Chapter 5: MULTI-HOP WIRELESS NETWORKS

Editors: Torsten Braun[1], Andreas Kassler[2], Maria Kihl[3], Veselin Rakocevic[4], Vasilios Siris[5], Geert Heijenk[6]

[1] University of Bern, Switzerland

[2] Karlstad University, Sweden

[3] Lund University, Sweden

[4] City University London, United Kingdom

[5] FORTH-ICS, Greece

[6] University of Twente, The Netherlands

Contributors: Hans van den Berg, Frank Roijers, Marcel Castro, Carles Gomez, Josep Paradells, Thomas Staub, Ruy de Oliveira, Jonas Karlsson, Stefano Avallone, Philipp Hurni, Barbara Staehle, Dirk Staehle, Patrick Goering, Fei Liu, Robbert Haarman, Peter Dely

5.1 Introduction

In cellular and wireless local area networks, wireless communication only occurs on the last link between a base station and the wireless end system. In multi-hop wireless networks there are one or more intermediate nodes along the path that receive and forward packets via wireless links. Multi-hop wireless networks have several benefits: Compared to networks with single wireless links, multi-hop wireless networks can extend the coverage of a network and improve connectivity. Moreover, transmission over multiple “short” links might require less transmission power and energy than over “long” links. Moreover, they enable higher data rates resulting in higher throughput and more efficient use of the wireless medium. Multi-hop wireless networks avoid wide deployment of cables and can be deployed in a cost-efficient way. In case of dense multi-hop networks several paths might become available that can be used to increase robustness of the network.

Unfortunately, protocols developed for fixed or cellular networks as well as the Internet are not optimal for multi-hop wireless networks. This is in particular the case for routing protocols, where completely new unicast, multicast, and broadcast routing protocols have been developed for (mobile) ad-hoc and sensor networks.

On the transport layer, the *Transmission Control Protocol (TCP)* is the de facto standard in the Internet and in order to allow interoperability, TCP must be supported in multi-hop wireless networks as well. However, many protocol mechanisms such as congestion control and error control based on acknowledgements do not work efficiently in multi-hop wireless networks due to various reasons such as contention and control packet overhead. Even on application level new concepts are required to support discovery of available applications and services.

Several concrete application scenarios for multi-hop wireless networks have been investigated during the last years. Initially, it has been proposed to deploy multi-hop networks to extend the coverage of cellular networks by relaying packets. Recently, wireless mesh networks have been proposed to provide broadband Internet services without the need of expensive cable infrastructures, in particular in areas sparsely populated. Wireless mesh networks consist of mesh routers and mesh clients, where mesh routers have minimal mobility and form the backbone of wireless mesh networks [Aky05]. They make use of heterogeneous network technology such as IEEE 802.11, 802.16, and cellular radio networks. Relaying nodes can also be mobile such as in case of vehicles. In that case the term mobile ad-hoc network is more appropriate. Vehicular networks as a special case of mobile ad-hoc networks make use of the frequently existing communication equipment in cars (either pre-installed or enabled by equipment carried by passengers). Wireless sensor networks are another emerging technology, can cover large geographical areas, and provide connectivity without having direct physical access to each sensor node. Sensor nodes can be configured and sensor data can be read using multi-hop networking.

The following sections discuss the contributions from COST Action 290 in research areas discussed above. Section 5.2 investigates the performance of forwarding and relaying in multi-hop wireless networks and discusses approaches to optimise wireless resource usage. Section 5.3 investigates routing protocols for unicast, multicast, and broadcast communication in multi-hop wireless networks, while novel mechanisms for transport protocols, in particular to support TCP, are presented in Section 5.4. Sections 5.5 and 5.6 are related to two promising application scenarios, namely wireless mesh and sensor networks. The issue of efficient self-management, e.g., to configure frequencies to be used automatically is in the focus of Section 5.5. The key issue in wireless multi-hop networks is how to operate these in an energy-efficient way. Cross-layer design approaches as well as appropriate models to evaluate such mechanism are discussed in Section 5.6. Finally, Section 5.7 presents new mechanisms to support efficient service discovery in (mobile) ad-hoc networks.

5.2 Packet Relaying in Multi-Hop Networks

In wireless multi-hop networks, nodes communicate with each other using wireless channels and do not have the need for common infrastructure or centralized control. Nodes may cooperate with each other by forwarding or relaying each others' packets, possibly involving many intermediate relay nodes. This enables nodes that cannot hear each other directly to communicate over intermediate relays without increasing transmission power. Such multi-hop relaying is a very promising solution for increasing throughput and providing coverage for a large physical area. By using several intermediate nodes, the sender can reduce transmission power thus limiting interference effects and enabling spatial reuse of frequency bands.

In ad-hoc networks, the medium is shared and nodes arrange access to the medium in a distributed way independent of their current traffic demand. In particular given standard ad-hoc routing protocols that try to minimize relaying nodes on the path, nodes closer to the network centre are more likely to become a relay node. This has the inherent drawback that a node that serves as a relay node for transmissions of multiple neighbouring nodes is prone to become a performance bottleneck. As it is necessary to understand performance of such relay networks, the next sub section provides an overview on performance analysis of a relay node.

When multiple relays are involved across an end-to-end path, it is important to control overhead for each single packet transmission. Unfortunately, current *Medium Access Control* (MAC) and physical layers for *Wireless Local Area Network* (WLAN) based multi-hop networks impose high overhead for the transmission of small data packets, which is common for *Voice over Internet Protocol* (VoIP). By combining several small packets into larger ones, per packet transmission overhead can be reduced significantly. Therefore, the following subsections provide an overview on efficient packet aggregation mechanisms.

5.2.1 Performance Modelling and Analysis of a Relay Node in IEEE 802.11 Wireless Ad-Hoc Networks

Performance studies on multi-hop ad-hoc networks are mostly based on simulations. Analytical studies are rare and mostly focus on packet-level effects, i.e., packet loss and delays, for details see Section 1 of [Ber06, TD(06)003]. This subsection, based on [Ber06, TD(06)003] and [Roi07, TD(07)016], presents an analytical study investigating *flow-level* metrics, in particular end-to-end transfer times of flows sharing a common relay node.

In [Ber06, TD(06)003] a simple, two-hop network consisting of a central node used as relay by a varying number of source nodes is analyzed via an idealized fluid-flow queuing model. Assuming equal sharing of the underlying radio transmission resources among source nodes and relay node, a closed-form expression is obtained for the transfer time of a flow from source to destination via the central relay node. In [Roi07, TD(07)016] the fluid model is extended to the case where the relay node may obtain a different (higher) share of the capacity than the source nodes. This so-called “unequal resource-sharing” yields considerably shorter end-to-end flow transfer times. Unequal resource-sharing can be achieved in practical situations, e.g., by deploying the QoS differentiation capabilities of the IEEE 802.11e MAC protocol. In [Roi07, TD(07)016] it is shown how to map the IEEE 802.11e parameters on the parameters of the extended model. The modelling approach and parameter mapping is validated by extensive system simulations. Below, we will describe the set-up and results of the studies in [Ber06, TD(06)003] and [Roi07, TD(07)016] in some more detail.

5.2.1.1 Ad-hoc Network Scenario

We consider a two-hop network consisting of a number of source nodes that initiate flow transfers at random time instants, and a single relay node that forwards the traffic generated by the sources to the next-hop destination nodes, cf. Fig. 5.1. The source and destination nodes that are within each other’s sensing range are all within the transmission range of the relay node. Hence, there are no hidden nodes.

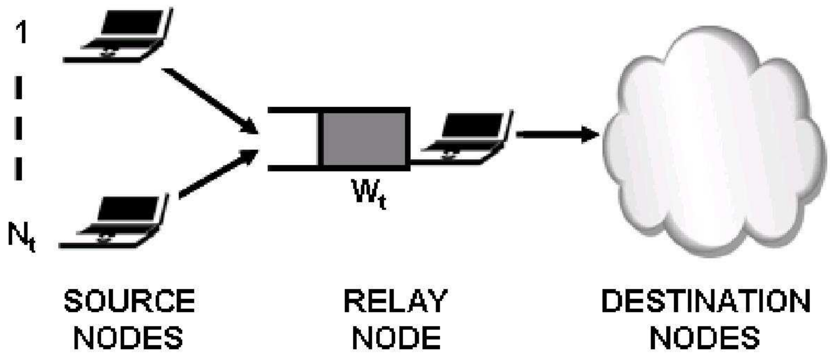

Fig. 5.1. Ad-hoc network scenario.

5.2.1.2 Fluid Model Description

We assume a large number of source nodes, which become active and initiate flow transfers to destinations via the relay node according to a Poisson process with flow arrival rate λ. The relay node relays all traffic of the source nodes in a first-come-first-serve discipline. Active source nodes and the relay node share the system capacity, which depends on the number of active source nodes n and is denoted by C_n. Once a source node has completed a flow transmission, the source node becomes inactive (although the last part of the flow may still be at the buffer of the relay node waiting for service). Flow sizes (in terms of the amount of traffic/fluid) are random variables (denoted by F) with finite mean f and second moment f_2. A source node has at most one flow transfer in progress.

First, we consider the case of so-called "equal resource-sharing". If n source nodes have a flow transfer in progress, any source node transmits its traffic (fluid) into the buffer of the relay node at rate $C_n/(n+1)$, while a rate $C_n/(n+1)$ is used by the relay node to serve the buffer (i.e., to forward the traffic stored in its buffer to the next node). The amount of work backlogged in the buffer is denoted by W_{buffer}. In case $W_{buffer} > 0$ and $n = 0$ the relay node receives the entire capacity C_0.

In case of unequal resource-sharing, the maximum ratio between the share of the relay node and a source node is denoted by $m_n \in R$, and the relay node may obtain capacity $m_nC_n/(n+m_n)$. The relay node will only obtain the maximum share, if it can actually use it, i.e., if the input rate exceeds the output rate ($n \geq m_n$) or if $W_{buffer} > 0$. Otherwise, the input and output rates are coupled, resulting in capacity share of $C_n/2$ for the relay node. The source nodes always share the remaining capacity equally. The main performance measures of interest are the steady-state buffer workload W_{buffer} at the relay node and the overall flow transfer time $D_{overall}$, i.e., the time required to completely transfer a flow from source to destination.

