SiMPLE: Survivability in Multi-Path Link Embedding

-Md Mashrur Alam Khan, **Nashid Shahriar**, Reaz Ahmed, Raouf Boutaba

David R. Cheriton School of Computer Science

University of Waterloo

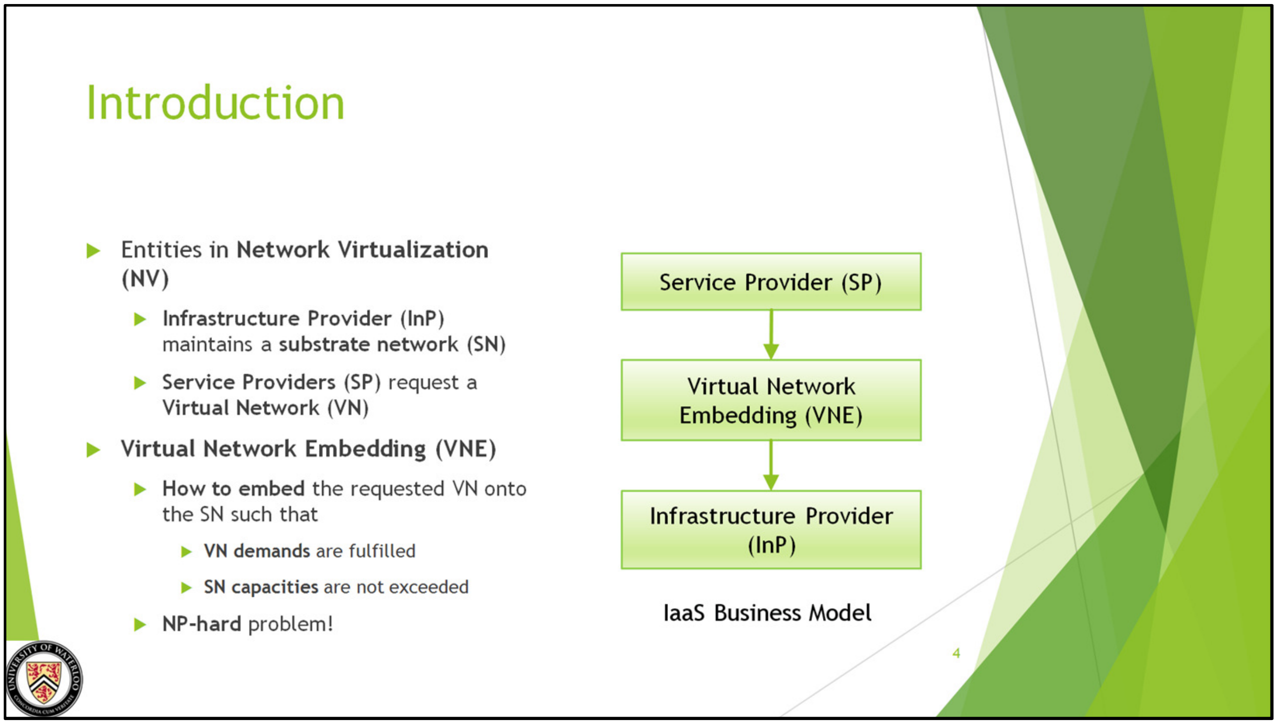
This slide presents the brief outline of this presentation. At first, I will present the introduction and motivation, followed by the state of the art. Afterwards, I will present the main contribution of this paper, which we call SiMPLE. Then, I will present the evaluation results, and finally conclude this presentation.

Outline

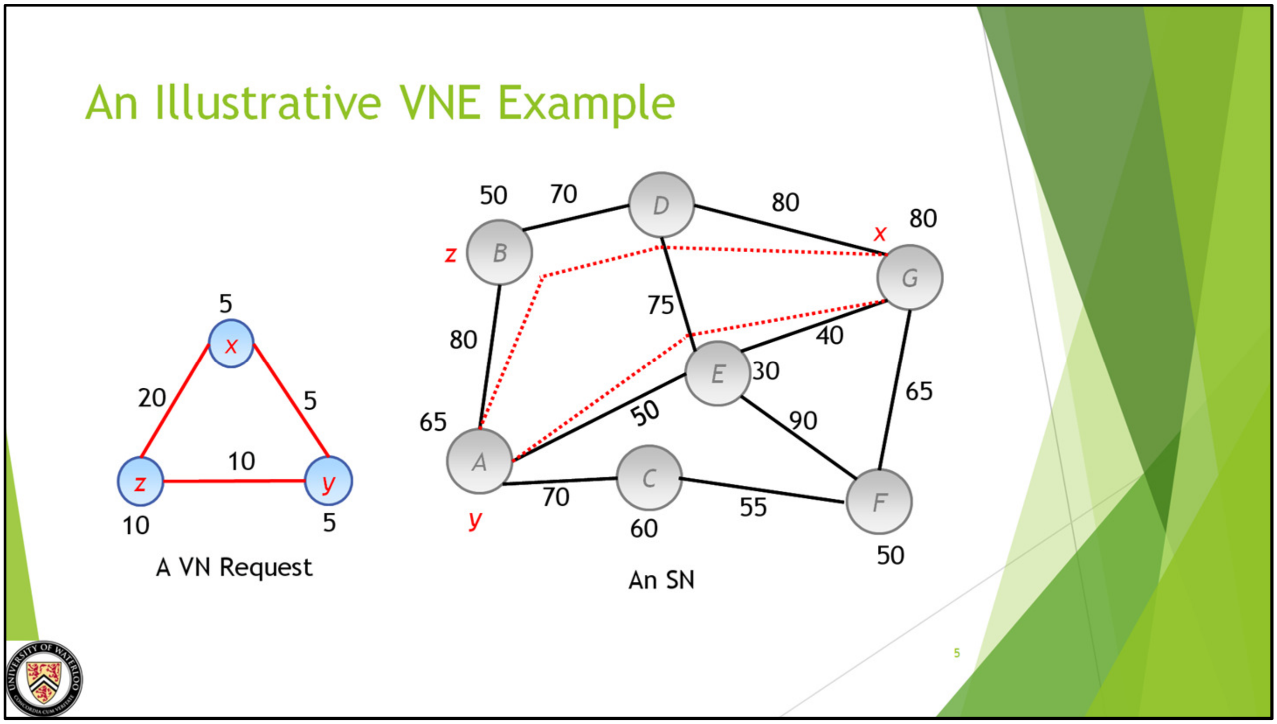
- **Introduction**
- Motivation
- State of the Art
- SiMPLE
- Evaluation
- Conclusion

3

Now I will present the introduction.



