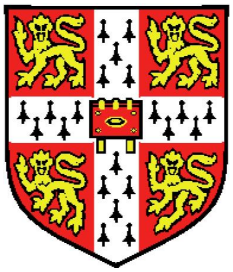
Social Network Support for Data Delivery Infrastructures



Nishanth Sastry

St. John's College

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy

2011

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This dissertation does not exceed the regulation length of 60 000 words, including tables and footnotes.

Social Network Support for Data Delivery Infrastructures

Summary

Network infrastructures often need to stage content so that it is accessible to consumers. The standard solution, deploying the content on a centralised server, can be inadequate in several situations.

Our thesis is that information encoded in social networks can be used to tailor content staging decisions to the user base and thereby build better data delivery infrastructures. This claim is supported by two case studies, which apply social information in challenging situations where traditional content staging is infeasible. Our approach works by examining empirical traces to identify relevant social properties, and then exploits them.

The first study looks at cost-effectively serving the "Long Tail" of rich-media user-generated content, which need to be staged close to viewers to control latency and jitter. Our traces show that a preference for the unpopular tail items often spreads virally and is localised to some part of the social network. Exploiting this, we propose Buzztraq, which decreases replication costs by selectively copying items to locations favoured by viral spread. We also design SpinThrift, which separates popular and unpopular content based on the relative proportion of viral accesses, and opportunistically spins down disks containing unpopular content, thereby saving energy.

The second study examines whether human face-to-face contacts can efficiently create paths over time between arbitrary users. Here, content is staged by spreading it through intermediate users until the destination is reached. Flooding every node minimises delivery times but is not scalable. We show that the human contact network is resilient to individual path failures, and for unicast paths, can efficiently approximate flooding in delivery time distribution simply by randomly sampling a handful of paths found by it. Multicast by contained flooding within a community is also efficient. However, connectivity relies on rare contacts and frequent contacts are often not useful for data delivery. Also, periods of similar duration could achieve different levels of connectivity; we devise a test to identify good periods. We finish by discussing how these properties influence routing algorithms.

Acknowledgements

First of all, I would like to thank Jon Crowcroft. For allowing me to discover my own thesis topic and approach, without any pressure. For opening up a whole host of collaboration opportunities. For quick and efficient feedback on my paper (and dissertation) drafts. For trademark-worthy names like Buzztraq and SpinThrift. One should be so lucky in their choice of supervisors.

I have had more than my fair share of excellent mentors. Karen Sollins took me under her wing at MIT during my first year. I thank her for making it possible for me to start my PhD at the other Cambridge, and for her gentle and caring support since then.

Discussions with collaborators and others have greatly shaped my thinking. D. Manjunath helped me with my first baby steps in using probabilistic arguments. Anthony Hylick pointed me towards storage subsystems, and this led to SpinThrift. An email exchange with Steve Hand and the wider netos group helped sharpen my thinking on Buzztraq. Discussions with Vito Latora led me to create and experiment with synthetic versions of the MIT and UCSD traces.

I have also benefited from lively and stimulating conversations on a wide range of topics with many of my FN07 officemates, past and present: Derek, Malte, Ganesh, Steve, Jisun, Mark, Henry, Chris, Stephen, Periklis, Theo, and frequent FN07 visitor Eiko. At MIT, I had Rob, Chintan and Ji in my office, and Mythili upstairs. Given the subject of this dissertation, it seems fitting to acknowledge that in a wider sense, the same purpose has been served by *my* social network support during these PhD years: Lava, Alex, Klaudia, Sriram, Alka, Amitabha, Mohita, Kiran, Lavanya, Vijay, Roopsha and Olga.

My wife, Viji, has played such a central role that I don't know where to begin. She is usually the first guinea pig on whom I try out my ideas. Having to explain to a non-computer scientist makes my thinking clearer and my ideas simpler. She suffers through unpolished versions of my talks. She stays up late during conference deadlines, for moral support. Once, she

even spell-checked my paper at 3 A.M. in the morning. At crucial junctures, she has completely taken over on the domestic front, allowing me to focus on work. In short, this dissertation is what it is because of (or inspite of) her. She is also responsible for ViVa, whom I need to thank separately for 1.5 years of pure fun.

Amma and Nana planted the seeds for this dissertation a long while ago, when they bought me the eminently readable *Chitra Kathegalu* (Picture Stories) and later (what must have been for them an expensive book), *The Children's Book of Questions and Answers*. I am forever in their debt. My sister Nikhila, five years younger, and therefore in need of teaching, was the first "student" I bossed over. That must have been the formative experience that set me off on an academic path.

Finally, my sincere thanks to my College, St. John's, for generously supporting me with a Benefactors' Scholarship, and to The Cambridge Philosophical Society, who awarded me a Research Studentship.

Contents

CHAPTER 1

Introduction

1.1 The content staging problem

In this dissertation, we are interested in supporting "eyeball" information flows, where the consumer or target of the flow is a human being. In many eyeball information flows, the producer or source of the flow is also human, but this is not essential. Examples of such flows include instant-messaging, VoIP conversations, email, or a user-generated video being streamed online.

A data communications network is only useful insofar as it is able to deliver content effectively to its consumers. Often, in eyeball information flows, the producer of the content and the consumers cannot be guaranteed to connect to the network infrastructure at the same time. In this situation, some intermediate infrastructure needs to play the key role of *staging the content* so that it becomes accessible to the consumer *asynchronously*.

A simple, well understood staging mechanism is to deploy the content on a centralised server. The server increases the availability of the content in the temporal dimension by acting as an always-on proxy for the producer. This kind of asynchronous access is enabled by Bulletin Board Services (BBSes), news servers, online video and photo sharing sites, etc.

However, a centralised server may be inadequate either because of special characteristics of the content, or because of inadequate support from

the network itself[1]. For example, certain content, like rich-media video streams, have strict delivery constraints and require tight control over latency and jitter. This may be difficult to achieve with a centralised server and a globally distributed consumer base, given the best-effort nature of the current Internet. To resolve this, the producer needs to be proxied in space as well as in the time dimension, by mirroring the content in different parts of the network. Specialised global replication infrastructures called Content Delivery Networks (CDNs) have been developed for this purpose.

Delay-tolerant networks like the Pocket Switched Network (PSN) [HCS+05] offer even lesser support and might not even guarantee any-any connectivity. In a PSN, content is opportunistically transmitted hop-by-hop over face-to-face contacts and connectivity is achieved over time through a sequence of social contacts. Clearly, each user involved as an intermediate node is mobile and can be thought of as increasing the availability of the content in a particular region of space-time where they are located, by acting as a proxy for the producer. Because of the extremely constrained connectivity, the coverage of individual proxies can be limited.

We can make the following observations from the above two scenarios:

1. As the number of dimensions increase and the coverage areas of individual proxies decrease, content staging becomes more expensive because the number of replicas required increases.

2. There is a natural trade-off between the number of replicas and the cost incurred. For instance, in PSNs, the best coverage (and minimum delivery time) is achieved by flooding to every possible intermediate node, but this also maximises the cost of delivery.

3. The cost of making replicas is amortised over the number of consumers who access the content. Thus, global replication via CDNs is cost-effective for popular, widely accessed content.

[1] A third reason is possible: a centralised server architecture might be unable to handle peak loads such as those due to flash crowds. Although this is usually handled in practice by replicating via CDNs, in theory, the server and/or network can be provisioned to handle the peak load. Both these solutions reduce to ones we have discussed here.

These observations lead us to the following problem, which is the subject of this dissertation:

> How do we stage content in a cost-effective manner, when the number of consumers is small and the cost of staging content is high?

Remark 1.1 (Generality). The question we have framed applies not only to the two examples discussed above (PSN and CDN), but also to other content staging mechanisms such as peer-to-peer networks. However, the solutions (as well as the precise specification of the problem) are intimately connected with the system being considered. Hence, this dissertation will focus on these two examples and develop independent solutions for each, as case studies of how to approach the problem in other cases.

Remark 1.2 (Problem formulation). The problem is deliberately vague and underspecified: How do we measure "cost-effectiveness"? When do we enter the desired regime of "small number" of consumers? How do we decide whether the cost of staging content is "high" in a given scenario? The answers to these depend on the case under study, but even without explicitly resolving the answers to these questions, the problem we have framed is usually understandable at the level of intuition, given a scenario.

For concreteness, we adopt the following conventions: We measure cost in terms of the number of replicas, and, if more detailed accounting is required, the resources consumed on individual replicas (e.g. energy incurred in storing and serving an item). Thus, in a CDN, the cost might be the number of regional replicas made. In a PSN, it could be measured as the number of intermediate nodes staging the content.

We *do not* discuss when we enter the desired regime of "small number" of consumers and "high costs". This could be decided based on a cost-benefit analysis on a case-by-case basis. However, given an item which is deemed to fall under our regime, we wish to develop techniques to deliver it without impacting upon performance.

In general terms, there are at least two options: The first is a "do

nothing" option, which incurs the minimum cost. In the CDN scenario, this could mean staging the content on a single replica (which boils down to using a centralised server). In the context of a PSN, this could mean, for example, involving no more than one intermediate node at each step[2]. A second option is one that involves the maximum cost: global replication in a CDN, or flooding all nodes in a PSN. We are looking to develop effective strategies that are less expensive than the second option, but offers better performance for the consumer than the first one.

Remark 1.3 (Designing for the worst case). The problem we have framed is essentially a worst-case scenario in which the cost incurred is high and per-item benefits are small. Many systems find it sufficient to optimise for the common case, and silently ignore or offer a degraded performance for the worst case. It should be emphasised that the handling of the common or normal case should never be compromised in order to accommodate a solution for the worst case; frequently it may be desirable to handle the normal case separately from the worst case, by developing entirely different strategies to handle each [Lam83].

The problem in context

We conclude this section by refining and adapting the generic problem statement above to our case study scenarios. For each scenario, we justify the need to treat the worst-case as importantly as the common case.

In the case of rich-media streaming, a small number of popular items often account for most of the accesses. Global replication via a CDN pays off for such content because the cost of replication is amortised over a large number of accesses. Much of the benefit of a CDN can be cost-effectively obtained by optimising for this common case, replicating only the popular items, and providing a degraded service for the others (say by using a centralised server).

[2]Direct delivery when the source meets destination is cheaper, but it does not involve the PSN.

However, some content catalogues consist mainly or entirely of so-called *user-generated content*. Virtually anyone may add new items. This leads to large numbers of items and makes it difficult to curate. It is not possible to tell whether an uncurated item is important simply by its popularity– replicating only popular items would be unsatisfactory if the unpopular items (in terms of number of accesses) represent important niche interests. §3.2 and §3.3 discuss other reasons for why it is important to replicate unpopular user-generated content. For instance, although such "Long Tail" items may individually not have many consumers, collectively, a large portion of the user base expresses a preference for tail items. These considerations lead us to the first case study problem:

Problem 1.1. *How can we cost-effectively replicate items from the Long Tail of rich-media user-generated content close to their viewers?*

PSNs offer a different trade-off between replication and communication. In this dissertation, we are mainly interested in how the PSN enables unicast. Unicast flows, by definition, have only one destination node, so there is never an opportunity to amortise costs across multiple consumers. However, it may be easier (cheaper in terms of intermediate nodes required) to connect some node pairs than others. Even if communication between such nodes represents the common case usage of a given PSN, it is less interesting because nodes which need to communicate often could likely organise their contacts to meet directly. The need for PSN support is greater when the nodes are only weakly connected in a social sense. To measure the utility of the PSN, we need to study whether and how well it can connect any node to any other arbitrary node. Therefore, our second case study focuses on the ability of human social contacts to acheive any-any connectivity:

Problem 1.2. *What properties of human social contacts enable effective data delivery between arbitrary pairs of nodes in a Pocket Switched Network?*

1.2 Thesis and its substantiation

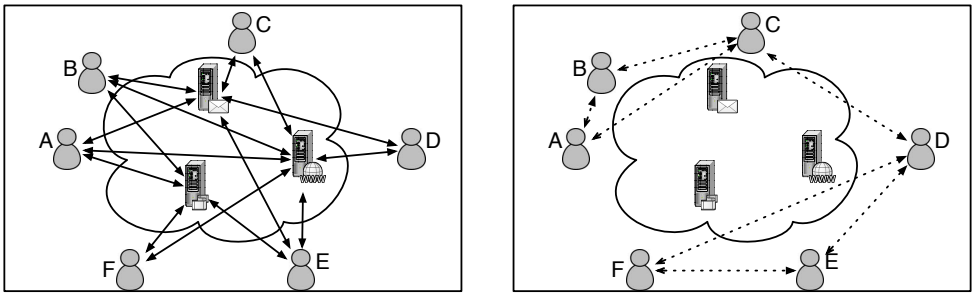
We approach the above problems by observing that in contrast with the case when an item is popular and widely accessed, an item which has few consumers does not need to be staged everywhere, and at all times. Therefore, if we can develop a predictor for where and when accesses occur, we can save costs by selectively staging the content.

> Our thesis is that information encoded in social networks can be used to tailor content staging decisions to the user base and thereby build better data delivery infrastructures.

We substantiate this thesis by applying social information-based solutions to the two content staging problems posed above. The solutions were developed independently of each other, and are highly specific to their environments. However, they share the same **high-level approach**: We first examine how the participants in the environment—the consumers and other human actors, if any—are related to each other. This is captured implicitly or explicitly as a *social graph*. The social graph represents each human actor as a node and the relationships between the actors as links or edges between the corresponding nodes. The structure of this graph is then used to develop a cost-effective content staging strategy. Our high-level approach can be seen with the following toy example.

1.2.1 Staging content based on social properties

To understand how the structure of the social graph can inform content staging, consider the following toy example, which is pictorially depicted in Figure 1.1. Suppose globally distributed users are asynchronously producing and consuming content by staging them on intermediate servers (Figure 1.1 (a)). We can draw graphs with the users as nodes, and edges depicting various kinds of relationships between them (Figure 1.1 (b)).



(a) Actual information flow (b) Logical information flow

Figure 1.1: *A toy example illustrating the utility of elementary social network analysis for content staging. Left: Actual information flows between six users. Each flow is first staged on one of three servers by the producer, and later fetched by the consumer. Right: Logical information flows, depicting (possible) interactions between users as a social graph. Different graphs can be drawn for different kinds of interactions (see text for interpretation).*

Such graphs can be used to tailor content delivery infrastructures in interesting ways. We will illustrate this by assigning different interpretations to the edges in Figure 1.1 (b). Suppose an edge $X \rightarrow Y$ is drawn whenever a node Y consumes content produced by node X, and the relative positions of nodes are indicative of their network distance. The structure of the graph suggests a natural partition of nodes into two clusters: A, B and C on the one hand, and D, E and F on the other[3]. This suggests that traffic across the network can be decreased by deploying a separate server for each cluster of users instead of deploying centralised servers that cater for all users.

Now consider the case when edges connect nodes which are colocated in time (i.e. edges connect nodes that tend to stay connected to the network during the same time periods) or space (e.g. edges connect nodes within the same Internet Service Provider or ISP). This kind of graph can be useful in deciding whether to replace the centralised servers with a peer-to-peer (P2P) content staging infrastructure. For instance, the connectedness of the graph under edges induced by temporal colocation implies that any peer can successfully reach any other peer either directly or by staging the

[3]We focus on providing an intuitive understanding here, but this can be formally shown, for example, by proving that the min-cut corresponds to the edge $C \leftrightarrow D$.

content on a series of intermediate nodes such that successive nodes are temporally colocated. Next, suppose that in Figure 1.1 (b), the nodes A, B and C are with one ISP, D, E and F are with a second ISP, and C, D are multi-homed and connect to each other via a third ISP. By forming an overlay network amongst themselves, all the six nodes can reach each other without ever causing cross-ISP traffic, and therefore might be able to avoid the P2P traffic shaping that some ISPs implement at their borders in order to decrease their transit payments.

Essentially, this approach can be thought of as performing a social network analysis on the population of consumers and other participants in the content delivery process, with a view to deduce properties that can be useful for making content staging decisions. The same approach could be extended to other systems-level design choices.

Traditionally, social network analysis has been used not as a tool to make systems-level decisions, but as an approach to understand societies and organisations [Sco00, WF95]. Classical applications include identifying important players in teams and organisations, social affiliations and groupings, susceptibility to the spread of information and so on. For instance, targeted advertising strategies can be devised if the edges are drawn to be indicators of similar consumption patterns. As a trivial example, if product adoption is known to spread mainly along the links shown in Figure 1.1 (b), then it is easy to see that adoption by nodes C and D is important for global adoption in the entire network.

1.2.2 On deriving useful social properties from trace-driven social network analysis

In both case studies, we derive properties of interest by examining empirical traces. Unlike traditional systems where it is usually possible to understand the design at an intuitive level, systems which rely on social network support often need to make use of non-obvious social properties. Trace-driven analysis serves the important purpose of grounding such designs in reality, and helps provide a better understanding of why the system works.

Social networks have many well-known properties and it may be possible to exploit these in a given system. But, as demonstrated in this dissertation, we may require new social properties. We sketch some thoughts in this subsection on how to derive new and useful social properties starting from empirical traces of a system's users.

A basic requirement for social properties is that they should be valid beyond the examples from which they are derived. Clearly, some of the examples in §1.2.1 recommend actions that are specific to the nodes being considered. Similarly, the designs described in the case studies make content staging decisions based on example networks drawn from real-world traces. In order to lift up individual node- or link-level observations to the level of a property that can be reused beyond the networks described by the traces, the case studies first test to see whether it holds for a class of nodes (or links) rather than individual nodes. Additionally, to establish generality, it would be desirable for the property to hold in multiple networks; each case study uses at least two different networks and establishes that the property holds in both. Note that it is perfectly acceptable if a property does not hold for *all* the nodes in a class—some of the properties are stochastic in nature, and only hold in distribution. From a systems viewpoint, it would still be possible to use such a property if the expected benefits outweigh the penalties of being wrong.

Unfortunately, there seem to be no easy or failsafe methods to identify all the properties that might be useful in a given context or system. However, during the course of our work, we have come up with two heuristic techniques that might be of independent interest:

1. The first heuristic is to use known social theories as a starting point. In §3.3.6, we attempted to verify whether theories of viral propagation popularised by Malcolm Gladwell's Tipping Point [Gla02] hold in our context. As demonstrated in this case, there is no guarantee that such generic theories will be valid outside of their initial contexts, but they can still yield useful insights, or new ways to look at the system under consideration.

2. A second technique is used in §4.3. Here we create a synthetic trace from the original trace by disrupting the property we wish to study. By studying the differences between the two traces, we establish the effects of the property. This method is inspired by similar techniques used in other fields. For instance, the functions of proteins in biological network pathways are routinely studied by disrupting the protein, and studying its effect on the pathway.

1.2.3 Case Studies

Chapter 3 and Chapter 4 give a detailed and self-contained account of how social network support can address the problems posed in §1.1. Here we give a brief overview of the two case studies in the light of the approach developed above.

Case study I: staging the long tail of user-generated content

As described in §1.1, serving rich-media user generated content such as streaming videos is compounded by three factors: *a*) They require large storage and typically need to be served from disk. *b*) They need tight control over latency and jitter and need to be staged on replicas close to viewers. *c*) The majority are so-called "Long Tail" items which individually receive few accesses but collectively are important for a large fraction of the user base.

To mitigate these issues, we rely on the observation made at the beginning of this section (§1.2): an item which has few consumers does not need to be staged everywhere, and at all times. First, let us consider *when* to stage content. Ideally, an item only needs to be staged when it needs to be accessed by a user. Since each Long Tail item individually receives few accesses, this leads us to formulate a novel energy-saving content staging strategy that stores items on disk at low-power mode, and *reactively stages them when there is an access*, by bringing the disk back to full power.

Note that for a disk to be in low-power mode, *none* of the items on the disk should receive accesses. Therefore, to realise the energy savings, we

need to segregate the popular content from the unpopular. By examining traces of user-generated content, we establish that interest in popular items is seeded independently in multiple parts of the social network and spreads non-virally, whereas interest in the unpopular "Long Tail" remains localised to some part of the social network, spreading virally. Based on this, we develop SpinThrift, which uses the relative proportion of viral and non-viral accesses to segregate popular and unpopular content on separate disks and saves energy by spinning down disks containing unpopular items when possible.

Next, we focus on *where* to stage the content. Observe that ideally, items only need to be staged in regions where there are users. This works to the advantage of items in the Long Tail, which do not have many users– we can save costs by selectively replicating only to regions where there are users. However, in order to effectively make use of this strategy, we need to *predict* which regions will have users. Here, we use the earlier observation that interest in Long Tail items spreads virally, from friend to friend in the social network. Hence, we can expect that future accesses will come predominantly from regions where *friends of previous users* are located. This forms the basis of our selective replication scheme, Buzztraq, which replicates a video to the top-k regions where friends of previous users are located.

Case study II: staging content in pocket switched networks

In Pocket Switched Networks, paths are opportunistically created over time by staging content on a series of intermediate nodes. Content is transferred between nodes during human social contacts. We are interested in understanding how properties of the human contact network affect data delivery. We study the achievable performance of the contact network in terms of the fraction of data delivered (delivery ratio), as well as the time to delivery.

Our primary interest is the case of unicast, where a single source node or producer needs to be connected with a single destination node or consumer over time. Note that even though multiple paths may form between a

given pair of nodes, only one path needs to be completed for unicast data to be delivered between them. Thus, as in the previous case study, we can save costs by not staging content everywhere and at all times. As discussed in §1.1, each intermediate node can be thought of as staging the content in a particular region of space-time; hence we can save costs by carefully choosing intermediate nodes based on how this choice affects the performance of the network. Equivalently, we can save costs by being selective about which contacts are used to spread the content on more nodes.

In effect, we can save costs by having PSN routing algorithms take into account the properties of human contact networks. Below we summarise our findings on how the human contact network shapes data delivery and how these can be used to make routing decisions.

In our empirical traces, we find that the delivery ratio is determined by the distribution of contact occurrences. Most node pairs meet rarely (fewer than ten times), whereas a few node pairs meet much more frequently (hundreds of times). The network's ability to connect two nodes over time depends crucially on the rare contacts. In contrast, frequent contacts often occur without there being new data to exchange, even when flooding is the route selection strategy used. Further, we find that for arbitrary source-destination pairs, routing only on rare edges does not significantly affect delivery time distribution. These findings suggest developing new routing algorithms that favour rare edges over frequent ones. However, nodes which are well connected with each other may benefit from using the more frequently occurring edges. We note that our demonstration of the importance of rare contacts provides empirical evidence for Granovetter's celebrated theory on the role of weak ties in information diffusion [Gra73].

Next, we examine time windows of fixed duration and find that there is a significant variation in the achieved delivery ratio. We discover that in time windows in which a large fraction of data gets delivered, rather than all nodes being able to uniformly reach each other with the same efficiency, there is usually a large *clique* of nodes that have 100% reachability among themselves. We show how to identify such time windows and the nodes involved in the clique by computing a clustering co-efficient on the contact

graph. This can be used in multiple ways: For instance, nodes with higher clustering co-efficient could be preferred for routing. When the clustering co-efficient is low and a low delivery ratio is predicted, the routing algorithm could be re-tuned to aggressively send multiple copies; alternately, data which need a better delivery guarantee could be sent through other, more expensive means (e.g. using a satellite connection instead of the PSN).

Finally, we examine the effect of path failures on data delivery. The best delivery times are achieved by flooding at every contact opportunity. Over time, this creates a number of paths between each source and destination. Unless all these paths fail, data will eventually be delivered. However, since there is a wide variation in the delivery time distributions of the quickest and slowest paths between node-pairs, time to delivery is expected to increase if some of the quicker paths fail. To understand this, we examine the impact of failing a randomly chosen subset of the paths found by flooding between each sender and destination and find that the delivery time distribution is remarkably resilient to path failures.

Specifically, we study two failure modes among paths found by flooding: The first, proportional flooding, examines delivery times when only a fixed fraction μ of paths between a sender and destination can be used. The second, k-copy flooding, assumes that at most a fixed number $k > 1$ of the paths between a node-pair survive. In both cases, we find that the delivery time distribution of the quickest paths can be closely approximated with relatively small μ and k. It is shown that a constant increase in μ (respectively, k) brings the delivery time distribution of proportional (k-copy) flooding exponentially closer to the delivery time distribution of the quickest paths found by flooding.

The success of $k > 1$-copy flooding can also provide a loose motivation for heuristics-based routing algorithms that explore multiple paths simultaneously. It also helps explain simulation studies [HXL+09] which have shown that delivery ratio achieved by a given time is largely independent of altruism levels, or the propensity of nodes to carry other people's data. We also find that randomised sampling of paths could lead to better resilience and load balancing properties.

1.3 List of publications

During the course of my PhD, I had the following nine publications. Chapter 3 is based on [SYC09, SC10]. Chapter 4 is based on [SMSC11] and extracts from the book chapter version [SH11].

Part of [SMSC11] was originally published as a conference paper in [SSC09]. [SC10] extends a version published as an invited paper in [SHC10].

[SCS07] Nishanth Sastry, Jon Crowcroft and Karen Sollins. Architecting Citywide Ubiquitous Wi-Fi Access. In *Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets-VI)*, Atlanta, GA, USA, November 2007.

[Sas07] Nishanth Sastry. Folksonomy-based reasoning in opportunistic networks. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–2, New York, NY, USA, December 2007. ACM. (PhD Workshop).

[SYC09] Nishanth Sastry, Eiko Yoneki, and Jon Crowcroft. Buzztraq: Predicting geographical access patterns of social cascades using social networks. In *Proceedings of Eurosys Social Network Systems Workshop*, Nuremberg, Germany, February 2009.

[SSC09] Nishanth Sastry, Karen Sollins, and Jon Crowcroft. Delivery properties of human social networks. In *Proceedings of the IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009. (Miniconference).

[HS09] Pan Hui and Nishanth Sastry. Real world routing using virtual world information. In *Proceedings of the IEEE SocialCom Workshop on Leveraging Social Patterns for Privacy, Security, and Network Architectures (SP4SPNA09)*, Vancouver, Canada, August 2009.

[SHC10] Nishanth Sastry, Anthony Hylick, and Jon Crowcroft. SpinThrift: Saving energy in viral workloads. In *Proceedings of the International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, India, January 2010. (Invited paper).

[**SC10**] Nishanth Sastry and Jon Crowcroft. SpinThrift: Saving energy in viral workloads. In *Proceedings of the first ACM SIGCOMM workshop on Green networking*, New Delhi, India, August 2010.

[**SH11**] Nishanth Sastry and Pan Hui. Path Formation in Human Contact Networks. In My T. Thai and Panos Pardalos, editors, *Handbook of Optimization in Complex Networks*. Springer-Verlag, 2011. (Invited Book Chapter. In press.)

[**SMSC11**] Nishanth Sastry, D. Manjunath, Karen Sollins, and Jon Crowcroft. Data delivery properties of human contact networks. *IEEE Transactions on Mobile Computing*, 2011. (To appear. Early access version available from http://dx.doi.org/10.1109/TMC.2010.225.)

1.4 Contributions and chapter outlines

In summary, this dissertation demonstrates how to exploit social structures for data delivery. Our contributions fall into three categories:

1. The codification of an approach that applies social network information to make content-staging and other systems-level decisions. This chapter, introduced and motivated the approach. In Chapter 5 we conclude by reflecting on its promise and limitations.

2. The development of solutions and recommendations for content staging in two real-world scenarios, as case studies of applying the above approach (Chapters 3 and 4).

3. The derivation of new social network properties from empirical traces. (In §3.3 and Chapter 4). §1.2.2 gives hints on deriving new properties.

Chapter 2 gives an overview of existing social networks, social network properties and social network-based systems, and discusses how the present dissertation fits in (§2.5).

CHAPTER 2

Social Networks: an overview

According to boyd, a primary and essential feature of a social network is a "machine-accessible articulation of users' personal relationships" to other users [bE08]. The web of personal relationships or "Friendships" is typically viewed abstractly as a social network graph or "social graph" with the site's users as nodes, and each "Friendship" corresponding to an edge between two nodes.

In this chapter, we attempt to give a broad overview of social networks based on the above definition. Our emphasis is on providing an understanding of social networks as a useful addition to the standard tool-box of techniques used by systems designers.

Chapter layout

§2.1 discusses different varieties of social networks and their applicability. §2.2 highlights properties which are thought to be common features of social networks and how they can be exploited. §2.3 discusses emerging opinions in the literature which present evidence against some deeply and widely held beliefs about social networks. §2.4 provides examples of a few systems which are based on social networks or properties. §2.5 draws lessons from the research we survey, and concludes with potentials for future work. In

particular, we discuss how this dissertation differs from and contributes to the growing literature.

2.1 Different varieties of social networks

The primary reason we are interested in social graphs from a systems standpoint is that such graphs are thought to capture a reasonably large fraction of real offline relationships. However, there are several different social networks, and underneath the simple unifying framework of the social graph, there are deep semantic and structural differences between them. Different networks may be suitable for different applications. Below, we first discuss a framework for understanding these differences in terms of link semantics and structure (§2.1.1 and §2.1.2), and highlight the implications for systems designers. Then we provide examples of different social networks which have been analysed in the literature (§2.1.3).

2.1.1 Link semantics

At a semantic level, differences arise between different social networks because they capture different kinds of relationships between their users. Facebook[1], for example, tries to recreate the friendships of individuals, whereas LinkedIn[2] attempts to capture their professional contacts. Still others, such as the educational social network, Rafiki[3], encourage and foster relationships between teachers and students across the world. Geospatial social networks like Gowalla[4] and foursquare[5] engender connections between people with ties to similar places. Interest-based social networks like the epinions online community[6], last.fm social network[7] etc. tend to form links between people who like the same items or the same *kind* of items.

[1] http://www.facebook.com
[2] http://www.linkedin.com
[3] http://rafi.ki
[4] http://www.gowalla.com
[5] http://www.foursquare.com
[6] http://www.epinions.com
[7] http://www.last.fm

Implication: Clearly different social networks are suitable for different needs. For example, in Chapter 3, we use interest-based social networks to capture affinities between people and content that interests them. In Chapter 4 we use an entirely different network, based on human contact patterns, and find opportunities to deliver data over a series of such contacts.

2.1.2 Link structure

At a structural level, the friendship edges can be drawn in a variety of ways. Most social networking sites require users to explicitly declare their friends. Some, such as Facebook and LinkedIn, have bidirectional (or equivalently, undirected) edges which are formed only when both parties confirm their friendship to the site. Others, such as Twitter[8], have unidirectional (directed) edges in which one user can "follow" another user without the other user's permission. Unidirectional links are particularly suited for interest-based social networks where reciprocation is not expected or required. For instance, Digg[9] and vimeo[10] allow users to add other users as contacts and unilaterally subscribe to their content uploads and recommendations.

Social graphs can also be inferred automatically (implicitly) as a side-effect of user actions. Human contact networks such as the ones we examine in Chapter 4 draw edges between people who are in close proximity to each other [EP, UCS04]. Other examples include email networks, where edges are drawn as a result of email interaction. Note that automatically inferred links still meet boyd's generic definition of social networks [bE08] because the links are still explicitly articulated in a machine accessible manner.

Implication: Bidirectional links are more trustworthy because they are more difficult to make: spammers or other nodes can indiscriminately make unidirectional links to several unrelated nodes whereas bidirectional links have to be vetted by both parties. However, the directionality of unidirectional links can be used to create a reputation mechanism similar to PageRank [PBMW99]. TunkRank [Tun09], developed as a direct analogue

[8] www.twitter.com
[9] http://www.digg.com
[10] http://www.vimeo.com

of PageRank, measures user influence in Twitter based solely on link structure. TwitterRank [WLJH10] gives a topic-sensitive ranking for Twitter users, using a random surfer model similar to PageRank.

Both explicitly declared and implicitly inferred social graphs can suffer from irregularities such as missing or spurious links. In explicitly declared social networks, these errors arise due to user actions. When links are automatically (implicitly) inferred, the irregularities could arise due to technology limitations. For example, one way to infer a human contact network is to use Bluetooth device discovery to identify users who are close to each other, and drawing a link between them. This technique was used to collect the MIT/Reality data set[EP]. However, to conserve batteries, device discovery was only initiated once every five minutes. Therefore contacts shorter than this interval could be missed. Similarly, the ten metre range of Bluetooth along with the fact that it can penetrate some types of walls, means that spurious links can be created between people who are not physically proximate.

2.1.3 Examples of social networks

In recent years, there has been an explosion of interest in social networking sites such as Facebook, Twitter and LinkedIn. Whilst there have been older avenues for online social interaction, such as Ward Christensen's Computerised Bulletin Board System or CBBS (1978) and Richard Bartle and Roy Trubshaw's original Multi-User Dungeon or MUD (1977)[11], the current social networking phenomenon is distinguished by the massive scale of its adoption in the general population. For example, Facebook claims to have over 500 million active members, over 50% of whom log on in any given day [Fac]. [LH08] claims the record for the largest network analysed: 30 billion instant messaging conversations among 180 million users over the course of a month. Thus, the modern social networking phenomenon has

[11]For a brief and vividly sketched history of Social Networking, see The Cartoon History of Social Networking [Lon11]. Other brief accounts with similar material include [Nic09, Sim09, Bia11].

generated extremely large social graphs, which are sometimes a challenge to analyse.

Below we discuss different kinds of networks which have been analysed by researchers. Many of these have been obtained by brute-force crawling. Most Web 2.0 sites expose well-known APIs which provide the data in a structured form, but it may be time-consuming to obtain data at large scales because the API calls are typically rate limited. Prior to Web 2.0, it was not common to explicitly record and store the graph of social relationships. Thus, older social graphs, including some about offline social relationships had to be inferred from trails of interactions.

Offline social relationships

A number of offline social networks have been studied over the years, mostly by sociologists and physicists from the complex networks community. These include the network of movie actors [BA99], scientific collaboration networks [New01] and sexual relationships [LEA+01, HJ06]. Information about some offline relationships may be found more easily online: [HEL04] studies how social communities evolve over time by studying an Internet dating community.

Pre Web 2.0 online communities

Before Web 2.0, online communities typically did not store an explicit social graph. Thus many graphs in older online communities are inferred relationships based on records of interactions. Cobot gathered social statistics from the LambdaMOO MUD [IKK+00]. Even earlier, [SW93] looked at the social graph induced by email conversations. [LH08] examined the network of instant messaging conversations and found strong evidence for homophily.

ReferralWeb [KSS97] synthesized a social network by mining a variety of public sources including home pages, co-authorships and citations, net news archives and organisation charts, and used this to effectively locate experts for different topics. This technique may be useful in other systems as well: In many applications, including at the systems level, it may be relevant

or useful to capture multiple modes of relationships or interaction. For example, email and instant messaging, when combined with phone call logs, could yield a more complete picture of interaction between two individuals, than can be seen by examining each mode of interaction separately.

One early example of explicitly stored social graphs comes from webs of trust. Prominent early webs of trust include PGP [Sta95] and Advogato [Lev00]. However, these are usually stored in a decentralised fashion, and may be difficult to capture completely. More centralised records of trust and reputation relationships may be found on online shopping sites, where trust webs have been exploited to enhance confidence in sellers and items, as well as to provide useful recommendations. Many such networks have been examined for inferring relationships between users and their consumption patterns. Examples include Epinions [RD02], Amazon [LAH07] and Overstock [SWB+08].

Modern social networks

One of the earliest studies looking at recent Web 2.0 online social networks was [MMG+07]. It showed that online social networks are scale-free, small world networks and have a power-law degree distribution. It also identified a form of reciprocity: in-degree distributions match out-degrees. [FLW08] studied a chinese social network (Xiaonei, now known as Renren[12]) and a blog network and showed that social networks are assortative whereas blogs are dissortative. Other prominent Web 2.0 communities analysed include the Flickr photo sharing community [CMG09a] and video sharing on YouTube [CKR+09].

Some networks have received considerable attention because of their popularity or importance. We discuss a sample of studies for three of the most common networks on which people interact: Facebook, Twitter and the network induced by mobile phone calls.

Facebook: Apart from basic topological properties, various different social *processes* have been studied on Facebook. For example, [SRML09]

[12] http://renren.com

show that information contagion, in terms of people "liking" pages on Facebook, starts independently as local contagions, and becomes a global contagion when different local contagions meet. Similarly, [WBS+09, VMCG09] showed that the network of nodes with interactions between them is much more sparse than the graph of declared relationships. We discuss this further in §2.3.1, in relation to the trustworthiness of interaction-based links.

Twitter: [JSFT07] discusses early uses of Twitter. [KGA08] offers an updated view and offers a classification of users. [CHBG10] discusses three different measures of user influence on Twitter. [KLPM10] performed a large-scale study of Twitter and found low levels of reciprocity, unlike other social networks. Instead they found that after the first retweet, most tweets reach around 1000 people, regardless of where it started; and that URLs are mostly news URLs. Based on these findings, they proposed an alternate view of Twitter as a news media site.