5.2.1.3 Analysis of Fluid Model with Equal Resource Sharing

In [Ber06, TD(06)003] insightful, explicit formulas for the mean values of the performance measures are presented. The analysis focuses on the case of equal resource sharing with constant capacity, i.e., C_n is constant for all n (cf. Section 3.1 of [Roi07, TD(07)016]), for simplicity denoted by C, which allows us to define the load of the system by $\rho = \lambda f/C$. The overall flow transfer time $D_{overall}$ of a flow is the sum of its flow transfer time D_{source} and the buffer delay of its last particle D_{buffer}. Hence,

$$D_{overall} = D_{source} + D^*_{buffer}\,. \tag{5.1}$$

Notice that D_{source} and D^*_{buffer} are not statistically independent. The behaviour of the source nodes is described by a generalized processor sharing queuing model [Coh79] for which the stationary distribution, here denoted by π_n, is known. Little's law on the mean number of active source nodes yields

$$\mathrm{E}D_{source} = \frac{\mathrm{E}N}{\lambda} = 2\frac{f/c}{1-\rho}, \tag{5.2}$$

which is insensitive to the flow-size distribution apart from its mean. The buffer delay D_{buffer} is derived from the buffer workload W^{*}_{buffer} seen by the last particle, which is the sum of the workload W_{buffer} upon flow arrival and the buffer increase ΔW_{buffer} during D_{source}. Explicit expression for W_{buffer} and ΔW_{buffer} can be derived by relating the total amount of work in the total system to that in a corresponding M/G/1-queue. Then, we obtain the following expression for the amount of work W^{*}_{buffer} that a last particle will find upon arrival at the relay node

$$\mathrm{E}W^{*}_{buffer} = \mathrm{E}W_{buffer} + \mathrm{E}\Delta W_{buffer} = \frac{2\rho^2\, f_2/fc}{(1-2\rho)(1-\rho)} + \frac{2\, f\rho/c}{1-\rho}. \tag{5.3}$$

Observe that the buffer delay of the last particle D^{*}_{buffer} is the time required to serve the amount of work W_{buffer}. As the resource sharing between source nodes and relay node is purely processor sharing, we approximate the buffer delay of the last particle by

$$\mathrm{E}D^{*}_{buffer} \approx \sum_{n=0}^{\infty} \pi_n \mathrm{E}X_n\left(\mathrm{E}W^{*}_{buffer}\right), \tag{5.4}$$

where $\mathrm{E}X_n(\tau)$ is the so-called response time for a job of size τ in an M/M/1-PS queue [Cof70]. For further details about the approximation we refer to [Ber06, TD(06)003]. Observe that we have an expression for $\mathrm{E}\,D_{overall}$ as we have derived expressions for the means of both parts of (5.1).

5.2.1.4 Numerical Results

The model and the analysis have been extensively validated. Fig. 5.2 presents a validation of the overall flow transfer time consisting of a comparison of i) detailed simulations of the ad-hoc network scenario described above, ii) simulation of the fluid-flow model, and iii) the analytical results. The results illustrate that the bottleneck model captures the behaviour of the ad-hoc network scenario including the influence of the load and the flow-size distribution. Further, the analysis is also very accurate.

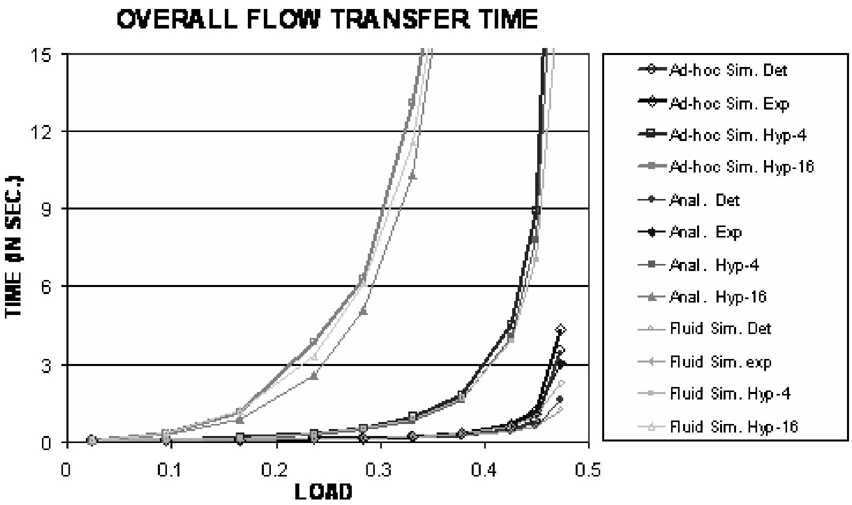

Fig. 5.2. Overall flow transfer time in equal resource-sharing bottleneck model.

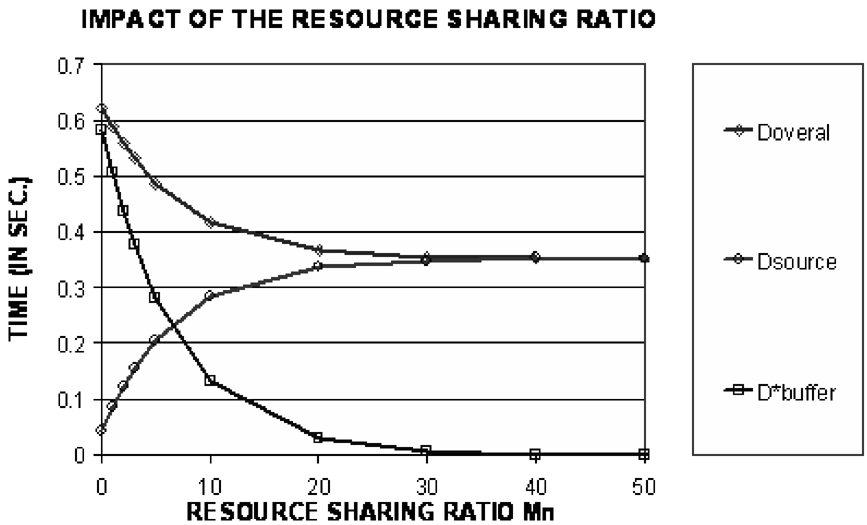

Fig. 5.3. Impact of the resource sharing ratio m for load 0.43.

Fig. 5.3 illustrates the impact of the resource sharing ratio m_n; it illustrates the trade-off between D_{source} and D_{buffer} for a given load (here chosen as 0.43). When m_n increases, it becomes less probable that $W_{buffer} > 0$, and the relay node will mostly obtain a share of C/2. Hence, there is hardly any queuing at the relay node. From the right graph we conclude that resource sharing ratio $m_n = \infty$, i.e., always granting a share of C/2 to the relay node, is optimal for the overall flow transfer time. For the mapping of m_n on the IEEE 802.11e parameter setting, and its validation by detailed system simulations, we refer to [Roi07, TD(07)016].

5.2.2 Packet Aggregation for VoIP in Wireless Meshed Networks

The provision of VoIP in wireless mesh networks is an important service for the future wireless internet. However, the transmission of small (voice) packets imposes high MAC and physcial layer overhead, which leads to low capacity for VoIP over IEEE 802.11-based multi-hop mesh networks. The idea of packet aggregation is to combine several small packets into a larger aggregated one so that overhead on the wireless medium can be significantly reduced. While such aggregation mechanisms have been proposed for single-hop infrastructure wireless local area networks, designing an aggregation strategy for multi-hop wireless mesh networks is a hard problem. In infrastructure wireless local area networks, the sender has complete knowledge about the link characteristics of one hop neighbours and can thus calculate an optimal packet size for aggregation [Lin06]. In a multi-hop environment, signal quality and congestion for each link are different. When mesh relay nodes aggregate small packets, there is an inherent trade-off regarding packet size. Aggregating more packets leads to larger aggregated ones, reduces the overall number of packets in the mesh and leads to reduced multi-hop contention and packet loss due to collisions. However, such larger aggregated packets can result in higher packet loss for a link that operates at low signal quality. For such links, aggregating fewer packets can be beneficial.

For efficient packet aggregation it is essential to have enough packets in the local queue to be aggregated. Therefore, packets are artificially delayed to increase the aggregation ratio, which might lead to higher end-to-end delay. On the other hand, aggregation reduces the overall number of packets in a collision domain, decreasing multi-hop contention, collisions, re-transmissions and, therefore, MAC layer utilization, which may reduce the end-to-end delay (cf. Fig. 5.4).

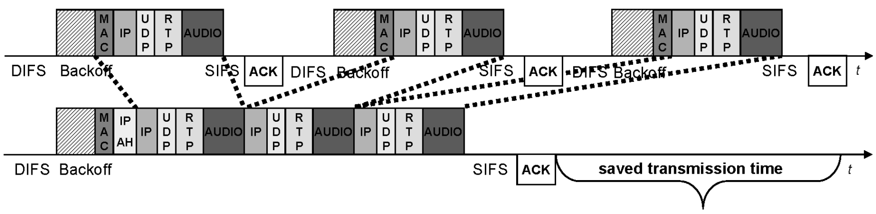

Fig. 5.4. Packet Aggregation saves transmission time and reduces overhead

Packet aggregation can be classified as end-to-end or hop-by-hop. In end-to-end aggregation, all packets towards a common destination are aggregated. In hop-by-hop aggregation, aggregation and de-aggregation is done at every node, which leads to higher complexity and potentially higher delay. However, it yields better aggregation possibilities as packets for different destination addresses but with the same next hop could be aggregated. In a realistic wireless mesh network deployment, link characteristics and load will be different for each hop. Therefore, a hop-by-hop aggregation scheme enables an optimization of the packet size used for aggregation for each hop. This allows to trade-off packet loss due to contention and bit errors. Hop-by-hop aggregation outperforms end-to-end aggregation strategies, because the overall aggregation along a whole path will not be constrained by the weakest link, leading thus to significant performance improvement compared to end-to-end aggregation mechanisms.

5.2.2.1 Link Quality Based Adaptive Packet Aggregation

Finding an optimal aggregation size is difficult to achieve as end-to-end QoS constraints need to be maintained. For example, using G.729a voice codec requires end-to-end delay below 150 ms at less than 2% packet loss in order to provide acceptable quality [TSB06]. Due to the retransmission scheme of the IEEE 802.11 MAC layer, a reduction of the packet loss ratio has also beneficial effects on jitter and delay, so a good aggregation scheme for VoIP must reduce packet loss while keeping end-to-end delay low.

Larger packets have better efficiency, but are more likely to be dropped due to frame errors than small packets for a given *Bit Error Rate* (BER). For a given physical coding scheme and card sensitivity, a bit error probability can be found for a given link *Signal-to-Noise Ratio* (SNR) value [Xiu04]. For a given BER the frame error rate can be approximated as $1\text{-}(1\text{-BER})^n$, where n is the frame length in bits. Therefore, the SNR can be used to predict loss probabilities of frames with different lengths. Although [Agu04] argues that such mapping is hard to obtain, [Sou06, Lal03] show that SNR can significantly improve link quality prediction and hence packet loss estimation. The SNR of a link is a function of signal strength and noise, which might be different at the sender and the receiver.

In the adaptive aggregation mechanism [Kas07, TD(07)020], every node measures the SNR for received packets, stores a moving average for each neighbour and exchanges this information in extensions to Hello messages, which are sent periodically to maintain neighbour link information in routing protocols. When receiving such a message, every node updates its routing table to additionally keep track of the optimal packet size estimate $SIZE_{max}$ used for aggregation for the next hop. In order to control additional delay added by aggregation and maintain end-to-end delay bounds, the algorithm can be controlled by MAX_{delay}, which determines the maximum additional delay that each packet could experience while waiting to be aggregated.