"Network Virtualization" is widely regarded as a key enabler for the future Internet, and it has the Infrastructure as a Service (IaaS) model at the core. As shown in the figure, the IaaS business model decouples the role of the ISPs in to Infrastructure Providers (InPs) and Service Providers (SPs). An InP owns and maintains a substrate network (SN), and an SP offers services to its clients through virtual networks (VNs). Virtual Network Embedding (VNE) deals with a feasible embedding of the VNs on to the SNs, subject to the VN demand and SN capacity constraints. It is an NP hard problem as shown in the literature.



In this example, for simplicity, we show how we can embed a small VN onto a small SN. We consider a VN of two nodes and one link, and an SN of seven nodes and ten links. Each substrate and virtual component is labeled with its capacity and demand, respectively. As we see, virtual nodes x and y are embedded to substrate nodes G and A, respectively. The virtual link xy has been embedded on to two link-disjoint paths – the red path denote primary, and the blue path denote backup. The rationale to selecting two paths is that if any link in the primary path fails, then the backup path can support the required demand of the virtual link.

Outline

- Introduction
- **Motivation**
- State of the Art
- SiMPLE
- Evaluation
- Conclusion

6

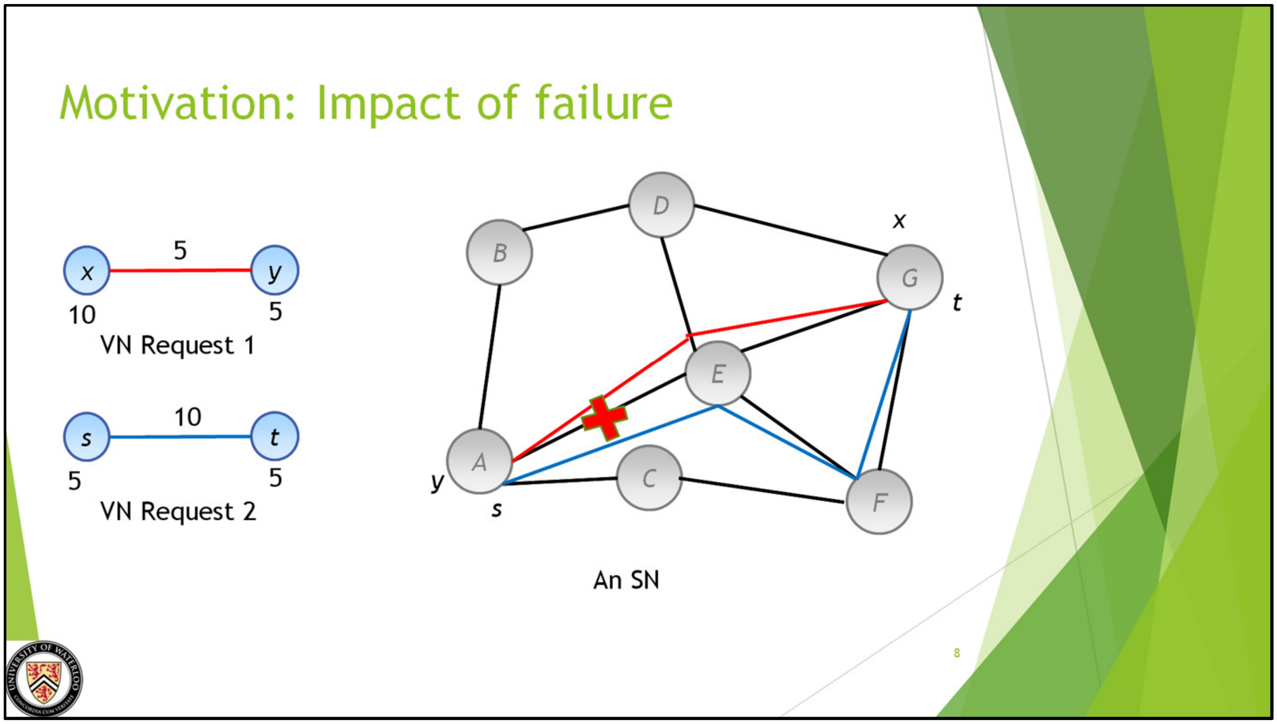
Now we talk about the motivation of this paper.

Motivation

- **Some unfortunate facts**
  - A survey of 200 companies across North America and Europe conducted by CA Technologies finds that IT outages are frequent and lengthy[1]
  - Online businesses in North America lost more than **$26.5 billion in revenue** due to service downtime in 2010[1]
  - Every hour of downtime can typically cost an organization **$300,000** per hour[2]
- **Survivable Virtual Network Embedding (SVNE)**
  - Solve VNE such that it **survives Substrate node and/or link failures**
  - Require **redundant SN resources**

1. Source: InformationWeek, “IT Downtime Costs” May 24, 2011. http://www.informationweek.com/it-downtime-costs-$265-billion-in-lost-revenue/d/d-id/1097919
2. Source: Gartner, “The Cost of Downtime” July 16, 2014. http://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/

A number of works studied the characteristics of link failures in both data center and ISP networks. To summarize, we can classify link failures into single and multiple failures. More than half of the link failures are single link failures, i.e., no other link failure is present at that time in the SN. Provisioning guaranteed VN survivability in these cases can be challenging, since it requires to balance a trade-off between the level of survivability and the amount of used resources. The multiple failure scenario is less frequent than single failure scenario, since it involves a failure with high MTTR, or router/switch failures. However, these failures can jeopardize the embedded VNs, and can cause Service License Agreements (SLA) violation. In addition, bandwidth is considered an expensive resource, and minimizing bandwidth consumption decreases the embedding cost significantly.