Mobile phone call graphs: Unlike Facebook or Twitter, links on the mobile phone call graph have to be inferred from calls made. Large scale data on phone calls is not easily available, nor can it be mined by crawling data sources. Hence, this area has been dominated by the Barabasi Lab, who have managed to obtain country-scale data about mobile phone calls. In agreement with similar results we obtain (see §4.3), [OSH+07] finds that the mobile phone call graph falls apart when weak ties are broken, and that this reliance on weak ties slows down information diffusion. [PBV07] finds that small groups persist if their membership does not change, whereas large groups persist if members are allowed to break away. [GHB08] examines human mobility patterns through cell tower associations and finds that they are highly regular. Finally, [SMS+08] studies the distributions of several per-customer scalars in mobile phone networks (namely, number of phone calls, total talk time, number of distinct calling partners) and finds that they fit a Double Pareto Log-normal distribution.

2.1.4 Synthetic social networks

The data of many social networks is not easily obtained. Alternately, the terms of service of a social network may not allow the offline use and storage of their users' data. Thus, during development of social network-based systems, it may be desirable to use synthetic but realistic social graphs. Recent results show how to generate synthetic graphs for Facebook [SCW+10] and Twitter [EYR11]. Older methods for synthetic social graphs were built upon generative models which capture particular properties of social networks. For example, Barabasi's Preferential Attachment model captures scale-freeness and power-law degree distributions (§2.2.1). Graphs generated according to the Kleinberg or Watts-Strogatz model capture the small world phenomenon (§2.2.2). See [CF06] for a survey of different generators. More recent generators include ones based on Kronecker multiplication, which capture densification laws and shrinking diameters [LF07].

2.2 Properties of social networks

Although the complete social graph may never be articulated online in its entirety, it is thought to possess certain characteristic features, some of which have even become catch-phrases in popular culture. Many of these properties are not unique to social networks; rather they are common to other "complex networks". It should be noted that graph properties can evolve over time. For instance, it has been observed that the number of edges grows superlinearly in the number of nodes, creating denser graphs, and the diameter of the graph often shrinks over time [LKF07].

2.2.1 Properties of node degrees

Social graphs are thought to be *sparse*: most node pairs do not have direct edges between them. The degree of nodes, the number of other nodes to which a node is connected, typically has a *right-skewed distribution*: the majority of nodes have a low degree but a few nodes have a disproportion-

ately large degree. The precise distribution often follows a power-law or exponential form.

Social graphs differ from other networks in having positive correlations between node degrees [New02]. In other words, nodes with high degree tend to connect to other nodes with high degrees, and nodes with low degree tend to interconnect among themselves. This property is known as *assortativity*. By contrast, biological and technological complex networks tend to be dissortative, i.e., there is a negative correlation of node degrees, with high degree nodes preferentially connecting to low degree ones. Systems designers should note that assortative networks percolate more easily than dissortative networks and are also more resilient to targeted attacks which remove high-degree nodes [New03].

2.2.2 Transitivity of friendships and small worlds

One finds a larger than expected (when compared with random graphs) number of triangles, or triads, in the social graph[13]. In sociology, this property is often termed *network transitivity*, because it implies that friendship is transitive: friends of friends are also likely to be friends.

Triangles are the smallest possible cliques (3-cliques) in a graph. Therefore, the excess of triangles can also be seen as evidence of "cliquishness" in a local neighbourhood. Each edge in a triangle is a *short-range connection* between two nodes which are already close to each other (because of the two hop path through the third node in the triangle). Hence, in complex networks literature, this property is usually referred to as *clustering*, in analogy with regular lattices, where edges are formed between nodes which are close to each other.

Despite the lattice-like clustering and the large number of short-range connections, several complex networks, including social networks have been

[13]Here we restrict our focus to triads. However, it has been extremely useful to generalise this to other local patterns which may occur in graphs, including signed versions. Such patterns, termed as network motifs [MSOI+02], have been useful in classifying different networks. For instance, a study of the relative abundance of different kinds of triads shows that social networks and the WWW may belong to the same "superfamily" of networks [MIK+04].

shown to have a small characteristic path length and small diameter, like random graphs. Popular culture knows this as the concept of "six degrees of separation", after Stanley Milgram's famous experiment [TM69], and the phenomenon of short paths has been called the *small world effect*.

In a seminal paper [WS98], Watts and Strogatz showed how complex networks reconcile clustering at a local level (which leads to long paths in lattices) with the global property of small paths, by rewiring a lattice structure on a ring to include a few random long-range edges. Even if there are only a few rewired edges, they serve as global short-cuts and decrease the path lengths for all nodes. In the Watts-Strogatz model, social and other small world networks are seen as a middle ground between a totally ordered system (a lattice, where edges are formed according to a geometric pattern) and a completely disordered system (a random graph, where edges connect randomly chosen nodes). An important systems-level implication of Watts and Strogatz is that signals as well as infections can propagate at high speeds through such networks.

It should be noted that even though short paths exist, it may be impossible to find them using decentralised or local search schemes unless certain conditions are satisfied. Kleinberg [Kle00] considered a graph with short-range links created according to a lattice-based model (similar to the Watts-Strogatz model), but generalised the probability of a long-range link between nodes to decay according as $r^{-\alpha}$, where r is the manhattan distance between nodes and α is a contant (In contrast, Watts-Strogatz rewires uniformly at random). In the Kleinberg model, a decentralised greedy search finds its targets, but only if $\alpha = d$, the dimension of the lattice. Interestingly, the success of Milgram's experiment suggests that social networks must be *navigable*, i.e., individual nodes have *demonstrated* the ability to find routes to other nodes through local search mechanisms. Clearly, navigability can aid a large number of applications and there has been a great deal of interest in devising routing schemes for small world networks [Kle06]. One promising recent approach is to use an embedding of the network in an appropriate hidden metric space [BKC08].

2.2.3 Community structures

Communities are subsets of nodes which are more "densely" connected to each other than to other nodes in the network. Community structures are a fundamental ingredient of social networks. Newman and Park [NP03] showed that both transitivity of friendships (clustering) and assortativity can be explained by a simple model that considers social networks as arising from the projection of a bipartite graph of nodes and their affiliations to one or more communities.

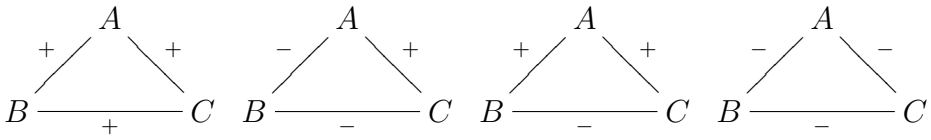
Unfortunately, there is no clear consensus on how to define a community, and various measures have been proposed [WF95, GN02, FLGC02, NG04, RCC+04]. [New04, DDGDA05, FC07] survey different algorithms which have been proposed based on the above measures. While the quality of the different algorithms, in terms of meaningful communities detected, can vary from case to case, most of them are computationally intensive and do not scale to large networks. The Louvain method [BGLL08] is a fast algorithm based on Newman and Girvan's modularity measure [NG04] that has good scalability properties.

Practitioners should be aware that the Louvain method (and other modularity-based algorithms) can produce inconsistent results depending on node orderings [KCE+09]. [LLM10] compares different objective functions for defining communities and the quality of communities found by some common algorithms. Using graph conductance as a measure, [LLDM08] shows that at small scales (≈ 100 nodes), tight communities can be found (i.e., the best communities of small sizes are separated from the larger graph by a small conductance cut), but at larger scales, the communities "blend" into the larger network (i.e., the community's cut has large conductance).

2.2.4 Structural balance

While most social networks only capture positive relationships or friendships, it is also possible to add negative or antagonistic relationship edges. For instance, users in the Slashdot[14] community can mark each other as

[14] http://www.slashdot.org



(a) balanced (b) balanced (c) unbalanced (d) unbalanced

Figure 2.1: *Possible triad configurations in a signed network, and their balance statuses. (a) This represents three nodes who are mutual friends. This is stable, and can be seen as a case of "a friend of my friend is also my friend." (b) This represents the case when two users have a mutual enemy, and is also stable. This is a case of "enemy of my enemy is my friend.". The other two labelings introduce psychological "stress" or "instability". In (c), A's friends B and C are enemies. This would cause A to persuade B and C to become friends, reducing to case (a), or else, to side with one of them, reducing to case (b). (d) has three nodes who are mutual enemies. This is also unstable, because there is a tendency for one pair to gang up against the other. However, this tendency is lesser than the stress induced in case (c); hence weak structural balance considers this to be balanced.*

friend or foe [KLB09]. Users in Essembly[15], an online political forum, can label each other as friends, allies or nemeses [HWSB08]. Guha et al. [GKRT04] were the first to examine *signed relationships* in online social networks, when they considered trust and distrust in the Epinions community.

Unlike positive relationships, the presence of antagonistic or negative relationships can induce stresses in the social graph, which make certain configurations of relationships unstable. The theory of *structural balance* governs the different relationships possible. There are two variants. The structural balance property originally proposed by Heider [Hei46] holds if in every triad, either all edges are positive, or exactly one edge is positive. Davis [Dav67] proposed a theory of weak structural balance which allowed all combinations except for a triad with two positive and one negative relation. Figure 2.1 shows the possible triads and the social psychology-based motivation for their classification as stable or unstable.

[15] `www.essembly.com`. Site has been down since May 2010, according to Wikipedia (`http://en.wikipedia.org/wiki/Essembly`. Wikipedia URL last accessed on March 31, 2011.)

A recent result [SLT10] finds more support for the weaker formulation of structural balance in multi-player games. Another recent result [LHK10] suggests that signed networks with *directed edges* are better represented by a theory of status rather than structural balance. Structural balance differs from status by being transitive: If A gives higher status to B with a positive directed link, and similarly, B gives higher status to C, status theory predicts that a link from C to A, should one exist, would be negative, whereas balance theory predicts a positive link. Finally, we note that while this discussion has focused on triads, structural balance theory can be extended to other patterns; the extension requires an even number of negative edges in every loop.

Structural balance can be useful in large systems because the constraints it imposes on local structure leads to very simple structures at the global scale. Harary [Har53] showed that in a complete graph, Heider's structural balance property results in graphs which can be clustered into two partitions, such that links within each partition are positive and the links across the partitions are negative. With weak structural balance, we can still achieve cohesive partitions with positive intra-partition links and negative inter-partition links, but there may be more than two clusters of nodes. Graphs which can be partitioned in this way are said to be *clusterable*.

2.2.5 Homophily, segregation and influence

Homophily, the tendency of people to form links with other people having similar attributes as them, is a powerful driver for social link formation [MSLC01]. Studies have shown that rising similarity between two individuals is an indicator of future interaction [CCH+08]. Systems can use this finding to optimise for links that are about to form.

However, homophily can also lead to segregation. The Schelling model shows that global patterns of segregation can occur from homophily at the local level, even if no individual actively seeks segregation [Sch72]. Thus, homophily could be used to effectively partition data.

Recent results indicate that peer influence in product adoption (i.e., vi-

ral marketing) is greatly overestimated, and that homophily can explain more than 50% of perceived behavioural contagion [AMS09]. Hence, homophily could be used to explain and anticipate product adoption or data consumption patterns.

2.3 Anti-properties

This section discusses two "properties" of social networks, which are widely believed to be true, but recent evidence is emerging which suggests otherwise.

2.3.1 Trustworthiness of links

Most social systems implicitly make the assumption that users only form links with other users they trust, and further, that correct meanings are attached to the links. Even if users can be trusted to be honest, relying on the correctness of links requires an assumption about the hardness of *social engineering* attacks, which aim to fool honest users into creating false relationships. The security firm Sophos demonstrated the surprising effectiveness of such attacks. It created two fictitious Facebook users, complete with a false profile and an age-appropriate picture[16]. Facebook friend requests were sent out to 200 Facebook users, and nearly 50% of them voluntarily created a link [sop09]. Similarly, Burger King ran an advertising campaign called Whopper Sacrifice, in which it was able to induce a number of users to *shed* 10 Facebook friends each, in return for a free burger [Wor09].

A new and powerful class of social engineering attacks makes use of the tendency of people to trust their friends or friends of friends. [BSBK09] describes attacks based on cloning data from a victim's online profile and sending friend requests (either on the same social network where the cloned data was obtained, or on a different social network, where the victim is not yet registered) to the victim's friends. [BMBR11] shows that it is easy to

[16] 21 year old "Daisy Felettin" and 56 year old "Dinette Stonily", whose names are anagrams of "false identity" and "stolen identity" respectively. This expands on a 2007 study by the same firm, using a profile name "Freddi Staur", an anagram of "fraudster".

expand from one attack edge: Friends of an initial victim are more likely to accept friend requests from an attacker (i.e., attacks are more successful with new victims whose friends are "friends" of the attacker). Disturbingly, these attacks have been shown to have a success rate $> 50\%$.

A similar problem arises with the correctness of node identities. To be sure, many social network operators would like users to have identities that can be relied upon: Facebook's Zuckerberg famously wants people to have a single identity online and offline [Kir10, p. 199]. Rafiki presents an extreme example of ensuring the integrity of identities. Because it explicitly targets school children, Rafiki has moderators who ensure the integrity of identities, and even go to the extent of monitoring interactions to make sure that the guidelines of the site are not violated. Howeever, such support is not available or feasible on most social networking sites. False identities by themselves are not usually of much concern as long as they are not able to make real users link to them. But attackers do not need fake identities if they can compromise real users. A compromised user automatically gives the attacker access to her "real" links into the social network. The reality of this threat can be seen from numerous reports of compromised profiles of honest users on different social networks [McM06, Zet09].

Activity networks: a potential solution considered

It has been argued that all links on a social network do not represent the same level of trust. For instance, some users might find it difficult or socially awkward to refuse friendship requests and form links out of "politeness". One point of view [WBS+09] holds that different social links represent different levels of trust, and systems requiring a high degree of trust should not use all explicitly declared links. Instead edges should be filtered based on the presence of social activity (e.g. bidirectional conversation records).

Unfortunately, while the studies in [CKE+08] show that such *activity networks* have qualitatively similar topological characteristics as the original social networks, the evaluations in [WBS+09] show that systems such as RE: [GKF+06] and SybilGuard [YKGF06] which depend on links being trustworthy, do not perform as well when using the sparser and fewer edges

of the activity networks, as they do on the denser native social graph. A further complication is that the activity graph is highly dynamic, with the strength of interaction between node pairs changing over time [VMCG09].

A different point of view suggests that people may be interacting more often than suggested by records of actual conversations. [BRCA09] study clickstream data of user actions on online social networking sites and find that 92% of all activity on social networks involves browsing friends' profiles. A similar conclusion is reached in [JWW+10], which mines the extraordinarily rich and accessible records of the Renren social network (These records include visitor logs to each profile, etc.).

Thus one potential solution for verifying trustworthiness of links could be to use networks inferred from consistent and bidirectional interactions, either based on silent browsing, or explicit conversational activity. However, even this may not work: The social success of bots such as Cabot on LambdaMOO [IKK+00] suggests that spammers might be able to fool users even when interaction is required.

Implication: From the viewpoint of a systems designer, the weakness of online identities and links implies that systems have to be designed defensively to be robust against compromised or false identities and links, even though most social network links encode a high-level of trust between the human actors involved (especially when they are bidirectional). Some systems, like Sybilguard [YKGF06] and SybilLimit [YGKX10] parameterise the security they offer based on the trustworthiness of links. For instance, SybilLimit guarantees that no more than $O(\log n)$ false (Sybil) nodes will be accepted as long as the number of *attack edges*, between a compromised node and an honest node, is bounded by $\Omega(\sqrt{n}\log n)$.

2.3.2 Fast-mixing

As we will discuss in §2.4, many social systems (e.g. [YKGF06, YGKX10, DM09, LLK10]) have been built on the assumption that social networks are *fast-mixing*. This is a technical property which essentially means that a random walk on the network will converge to its stationary distribution

after very few steps, i.e., every edge in the graph has an equal probability of being the last hop of the walk, or equivalently, the probability of the walk ending on a node is proportional to its degree[17]. If the mixing time of a graph is T, then after T steps, the node on the random walk becomes roughly independent of its starting node. Typically, an n-node graph is considered to be fast-mixing if $T = \Theta(\log n)$.

The first use we have come across of fast-mixing in social systems is in the design of Sybilguard [YKGF06]. There the authors assert that social networks are fast-mixing, citing [BGPS06][18], who in turn note that stylised social network models such as [Kle00], which are regular graphs with bounded degree, are fast-mixing. Another theoretical proof comes from [TSJ07] who show using spectral arguments that the mixing time of a *cycle* graph decreases from $\Omega(n^2)$ to $O(n \log^3 n)$ when shortcut edges are added with a probability $p = \epsilon/n$ (i.e., every one of the $n(n-1)/2$ edges is chosen as a shortcut with a probability p). These shortcuts effectively convert the cycle into a small world graph similar to the Watts-Strogatz model discussed earlier. However, this result is higher than the $O(\log n)$ mixing time assumed by the systems mentioned above.

Unfortunately, the theoretical results about fast mixing do not always generalise to real-world scenarios: [Nag07] simulates the Kleinberg model and a social network drawn from LiveJournal, and showed that they mostly have good mixing properties, worse than but close to that of expander graphs. However certain parameter choices, such as a large density of short-range links in the Kleinberg model, are shown to severely affect convergence times. [MYK10] conducts an extensive survey of mixing times in different social network data sets, and finds two sources of variation in convergence times: First, within a given social network, random walks starting from some nodes converge more slowly than others. Since mixing time, by definition, is the maximum of the different convergence rates, this implies slow

[17] This terminology derives from the theory of Markov chains, where the term *mixing time* is used to denote the speed with which a Markov chain converges to its stationary distribution.

[18] Actually, the conference version [BGPS05] is cited, but this does not directly talk about social networks.

mixing. Second, some social networks mix more slowly than others. The authors observe that networks such as scientific co-authorship networks, which may encode high degrees of trust, are slower. They note that mixing times improve dramatically if the lowest degree nodes are deleted from the social network, as done in the evaluation of Sybilguard and similar systems. **Implication**: The empirical results above seem to suggest that fast-mixing may not be a universal property that applies in all social graphs. For *some* nodes, such as those of large degree, or in some densely connected social networks, it may be possible to use random walk-based methods that rely on converging fast to a stationary distribution, but convergence times should be explicitly tested before applying such methods.

Note that the correctness of random walk-based techniques is guaranteed as long as the walk is long enough to ensure convergence to the stationary distribution. Thus, as long as one is willing to pay the performance penalty of long random walks, such systems can still be used. In other words, a weaker property that can always be relied upon is that a sufficiently long random walk in social networks will always lead to a node which is independent of the initial node. This property is guaranteed to hold as long as the social graph is connected and non bipartite [Fla06].

Unfortunately, the security guarantees of random walk-based Sybil defenses like Sybilguard may be compromised if the walk ends up being too long, and thereby allows walks starting from honest nodes to "escape" to a region controlled by Sybils.

2.4 Social network-based systems

In recent years, numerous systems have been proposed to make use of the properties of social networks. Many of these systems attempt to leverage the trust encoded in social network links to provide additional security. Accordingly, several of the systems are anti-spam or anti-sybil attack solutions. Another major application area is to modify social networking systems themselves, by taking advantage of locality properties of social network data. This section provides a brief survey of such systems.

2.4.1 Systems to combat spam

[BR05] note that spammers use temporary email addresses and send emails to a number of unrelated people. Thus, triangle structures (§2.2.2) are largely absent in their subnetworks in the email social graph. Thus the absence of clustering, or a low clustering co-efficient (see §4.4.3 for a definition), is used to label spam. [LY07] uses a machine-learning approach to semi-automatically classify spam based on a number of social network characteristics including clustering co-efficient.

RE: [GKF+06] is based on a similar idea. RE: introduces a novel zero knowledge protocol that allows users to verify their friends and friends of friends using secure attestations. This is used to construct an email whitelist. Because a majority of legitimate emails originate close to the receiver's position in the email social network, they demonstrate, using real email traces, that up to 88% of real emails which would have been misclassified as spam, can be identified.

Ostra [MPDG08] is an elegant system, loosely based on Hawala, the ancient Indian system of debt value transfers (still prevalent in India and a few countries in Asia and Africa), that prevents spam by requiring a chain of credit transfers in the social network to send emails. Sending emails decreases the available credit balance. This balance is eventually restored if the recipient accepts an email. This system allows legitimate senders to operate freely, but sending bulk emails (or falsely marking as spam) quickly pushes the sender (or a lying recipient) against the credit limits, and prevents further damage. Note that Ostra is not overtly based on any social network property, but relies on nodes cutting links to untrustworthy spammers in order to preserve their own credit lines. [PSM11] is a successor, which introduces a max-flow computation and secures reputations on online marketplaces from Sybil and whitewashing attacks.

2.4.2 Systems to combat Sybil attacks

The Sybil attack [Dou02] is a powerful attack on distributed systems in which a single entity poses as multiple identities, and thereby controls a

substantial portion of the system. One of the most appealing, if flawed, applications of social networks has been the development of defenses to this attack by leveraging trust embedded in social network links.

These systems are all based on the following central assumption: While an adversary can create any number of Sybil identities and relationships between them, it is much harder to *induce* an *attack edge* between a Sybil node and an honest one. All Sybil defenses are based on assumed scarcity of attack edges. Note that on one side of the set of attack edges is the set of honest nodes, or the *honest region*; on the other side, the set of Sybil nodes, or the *Sybil region*.

Two major approaches have developed to separate nodes in the honest region from the Sybil nodes. The first [YKGF06, YGKX10] relies on performing multiple random walks on the social network. The random walks are unlikely to cross attack edges because they are scarce. However, random walks from two nodes on the same side of the attack edge can easily intersect because they do not have to cross the attack edge. By performing enough number of random walks from each node, the probability of intersection can be made high enough for two nodes in the same region. Thus, the central protocol involves a verifier node and a suspect node. The verifier accepts a suspect node if their random walks intersect. Performing such random walks is typically assumed to be easy because social networks are considered to be fast-mixing (see §2.3.2)

The second approach [TLSC11] is based on the realisation that when attack edges are scarce, the maximum flow between the honest and Sybil regions is small. Instead of actually computing the flow, the approach uses a ticket distribution mechanism. Each verifier node distributes a set of tickets in a breadth-first manner starting from itself and the protocol accepts a node if it has enough tickets. If a verifier node is close to the Sybil region, then Sybil nodes could end up with some tickets. By carefully choosing multiple verifier nodes randomly across the network, the protocol removes this sensitivity to the positions of individual verifiers.

Other systems based on similar approaches

[DM09] first creates a trace based on random walks, and then, accepts or rejects new sets of nodes using the trace and a Bayesian model to decide if the nodes belong to the Sybil or honest region. Nagaraja [Nag10] shows how the privacy of a shared weak secret can be amplified by performing random walks. Whanau [LLK10] develops a Sybil-proof Distributed Hash Table based on random walks. SumUp [TMLS09] is a precursor to [TLSC11] that describes a secure vote collection protocol based on ticket distribution.

* * *

[VPGM10] analyses a number of social network-based Sybil defenses and shows that they all work by detecting local communities. This suggests that social graphs with well-defined communities are more vulnerable and that adversaries which can create targeted links can be more effective.

2.4.3 Social network support for online social network systems

It is natural that properties of social networks could be used to support online social network systems. The primary issue here is that social data typically involves many-many communication and is both produced and consumed within the social network (For example, consider Facebook Wall posts or email communications). This creates a high degree of inter-dependency, which makes it difficult to distribute across multiple locations.

[WPD+10] shows that Wall traffic on Facebook is mostly localised between friends in the same geographic region. Thus, by introducing regional TCP proxies, both the load on Facebook's central servers as well as the latency of user access can be reduced. [KGNR10] and [PES+10] consider data from a company's email network and Twitter respectively and show that users can be partitioned efficiently using versions of the METIS[19] graph partitioning tool. [STCR10] show how to deliver the latest data and improve performance at the same time, by selectively materialising the views of low-rate producers and querying the latest data from high-rate producers.

[19] http://www.cs.umn.edu/~metis

2.5 Present dissertation and future outlook

This chapter surveyed the different types of social networks available (§2.1), their common properties (§2.2) and misconceptions (§2.3), and some early systems which make use of social networks and their properties (§2.4). We conclude by discussing how future systems can learn from these early efforts, and how the approach proposed in our thesis (§1.2) differs from previous work and contributes to the area.

First, we note that many of the early social network-based systems share similar approaches based on graph-theoretic concepts: random walks, max-flow and graph partitioning. We expect that these approaches will be used more commonly.

Second, we observe that very few of the properties of social networks mentioned in §2.2 are being directly used. We believe that there is great potential for future work, exploiting other properties of social networks.

Curiously, the very properties which are least likely to be widely true (trustworthy nodes and links, and fast-mixing; see §2.3), are also the most commonly relied upon properties. This indicates that future social network-based systems should first empirically establish the validity of a property before using it.

Our dissertation is a step in this direction. The approach we outline in §1.2 first measures a relevant property from empirical traces and then develops ways to exploit it. We note that some concurrently developed systems mentioned in §2.4.3 (all of them published in 2010, after we finished our work) also use empirical trace-based analysis to drive their designs. However, our work differs in two important respects: First, we have identified several new properties of social networks (§3.3 and Chapter 4), and base our designs on these properties. Second, social network systems are a natural area for applying social network-based system designs. In applying social network properties to support data delivery infrastructures, we have identified a new, and perhaps unexpected, application area for social network-based systems.

CHAPTER 3

Cost-effective delivery of unpopular content, or Akamaising the Long Tail

This chapter presents the first case study on adding social network support for data delivery infrastructures (cf. Problem 1.1). We focus on rich media content and show how social networks can help deliver so-called "Long Tail" content items which are individually unpopular but collectively are important to a large fraction of the user base and therefore important to deliver effectively.

3.1 Introduction

Recently there has been a greatly increased demand for rich media content because of the popularity of sites like Netflix and Hulu in the USA, BBC iPlayer in the UK, and equally importantly, the growth of user-generated video sites like YouTube, vimeo, metacafe, dailymotion etc. Rich media content such as streaming videos have strict latency and jitter requirements—a video frame that arrives at the client after its playback time is not useful.

One way to meet these requirements is to use specialised content delivery networks (CDNs) like Akamai. CDNs are essentially a geographically diverse set of servers which help place the content closer to the users and thereby help ensure the strict requirements of streaming videos are met. In recent years, concurrently with the rise of online videos, CDNs have started to play a central role in the Internet, to the point that "most Internet inter-domain traffic [now] flows directly from large content providers and hosting/CDNs to consumer networks" [LIJM+10].

However, the cost of deploying content on current CDN architectures is only offset based on the assumption that the content is popular. Indeed, YouTube is known to use a CDN provider[1] for delivering its popular content, but serves the unpopular tail content, which may receive as few as 10–20 accesses per day, from its own servers distributed across multiple CoLo sites in the USA [Do07, 10:10 minutes into video]. While this strategy is sufficient for well-connected clients located relatively close to the servers, user experience and quality of service can suffer for those who are far away[2] (e.g. countries in Africa) [Hec10].

This chapter is concerned with the problem of developing *cost-effective* content delivery strategies for rich-media user-generated content which may have a small user base. Such "unpopular" content is said belong to "The Long Tail"[3] [And04]. Individually, items in the tail receive very few accesses. Collectively, however, they are accessed a non-trivial number of times (more than would be expected in the tail of a normal distribution; hence the term long tail), and therefore need to be delivered effectively.

[1] Currently (2010-11), it uses Google's own CDN for the majority of its needs [TFK+11, Hec10], but earlier it used an external CDN provider, widely believed to be Limelight Networks [Kir06]. [HWLR08] also states that YouTube used Limelight, but without a citation.

[2] Geographical distance will be used as a rough approximation for network distance in this chapter. In reality, if there are no cost constraints, a well-provisioned link could be engineered between two points which are physically far away.

[3] We follow Anderson in capitalising The Long Tail in this chapter.

Contributions

Examining characterstics of tail content taken from a social video sharing website (vimeo) and from a social news sharing website (digg), we find that the unpopular tail content is primarily accessed virally, spreading from friend to friend, whereas accesses to popular content are seeded independently in several parts of the social network, making the accesses predominantly non-viral.

Based on this insight, we first design SpinThrift[4], a strategy for saving energy in the storage subsystem. Using the ratio of viral to non-viral accesses to predict whether an item will be unpopular or popular, we segregate the popular content from the unpopular. This allows the disks containing unpopular content to be put into an idle mode when their content is not being accessed, thereby saving energy.

Notice that the ratio of viral to non-viral accesses could also be used by a content provider to decide which items to serve using expensive CDN infrastructure (the popular ones, with predominantly non-viral accesses), and which ones to serve in a more cost-effective manner (the unpopular tail content).

Next, we design a cost saving measure for delivering the tail content. Our solution, Buzztraq[4], saves cost by limiting the number of replicas for Long Tail content. In order for such a strategy to be effective, the locations for the replicas need to be carefully selected. We select the locations based on the observation that the predominantly viral nature of the accesses to tail content makes it more likely for future accesses to come from the *locations of the friends* of previous users. We show that selecting replicas based on this strategy performs better (results in more local accesses) than simply using the history of previous users to decide top locations.

An important part of our contribution is the characterisation of tail content using data from vimeo and digg. Apart from the predominantly non-viral nature of accesses to popular stories, we also find that daily popularity ranks of items can vary dramatically, and that as a ratio of the total

[4]Names suggested by Prof. Jon Crowcroft.

number of accesses, accesses to tail items are geographically more diverse than accesses to head items. We also discover that a large fraction of users access items in the tail, and exhibit a preference for tail content, which justifies the need to effectively distribute such items.

Chapter layout

§3.2 discusses the notion of The Long Tail and two conflicting theories about the relative popularity and utility of the head and tail in Internet-based catalogues of physical items like books and DVDs. §3.3 then characterises the tail of user-generated content using data from digg and vimeo. §3.4 builds on these results and develops a way to segregate popular content from the unpopular and save energy required for storage. Then we turn to delivering long tail content. §3.5.2 discusses various design choices to be made, and justifies our decisions. §3.6 develops a selective replication scheme that works well for unpopular (viral) work loads. §3.7 concludes.

3.2 The tail of Internet-based catalogues: Long Tail versus Super Star Effect

User-generated content is not the first online arena where items in the tail have come under attention. The Long Tail phenomenon in Internet-based catalogs of items such as books, movie rental libraries etc. has been debated extensively by economists and management specialists. Before we begin, we review the literature in this area to understand the nature of the tails of large online collections, and highlight research problems for content delivery.

At the risk of over simplification, there are two main viewpoints [BHS10a]. The first view, popularised by Chris Anderson in Wired Magazine [And04] and later expanded to a book [And06], is based on the observation that unlike physical "brick-and-mortar" retailers which are limited by building sizes and the cost involved in stocking and distributing to a geographically diverse set of stores, online retailers can offer a much larger catalogue, including so-called tail items whose individual sales are low enough to be cost ineffective

for brick-and-mortar retailers. One of the earliest works in this area [BS03] examined Amazon book sales and showed that the tail items contribute to a significant fraction of Amazon's total sales. Subsequent work has revealed that in the last few years, the tail has become "longer" and accounts for a larger fraction of Amazon's sales [BHS10b]. This demand for the tail items seems to be helped or driven by recommendation systems that present a tailor made suggestion of new items based on prior consumption patterns, and furthermore, this effect is enhanced by higher assortative mixing and lower clustering in the network, and is greater in categories whose products are more evenly influenced by recommendations [OSS09]. Curiously, it appears that the more internet-savvy users have a tendency to gravitate towards obscure items, whereas the more savvy users of physical catalogues tend to consume the popular items [BHS07, BHS06].

The second view, dubbed the "Superstar" effect, draws on the economics of Superstars [Ros81, FC95], and posits that since the Internet makes it *easier* for the consumers to discover the true hits (the superstars), and for these items to be delivered to the consumers wherever they are geographically, popular products can become disproportionately profitable over time [EOG06, TN09]. According to McPhee's Theory of Exposure [McP63] the positions of items in the head and tail get reinforced because the popular items have a "natural monopoly" over the light users who may not be sophisticated enough to know about the more obscure items, whereas the unpopular items have a "double jeopardy" in that they are not well-known and when they do become known it is by sophisticated heavy users who "know better" and prefer the popular products. Using this as justification and based on her work on online DVD rentals [Elb08b] which exhibits the long tail effect but also confirms the "double jeopardy" effect, Elberse suggests that firms should continue to invest in blockbusters rather than the tail [Elb08a]. However, in a later work, she finds that consumers who have a greater share of products in the long tail not only consume with a higher frequency but also are less likely to leave, making them a valuable group to cater for [ES09]. [FH09] shows that some common recommender systems such as collaborative filters can cause a rich-get-richer effect because

they cannot recommend items with limited historical data, thereby decreasing aggregate diversity, even as the new product recommendations increase diversity at an individual level by pushing new products to users. Using Web-based music downloads as a controlled experimental setting, [SDW06] shows that even a weak social signal such as how many others have consumed an item can increase the inequality of success between items (where inequality is measured in terms of the Gini coefficient, and success is counted by the number of times an item is downloaded). However, they find that social signals also increase the unpredictability of success (unpredictability is measured as the difference in market share achieved across different realisations of their experiment).

3.2.1 Commentary: is there value in the tail?

It is natural to ask whether these two views describe the same reality and whether it is worth distributing the tail effectively. We suggest that there are two reasons to see value in the tail, regardless of the two views above: First, the tail items which are unavailable through other channels represent additional value to the consumer, regardless of the numbers consumed. Because online retailers only incur marginal additional costs for listing and stocking tail content centrally, they can afford to sell/rent such items. Second, presenting the entire catalogue of items to the consumers lets them directly choose which items to consume. This may lead to some unexpected "hits", which may not otherwise have been discovered.

Tail items unavailable through other channels

In response to Elberse's recommendation against investing in the Long Tail [Elb08a], Anderson suggested a way to reconcile the two views [And08], by pointing out that the Long Tail proponents use an absolute number (typical inventory size of brick-and-mortar retailers) to define the head of the distribution, whereas Elberse, Tan and others define the head as a predefined proportion of the most popular items (10% or 1%). When the percentage is translated to an absolute number, it can still be too large for

a physical store. This suggests that there is added value for online retailers who provide consumers access to the tail, regardless of the geographical location of consumers. Indeed, regardless of whether the head becomes more influential over time than the tail, it has been observed that the items in the tail represent a non-trivial increase in consumer surplus [5][BS03].

Unexpected hits

Rajaraman [Raj08] offered the deeper explanation that the improvement offered by online retailers is not necessarily that the consumers shift their preferences to the tail but that the hits are decided democratically by consumer feedback, rather than by media executives of a publishing channel choosing which items to market in a selective catalogue. There have been well publicised instances of artists and bands such as Arctic Monkeys and Lily Allen who have subverted the traditional selection process by recording industry executives and gained popularity through self-promotion on sites such as MySpace [Wei08, Wag07]. These success stories suggest that some items in the "traditional" tail could be less popular not because they are inherently lower quality items but rather because they have not been distributed and highlighted in the same way as the hits in the traditional head[6].

3.2.2 Takeaways and problems for research

Finally, from the above discussion we would also like to extract principles for distributing user-generated content. Note that unlike with items which have traditional distribution channels to physical stores, we are not directly concerned with expanding inventory size. Rather, we are interested in expanding the use of CDNs in a cost-effective manner. To the extent that the content provider is resource- or budget-limited, the savings thus enabled could effectively increase the number of items distributed through

[5] An economic measure of the benefit gained from consumption of items.
[6] This insight is due to Prof. Jon Crowcroft.

the content delivery network, and thereby expand the "inventory" of items delivered from an optimal location for the client.

From the above discussion, we can identify at least three different areas where improvements could be made for content delivery: First, discoverability of content is crucial in systems where the size of the catalogue is extremely large. Traditionally, online retailers have used recommender systems to help users find items they might be interested in. The current generation of user-generated video sites such as YouTube and vimeo also recommend new content based on users' past history. However, they are still limited to videos found in the catalogue of the site. Unlike traditional items such as printed books or movies, user-generated video is created at a much higher rate, and no single provider's site contains the entire catalogue; even keeping the catalogue always current on all the globally distributed replicas of a single content provider becomes a challenge. It is an interesting open problem to find a good way to help users discover user-generated content across different sites. The finding (mentioned earlier) that assortative mixing and lower clustering can improve the effect of recommendations [OSS09] could provide a unique angle of attack from the social network perspective.