The aggregation algorithm then marks received packets with a timestamp at every hop and stores them in a queue located between the routing module and the

MAC layer. When the MAC layer becomes idle, an aggregation packet is created, which is composed of all potential packets with the same next hop. The cumulative size of those potential packets needs to be larger than $SIZE_{min}$ and smaller than $SIZE_{max}$. If the size is below $SIZE_{min}$, only the packets that are older than MAX_{delay} are aggregated. If none is older, the queue stays idle and nothing is sent. If exactly one packet is older, the queue sends the packet as it is. If at least two are older, they are aggregated and passed to the MAC layer.

5.2.2.2 Implementation and Results Obtained

The proposed adaptive packet aggregation was implemented in ns-2. For comparison, we also implemented a static hop-by-hop aggregation mechanism that adds constant forced delay without considering link quality. With an arrow topology and 2 hops, around 40 flows without aggregation can be supported, while for static aggregation the number of supported flows can be increased to around 60 flows. For adaptive aggregation 120 flows can be supported, leading to an increase of 200% compared to "no aggregation". In a grid topology, an increase of 243% supported flows compared to "no aggregation" has been achieved. The performance increase of the adaptive aggregation algorithm was verified by changing the distance and thus link quality between mesh relay nodes. More detailed results can be obtained from [Kas07, TD(07)020].

5.3 Routing Protocols

The objective of routing is to route data from a sender to one or more destinations. Routing in a mobile wireless multi-hop network, and in particular in mobile ad-hoc networks, is a challenging task. Routing protocols in mobile ad-hoc networks are usually divided into proactive, reactive and hybrid routing. A proactive protocol evaluates routes to all reachable nodes and attempt to maintain consistent up-to-date routing information. In a reactive protocol, routing paths are searched only when needed. Hybrid protocols combine proactive routing with reactive routing in hierarchical network structures.

The mobility of nodes in combination of the noisy links calls for new approaches in order to obtain optimal network performance. Also, new applications and systems require more than the traditional unicast routing protocols. For example, broadcasting and multicasting protocols targeted at mobile wireless networks are needed. In this section, various investigations of routing protocols for mobile wireless multi-hop networks are discussed.

First, real experiments with three of the most popular mobile ad-hoc network routing protocols are described: *Ad-hoc On-demand Distance Vector protocol* (AODV), *Optimized Link State Routing protocol* (OLSR), and *Dynamic Source*

Routing protocol (DSR). The main focus of the experiments was to evaluate the reactivity of the protocols compared to power and bandwidth consumption. The next section then discusses the issue of broadcasting for multi-hop wireless networks. It also proposes and evaluates a new protocol for stateless broadcasting, the *Dynamic Delayed Broadcasting* (DDB) protocol. Multi-path routing, which is discussed in Section 0 allows the establishment of multiple paths between source and destination in wireless mesh networks. Then, multicast routing for mobile ad-hoc networks including two new protocols are presented: QAMNet, which is an approach to improve the *Quality of Service* (QoS) for multicast communication, and *RObust VEhicular Routing* (ROVER), which is a reliable multicast protocol for vehicular networks. Finally, an intelligent navigation system based on traffic monitoring with multi-hop communication for vehicular networks is proposed. It is shown that with the use of multicast routing, intelligent navigation systems that makes re-routes in case of accidents or traffic congestion can be developed.

5.3.1 Performance Comparison of Mobile Ad-hoc Network Routing Protocols

Mobile ad-hoc networks have several features that limit the achievable performance of data communications, such as node mobility, radio link problems, energy constrained operation and the lack of infrastructure itself. A key element with impact on network efficiency is the routing protocol. Ideally, a mobile ad-hoc network routing protocol should be able to provide optimal routes quickly, even in the case of link failures along an active path, with minimum impact on data latency, available bandwidth and device power consumption for any data traffic pattern.

This subsection presents experience testing real ad-hoc scenarios using OLSR, AODV and DSR, respectively [Gom05a, Gom05b, Gom06, Per03, Cla03, TD(07)053]. These protocols account with the most popular implementations available at the time of writing this book. Focus of the study is on the trade-off between reactivity against topology changes as well as bandwidth and power consumption. Some aspects regarding real world mobile ad-hoc network routing protocol implementations are highlighted. Note that simulators contain assumptions that may not reflect actual network operation.

One of the features of a mobile ad-hoc network routing protocol with significant influence on network performance is *Local Connectivity Maintenance* (LCM). Most mobile ad-hoc network routing protocol specifications cover a range of layer two and three link failure detection strategies. The main advantages of a layer two approach are: i) when available, such mechanisms come at no cost and ii) they allow a fast detection of a link break. However, a majority of routing protocol implementations use by default a layer three based LCM mechanism, e.g., Hello messages or layer three *Acknowledgements* (ACKs) generated by the routing protocol itself at each next hop. One reason for such a design choice is that link layer feedback is, of course, link layer dependent. Thus, a layer three strategy does not restrict the usage of an implementation to a specific network interface driver. Nevertheless, a layer three LCM mechanism is expected to incur significant link break detection latency and will consume additional bandwidth and power resources.

A parameter called *Route Change Latency* (RCL) has been defined to measure the degree of reactivity of a routing protocol against route changes [Gom05a, Gom05b, Gom06]. RCL is the total delay between the instant of a link failure in an active path (i.e., a path along which data is sent) and the moment, at which a sending node starts using an alternative route, if such one exists. The formulae for calculating the expected RCL [Gom05a, Gom05b] in OLSR and AODV depend on the *HELLO_INTERVAL* parameter as follows:

$E[RCL_{OLSR}] = 3$ HELLO_INTERVAL

$E[RCL_{AODV}] = 1.5$ HELLO_INTERVAL

The other parameters are assumed to be configured as proposed by default [Per03, Cla03]. Hello messages do not have the same functionality in both protocols. Note that expected RCL is denoted by $E[RCL_{OLSR}]$ and $E[RCL_{AODV}]$ for OLSR and AODV, respectively. RCL measured with default-configured real protocol implementations in scenarios with two different two-hop paths between source and destination is equal to 0.27 s with DSR, 1.53 s with AODV and 13 s with OLSR [Gom05a, Gom05b, Gom06]. Although the expected RCL in OLSR with an ideal implementation is equal to 6 s, this value is too large for highly dynamic environments.

On the other hand, it must be noted how DSR and AODV yield different RCLs (see Fig. 5.5) despite the similarities between both protocols. The reason is that RCL strongly depends on the LCM mechanism used and how it is configured.

Experiments allow quantifying the trade-off that exists between reactivity to topology changes and bandwidth/power consumption in some scenarios [Gom05a, Gom05b]. While significant reductions in RCL can be obtained by increasing control message frequency, the decrease in end-to-end bandwidth and battery lifetime can be tolerable, depending on specific environments and requirements. For instance, in AODV, increasing by 5 the control message rate decreases RCL in the same factor, while only 10% of available bandwidth is lost in a 4-hop string topology scenario with IEEE 802.11b radios. As it can be seen in Fig. 5.6, battery lifetime decreases only by 3.1% in that scenario [Gom05a].

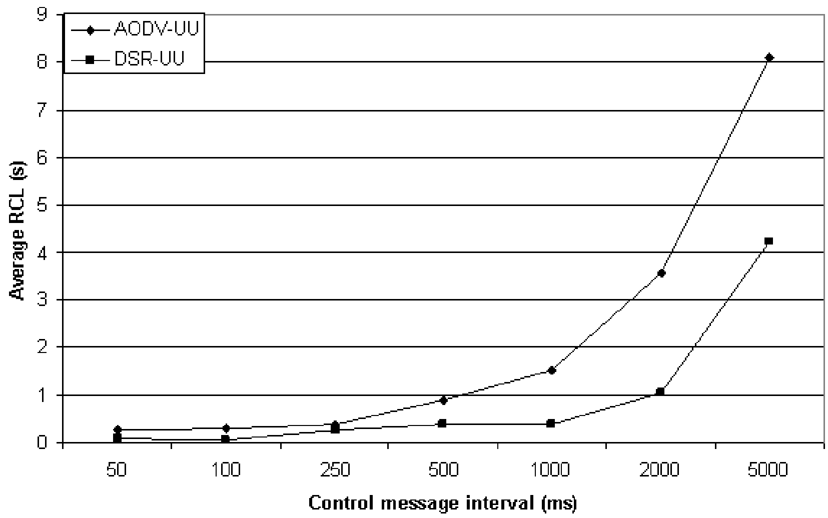

Fig. 5.5. Measured RCL in a two-branch, two-hop path test-bed using AODV and DSR.

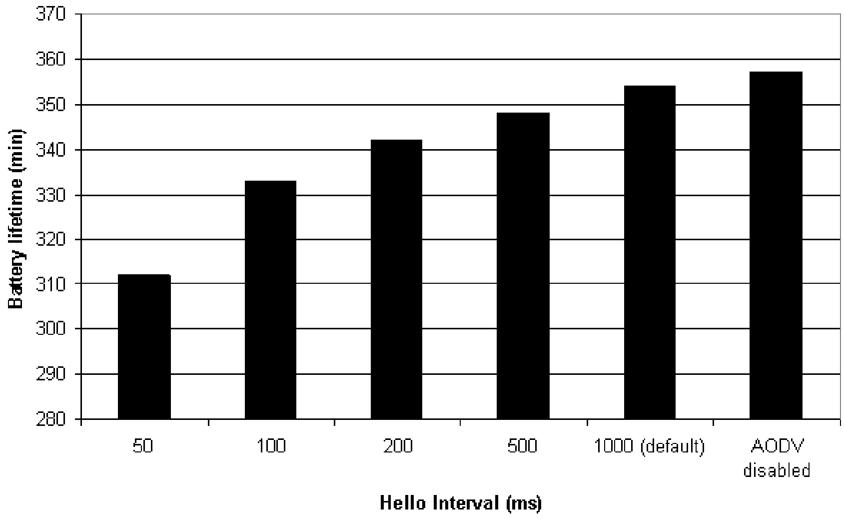

Fig. 5.6. Impact of AODV Hello Interval on laptop battery lifetime [Gom05a].

Experiments have also been carried out with a simplified version of AODV called *Not So Tiny AODV* (NST-AODV) in an IEEE 802.15.4 mesh sensor network test-bed [Gom06]. The platform allows usage of a layer two LCM mechanism. A significant result is that RCL is close to 50 ms in this case.

Hence, this approach, based on a powerful cross-layer mechanism, is between one and two orders of magnitude more reactive than a Hello message mechanism. Note that this result is independent from the radio technology used. The reason is that default / typical HELLO_INTERVAL values are significantly larger than typical layer two acknowledgement waiting times and frame transmission times. Moreover, these benefits come at no cost, since no additional layer three control messages are sent, thus avoiding further bandwidth and energy consumption.