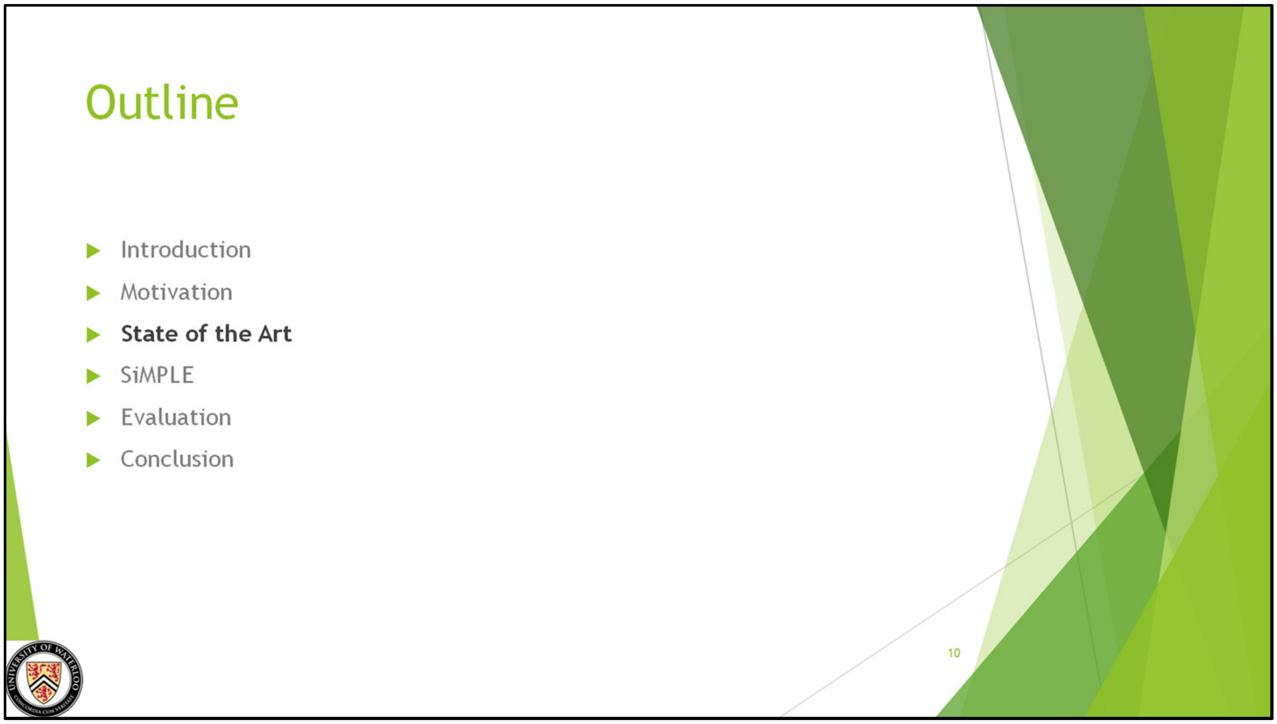
In this example, for simplicity, we show how we can embed a small VN onto a small SN such that it survives a single substrate link failure. We consider a VN of two nodes and one link, and an SN of seven nodes and ten links. Each substrate and virtual component is labeled with its capacity and demand, respectively. As we see, virtual nodes x and y are embedded to substrate nodes G and A, respectively. The virtual link xy has been embedded on to two link-disjoint paths – the red path denote primary, and the blue path denote backup. The rationale to selecting two paths is that if any link in the primary path fails, then the backup path can support the required demand of the virtual link.

Motivation

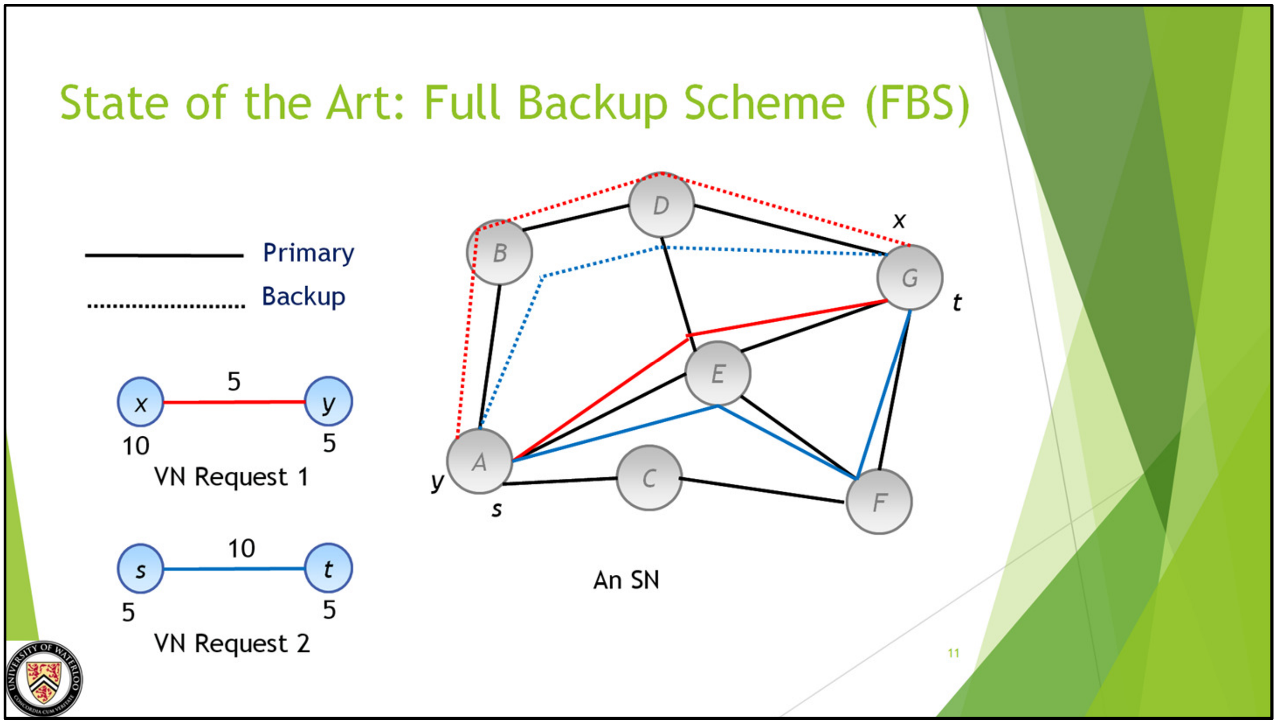
- **Single Link Failures**
  - **More than half** of the total failures are **single** link failures[1, 2]
- **Multiple Link Failures**
  - **Less frequent than single failure**
  - **Mean Time to Repair** (MTTR) for some failed links can be **hours** to **several days**[1, 2]
    - A **new failure** in this period can generate an instance of **multiple failures**
- **Trade off** between **survivability** and **redundancy**
- How to achieve **Maximal survivability** using **minimal redundancy?**
  - Reduces downtime without incurring **excessive cost**