The second and third research problems derive from the observation that the items in the head gain more value over time, and therefore benefit from more investment [EOG06, TN09]. In the context of content delivery, this suggests that the more popular items should be delivered using the expensive but effective CDN infrastructure that is currently in place. The second research problem is deciding the criteria on which to base the decision of whether to use traditional CDNs for a given video. Which items should be distributed using traditional CDN infrastructure? When, and on what criteria, should an item be moved to the CDN, and equally, when should it be moved out of the CDN, to cut down expenses? One way to approach these questions is by framing it as a multi-dimensional optimisation problem depending on several factors such as the budget of the content provider, the access patterns for the video, the nature of peering arrangements and service level agreements, and so forth. The third area for improvement is finding cost-effective ways to deliver content which cannot be accommodated into

the the traditional infrastructure.

Note that the answer to the second problem would depend to a large extent on the third problem (how much cost savings is achieved), and also on the first question (how new content discoverability methods will affect access patterns). This chapter leaves open the first two problems for future work and develops two strategies towards cost-effective delivery of tail content. The goal is to improve over the current state-of-the-art, which simply delivers the unpopular content from a centralised website (or a few CoLos—data centers which are well connected to a reasonably large portion of the Internet) operated by the origin content provider.

3.3 The tail of user-generated content

In this section, we will explore the characteristics of the tail of user-generated content, using data from a social news site and a video sharing site. The focus is on identifying trends in access patterns with a view to derive principles for more efficiently delivering such content.

First we discuss how we define the "tail" and "head". Then, we describe the data sets used in this section and in the rest of the chapter. Following this, we discuss four different aspects of the popularity distribution and use these to guide our design, as summarised below:

The importance of the tail We find that although in terms of the number of accesses the "weight" of the tail is small compared to the head, the tail is still important because of the number of users who like videos in the tail. This justifies our focus on distributing tail items. However, the small numbers of tail accesses also dictates our design choice of being cost-effective.

Geographic diversity of accesses Compared to popular items, a larger fraction of the audience for an unpopular video come from a unique country. In §3.5.2, we use this insight to argue for push-based CDNs.

Dynamics of popularity The popularity rank of individual items can

vary dramatically over a short duration of time. This motivates the design of the energy saving scheme in §3.4. While the goal of the scheme is to save energy by segregating popular items from the unpopular, we require the algorithm to be robust in the face of dynamic changes to popularity rank order. To meet this goal, we design a binary classification scheme and show that it performs better than a rank-based classification scheme.

Impact of viral accesses Contrary to the popular picture of videos which become wildly successful by "going viral", we find that most videos which have predominantly viral accesses are unpopular. We use the finding both to guide the design of a selective replication mechanism for unpopular content (§3.6) and a binary classifier of content as popular or unpopular (§3.4).

While it is hard to make concrete claims about the representativeness of our data sets for access patterns involving other kinds of user-generated content, we will provide a brief corroboration for each aspect discussed below, by mentioning related work with comparable statistics from other online catalogues and social networks. We stress that the results cannot in many cases be compared directly because of the different types of media and social networks used. However, the similarities seen can be interpreted as an indication of underlying global trends that hold across different kinds of media and social networks. To the best of our knowledge, there are no apparent or major contradictions in the literature.

3.3.1 Defining the head (popular) & tail (unpopular)

Unlike traditional items such as books and movies, user-generated content does not have an alternate physical distribution channel. Thus we cannot easily pick an absolute number (the inventory size of the "brick-and-mortar" stores) to define the head[7]. Picking a fixed percentage (say $x\%$) of the total

[7]One plausible candidate is the number of videos picked by sites such as YouTube, to be distributed by external CDN providers; unfortunately we do not have access to

number of videos also becomes equally arbitrary: this ignores the variance in the number of accesses a video gets—If the deviation is large and a handful of videos account for the majority of accesses, then picking too large a value for x will cause much less popular videos to be included in the head along with the truly popular handful.

We get around this problem by defining the head as items whose consumption counts are at least one standard deviation above the mean consumption. Items with a smaller consumption count belong to the tail.

In the rest of the chapter, we will adopt the above definitions of head and tail. Also, the adjective "popular" will be used exclusively for items in the head. Items in the tail are referred to as "unpopular" items.

3.3.2 Datasets

We use two main traces in our study, both of which were collected by crawling the respective websites using public APIs made available by the sites. Our first trace is based on one week of data from Aug 29, 2008–September 6, 2008 from Digg[8]. Digg is a social news website: Users submit links and stories. Digg users collectively determine the popularity of a story by voting for stories, and commenting on it. Voting for a story is termed as "digging", and the votes are called "diggs". Collectively, "diggs" and comments are termed item activities. Some stories are identified as "popular" by digg, using an undisclosed algorithm. We use these for ground truth about popularity values[9].

Digg also has a social networking aspect. Users can follow the activities of other users by "friending" them. Note that these are one way links, and are frequently not reciprocated. The Digg website highlights friends' activities and stories "dugg" by friends.

Our second trace is sampled from videos and user ids found in all the

this number. We do not even know whether YouTube picked a fixed number of videos to be distributed by its CDN provider.

[8] http://www.digg.com

[9] Our results are qualitatively similar if we equivalently use the method described below for vimeo on the digg data.

groups and channels of the video-sharing website vimeo[10]. Similar to Digg, users can "like" videos; these are counted as item activities. A video whose "likes" number more than one standard deviation in excess of mean is counted as "popular". Users also have contacts, similar to Digg friends. The vimeo website highlights the activities of contacts by allowing users to follow their friends' channels, subscriptions, likes etc. The details of both data sets are summarised in Table 3.1.

	digg	vimeo
Item statistics		
Number of items	163,582	443,653
"Popular" items	17,577	7,984
item activities	2,692,880	2,427,802
Graph statistics		
Number of users	155,988	207,468
directional links	1,516,815	718,457

Table 3.1: *Trace details*

While digg is clearly not a site for *rich-media* user-generated content, the similarities we show between the results for digg and vimeo lends some credence to our belief that other user-generated content sites, including those focusing on rich-media user-generating content, are likely to exhibit similar trends as seen below.

3.3.3 How important is the tail?

To start with, we examine what fraction of activity is due to the tail. Figure 3.1 shows the fraction of views contributed by the least popular videos. Two different measures of views are available: The number of plays, and the number of likes. Likes is a much stronger measure, and indicates that users preferred a video over others they have seen and not "liked".

First, notice that the distributions for "likes" and "plays" are similar. We use this as justification for substituting likes for plays. We need to

[10] http://www.vimeo.com

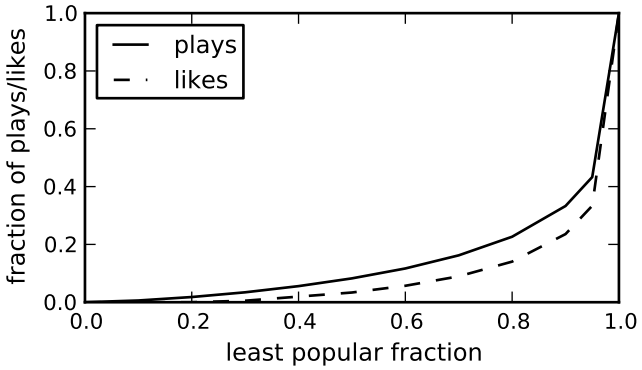


Figure 3.1: *The fraction of views contributed by the least popular x% of videos, ordered by two different measures: number of likes and number of plays. Although for individual videos there may be no correlation between number of likes and plays, at an aggregate level the contribution of the tail by both the measures are similar, and exhibit a familiar 80-20 split (Vimeo trace).*

do this at several places because of a limitation of the Vimeo dataset. The dataset only provides the aggregate number of plays for each video. Likes on the other hand, are listed with the name of the user or the time of action. We use likes as a proxy for plays when we need the additional details. Whilst we have no knowledge of how the plays are distributed in time or across users, we assume that this would be similar to the corresponding distribution of likes (except that the total numbers of likes would typically be much fewer than plays).

Next, observe that with both measures, we see the familiar 80-20 split: The least popular 80% of videos only contribute approximately 20% of plays or likes. While the tail is long, this figure seems to reinforce the "Super Star" argument and could be because of a number of different reasons. On the one hand, some popular videos might have received a disproportionate number of accesses simply because they were featured on a high-profile page, for instance, on the home page of vimeo. On the other hand, a video could be genuinely of niche interest: for example, videos of home vacations, which might be of interest only in the video author's immediate social circle.

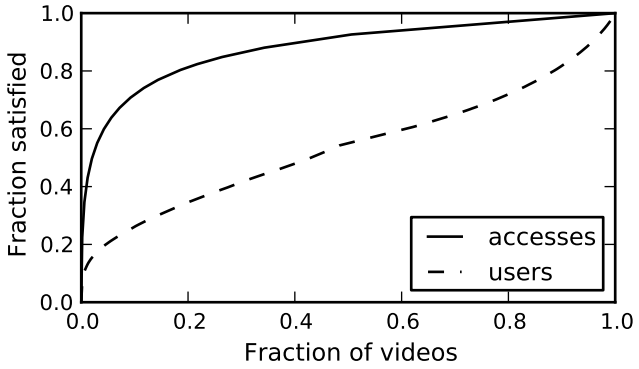


Figure 3.2: *The tail is long in users. If the tail is defined in terms of number of accesses or likes, the familiar 80-20 rule is obtained, with the top 20% of videos satisfying 80% of accesses or likes. However, when we look at the fraction of users all of whose "likes" would be fulfilled by the top x% of videos, more than 80% of videos need to be included to satisfy 80% of the user base (Vimeo trace).*

While the above suggests that there is not much value added by distributing the large number of items in the tail, Figure 3.2 shows that by a different measure, the tail is extremely important. In contrast with the 20% of videos which can satisfy nearly 80% of accesses, more than 80% of videos need to be included to satisfy 80% of the user base. In other words, most of the users have *some* interest in the tail.

Further, Figure 3.3 shows that users are disproportionately interested in tail videos. Nearly half the users have 60% or more of their likes in the tail. Just under 30% of users have *all* their likes in the tail. Thus, to have a satisfied user base, it is necessary to effectively distribute nearly their entire catalogue of videos.

It may appear paradoxical that despite nearly half (47%) of users having a majority (at least 60%) of their likes in the tail, the tail adds up to just 20% of the total number of likes. This conundrum is resolved by looking at the other half of the user base. Not only do they have at least 40% of their likes in the head, but they also have more likes individually (a median of 8 likes per user, in comparison with 4 likes among users who have a majority

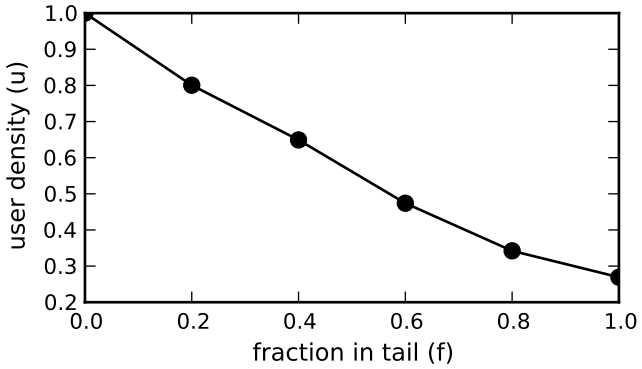


Figure 3.3: *The density of the tail likes: u% of users have a fraction of f or more of their likes in the tail. (Vimeo trace).*

of likes in the tail). Thus, collectively, the head gets more likes than the tail.

Corroboration: Similar to our results, [GBGP10] explores the nature of the tail of user preferences and finds that in movies and music, most users are "eccentric" in that they have an interest in and consume items from the tail. Note however that they use Anderson's definition for the tail (items which cannot be accommodated in the inventory of a physical store). They also find that the eccentricity is lesser than would be predicted by a random model in which accesses are purely popularity-driven—most users are found to have an a priori preference for the products in the head or products in the tail.

3.3.4 Geographic diversity of accesses to tail items

The primary difficulty in streaming rich-media content is that it needs to be served from close to the viewers. To better understand this, we next look at the geographic diversity of likes. To obtain this data, we need to map users to a geographic location. In Vimeo, each user can enter an arbitrary string description of their location. We use Google and Yahoo geocoding APIs to first map these strings to a latitude-longitude pair, and then obtain the country of the user. We find that there is great diversity,

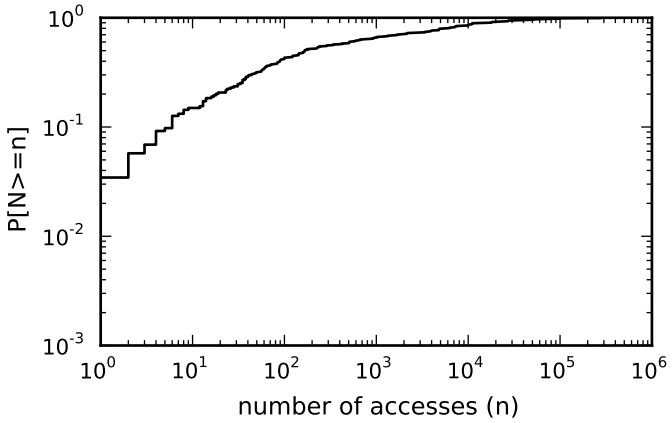


Figure 3.4: *Complementary CDF of the number of likes coming from users in a country. (Vimeo trace, loglog scale).*

with likes coming from 174 countries (The current number of internationally recognised countries stands at 194). There is also a great disparity in the amount of use: One country, United States, accounts for approximately 20% of the total. The top two countries (US and UK) together account for 31% of the total. Figure 3.4 shows the distribution of the number of likes coming from a country.

We next examine the per-item diversity of likes. We define the geographic diversity ratio of a video as the fraction of its likes that come from a different country, i.e., it is the ratio of the number of distinct countries from which the likes of the video originate to the total number of likes of the video. Figure 3.5 compares the distribution of this ratio for both popular (head) and unpopular (tail) videos. Observe that the unpopular videos have a much larger fraction of their likes coming from a new country. This implies, for instance, that peer-to-peer solutions are unlikely to work well because peers who are interested in an unpopular video are spread out in different countries. Similarly, observe that in a caching strategy which deploys one cache per country[11], the geographic diversity ratio represents the fraction of accesses which access a new cache for the first time. In other

[11] As discussed in §3.5.2 (Server placement), discusses why this is a reasonable choice.

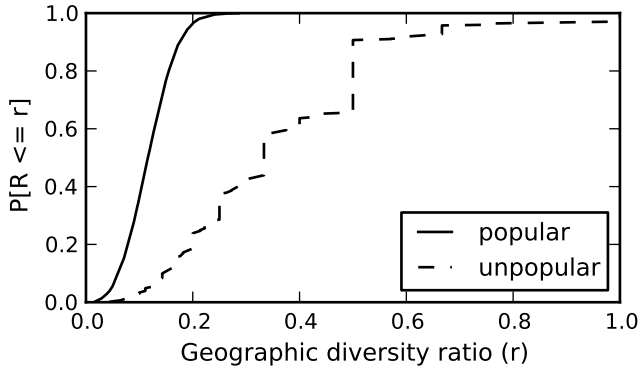


Figure 3.5: ***Geographic diversity ratio****: The geographic diversity ratio measures the fraction of likes of an item that come from a new country. The CDF of this value for popular and unpopular videos is shown. Unpopular videos have a much larger fraction of their likes drawn from a different country. (Vimeo trace).*

words, the geographic diversity ratio also represents the miss rate due to cold caches. We use both these insights to argue for a push-based CDN in §3.5.2.

Corroboration: While not much academic work appears to have examined the geographic distribution of users on different websites, [KGA08] find a similar disparity of usage across geographies in Twitter.

3.3.5 Dynamics of popularity

Next, we examine the popularity distribution of data activities to determine how best to optimise storage and content distribution. Figure 3.6 shows that popular stories get accessed over a much longer time window than unpopular stories. For instance, nearly 80% of unpopular stories in digg have a time window of less than 2 days, whereas the time window for popular stories is spread nearly uniformly over the entire duration of the trace.

This implies that for most popular stories, there is a *sustained* period of interest rather than a brief surge. This in turn suggests a content distribution strategy which treats popular stories differently from unpopular

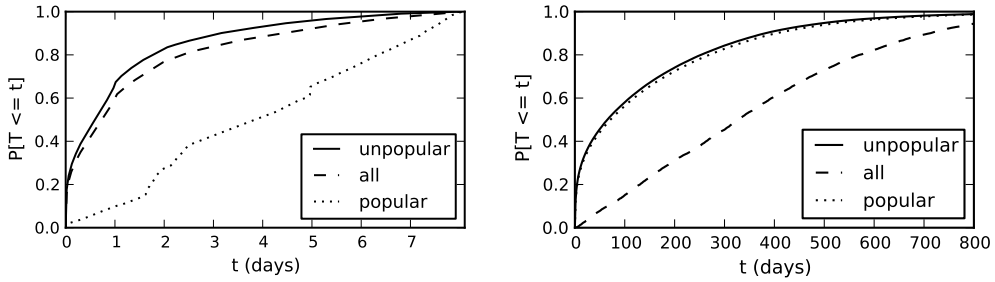


Figure 3.6: ***Popular items have sustained period of interest****: CDF of time window of interest (time of last access minus time of first access) shows that unpopular items have much briefer windows of interest than popular items. Left: Digg Trace, Right: Vimeo Trace.*

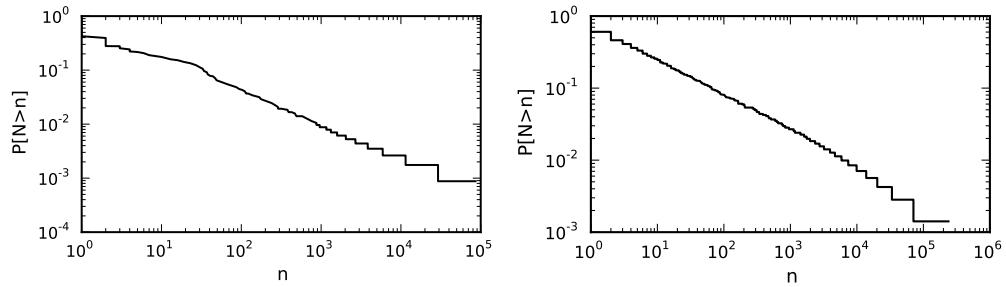


Figure 3.7: *Complementary CDF of number of accesses follows a straight line when plotted on a log-log scale, suggesting a power law. Left: Digg, Right: Vimeo.*

stories. In §3.3.6, we develop a simple scheme to predict which stories will be popular and have a different window of interest than other stories.

It must be noted that there is a boundary effect at play. Some of the digg news articles (similarly videos) could have a longer interest span than the one week we consider. It is equally possible that interest in an old article could have just ended one or two days into our study. Regardless, it can be seen that there is a significant difference in the distributions of the time windows of interest between popular vs. unpopular stories.

Next, we look at the number of accesses received by a story through the entire duration of our trace. Fig 3.7 depicts the complementary cumulative distribution of the number of the accesses, showing a distinct power law-like

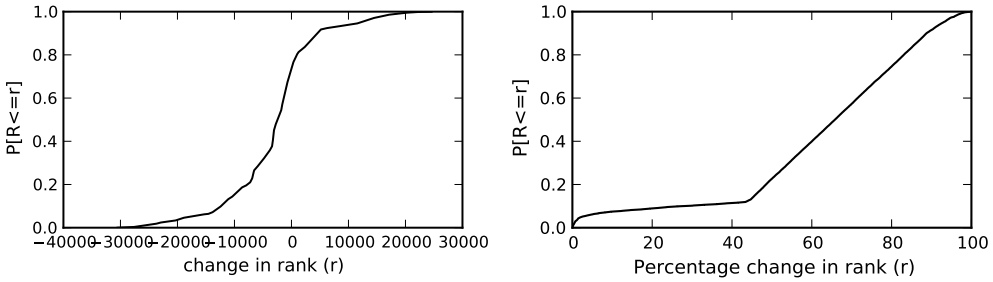


Figure 3.8: ***Dynamic popularity****: CDF of difference in popularity ranks between two successive days shows that ranks can vary dramatically. Left: Digg trace, showing the signed change in popularity. Right: Vimeo trace, showing the change in popularity rank as a percentage of the total number of items. Vimeo shows the change as an absolute number, disregarding whether rank goes up or down.*

distribution—a select few stories receive well over 10,000 accesses whereas a large number of stories receive very few (< 100) accesses.

Fig 3.7 by itself would suggest that popular stories can be segregated from the unpopular ones simply by a rank ordering of popularity. The popularity ranking at any given moment could be used, for instance, to decide whether to use a traditional CDN infrastructure to distribute content, or a more cost-effective mechanism. Similarly, it could be used for saving energy by placing the popular stories on "hot" disks, and storing the unpopular stories on other disks that can be switched off or put into a low-power mode when not being accessed.

However, this ignores the dynamics of popularity rankings. We examine this further by looking at the evolution of popularity between two successive one-day windows (chosen randomly) in the traces. Stories are first ranked by the number of accesses. If a story is accessed in only one of the days, then on the day it is not accessed, it is assigned a maximum rank equal to the number of stories seen across both time periods.

Fig 3.8 shows the distribution of the change of popularity ranks, providing a perspective both as a signed change in popularity ranks, and as a proportion of the total number of items being ranked, disregarding the

direction (sign) of rank change. Both views depict rapidly changing popularity ranks. For instance, in vimeo, nearly half the videos have a rank change of more than 60%. This dramatic change in ranks is driven by the large numbers of new items that are added each day. This implies that a strict popularity ordering of items across disks would require excessive shuffling of items if items were stored ordered by the current rank.

Corroboration: Skewed popularity patterns are common for most kinds of content. In particular, various trends such as skewed access distributions have been found to hold for other video-sharing sites like YouTube as well [CKR+09]. In YouTube, the tail 90% of videos account for 20% of the views. Another study [YZZZ06] looked at a Video-on-Demand setting and found that the 90% of the videos that comprise the tail account for 40% of accesses. [CKR+09] also reported that ranks can change dynamically across a coarse-grained time window of one week. Here we show that ranks can change across a more fine-grained a one day interval, which has stronger implications for energy savings and cache placement.

3.3.6 Impact of viral accesses

Crucially, both the vimeo and digg data sets have information about the social network of the users as well as their access patterns. We next examine how the social network impacts on the popularity. Inspired by theories of viral propagation of awareness about stories, products, etc. that have become prominent recently (e.g. [Gla02]), we tested the hypothesis that the stories which become popular are the ones that spread successfully within the social network. Our results in this subsection collectively suggest that (contrary to expectation) stories are unlikely to become popular when viral accesses predominate over non-viral accesses. Note that we are only able to count accesses which resulted in a "like".

Our definitions of viral and non-viral accesses are adopted as follows. If an item is accessed (liked) by a user after a direct friend on the social network has accessed (liked) it, we term the access as *viral*. In contrast, if no direct friend has accessed (liked) the item, then the access is termed as

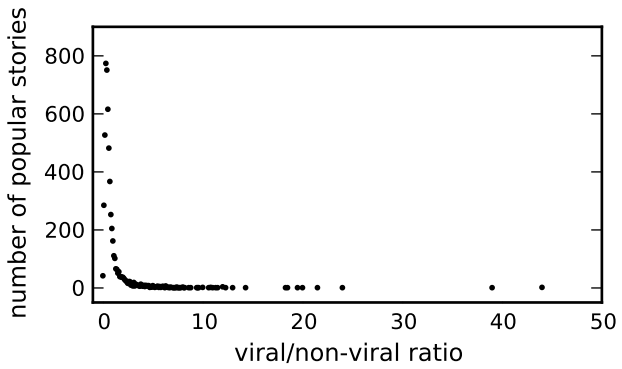


Figure 3.9: ***Non-viral accesses predominate in popular stories****: x-axis shows the ratio of number of viral accesses to non-viral accesses. Y axis shows the number of popular stories that have the corresponding ratio. To clearly show zero valued points, the origin (0,0) is slightly offset from the corners of each graph. (Vimeo trace.)*

non-viral. Relative proportions of viral and non-viral accesses are used to infer popularity.

Figure 3.9 bins videos in the Vimeo trace by the ratio of viral accesses to non-viral accesses (rounded to one decimal place). It then measures the number of popular stories, counted as the number of stories with more than the average number of likes. It can be seen that as the ratio of viral to non-viral accesses increases, the number of popular stories falls drastically.

A qualitatively similar result can be obtained for digg. However, we use the "digg" data in a slightly different manner, to distinguish between "truly" non-viral accesses, which results from different users independently[12] discovering about and liking a story, and non-viral accesses which result from being highlighted. digg, vimeo and many other websites hosting user-generated content typically highlight a few items on the front page. Such items can be expected to be popular in terms of number of accesses. Furthermore, many of the accesses are also likely to be from people unrelated to each other (non-viral). To discount this effect, we examine stories on digg *before* they are highlighted on the front page as "popular" stories. Digg de-

[12]i.e., without Digg's help. The story could have been highlighted elsewhere, e.g., on another Swebsite, but the entire user base of Digg may not have heard of it.

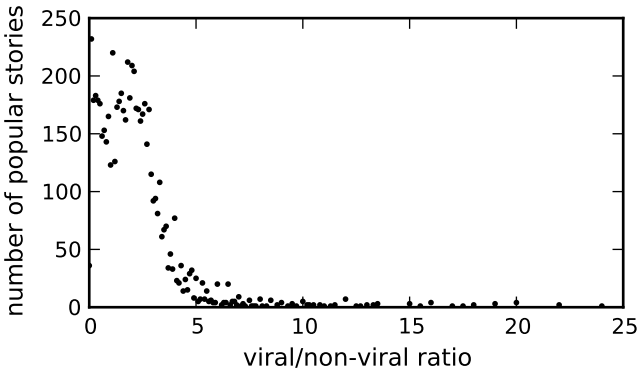


Figure 3.10: ***Non-viral accesses predominate even in popular stories which are not highlighted****: x-axis shows ratio of number of viral accesses to non-viral accesses. Y axis shows the number of popular stories that have the corresponding ratio. To clearly show zero valued points, the origin (0,0) is slightly offset from the corners of each graph. (Digg trace, focusing only on accesses made before digg marks a story as popular and highlights it on digg's front page.)*

notes a small subset of stories as "popular" using an undisclosed algorithm that reportedly [Dig]

> "takes several factors into consideration, including (but not limited to) the number and diversity of diggs, buries, the time the story was submitted and the topic."

Using Digg's API, we obtained information on which stories are marked popular and the timestamp when the story was made popular. We then examine diggs to the story before the timestamp. The result, shown in Figure 3.10, indicates that even in this subset of accesses for "yet to be marked popular" stories, a predominance of viral accesses greatly decreases the possibility that a story is popular.

We conjecture that while there may be individual successes of "viral marketing" strategies, in general, a story which spreads only by viral propagation remains stuck in one part of the social network. In contrast, an inherently popular story is independently (i.e., non-virally) seeded at several places in the social network and therefore becomes popular. Note that

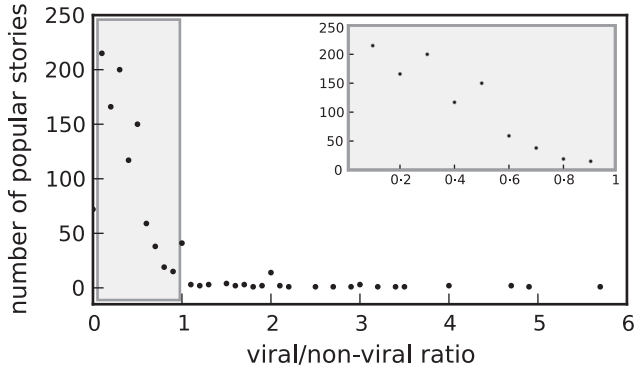


Figure 3.11: *The relation between viral/non-viral ratio and popularity among stories marked by digg as popular. Observe that digg's well-tuned algorithm for detecting story popularity can be approximated simply by detecting whether the non-viral component is more than the viral. The inset figure zooms in on the gray region on the main axis (between 0 and 1 on the x-axis), showing an almost linear dependence between the viral/non-viral ratio and the number of popular stories.*

even the local success of viral strategies could be partly attributed to homophily, the tendency of friends to like the same things, rather than users actively influencing their friends to watch a video or news story [AMS09].

Finally, Figure 3.11 counts the ratio of viral/non-viral accesses in stories *deemed by digg* to be popular. Interestingly, this graph has a knee around a viral to non-viral ratio of 1. There are hardly any popular stories with this ratio greater than 1, i.e., when a story has more viral than non-viral accesses, the probability that the story is popular is almost negligible. When the viral to non-viral access ratio is less than one, the probability that the story is popular is proportional to the ratio[13]. This is clearly seen from the inset graph which zooms in on the gray region (between 0 and 1 on the x-axis) shown on the main graph. In other words, it appears that digg's well-tuned algorithm for marking a story as popular can be closely approximated by a simple algorithm that marks a story as popular with a probability proportional to the ratio of viral to non-viral accesses. In §3.4,

[13]The actual counts shown in the figure can be re-scaled by the total number of popular stories to obtain the probability.

we will adopt a similar strategy to mark a story as popular or unpopular.

Corroboration: In effect, this section looked at information propagation on digg and vimeo. Others have looked at information propagation in Flickr [CMG09b], Amazon [LAH07] etc. As in our study, they find evidence that purely viral propagation of information is largely ineffective. Previous studies have also found a predominance of viral accesses. [WIVB10] finds that nearly 45.1% of views are "social" (i.e., by directly typing the obscure URL or from external links or embeds in social networks). [LG10] examines the spread of information in digg and twitter and finds a predominance of viral accesses. In digg, it is found that votes come from fans with a probability of 0.74 for stories which have not been marked popular. After being marked popular, the probability decreases to around 0.3.

3.4 Saving energy in the storage subsystem

Energy expenses form a major part of the operating costs for content delivery networks [QWB+09]. The mushrooming of HD quality user-generated content [Big09] is increasing both the amount of storage required, as well as the energy costs of maintaining the stored content on the Web. This section develops a strategy to save energy in the storage subsystem by exploiting the skewed access patterns observed earlier. The essence of the strategy is to segregate the highly popular videos from the unpopular ones. Disks containing unpopular videos can then be put in low power mode when there are no accesses for the videos contained, thereby saving energy. We utilise the ratio of viral to non-viral accesses of a video as a leading predictor of popularity, in order to classify videos as popular or unpopular.

3.4.1 Understanding the energy costs of storage

Before we begin, we offer a simple back of the envelope calculation to put the energy costs of storage into perspective. Consider the following statistic recently announced by YouTube [Wal10]: every minute, over 24 hours of videos are being uploaded. At a conservative (for HD) bitrate estimate of

3–5Mbps, this translates to 44–74 TB of data per day. Assuming conventional 512 GB SATA disks with a 12 W operating power[14], storing a day's uploads takes 25–42 kWh[15] of electricity per day. To put this into perspective, consider that ofgem figures put the typical daily consumption of UK households at around 9 kWh [Ofg11]. Thus, just a single day's worth of uploads consumes 2.8–4.7 times the electricity of a UK household.

Note that the above calculation has been extremely conservative. YouTube's 2010 limits of 2GB file uploads and 10 minute long videos allows for a full HD Video encoding of 27.3 Mbps. Thus, the required storage capacity and energy could be up to 9 times larger. If off-the-shelf components are being used and capacity can only be added in terms of additional blades onto a server rack, the calculations would have to consider the entire power consumed by a server, including the CPU, which is much more power hungry than storage. Depending on the processor, this could add an additional factor of 10 or more to the energy costs [HP07]. Together, these two factors could inflate the energy figures by a factor of 90. In other words, with a less conservative calculation, a day's uploads to YouTube could consume as much power as 250 UK households.

Further, observe that each day, *another* 44–74 TB of data get uploaded, according our earlier conservative estimate. Thus, in a system that starts from scratch and adds new videos at the same rate as YouTube, the daily energy consumption for storage doubles on the second day; the consumption for the third day is triple that of the first day, and so on. It is not hard to see that in just 13–17 days the storage system consumes as much energy (3300 kWH) as a UK household would over the course of an entire year, even if it is operating without any head room for additional storage.

[14] Although Solid-State drives consume much less energy, they are currently too expensive for servers [NTD+09]. Therefore, we only consider SATA drives in this section.

[15] The ranges presented in the next few numbers are calculated based on whether the bitrate is taken as 3 Mbps or 5 Mbps.

3.4.2 Data arrangement schemes for saving energy

Next, we investigate the use of intelligent data arrangement to save energy in the storage sub-system. From the perspective of the storage sub-system, most of the accesses are reads. Further, as discussed above, there is a skewed popularity distribution across different items, and popularity ranks of individual videos can change dramatically. The solution we propose in this section exploits and makes accommodations for these features of our work load. Because the solution is work load specific, it is not guaranteed to work well under other conditions.

The basic idea is to arrange data so as to skew the majority of access requests towards a small subset of disks. Other disks can then be put in an idle mode at times when there are no access requests for content on those disks. The highly skewed nature of the number of accesses (Figure 3.7) suggests the basic data arrangement strategy of organising data according to their popularity. Within this space, we explore two alternatives: The first, Popular Data Concentration, uses the Multi Queue algorithm to maintain a popularity ordering over all files. The second, SpinThrift, uses the relative proportions of viral and non-viral accesses to distinuish popular and unpopular data. Within each class, data is ordered so as to minimise the number of migrations.

Our simulations indicate that both Popular Data Concentration and SpinThrift end up with similar energy consumptions, ignoring energy involved in periodically migrating data. SpinThrift results in significantly fewer data migrations, thereby requiring lesser energy overall, as compared to Popular Data Concentration.

As detailed in the evaluation below, disks take a non-trivial amount of time ($\approx$ 6 secs) to move from the low-power mode back to full-power. However, we will assume that this latency is hidden simply by buffering the first 6 seconds of each video in memory or on a separate disk which always runs at full-power. This head optimisation technique applies to both Popular Data Concentration and SpinThrift.

Popular Data Concentration

We use Popular Data Concentration (PDC) as specified by Pinheiro and Bianchini [PB04], with minor changes[16]. PDC works using the Multi Queue (MQ) cache algorithm to maintain popularity order. The MQ cache works as follows [ZPL01]:

There are multiple LRU queues numbered in Q_0, Q_1,..., Q_{m-1}. Following Pinheiro and Bianchini, we set $m = 12$. Within a given queue, files are ranked by recency of access, according to LRU.

Each queue has a maximum access count. If a block in queue Q_i is accessed more than 2^i times, this block is then promoted to Q_{i+1}. Files which are not promoted to the next queue are allowed to stay in their current LRU queue for a given lifetime (The file-level lifetime is unspecified by Pinheiro and Bianchini. We use a half hour lifetime). If a file has not been referenced within its lifetime, it is demoted from Q_i to Q_{i-1}. Deviating from PDC, we do not reset the access count to zero after demoting a file. Instead, we halve the access count, giving a file a more realistic chance of being promoted out of its current queue during our short half-hour lifetime period. In our method, all files within a queue Q_i obey the invariant that their access count is between 2^{i-1} and 2^i.

Periodically (every half hour, in PDC), files are dynamically migrated according to the MQ order established as above. Thus, the first few disks will end up with the most frequently accessed data and will remain spinning most of the time. Disks with less frequently accessed data can be spun down to an idle power mode, typically after a threshold period of no accesses (17.9 seconds, as used by Pinheiro and Bianchini).

SpinThrift

The periodic migration of PDC, coupled with the changing nature of popularity ranks (see Figure 3.8) can lead to a large number of files being moved across disks at each migration interval. This is undesirable, not only be-

[16]PDC is specified in detail for handling block-level requests. At the file level, it is only specified that the operation is analogous to the block-level case. Details which are left unspecified for the file case are filled in by us.

cause it keeps the disks busy unnecessarily but because the migration itself consumes energy as the disks need to be on at full power during the process. To alleviate this, we propose SpinThrift, which separates the popular data from the unpopular, without imposing a complete ordering.

We use results from §3.3.6 to find popular data: SpinThrift uses the social network graph of the users, and keeps track of whether an access to a story is viral or non-viral. As before, an access by a user is *viral* if one of the friends of the user has accessed the same story before. In contrast, if the user who is accessing a story has no friends amongst previous users of the data, we deem that access as *non-viral*.

SpinThrift implements a policy of labeling a story as popular when the number of non-viral accesses exceeds the number of viral accesses. Popular stories are much fewer in number than unpopular ones. Therefore unpopular data is sub-classified based on the median of the median hour of access of previous users. Since the window of interest for unpopular stories is much smaller than for popular stories (see Figure 3.6), this organisation could reap additional locality benefits from time of day effects in user accesses.

The above classification is used to construct a ranking scheme for data, which is then used to periodically migrate data to different disks. The ranking works as follows: At the top level, popular stories are ranked above unpopular ones. Unpopular data are further ranked based on the median hour of previous accesses as above. Data within each sub group maintain the same relative ranking to each other as before, thereby reducing the number of migrations required.