5.3.3 Stateless Broadcasting

Broadcasting is most simply realized by flooding, where nodes broadcast each received packet exactly once. Especially in dense networks, flooding causes many redundant transmissions as well as contention and congestion due to almost simultaneous transmissions. This so-called broadcast storm problem [Ni99] heavily consumes scarce network and energy resources. Two important objectives of any broadcast algorithm in ad-hoc networks are reliability and optimized resource utilization [Wil02]. These objectives are often complementary. Minimizing the number of transmissions also may help reliability and decrease delay [Ni99].

Broadcast protocols can be broadly classified in probability-based, location-based, neighbour-designated, self-pruning, and energy-efficient algorithms. In ***probability-based*** approaches, each node rebroadcasts a message with a certain probability p and drops the packet with a probability of 1−p [Ni99, Haa02]. In [Tse03], the threshold is no longer fixed but adapts to the number of neighbours. In [Car03], the authors proposed to adjust the probability depending on the distance to the last visited node. In ***location-based*** schemes [Ni99], the forwarding decision is solely based on the position of the node itself and the position of the last visited node as indicated in the packet header. Nodes wait a random time and only forward a message, if the distance to all nodes from which they received the message is larger than a certain threshold value. In ***neighbour-designated*** schemes nodes are aware of their neighbourhood. Each node selects a set of forwarders among its one-hop neighbours such that the two-hop neighbours can be reached. A node only forwards packets from the set of neighbours out of which it was selected as a forwarder, thus reducing the total number of transmitted messages. In *Multi-Point Relaying* (MPR) [Lao01], all two-hop neighbours should be covered by the selected one-hop forwarder. Unlike in the neighbour-designated method, in the ***self-pruning*** approaches each node decides for itself on a per packet basis, whether it should rebroadcast the packet. In [Lim00], a node piggybacks a list of its one-hop neighbours on each broadcast packet and a node only rebroadcasts the packet, if it can cover some additional nodes. Other approaches are based on (minimal) connected dominating sets. As the problem of finding such a set is proven to be NP-hard [Mar95], distributed heuristics are proposed in [Sto02, Wu99, Sus00]. In ***energy-efficient*** approaches, nodes have adjustable transmission power. [Wie00] proposed an incremental power algorithm, which constructs a tree starting from the source node and adds in each step a node not yet included in the tree that can be reached with minimal additional power from one of the tree nodes.

The probability- and location-based schemes, as well as simple flooding belong to the category of stateless algorithms as they do not require any neighbour knowledge. The neighbour-designated, self-pruning, and energy-efficient schemes all belong to the stateful protocols. They require at least knowledge of their one-hop neighbours; sometimes even global network knowledge is required. Stateful protocols are barely affected by high traffic loads and collisions [Wil02]. Their performance suffers significantly in highly dynamic networks as frequent topology changes induce excessive control traffic. Furthermore, stateful algorithms may also never converge and reach a consistent state, if changes occur too frequently. On the other hand, stateless algorithms are almost immune to frequently changing network topologies [Wil02]. The main drawbacks of stateless protocols are that the number of rebroadcasting nodes is very high in networks with high node density and that the random delay introduced at each node before rebroadcasting a packet is highly sensitive to the local congestion level [Haa02]. Because stateless protocols use fixed parameters, they are not flexible enough to cope with a wide range of network scenarios. Energy-efficient schemes may not be suited for mobile networks with frequently changing topologies. They require a large computational and communication overhead to construct a power-efficient network structure.

The stateless *Dynamic Delayed Broadcasting* (DDB) protocol [Hei06, TD(06)016] has all the aforementioned advantages of stateless protocols. Unlike other stateless protocols, however, DDB allows making locally optimal rebroadcasting decisions by applying the concept of *Dynamic Forwarding Delay* (DFD) allowing ”better” nodes to rebroadcast first and suppress the transmissions of other neighbours. In other stateless protocols, the sequence of rebroadcasting neighbours is random, hence the unnecessary transmissions occur. Nodes do not rely on information about neighbours and allow avoiding beaconing completely. Beacon-less routing protocols [Hei07] exploit the broadcast property of the wireless medium to determine in a completely distributed way the next node after the packet has been transmitted. DDB is different from the protocol described in [Ni99], which also used location information for designing a broadcast algorithm: The timing of the rebroadcasting in DDB is not random, but nodes apply the concept of DFD to allow optimal rebroadcasting decisions locally. DDB is designed with a cross-layer perspective by coupling the MAC and network layer. DDB can also nicely adapt to changing network conditions and is less sensitive to local congestion. Finally, DDB may be improved to extend the network lifetime by accounting also for the battery level of nodes in the forwarding decision. In the following, we discuss three variants of DDB.

The objective of the **DDB 1** scheme is to minimize the number of transmissions and at the same time to improve the reliability of the packet delivery. We assume that the nodes are aware of their absolute geographical location. Nodes that receive the broadcasted packet use the DFD concept to schedule the rebroadcasting. From the position of the last visited node stored in the packet header and its own current position, a node can calculate the estimated additional area that it would cover with its transmission. Depending on the size of this additionally covered area, the node introduces a delay before relaying the packet [Hei06]. Fig. 5.7 shows in dark grey the area that can be additionally be covered by node B after receiving a message from A.

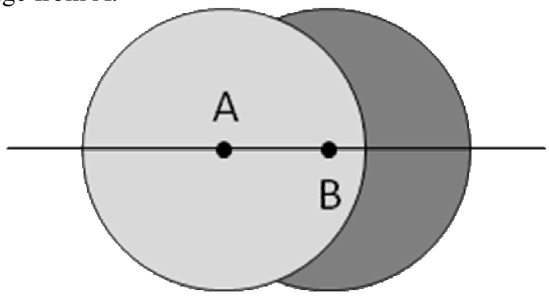

Fig. 5.7. Additionally Covered Area (AC).

The DFD function should yield larger delays for smaller additional coverage and vice versa. We assume the unit disk graph as the network model and we propose a DFD that is exponential in the size of additional covered area. In [Non99] it was shown that exponentially distributed random timers can reduce the number of responses. Let AC denote the size of the additionally covered area, i.e. AC [0, ACMAX],

$$Add\_Delay = Max\_Delay\sqrt{\frac{e - \frac{AC}{eACMAX}}{e-1}}$$, where Max_Delay is the maximum delay a packet can experience at each node and ACMAX is the maximum possible additionally covered area.

Location information may not always be available. In order to minimize the number of transmissions, nodes can use the incoming signal strength as input to the DFD function instead of the additional covered area (**DDB 1 with signal strength**). No position information is required then. For higher signal strength, the DFD should calculate a larger additional delay as we may assume that we are close to the transmitting node, i.e., only cover little additional area.

In the **DDB2** scheme, the delay calculated by DFD depends solely on the remaining battery level of a node and does not take into account the additionally covered area and the signal strength. Nodes with an almost depleted battery schedule the rebroadcasting of the packet with a large delay whereas nodes with a lot of remaining battery power forward the packet almost immediately. Consequently, energy is conserved at almost depleted nodes, which increases their lifetime.

The performance of DDB has been compared with the protocol proposed by [Ni99] and MPR, which uses neighbour knowledge obtained through hello messages. The protocol proposed by [Ni99] was not able to perform well over a wide range of network conditions. The performance degrades under heavy traffic load and high node density, as also observed in [Wil02]. However, DDB did not suffer from these drawbacks, but the performance of DDB even improved for those scenarios of high traffic load and high node density. MPR performed well in most scenarios, except in highly dynamic networks where the delivery ratio collapsed. The delay of MPR was the shortest in all simulated scenarios closely followed by DDB whose delay was approximately 10% longer, except in highly congested networks. On the other hand, DDB outperformed MPR significantly considering the efficiency of the algorithm. Compared to MPR, DDB only required about half of the transmissions to deliver the packet reliably to all nodes. Furthermore, as DDB is stateless, its performance was completely unaffected in highly dynamic networks.

5.3.3 Multi-Path Routing in Wireless Mesh Networks

Wireless Mesh Networks (WMNs) provide a cost efficient way to interconnect existing wireless networks as well as to supply larger areas with network access. WMNs offer a more robust and redundant communication infrastructure than the wireless networks deployed today. They offer communication facilities in situations where certain systems, e.g., Global System for Mobile Communications (GSM), might be overloaded.

The unreliability of the wireless medium, resource-constrained nodes and dynamic topologies make wireless mesh and mobile ad-hoc networks prone to transmission failures, node failures, link failures, route breaks, and congested nodes or links [Mue04]. One important approach to overcome this problem and to exploit wireless mesh networks for robust real-time communication is path diversity. For each destination multiple routes are provided by a multi-path routing protocol, e.g., to support real-time data transfer. Appropriate coding and path allocation is selected for the given network conditions and, therefore, the degree of redundancy in transmission is set.

Multi-path routing allows the establishment of multiple paths between source and destination. This provides increased reliability of the data transmission and fault tolerance or load balancing. Several multi-path routing approaches enhance the well-known single path routing protocols AODV [Per03] or DSR [Joh07]. *Split Multi-Path Routing with Maximally Disjoint Paths in Ad Hoc Networks* (SMR) [Lee01] extends DSR to create two maximally disjoint paths. The routing scheme prohibits intermediate nodes from replying on route requests. Intermediate nodes forward duplicate *Route Request* (RREQ) messages, if they arrive through a different link and if their hop count is equal or lower than the previously received one(s). The destination responds to the first RREQ with a *Route Reply* (RREP) message as it represents the shortest delay path. From subsequently received RREQs the destination selects the maximally disjoint path and establishes a second path by sending a RREP. Both paths are then equally used for data transmission.

Node-Disjoint Multipath Routing (NDMR) adapts the same SMR scheme for AODV. The criteria for forwarding the RREQs are the same as in SMR, but the behaviour of the destination is changed. After setting up the shortest delay paths the destination only selects paths that are node-disjoint to the already established one(s).

Ad-hoc On-demand Multipath Distance Vector protocol (AOMDV) [Mar02] and *Ad-hoc On-demand Distance Vector Multipath* protocol (AODVM) [Ye03] represent other multi-path variants of AODV. AOMDV discovers multiple loop-free paths during a single route discovery. AOMDV replaces the hop count of AODV by an advertised hop count to a destination, which represents the maximum hop count for all available routes to the destination. The routing entries further contain a list of next hops with hop counts instead of one simple next hop for each destination. RREQ or RREP packets update the routing information at a node either for a reverse or forward path. Duplicates of such route advertisements may define alternate paths to destination or source. Like in AODV, sequence numbers guarantee the freshness of the routing information. In order to avoid routing loops, alternate paths are only accepted if their hop count is smaller than the advertised hop count for the same destination sequence number. The reception of a newer destination sequence number reinitializes the advertised hop count as well as the next hop list for this destination. AOMDV may either find node-disjoint paths or link-disjoint paths. For node-disjoint paths, each node simply accepts RREQs arriving from different neighbours. The support of only link-disjoint paths requires further changes.