1. Markopoulou, A et al. "Characterization of Failures in an Operational IP Backbone Network," *Networking, IEEE/ACM Transactions on* , vol.16, no.4, pp.749,762, Aug. 2008
2. Phillipa Gill et al. Understanding network failures in data centers: measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2011 conference* (SIGCOMM '11)

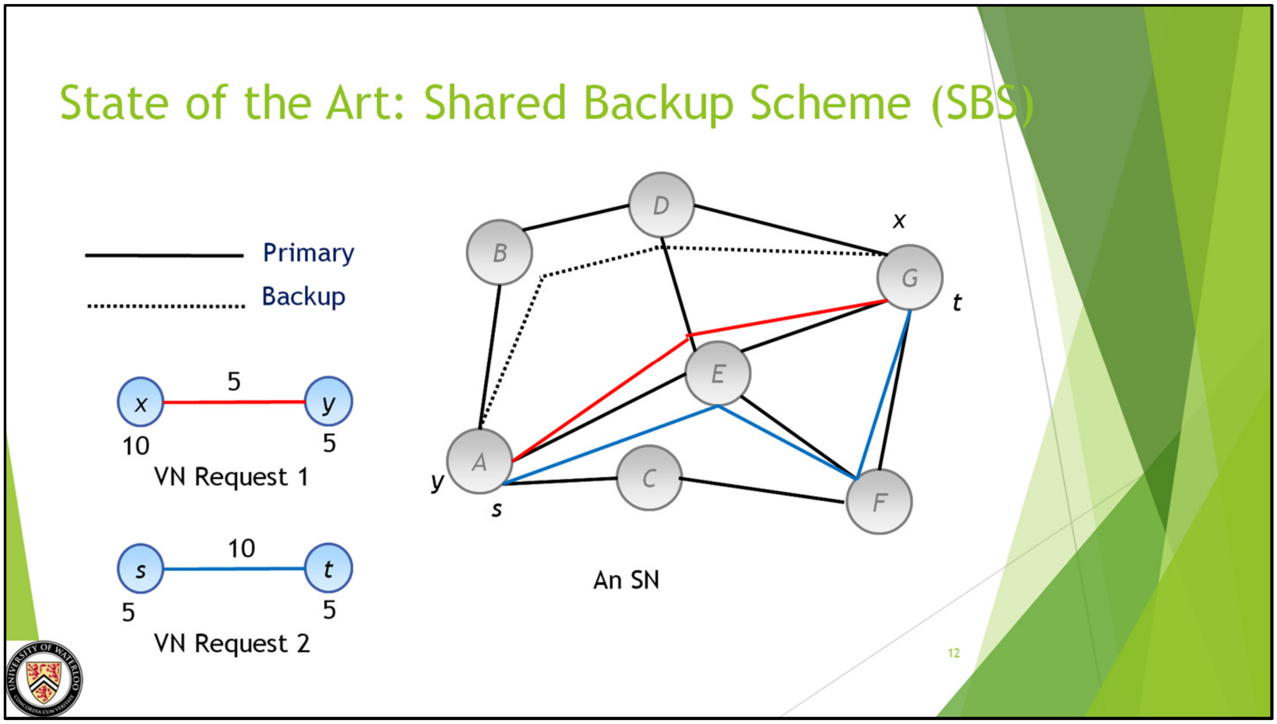
A number of works studied the characteristics of link failures in both data center and ISP networks. To summarize, we can classify link failures into single and multiple failures. More than half of the link failures are single link failures, i.e., no other link failure is present at that time in the SN. Provisioning guaranteed VN survivability in these cases can be challenging, since it requires to balance a trade-off between the level of survivability and the amount of used resources. The multiple failure scenario is less frequent than single failure scenario, since it involves a failure with high MTTR. However, these failures can jeopardize the embedded VNs, and can cause Service License Agreements (SLA) violation. In addition, bandwidth is considered an expensive resource, and minimizing bandwidth consumption decreases the embedding cost significantly.



Now we present the state of the art.



In this example, for simplicity, we show how we can embed a small VN onto a small SN such that it survives a single substrate link failure. We consider a VN of two nodes and one link, and an SN of seven nodes and ten links. Each substrate and virtual component is labeled with its capacity and demand, respectively. As we see, virtual nodes x and y are embedded to substrate nodes G and A, respectively. The virtual link xy has been embedded on to two link-disjoint paths – the red path denote primary, and the blue path denote backup. The rationale to selecting two paths is that if any link in the primary path fails, then the backup path can support the required demand of the virtual link.



In this example, for simplicity, we show how we can embed a small VN onto a small SN such that it survives a single substrate link failure. We consider a VN of two nodes and one link, and an SN of seven nodes and ten links. Each substrate and virtual component is labeled with its capacity and demand, respectively. As we see, virtual nodes x and y are embedded to substrate nodes G and A, respectively. The virtual link xy has been embedded on to two link-disjoint paths – the red path denote primary, and the blue path denote backup. The rationale to selecting two paths is that if any link in the primary path fails, then the backup path can support the required demand of the virtual link.