Just as with PDC, a migration task runs every half hour, rearranging stories according to the above order. Similarly, disks which do not receive any access requests are spun down to an idle power mode after a threshold period of 17.9 seconds. The migration interval and idle threshold values were chosen to be the same as PDC for ease of comparison below.

Note that this classification is solely on the basis of the proportion of viral accesses and completely ignores recency of access. This can lead to inefficiencies as in the following scenario. Consider a video that was classified as popular because its non-viral accesses exceeded the number of viral ac-

cesses. This video will continue to be classified as popular even if it receives no further accesses (viral or non-viral). Thus, it will be arranged alongside more recently accessed popular videos, occupying valuable space on a "hot" disk despite not receiving any accesses.

SpinThrift does not attempt to address this kind of inefficiency because of the following reasons. First, the interest window (the time duration between first and last access) of popular items seems uniformly distributed (see Figure 3.6). Thus, in contrast with unpopular videos, it is not straightforward to predict the last access of a popular video. Second, the number of stories that end up being classified as popular is much smaller than the number classified as unpopular. Hence the aggregate effect of inefficiencies on the energy consumption is small. One solution would be to have a fixed number of "hot" disks for popular videos. If the cumulative size of videos classified as popular is more than can fit into these disks, an additional LRU weighting factor could be used to decide which videos to discard from the "hot" set. However, we do not explore this option below.

Evaluation: Spinthrift vs PDC

We investigate the relative merits of PDC and SpinThrift using a simplified simulation scheme, similar to that used by Ganesh et al.[GWBB07]. We assume that there is an array of disks, sufficient to hold all data. A disk moves into operating mode when there is an access request for data held in that disk. After an idleness threshold of 17.9 seconds, the disk is moved back into an idle mode. Disks consume 12.8 Watts when operating at full power, as compared to 7.2 Watts in idle mode. The transition between modes consumes 13.2 * 6 = 79.2 Joules. These details are summarized in Table 3.2.

Our simulations are driven by access requests from the vimeo and digg data sets described before. Note that our data sets do not indicate the sizes of the data as stored on the disk. We account for this using disks of different capacities, by counting the number of data items that fit on each disk. Extending the example from §3.4.1, notice that only about 1,400 10

Operating power	12.8 Watts
Idle power	7.2 Watts
Idle threshold	17.9 Secs
Transition power	13.2 Watts
Transition time	6.0 Secs

Table 3.2: *Power parameters*

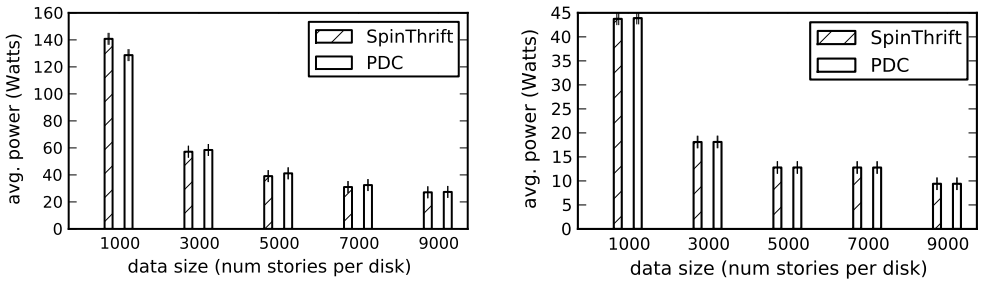


Figure 3.12: *Relative energy profiles: SpinThrift consumes roughly same energy as Popular Data Concentration, without taking migration energy or number of migrations into account. Errorbars show 95% confidence intervals. Left: Digg trace. Right: Vimeo trace.*

minute long clips can be stored on a 512 GB disk if the bitrate is 5 Mbps. Based on this, we use conservative numbers of between 1000 and 10,000 data items per disk.

Our first simulation uses requests drawn from a random 3 day interval from the digg trace, and measures the average power consumption on the final day after an initial warm up period. For the vimeo trace, a random 30-day interval is used, and the average power consumption on the final day is measured. Figure 3.12 shows the relative power profiles for disk arrays using disks of different sizes. Observe that both PDC and SpinThrift require similar power, with PDC doing slightly better because it tracks the popularity more accurately.

The above experiment does not take into account the energy involved in data migration. The next simulation, in Figure 3.13, plots the number of files needing to be migrated for a disk which can accommodate 5000 data

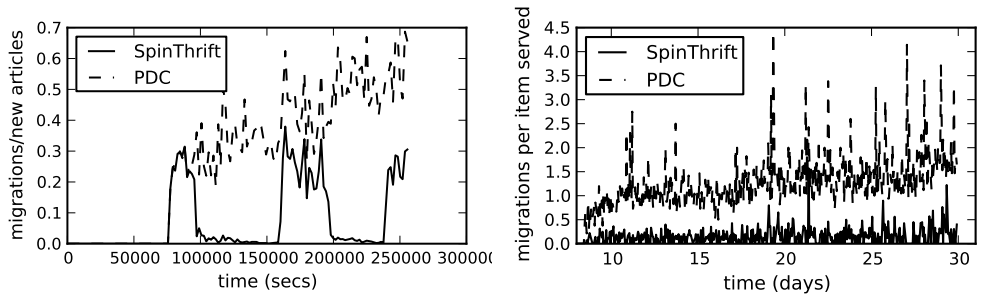


Figure 3.13: *SpinThrift has significantly fewer migrations than Popular Data Concentration (PDC). Left: Digg trace. Right: Vimeo trace.*

items. The number of files migrated at every half-hourly migration point is shown as a fraction of the total number of new articles that arrived during the interval.

Observe that after an initial warm up period, the number of files requiring migration under PDC keeps growing continuously. This is a result of the changing popularity of content items – as new articles are introduced into the system and become popular, they move to the head of the list, requiring all articles before them to be moved lower on the ranking list. In contrast, SpinThrift requires many fewer migrations.

The energy savings are strictly positive, but the precise amount of energy saved depends on the sizes of the items being migrated and we lack this information about the work load. The energy saved is proportional to the time taken to transfer the excess items, which in turn is proportional to the number of items transferred, for a given average item size and disk bandwidth. Thus the energy savings can be readily calculated, and is always proportional to the number of migrations.

In summary, the evaluation above suggests that SpinThrift is able achieve a similar power profile as PDC, with many fewer migrations. We emphasise that our simulations are highly simplified. In particular, they do not consider the effect of repeated data migration or power cycling on disk reliability. We also assume that disks are not bandwidth constrained, i.e., that the most popular disks can support all the data requests without increasing request latency. We also do not directly model the costs of increased

latency for accesses directed at disks in idle/low-power mode.

3.4.3 Related approaches

Our scheme improves upon Popular Data Concentration, which was designed explicitly for highly skewed, zipf-like work loads[PB04]. Popular Data Concentration, in turn improves upon an older scheme, Massive Array of Idle Disks (MAID), which attempts to reduce disk energy costs by using temporal locality of access requests[CG02]. The above family of disk energy conservation techniques look at read-heavy work loads, and can be viewed as orthogonal to techniques which look at write work loads (e.g. [NDR08, GWBB07]).

Our work looks at the trade-off between saving energy in the storage subsystem using intelligent data arrangement policies, and the number of file migrations that the policy requires. Various other trade-offs exist in the space of disk energy savings and could be applied in addition to our technique, depending on the work load. For example, Gurumurthi et al. [GZS+03] looks at the interplay between performance and energy savings. Zheng et al.[CSZ07] examine the trade-off between dependability, access diversity and low cost.

3.4.4 Discussion

We conclude this section with a few comments about aspects of the solution:

Applicability The design and evaluation of SpinThrift only made two assumptions: that the popular content will predominantly have non-viral accesses, and that the content is stored on an array of disks (or some small unit of storage which can be independently put in low power mode). In particular, no assumptions were made about the delivery infrastructure used. Thus, the solution applies equally to a centralised content server, as it does to a highly distributed content delivery network.

Different application scenario In essence, the work described in this section draws directly upon our empirical observations in §3.3.6 to develop a predictor for which content will be popular and which will be unpopular. This section applied the predictor to segregate popular content from the unpopular content on an array of disks. Notice that it could be also be used in deciding which items to deliver using the expensive but effective traditional CDN infrastructure.

Criteria for segregating popular from unpopular The advantage of SpinThrift over schemes like PDC comes from making a binary classification between popular and unpopular, rather than maintaining a strict popularity ranking, which can change dynamically (§3.3.5). We make the classification based on the ratio of viral to non-viral accesses. It is equally possible to keep a strict popularity ranking, but make a binary segregation between popular and unpopular by establishing an arbitrary cutoff rank, declaring the top n items as popular and the rest as unpopular. Such a strategy is simpler, and could be more appropriate when the cost budget can only accommodate a fixed number n of popular items.

On the other hand, our method offers the advantage that by design, the cutoff point also identifies items which have predominantly viral accesses as unpopular. Thus, we can develop strategies tailored for viral work loads. For instance, §3.6 develops buzztraq, a selective replication scheme that cost-effectively delivers items which are accessed virally. Using the above suggestion, the "popular" non-virally accessed content can be delivered using traditional CDNs and buzztraq can be used for the "unpopular" tail items.

3.5 Tailoring content delivery for the tail

Having seen how to cost-effectively store tail content, we now turn to the question of how to deliver such content. This section explores different design options for tailoring the content delivery infrastructure to suit tail

items. While we mostly end up choosing from well-known options, it is instructive to see how the characteristics of the work load (unpopular tail videos) can affect different aspects of a large complex system. This section also sets the stage for our own contribution, the selective replication scheme Buzztraq (§3.6), by showing how it fits into the larger content delivery infrastructure as a content placement algorithm.

3.5.1 Streaming vs. whole file downloads

There are two main ways to deliver content: whole file downloads, or streaming. Traditional file downloading has a disadvantage in that the entire file has to be downloaded before viewing can begin. The advantage, however, is that peer-to-peer techniques can be used to offload some of the burden of distribution to the clients. Furthermore, unlike streaming which has strict playback requirements, file downloading is the classical "elastic" bandwidth application.

Combining the advantages of the two approaches, a number of peer-to-peer streaming and video-on-demand solutions such as Bullet [KRAV03], SplitStream [CDK+03], RedCarpet [AGG+07], CoolStreaming [ZLLY05], PPLive [HLL+07] and UUSee [LWLZ10] have been proposed. Unfortunately, even though some of these solutions have been widely adopted and proven to work in the wild, they are more suited for popular streams with large simultaneous audiences who can help each other download, and do not work well for unpopular (tail) content.

With the advent of sites like YouTube, streaming from a server has become the overwhelmingly popular choice for users, since it allows playback to start nearly instantaneously. For the content provider and distributor, streaming offers much better control for the following reasons:

1. Streaming makes it easier to remove offending videos, e.g., those that violate copyright restrictions. YouTube runs a massive copyright verification program called Content ID[17] that matches uploads with known content. Copyright owners can block the upload, share in

[17] http://www.youtube.com/t/contentid

the advertising revenue resulting from the upload, or track viewership metrics.

2. Streaming allows targeted ads to be placed, depending on the viewer.

3. Streaming can provide accurate viewership statistics.

We note in passing that at an implementation level, the technology used for streaming appears to be going through an inflexion point. "True" streaming technology uses RTSP [SRL98] or Adobe's proprietary RTMP protocols [Rei09], which allows seeking to arbitrary points in the video and the quality of content to be adapted on-the-fly to suit available bandwidth. But this approach requires a specialised streaming server and will not work if the client is behind overly restrictive firewalls which block non-HTTP traffic. HTTP tunneling of RTSP/RTMP is an option, but remains susceptible to deep packet inspection and filtering. An alternate technology is progressive download, in which the content is sent over an ordinary HTTP connection from an ordinary Web Server and played back as it gets downloaded to a temporary folder on the client machine[18]. Progressive download requires the entire video to be downloaded in byte order. Most importantly, playback can stutter if available bandwidth decreases during playback. To combat this, a slight variation, called HTTP *adaptive* streaming, is increasingly being adopted. In HTTP adaptive streaming, the server stores the same video at multiple bitrate encodings and based on feedback about currently available bandwidth, switches the video to the appropriate bitrate on-the-fly [DCM10, CMP11, ABD11]. We stress that these variations in streaming technology does not affect the arguments above, in favour of streaming: that it is more suitable than other options for unpopular content, and that it offers more control than other technologies.

[18]Downloading to the viewer's disk makes it easier to violate protected rights. Therefore, streaming servers are typically employed if DRM is important.

3.5.2 Choices in the CDN infrastructure

Because of its stringent playback requirements, server-based streaming requires that the servers be close to the viewers. One way to mitigate this requirement is to replicate the content on a widely distributed set of servers, and direct viewers to the closest available server to them. We examine such Content Delivery Network infrastructures next and indicate preferred ways to deploy such infrastructures for unpopular content.

A brief primer

Before we delve into how CDNs can be tailored for tail content, we offer a brief primer on CDNs: CDNs work by maintaining multiple Points of Presence (PoPs), with each PoP containing clusters of *surrogate* servers, called so because they serve the client requests *instead* of the *origin* server owned by the content publisher. CDNs also typically incorporate an extensive measurement infrastructure that continuously evaluates network conditions, and thereby can determine the best surrogate server for a given client.

The measurement infrastructure feeds into a request routing infrastructure, which redirects client requests to the appropriate surrogate. Several redirection methods are known [BCNS03]. Many commercial CDNs perform the redirection at the DNS level—The authoritative server for the CDN provider directs different clients to different IP addresses by returning different results for the same CNAME lookup. Other methods include dynamic URL rewriting (embedded links to rich-media are changed to point to the appropriate surrogate) and HTTP redirection (using HTTP 3xx response codes).

More extensive overviews can be obtained from the works by Vakali and Pallis [PV06, VP03]. Pathan and Buyya [PB07] offer a taxonomy of the different ways in which CDNs can be tweaked. [Lei09] provides an overview of current directions and research issues. [Aka00, DMP+02, Mah01] discuss the working of Akamai, the first commercially successful CDN. A similar White Paper is available for Limelight Networks [Lim09]. Examples of

research CDN systems include Globule [PVS06], Coral [FFM04], CoDeeN and CoDeploy [WPP+04, PP04].

* * *

CDN providers have to make several choices in deploying content on their extensive infrastructures. At a high-level, they can be broken down into the following choices:

Server placement

In academic work, server placement has been treated as an instance of the Facility Location Problem [LSO+07], and has generally been solved using various optimisation approaches and heuristics [LGI+02, JJJ+02, JJK+02, CKK02, QPV02, KRS00].

In practice, surrogate servers are placed according to one of two design philosophies: "enter deep into ISPs", or "bring ISPs to home" [HWLR08]. The first approach is to locate the surrogates within the viewers' ISP networks. The second approach places surrogates in a few important peering points. Such points are well connected with large portions of the Internet, and therefore represent an effective way for the CDN to become proximate with a large number of potential viewers. The smaller footprint of the second approach makes it more easy to manage. However, it has been argued [Lei09] that there are a number of advantages obtained by entering deep into ISPs. First, economically, it may in fact be *cheaper* for the CDNs to "enter deep into ISPs", because many consumer ISPs host the CDN's servers for free (ISPs have an economic motivation for hosting the CDN server within their own networks: doing so reduces or eliminates transit traffic for flows to the CDN). Secondly, the "bring ISPs to home" approach relies a great deal on peering agreements between ISPs, and is therefore vulnerable to problems in the middle, which may temporarily disconnect parts of the Internet from each other. Finally, back of the envelope calculations show that the aggregate bandwidth available is higher in the first approach, because of its highly distributed presence.

In the evaluations below (§3.6.5), we experiment with two approaches.

The first approach assumes that the CDN has a server in each country. One effective way to arrange this is have a presence at the major peering point of the country. Most countries typically have just one or two major exchange points where the majority of country's ISPs peer. Thus this method is similar in spirit to the "enter deep into ISPs" philosophy and can obtain excellent coverage. In the second, we assume a CDN with surrogates in 10 locations, chosen to minimise the distance to the known locations of viewers in our trace. Lacking precise information about network distance, we place servers so that we minimise the geographic distance to the viewers. This method could be more cost-effective and easier to manage because of the limited number of locations, and is closer to the "bring ISPs to home" philosophy.

Content selection and placement

Content selection is the problem of selecting *which* content is to be replicated. The placement algorithm decides *where* (i.e., on which surrogates) the content goes. Similar to server placement, this problem has been studied extensively [PVS+05, Tse05, CQC+03, KRR02, VWD01, KPR99], with each optimising using a different heuristic, or optimising with different constraints (e.g. WAN bandwidth or space consumed on different replicas). Given a content item and the history of previous users who have accessed it, our solution in §3.6 gives a strategy for deciding where the content gets placed. While it is agnostic on which content is chosen for replication, the solution is designed to work better for content which has more viral accesses.

Content outsourcing: push or pull

Content outsourcing is the strategy used for staging the content from the origin server onto the surrogate servers [PV06]. The main choice here is between a pull-based approach, in which the surrogate fetches the content from the origin the first time a client asks for it, and a push-based approach, in which the content is proactively pushed to the appropriate surrogates. A further choice is whether a surrogate server can co-operatively fetch content

from another surrogate if a client asks for content it does not have. This is clearly beneficial but adds book-keeping complexity. Although many popular CDNs use un-cooperative pull-based content outsourcing [PV06], co-operative pushing can perform better in theory [VYK+02, CQC+03].

For popular content, the primary benefit from co-operative pushing derives not from the one hit obtained instead of the compulsory miss incurred by the pull-based approach, but because the surrogates can co-operate together [CQC+03], thereby allowing a more globally optimised placement under limited budgets.

For unpopular content, the cost of the compulsory miss of the pull-based approach is amortised over many fewer accesses. The geographic diversity of accesses (§3.3.4) compounds the difficulty. For instance, nearly half the unpopular items have one in three or fewer requests come from an entirely different country. These characteristics indicate using a push-based approach. Pushing has the additional advantage that content can be sent to the surrogates using an appropriately constructed overlay multicast tree, allowing WAN bandwidth to be conserved. Furthermore, because the push happens before client access, and is therefore not under a time constraint, cheaper and more efficient *background transport* protocols such as LEDBAT [Sha09] or TCP Nice [VKD02] can be used.

The research CDNs mentioned above use co-operative techniques. Surrogates in Coral [FFM04] co-operatively pull files from other close surrogates using a distributed sloppy hash table to locate appropriate servers. CoDeeN [WPP+04, PP04] is geared towards large files. The surrogate closest to the viewer serves as a proxy, fetching parts of the file from various other surrogates, reassembles and delivers to the viewer. Thus a surrogate can serve as a proxy both in forward and reverse modes. Globule [PVS06] pushes a fixed number, k, of replicas, and decides which surrogate gets a replica based on network positioning techniques [SPvS06].

3.6 Selective replication for viral workloads

The goal of this section is to devise a strategy that helps mitigate the difficulty of serving content not yet popular enough to be globally replicated. Given a history of previous accesses, we wish to predict the geographies of the next few accesses, so that replicas may be intelligently provisioned to minimise future access times. Our predictions are based on the means by which a user-generated content (UGC) object becomes known to potential users.

Knowledge of a UGC object can spread in two ways; *broadcast highlights* or *viral propagation*. The first happens when the UGC object is featured or highlighted on a central index page. Examples include being featured on the home page of the hosting sites (such as the featured videos list on YouTube); being promoted on an external social bookmarking site (e.g. if slashdotted, or featured on Digg, Reddit, Del.icio.us "hotlists", etc.); or ranking high on a google search. UGC objects in this class have to be popular according to the indexing algorithm used. Such high-visibility objects will likely be accessed many times and from all over the world, and are best served by replicating globally via CDNs.

The second possible means of propagation is by word-of-mouth, by sharing explicitly with a group of friends. This can happen through online social networks, emails, or out-of-band (or face-to-face) conversations. This kind of viral propagation has been termed as a *social cascade* and is considered to be an important reason for UGC information dissemination [CMAG08].

The links between friends on an online social network explicitly captures the *means* of propagation for social cascades. Furthermore, many social networking sites include approximate geography information. Thus, information about the friends of previous users and their geographical affiliations could be used to predict the geographical access patterns of future users.

In reality, content access is driven by a diverse mixture of random accesses and social cascade-based accesses. The content provider must place replicas to handle this access pattern by choosing the geographic regions in

which to place replicas of the content. The goal is to maximise the number of users who can be served by a local replica.

Two replica placement strategies are considered. The first, *location based placement*, uses the geographical location of recent users[19] to place replicas. The second strategy, which we call *social cascade prediction*, places replicas in regions where the social cascade "epidemic" is densest, as determined by the average number of friends of previous users.

In the specific case where a fixed number, k, of replicas is chosen, location based placement amounts to placing the replicas in the top k regions ranked by number of recent users. Social cascade prediction ranks regions by the number of friends of previous users and places replicas in the top k regions.

Our main result is that when the number of replicas is fixed, social cascade prediction can decrease the cost of user access compared to location based placement; i.e., more users are served by local replicas. The cost decrease is greatest when the cascade is responsible for most requests. Costs also decrease when cascades are responsible for fewer requests than random accesses.

Based on this, we have built a prototype system, Buzztraq, that provides hints for replica placement by using social cascade prediction. Intuitively, Buzztraq relies on the presence of a social cascade component, which makes the geographies of user requests non-random. Location based placement predicts that future requests will come from the same geographies as those of past requests. If instead, the requests shift to a new region, it is slower to react – until enough requests come from the new region to displace one of the old top-k, replicas are not moved. In contrast, Buzztraq's social cascade prediction strategy starts counting friends of previous users who are in the new region even before a request originates from the region. Thus, Buzztraq's strategy is faster to shift replicas and incurs fewer remote accesses.

[19]This can be determined from the IP address block of the user. Commercial CDNs may employ similar strategies [Mah01].

3.6.1 Related work

Recently, a number of studies [GALM07, CDL08, CKR+07, ZSGK09, AS10] have looked at various macroscopic properties of YouTube videos and concluded that caching of one form or another is beneficial for the web site performance. One of the earliest suggestions was that Web 2.0 objects such as YouTube videos have metadata information like user popularity ratings, which should be made use of in deciding which videos to cache [GALM07]. By studying YouTube traffic at the border of a university campus network, [ZSGK09] finds that there is no correlation between global and local popularity, and argues for proxy caching at the edge to reduce network traffic significantly. [CKR+07] also shows that proxies can help, using a simplified simulation with a synthetic access pattern generated from daily snapshots of view counts[20]. Both [GALM07] and [ZSGK09] find that peer assisted downloads can help, but the gains are highly sensitive to peers' uptimes. [SMMC11] considers the propagation of videos as a social cascade and develops a cache replacement policy that weights videos based on the observation that most cascades are geographically local. However, none of the above proposals are applicable for the videos in the tail, which are more geographically diverse than the popular videos (see §3.3.4) and less likely to be accessed predominantly from the same edge network. Further, most proposals implicitly or explicitly assume a pull-based CDN; a push-based CDN is more appropriate for the tail (§3.5.2).

[CDL08, AS10] find that popularity of related videos on YouTube are highly correlated, and because one viewing pattern is to see related videos one after another, suggests prefetching the heads of related videos while watching the first video. While this is limited to one viewing pattern, this solution can be adopted in conjunction with our cache placement algorithm to yield additional benefits.

[KMS+09] recommends that queueing delay (time spent waiting to transmit a packet at a loaded server) should also be taken into account when

[20] [CKR+07] has an updated journal version [CKR+09], which should be consulted for most aspects of this work. However, the effect of caching proxies and P2P is explored in more detail in the conference version [CKR+07].

deciding which server to which a client is being redirected. [AJZ10] looks at effects that ISP and content-provider policies such as early-exit routing and location-oblivious server selection might have on the overall traffic matrix between the ISP and content provider. Both these issues are beyond the scope of our work in this section. Here we develop a selective content placement strategy, and this can be adapted to work in conjunction to any solutions developed to problems in other areas of the CDN infrastructure, such as server selection and routing.

3.6.2 Inputs to Buzztraq

Buzztraq takes users' declared social links and geographic affiliations and produces hints on where to place replicas. Below we discuss how Buzztraq obtains the declared social links and geographic affiliations.

Buzztraq needs the declared social links of users. Previously this information was confined to social networking sites. However, APIs such as Facebook Connect[21] and MySpace Data Availability[22] are starting to make this data available to external web sites. These new APIs allow a user to login to external web sites using their identity on the corresponding social network. The external web site is authorised to retrieve and add related information about the user. Buzztraq uses the Facebook Platform API to retrieve each user's friends, and their publicly available affiliation information. We then attempt to deduce a geographic location for each retrieved affiliation using Google's geocoding API[23]. While users can choose to enter any arbitrary string for their affiliations, the geography lookup is mostly successful because the API corrects for common spelling mistakes and recognises different commonly used terms or abbreviations for popular locations. For example, our evaluations below (§3.6.5) use a dataset of around 20,000

[21] http://developers.facebook.com/connect.php. Last accessed in April 2009.

[22] http://developer.myspace.com/community/myspace/dataAvailability.aspx. Last accessed in April 2009.

[23] http://code.google.com/apis/maps/documentation/geocoding/index.html. Last accessed in April 2009.

Facebook users who collectively have 1,660 distinct affiliations, of which 1,181 (over 70%) could be translated into latitude-longitude coordinates.

Although social networks typically contain information about users' current geographic locations, this is not the right granularity for our purpose because users may access from a different location. In practice, we also find that the current location information is not entered by a vast majority of users on Facebook. Therefore, we use all the declared affiliations of the user in predicting the location of next access.

By giving equal weight to all locations, we are ignoring the complexities involved in word-of-mouth propagation, and may end up introducing false positives. For instance, depending on the nature of the content, a user may be much more likely to access item or spread information about it in only a subset of her affiliations/communities.

One mitigating factor is that Buzztraq hints are restricted to regions of the world. If the geographic affiliations of a user all belong to a single region, then the hints will be correct. Furthermore, to the extent that the user has more friends in the geographic region where she is most likely to spread a social cascade, Buzztraq hints will be correct even if the friends are not declared explicitly in the social network.

Going forward, it is perfectly possible that social networks would be able to provide more accurate location information of users, not only in terms of geography, but also in terms of which ISP's network users are likely to access content from. First, users are engaging more and more with social networks. For instance, Facebook reports that over 50% of its user base logs on at least once a day [Fac]. Thus, the IP address from which a highly engaged user initiated the last session to Facebook is likely a good approximation of their current network location. A second source for user location comes from geographic social networks like Gowalla[24] and foursquare[25] on which users are voluntarily disclosing their current locations.

[24] http://www.gowalla.com
[25] http://www.foursquare.com

3.6.3 Predicting future accesses

The UGC provider specifies a single UGC object or a collection of related content by a content-id. Buzztraq keeps note of users accessing content identified by each content-id. Using information about these users' friends and affiliations, hints are generated on where to place replicas of the content.

For purposes of exposition, we discuss how this is done in the context of a possible UGC provider architecture. Note that the basic concepts underlying Buzztraq do not rely on this specific model.

A basic UGC provider model

We assume that users first log in at a central site and are redirected to one of a fixed number of replica sites to access the requested rich media UGC. Redirection to a replica site local to the user is considered to be preferable. For instance, this may enable better delay and jitter guarantees.

The central site runs Buzztraq, which obtains the user's declared friends and geographical affiliations as detailed in Section 3.6.2. After each user access, Buzztraq uses this information to predict the top-k regions from which future accesses to the requested UGC are likely to originate. The UGC provider can use these hints to decide where to locate each UGC object. Specifically, in our evaluation, we look at the case where each content object is placed in a fixed number of replicas.

Buzztraq predictions are to be treated as *hints* for replica placement. While hints are generated after each access, the provider is not required to reconfigure replica locations each time. This is not critical since the set of top-k regions is not expected to change frequently.

There may be regimes where it is practical to reconfigure replicas after each user. For instance, suppose each region contains all the UGC objects hosted by the provider, but only those most likely to be accessed are kept in main memory. Buzztraq hints can be used to decide which k regions will keep a given object in main memory. Changing this set is less expensive than shipping the content over the network, and could potentially be done after each user access.

3.6.4 Generating replica placement hints

Without social network links, the best a UGC provider can do is **location based placement**. This strategy keeps per-region histories of user accesses and places replicas in regions which have historically contributed the maximum number of users. This is similar to the placement algorithm used in the Globule Collaborative CDN, which breaks the network into cells, and places each item in the top-k cells where it has been accessed before [SPvS06].

Buzztraq uses an alternate strategy, **social cascade prediction**, which predicts the next accesses by taking social cascade into account. If user accesses are being driven by word-of-mouth propagation, we expect that some of the future accesses will be made by friends of previous users. Thus, our strategy is to place the replicas in the top-k regions where the number of potential future users, as measured by the number of friends of previous users, is highest.

Unlike location based placement, which only counts the number of *previous* users in each region, social cascade prediction additionally attributes non-local friends to their appropriate regions as potential *future* users.

If the cumulative number of friends of previous users ranks a new region in the top-k, Buzztraq predicts that more accesses will originate from this region, owing to social cascade. Location based placement will not rank this new region in the top-k until the new region generates enough requests to displace one of the previous top-k. During this transition period, location based placement will cause non-local replica access for users from the new region, leading to higher costs.

If a user's friends are local to her region, then both social cascade prediction and location based placement will recommend placing replicas in the same regions.

The approach of counting friends of previous users is similar to the concept of the reproductive number R in epidemiology, which measures the average number of secondary cases caused by each case of an infectious disease [AM79]. If $R > 1$, then the infection will be sustained in the region.

In this language, we are counting the number of potential secondary accesses that could be caused by a previous infected user. Buzztraq's output of top-k regions gives the regions where the intensity of infection is highest. Since each access generates new hints, only the current infection intensity is counted. We do not normalise to predict whether the infection will be sustained.

3.6.5 Evaluation

We evaluate the relative costs of location based placement and social cascade replication using a synthetic workload generated with user requests coming as a mixture of social cascade and random accesses. We compare the relative costs of the two different strategies for replica placement and show that, on average, social cascade prediction can help place replicas closer to the viewers, than location based placement.

Workload

Evaluation is driven by a simple workload. It is not intended to capture all the complexities of user request arrivals. User accesses from across the globe are assumed to arrive at the central site in some serialisable fashion. Only the sequence of requests matters; there is no notion of real time. We also assume that user accesses are generated either by a social cascade or by a random process. Additionally, each user performs at most one access.

The main goal of the workload is to have a tunable amount of social cascade-based user accesses. User requests are assumed to arrive because of social cascade with probability p_s, or as a result of a random access, with probability $(1-p_s)$. Thus, with probability p_s, the next user is chosen to be a friend of a previous user; with probability $1-p_s$, the next user is a random user. We incorporate a notion of recency in the social cascade process – only friends of the last `TTL` users are chosen for non-random accesses.

Given this workload, the UGC provider has to place replicas so that access cost is minimised. If the provider has a replica in the region of the next user, it is deemed to be a local access; otherwise it is a remote access.

The cumulative cost is measured by a cost function which is arbitrarily defined so that a remote access is $c_r = 20$ times costlier than a local access. The provider's goal is to minimise the total cost of all user accesses. Note that any value of $c_r > 1$ will capture the relative difference in the long term costs of two replica placement strategies.

User characteristics

In addition to the vimeo dataset described in §3.3.2, we also evaluate using a dataset of Facebook users, drawn from 20,740 Facebook profiles from the Harvard network with profile IDs $< 36,000$. There are 2.1 million links between them, with a mean degree of 63 and a maximum degree 911.

We derive geographical locations for users using Google and Yahoo Geocoding APIs to derive latitude-longitude co-ordinates for strings in the user's profile which represent location information. In vimeo, each user has a single location string. In Facebook, users can declare multiple affiliations. If more than one of these can be successfully translated into a geographic location, we assume the user can be located in any of these locations.

Server placement

The server placement policy directly affects the costs incurred. We experiment with two policies. For the vimeo dataset, we assume that each country has a different server. For the smaller Facebook dataset, we examine the implications of choosing a fixed number (10) of servers.

In the Facebook dataset, to obtain an optimal placement for the 10 servers, user locations are treated as points and are clustered into 10 regions using the k-means algorithm [Mac67]. To decide cluster membership, Vincenty's formula is used to calculate geodesic distances between points [Vin75]. The clusters define fixed regions across the world, and the UGC provider can place replicas in any of the identified regions.

Our algorithm found separate clusters for North Africa, South and Central Africa, Europe and the Middle East, Australia and the Far East, South

America, and the Indian Sub-continent. Predictably, there were multiple (4) regions within the United States.

Number of replicas

The results will also be sensitive to the replica budget allowed for each item. For instance, the Facebook dataset contains more declared affiliations for places within USA than any other country. Thus a safe strategy would be to concentrate all replicas in US regions.

To prevent such naïve strategies from succeeding, we fix the number of replicas allowed to $k = 3$. Since the clustering gives us four different US regions, any placement strategy is forced to choose at least one US region to serve remotely. This counteracts any inherent geographical bias and brings out the relative difference in the costs of the two strategies.

The policy of allowing each country to have a server location is less sensitive to this kind of bias. However, we still fix a replica budget of $k = 3$ for the vimeo dataset. In other words, only the top three countries chosen by a replica placement strategy can have local accesses.

Relative cost of social cascade prediction

In effect, location based placement uses the history of previous accesses to predict future accesses. Social cascade prediction explicitly captures a user's friends as potential future users. Thus, social cascade prediction should be expected to work better if there is a strong social cascade component driving the user accesses.

To verify this, we simulate the same workload (with $p_s = 0.5$) on two UGC objects which are placed using social cascade prediction and location based placement strategies, respectively. We measure the cumulative cost of serving the first n requests, as n increases.

If the UGC provider is able to serve more users local to regions where it has placed replicas, its cost is lower. Figure 3.14 plots the result. The x-axis shows n and the y-axis plots the ratio of the cumulative costs of serving the first n requests using the social cascade prediction strategy to the

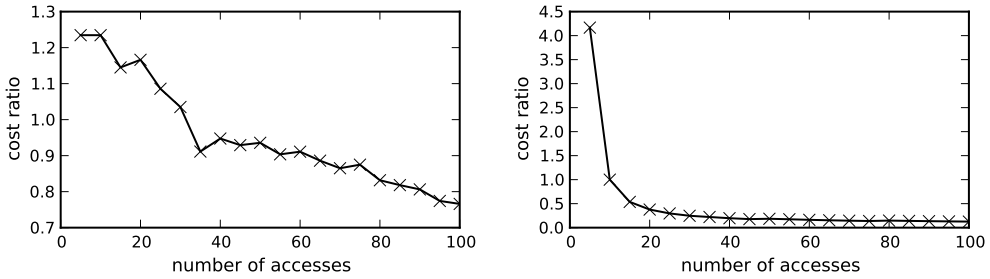


Figure 3.14: *Cost comparison of social cascade prediction to location based placement. $p_s = 0.5$. When cost ratio is less than 1, social cascade prediction is cheaper. Left: Vimeo. Right: Facebook.*

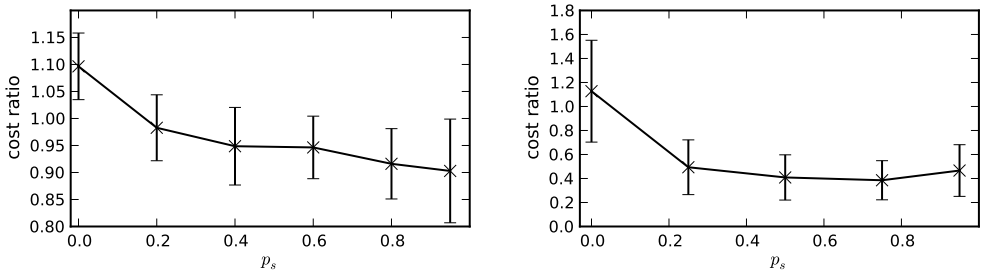


Figure 3.15: *Average cost ratio for different values of p_s. As p_s, the probability that social cascade drives user access, increases, Buzztraq's strategy becomes more efficient. Left: Vimeo. Right: Facebook. Errorbars show one standard deviation across 25 simulation runs.*

cumulative costs using location based placement. Initially, when there is no discernable social cascade, location based placement outperforms. However, as the number of accesses increases, social cascade prediction becomes the more efficient strategy.

Figure 3.15 examines the relative efficiency of the two strategies for different values of p_s, the probability that the next user accesses because of a social cascade. A sequence of 100 requests are performed, and the relative cumulative cost of serving the last ten requests is measured, for different values of p_s. The cost ratio remains less than 1 (i.e. social cascade prediction is cheaper) for all the p_s values we measure. As the probability of a social cascade choice increases, the cost ratio drops, showing that the

social cascade prediction does detect the underlying process generating the user inputs.