RREQs include the first hop taken by them and the nodes store a first hop list for each received RREQ. At intermediate nodes, duplicates of RREQs update the reverse path if they include a new first hop beside the freshness and hop count criteria. In either case, node- or link-disjoint, the destination replies to a predefined number of RREQs arriving from different neighbours with a RREP and sets up multiple paths.

Intermediate nodes in AODVM [Ye03] do not drop duplicate RREQs. Duplicates are stored in a RREQ table at each intermediate node. The destination replies to all RREQ received from different neighbours with a RREP. Nodes on the path overhear these RREPs. If a node is assigned to a route, it is deleted from its neighbours' RREQ tables. AODVM therefore finds node-disjoint paths.

The authors of [Sun03] propose a scheme based on SMR that offers Quality of Service (QoS) support in wireless ad-hoc networks. The different QoS requirements are achieved by adjusting the number of paths, the parity length of the *Forward Error Correction* (FEC), and the traffic rate on each path. Local link information used for the calculation is collected or predicted at each node and distributed via the routing messages.

Resilient Opportunistic Mesh Routing (ROMER) [Yua05] is a routing solution based on multipath routing. It directly uses the path diversity to enhance the robustness of the routes. A run-time forwarding mesh is established on per-packet basis including the long-term minimal cost path. This mesh offers the current candidate routes. ROMER selects the highest-rate link for the main data transmission and delivers redundant data randomly over other high-rate links to increase resilience against lossy links. Various other multi-path routing protocols exist: *Similar Node-Disjoint Multipath Routing* (SNDMR) [Xu05] as an enhancement of NDMR, the *AODV Backup Routing* (AODV-BR) [Lee00], *Multipath-DSR* (MP-DSR) [Leu01], *Multipath Source Routing* (MSR) [Wan01], and *Caching and Multi-Path routing protocol* (CHAMP) [Val03].

AODVM and SMR have been implemented in the Linux based mesh network [TD(06)051]. The implementations are based on code from the University of Uppsala (AODV-UU, DSR-UU) [Cor07]. Initial tests in the experimental WMN indicate that in order to fully exploit path diversity the multi-path routing has to be enhanced with multi-channel functionality. Mutual interference between the alternate paths as well as between links on the same route has to be considered in the routing decision. The selection of appropriate coding and path allocation for the support of real-time communication in wireless mesh networks is another open research issue.

5.3.4 Multicast Routing

In mobile ad-hoc networks, efficient support of multipoint communications is essential in order to provide services like group audio and video conferencing, dissemination of data to a set of receivers or collaboration of a group of users. Also, most of the important interactive group services such as gaming or conferencing have very strong QoS requirements regarding delay and bandwidth.

Multicast routing protocols for mobile ad-hoc networks can be classified into tree- or mesh-based depending on the underlying forwarding structure that they use. Tree-based schemes such as [Roy99, Ji98, Jet01b] construct a multicast tree from each of the sources to all the receivers using either source based trees or shared trees. Mesh-based approaches such as [Lee02, Gar99] compute several paths between sources and destinations. Hybrid protocols such as [Bom98, Sin99] try to combine the robustness of mesh-based ad hoc routing and the low overhead of tree-based protocols. Finally, stateless multicast protocols such as [Ji01, Jet01a] do not maintain forwarding states on the nodes as for example the set of nodes to traverse is included in the data packets themselves.

Many vehicular network applications require position-based multicasting, e.g., for disseminating traffic information to vehicles approaching the current position of the source [Sic07]. Geocast protocols that forward messages to all nodes within a *Zone of Relevance* (ZOR) [Mai04] are the natural match for this type of routing. Some applications will require multicast transmission with end-to-end QoS. Flooding-based geocast protocols are not intended for these types of applications.

Therefore, there is a need to develop multicast protocols for *Vehicular Ad-hoc Networks* (VANETs) that can support end-to-end QoS mechanisms implemented in a transport layer protocol.

5.3.4.1 QAMNet

QAMNet [Teb04, TD(06)029] is an approach to improve the Quality of Service (QoS) for multicast communication in *mobile ad-hoc networks* (MANETs). We extend existing mesh based multicast routing protocols by introducing traffic prioritization, distributed resource probing, admission control mechanisms, and adaptive rate control of non-real-time traffic based on MAC layer feedback to maintain low delay and required throughput for real-time multicast flows.

When a QAMNet node in a MANET has real-time traffic to send to a multicast group, it starts with flooding the entire network with a control message to advertise the multicast source to receivers, which carries the first data packet using piggybacking. This special Join-probe message contains *Bottleneck Bandwidth* (BB) and *Required Bandwidth* (RB) fields. Upon reception of the first, non-duplicate, Join-Probe packet, intermediate nodes set pointers towards their upstream nodes and rebroadcast it, after modifying the probing request information. Each intermediate node additionally updates the bottleneck bandwidth field, if the local bandwidth availability at the given node is lower than the current value. Bandwidth availability at the local node is calculated based on MAC layer utilization.

Once a Join-Probe packet reaches a multicast receiver, BB indicates the bottleneck bandwidth found along the path. The receiver collects several Join-Probe packets received from other branches of the multicast mesh, evaluates whether the BB with the largest value is greater than RB and if so creates a Join-Reply, piggybacking a Probe-Response, which contains the largest BB and the same value in the RB field that it received in the Join-Probe. The Join-Reply is relayed by the intermediate nodes all the way from the sink to the source following the pointers established during the propagation of Join-Probes. Each intermediate node waits a short time to collect Probe-Response packets from other branches of the multicast mesh and updates the BB value in the Probe-Reply packet with the maximum value of all received Probe-Response messages when forwarding to the source. It also sets a flag for the given multicast group, if the forwarded BB value is larger than RB.

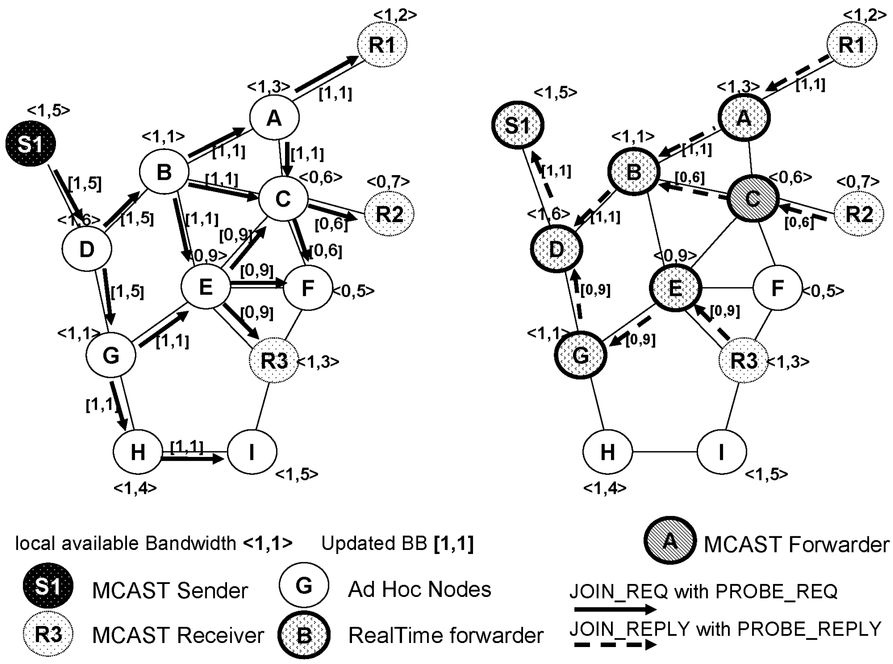

Fig. 5.8. Building of resource aware multicast mesh structure in QAMNet

Once the Join-Reply reaches the source, it multicasts (real-time) packets with the help of the (real-time) forwarder nodes through the forwarding mesh. For all packets requiring preferential treatment, the source sets the *Type of Service* (ToS) bit in the IP header and sends it via MAC layer broadcast. Before an intermediate forwarding node re-broadcasts the packets, the classifier of that node checks whether the flag for the given group is set. Then the packet bypasses the nodes' shaping mechanism, remains unregulated and is directly passed to the MAC layer for re-broadcasting. If the flag is not set, the node will unmark the ToS bit in the header and put the packet into the shaper.

QAMNet has been implemented in ns-2 and simulation results showed that by reusing control messages already used by those protocols, the approach does not significantly increase control overhead nor state stored at nodes. The average delay of real-time packets can be controlled efficiently by QAMNet even for high traffic scenarios for all admitted real-time flows. It only marginally increases with higher mobility. The drawback of the approach is the additional delay for the best-effort packets as those packets are regulated by the shaper.

5.3.4.2 ROVER

The RObust VEhicular Routing (ROVER) protocol [Kih07, TD(06)049] offers reliable geographical multicast. Its objective is to transmit a message, M, from an application, A, to all other vehicles within an application-specified ZOR, Z. The ZOR is defined as a rectangle (although other definitions can be easily accommodated) specified by its corner coordinates. The *Zone of Forwarding* (ZOF) is defined as 15 meters outside the boundaries of the ZOR.

The first time the routing layer receives a packet [A, M, Z] from the application layer, a route discovery process is triggered. The objective of the route discovery process is to build a multicast tree from the source vehicle to all vehicles within ZOR Z. The route discovery process is initiated when the originator vehicle floods a *Zone Route Request* (ZRREQ) message containing its identification number, location, the current ZOR, and a *Route Sequence Number* (SS) throughout the ZOF. Any vehicle that receives a ZRREQ for the first time for this session sequence number accepts the message, if the vehicle is within the ZOF, and is not too far away from the sender.

If a vehicle accepts a ZRREQ, it replies to the one-hop vehicle that forwarded the ZRREQ with a *Zone Route Reply* (ZRREP) message, containing its *Vehicle Identification Number* (*VIN*). The VIN is deployed in such way that each vehicle is anonymous to all other vehicles. It also stores the information [*SS*, *Z*] in a routing table. Finally it re-broadcasts the ZRREQ, including the original *VIN*, *ZOR*, and *SS*. The vehicles in the ZOF but not in the ZOR do not reply to ZRREQ messages unless they receive a reply themselves. The sequence number *SS* in conjunction with the *VIN* of the source vehicle (originator) is used as a unique identifier in the routing tables formed by the route discovery process. After flooding the ZRREQ throughout the ZOF, the ZRREP messages are only transmitted to the node from where the ZRREQ came.

Since each vehicle stores next-hop(s) information about the source *VIN* and *SS*, data will be forwarded through the tree as a function of those numbers. The source forwards the data packets immediately after it receives a ZRREP message. The source and all forwarding nodes in the multicast tree send the message *M* per unicast to all the vehicles from which they have received a ZRREP. The message is also stored in a buffer for a short time in case a ZRREP arrives after it receives the message. Thus, each message is propagated through the multicast tree according to the “route table” stored during the route discovery process. All receivers also deliver the packet, if they are within the ZOR. Since the data is transferred using unicast, it benefits from the normal MAC layer acknowledgments.