State of the Art

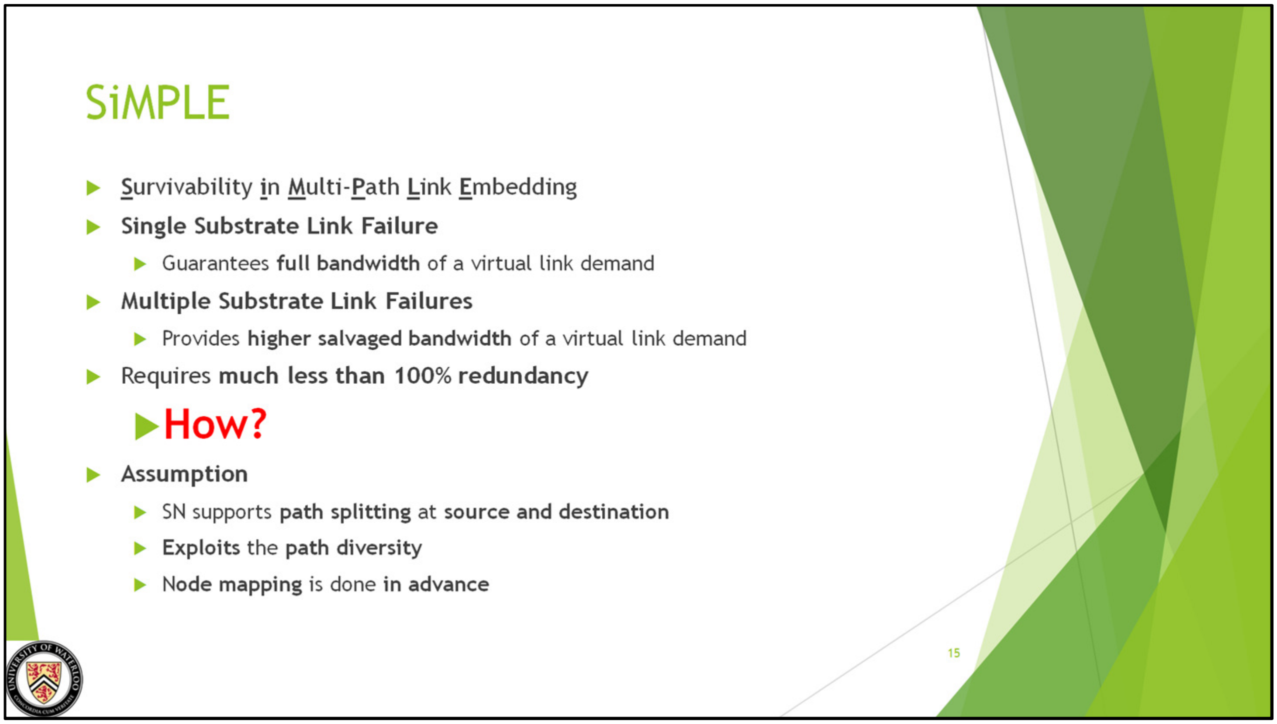
VNE	Path Splitting	Shared Backup	Guaranteed Backup	Method	Embedding
FBS	No	No	Yes	Proactive	Shortest Path
[1]	Yes	No	No	Hybrid	MCF
SBS [2]	No	Yes	No	Proactive	Shortest Path
[3]	Yes	No	No	N/A	MCF
[4]	Yes	No	No	Proactive & recovery	ILP
[5]	Yes	No	No	Proactive & recovery	Simulated Annealing

[1] **M. R. Rahman** et al, Survivable Virtual Network Embedding, **NETWORKING 2010, TNSM 2013**
[2] **T. Guo** et al, Shared Backup Network Provision for Virtual Network Embedding, **ICC 2011**
[3] **M. Yuy** et al, Rethinking VNE: Substrate Support for Path Splitting and Migration, **Sigcomm CCR 2008**
[4] R. R. Oliveira et al, Dos-resilient virtual networks through multipath embedding and opportunistic recovery, SAC 2013
[5] **R. R. Oliviera** et al, No More Backups: Toward Efficient Embedding of Survivable Virtual Networks, **ICC 2013**

We present a number of recent SVNE works in this slide, as presented by the left column of the table. The first VNE proposal, FBS, stands for Full Backup Scheme. FBS provisions two dedicated link disjoint paths for each virtual link, one primary and one backup, as we have seen in our earlier example. Ref [1] finds a backup path for each substrate link, and uses MCF to embed each virtual link. Ref [2] proposes the shared backup scheme (SBS), which finds the disjoint primary and backup paths, and the backup paths are shared among other virtual links. Ref [3] embeds each virtual link by splitting them into a number of paths by Simulated Annealing, and when one path fails, it redistributes the bandwidth among other paths. Ref [4] introduces path splitting in VNE context. Ref [5] presents a variant of SBS. In this work, when the VNE engine cannot embed any other VN, it reconfigures the idle backup resources to improve acceptance ratio. We see from this table that both path splitting and shared backup schemes are adopted by a number of papers. However, except for FBS, no other paper provides provable guarantees to survive a single link failure scenario. Some of the papers adopt proactive manner, that is, provide backups for virtual links before any failure occurs, whereas the others adopt the reactive strategy, that is, recover affected virtual links after failure occurs. The most popular VNE algorithms use MCF or shortest path strategy. Note that in MCF, any node can do the path splitting, which is not necessarily true for the shortest path embedding case.

Outline

- Introduction
- Motivation
- State of the Art
- **SiMPLE**
- Evaluation
- Conclusion



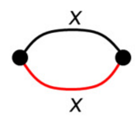
A number of works studied the characteristics of link failures in both data center and ISP networks. To summarize, we can classify link failures into single and multiple failures. More than half of the link failures are single link failures, i.e., no other link failure is present at that time in the SN. Provisioning guaranteed VN survivability in these cases can be challenging, since it requires to balance a trade-off between the level of survivability and the amount of used resources. The multiple failure scenario is less frequent than single failure scenario, since it involves a failure with high MTTR, or router/switch failures. However, these failures can jeopardize the embedded VNs, and can cause Service License Agreements (SLA) violation. In addition, bandwidth is considered an expensive resource, and minimizing bandwidth consumption decreases the embedding cost significantly.

SiMPLE Embedding Concept

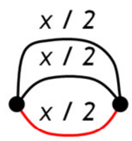
Virtual Link Demand = x

Primary flow

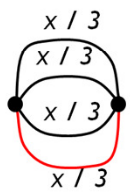
Backup flow



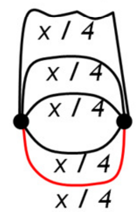
Base case (FBS)
BW Requirement = $x + x$
Backup BW Saving = ***0%***