3.7 Conclusions

Web 2.0 sites have made networked sharing of user generated content increasingly popular. Serving rich-media content with strict delivery constraints requires a distribution infrastructure. Traditional caching and distribution algorithms are optimised for globally popular content and will not be efficient for user generated content that often show a heavy-tailed popularity distribution. New algorithms are needed.

This chapter showed that information encoded in social network structure can be used to predict and act upon access patterns which may be partly driven by viral information dissemination. Specifically, we developed Buzztraq, a strategy for saving costs incurred for replica placement, and SpinThrift, an energy saving strategy. Buzztraq uses knowledge about the number and locations of friends of previous users to generate hints that enable the selective placement of content replicas closer to future accesses. SpinThrift saves energy in the storage subsystem by segregating the unpopular (predominantly viral) content from the popular (predominantly non-viral) content and placing them on disks which are put in a low power mode when not being accessed.

CHAPTER 4

Data delivery properties of human contact networks

In this chapter, we switch gears and focus on the second case study on adding social network support for data delivery infrastructures (cf. Problem 1.2). This case study explores the connectivity properties of the Pocket Switched Network (PSN) [HCS+05], a radical proposal to take advantage of short-range connectivity afforded by human face-to-face contacts. The people act as the (mobile) nodes in this network, and data hops from node to node via short-range transfers during face-to-face contacts. The PSN creates paths over time by having intermediate nodes ferry data on behalf of the source. Our goal is to characterise the achievable connectivity properties of this dynamically changing mileu.

Summary of findings

At a macroscopic level, human contacts are found to be not very efficient at achieving any-any connectivity: contacts between node pairs which meet frequently are often not useful in creating new paths, and global connectivity crucially depends on rare contacts. This result is important not only because of its impact on data delivery but because it offers direct empir-

ical evidence for Granovetter's theory on the importance of weak ties in information diffusion [Gra73].

Connectivity is also highly uneven, with significant differences between time windows of similar duration, as well as between different sets of nodes within the same window: in time windows in which a large fraction of data gets delivered, rather than all nodes being able to uniformly reach each other with the same efficiency, there is usually a large *clique* of nodes that have 100% reachability among themselves; nodes outside the clique have a much more limited reachability between them. We show how to identify such time windows and the nodes involved in the clique by computing a clustering co-efficient on the contact graph.

These inefficiences are compensated for by a highly robust distribution of delivery times. We examine all successful paths found by flooding and show that though delivery times vary widely, randomly sampling a small number of paths between each source and destination is sufficient to yield a delivery time distribution close to that of flooding over all paths. This result suggests that the rate at which the network can deliver data is remarkably resilient to path failures. Also, randomised path selection is shown to balance the burden placed on intermediate nodes and therefore further improve robustness. Finally, human contact networks are found to be efficient at "communitycast", in which a sender needs to multicast the same data to an identifiable community of nodes.

Chapter layout

§4.1 introduces the notion of a Pocket Switched Network. §4.2 describes the empirical contact networks we study and how we measure the connectivity achieved in such networks. By synthetically deriving modified traces from the original, §4.3 isolates individual properties and investigates the central question of what properties of human contacts determine efficient network-wide connectivity. It is shown that connectivity depends on the contact occurrence distribution, and that rare contacts are crucial. §4.4 examines different time windows of the same duration for variations in connectivity,

and demonstrates that connectivity achieved can be highly uneven. §4.5 studies properties of successful paths found by flooding over fixed duration windows. Based on this, §4.6 shows that the human contact network can be made robust to path failures. §4.7 demonstrates that multicast within a community is efficient. §4.8 discusses related work. §4.9 considers the implications of our findings for the design of PSNs and concludes.

4.1 The Pocket Switched Network

Consider a scenario in which Alice wants to send some item to Carol. If Bob happens to meet Alice first and then Carol, he could potentially serve as a messenger for Alice. Essentially, this method of communication exploits human contacts to create a path *over time* between a source and destination. Of necessity, the paths are constructed in a *store-carry-forward fashion*: Various intermediate nodes *store* the item on behalf of the sender and *carry* it to another contact opportunity where they *forward* the data to the destination or another node that can take the data closer to the destination.

Two factors normally limit the transfer of items or information in this manner over social contacts. First, each hop requires manual intervention. For example, Alice may need to request Bob, "When you see Carol, please give her this". Second, a knowledge of future contacts is tacitly presumed. In the above example, Alice and/or Bob need to know that Bob will be meeting Carol in the future.

[HCS+05] applies this method to the transfer of data and suggests workarounds to these two limitations: Manual intervention can easily be avoided by automatically exchanging data between mobile devices carried by the human actors. Widely supported short-range data-transfer protocols such as Bluetooth or Wi-Fi can be used for this purpose. If we do not have knowledge of future contacts, data can still be forwarded *opportunistically* from node to node, but without a guarantee that it will reach the intended destination.

This idea, of leveraging human social contacts, and using ubiquitious

mobile devices in people's pockets to create data paths over time, has been termed as a **Pocket Switched Network (PSN)**. At each contact opportunity, mobile devices carried by the humans exchange data using short-range protocols such as Bluetooth or Wi-Fi. By chaining such contacts, the PSN opportunistically creates data paths that connect a source and destination *over time*. Intermediate nodes in the path store data on behalf of the sender and carry it to the next contact opportunity where it is forwarded further.

4.1.1 Application scenarios

Clearly, opportunistic forwarding involves long and uncertain delays. Under what circumstances would it be useful? Here we present a few possible scenarios.

Designing for, and exploiting human mobility in this manner becomes important in situations where there is no alternative: when traditional networking infrastructure has been damaged (e.g. after a disaster), or does not exist (e.g. in remote areas and in some developing countries).

Even when traditional communication infrastructures such as cellular networks exist, it may be beneficial to use pocket switching when the producers and consumers of data are local to each other. For instance, consider a farmer selling livestock to local buyers[1], who wants to advertise the animal by making a high resolution video. It is likely to be very expensive both for the seller to upload the video to a central server, and for potential buyers to download it for viewing. Also, the central server may only be accessible via long-haul links for ease of administration. The capacity and bandwidth on long-haul links are often much lesser than the capacity and bandwidth of short-haul links which use technologies like Wi-Fi. In such cases, pocket switching will scale better.

By the same token, some applications, such as email, database synchronization and certain types of event notifications, are inherently asynchronous and can tolerate relatively long delays. Mobility can be exploited to provide multiuser diversity gains for such applications [GT02]. As

[1]Thanks to Prof. Huzur Saran for suggesting a version of this application.

with other gains, multiuser diversity increases the capacity of the network. A PSN can be therefore effective as a multi-hop "sneakernet" for high-bandwidth applications that can tolerate delays. New applications, such as seeking and finding information [MSN05], media file-sharing [MMC08], dynamic news feed updates [ICM09] and mobile micro-blogging [ACW10] are being proposed to take advantage of short-range connectivity between mobile devices.

4.1.2 An abstraction for the PSN

Abstractly, the Pocket Switched Network can be thought of as a temporally evolving contact graph. Each contact corresponds to a momentary undirected edge and involves a two-way data exchange between the node pair involved. Edges appear and disappear according to some underlying stochastic process that corresponds to the social contacts. The key insight behind the concept of pocket switching is that by integrating the effects of individual local contacts over time, we can achieve extended connectivity. Figure 4.1 illustrates this with the earlier example.

The sequences of edges (contacts) that occur constitutes a trace of the PSN. An empirical *contact occurrence distribution* can be defined for a trace, as the probability $p(f)$ that an edge (contact) constitutes a fraction f of the trace, i.e., the nodes that form the edge contact each other fn times in a time window with n contacts. In §4.3, we will examine how the contact occurrence distribution determines the connectivity achieved.

4.1.3 A note about diversity

At a high-level, the PSN can be thought of as creating connectivity by exploiting two forms of diversity. The first is the diversity of contacts made by different users. The PSN "stitches" new paths by combining contacts made by different users. The second source of diversity is time: Given a source and destination, over time, the same paths are recreated due to repeated contacts, or an alternate path could be found, in which some of the

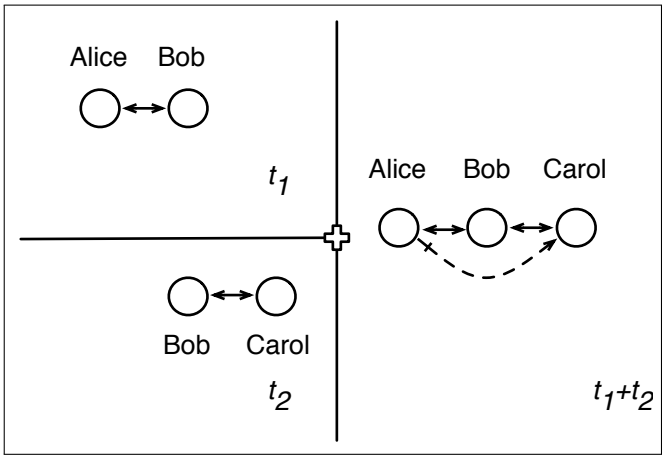


Figure 4.1: *Model of the Pocket Switched Network: Each edge occurs temporarily (panels on the left), but by integrating the effects across time, we can achieve additional connectivity (right panel). Note that there is an opportunity for two-way data exchange during local, face-to-face contacts (solid edges), whereas the opportunistically induced connectivity is directional (dashed edge). The direction of the edge is defined by the time order of contacts.*

intermediate nodes from the original set of paths are involved, but happen to get the data from a different upstream node.

The first of these two forms of diversity is more fundamental: While the recreation of paths over time can be crucial in repairing path failures, without diversity of contacts, additional connectivity will not be created over time. Furthermore, paths can typically be repaired much faster by exploiting all of the different paths forming over time due to user diversity.

Hence, this chapter focuses on how the data delivery properties of the PSN depend on the properties of the underlying process that corresponds to the creation of social contacts and hence governs the multi-user diversity that can be derived. To this end, we use a flooding process to study the diversity of paths that form *starting from some point in time, and in which each intermediate node gets involved at the first possible opportunity.* Each intermediate node remembers the first time it received data so that subsequent paths created by diversity over time are not recorded. The next section provides a more thorough description of this method of flooding,

and the "flood-tree" created in the process.

4.2 Setup and methodology

This section motivates the choice of traces, the simulation setup and the performance measures used.

4.2.1 Traces

We imagine the participants of a PSN would be a finite group of people who are at least loosely bound together by some context—for instance, first responders at a disaster situation, who need to send data to each other. Multiple PSNs could co-exist for different contexts, and a single individual could conceivably participate in several different PSNs. Note that this is in contrast to a single unboundedly large network of socially unrelated individuals as in the famous "small-world" experiment [TM69] that examined a network essentially comprising all Americans and discovered an average 5.2 (≈ 6) degrees of separation.

Our model that PSN participants form a cohesive group places the requirement that an ideal PSN should be able to create paths between arbitrary source-destination pairs. This is reflected in our simulation setup, where the destinations for each source node are chosen randomly. Also, our traces are picked to be close to the limits of Dunbar's number (=147.8, 95% confidence limits: 100.2–231.1), the average size for cohesive groups of humans [Dun93].

The first trace comes from a four week subset of the UCSD Wireless Topology Discovery [UCS04] project which recorded Wi-Fi Access Points seen by subjects' PDAs. We treat PDAs simultaneously in range of the same Wi-Fi access point as a contact opportunity. This data has $N = 202$ subjects. The second trace consists of Bluetooth contacts recorded from 1 Nov. 2004 to 1 Jan. 2005 between participants of the MIT Reality Mining project [EP]. We conservatively set five minutes as the minimum allowed

data transfer opportunity and discarded contacts of durations smaller than this cutoff. This trace has contacts between $N = 91$ subjects.

The subjects in the MIT trace consist of a mixture of students and faculty at the MIT Media Lab, and incoming freshmen at the MIT Sloan Business School. The UCSD trace is comprised of a select group of freshmen, all from UCSD's Sixth College. As such, we can expect subjects in both traces to have reasons for some amount of interaction, leading to a loosely cohesive group structure. Prior work on community mining using the same traces supports this [YHC07].

It is important to emphasize that our focus is solely on the capability and efficiency of the human contact process in forming end-to-end paths. The precise choice of the minimum data transfer opportunity is less important—it is entirely possible that a new technology would allow for faster node-node transfers. Indeed, our results are qualitatively similar for other cutoff values tested. Similarly, a different technology for local node-node transfers could have different "reach," allowing more nodes to be in contact with each other simultaneously. Nevertheless, the substantial similarities between results we present based on two different technologies and traces—the Wi-Fi based UCSD trace and the Bluetooth based MIT trace—gives us some confidence that the results below may be applicable beyond the traces and technologies we have considered.

4.2.2 Simulation method

All the experiments reported here are run using a custom discrete-event simulator and analysis engine written in Python ($\approx$ 5000 lines of code). At its core, the simulator takes as input a time-ordered trace of edges and replays them one-by-one, keeping track of the cumulative effects of edge occurrences[2]. Each simulation typically runs over different *windows* of the trace, where a window is defined as a consecutive set of edges. At the end of simulation, the analysis engine can be queried for various properties of interest.

[2]In effect, it tracks the flood trees created as detailed in §4.2.3

At the beginning of simulation, data is created, marked for a randomly chosen destination, and associated with the source node. An oracle with complete knowledge of the future can choose to transfer data at appropriate contact opportunities and thereby form the quickest path to the destination. To simulate this, we enumerate all possible paths found by flooding data at each contact opportunity, and choose the quickest.

The simulations run flooding using the following protocol, unless otherwise specified in the text: Each experiment runs over a time-ordered window of consecutive contacts happening between all possible node pairs the network. Each window is started at a random point in the trace. In some experiments, primarily in §4.3, the window ends after a fixed number (6000, unless otherwise specified) of contacts. In other experiments, primarily in §4.4 and §4.5, the window ends after a fixed time duration, say 1 week or 3 days. In all cases, the experiments are run multiple (typically 25 or more) times with different randomly chosen starting points, to verify that the results are not anomalous to some part of the trace. Where relevant, results are averaged, and confidence intervals are presented.

4.2.3 Flood tree

In our method of flooding, every non-destination node (including the source) forwards data to each non-destination node it meets that does not yet have a copy of the data. Each intermediate node receives a data item at most once. The destination node accepts from any intermediate node that forwards to it (at most once from each distinct node), but does not forward the data further. Note that this does not uncover all the paths that form over time in the contact graph. Rather, since each non-destination node receives data at most once, a tree of paths, rooted at the source, forms over time. We call this the *flood-tree*.

Figure 4.2 depicts an example of flood-tree evolution and discusses various corner cases that arise as a result of our definition. The flood-tree formulation was chosen to probe the diversity of paths that can form between a given source and destination, simply by involving new intermediate

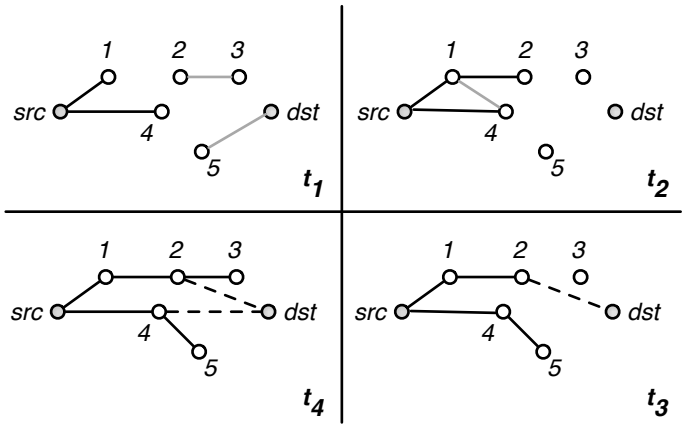


Figure 4.2: *Growth of the flood-tree from snapshots taken at four time instants, $t_1 < t_2 < t_3 < t_4$; not necessarily consecutive. The flood-tree is rooted at the source src, and is growing towards the destination node, dst. It can potentially include all nodes in the network, except the destination. At t_1, notice that edges occurring between nodes 2–3 and 5–dst do not grow the flood-tree because neither node in the edge is part of the flood-tree. In contrast, at t_2, edge 1–4 does not count because* both *nodes are part of the flood-tree. The flood-tree only grows when one of the nodes in an edge is part of the flood tree, and the other node is not. Observe that lower hop count need not mean faster paths: The three-hop path src–1–2–dst completes at t_3, before the two-hop path src–4–dst at t_4. Intermediate nodes continue to grow the flood tree even after delivering (edge 2–3 at t_4). On the other hand, some edges such as 4–5 at t_3 turn out to be unnecessary as the intermediate nodes find a more direct path at a later time instant (edge 4–dst at t_4).*

.

nodes. It avoids all repair/repeat paths that form over time, by having nodes record the first copy, and rejecting all paths formed by subsequent copies. Thus, nodes continue to forward to other intermediate nodes even after directly delivering to the destination (Edge 2–3 at t_4), whereas paths which involve a node that could have been recruited into the flooding process earlier are ignored (Path *src*–1–4 at t_2).

4.2.4 Performance measure

Consider the time-ordered sequence (with ties broken arbitrarily) of contacts that occur between all possible node pairs in the network. Since there are $N(N-1)$ quickest paths between different sender-destination pairs, a maximum of $N(N-1)$ contacts in the sequence of contacts act as path completion points. The actual number could be lesser because an intermediate node could deliver multiple items to a destination node in a single contact, completing multiple paths. Of these, Nd path completion points become "interesting" when there are d destinations per sender. Since the destinations are chosen randomly, we might expect that on average, if k path completion points have occured, the *fraction* of these that are interesting is independent of d: When d is greater, more data gets delivered after k path completion points, but there is also more data to deliver.

The above discussion motivates our method of measuring the efficiency of the PSN: At any point in the simulation, the *delivery ratio*, measured as the fraction of data that has been delivered, or equivalently, the number of "interesting" path completion points we have seen, is taken as a figure of merit. The more efficient the PSN is, the faster the delivery ratio evolves to 1, as the number of contacts and time increase.

We confirm our intuition in Figure 4.3, which shows that the delivery ratio evolves similarly, whether d is 1 or a maximum of $N-1$ destinations per sender. We note that the graph also represents the fastest possible evolution of the delivery ratio under the given set of contacts, due to the use of flooding.

Our second performance measure is time to delivery. The delivery ratio at the end of a time window is indicative of the fraction of node pairs connected during the window and is therefore a measure of the connectivity achieved by the network. If the empirical probability that the delivery time is less than t is r, then a fraction r of the data that eventually get delivered have been delivered by time t. The empirically observed cumulative distribution of delivery times can also be interpreted as the evolution in time of delivery ratio, normalised by the ratio eventually achieved at the

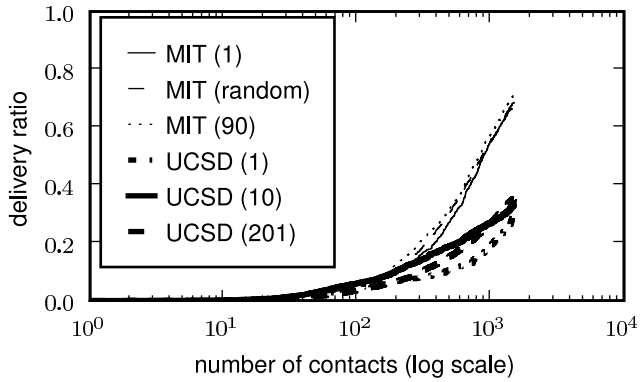


Figure 4.3: *Fraction of data delivered as a function of the number of contacts, for the MIT and UCSD traces (number of destinations per sender shown in brackets). The curves for each network are clustered together (the thin lines of the MIT trace are clustered together, but separately from the cluster of thick lines representing UCSD), showing that the delivery ratio evolves independently of the load.*

end of the time window, and thus represents the rate at which connectivity is achieved.

4.3 Delivery over fixed number of contacts

A PSN contact trace is determined by the distribution of contact occurrences and the time order in which these contacts occur. In this section, we examine how these properties affect delivery ratio evolution.

Our goal is to find out how efficient the network of human contacts is at achieving global any-any connectivity. Given two traces, the more efficient one will manage to achieve a given delivery ratio with fewer number of contacts. Our approach is to create a synthetic trace from the original trace by disrupting the property we wish to study. Comparing delivery ratio evolution in the original and synthetic traces over a fixed number of contacts informs us about the effects of the property.

Our main findings are that in both the traces we examine, time correlations between contacts that occur too frequently lead to non-effective

contacts in which no new data can be exchanged, and that the progress of the delivery ratio as well as the connectivity of the PSN itself are precariously dependent on rare contacts.

A rare contact involves much less time commitment from the nodes involved than a contact which occurs frequently, and can be classified as a "weak tie" in Granovetter's terms [Gra73]. His theory predicts that such ties are crucial in information diffusion processes that connect arbitrary node pairs. Thus, our results can be seen as empirical evidence of the correctness of his theory of the strength of weak ties.

4.3.1 Frequent contacts are often non-effective

To investigate the effect of the time order in which contacts occur, we replay the trace, randomly shuffling the time order in which links occur. Specifically, edges in the shuffled trace share the same timestamps as the original trace, but the sequence of edge occurrences is a random permutation of the original. Recall that the simulator plays back the trace as a time-ordered sequence of edge occurrences. Thus, the shuffled trace essentially is a random permutation of the original set of edge occurrences but with the times of successive events preserved from the original trace. For instance, given the sequence of edge occurrences (t_1, a, b), (t_2, a, b), (t_3, a, c) and (t_4, c, b), the random permutation might yield the following shuffled trace: (t_1, a, c), (t_2, a, b), (t_3, c, b) and (t_4, a, b).

Observe in Figure 4.4 that the curve marked "shuffled" evolves faster than "trace" implying that the delivery ratio increases faster after random shuffling. The random shuffle has the effect of removing any time correlations of contacts in the original trace. Thus the improved delivery ratio evolution implies that time correlations of the contacts in the original data slowed down the exchange of data among the nodes, causing them to be delivered later.

Manual examination reveals several time correlated contacts where two nodes see each other multiple times without seeing other nodes. At their first contact, one or both nodes could have data that the other does not,

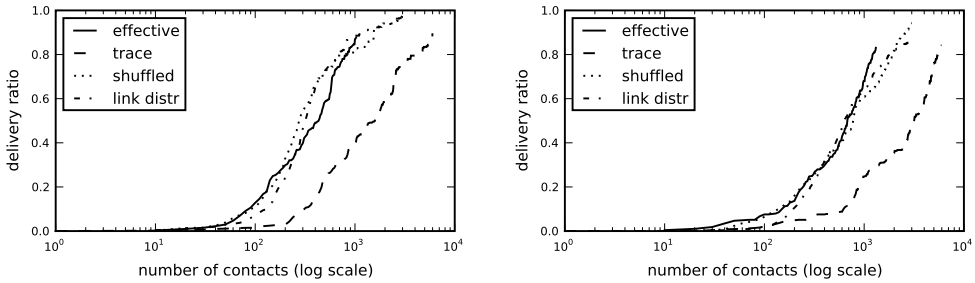


Figure 4.4: *Delivery ratio evolution for synthetically derived variants of MIT (left), UCSD (right) traces. 'Trace' is the original. 'Shuffled', the same trace with time order of contacts randomly shuffled. 'Effective' replays 'trace', counting only contacts where data was exchanged. 'Link distr' is an artificial trace with the same size and contact occurrence distribution as the original.*

which is then shared by flooding. After this initial flooding, both nodes contain the same data—subsequent contacts are "non-effective", and only increase the number of contacts happening in the network without increasing the delivery ratio.

To quantify the impact, in the curve marked "effective" on Figure 4.4, we plot delivery ratio evolution in the original trace, counting only the contacts in which data could be exchanged. This coincides well with the time-shuffled trace, showing that non-effective contacts are largely responsible for the slower delivery ratio evolution in the original trace.

Next, we construct a synthetic trace that has the same number of nodes as the original trace, as well as the same contact occurrence distribution. By this, we mean that the probability of contact between any pair of nodes is the same as in the original trace. The delivery ratio evolution of this trace, depicted as "link distr" in Figure 4.4, is seen to evolve in a similar fashion as the time-shuffled trace. This indicates that once time correlations are removed, the delivery properties are determined mainly by the contact occurrence distribution.

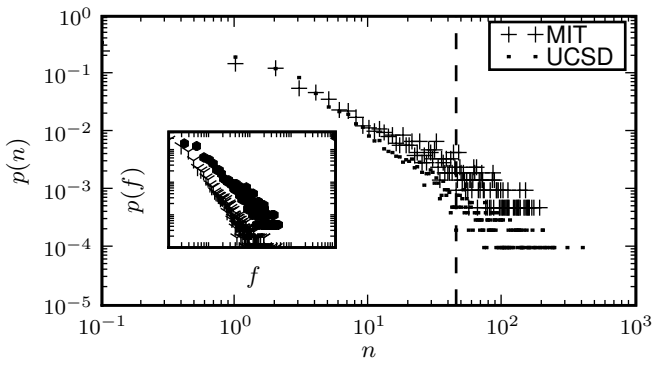


Figure 4.5: *Contact occurrence distributions (log-log): A random edge appears n times with probability $p(n)$. To the left of the dashed line at $n = 45$, the distributions for both traces coincidentally happen to be similar. The inset shows the difference when normalised by the number of contacts in the trace. In the inset, a random edge constitutes a fraction f of the trace with probability $p(f)$.*

4.3.2 Connectivity depends on rare contacts

The fact that three different traces (shuffled, effective, and link distr), which are based on the same contact occurrence distribution, essentially evolve in the same manner leads us to examine this distribution further.

Figure 4.5 shows that the contact occurrence distribution has both highly rare contacts (involving node pairs that meet fewer than ten times in the trace) as well as frequent contacts (nodes which meet hundreds of times). A randomly chosen contact from the trace is much more likely to be a rare contact than a frequent one.

Figure 4.6 shows that the rare contacts are extremely important for the nodes to stay connected. When contacts that occur fewer than a minimum cutoff number of times are removed, the number of nodes remaining in the trace falls sharply. This implies that there are a number of nodes which are connected to the rest of the nodes by only a few rare contacts.

On the other hand, removing the frequent contacts (by removing contacts occuring more than a maximum cutoff number of times) does not affect connectivity greatly. For instance, the MIT trace remains connected

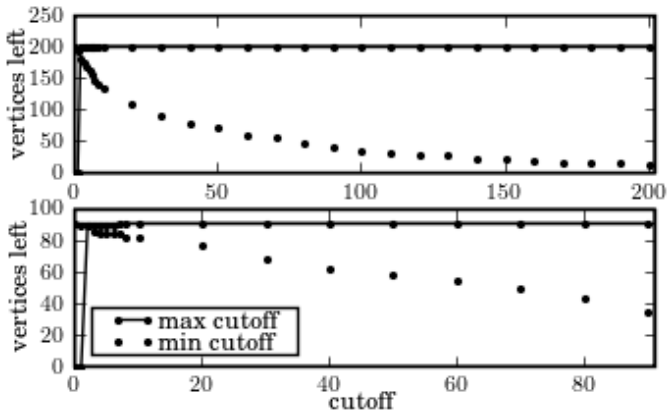


Figure 4.6: *Robustness to cutoff: MIT (below), UCSD (above). Max cutoff specifies a maximum cutoff for the frequency of contacts, thus removing the most frequently occurring ones. Min cutoff specifies a minimum frequency of contacts—removing the rarest contacts causes the number of nodes that are connected to drop precipitously.*

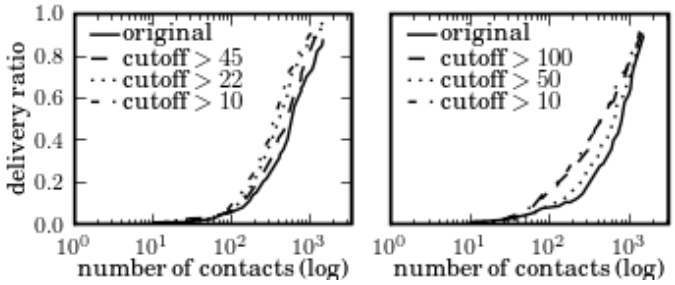


Figure 4.7: *Evolution of delivery ratio with contacts that occur more than* `cutoff` *times removed. MIT (left), UCSD (right). The network still remains connected, and manages to deliver data with* fewer *contacts.*

even when the maximum cutoff is as low as 10 (i.e., contacts occurring more than ten times are removed). This suggests that nodes which contact each other very frequently are also connected by other paths, comprising only rare edges.

Interestingly, Figure 4.7 shows that with the most frequent edges removed, achieving a given delivery ratio can take fewer contacts. This appears paradoxical but can be explained as follows: In terms of time, data delayed waiting for the occurrence of a rare contact still take the same

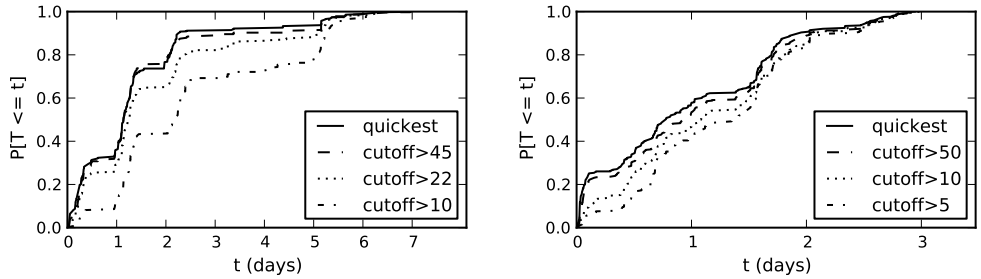


Figure 4.8: *Routing on rarest edges: CDFs of delivery time distributions with edges that occur more than* `cutoff` *times removed, in comparison with the* `quickest` *paths found by flooding. Left shows a random week-long time window from the MIT trace. Right shows a random 3 day window from the UCSD trace.*

amount of time to reach the destination, and data previously sent on paths containing more frequent edges alone are delayed, because they now have to be re-routed over rare contacts. However, the reliance on rare contacts allows "batch-processing": Each node involved in the rare contact has more data to exchange when the contact happens, thus decreasing the overall *count* of contacts taken to achieve a given delivery ratio.

Figure 4.8 shows the distribution of delivery times when routing only on the rare edges, which occur fewer than a specified cutoff number of times. Relying only on infrequent edges does increase the delay (the Cumulative Distribution of delivery times moves downwards and to the right). However, the increase in delay does not appear to be too much even when routing only on edges occurring fewer than five or ten times in a 3 day and 1 week-long time window respectively. We remark on this further in §4.9.

Finally, we briefly quantify the rate at which a connected component forms, by comparing it with the canonical Erdos-Renyi random graph model $\mathcal{G}_{n,p}$. $\mathcal{G}_{n,p}$ represents the ensemble of graphs which can be generated by independently and identically choosing each possible edge in an n-node graph with a probability p. Thus, each graph in $\mathcal{G}_{n,p}$ has an expected $pn(n-1)/2$ edges. One of the earliest and most striking results of the theory of random graphs is that for $np > 1$ and large n, the graphs in $\mathcal{G}_{n,p}$ almost surely contain a giant ($O(n)$ size) component, whereas for $np < 1$,

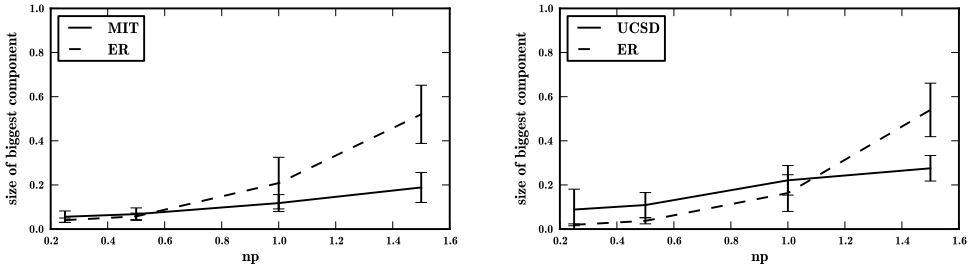


Figure 4.9: *Comparing the fraction of nodes in the largest components of random graphs generated according to two different models. Both models have the same fixed number of nodes n. Models of different sizes are generated by choosing a different number of edges. In the first model ("ER"), each possible edge is selected independently and identically with a probability p, giving an expected $pn(n-1)/2$ edges for a given p. This is compared with a second model in which a fixed number of edges, equal to the expected number above, are chosen, according to the contact occurrence distribution (Left: MIT, Right UCSD). For $np > 1$, the ER graphs start to have larger components, while the contact occurrence graphs fall behind. Error bars at one standard deviation distance.*

the largest component is $o(log(n))$ [Bol01, CL06].

The Pocket Switched Network can be viewed as creating a random graph over time by accreting the effects of individual edges, each of which is chosen from the contact occurrence distribution. To see the difference between the two approaches to choosing edges, we compare the size the largest component when $pn(n-1)/2$ edges are chosen randomly from the contact occurrence distribution with the size of the largest component for random graphs in $\mathcal{G}_{n,p}$ when each edge is chosen with probability p (which gives an expected $pn(n-1)/2$ edges in the ER model).

Figure 4.9 compares the largest components as increasing numbers of edges are chosen according to the two models. Each error bar on the figure shows the distribution of sizes obtained across 25 independently chosen samples. Each edge in $\mathcal{G}_{n,p}$ is chosen with equal probability but some edges occur much more frequently than others in the contact occurrence distribution. Notice that as np increases beyond 1, the ER graphs start to have larger components, consistent with theory. However, the graphs drawn

from the contact occurrence distributions do not experience in this jump in connectivity, confirming again that human contact networks are slow to achieve connectivity because of the prevalence of a few edges which occur much more frequently.

This view of the Pocket Switched Network as a random, static graph over estimates the actual connectivity achieved since it does not take into account the time order in which the edges were created. Regardless, the difference in connectivity observed implies that human contact networks are not optimised for achieving global connectivity (there exists at least one other contact occurrence distribution—that of choosing each edge uniformly at random—which can achieve connectivity with fewer number of contacts).

It is an interesting open problem as to whether, for a given contact occurrence distribution, a closed form solution can be found for the number of contacts required for a giant component to emerge. One potential approach would be to calculate the distribution of vertex degrees induced by the contact occurrence distribution. This could then be employed in the formula given by Molloy and Reed [MR95, MR98, NSW01] to obtain the condition for a giant component to emerge.

4.4 Delivery over fixed duration windows

The previous section showed that at a macroscopic level, a PSN is a challenged network, with connectivity crucially dependent on rare contacts, and frequent contacts non-effective for data transfer. This section examines time windows of fixed duration. It is observed that there can be a large variation in the delivery ratio achieved between windows of similar duration. Time windows which achieve a high delivery ratio are characterised by unequal connectivity, with a large clique of nodes having 100% connectivity amongst themselves and much worse connectivity among the other nodes.

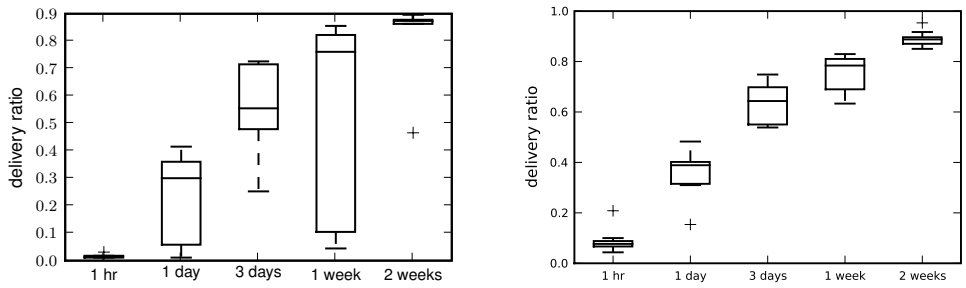


Figure 4.10: *Distribution of delivery ratios over different time windows of fixed duration. Allowing more time generally results in more data being delivered, but there is significant variation. Each box extends from the lower to upper quartile values, with a line at the median. Whiskers extend from the box to show the range. Outliers past the whiskers are plotted individually. Left: MIT trace. Right: UCSD trace. Each window size was tested using 25 independent window samples.*

4.4.1 Connectivity stable given number of contacts, but varies over similar duration windows

First we report a gross study of connectivity achieved over windows of fixed duration, and compare it to connectivity achievable over a fixed number of contacts.