We have evaluated ROVER using the realistic simulation package Jist/SWANS with the STRAW module for vehicular movements along streets. We constructed a straight highway in TIGER [Cen07] format and then used this road in the simulations. To evaluate the performance of the proposed routing protocol, we used a generic data transfer application. In this application a vehicle sends a message to vehicles behind it. The results show that ROVER delivers the data with reasonable delays to 100% of the intended vehicles for almost all scenarios. Also, ROVER could be used by applications that require end-to-end QoS, by implementing a transport layer protocol that uses the multicast tree set up by ROVER.

5.3.4.3 An Intelligent Route Guidance System

Since the concept of *Intelligent Transportation System* (ITS) was put forward in 1991 by the U.S. Department of Transportation, it has been viewed as a promising way to tackle modern traffic problems. One part of the ITS architecture is inter-vehicle communication, which would enable a vehicle to, for example, monitor the traffic situation around itself. Such a traffic monitoring application could be used in, for example, an intelligent route guidance system that can minimize the travel time for the driver by making rerouting suggestions when it receives information about traffic congestion or accidents. In this section, we will present and evaluate an intelligent route guidance system that relies on a traffic monitoring application based on inter-vehicle communication [TD(07)032].

Dissemination of data in the traffic monitoring application is performed with so called geocast routing [Jos06]. Geocast is a location based technique for multicast routing of data, in which a message is spread to a selection of vehicles in a ZOR. The traffic monitoring application uses the digital road map that each vehicle is equipped with. The digital map is divided into segments of certain length, which may vary depending on different roads. In an experiment a grid has been used to simulate a city map. The road between two intersections is one segment, which are identified by its ID. Each node (vehicle) will disseminate data messages containing not only its *Global Positioning System* (GPS) position, but also the ID of the segment it is on. Each vehicle transmits data about itself with time intervals of *2* s. The data is spread to all vehicles within a square of 1000 m x 1000 m. The data transmitted is the vehicle's position and speed. All vehicles keep a database with current data for each road segment. The maximum value among all the data of a road segment received in one period is chosen as the speed for the traffic flow on that segment during that period.

The objective of the intelligent route guidance system is to ensure that the driver always drives on the best feasible route to his/her destination. The route guidance system uses real-time traffic information received from the traffic monitoring application described above. In this text, the best route is the one that minimizes the travel time for the driver. The route guidance system continuously monitors the surrounding traffic situation. If it detects that the speed on a road segment has changed significantly between two periods (we use a threshold of 7 m/s), it will try to make a re-route to a faster route using the most updated data. A shortest path algorithm is used to calculate the optimal route. The cost of each road segment is calculated by dividing the length with its current speed. Each road segment has a default speed (i.e., the speed limit) that will be used in cases where no updated information exists.

We implemented the system in a modified version of the open source simulator JiST/SWANS with a mobility module called STRAW. Jist/SWANS is a Java-based simulator for mobile ad-hoc networks. The STRAW module implements a mobility model with real digital maps, which are based on the TIGER system available from the US Census Bureau Geography. We used a road grid model as a simplified city road map. Each road block was 128 m x 100 m, and the total size of the road map was 2688 m x 2000 m. We assumed that all roads were single lane, so it was not possible for cars to change lanes. In order to highlight the function of the route guidance system, we used 10 nodes in the simulations. All the nodes are heading to the same destination. In the simulation, node 4 will slow down after 35 seconds, causing congestion. Node 5 is on the same segment as node 4. Node 6 is on the segment immediately after node 4's. Nodes 1-3 also use the congested road segment. Nodes 7-10 have routes that should not be changed.

The speed information will be disseminated by the traffic monitoring applications to all other vehicles in the vicinity. At the end of the monitoring period, nodes have received data from all the nodes on one segment and thereby can determine that the speed on that segment has slowed down dramatically. This will trigger the rerouting process. Since nodes 7-10 will not take the congested segment anyway, they will not re-route. Table 5.1 shows how much travel time each driver has saved by re-routing.

Vehicle	Travel Time(s) (no reroute)	Travel Time(s) (reroute)	Saved Time (percentage)
1	250	129	48
2	250	135	46
3	250	119	52
5	243	160	34
6	246	178	28
7	102	102	0
8	117	117	0
9	77	77	0
10	77	77	0

Table 5.1. Time Saved by the Dynamic Route Guidance System

5.4 Transport Protocols for Ad-hoc Networks

Reliable data transport in networks is a required service provided and controlled by transport-layer protocols. Currently, the dominant protocol for the end-to-end transport of data is TCP. The original TCP provides full-duplex in-order delivery of data, accompanied with flow and congestion control mechanisms. The sending rate of the TCP sender is controlled using a sliding window algorithm, where the window size is constantly changing, according to the TCP flow and congestion control mechanisms. The basic principle of the TCP congestion control algorithm is that any lost packet is lost because of network congestion.

In MANETs, the spectrum of reasons for packet loss is much larger. Interference in the wireless medium, inter-flow instability, dynamic topologies and hidden and exposed terminal effects all contribute to an increased packet loss. Designing efficient transport protocols for MANETs is therefore a complex problem.

The majority of the research effort on transport protocols for ad hoc networks is still being put in understanding the difficulties TCP mechanisms face in MANETs and in developing TCP modifications capable of improving the data delivery. The main drive behind this is the importance of integrating MANETs into the global Internet and thus enabling efficient and transparent communication of MANET nodes with the rest of the network.

The analysis given in this section identifies as the main problem the need for TCP to be able to recognize the cause of packet loss and to react in the most efficient way. MANETs are characterized with low bandwidth-delay product and constantly changing routes, and most existing solutions advocate a close control of the sending rate, especially for TCP connections that span several nodes along a multi-hop route. TCP has been designed to maximize the use of network resources and while in wired networks this produces excellent performance, in MANETs TCP grows its sending window beyond its optimal value and overestimates the bandwidth-delay product, thus creating increased delays and unnecessary retransmissions.

Additionally, in 802.11 MANETs link-layer activities can add variable delay in the TCP segment delivery. The unfairness of the Medium Access Control in 802.11 networks, for example, can produce unexpected delays, especially in large networks. Hidden and exposed terminal effects in chain network topologies add more link-layer delays and make RTT prediction complex and unreliable. Detailed understanding of all of these effects contributes greatly to modifications that are required for TCP to perform in a more optimal way in MANETs. The sending window control is necessary – it can be achieved either by fixing the sending rate based on the network topology knowledge, or by utilizing cross-layer mechanisms to obtain information from the link layer or by closely analyzing the impact of delaying TCP acknowledgements and choosing the right strategy.

In this section, the problem of TCP stability is analysed further, providing a selection of solutions for improving the TCP operation. First, the performance of different TCP flavours in MANET chain topology with moving nodes and variable chain length is analysed. This is followed by a description of cross-layer design solutions which utilize link-layer information to adjust the TCP sending rate. Finally, the section gives a comparative analysis of the existing work on minimizing the traffic overhead caused by redundant acknowledgements. Experience with delaying TCP acknowledgements is examined, together with more general work of avoiding spurious TCP retransmissions in wireless networks.

5.4.1 Performance Comparison of Different TCP Flavours

Traditional TCP(was designed for wired networks but as more people move from wired towards wireless connections the demands for performance and ease of use of wireless communication becomes equivalent to those of wired communication. Wired networks are stable and have usually a high bandwidth-delay product, packet collisions and bit-errors are exceptions. This is a fundamentally different environment than in a mobile ad-hoc network, where the medium and transmissions routes are fluctuating causing the transmission between nodes to be unstable and unpredictable. The bandwidth-delay product is low compared to a wired network. Bit errors and collisions due to intra/interflow interference are common. In an Internet connected mobile ad-hoc network it is presumed that a majority of the traffic will be destined for the Internet. This increases the contention and possibilities for collision, hidden / exposed terminals etc. around the gateway node. Moreover, since TCP uses sliding window to control the pace of how many packets to send, there is a large possibility that several of the packets in transit get lost when a handover to a new gateway takes place. However, due to the large number of users and existing equipment, one cannot change TCP completely.

Breaking TCP end-to-end semantics has several drawbacks, e.g., security and handover. Therefore, the transport protocol in an Internet connected mobile ad-hoc network must be TCP compatible. Furthermore, in a nearly fully utilized network, it is essential for TCP flow fairness that the MAC layer is fair. Unfortunately, IEEE 802.11 MAC is not fair and this results in disadvantageous behaviour [Nan00]. It has been shown that fairness among flows are reduced when TCP flows local to a mobile ad-hoc network compete with flows from/to the Internet [Kai02] and when flows over a few hops compete with longer hop count flows. In a wired network, for which TCP [Flo99, Bra99] was designed, packets drops are most likely due to an overload in some part of the transmission route, i.e., a state of congestion at a node. Therefore, TCP's congestion control mechanism aims at quickly reducing the sending rate to remedy the congestion as soon as a packet drop has been discovered. In a mobile ad-hoc network this is often a misinterpretation, because packet drops are often the result from bit errors, route changes, or a handover to another gateway. There, the correct procedure instead would be to resend the packet as fast as possible. However, sending data too quickly is also a problem in a wireless network. This is a complex cross-layer interaction between the MAC, routing, and transport layer. For example, when TCP probes for bandwidth during the slow start phase there is a high probability for MAC layer contention induced packet loss, which will cause the routing protocol to trigger route error messages regardless whether the route is valid or not and therefore further increase contention [Nah05]. Even in congestion avoidance phase with only an additive increase of the amount of packets sent for every Round Trip Time (RTT), TCP might send too many packets into the network. In a wired network an increase by 1 packet may be only 0.1-1% of the available network capacity. However, in a mobile ad-hoc network an increase of 1 packet may be several 10% of the available network capacity and will rapidly lead to MAC layer contention.

We studied the performance of TCP in hybrid mobile ad-hoc networks [Kar07, TD(07)051] by simulating three TCP variants using ns2 [Est07, [Nsn07]. In the simulation AODV(-UU [Cor07] was used as routing protocol, TCP packet size was 1460 bytes and the queue size of nodes was 50 packets. During the simulation, one or two mobile ad-hoc network nodes uploaded files to a wired host. The transmission range of mobile ad-hoc network nodes was set to 250 m, interference range to 550 m and physical layer bandwidth was fixed to 2 Mbps in order to avoid effects of automatic rate adaptation and focus on TCP performance difference. NewReno was compared to TCP Vegas (using $\alpha=\beta=2$) and *TCP with Adaptive Pacing* (TCP-AP) [ElR05] (weighting factor $\alpha=0.7$ and history size $N=50$. In the example simulation, the sending node moves constantly from left to right and back at varying speeds, hovering above a chain of 5 nodes. The nodes are at a 200 m distance from each other and the middle node in the chain is the gateway. Consequently the route to the gateway changes together with the number of hops between the gateway and the mobile node. The receiving node is directly connected to the gateway via a fixed uplink where the capacity manually can be changed between 100 Mbps / 2 ms delay (simulating a WLAN uplink) and 756 kbps / 25 ms delay (simulating an *Asymmetric Digital Subscriber Line* (ADSL) uplink).