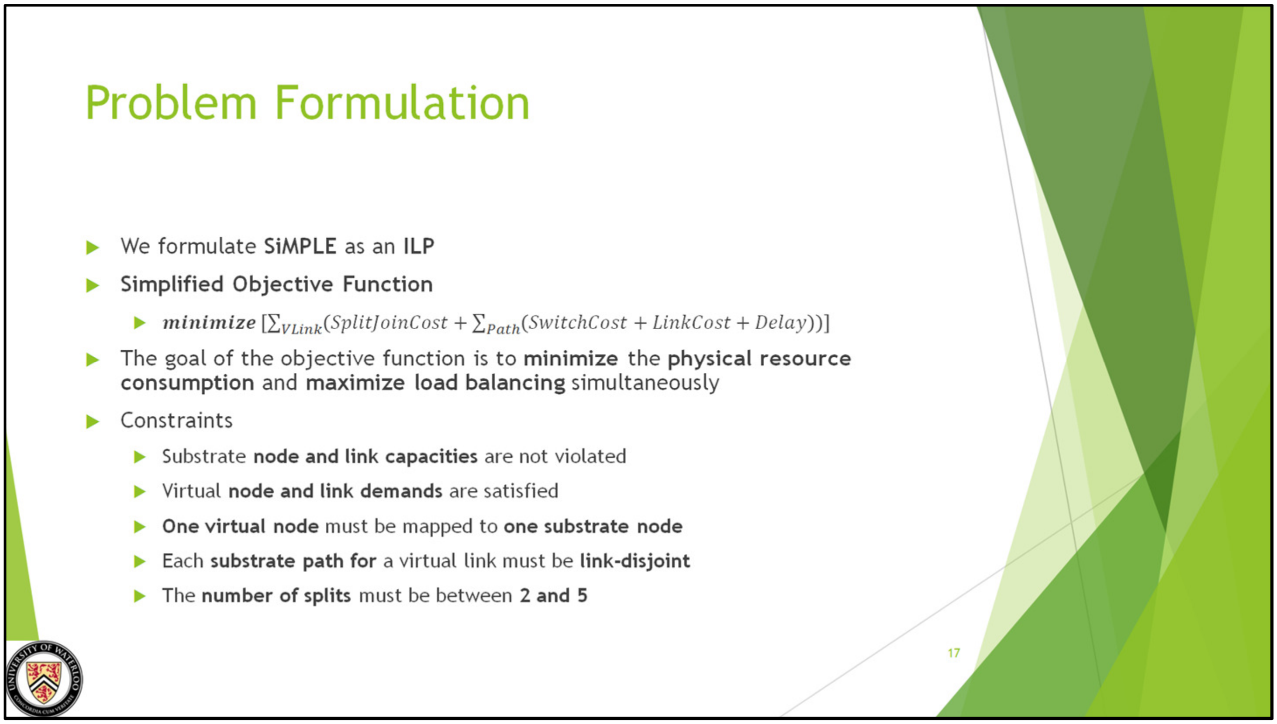
Three splits
BW Requirement = $x + x / 2$
Backup BW Saving = ***50%***



Four splits
BW Requirement = $x + x / 3$
Backup BW Saving = ***67%***



Five splits
BW Requirement = $x + x / 4$
Backup BW Saving = ***75%***



SiMPLE is formulated as an ILP. The objective function is to minimize the embedding cost, which has a number of components. The first component is the split and join cost, which is due to splitting and merging the each data stream at the ingress and egress switches. The second cost is the switching cost, which is due to forwarding the fragmented data stream between the source and destination substrate nodes. The third cost is the substrate link cost, which represents the amount of bandwidth used for embedding. The fourth cost is the accumulated delays along the substrate paths. This objective function minimizes the physical resource consumption and maximizes load balancing simultaneously. The constraints for the ILP formulation includes the SN capacity constraints, VN demand constraints, virtual node unsplittability constraint, link disjointness constraint, and finite number of splits per virtual link constraint.

Embedding Heuristics

SiMPLE-GR
Greedy algorithm for SiMPLE Link Embedding
Computes the first *k* disjoint shortest paths for *k = 2, 3, 4, 5* by using Dijkstra's shortest path algorithm with our cost function
Returns the embedding yielding the lowest cost

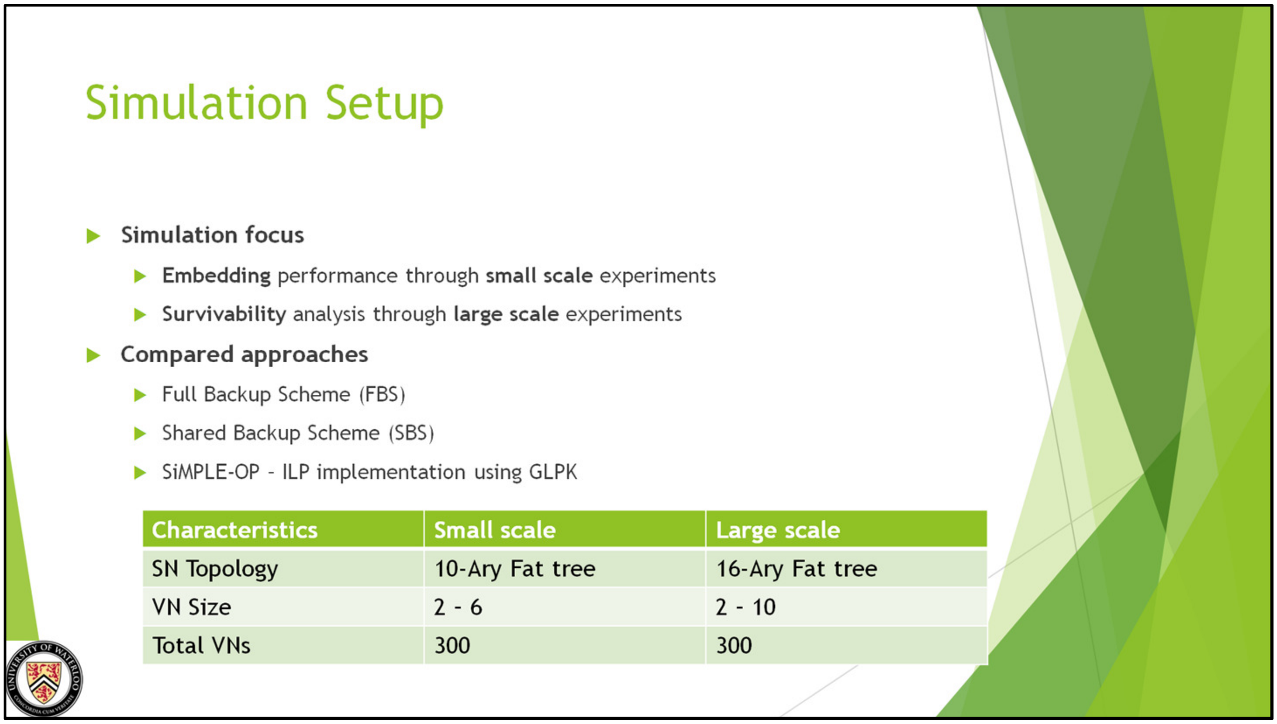
We also propose two greedy algorithms for embedding – each representing one stage in SiMPLE embedding concept. SiMPLE-PR stands for SiMPLE proactive allocation, and it computes the first k shortest paths for k = 2, 3, 4, 5, and returns the embedding with lowest cost. SiMPLE-RE, on the other hand, stands for SiMPLE reactive recovery, and it recovers each virtual link affected by physical failures. For recovering purpose, it considers three options as described earlier – provisioning new path, fixed allocation, and variable allocation, and returns the embedding with lowest cost.