Figure 4.10 uses a box-and-whisker plot to show the distribution of delivery ratios achieved by flooding data between every possible source-destination pair over time windows of different sizes. On average, allowing more time increases the delivery ratio. This is expected because the number of contacts can only increase over time. However, there is still significant variation, especially within windows of shorter duration, and the distributions are clearly skewed (observe the positions of the median). In order to discern the root of this variation, we examine the distributions of delivery ratio achieved over windows with a fixed number of contacts. First, we need to remove the effect of non-effective contacts (See §4.3.1). We do this by running the simulator over the time-shuffled version of the original trace, as defined in §4.3.1. The results, shown in Figure 4.11, reveals that windows with a fixed number of contacts have highly predictable levels of connectiv-

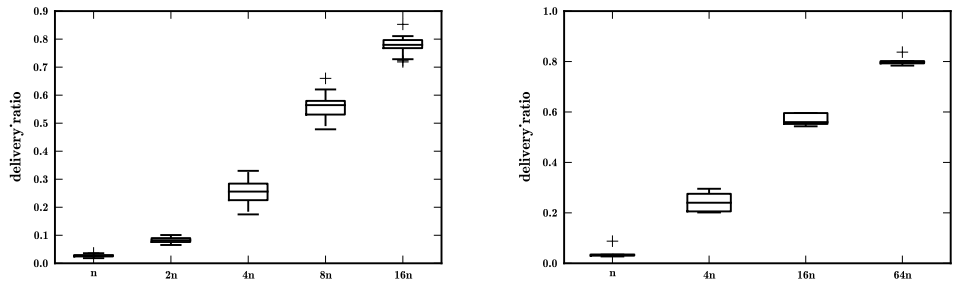


Figure 4.11: *Distribution of delivery ratios over different time windows with fixed number of contacts. Each window is independently chosen from time shuffled traces derived from the MIT (Left) and UCSD (Right) traces. In each distribution, the number of contacts chosen is kn where n is the number of nodes in the trace and k is a small constant. Observe that this procedure leads to much less variation than Figure 4.10.*

ity, as seen by the very narrow distribution of delivery ratios. This suggests that the wide distributions seen in Figure 4.10 are due to the differences in the numbers of contacts happening in different windows of similar duration.

The causes of such differences in human contact rates could be several, including natural ebb and flow of human activity due to time of day, day of week and longer-term effects such as national holidays, when patterns of contacts, as recorded in workplace and university-like settings of our data sets change. Next, we show how to identify time periods in which good connectivity is achieved.

4.4.2 Large cliques correlate with good connectivity

Over fixed time windows, the temporal contact graph of the PSN can be viewed as constructing a *static* reachability graph where a directed edge is drawn from node s to t if the sender s can transfer data to destination t during that window. The reachability graph is constructed by flooding data during the window between every possible source-destination pair. We examine this graph for clues about successful time windows which achieve high delivery ratios.

A preliminary examination (Figure 4.12) shows that the average number

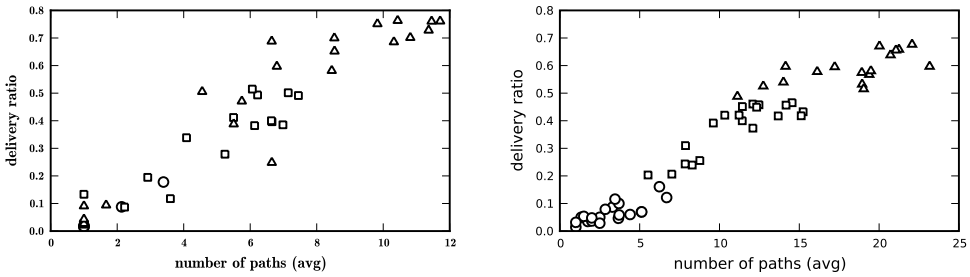


Figure 4.12: *Scatter plot showing a correlation between delivery ratio during random time windows of different sizes and the mean number of paths connecting node pairs. Left: MIT trace. Right: UCSD trace. Squares, circles and triangles represent windows of one-hour, one-day and 3 days, respectively.*

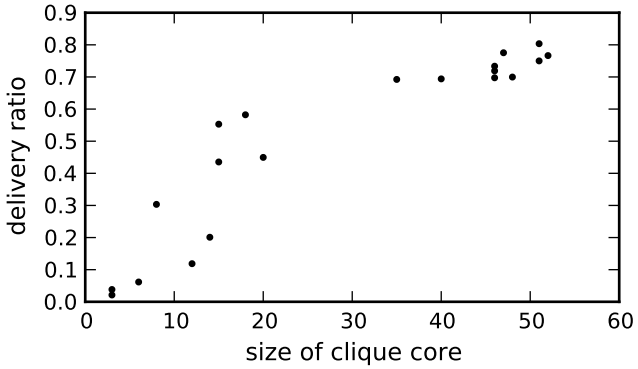


Figure 4.13: *Delivery ratio in the contact graph correlates with size of maximum clique observed in the reachability graph (MIT trace, non-overlapping 3-day windows).*

of paths connecting a source and destination in the contact graph exhibits a significant correlation with the achieved delivery ratio.

To uncover the reason, we focus on the MIT trace and divide the entire duration of the trace into non-overlapping 3-day windows and examine the reachability graph of each window for subsets of nodes with large numbers of paths. We find that windows with high delivery ratio tend to have a large subset of nodes that form a clique in the reachability graph (Figure 4.13). By definition, each member of a clique in the reachability graph can reach

every other member of the clique, leading to large numbers of paths when data is flooded.

While we expect delivery ratio to be high when the reachability graph has a large clique (implying that there is complete connectivity between a large fraction of nodes), it is rather surprising that the converse is true, viz. whenever the delivery ratio is high, there is a large clique in the reachability graph. To understand the implication, consider as example an arbitrarily chosen 3-day window in the MIT trace, with 77 active nodes, a clique of size 46 and an overall delivery ratio of 0.68. If nodes were equally connected, most nodes should be able to reach $\approx 68\%$ of the other nodes during this time window. The clique implies that a subset of 46 nodes ($\approx 60\%$ of the nodes) actually have 100% reachability amongst themselves. The 31 nodes outside the clique form $31 * 30$ source-destination pairs, of which only 59% have paths between them. The table below details the skewed reachability between these classes:

From\To:	**clique**	**outside**
clique	100.00%	78.61%
outside	76.44%	59.35%

In the same window as above, Figure 4.14 looks at the quality of the paths between nodes in the clique, outside the clique, as well as the paths that go from source nodes in the clique to destinations outside it, and vice-versa. Plotting the cumulative distribution functions of the delivery times of the quickest paths for each category shows that data is transferred faster when both the sender and receiver are members of the large clique.

Thus, during time intervals when there is a large clique in the reachability graph, the PSN is very successful, but only for the subset of nodes in the maximum clique observed. It is hard to predict the nodes involved because clique membership changes significantly (an average of 33 nodes are added or deleted over successive windows). Clique sizes also vary widely, with a mean size of 28.8, and a standard deviation of 18.2.

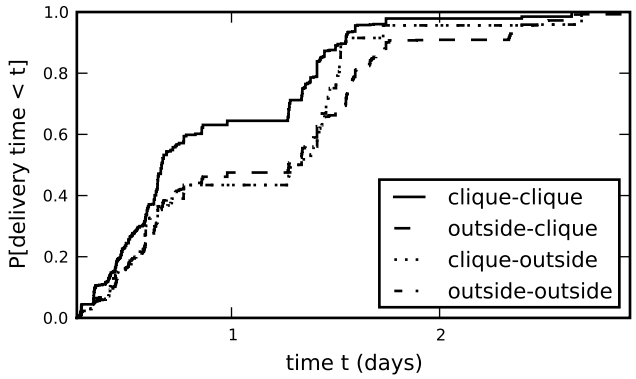


Figure 4.14: *CDFs of delivery times during a 3-day window with a 46 node clique. The four categories shown are different combinations of sender-receiver pairs when the source (or destination) is inside (or outside) the clique. clique-clique transfers are faster than other combinations (MIT).*

4.4.3 Clustering coefficient predicts delivery ratio

The clique occurs in the *reachability* graph, and cannot be easily detected without flooding all paths and performing extensive computation. We now show that the "cliquishness" of the *contact graph* can serve as an approximation.

Suppose a vertex v has neighbours $\mathcal{N}(v)$, with $|\mathcal{N}(v)| = k_v$. At most $k_v(k_v - 1)/2$ edges can exist between them (this occurs when v is part of a k_v-clique). The clustering coefficient [WS98] of the vertex, C_v, is defined as the fraction of these edges that actually exist. The clustering coefficient of the graph is defined as the average clustering coefficient of all the vertices in the graph. In friendship networks, C_v measures the extent to which friends of v are friends of each other, and hence, approximates the *cliquishness* of the graph. Figure 4.15 shows that the average clustering coefficient of the contact graph correlates well with the delivery ratio achieved during time windows of various sizes.

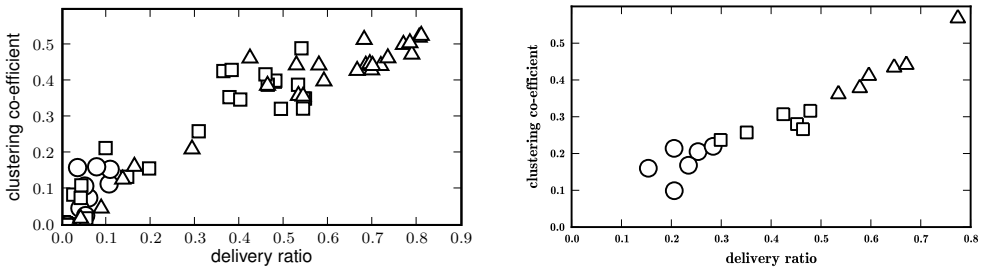


Figure 4.15: *Scatter plot showing the correlation between delivery ratio during random time windows of different sizes and the clustering coefficient of the* contact *graph for that time window. Left: MIT trace. Squares, circles and triangles represent windows of one-hour, one-day and three day windows, respectively. Right: UCSD trace. Squares, circles and triangles represent six-hour, one-day and three day windows, respectively.*

4.5 Understanding path delays

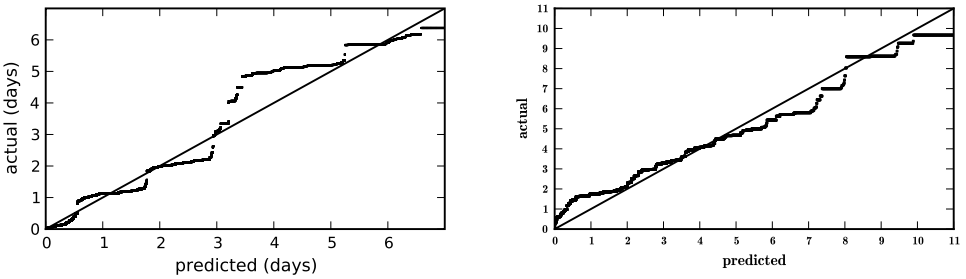
We move from considering the fraction of data delivered to the time taken to deliver data. This section focuses on understanding the delays on unicast paths that form during a fixed time window.

First, we examine the quickest paths. Then, we obtain an expression for path delay on any path discovered by flooding, in terms of the delay per hop and number of hops. Finally, we look at the distribution of the number of paths between all possible source-destination pairs. The ultimate goal is to obtain an approximate expression for delivery time, as the minimum of the path delays of a random number of paths.

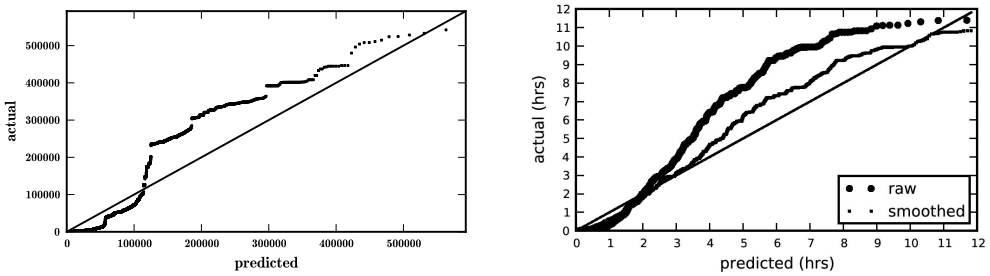
Table 4.1 summarises the notation and variables used in this section and the next.

4.5.1 Delivery time: path delay on the quickest path

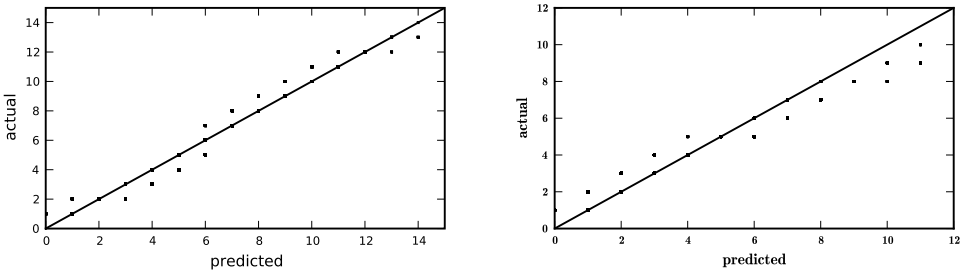
Delivery time is the time taken by the first path to reach the destination. Hence, it is the minimum of the path delays along all paths connecting the source and destination over time. Although there is a huge difference in the path delays of the first (quickest) and the last (slowest) paths that form



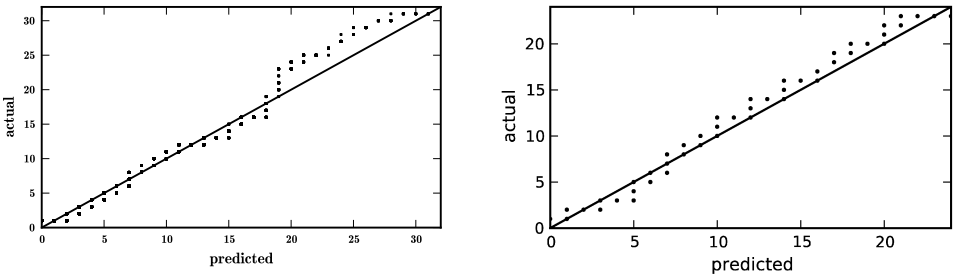
(a) Delivery time: Almost exponential. L: MIT, 1 week wnd. R: UCSD, 12 hour wnd.



(b) Hop delay: Roughly exponential. L: MIT, 1 week window, R: UCSD, 12 hr window.



(c) Number of hops: Poisson. L: MIT, one week window. R: UCSD, 6 hr window.



(d) No. of paths (also degree): Neg. Binomial L: MIT, 3 day wnd, R: UCSD, 6 hr wnd.

Figure 4.16: *Distributions related to successful paths. Each Q-Q plot shows fit through correspondence between sample deviates generated according to the theoretical distribution (predicted) and empirical (actual) values. Closeness to predicted=actual diagonal indicates better fit. Different combinations of trace and time window sizes are used to show generality of fit.*

H	Hop delay, or time to next hop. Time until a path expands by one more node. $H \sim Exp(rate = \nu)$
N	Number of edges per path. $N \sim Poisson(mean = \lambda)$
L	Number of paths between a random src-dest pair (also node degree). $L \sim NegBin(mean = \eta, overdispersion = \theta)$
D	Path delay for a random path
D^*	Delivery time (minimum path delay across all paths between a randomly chosen source & destination)
$G_X(s)$	Probability-generating function of Random Variable X
$M_X(s)$	Moment-generating function of X. $M_X(s) = G_X(e^X)$

Table 4.1: *Summary of notation used in §4.5 and §4.6*

during a time window, the Quantile-Quantile plot in Figure 4.16(a) shows that delivery time distribution for the quickest paths is almost exponential. Thus, most source-destination pairs have quick paths.

[SH03, ZNKT07] derive analytical expressions which are similar in spirit. However, these assume a constant (averaged) contact rate, whereas, as shown earlier in this chapter, the contact rates in our empirical traces are highly heterogeneous. Plugging in the average contact rate from the empirical traces into those expressions yields bad fits.

4.5.2 Characterising path delays

Next, we examine all successful paths, and express path delay in terms of the delay per hop and number of hops.

Hop delay (H)

The hop delay distribution captures the time that elapses between successive hops on the same path. This corresponds to the duration that a node has to carry data before it meets a new node that does not already have a copy of the data. Figure 4.16(b) shows a one week MIT window can be fitted to an exponential distribution with rate $\nu \approx 1.79 \times 10^{-5}$, giving a mean time of 15.5 hours to next hop. A similar fit can be obtained for the

UCSD window ($\nu \approx 0.0001$, corresponding a mean time of 2.6 hours to next hop).

For both the MIT and UCSD traces, the fit for the entire curve is approximate, with goodness of fit varying between different time windows. The fit can be improved by removing outlier values which are likely an artifact of the data set and the way in which our simulation plays back the traces. For instance, contacts in the UCSD trace are recorded as pairs of nodes which are simultaneously at the same Wi-Fi AP. Thus, when a number of nodes are simultaneously at the same AP, their pairwise contacts are recorded according to some arbitrary serialization, but with the same time stamp. When data reaches one of the nodes, the simulation creates the flood tree by playing back the contacts according to the serialization order. This leads to a number of hops within a small time window, when a single hop would have sufficed. Removing small hops results in a better fit as shown in Figure 4.16(b) (right side). The original fit is marked by the points marked "raw"; after removing the outliers, we obtain the points marked "smooth", which are closer to the actual=predicted diagonal, and hence represent a better fit. In the MIT trace, the outliers are a few rare hops (for example, $\approx 0.1\%$ of the hops in a window) which occur after a greater than four day wait in a one week window.

While the fit is not exact in many windows, we will assume that H is exponential, to simplify the analysis in the following section. This does not limit the applicability of our analysis. We will later show that the key results of §4.6 will apply equally for any other distribution which has a moment-generating function.

Note that there is no conflict between the nearly exponential distribution of hop delays and the previously reported power laws (with exponential tails) for inter-contact time distribution [KLBV07, CHC+07]. Inter-contact time is the time between repeated meetings of the same pair of nodes, whereas hop delay measures the time taken by the flooding process: from the time a node receives some data to the time it meets a new node that does not already have a copy of the data. The longer the duration a node carries the data, the greater the chance that data has already been flooded

to the nodes it meets. Because of flooding, the hop delay distribution decays rapidly, unlike the inter-contact time distribution.

Number of hops (N)

Several factors work together to limit the number of hops in a successful path. First, we only consider paths that form during a fixed time window. Second, the small-world nature of the human contact graph makes for short paths to a destination; and paths are frozen at the destination because the destination does not forward data further. Third, each node can join the flood-tree at most once. As the tree grows, the number of nodes available to grow the tree and extend a path decrease. Thus extremely long paths are rare. Figure 4.16(c) shows that the number of hops in paths that reach the destination during a one week time window of the MIT trace closely follows a Poisson distribution (The mean number of hops is $\lambda = 5.58$ in the window shown, and has been found to vary between 5–6 in different windows tested.). A similar fit can be obtained for the UCSD trace.

Path delay (D)

From the above empirically found distributions, we can derive the distribution of the path delay D on a random path as the sum of N hop delays. Thus, D can be written in terms of its moment-generating function (see Table 4.1 for notation and a summary of the component distributions):

$$M_D(s) = G_N(M_H(s)) = e^{\lambda(\frac{\nu}{\nu-s}-1)} \tag{4.1}$$

The average path delay is simply $M'_D(0) = \lambda\nu^{-1}$. Applying a Chernoff-type bound,

$$P[D \geq t] \leq \min_{s>0} e^{-st} M_D(s) = \min_{s>0} e^{\lambda(\frac{\nu}{\nu-s}-1)-st} \tag{4.2}$$

Minimizing by setting $s = \nu - \sqrt{\lambda\nu/t}$, we get (for $t < \sqrt{\lambda/\nu}$)

$$P[D \geq t] \leq e^{-(\sqrt{\nu t}-\sqrt{\lambda})^2} \tag{4.3}$$

Remark 4.1. The form of (4.3) indicates that the path delay on a random path is also close-to-exponentially distributed.

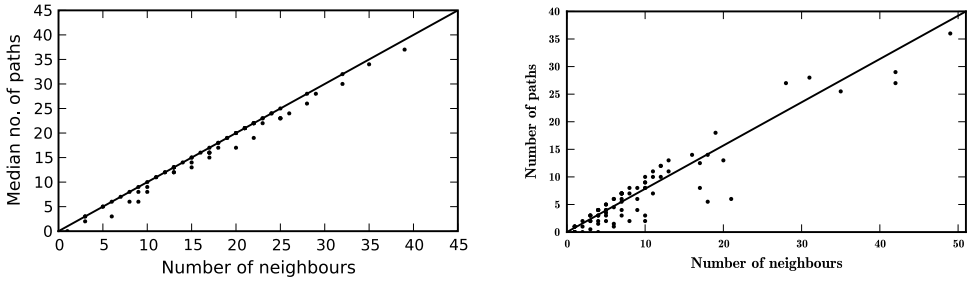


Figure 4.17: *Median number of successful paths reaching a destination node correlates with the number of distinct neighbours it has. Diagonal shows x axis=y axis. Left: MIT trace, one week window. Right: UCSD trace, 12 hour window.*

4.5.3 Characterising the number of paths (L)

Since each node joins the flood-tree at most once, there can be at most $N-1$ nodes (nodes other than destination) on the tree. Therefore there can be at most $N-1$ paths reaching a destination during the time window.

The actual number of paths depends on the number of unique nodes met by the destination: If the PSN is well mixed and the window is long enough, eventually all intermediate nodes become reachable from the source and get attached to the flood-tree. Thus the number of paths to a destination is determined by the number of distinct neighbours met by the destination over the time window.

Figure 4.17 empirically confirms this argument, by showing that the median (mean can also be used, instead) number of paths reaching a destination correlates with the number of distinct neighbours it has. As a consequence of this "eventual reachability" phenomenon, the distribution of the number of paths to a destination is simply given by the degree distribution of the who-met-whom graph of the PSN, taken over the entire time window.

Figure 4.16(d) shows that the degree distribution (and the number of paths) fits a negative binomial distribution. The best fits are obtained with the following parameters:

	mean	overdispersion
MIT	14.44	1.4
UCSD	8.15	1.25

Generative models for number of neighbours: a digression

The fact that the number of group members that an individual has contact with (the number of neighbours seen) follows the same distribution in both the traces, across time windows of different sizes suggests the possibility of an underlying stochastic mechanism. We discuss three different plausible models below. It should be stressed that our data sets do not have additional information that we could use to distinguish between these alternatives.

The negative binomial is a versatile distribution that can arise in a number of ways [BP70]. One possible way the negative binomial arises is as a continuous mixture of Poisson distributions where the mixing distribution of the Poisson rate is a gamma distribution. The model assumes that people acquire *new* neighbours according to a Poisson process with rate λ_p. The distribution of number of neighbours, K, is given by:

$$P(K = k|\lambda_p) = \frac{e^{-\lambda_p}\lambda_p^k}{k!}$$

Heterogeneity in the population is modeled by drawing λ_p from some population distribution $P(\lambda_p)$. If $P(\cdot)$ follows the gamma distribution with scale parameter η_1 and shape parameter η_2

$$P(\lambda_p = \lambda) = \frac{e^{-(\lambda/\eta_1)}(\lambda/\eta_1)^{(\eta_2-1)}}{\eta_1\Gamma(\eta_2)}, \tag{4.4}$$

it is known that the observed distribution of the number of neighbours seen by individuals in the group would then fit the negative binomial distribu-

tion:

$$\begin{aligned} P(K=k) &= \int_{\lambda=0}^{\infty} P(K=k|\lambda)P(\lambda)d\lambda \\ &= \int_{\lambda=0}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!} \frac{e^{(-\lambda/\eta_1)}\lambda^{(\eta_2-1)}}{\eta_1\Gamma(\eta_2)} \\ &= \frac{1}{k!\Gamma(\eta_2)\eta_1^{\eta_2}} \int_{\lambda=0}^{\infty} e^{-\lambda(1+1/\eta_1)}\lambda^{\eta_2+k-1}d\lambda \end{aligned}$$

Notice that the integrand can be made to resemble the Gamma distribution (Equation 4.4) by substituting $\eta_1 \leftrightarrow \eta_1/(\eta_1+1)$ and $\eta_2 \leftrightarrow \eta_2 + k$ and multiplying and dividing by $\Gamma(\eta_2+k)(\eta_1/(\eta_1+1))^{(\eta_2+k)}$. With this, the integral becomes $\int_0^\infty P(\lambda)d\lambda = 1$, and we get

$$\begin{aligned} P(K=k) &= \frac{\Gamma(\eta_2+k)}{k!\Gamma(\eta_2)} \left(\frac{\eta_1}{\eta_1+1}\right)^{\eta_2+k} \left(\frac{1}{\eta_1}\right)^{\eta_2} \\ &= \frac{\Gamma(\eta_2+k)}{k!\Gamma(\eta_2)} \left(\frac{1}{\eta_1+1}\right)^{\eta_2} \left(\frac{\eta_1}{\eta_1+1}\right)^{k} \end{aligned}$$

This is nothing but a negative binomial distribution giving the number of successes k before having η_2 failures, if each trial is Bernoulli with a success probability $p_c = \eta_1/(\eta_1+1)$. In our canonical formulation (see Table 4.1), this corresponds to a negative binomial distribution with overdispersion parameter $\theta = \eta_2$ and mean $\eta = \eta_1\eta_2$. [HJ06] consider a similar model for the distribution of new sexual partners acquired by people in a population.

Another possible explanation is based on the hypothesis that each person contacts new neighbours in order to satisfy some goals/needs. Suppose each fixed duration window contains a Poisson number of such neighbour-seeking events. Further, suppose most goals/activities follow a long-tailed distribution in which most needs are met with a small number of new neighbours, but occassionally require a large number of new neighbours. If the number of new neighbours required per goal/activity follows the Fisher log series distribution [FCW43], then the number of new neighbours is essentially the sum of a poisson number of logarithmic random variables; this gives rise to the negative binomial distribution.

Indeed, if the number of new contacts required per goal is given by a sequence of independent, logarithmic random variables X_i,

$$X_i \sim \frac{-1}{\ln(1-p)}\frac{p^x}{x}$$

and there are a Poisson number Y of activities with rate Θ, we obtain the total number of new neighbours as

$$Z = \Sigma_{i=1}^{Y} X_i$$

In terms of generating functions,

$$\begin{aligned} G_Z(s) &= G_Y(G_{X_i}) \\ &= \exp\left(\Theta\left(\frac{ln(1-ps)}{ln(1-p)} - 1\right)\right) \\ &= \exp\left(\ln\left(\frac{1-ps}{1-p}\right)^{\Theta/\ln(1-p)}\right) \\ &= \left(\frac{1-p}{1-ps}\right)^{-\Theta/\ln(1-p)} \end{aligned}$$

This corresponds to a negative binomial distribution with overdispersion $\theta = -\Theta/\ln(1-p)$ and mean $\mu = \theta(1-p)/p = -\Theta(1-p)/(p\ln(1-p))$

Finally, the process of acquiring new neighbours can be modeled as a pure birth process. Suppose each person acquires a new neighbour independently at a rate α, and each current neighbour introduces a new neighbour at a rate β. Thus the rate at which a node with x neighbours acquires new neighbours is given by

$$\lambda_x(t) = \alpha + \beta x, \alpha > 0, \beta > 0$$

. The probability distribution of the number of neighbours x at time t is given by the difference differential equations

$$P_x^{'}(t) = -\lambda_x P_x(t) - \lambda_{x-1} P_{x-1}(t), \qquad x > 0, \tag{4.5}$$

$$P_0^{'}(t) = -\lambda_0 P_0(t) \tag{4.6}$$

$$\tag{4.7}$$

subject to the intial conditions $P_0(0) = 1$, $P_x(0) = 0$ for $x \geq 1$.

It can be seen that this is satisfied by the negative binomial

$$P_x(t) = \frac{\Gamma(x+k)}{x!\Gamma(k)} p^k (1-p)^x \tag{4.8}$$

where $p = e^{-\beta t}$ and $k = \alpha/\beta$.

To show this, we make use of the identity

$$\begin{aligned} P_x(t) &= \frac{\Gamma(x+k)}{x!\Gamma(k)} e^{-\alpha t} \left(1 - e^{-\beta t}\right)^x \\ &= \frac{x+k-1}{x} \left(1 - e^{-\beta t}\right) P_{x-1} \end{aligned} \tag{4.9}$$

With a little algebra we obtain

$$\begin{aligned} P_x'(t) &= \frac{\Gamma(x+k)}{x!\Gamma(k)} \left\{ -\alpha e^{-\alpha t} \left(1 - e^{-\beta t}\right)^x + e^{-\alpha t} \left(1 - e^{-\beta t}\right)^{x-1} x e^{-\beta t} \beta \right\} \\ &= -\alpha P_x(t) + \beta e^{-\beta t} (x+k-1) P_{x-1} \\ &= -\lambda_x P_x(t) + \beta x P_x(t) + \beta e^{-\beta t} (x+k-1) P_{x-1} \\ &= -\lambda_x P_x(t) + \beta (x+k-1) P_{x-1} \left(1 - e^{-\beta t} + e^{-\beta t}\right) \\ &= \lambda_x P_x(t) + \lambda_{x-1} P_{x-1} \end{aligned}$$

satisfying (4.6). It is easy to see that (4.8) also satisfies (4.7) and the initial conditions.

4.6 Evaluating the effect of failures

Many studies on Pocket Switched Networks, including ours, implicitly assume that every contact on any path which occurs can be used to successfully transfer data. In reality, many factors could prevent a path from being useful: An intermediate node may run out of storage; the time available during a contact opportunity may not be sufficient; or a node can simply fail.

Since only one path between every source-destination pair needs to succeed for data delivery, individual path failures do not greatly impact the

delivery ratio achieved at the end of a time window (unless all paths between a source-destination pair fail, disconnecting the network). However, it can affect the rate at which the delivery ratio evolves: Suppose the quickest path between a pair of nodes would have arrived at t_1, but cannot be used because of a failure. If the first usable path connects the nodes at time $t_2 > t_1$, then between t_2 and t_1 the fraction of data delivered is decreased on account of the path failure. In other words, there is a delay in data delivery, which temporarily shifts the cumulative distribution of delivery times to the right.

Given a sequence of contacts, flooding achieves the best possible delivery times by exploring *every* contact opportunity and thereby finding the path with the *minimum* path delay. This section looks at the degradation in the delivery time distribution when not all of the paths found by flooding can be explored.

Specifically, we study two failure modes: The first, proportional flooding, explores a fixed fraction μ of the paths found by flooding between each source and destination. We show that a constant increase in the fraction of paths explored brings the delivery time distribution of proportional flooding exponentially closer to that of flooding over all paths. The second failure mode, k-copy flooding, explores no more than a fixed number $k > 1$ of the paths found by flooding between each source and destination. Again, a constant increase in k brings the delivery time distribution exponentially close to the optimal delivery time distribution of flooding all paths. Empirically, even small values of k (e.g., $k = 2$ or $k = 5$) closely approximate delivery times found by flooding.

The results of this section imply that the human contact network is remarkably resilient to path failures and the delivery ratio evolves at a close-to-optimal rate even when the majority of paths fail and only a small fraction or a small, bounded number of paths can transport data to the destination.

Note that we only admit paths from the original flood-tree, and do not include new paths that repair failures by joining the affected nodes to the flood tree at later contacts. Thus, if there is a path failure at an edge a-

b in the flood-tree, b and all its descendants are excluded from the paths that reach the destination. In reality, nodes from b's sub-tree could rejoin the flood-tree by obtaining a copy of the data during a later contact and creating a new path to the destination. This implies that our results in fact undercount the number of good paths and underestimate the resilience of the human contact network.

The success of k-copy flooding can provide a loose motivation for routing algorithms that use multiple paths between each sender and destination pair since this could obtain a close-to-optimal delivery time distribution. However, heuristics-based routing algorithms may not find the same paths as those on the flood-tree. Thus, the correspondence is not exact.

4.6.1 Proportional flooding

Consider an arbitrary source-destination pair. We will model the path delays between them as being chosen independently and identically from the distribution in (4.1). Suppose copies of the data are sent along l randomly chosen paths between them. The obtained delivery time D_l^* is the *minimum* of the path delays across all l paths. Using (4.3) we can write

$$P\left[D_l^* \leq t\right] = 1 - \prod_{i=1}^{l} P\left[D \geq t\right] \geq 1 - e^{-l(\sqrt{\nu t}-\sqrt{\lambda})^2} \tag{4.10}$$

Note that the above assumes that the l path delays are independent. In reality, paths found by flooding all fan out from a single source node, and the first few hops, close to the source, are typically shared with other paths, violating the independence assumption. Therefore, the model in this section is to be considered only as a simple formulation designed to gain insight into proportional flooding. It is worth mentioning however that in the empirical data sets, we frequently find that the major component of path delay is contributed by the part of the paths closest to the destination, which are not shared with other paths. Also, in the case when only a few paths on flood-tree are being randomly sampled, the number of hops shared is limited.

Consider source-destination pairs with $L = m$ paths connecting them. Full flooding finds the quickest of all m paths and obtains a delivery time distribution $P\left[D_L^* \leq t | L = m\right]$. Proportional flooding chooses a fraction μ of them. From (4.10), the difference $\Delta(t;\mu)$, in the delivery time distributions between full and proportional flooding, is upper bounded by

$$\Delta(t;\mu) \leq P\left[D_L^* \leq t | L = m\right] - 1 + \ e^{-\mu m\left(\sqrt{\nu t}-\sqrt{\lambda}\right)^2} \tag{4.11}$$

Remark 4.2. A constant increase in μ has an exponential effect on Δ: For any t, if μ is increased by some constant, the fraction of data delivered by proportional flooding during $[0, t]$ becomes exponentially closer to that delivered by full flooding. Thus, proportional flooding quickly becomes very effective as μ is increased.

While the above is not unexpected given the model and the resulting distributions as obtained in §4.5.2, this observation can be generalised: For a hop delay distribution with moment-generating function $M_H(s)$, (4.1)–(4.11) can be rederived to get

$$\Delta(t;\mu) \leq P\left[D_L^* \leq t | L = m\right] - 1 + \exp(\mu m F_H(t)), \tag{4.12}$$

where $F_H(t) = \lambda M_H(s_{\min}(t)) - s_{\min}(t)t - \lambda$ and $s_{\min}(t)$ minimises s in (4.2)[3]. Thus the exponential decrease in Δ with a constant increase in μ is obtained as long as $F_H(t) < 0$. In other words, our results hold when there are a Poisson number of hops in paths formed over fixed time windows, for any hop delay distribution H that has a moment generating function and satisfies $F_H(t) < 0$.

Also, since

$$\frac{\partial \Delta}{\partial \mu} = m F_H(t) e^{\mu m F_H(t)} < 0,$$

Δ decreases when μ is increased. Furthermore, the rate of decrease is higher for smaller μ—increasing μ from $\mu = 0.1$ to $\mu = 0.2$ results in a greater decrease than an increase from $\mu = 0.6$ to $\mu = 0.7$.

[3]When H is exponentially distributed, we get $F_H(t) = -\left(\sqrt{\nu t} - \sqrt{\lambda}\right)^2$. Plugging this value into (4.12) yields (4.11).

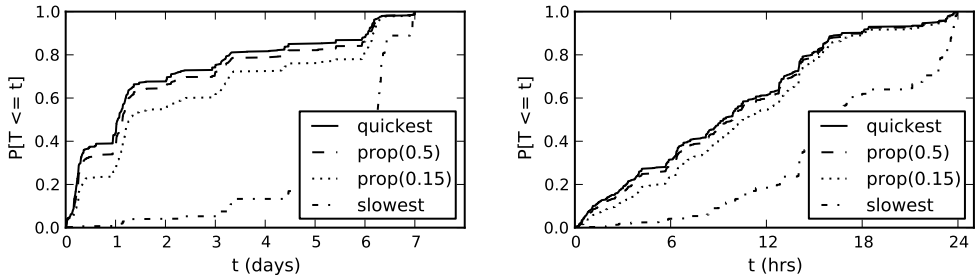


Figure 4.18: *Proportional Flooding: Source and destination are connected by multiple paths of different delays. (CDFs of the* quickest *and* slowest *paths are shown). Proportional flooding randomly selects a fraction μ of these paths. As μ increases, proportional flooding recovers much of the benefit of flooding over all paths by closely approximating the optimal delivery time distribution. $\mu = 0.5$ (prop(0.5)) and $\mu = 0.15$ (prop(0.15)) are shown. Left: MIT trace, one week window. Right: UCSD trace, one day window.*

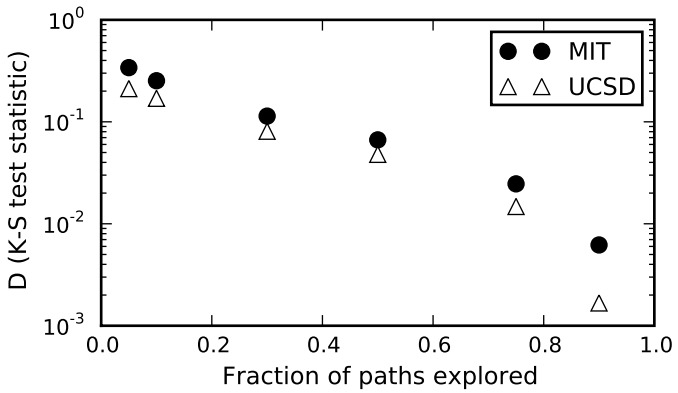


Figure 4.19: *K-S statistic (D) measuring the difference between the delivery time distributions of full flooding and proportional flooding for different μ. X-axis is linear, Y-axis is log-scale. MIT: 1 week window, UCSD: 6 hour window.*

Empirical validation

Figure 4.18 shows that empirically, even small values of $\mu = 0.15$ or $\mu = 0.5$ can closely approximate the optimal delivery time distribution of flooding (the *quickest* line). We measure the closeness of approximation for different values of μ next.