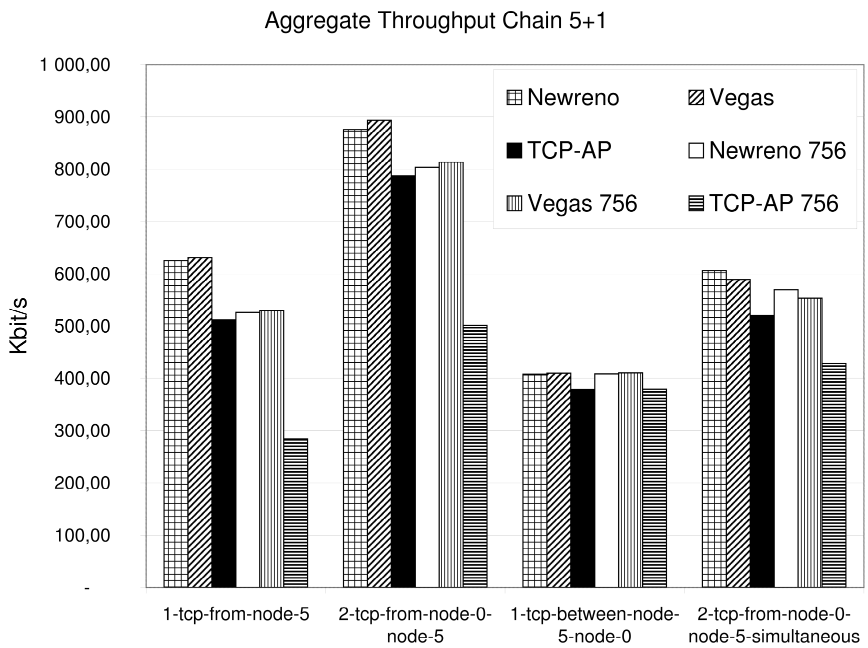

Fig. 5.9. Throughput (kbps) in a 5 node chain with different uplink characteristics.

As it can be seen from Fig. 5.9, TCP Vegas performs as good as or better than the two other TCP variants. The impact of uplink bandwidth and delay (indicated as 756 in Fig. 5.9 for ADSL) on NewReno and Vegas is around 10%, whereas the performance of TCP AP is reduced by almost 50% when the uplink is changed. The throughput for two competing flows are lower than for the single flow, showing the negative effect the MAC layer contention and the adverse interactions between the MAC, routing and transport layer, when both nodes compete to reach the gateway.

Fig. 5.10 shows the goodput for one single flow from the moving node towards the fixed host over time. As expected, the higher number of hops, the lower is the goodput. The amount of route changes during this time period is interesting. The route is changed eight times when Vegas and TCP AP are used, which is in line for having an optimal route. When NewReno is used, the route changes 26 times due to overload of the network. When packets cannot be delivered by the MAC layer, an error to the routing layer is returned. The routing layer (AODV-UU in this case) considers this as a route error and starts an unnecessary route search. This is an indication of the problems that follows the more aggressive bandwidth probing and reactive congestion detection of NewReno.

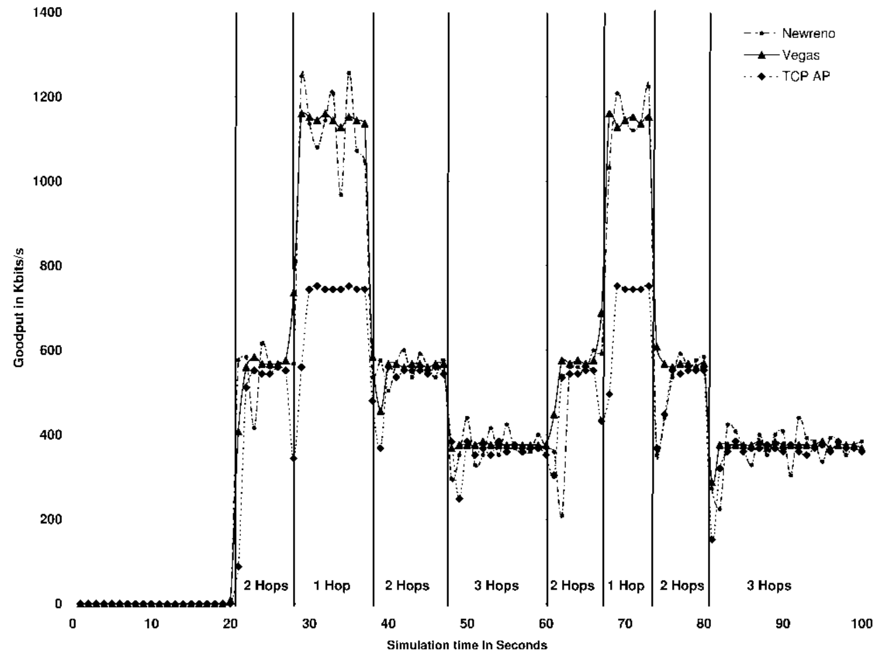

Fig. 5.10. Goodput for flow from node 5 towards fixed host over time.

5.4.2 Reliable TCP Delivery

Designing reliable data delivery mechanisms in ad hoc networks is a challenging task. Ad-hoc networks introduce a dynamic and unpredictable environment, which is fundamentally different to the wired Internet environment for which traditional, TCP(-based reliable delivery was designed. To analyse this problem further, we give a brief analysis of the deployment of TCP in ad hoc networks that use IEEE 802.11 technology.

In TCP, the effective limit on outstanding data known as *send window* (swnd), is set as the minimum of the *congestion window* (cwnd) and the available receiver window. The performance of TCP directly depends on the swnd. It is well known that the optimal value for swnd should be proportional to the bandwidth-delay product of the entire path of the data flow [Che04a]. As shown in [Fu02a, Xu01, Che04b] the bandwidth-delay product of a TCP connection over multi-hop IEEE 802.11 networks tends to be very small. This is mainly because in IEEE 802.11 the number of packets in flight is limited by the per-hop acknowledgements at the MAC layer. Such property is clearly quite different from wired networks, where multiple packets can be pushed into a pipe without waiting for the first packet to reach the other end of the link. The key problem for TCP in ad hoc networks is that, as shown in [Fu05], the TCP grows its congestion window far beyond its optimal value and overestimates the available bandwidth-delay product.

In general, TCP instability can be broken down into two broad categories: TCP inter-flow and TCP intra-flow instability, where the former happens when nodes belonging to different connections interact, and the latter refers to the situation where successive transmissions of packets in a single TCP flow interfere with each other and result in a large number of contention related packet drops and hence TCP instability in the network. In particular, when TCP runs over IEEE 802.11, the intra-flow interference can be broken down into the following categories: interference of TCP packets with each other, interference between TCP packets and 802.11 control packets, and interference of IEEE 802.11 control packets with each other. Here, TCP packets refer to either TCP DATA or TCP ACK packets, and 802.11 control packets include a MAC ACK (IEEE 802.11 acknowledgements) and *Request To Send* (RTS) / *Clear To Send* (CTS) if used. We should note that according to the IEEE 802.11 MAC standard, if a node cannot reach its adjacent node within the limited number of allowed retries, it will drop the packet. These packet drops are wrongly perceived as congestion by the TCP and result into false triggering of TCP congestion control algorithm, frequent TCP retransmissions and therefore TCP instability. This instability is dangerous as it may create a situation where the receiver (data sink) does not receive any packets for a period of time causing the connection throughput to drop to zero or fluctuate rapidly.

Fig. 5.11 shows the change of cwnd and the instances of TCP retransmission in a simulation of a 4 hop chain topology using 802.11 MAC. Here, the only cause of packet drop in the network has been set to contention losses to verify the problem of TCP and link layer interaction in ad hoc networks. The results fully support the above argument and confirm that TCP behaviour towards overloading the network causes extensive packet contention drops in the link layer. This observation is also confirmed in many studies such as [Fu02a, Fu05, Xu02] by showing that TCP with a small congestion window (e.g., 1 or 2) tends to outperform TCP with a large congestion window in 802.11 multi-hop networks. To enforce the congestion window to a small value, the authors in [Che04a] showed that the bandwidth-delay product of ad hoc networks is limited to *Round Trip per Hop Count* (RTHC). They then refine this upper bound based on the 802.11 MAC layer protocol, and show that in a chain topology, a tighter upper bound of approximately 1/5 of the round trip hop count of the path outperforms in comparison to default TCP(. The authors in [Fu05] impose a hard limit of 1/4 of chain length based on transmission interference in 802.11.

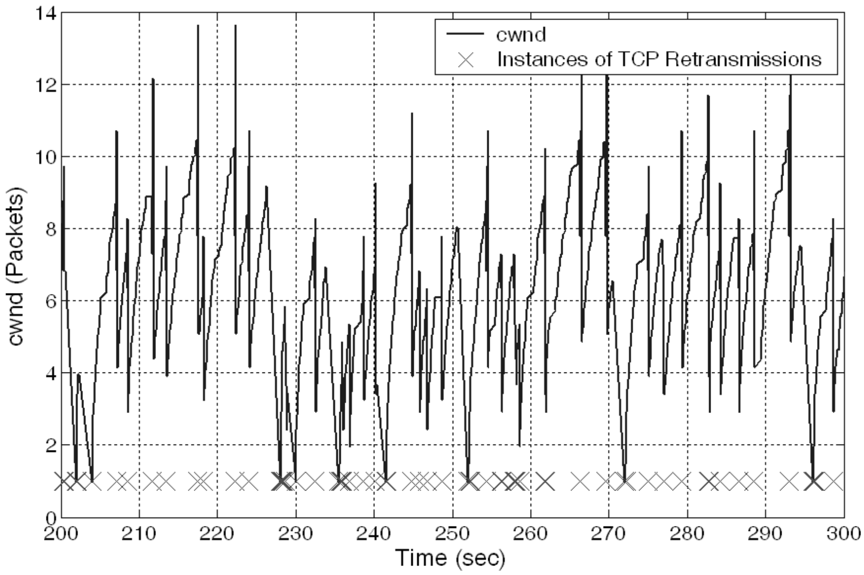

Fig. 5.11. Change of *cwnd* and the instances of TCP retransmission in a 4 hop chain topology.