Outline

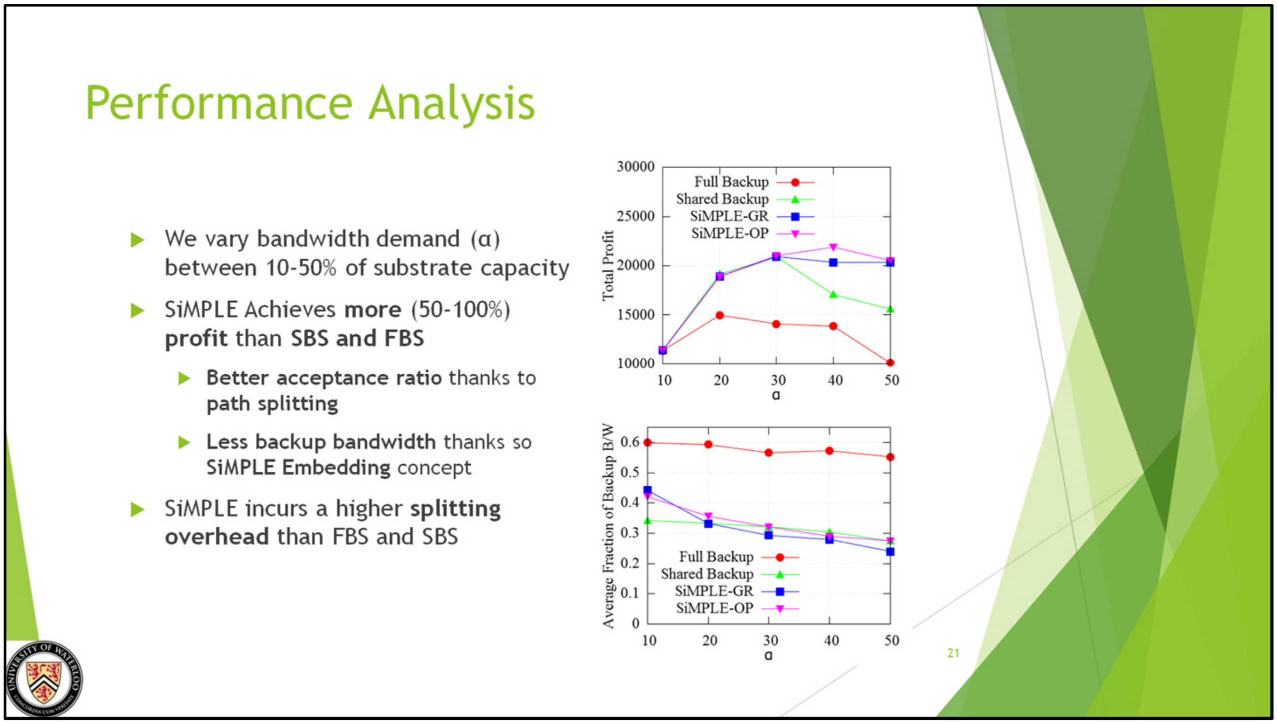
- Introduction
- Motivation
- State of the Art
- SiMPLE
- **Evaluation**
- Conclusion

19

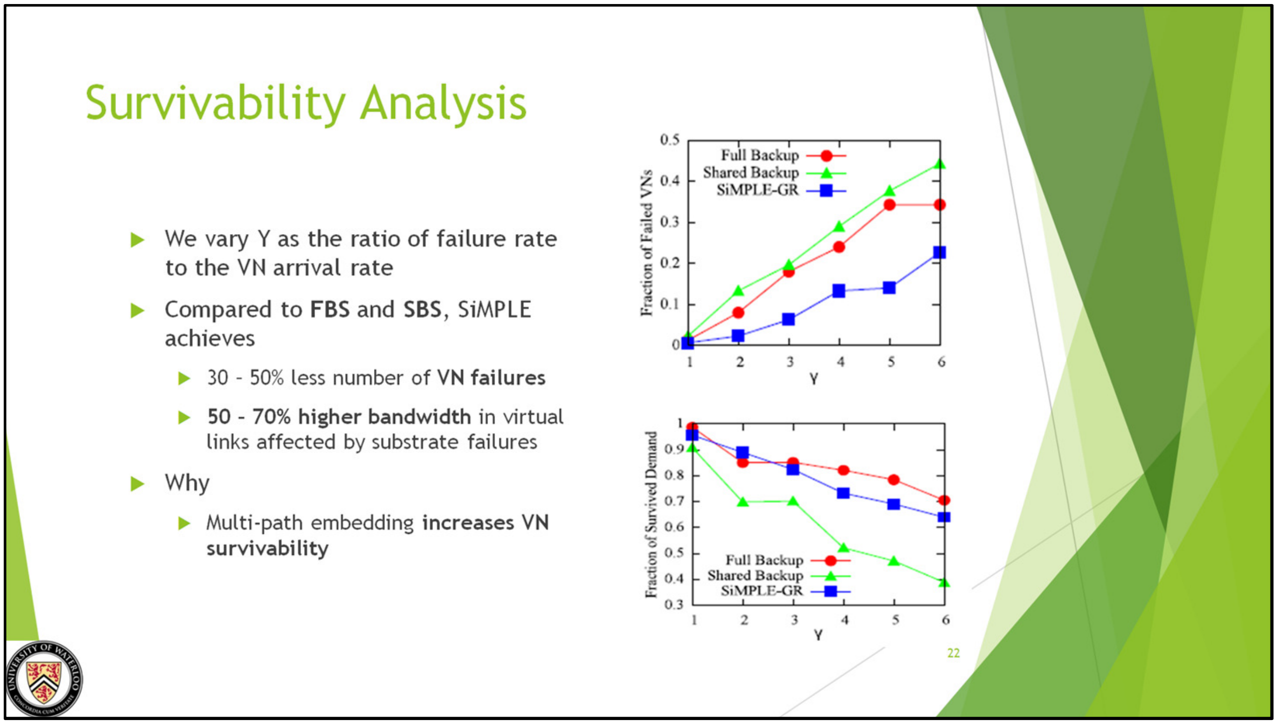
Now we present the evaluation results.