Figure 4.19 empirically shows the difference between $D^*(t)$, the delivery time distribution obtained by flooding over all paths, and $D^*_\mu(t)$, the delivery time distribution for proportional flooding using a randomly selected fraction μ of paths between every source and destination. The difference is measured using the Kolmogorov-Smirnov statistic given by $D = \max_t (D^*(t) - D^*_\mu(t))$. Note that the Y-axis is log scale; a constant increase in μ shows an exponential decrease in D.

Figure 4.19 can also be interpreted as the maximum instantaneous slowdown incurred over the entire time window, due to not exploring all paths. Thus the figure indicates that by exploring just 25% of the paths, we can reach within 10% of flooding, the optimal strategy (i.e., we can deliver at least 90% as much data as delivered by flooding. Note that this bound holds on the instantaneous delivery ratio *at any point in time*). At the same time, to reach within 1% of the optimal (i.e., to deliver 99% as much data as flooding), we would need to explore more than 80% of the paths.

4.6.2 From proportional to bounded number of paths

§4.5.3 showed that the number of paths to a destination (L), is a random variable that is well modeled by the negative binomial, a positively (or right) skewed distribution. Thus a majority of sender-destination pairs have a small (fewer than average) number of paths, but a minority have a large number of paths, which pulls the average higher. The expected number of paths that need to work for successful proportional flooding is given by $\mu\mathbb{E}[L]$, which is higher than it would be if the minority of node pairs with large numbers of paths were not considered. In the worst case, when there are $(N-1)$ paths between a sender and destination, proportional flooding requires $\mu(N-1)$ of these to be functional.

This suggests an alternate bounded cost strategy that explores at most a fixed number, k, of the paths between every sender and destination. We call this k-copy flooding. Unlike proportional flooding, k-copy flooding explicitly limits the number of paths explored, and therefore can tolerate

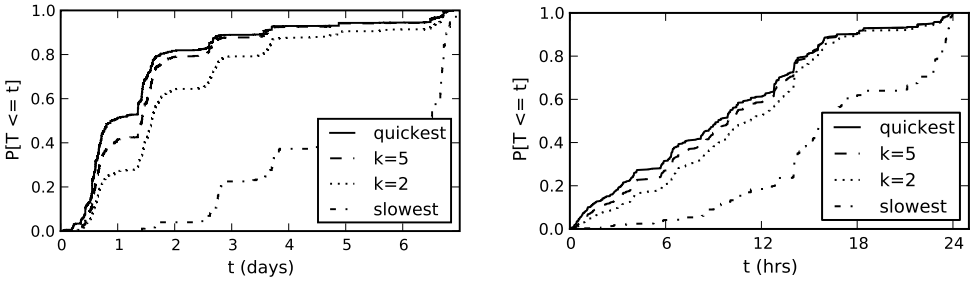


Figure 4.20: *k-copy flooding: Nodes are connected by multiple paths with different delays (CDFs of the* quickest *and* slowest *are shown). Yet, randomly choosing at most k of the paths to each destination closely approximates the quickest, even for small k. Left: MIT trace, one week window. Right: UCSD trace, one day window.*

a larger number of path failures in the worst case, when there are a large number of paths between a node-pair.

Figure 4.20 shows empirically that in our data sets, even for small k $(= 2, 5)$, the delivery time distribution of k-copy flooding starts to closely approximate full flooding. To see why, consider the *equivalent fraction* μ_k of paths in proprotional flooding that gives the same expected number of paths as k-copy forwarding:

$$\sum_{l=0}^{k} lP\left[L = l\right] + kP\left[L > k\right] = \mu_k \mathbb{E}\left[L\right] \tag{4.13}$$

Suppose k is increased by a constant h, resulting in a new equivalent fraction μ_{k+h}. (4.13) becomes

$$\sum_{l=0}^{k} lP\left[L = l\right] + \sum_{j=1}^{h}(k+j)P\left[L = k+j\right] + (k+h)P\left[L > k+h\right] = \mu_{k+h}\mathbb{E}\left[L\right]$$

Regrouping, we get

$$\sum_{l=0}^{k} lP\left[L = l\right] + k\left(\sum_{j=1}^{h} P\left[L = k+j\right] + P\left[L > k+h\right]\right) + \sum_{j=1}^{h} jP\left[L = k+j\right] + hP\left[L > k+h\right] = \mu_{k+h}\mathbb{E}\left[L\right]$$

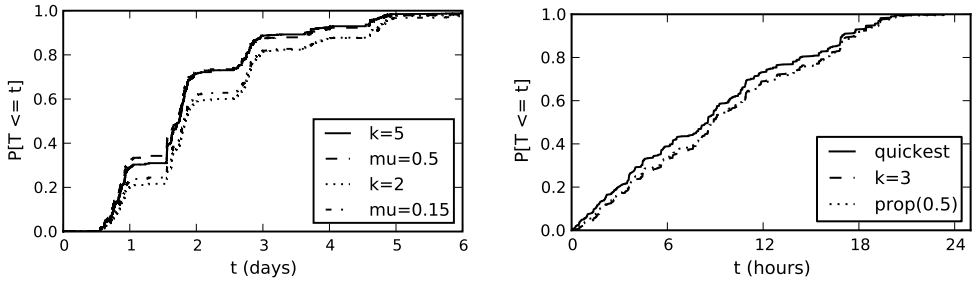


Figure 4.21: *Comparing Proportional flooding with k-copy flooding. Left: MIT, one week window.* $\mu_2 = 0.15$ *of paths has similar delivery times as* $k = 2$*-copy flooding. Similarly* $k = 5$ *corresponds to* $\mu_5 = 0.5$*. Right: UCSD, on day window.* $k = 3$ *corresponds to* $\mu_3 = 0.5$*.*

Comparing with (4.13), we can write

$$\mu_k \mathbb{E}[L] + \sum_{j=1}^{h} jP[L = k + j] + hP[L > k + h] = \mu_{k+h} \mathbb{E}[L]$$

Thus the *increase* in the equivalent fraction of paths is

$$\begin{aligned} \mu_{k+h} - \mu_k &\geq \frac{h}{\mathbb{E}[L]} \left(\sum_{j=1}^{h} P[L = k + j] + P[L > k + h] \right) \\ &= h \left(P[L > k] / \mathbb{E}[L] \right) \end{aligned} \tag{4.14}$$

Remark 4.3. A constant increase in k is equivalent to at least a (scaled) constant increase in the fraction of paths explored by proportional flooding. Thus, as a simple consequence of Remark 4.2, a constant increase in the number of paths explored in k-copy forwarding moves its delivery time distribution exponentially closer to that of full flooding.

Note however that the constant of scaling decreases as k increases. For the same h, the increase in the equivalent fraction is higher for smaller k.

This explains why exploring at most a small number k of paths has a delivery time distribution approaching that of flooding over all paths. Figure 4.21 empirically shows the equivalent fractions μ_k for different k values. We next derive a closed form for μ_k when the number of paths follows the negative binomial distribution $L \sim \text{NegBin}(\text{mean} = \eta, \text{dispersion} = \theta)$. The

probability mass function for L can be written as:

$$P\left[L=l;\eta,\theta\right]=\frac{\Gamma(\theta+l)}{(l)!\,\Gamma(\theta)}\left(\frac{\theta}{\eta+\theta}\right)^{\theta}\left(\frac{\eta}{\eta+\theta}\right)^{l}$$

First, using $p=\theta/(\eta+\theta)$, we have [KKK05]

$$P\left[L>k\right]=1-P\left[L\leq k\right]=1-I_p(\theta,k+1) \tag{4.15}$$

where $I_x(a,b)=B(x;a,b)/B(a,b)$ is the regularized incomplete Beta function. Next,

$$\begin{aligned}\sum_{l=0}^{k}lP\left[L=l\right]&=\sum_{l=1}^{k}\frac{\Gamma(\theta+l)}{(l-1)!\,\Gamma(\theta)}\,p^{\theta}(1-p)^{l}\\&=\theta\frac{(1-p)}{p}\sum_{l=1}^{k}\frac{\Gamma(\theta+1+l-1)}{(l-1)!\,\Gamma(\theta)}\,p^{\theta+1}(1-p)^{l-1}\\&=\theta\frac{(1-p)}{p}\sum_{l'=0}^{k-1}P\left[L=l';\eta,\theta+1\right]\\&=\eta I_p(\theta+1,k)\end{aligned} \tag{4.16}$$

Substituting (4.15) and (4.16) into (4.13),we get

$$\mu_{_k}=I_p(\theta+1,k)+\frac{k}{\eta}I_{1-p}(k+1,\theta) \tag{4.17}$$

4.6.3 Resilience and load balancing

The results above suggest that the human contact network is remarkably resilient to path failures and the network's optimal rate of data delivery can still be approximated even when many of the paths do not succeed in delivering data. To conclude, we briefly elucidate this from the perspective of intermediate node failures.

The source and destination rely on the rest of the network to serve as intermediate nodes and form connecting paths. The use of multiple paths provides a degree of resilience against path failures, since only one of the paths needs to succeed. Deterministic strategies that selectively favour particular next hop nodes or even particular paths can potentially overburden

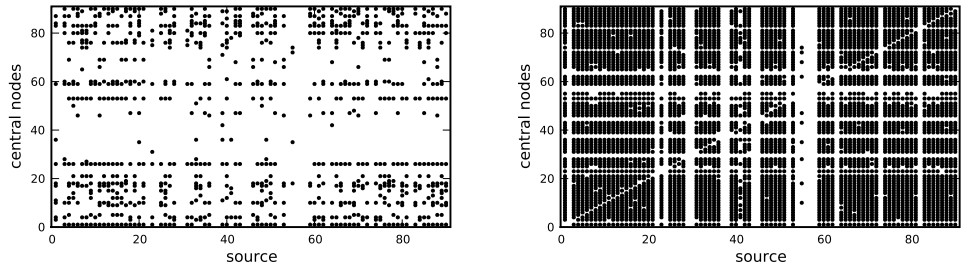


Figure 4.22: *Scatterplots showing node numbers of source on the X-Axis, and on the Y-Axis, nodes numbers which cross a threshold (=5) betweenness centrality for the corresponding sources. Left figure shows the central nodes when paths are picked according to a deterministic strategy (the quickest path). Right figure shows the central nodes when up to five paths are randomly selected. The random strategy selects more central nodes and spreads the load more evenly. A similar set of figures can be obtained by looking at the most central nodes for a given* destination*. (MIT trace, one week window. Empty columns in the middle correspond to nodes which were inactive during the window.)*

certain intermediate nodes that end up getting selected more often, but randomised strategies such as the ones we discuss are more resilient and have fewer bottlenecks.

For any strategy, we can measure the burden placed on a given intermediate node by any given source (respectively, destination) by counting the number of paths of the sender (destination), chosen according to the strategy, in which the node figures as an intermediate hop. We call this number the betweenness centrality of the node for a given source (destination).

Source nodes (destinations) with very few central nodes are vulnerable to being disconnected if the central nodes fail. As Figure 4.22 shows, deterministic strategies such as picking the quickest possible path can result in a network that is sparse in central nodes. A randomised strategy results in more central nodes and thus renders the network more resilient against failures. A sender (destination) that has few central nodes is also likely to face congestion if the central nodes hit capacity bottlenecks. The availability of more central nodes opens the possibility of load balancing in such cases.

4.7 Multicast within communities

The previous section showed that paths occuring over time in human contact networks have a diversity which renders them resilient to failures. In this section, we characterise these paths further by showing that paths within detectable communities are faster than those which go across communities. We demonstrate this by showing that multicast by restricting forwarding to nodes within a community is nearly as fast as flooding the entire network.

Consider data multicast from a single source to multiple destinations. Although k-copy flooding can be extended to this case by sending k separate copies to each destination, we can do better by observing that some of the destinations can be intermediate nodes in the paths to another destination. Below, we explore the delivery times obtained if the intermediate nodes consisted entirely of other destination nodes.

In other words, we are interested in the distribution of delivery times across all intended receivers when we flood the multicast data, but restrict the nodes involved to the source and the set of intended destination nodes. We call this *communitycast* because the intermediate nodes include only the community of interested nodes. The obtained delivery times are compared to the delivery times from flooding the entire graph. We call this *fullcast*; it is similar to standard epidemic flooding.

In general, we expect fullcast to perform better, because exploring a larger number of paths can find faster paths. However, when the community of interested nodes coincides with a detectable community in the who-met-whom graph, taken over a time window, we find that communitycast performs nearly as well. This implies that most of the fast paths are in fact contained within the community.

We detect communities using the modularity measure defined by Newman and Girvan [NG04]. Modularity is a fitness function that measures the expected number of edges within nodes of a community, as compared to the number of edges that would occur in a null model, typically a random graph which preserves the degree distribution. In an m edge random graph,

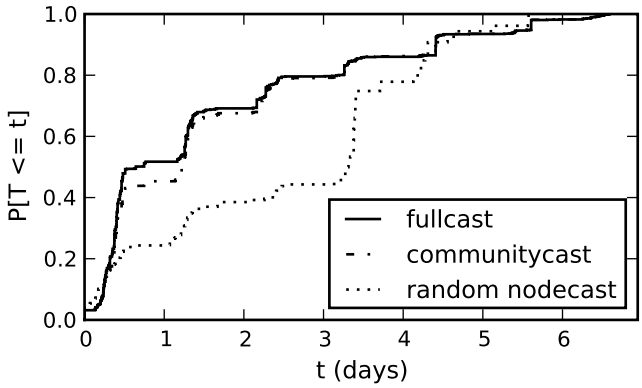


Figure 4.23: *communitycast = fullcast: When the receivers are all within a community, restricting flooding to the set of intended receivers (communitycast) has delivery time distribution similar to flooding the entire graph (fullcast). This does not hold when recievers are in different communities (random nodecast). MIT, one week window*

between two nodes i, and j with degrees k_i, k_j respectively, there is an edge with probability $k_i k_j/(2m)$. If A_{ij} is the element in the adjacency matrix giving the number of edges between i and j, modularity is written as:

$$Q = \frac{1}{2m} \sum_{i,j \in \text{same community}} \left[A_{ij} - \frac{k_i k_j}{2m} \right],$$

An iterative algorithm [BGLL08] is used to partition nodes into communities, so that modularity is maximised. Note that we run this algorithm on the static graph of known contacts between nodes.

The who-met-whom graph gives equal weight to both rare and frequent contacts and therefore does not directly predict delivery times for aribtrary source-destination pairs. Nevertheless, Figure 4.23 shows that for nodes within a community, fast paths are found within the community itself.

Our experimental setup is as follows: We randomly select a subset of nodes within a community. Data is multicast from each node selected. Each datum goes out to all the other nodes in the chosen subset. We consider the two strategies. The curve "fullcast" shows delivery times from multicast by flooding all data through every possible contact happening in the entire

pocket switched network. This will achieve the best possible delivery times, but at considerable load to the entire network. The curve "communitycast" finds delivery times by flooding each multicast datum amongst the set of intended receivers alone. Note that "communitycast" is close to "fullcast". However, when a random subset of nodes with the same size as the multicast community is chosen for communitycast (curve "random nodecast" in the figure), we find that restricting the paths to the destination nodes alone results in slower delivery times.

4.8 Related work

Conceptually, PSNs are Delay-Tolerant Networks [Fal03], and generic results from that framework apply. For instance, a forwarding algorithm that has more knowledge about contacts is likely to be more successful [JFP04], and the best performance is achieved by an oracle with knowledge of future contacts.

Nevertheless, the fact that our underlying network is made up of human contacts and is less predictable has a large impact: For instance, reasonably predictable traffic patterns of buses allow a distributed computation of route metrics for packets in vehicular DTNs [JFP04, BLV07]. Similarly, fixed bus routes allow the use of throwboxes [ZCA+06] to reliably transfer data between nodes that visit the same location, but at different times.

The variability of PSNs has naturally led to a statistical approach: The inter-contact time distribution of human social contacts has been used to model transmission delay between a randomly chosen source-destination pair [CHC+07, KLBV07]. In this work, we take a more macroscopic view and look at the ability of the PSN to simultaneously deliver data between multiple source-destination pairs. This leads us to look at the distribution of the *number* of contacts between randomly chosen source-destination pairs, and find that this distribution is not only crucial for global data delivery performance, but also for the connectivity of the PSN itself.

This paper uses variants of flooding to obtain a better understanding of *achievable* data delivery properties of human contact networks. However,

unbounded flooding is expensive. To mitigate this, various routing protocols have been proposed. These typically use various ad-hoc metrics, such as betweenness centrality [DH07], history of previous meetings [LDS04], and inferred community structure [HCY08]. Computing such metrics can be costly and the computation can be inaccurate due to the high variability inherent in PSNs. Our results point to simpler techniques that could exploit time windows of good connectivity or the use of multiple paths.

[SH03, GNK05, ZNKT07] model the performance of epidemic routing and its variants. In particular, they derive a closed form for delivery time distribution, and show it to be accurate for certain common mobility models. However, several simplifying assumptions are made, including an exponential inter-contact time between node pairs. Unforunately, human contact networks are known to have power law inter-contact times with exponential tails [CHC+07, KLBV07]. Furthermore, [SH03, GNK05, ZNKT07] use a constant contact rate, whereas our studies show that human contacts are highly heterogeneous. [MN05] considers heterogeneous contact rates between mobile devices but only in the context of establishing an epidemic threshold for virus spread.

The number of paths found by flooding is crucial to the success of proportional and k-copy flooding. Counting differently, [ECCD07] reports a phenomenon of "path explosion" wherein thousands of paths reach a destination shortly after the first, many of which are duplicates, shifted in time. In contrast, duplicate paths are prevented in our method of counting, by having nodes remember if they have already received some data, resulting in a maximum of $N - 2$ paths between a source and destination.

The power of using multiple paths has been recognised. Binary Spraying, which forms the basis for two schemes (spray and wait, spray and focus) has been shown to be optimal in the simple case when node movement is independent and identically distributed [SPR08]. [ECCD08] noted that among routing schemes evaluated, those using more than one copy performed better. Furthermore, all algorithms employing multiple paths showed similar average delivery times. The success of k-copy flooding suggests a possible explanation for this result. Similarly, [HXL+09] finds that

the delivery ratio achieved by a given time is largely independent of the propensity of nodes to carry other people's data. They suggest the existence of multiple paths as an explanation. At an abstract level, the refusal of a node to carry another node's data can be treated as a path failure. Thus §4.6 corroborates [HXL+09] and provides a direct explanation.

We mention in passing that our finding of large cliques in the reachability graphs (§4.4.2) is loosely analogous to the giant strongly connected component in the WWW graph that accounts for most of its short paths [BKM+00]. Similarly, our finding that the human contact graph is resilient to path failure (§4.6) is echoed in the attack tolerance demonstrated for static graphs of many complex networks[AJB00]. Most importantly, the result that rare contacts are much more important than frequent ones for connecting arbitrary source-destination pairs (§4.3) offers direct support for Granovetter's theory of the strength of weak ties in diffusion processes and spreading innovation [Gra73]. [OSH+07] obtained similar results in mobile phone call graphs.

Multipath has been used for other purposes in different types of networks. For example, [DBN03] uses multiple copy forwarding for reliability in sensor networks. [KMT07] (and several works cited in there) consider multipath routing from the perspective of increasing throughput, etc. In the case of PSNs, minimizing latency or path delay is more important and is our focus here. For instance, in multipath routing as considered in the Internet setting, all paths are used concurrently to send more data than could be sent over a single path. Alternately, multiple paths are used for providing failover availability. Here, we *concurrently* flood *copies* of the same data along different paths, to decrease expected path delay.

4.9 Design recommendations

From a systems perspective, the ultimate goal of a study such as the above is, of course, to design better routing algorithms for Pocket Switched Networks. While designing, implementing and evaluating a new routing algorithm is beyond the scope of our current study, we conclude with some

recommendations, based on insights and experience gained here.

§4.3 showed that rare contacts are crucial for connectivity, and that frequent contacts are usually not effective for data delivery. This suggests developing new routing algorithms that weight rare edges over frequent ones. In particular, Figure 4.8 shows the promising result that when connecting arbitrary source-destination pairs, routing only on the rare edges does not greatly affect the delivery time distribution. However, nodes which may be well-connected with each other could benefit from using the more frequently occurring edges.

§4.4 demonstrated that delivery ratio achieved can vary significantly over time windows of similar duration, and showed that good time windows can be predicted by the average clustering coefficient of the contact graph. Clearly this information can be used for routing purposes. In practice, a node can compute its current clustering coefficient by obtaining a list of recent contacts from each node it meets. The average clustering coefficient of the network can be approximated by propagating current local estimates by adapting a distributed algorithm to compute aggregates, such as [ZGE03].

By using clustering coefficient as a predictor for delivery ratio, senders (or other nodes on their behalf) can make informed decisions about how to deliver data. For instance, nodes with high clustering co-efficients could be preferred by routing algorithms. Similarly, a node which observes a low clustering coefficient can aggressively send multiple copies, resend the same data over time, or even bypass the PSN entirely and use a more expensive infrastructure-based communication mechanism such as a satellite connection.

§4.5 examined distributions of random variables that contribute to the delay on paths that successfully connect source nodes to their destinations. Among other results, it was found that the number of hops on a path follows a Poisson distribution. Primarily as a consequence of this, it was shown (§4.6) that the human contact network exhibits a remarkable resilience to random path failures. Increasing the number or fraction of paths explored between source and destination by a constant brings the delivery time distribution exponentially close to the optimal. Thus, the network can

continue to deliver data at a near optimal rate even when a large number of random path failures occur.

Two immediate consequences of this result were discussed: First, Routing algorithms which concurrently explore more than one path are likely to be successful. Randomised sampling amongst paths considered could lead to better resilience and load balancing properties. Second, because of the robustness to failure, the network as a whole will continue to perform well even if some nodes decide to be selfish and cause path failures.

Finally §4.7 showed that when the intended receivers of a multicast message are within a community, they can efficiently be reached without the message leaving the community. This implies that a message can be securely delivered to all intended receivers within a community without sacrificing the speed of delivery.

In summary, the results of this chapter could be used to adopt a principled approach to the development of routing algorithms for Pocket Switched Networks, rather than relying on heuristics as motivation.

CHAPTER 5

Reflections and future work

The vast amount of information being divulged by users online, especially in the context of social networks, is widely considered to be a valuable commodity. But surprisingly, little commercial use is being made of it. Only at the application layer is it being exploited, albeit in an elementary way. For example, external websites such as online newspapers are starting to incorporate social plugins that enable sharing a link to a story with friends on the social network website.

The previous chapters showed that *cross-layer* designs which employ social information and structure for lower level operations such as supporting data delivery can indeed be practical. If applied in a principled fashion, such designs could greatly expand the impact of social networks. In this chapter, we conclude by discussing how future work can generalise from the examples presented in this dissertation.

We first review the two case studies presented in previous chapters (§5.1). Then we propose a Social Information Plane (§5.2), as a way of unifying such designs. We finish with a few comments on the systems-level consequences of building systems on social properties and the difference in focus, when compared with previous network science-based approaches to social networks (§5.3).

5.1 Summary of contributions

This dissertation presented two very different case studies, both of which showed how properties from a relevant social network can help data delivery.

In Chapter 3, we explored the problem of serving the tail of rich-media user-generated content. Effectively serving rich-media content such as streaming videos requires using an expensive Content Delivery Network. Such costs are only offset for the popular and widely accessed items, and by definition, individual items in the tail do not meet this requirement. However, we found that the tail is long and contains the vast majority of the videos. Collectively, the tail items command a great deal of interest—nearly half the users have 60% or more of their preferred videos in the tail—making it important to serve effectively. We then examined the properties of the social network of users who access individual items, and found that unpopular tail items can be distinguished from the popular head items by their relative proportions of viral and non-viral accesses. Using this, we first devised SpinThrift, a strategy to save energy by storing the infrequently accessed tail items on disks at low-power mode. Additionally, our studies found that popularity ranks could vary dramatically; SpinThrift was explicitly designed to be robust in the face of such changes. Then we observed that because tail items are predominantly accessed virally, the locations of those accesses are non-random: the viral accesses are confined to the locations of friends of previous users. This insight led to the design of Buzztraq, a selective replication strategy that decides replica locations by their potential for contributing viral accesses.

In Chapter 4, we studied properties of human social contacts that could help in opportunistically forming paths over time between a source and its intended destination. Several heuristics-based routing protocols have previously been designed to discover such paths; this study is intended to inform the design of such protocols. We found that the fraction of data delivered is determined by the contact occurrence distribution. There is a great disparity in contact occurrences: some pairs of nodes meet very frequently, but many other pairs contact each other rarely. Unfortunately,

the rare contacts are crucial for connectivity, and frequent contacts can be avoided without greatly affecting speed of delivery. We discussed two implications of this result: First, this suggests that routing decisions should weight rare contacts over frequent contacts. Second, it offers a confirmation of the "strength of weak ties" theory [Gra73]. Connectivity achieved was seen to be highly uneven, with significant differences between different windows of similar duration, and also between different nodes within the same window. We showed how the clustering co-efficient on the contact graph can be used to predict the connectivity achieved. We then examined the multiple paths that form between different source-destination pairs by using a flooding process and demonstrated that the delivery time distributions of flooding can be closely approximated even when only a small number $k > 1$ (or a fraction μ) of these paths survive. This remarkable resilience to path failures hints that routing algorithms which choose more than one path perform better. This result also yields insights into previous research that found the human contact network continues to deliver data nearly as well even if many nodes selfishly or selectively refuse to forward data.

5.2 A Unifying Social Information Plane

As it stands, the social information-based networked systems proposed in this dissertation have each been developed separately, to suit the problems at hand. Moving forward, we envision a unifying *Social Information Plane* (SIP) that delivers social information to various network level elements.

The benefits of such an infrastructure can be obtained on at least two levels, as demonstrated in this dissertation: On one level, social information could direct the allocation of resources, as in Chapter 3. At a more fundamental level, networks such as the Pocket Switched Network, which are structured on top of (or embrace the structure of) social networks, can provide guarantees based on the macroscopic properties of the social network, as exemplified in Chapter 4.

In many ways, the idea of an *all-encompassing* social information plane

is akin to the previously proposed knowledge plane [CPRW03]; which is perceived by some as a vague and high-level construct. However, the SIP has a narrower scope than a generic knowledge plane; the social graph provides a recognisable structure which could be harnessed. Furthermore, it is possible to build up a network directly on top of stable macroscopic social properties (Chapter 4), whereas the knowledge plane can only inform decisions made by other network elements. The research question, therefore, is to what extent can we generalise from the current piecemeal solutions to a generic social information plane.

While the concept may end up meaning different things to different people, any instantiation of the SIP will need to resolve the following sources of conflict:

inclusion There are in fact multiple networks with a social angle: "Traditional" social networks like Facebook, email, interest-based networks (such as digg, or last.fm) etc. Therefore social information from multiple sources will need to be aggregated, and any differences must be reconciled by SIP or exposed to the applications that consume the information.

reusability Data provided by SIP needs to be generic enough to be usable by multiple applications; at the same time, it should be specific enough to satisfy the needs of individual applications.

evolvability The social network revolution is still ongoing, and new *kinds* of social networks are constantly emerging (For example, location-based networks like gowalla and foursquare are recent additions). A SIP needs to be generic enough to accept as-yet-unknown forms of social information easily.

Similarly, building and evaluating a SIP will likely involve multiple challenges:

1. Large-scale or macroscopic social properties, which are the most useful, become visible only with a large amount of data, and therefore SIP needs a strong aggregation infrastructure.

2. The piecemeal solutions constructed thus far rely on very little processing power. A generic SIP, by contrast, could need to run compute-intensive calculations.

3. Needless to say, finding ways to disseminate social information while still protecting the privacy and rights of users will be of paramount importance.

From previous experience, finding ways to exploit social structure and information goes hand in hand with mining datasets for new properties, and developing models to gain further insight. Our future work is likely to incorporate all these different facets.

5.3 On adding social network support at a systems level

The work presented in this dissertation applies results from the domain of social networks to build networked systems. We conclude by reflecting on how this differs from prior work on social networks and the consequences it has on traditional systems building.

5.3.1 From network science to engineering

In seeking to find applications for properties of the social network, work such as ours makes a departure from prior lines of research in social networks which focused on *understanding* their properties. Such research comprises both empirical work which map different properties of real social networks, as well as modeling efforts which aim to give generative or mechanistic explanations for why certain properties arise.

In contrast with these previous *network science* studies, we are interested in characterising a property mainly with a view on how applicable it is to a problem at hand. In *network engineering* efforts such as ours, the mechanistic origins of the property are of secondary interest.

However, it is still important to establish that a property holds. For instance, systems such as SybilGuard [YKGF06], SybilLimit [YGKX10], SumUp [TMLS09] and Whanau [LLK10] essentially assume that social networks are fast-mixing and show how this can be used to thwart Sybil attacks. The correctness of such systems depends on the fast-mixing property. However, recent results seem to indicate that the mixing time of real social graphs is much larger than anticipated [MYK10], which implies that current security systems based on fast mixing have weaker utility guarantees or have to be less efficient, with less security guarantees, in order to compensate for the slower mixing.

5.3.2 Consequences of building on social networks

Building a system on top of (or taking advantage of) social networks also represents a departure from traditional systems-building. In socially supported systems, we do not control all parts of the system, and will need to find ways to incorporate the peculiarities of the social network under consideration. Note also that there is a possibility of a feedback loop in that the changes we make to the system could in turn change the properties we rely upon. Systems designers also have to take into account the fact that some properties of social networks can change over time [LKF07].

At a systems level, the *properties of the social property* can also have consequences. For instance, in Chapter 3, we build our system on a "gameable" property[1]. Specifically, we predict that videos whose accesses are predominantly non-viral will be popular, and use this to segregate the popular and unpopular videos to save energy, and also to make selective replication decisions. This measure for predicting popularity could be subverted by a sybil attack which creates a large number of fake identities with no explicit link between them, who then proceed to access an unpopular video, thereby

[1]That the properties of a property can have systems level implications is hardly unique to networked systems with social support. The correctness of TCP, for example, relies on senders responding to congestion indications. This is a gameable property: a selfish sender can simply choose to not respond to such indications.

marking it as popular[2]. The popularity measure is gameable because it is a "local" property over which individual nodes can have undue influence. In contrast, the path resilience we find in Chapter 4 is a "global" property that *emerges* from the network as a whole, and is therefore resistant to such sybil attacks.

5.3.3 Privacy considerations

The viability of social network-based systems depends crucially on large-scale data about social networks. Much of this is highly sensitive Personally Identifiable Information (PII), and needs to be handled carefully. In this dissertation, we do not address this requirement, but leave it to future work as a serious challenge to be addressed (See §5.2). Below, we present some initial thoughts on the issue, based on classifying the dependence on social network data as either a direct runtime dependency on dynamically changing factors, or a more static dependency on some social network property.

When there is a direct runtime dependency, the system typically needs to take different decisions based on the results of computations performed on current (dynamically changing) social network data. One way to handle this securely is to have a trusted third party perform the computation and return only the result to the system. Social networks like Facebook and Myspace already expose APIs through which applications can make specific queries without needing to directly access raw social network data. Indeed, our prototype of Buzztraq was built on an early version of such an API (See §3.6.2). Unfortunately, while this approach works, it has a number of limitations. First, the restricted interface of the API is limited to a few specific queries, and this may be insufficient for some applications. Second, it requires a third party, which presents a problem not only because it is difficult to get everyone to trust the same entity (e.g. not everyone trusts Facebook), but also because the trusted entity represents a single point of failure which can be subjected to concerted and co-ordinated attacks.

[2]While it is outside the scope of the current dissertation, we note that such Sybil attacks can be thwarted by excluding the Sybil nodes using measures like SybilLimit [YGKX10] or SumUp [TMLS09].

A more robust solution would be to store the social network data in an encrypted form and perform the computation using techniques like homomorphic encryption [Gen09]. Although at the moment this solution is not practically viable, researchers are widely hopeful that it is possible to do homomorphic encryption efficiently, and much research is being devoted to this issue. When only aggregated data is required, techniques like privacy-preserving data mining [AS00] could be employed.

The second kind of dependency is a static dependency on some social network property holding. For example, SybilGuard relies on the social network being fast mixing but does not require dynamic information from the social network. Similarly, our suggestion (§4.9) that using a small number $k > 1$ paths simultaneously leads to better delivery time distributions in Pocket Switched Networks does not require any sensitive social network information. However, as we argued in §5.3.1, it is still important in such systems to empirically verify that a property holds. Fortunately, this is only an offline computation and securing this is much easier.

During system development, it may be possible to use synthetic data for prototyping and debugging purposes. §2.1.4 gives more details about this.

* * *

Looking back, we can see a simple design pattern emerge for adding social network support into networked systems: First, we measure social networks, and derive relevant properties from them. Our designs then take into account or exploit such properties.

In the future, as the field matures, we can expect a bag-of-properties approach, in which systems builders choose from a set of well known social properties and apply the relevant ones to their system. This could obviate the first step of needing to measure and establish a relevant social property. Facilities such as the Social Information Plane proposed in the previous section could also play a role in mitigating some of the dangers discussed above. However, we expect that incorporating social properties into a system would still involve interesting design challenges, and that the application of a property to a given application domain might require creative thinking and non-trivial insights.

Bibliography

[ABD11] S. Akhshabi, A. C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In *Proc. ACM Multimedia Systems*, 2011. Cited on page 73.

[ACW10] S. Allen, G. Colombo, and R. Whitaker. Uttering: social micro-blogging without the internet. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, pages 58–64. ACM, 2010. Cited on page 95.

[AGG+07] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. R. Rodriguez. Is high-quality vod feasible using p2p swarming? In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 903–912, New York, NY, USA, 2007. ACM. Cited on page 72.

[AJB00] R. Albert, H. Jeong, and A.-L. Barabàsi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000. Cited on page 138.

[AJZ10] V. Adhikari, S. Jain, and Z. Zhang. YouTube Traffic Dynamics and Its Interplay with a Tier-1 ISP: An ISP Perspective. 2010. Cited on page 81.

[Aka00] Akamai Technologies. Fast internet content delivery with freeflow, April 2000. Cited on page 74.

[AM79] R. M. Anderson and R. M. May. Population biology of infectious diseases: Part i. *Nature*, pages 361—67, 1979. Cited on page 84.

[AMS09] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in

dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51), 2009. Cited on pages 30 and 61.

[And04] C. Anderson. The long tail. *Wired*, 12(10), October 2004. Cited on pages 40 and 42.

[And06] C. Anderson. *The Long Tail: Why the future of business is selling less of more*. Hyperion Books, 2006. Cited on page 42.

[And08] C. Anderson. Debating the long tail. http://blogs.hbr.org/cs/2008/06/debating_the_long_tail.html, June 2008. Cited on page 44.

[AS00] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 439–450, New York, NY, USA, 2000. ACM. Cited on page 148.

[AS10] A. Abhari and M. Soraya. Workload generation for YouTube. *Multimedia Tools and Applications*, 46(1):91–118, 2010. Cited on page 80.

[BA99] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999. Cited on page 21.

[BCNS03] A. Barbir, B. Cain, R. Nair, and O. Spatscheck. Known content network (cn) request-routing mechanisms. RFC 3568, July 2003. Cited on page 74.

[bE08] d. boyd and N. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2008. Cited on pages 17 and 19.

[BGLL08] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, page P10008, 2008. Cited on pages 27 and 135.

[BGPS05] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis and applications. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1653–1664. IEEE, 2005. Cited on page 33.

[BGPS06] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *Information Theory, IEEE Transactions on*, 52(6):2508–2530, 2006. Cited on page 33.

[BHS06] E. Brynjolfsson, Y. Hu, and M. Smith. From niches to riches: Anatomy of the long tail. *MIT Sloan Management Review*, 47(4):67, 2006. Cited on page 43.

[BHS07] E. Brynjolfsson, Y. Hu, and D. Simester. Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales. *Working Paper, Sloan School of Management, MIT*, 2007. Cited on page 43.

[BHS10a] E. Brynjolfsson, Y. Hu, and M. Smith. Research commentary—long tails vs. superstars: The effect of information technology on product variety and sales concentration patterns. *Information Systems Research*, 21(4):736–747, 2010. Cited on page 42.

[BHS10b] E. Brynjolfsson, Y. Hu, and M. Smith. The Longer Tail: The Changing Shape of Amazon's Sales Distribution Curve. *Working Paper, Sloan School of Management, MIT*, 2010. Cited on page 43.

[Bia11] L. Bianchi. The history of social networking. http://www.viralblog.com/research/the-history-of-social-networking/, January 2011. Last accessed, March 2011. Cited on page 20.

[Big09] B. Biggs. 1080p hd is coming to youtube, November 2009. YouTube Blog, http://youtube-global.blogspot.com/2009/11/1080p-hd-comes-to-youtube.html. Cited on page 62.

[BKC08] M. Boguñá, D. Krioukov, and K. Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2008. Cited on page 26.

[BKM+00] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33(1-6):309–320, 2000. Cited on page 138.

[BLV07] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *SIGCOMM*, 2007. Cited on page 136.

[BMBR11] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. On the penetration of online social networks: Exploiting the social structure. Under submission. Technical Report available online from http://lersse-dl.ece.ubc.ca/record/258/, March 2011. Cited on page 30.

[Bol01] B. Bollobás. *Random graphs*. Cambridge University Press, 2001. Cited on page 108.

[BP70] M. T. Boswell and G. P. Patil. *Random Counts for Scientific Work*, volume 1, chapter Chance Mechanisms Generating the Negative Binomial Distribution. Pennsylvania State University Press, 1970. Cited on page 121.

[BR05] P. Boykin and V. Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38(4):61–68, 2005. Cited on page 35.

[BRCA09] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 49–62. ACM, 2009. Cited on page 32.

[BS03] E. Brynjolfsson and M. Smith. Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science*, 49(11):1580–1596, 2003. Cited on pages 43 and 45.

[BSBK09] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 551–560, New York, NY, USA, 2009. ACM. Cited on page 30.

[CCH+08] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168. ACM, 2008. Cited on page 29.

[CDK+03] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: high-bandwidth multicast

in cooperative environments. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 298–313. ACM, 2003. Cited on page 72.

[CDL08] X. Cheng, C. Dale, and J. Liu. Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pages 229–238. IEEE, 2008. Cited on page 80.

[CF06] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys (CSUR)*, 38(1):2, 2006. Cited on page 24.

[CG02] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–11, 2002. Cited on page 70.

[CHBG10] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, 2010. Cited on page 23.

[CHC+07] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, pages 606–620, 2007. Cited on pages 118, 136, and 137.

[CKE+08] H. Chun, H. Kwak, Y. Eom, Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs interaction: a case study of cyworld. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 57–70. ACM, 2008. Cited on page 31.

[CKK02] Y. Chen, R. Katz, and J. Kubiatowicz. Dynamic replica placement for scalable content delivery. *Peer-to-Peer Systems*, pages 306–318, 2002. Cited on page 75.

[CKR+07] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007. Cited on page 80.

[CKR+09] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *Networking, IEEE/ACM Transactions on*, 17(5):1357–1370, 2009. Cited on pages 22, 58, and 80.

[CL06] F. Chung and L. Lu. *Complex graphs and networks*. American Mathematical Society, 2006. Cited on page 108.

[CMAG08] M. Cha, A. Mislove, B. Adams, and K. P. Gummadi. Characterizing social cascades in flickr. In *Proceedings of the first ACM Workshop on Online social networks (WOSN)*, 2008. Cited on page 78.

[CMG09a] M. Cha, A. Mislove, and K. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proc. of the 18th International World Wide Web (WWW) Conference*, 2009. Cited on page 22.

[CMG09b] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 721–730, New York, NY, USA, 2009. ACM. Cited on page 62.

[CMP11] L. D. Cicco, S. Mascolo, and V. Palmisano. Feedback control for adaptive live video streaming. In *Proc. ACM Multimedia Systems*, 2011. Cited on page 73.

[CPRW03] D. Clark, C. Partridge, J. Ramming, and J. Wroclawski. A knowledge plane for the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–10. ACM, 2003. Cited on page 144.

[CQC+03] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. Katz. Efficient and adaptive web replication using content clustering. *IEEE Journal on Selected Areas in Communications*, 21(6):979–994, 2003. Cited on pages 76 and 77.

[CSZ07] M. Chen, L. Stein, and Z. Zhang. Dependability, access diversity, low cost: pick two. In *HotDep'07: Proceedings of the 3rd workshop on on Hot Topics in System Dependability*, 2007. Cited on page 70.

[Dav67] J. Davis. Clustering and Structural Balance in Graphs. *Human relations*, 1967. Cited on page 28.

[DBN03] B. Deb, S. Bhatnagar, and B. Nath. Reinform: Reliable information forwarding using multiple paths in sensor networks. In *LCN '03: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, page 406, 2003. Cited on page 138.

[DCM10] L. De Cicco and S. Mascolo. An experimental investigation of the akamai adaptive video streaming. In G. Leitner, M. Hitz, and A. Holzinger, editors, *HCI in Work and Learning, Life and Leisure*, volume 6389 of *Lecture Notes in Computer Science*, pages 447–464. Springer Berlin / Heidelberg, 2010. Cited on page 73.

[DDGDA05] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005:P09008, 2005. Cited on page 27.

[DH07] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *MobiHoc*, 2007. Cited on page 137.

[Dig] Digg. Digg faq. http://digg.com/faq. Last accessed Feb 26, 2011. Cited on page 60.

[DM09] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. *NDSS. The Internet Society*, 2009. Cited on pages 32 and 37.

[DMP+02] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. *Internet Computing, IEEE*, 6(5):50–58, 2002. Cited on page 74.

[Do07] C. Do. Youtube scalability. http://video.google.com/videoplay?docid=-6304964351441328559, Last accessed, Feb 26, 2011., June 2007. Cited on page 40.

[Dou02] J. R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, 2002. Springer-Verlag. Cited on page 35.

[Dun93] R. I. M. Dunbar. Co-evolution of neocortex size, group size and language in humans. *Behavioral and Brain Sciences*, 16, 1993. Cited on page 97.

[ECCD07] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot. Diversity of forwarding paths in pocket switched networks. In *Proceedings of ACM Internet Measurement Conference*, October 2007. Cited on page 137.

[ECCD08] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *MobiHoc*, 2008. Cited on page 137.

[Elb08a] A. Elberse. Should you invest in the long tail. *Harvard Business Review*, 86(7/8):88–96, 2008. Cited on pages 43 and 44.

[Elb08b] A. Elberse. A taste for obscurity: An individual-level examination of "long tail" consumption. *Working Paper, Harvard Business School*, 2008. Cited on page 43.

[EOG06] A. Elberse and F. Oberholzer-Gee. *Superstars and underdogs: An examination of the long tail phenomenon in video sales.* Division of Research, Harvard Business School, 2006. Cited on pages 43 and 46.

[EP] N. Eagle and A. S. Pentland. CRAWDAD data set mit/reality (v. 2005-07-01). http://crawdad.cs.dartmouth.edu/mit/reality. Cited on pages 19, 20, and 97.

[ES09] A. Elberse and D. Schweidel. Who wants what's hot? popularity profiles and customer value. *Working Paper, Harvard Business School*, 2009. Cited on page 43.

[EYR11] V. Erramilli, X. Yang, and P. Rodriguez. Explore what-if scenarios with SONG: Social Network Write Generator. *Arxiv preprint arXiv:1102.0699*, 2011. Cited on page 24.

[Fac] Facebook statistics. http://www.facebook.com/press/info.php?statistics. Last accessed in Nov 2010. Cited on pages 20 and 82.

[Fal03] K. Fall. A delay-tolerant network architecture for challenged internets. In *"SIGCOMM"*, 2003. Cited on page 136.

[FC95] R. Frank and P. Cook. *The winner-take-all society: Why the few at the top get so much more than the rest of us.* Penguin Group USA, 1995. Cited on page 43.

[FC07] S. Fortunato and C. Castellano. Community structure in graphs. *Arxiv preprint arXiv:0712.2716*, 2007. Cited on page 27.

[FCW43] R. A. Fisher, A. S. Corbet, and C. B. Williams. The relation between the number of species and the number of individuals in a random sample of an animal population. *Journal of Animal Ecology*, 12(1):42–58, 1943. Cited on page 122.

[FFM04] M. Freedman, E. Freudenthal, and D. Mazieres. Democratizing content publication with Coral. In *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation-Volume 1*, page 18. USENIX Association, 2004. Cited on pages 75 and 77.

[FH09] D. Fleder and K. Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science*, 55(5):697–712, 2009. Cited on page 43.

[Fla06] A. Flaxman. Expansion and lack thereof in randomly perturbed graphs. Technical Report MSR-TR-2006-118, Microsoft Research, 2006. Cited on page 34.

[FLGC02] G. Flake, S. Lawrence, C. Giles, and F. Coetzee. Self-organization and identification of web communities. *Computer*, 35(3):66–70, 2002. Cited on page 27.

[FLW08] F. Fu, L. Liu, and L. Wang. Empirical analysis of online social networks in the age of Web 2.0. *Physica A: Statistical Mechanics and its Applications*, 387(2-3):675–684, 2008. Cited on page 22.

[GALM07] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 15–28. ACM, 2007. Cited on page 80.

[GBGP10] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *Proceedings of the third ACM international conference on Web*

search and data mining, pages 201–210. ACM, 2010. Cited on page 53.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM. Cited on page 148.

[GHB08] M. González, C. Hidalgo, and A. Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008. Cited on page 23.

[GKF+06] S. Garriss, M. Kaminsky, M. Freedman, B. Karp, D. Mazières, and H. Yu. RE: reliable email. In *Proceedings of the 3rd conference on Networked Systems Design & Implementation*, pages 22–22, 2006. Cited on pages 31 and 35.

[GKRT04] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004. Cited on page 28.

[Gla02] M. Gladwell. *Tipping Point*. Back Bay Books, 2002. Cited on pages 9 and 58.

[GN02] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821, 2002. Cited on page 27.

[GNK05] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Perform. Eval.*, 62(1-4):210–228, 2005. Cited on page 137.

[Gra73] M. Granovetter. The strength of weak ties. *The American journal of sociology*, 78(6):1360–1380, 1973. Cited on pages 12, 92, 103, 138, and 143.

[GT02] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, 2002. Cited on page 94.

[GWBB07] L. Ganesh, H. Weatherspoon, M. Balakrishnan, and K. Birman. Optimizing power consumption in large scale storage

systems. In *11th USENIX Workshop on Hot Topics in Operating Systems*, May 2007. Cited on pages 67 and 70.

[GZS+03] S. Gurumurthi, J. Zhang, A. Sivasubramaniam, M. Kandemir, H. Franke, N. Vijaykrishnan, and M. J. Irwin. Interplay of energy and performance for disk arrays running transaction processing workloads. In *ISPASS '03: Proceedings of the 2003 IEEE International Symposium on Performance Analysis of Systems and Software*, 2003. Cited on page 70.

[Har53] F. Harary. On the notion of balance of a signed graph. *The Michigan Mathematical Journal*, 2(2):143–146, 1953. Cited on page 29.

[HCS+05] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251. ACM, 2005. Cited on pages 2, 91, and 93.

[HCY08] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *MobiHoc*, 2008. Cited on page 137.

[Hec10] O. Heckmann. Personal Communication, April 2010. Mr. Heckmann is Director, Google Engineering, Zurich. Cited on page 40.

[Hei46] F. Heider. Attitudes and cognitive organization. *Journal of psychology*, 21(2):107–112, 1946. Cited on page 28.

[HEL04] P. Holme, C. Edling, and F. Liljeros. Structure and time evolution of an Internet dating community. *Social Networks*, 26(2):155–174, 2004. Cited on page 21.

[HJ06] M. S. Handcock and J. H. Jones. Interval estimates for epidemic thresholds in two-sex network models. *Theoretical Population Biology*, 70(2):125 – 134, 2006. Cited on pages 21 and 122.

[HLL+07] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A measurement study of a large-scale P2P IPTV system. *Multimedia, IEEE Transactions on*, 9(8):1672–1687, 2007. Cited on page 72.

[HP07] J. Hennessy and D. Patterson. *Computer architecture: a quantitative approach*. Morgan Kaufmann, 4 edition, 2007. Cited on page 63.

[HWLR08] C. Huang, A. Wang, J. Li, and K. Ross. Measuring and evaluating large-scale CDNs. In *Proc. of IMC*, 2008. Paper withdrawn at Microsoft's request. Cited on pages 40 and 75.

[HWSB08] T. Hogg, D. Wilkinson, G. Szabo, and M. Brzozowski. Multiple relationship types in online communities and social networks. In *Proc. of the AAAI Symposium on Social Information Processing*, pages 30–35, 2008. Cited on page 28.

[HXL+09] P. Hui, K. Xu, V. Li, J. Crowcroft, V. Latora, and P. Lio. Selfishness, altruism and message spreading in mobile social networks. In *Proc. of First IEEE International Workshop on Network Science For Communication Networks (NetSciCom09)*, 2009. Cited on pages 13, 137, and 138.

[ICM09] S. Ioannidis, A. Chaintreau, and L. Massoulié. Optimal and scalable distribution of content updates over a mobile social network. In *IEEE INFOCOM*, pages 1422–1430, 2009. Cited on page 95.

[IKK+00] C. Isbell, M. Kearns, D. Kormann, S. Singh, and P. Stone. Cobot in LambdaMOO: A social statistics agent. In *Proceedings of the National Conference on Artificial Intelligence*, pages 36–41. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000. Cited on pages 21 and 32.

[JFP04] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM*, 2004. Cited on page 136.

[JJJ+02] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentation. In *IEEE INFOCOM*, volume 1, pages 295–304. IEEE, 2002. Cited on page 75.

[JJK+02] S. Jamin, C. Jin, A. Kurc, D. Raz, and Y. Shavitt. Constrained mirror placement on the Internet. In *IEEE INFOCOM*, volume 1, pages 31–40. IEEE, 2002. Cited on page 75.

[JSFT07] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA, 2007. ACM. Cited on page 23.

[JWW+10] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, and B. Zhao. Understanding latent interactions in online social networks. In *Proc. of the ACM SIGCOMM Internet Measurement Conference*. Citeseer, 2010. Cited on page 32.

[KCE+09] H. Kwak, Y. Choi, Y.-H. Eom, H. Jeong, and S. Moon. Mining communities in networks: a solution for consistency and its evaluation. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 301–314, New York, NY, USA, 2009. ACM. Cited on page 27.

[KGA08] B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, pages 19–24. ACM, 2008. Cited on pages 23 and 55.

[KGNR10] T. Karagiannis, C. Gkantsidis, D. Narayanan, and A. Rowstron. Hermes: clustering users in large-scale e-mail services. In *Proceedings of the 1st ACM symposium on Cloud computing*, SoCC '10, pages 89–100, New York, NY, USA, 2010. ACM. Cited on page 37.

[Kir06] M. Kirkpatrick. Limelight networks lands $130m more to deliver the web's future. http://techcrunch.com/2006/07/26/limelight-networks-lands-130m-more-to-deliver-the-webs-future/, Jul 2006. Cited on page 40.

[Kir10] D. Kirkpatrick. *The Facebook Effect: The Inside Story of the Company That Is Connecting the World*. Simon & Schuster, 2010. Cited on page 31.

[KKK05] N. Kotz, A. Kemp, and S. Kotz. *Univariate Discrete Distributions*. Wiley, 3 edition, 2005. Cited on page 132.

[KLB09] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The Slashdot Zoo: Mining a social network with negative edges. In *Proceed-*

ings of the 18th international conference on World wide web, pages 741–750. ACM, 2009. Cited on page 28.

[KLBV07] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic. Power law and exponential decay of inter contact times between mobile devices. In *MOBICOM*, 2007. Cited on pages 118, 136, and 137.

[Kle00] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Annual ACM symposium on theory of computing*, volume 32, pages 163–170. Citeseer, 2000. Cited on pages 26 and 33.

[Kle06] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM)*, volume 3, pages 1019–1044, 2006. Cited on page 26.

[KLPM10] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010. Cited on page 23.

[KMS+09] R. Krishnan, H. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving beyond end-to-end path information to optimize CDN performance. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 190–201. ACM, 2009. Cited on page 80.

[KMT07] P. Key, L. Massoulie, and P. Towsley. Path selection and multipath congestion control. In *IEEE INFOCOM*, 2007. Cited on page 138.

[KPR99] M. Korupolu, C. Plaxton, and R. Rajaraman. Placement algorithms for hierarchical cooperative caching. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 586–595. Society for Industrial and Applied Mathematics, 1999. Cited on page 76.

[KRAV03] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat. Bullet: High bandwidth data dissemination using an overlay mesh. *ACM SIGOPS Operating Systems Review*, 37(5):282–297, 2003. Cited on page 72.

[KRR02] J. Kangasharju, J. Roberts, and K. Ross. Object replication strategies in content distribution networks. *Computer Communications*, 25(4):376–383, 2002. Cited on page 76.

[KRS00] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. *IEEE/ACM Transactions on Networking (TON)*, 8(5):568–582, 2000. Cited on page 75.

[KSS97] H. Kautz, B. Selman, and M. Shah. Referral Web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997. Cited on page 21.

[LAH07] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5, 2007. Cited on pages 22 and 62.

[Lam83] B. Lampson. Hints for computer system design. *ACM SIGOPS Operating Systems Review*, 17(5):33–48, 1983. Cited on page 4.

[LDS04] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *Proc. SAPIR Workshop*, 2004. Cited on page 137.

[LEA+01] F. Liljeros, C. Edling, L. Amaral, H. Stanley, and Y. Åberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, 2001. Cited on page 21.

[Lei09] T. Leighton. Improving performance on the internet. *Communications of the ACM*, 52(2):44–51, 2009. Cited on pages 74 and 75.

[Lev00] R. Levien. AdvogatoÕs trust metric, 2000. Available online from `http://www.advogato.org/trust-metric.html`. Cited on page 22.

[LF07] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *Proceedings of the 24th international conference on Machine learning*, pages 497–504. ACM, 2007. Cited on page 24.

[LG10] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Proceedings of the Fourth International AAAI*

Conference on Weblogs and Social Media (ICWSM). AAAI, 2010. Cited on page 62.

[LGI+02] B. Li, M. Golin, G. Italiano, X. Deng, and K. Sohraby. On the optimal placement of web proxies in the internet. In *IEEE INFOCOM*, volume 3, pages 1282–1290, 2002. Cited on page 75.

[LH08] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *Proceeding of the 17th international conference on World Wide Web*, pages 915–924. ACM, 2008. Cited on pages 20 and 21.

[LHK10] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1361–1370. ACM, 2010. Cited on page 29.

[LIJM+10] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. In *ACM SIGCOMM 2010*, pages 75–86, New York, NY, USA, 2010. ACM. Cited on page 40.

[Lim09] Limelight Networks. Limelight networks cdn: Network overview. Limelight Networks White Paper. Available online at `http://www.limelightnetworks.com/resources/LLNW_Network_Overview.pdf`, 2009. Cited on page 74.

[LKF07] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007. Cited on pages 24 and 146.

[LLDM08] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceeding of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008. Cited on page 27.

[LLK10] C. Lesniewski-Laas and M. Kaashoek. Whanau: A sybil-proof distributed hash table. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, page 8. USENIX Association, 2010. Cited on pages 32, 37, and 146.

[LLM10] J. Leskovec, K. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010. Cited on page 27.

[Lon11] A brief cartoon history of social networking 1930–2011. http://www.slideshare.net/peoplebrowsr/a-brief-cartoon-history-of-social-networking-19302011, March 2011. Commissioned by Peoplebrowsr. Illustrated by Adam Long. Additional commentary and highlights can be found on the Peoplebrowsr blog: http://blog.peoplebrowsr.com/blog/?p=780. Both URLs were last accessed in March 2011. Cited on page 20.

[LSO+07] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros. Distributed placement of service facilities in large-scale networks. In *IEEE INFOCOM*, pages 2144–2152, 2007. Cited on page 75.

[LWLZ10] Z. Liu, C. Wu, B. Li, and S. Zhao. Uusee: Large-scale operational on-demand streaming with random network coding. In *IEEE INFOCOM*, pages 1–9, 2010. Cited on page 72.

[LY07] H. Lam and D. Yeung. A learning approach to spam detection based on social networks. In *4th Conference on Email and Anti-Spam (CEAS)*, 2007. Cited on page 35.

[Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967. Cited on page 86.

[Mah01] R. Mahajan. How akamai works. http://research.microsoft.com/en-us/um/people/ratul/akamai.html., 2001. Last accessed Feb 26, 2011. Cited on pages 74 and 79.

[McM06] R. McMillan. Phishing attack targets myspace users. Available online at http://www.infoworld.com/d/security-central/phishing-attack-targets-myspace-users-614., October 2006. Last accessed January 2011. Cited on page 31.

[McP63] W. McPhee. *Formal theories of mass behavior*. Free Press of Glencoe New York, 1963. Cited on page 43.

[MIK+04] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538, 2004. Cited on page 25.

[MMC08] L. McNamara, C. Mascolo, and L. Capra. Media Sharing based on Colocation Prediction in Urban Transport. In *MOBICOM*, 2008. Cited on page 95.

[MMG+07] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 29–42, New York, NY, USA, 2007. ACM. Cited on page 22.

[MN05] J. W. Mickens and B. D. Noble. Modeling epidemic spreading in mobile environments. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, 2005. Cited on page 137.

[MPDG08] A. Mislove, A. Post, P. Druschel, and K. Gummadi. Ostra: Leveraging trust to thwart unwanted communication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 15–30. USENIX Association, 2008. Cited on page 35.

[MR95] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random structures and algorithms*, 6(2-3):161–180, 1995. Cited on page 109.

[MR98] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Comb. Probab. Comput.*, 7:295–305, September 1998. Cited on page 109.

[MSLC01] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27:415–444, 2001. Cited on page 29.

[MSN05] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, MobiCom '05, pages 243–257, New York, NY, USA, 2005. ACM. Cited on page 95.

[MSOI+02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824, 2002. Cited on page 25.

[MYK10] A. Mohaisen, A. Yun, and Y. Kim. Measuring the mixing time of social graphs. In *ACM SIGCOMM Conference on Internet Measurements. ACM*, 2010. Cited on pages 33 and 146.

[Nag07] S. Nagaraja. Anonymity in the wild: Mixes on unstructured networks. In *Privacy Enhancing Technologies*, pages 254–271. Springer, 2007. Cited on page 33.

[Nag10] S. Nagaraja. Privacy amplification with social networks. In *Security Protocols*, pages 58–73. Springer, 2010. Cited on page 37.

[NDR08] D. Narayanan, A. Donnelly, and A. Rowstron. Write off-loading: Practical power management for enterprise storage. *Trans. Storage*, 4(3), 2008. Cited on page 70.

[New01] M. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404, 2001. Cited on page 21.

[New02] M. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002. Cited on page 25.

[New03] M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126, Feb 2003. Cited on page 25.

[New04] M. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38:321–330, 2004. 10.1140/epjb/e2004-00124-y. Cited on page 27.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb 2004. Cited on pages 27 and 134.

[Nic09] C. Nickson. The history of social networking. http://www.digitaltrends.com/features/the-history-of-social-networking/, January 2009. Last accessed, March 2011. Cited on page 20.

[NP03] M. Newman and J. Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3):036122, 2003. Cited on page 27.

[NSW01] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64(2):026118, Jul 2001. Cited on page 109.

[NTD+09] D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron. Migrating server storage to SSDs: analysis of tradeoffs. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 145–158. ACM, 2009. Cited on page 63.

[Ofg11] Ofgem. Typical domestic energy consumption figures. Factsheet 96, January 2011. Available from http://www.ofgem.gov.uk/Media/FactSheets/Documents1/domestic%20energy%20consump%20fig%20FS.pdf. Cited on page 63.

[OSH+07] J. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A. Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332, 2007. Cited on pages 23 and 138.

[OSS09] G. Oestreicher-Singer and A. Sundararajan. Recommendation networks and the long tail of electronic commerce. *Working Papers*, 2009. Cited on pages 43 and 46.

[PB04] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In *ICS '04: Proceedings of the 18th annual international conference on Supercomputing*, pages 68–78, 2004. Cited on pages 65 and 70.

[PB07] A.-M. K. Pathan and R. Buyya. A taxonomy and survey of content delivery networks. Technical report, University of Melbourne, 2007. Cited on page 74.

[PBMW99] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120. Cited on page 19.

[PBV07] G. Palla, A. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007. Cited on page 23.

[PES+10] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez. The little engine(s) that could: scaling online social networks. In *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, SIGCOMM '10, pages 375–386, New York, NY, USA, 2010. ACM. Cited on page 37.

[PP04] K. Park and V. Pai. Deploying large file transfer on an http content distribution network. In *Proceedings of the First Workshop on Real, Large Distributed Systems (WORLDS'04)*, 2004. Cited on pages 75 and 77.

[PSM11] A. Post, V. Shah, and A. Mislove. Bazaar: Strengthening user reputations in online marketplaces. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, 2011. Cited on page 35.

[PV06] G. Pallis and A. Vakali. Insight and perspectives for content delivery networks. *Commun. ACM*, 49:101–106, January 2006. Cited on pages 74, 76, and 77.

[PVS+05] G. Pallis, A. Vakali, K. Stamos, A. Sidiropoulos, D. Katsaros, and Y. Manolopoulos. A latency-based object placement approach in content distribution networks. *Web Congress, Latin American*, pages 140–147, 2005. Cited on page 76.

[PVS06] G. Pierre and M. Van Steen. Globule: a collaborative content delivery network. *Communications Magazine, IEEE*, 44(8):127–133, 2006. Cited on pages 75 and 77.

[QPV02] L. Qiu, V. Padmanabhan, and G. Voelker. On the placement of web server replicas. In *IEEE INFOCOM*, volume 3, pages 1587–1596. IEEE, 2002. Cited on page 75.

[QWB+09] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the Electric Bill for Internet-Scale Systems. In *ACM SIGCOMM*, Barcelona, Spain, August 2009. Cited on page 62.

[Raj08] A. Rajaraman. The real long tail: Why both chris anderson and anita elberse are wrong. http://anand.typepad.com/datawocky/2008/07/the-real-long-tail-why-both-chris-anderson-and-anita-elberse-are-wron html, July 2008. Cited on page 45.

[RCC+04] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658, 2004. Cited on page 27.

[RD02] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 70. ACM, 2002. Cited on page 22.

[Rei09] R. Reinhardt. *Video with Adobe Flash CS4 Professional Studio Techniques*. Studio Techniques. Adobe Press, 1st edition, April 2009. Excerpt available online at http://www.adobepress.com/articles/printerfriendly.asp?p=1381886. Last accessed Feb 26, 2011. Cited on page 73.

[Ros81] S. Rosen. The economics of superstars. *The American Economic Review*, 71(5):845–858, 1981. Cited on page 43.

[Sch72] T. Schelling. A process of residential segregation: neighborhood tipping. *Racial discrimination in economic life*, pages 157–84, 1972. Cited on page 29.

[Sco00] J. P. Scott. *Social network analysis: A Handbook*. Sage Publications, 2000. Cited on page 8.

[SCW+10] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Zhao. Measurement-calibrated graph models for social network experiments. In *Proceedings of the 19th international conference on World wide web*, pages 861–870. ACM, 2010. Cited on page 24.

[SDW06] M. Salganik, P. Dodds, and D. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854, 2006. Cited on page 44.

[SH03] T. Small and Z. J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *MobiHoc*, 2003. Cited on pages 117 and 137.

[Sha09] S. Shalunov. Low Extra Delay Background Transport (LEDBAT). *IETF Draft, Mar*, 2009. Cited on page 77.

[Sim09] M. Simon. The complete history of social networking – cbbs to twitter. http://www.maclife.com/article/feature/complete_history_social_networking_cbbs_twitter, December 2009. Last accessed, March 2011. Cited on page 20.

[SLT10] M. Szell, R. Lambiotte, and S. Thurner. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*, 107(31):13636, 2010. Cited on page 29.

[SMMC11] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades. In *Proceedings of the 20th international conference on World Wide Web*, WWW '11, 2011. Cited on page 80.

[SMS+08] M. Seshadri, S. Machiraju, A. Sridharan, J. Bolot, C. Faloutsos, and J. Leskove. Mobile call graphs: beyond power-law and log-normal distributions. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 596–604. ACM, 2008. Cited on page 23.

[sop09] Facebook users at risk of "rubber duck" identity attack: 46% of users happy to reveal all to complete strangers. Sophos Press Release. http://www.sophos.com/pressoffice/news/articles/2009/12/facebook.html, December 2009. Last accessed January 2011. Cited on page 30.

[SPR08] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Trans. Netw.*, 16(1), 2008. Cited on page 137.

[SPvS06] M. Szymaniak, G. Pierre, and M. van Steen. Latency-driven replica placement. *IPSJ Journal*, 47(8), August 2006. http://www.globule.org/publi/LDRP_ipsj2006.html. Cited on pages 77 and 84.

[SRL98] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326, April 1998. Cited on page 73.

[SRML09] E. Sun, I. Rosenn, C. Marlow, and T. Lento. Gesundheit! modeling contagion through facebook news feed. In *Proc. of International AAAI Conference on Weblogs and Social Media*, 2009. Cited on page 22.

[Sta95] W. Stallings. PGP web of trust. *Byte*, 20(2):161–162, 1995. Cited on page 22.

[STCR10] A. Silberstein, J. Terrace, B. Cooper, and R. Ramakrishnan. Feeding frenzy: selectively materializing users' event feeds. In *Proceedings of the 2010 international conference on Management of data*, pages 831–842. ACM, 2010. Cited on page 37.

[SW93] M. Schwartz and D. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, 1993. Cited on page 21.

[SWB+08] G. Swamynathan, C. Wilson, B. Boe, K. Almeroth, and B. Zhao. Do social networks improve e-commerce?: a study on social marketplaces. In *Proceedings of the first workshop on Online social networks*, pages 1–6. ACM, 2008. Cited on page 22.

[TFK+11] R. Torres, A. Finamore, J. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting video server selection strategies in the youtube cdn. ECE Technical Reports 408, Purdue University, January 2011. Cited on page 40.

[TLSC11] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal Sybil-resilient node admission control. In *INFOCOM 2011. 30th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. IEEE, 2011. Cited on pages 36 and 37.

[TM69] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969. Cited on pages 26 and 97.

[TMLS09] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 15–28. USENIX Association, 2009. Cited on pages 37, 146, and 147.

[TN09] T. Tan and S. Netessine. Is tom cruise threatened? using netflix prize data to examine the long tail of electronic commerce. *Working Paper, University of Pennsylvania*, 2009. Cited on pages 43 and 46.

[Tse05] S. Tse. Approximate algorithms for document placement in distributed web servers. *IEEE Transactions on Parallel and Distributed Systems*, pages 489–496, 2005. Cited on page 76.

[TSJ07] A. Tahbaz-Salehi and A. Jadbabaie. Small world phenomenon, rapidly mixing markov chains, and average consensus algorithms. In *46th IEEE Conference on Decision and Control*, pages 276–281. IEEE, 2007. Cited on page 33.

[Tun09] D. Tunkelang. The Noisy Channel Blog. http://thenoisychannel.com/2009/01/13/a-twitter-analog-to-pagerank/. Online implementation available from http://tunkrank.com/, January 2009. Last accessed January 2011. Cited on page 19.

[UCS04] UCSD. Wireless topology discovery project. http://sysnet.ucsd.edu/wtd/wtd.html. Last accessed Feb 26, 2011, 2004. Cited on pages 19 and 97.

[Vin75] T. Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, XXIII(176), April 1975. Cited on page 86.

[VKD02] A. Venkataramani, R. Kokku, and M. Dahlin. TCP Nice: A mechanism for background transfers. *ACM SIGOPS Operating Systems Review*, 36(SI):329–343, 2002. Cited on page 77.

[VMCG09] B. Viswanath, A. Mislove, M. Cha, and K. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 37–42. ACM, 2009. Cited on pages 23 and 32.

[VP03] A. Vakali and G. Pallis. Content delivery networks: Status and trends. *Internet Computing, IEEE*, 7(6):68–74, 2003. Cited on page 74.

[VPGM10] B. Viswanath, A. Post, K. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 40(4):363–374, 2010. Cited on page 37.

[VWD01] A. Venkataramani, P. Weidmann, and M. Dahlin. Bandwidth constrained placement in a WAN. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 134–143. ACM, 2001. Cited on page 76.

[VYK+02] A. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif, and M. Dahlin. The potential costs and benefits of long-term prefetching for content distribution. *Computer Communications*, 25(4):367–375, 2002. Cited on page 77.

[Wag07] T. Wagemanns. The mobilisation of amateur music in convergence culture. Master's thesis, Maastricht University, 2007. Cited on page 45.

[Wal10] H. Walk. Oops pow surprise ... 24 hours of video all up in your eyes! YouTube Blog, http://youtube-global.blogspot.com/2010/03/oops-pow-surprise24-hours-of-video-all.html, March 2010. Cited on page 62.

[WBS+09] C. Wilson, B. Boe, A. Sala, K. Puttaswamy, and B. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 205–218. ACM, 2009. Cited on pages 23 and 31.

[Wei08] K. Weinen. The 'Do-it-yourself' Guide to Stardom: Artist-made music-marketing on the web 2.0 social network MySpace. Fast Forward online magazine, published by author, 2008. Available from http://www.fastforwardmag.eu/Do_it_yourself.html. Accessed 14 Feb 2011. Cited on page 45.

[WF95] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*. Cambridge university press, 1995. Cited on pages 8 and 27.

[WIVB10] M. Wattenhofer, Y. Interian, J. Vaver, and T. Broxton. Catching a viral video. In *Proc. Workshop on Social Interaction Analysis and Service Providers (SIASP), colocated with ICDM*, 2010. Cited on page 62.

[WLJH10] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010. Cited on page 20.

[Wor09] J. Wortham. The value of a facebook friend? about 37 cents. http://bits.blogs.nytimes.com/2009/01/09/are-facebook-friends-worth-their-weight-in-beef/, January 2009. Last accessed January 2011. Cited on page 30.

[WPD+10] M. P. Wittie, V. Pejovic, L. Deek, K. C. Almeroth, and B. Y. Zhao. Exploiting locality of interest in online social networks. In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 25:1–25:12, New York, NY, USA, 2010. ACM. Cited on page 37.

[WPP+04] L. Wang, K. Park, R. Pang, V. Pai, and L. Peterson. Reliability and security in the CoDeeN content distribution network. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, page 14. USENIX Association, 2004. Cited on pages 75 and 77.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. Cited on pages 26 and 114.

[YGKX10] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xia. SybilLimit: A Near-Optimal social network defense against sybil attacks. *IEEE/ACM Transactions on Networking*, 18(3):885–898, June 2010. Cited on pages 32, 36, 146, and 147.

[YHC07] E. Yoneki, P. Hui, and J. Crowcroft. Visualizing community detection in opportunistic networks. In *CHANTS'07: Proc. of the second ACM workshop on Challenged Networks*, pages 93–96, 2007. Cited on page 98.

[YKGF06] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278. ACM, 2006. Cited on pages 31, 32, 33, 36, and 146.

[YZZZ06] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. *SIGOPS Oper. Syst. Rev.*, 40(4):333–344, 2006. Cited on page 58.

[ZCA+06] W. Zhao, Y. Chen, M. Ammar, M. D. Corner, B. N. Levine, and E. Zegura. Capacity Enhancement using Throwboxes in DTNs. In *Proc. IEEE Intl Conf on Mobile Ad hoc and Sensor Systems (MASS)*, 2006. Cited on page 136.

[Zet09] K. Zetter. Weak Password Brings "Happiness" to Twitter Hacker. Threat Level. Available online from `http://www.wired.com/threatlevel/2009/01/professed-twitt/`., January 2009. Last accessed January 2011. Cited on page 31.

[ZGE03] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks. In *2003 IEEE International Workshop on Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE*, pages 139–148, 2003. Cited on page 139.

[ZLLY05] X. Zhang, J. Liu, B. Li, and T. Yum. CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming. In *proceedings of IEEE Infocom*, volume 3, pages 13–17. Citeseer, 2005. Cited on page 72.

[ZNKT07] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Comput. Netw.*, 51(10):2867–2891, 2007. Cited on pages 117 and 137.

[ZPL01] Y. Zhou, J. Philbin, and K. Li. The multi-queue replacement algorithm for second level buffer caches. In *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, 2001. Cited on page 65.

[ZSGK09] M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of YouTube network traffic at a campus network-Measurements, models, and implications. *Computer Networks*, 53(4):501–514, 2009. Cited on page 80.