We ran simulations on both data center networks and ISP networks, as represented by fat tree and synthetically generated topologies, respectively. To demonstrate the scalability of SiMPLE, we ran small scale simulations on embedding performance, and large scale simulations on VN survivability. We compared SiMPLE with two existing approaches – FBS and SBS. We also implemented the optimal solution for SiMPLE, SiMPLE-OP, by GNU Linear Programming Toolkit (GLPK). The table mentioned in this slide represents the different parameters of these simulations. We also compared with SiMPLE-OP only for small scale simulations, since it is not possible to compute the optimal solution for the large scale scenario.



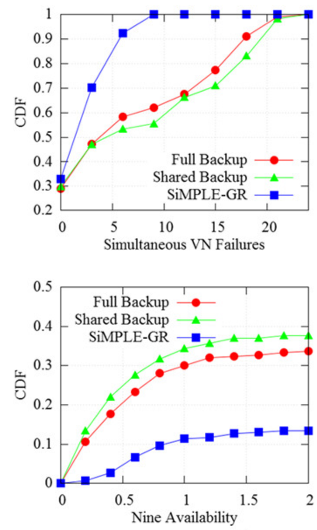
In this slide, we demonstrate the performance of SiMPLE-PR with respect to FBS, SBS, and SiMPLE-OP, as indicated by the legends in the graphs. These experiments were run for different alpha, which is defined as the percentage of an VN demand compared to the SN capacity. In the first graph, we show profit, which is defined as the VN lifetime multiplied by VN demand. Simulation results show that profit generated by all four approaches are same for small alpha, but as alpha increases, the profit falls for FBS and SBS, and SiMPLE achieves 50 – 100% higher profit. In the second graph, we show the percentage of backup bandwidth used for backup for different alpha. Here, SiMPLE consumes 40 – 50% less backup than FBS and performs almost identical to SBS. Since SBS uses the same backup for different VNs, the average backup bandwidth is very small, but unlike SiMPLE, it provides no guarantee for substrate failures. We also run simulations to demonstrate that SiMPLE achieves a higher acceptance ratio. SiMPLE's working principle – path splitting, contributes to these results. However, unlike FBS and SBS, SiMPLE incurs a higher path splitting overhead. With the built in path splitting capacity, the modern switches are expected to mitigate this overhead.



In the survivability experiments, we vary gamma – the ratio of failure arrival rate to VN arrival rate. Both of these graphs are plotted for gamma = 5. In the first graph, we show the CDF of simultaneous VN failures, and it is clear that SiMPLE-PR incurs only 10 simultaneous failures compared to 20 in FBS and SBS. The second graph demonstrates the CDF of nine availability – the number of nines in the uptime probability of a VN. This CDF shows that less VN has less nine availability in SiMPLE, e.g., the number of VNs with 0.5 nines (68% availability) in FBS and SBS is roughly four times than that in SiMPLE-PR. Other simulations show that SiMPLE incurs less number of total VN failures, and a higher bandwidth in virtual links affected by physical failures. The justification of these results is again the working principle of SiMPLE – path splitting – which increases VN survivability.

Survivability Analysis

- Compared to **FBS** and **SBS**, SiMPLE
  - Increases **VN survivability** in presence of multiple failures
  - Achieves higher **VN availability** (represented by nines)
- Why
  - Thanks to **multi-path embedding** and **SiMPLE Embedding** concept



23

In the survivability experiments, we vary gamma – the ratio of failure arrival rate to VN arrival rate. Both of these graphs are plotted for gamma = 5. In the first graph, we show the CDF of simultaneous VN failures, and it is clear that SiMPLE-PR incurs only 10 simultaneous failures compared to 20 in FBS and SBS. The second graph demonstrates the CDF of nine availability – the number of nines in the uptime probability of a VN. This CDF shows that less VN has less nine availability in SiMPLE, e.g., the number of VNs with 0.5 nines (68% availability) in FBS and SBS is roughly four times than that in SiMPLE-PR. Other simulations show that SiMPLE incurs less number of total VN failures, and a higher bandwidth in virtual links affected by physical failures. The justification of these results is again the working principle of SiMPLE – path splitting – which increases VN survivability.

Outline

- Introduction
- Motivation
- State of the Art
- SiMPLE
- Evaluation
- **Conclusion**

24

Now we present the conclusion and future works.

Conclusion

- In this paper, we have proposed
  - A **key concept** that facilitates bandwidth saving, and
    - **Guarantees VN survivability** for a single substrate link failure
    - **Maximizes VN survivability** on multiple substrate link failures
  - An optimization model that
    - Balances the **trade-off** between maximizing **VN survivability**, minimizing **redundant resources** and **splitting overheads**
  - A **greedy algorithm** to implement SiMPLE embedding concept
- Simulation results show that **SiMPLE** outperforms **FBS** and **SBS** on both performance metrics and survivability metrics

Future Works

- We would like to extend this work to
  - **Prototype Implementation** in SDN platform
  - **Coordinated Node** and **Link Mapping** while embedding
  - **Nested NV** environment

26

Software Defined Networking (SDN) is a research area that decouples the data plane from the control plane, and facilitates easier network management. SDN can be used to implement a prototype for SiMPLE.
Currently, SiMPLE assumes that the node mapping has been done. We can coordinate node mapping with link mapping, and obtain a potentially better VN embedding result.
Multiple layers of VNs can be embedded on the same SN, where the first layer of VNs act as the SN for the second layer of VNs, and so on. This is called nested NV, and this involves cross layer optimization. We can extend SiMPLE in these environments.

THANK YOU

Questions?