A comprehensive cross-layer solution of the problem is given in [Ham07, TD(07)026], where *TCP Contention Control* (TCTC) is proposed. The TCTC mechanism adjusts the TCP transmission rate to minimize the level of unnecessary contention in the intermediate nodes. To this aim, during fixed probe intervals, the TCP receiver monitors both the achieved throughput and the level of contention experienced by packets during that interval. Then, based on these observations, the receiver estimates the optimum amount of traffic to get the maximum throughput and the minimum contention delay for each connection. Finally, TCTC propagates the information back to the sender to adjust its transmission rate. Using this information, the TCP sender now sets its transmission rate not merely based on the level of congestion in the network and the available buffer size at the receiver but also on the level of medium contention experienced by intermediate nodes. More precisely, while TCP congestion control adjusts the TCP transmission rate to avoid creating congestion in the intermediate network buffers, TCP contention control adjusts the TCP transmission rate to avoid creating queue build up in the intermediate network buffers. Fig. 5.12 and Fig. 5.13 show the improvements that can be obtained in 4x4 grid topology and chain topology [Ham07]. The simulation results shown here include the overall number of packets in buffers and the overall number of TCP retransmissions.

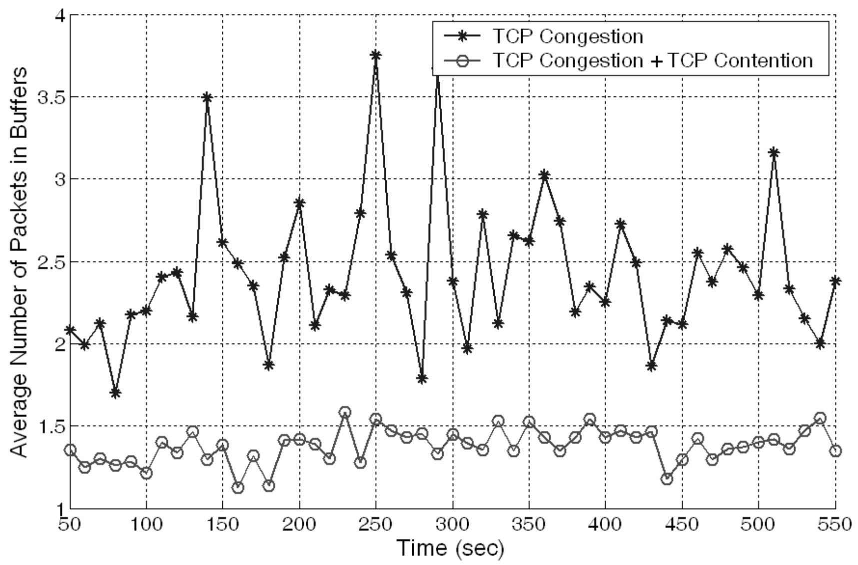

Fig. 5.12. Average number of packets in all buffers in a 4x4 grid topology.

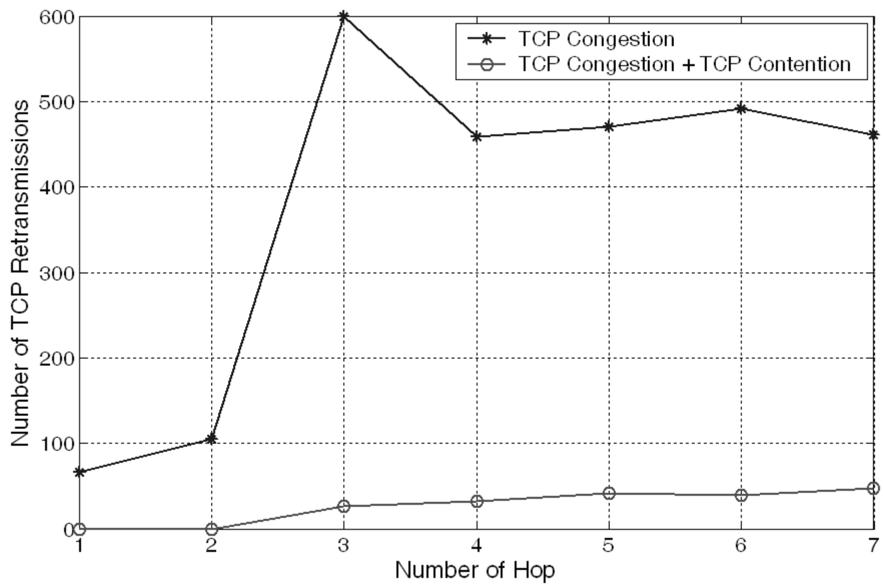

Fig. 5.13. Number of TCP retransmissions in a chain topology.

5.4.3 Adaptive TCP Acknowledgement

Several proposals for improving TCP(performance or replacing its mechanisms over multi-hop wireless networks have emerged in recent years [Bia98, Cha98, Fu02b, Hol99, Liu03, Liu01, Sun05]. The strategy of these proposals is to enhance the TCP sender to react properly to lost packets caused by reasons other than congestion. Hereafter, we focus on proposals that aim to minimize traffic overhead caused by redundant acknowledgements (ACKs). We also discuss related work on TCP spurious retransmissions.

In [Jim03] the impact of delaying more than two ACKs on TCP performance in multi-hop wireless networks was investigated. In a chain topology of nodes, substantial improvement may be achieved by delaying three to four ACKs. Unless the sender's retransmission timer expires, the receiver always delays four packets, except at session start-up. During start-up, the receiver begins delaying one ACK only and increases it until four based on the sequence number of the received data packets. The receiver uses a fixed interval of 100 ms for timing out and does not react to packets that are out-of-order or filling in a gap in the receiver's buffer, as opposed to the recommendation of the standard *Delayed Acknowledgments* (DA) algorithm [All99]. Their main mechanism, called *Large Delayed ACK* (LDA) hereafter, does not adapt to changing medium conditions.

The impact of using extended delayed acknowledgments on TCP performance was studied in [Joh95]. In experiments using a test-bed of workstations the kernel's TCP algorithm has been changed to allow different numbers of combined ACKs by the receiver instead of only two as proposed in the specification of [Bra89]. In this way, the receiver was adjusted to delay a higher number of ACKs, ranging from one to twenty ACKs. Delaying ACKs in large numbers is always beneficial in short-range networks, but may be inappropriate for long-distance networks, especially if congestion is present. This is a consequence of the high interference on RTT estimation caused by delayed ACKs. The longer the end-to-end connection, the longer it takes for the TCP sender to detect lost packets.

It was shown in [All98] that TCP performance may be decreased by delayed ACKs mainly during the slow start phase. One reason is that the exponential growth of the TCP congestion window in that phase may produce data bursts in the network, inducing packet drops in the routers buffer. Another problem lies in the ACK-clocked behaviour of TCP, in which the sender only increases its congestion window by one upon each received ACK. This limits the sender data rate in scenarios, where slow start is often invoked. The author proposed two mechanisms to handle the side effects of delayed ACKs: delayed ACKs after slow start and byte counting. The former requires signalling between sender and receiver to keep the receiver informed about whether slow start at the sender is active or not, so the receiver only delays ACKs when the slow start phase is over. Byte counting allows the sender to increase its congestion window on the basis of the number of bytes acknowledged by each ACK instead of the number of ACKs. This procedure can lead to prohibitive bursty traffic conditions, so the author also suggested limiting the number of packets sent in response to each incoming ACK to a value of 2.

Eifel [Gur03, Lud03, Lud05] and *Forward Retransmission Timeout* (F-RTO) [Sar05, Sar03] are approaches that propose to improve TCP performance against spurious retransmissions in wireless environments. The Eifel algorithm aims to eliminate the TCP retransmission ambiguity in order to solve the problems caused by spurious timeouts and spurious fast retransmissions. It uses the TCP timestamp option, so the sender may effectively determine whether a given packet is transmitted for the first time or whether it is a retransmission. By checking the timestamp in the ACKs, the sender is able to infer spurious retransmissions. If a retransmission is found to be spurious, the sender restores the parameters of the congestion control that were in place just before the unnecessary retransmission has occurred. As a consequence, the congestion window returns to its previous value and the transmission rate is not reduced wrongly. In a later version, the algorithm encompasses specific techniques for noisy networks, including a more appropriate way of updating the retransmission timer and a better policy for the congestion window restoration. F-RTO is an algorithm implemented at the sender side only and does not require any TCP options. It aims at detecting spurious TCP retransmission timeouts only. A sender using this algorithm keeps track of sequence numbers of the incoming acknowledgments after it has transmitted the first unacknowledged packet triggered by a timeout. In this way, it can decide whether to send new packets or retransmit unacknowledged ones.

The design of the novel adaptive algorithm called *TCP Dynamic Adaptive Acknowledgment* (TCP-DAA) [Oli05a, Oli05b, TD(05)001, Oli07] is based on the following observations: TCP reliability requires that transmitted packets are acknowledged by the receiver side. However, if the receiver acknowledges every incoming data packet, then the probability of collisions between data and ACK packets increases considerably. Moreover, since the receiver must also contend for the medium by using RTS/CTS control frames, the overall overhead at the MAC layer, for transmitting ACKs, is not negligible. The problems associated with the ACK overhead can be mitigated, if the receiver merges several acknowledgments into a single ACK, which is possible due to the cumulative ACK scheme used in TCP(. By delaying the acknowledgment too excessively, the receiver may trigger a retransmission by timeout at the sender. Thus, the receiver has to be well adjusted in order to avoid such spurious retransmissions. Solutions like F-RTO or Eifel might be useful here. The main problem with both the standard DA and the LDA scheme is the fixed timeout interval (100 ms) for generating ACKs, since the packet inter-arrival at the receiver changes not only with the channel data rate, but also with the intensity of the traffic going through the network. TCP-DAA combines the idea of a higher number of delayed ACKs with the dynamic reaction proposed in [All99], i.e., reaction to packets that are either out-of-order or filling in a gap. Furthermore, TCP-DAA adjusts itself to the channel conditions in that it adaptively computes the timeout interval for the receiver on the basis of the incoming packet inter-arrival time. In this way, the receiver delays just enough to avoid spurious retransmissions by the sender and is able to adapt to different levels of delays imposed by the wireless channel. As shown in [Oli05b], TCP-DAA outperforms the standard DA and LDA in several scenarios. TCP-DAA decreases the number of duplicate ACKs for triggering a retransmission by the fast retransmit mechanism from three to two packets, which is in line with [All01]. Moreover, the regular retransmission timeout interval RTO is increased fivefold for compensating the maximum of four combined ACKs. After start-up and having no losses for four received data packets, the receiver replies with a single ACK. The delay management is performed through a delaying window at the receiver that limits the maximum number of ACKs to be delayed. Under normal conditions, the delaying window is initialized to one and increases gradually for each received data packet until it reaches four. The limit of four is imposed by the sender congestion window limit that is also set to four. When facing losses, however, the delaying window should be reduced in order to avoid further performance degradation. To detect a constrained channel, the receiver keeps a timer that is reset whenever it receives a data packet that is going to have its ACK delayed. Additionally, the receiver keeps track of the sequence numbers of incoming data packets. Whenever the receiver gets a packet that is either out-of-order or filling a gap in the receiver's buffer, or when its timer expires, it immediately sends an ACK to the sender and reduces the delaying window to the size of two packets (Fig. 5.14).

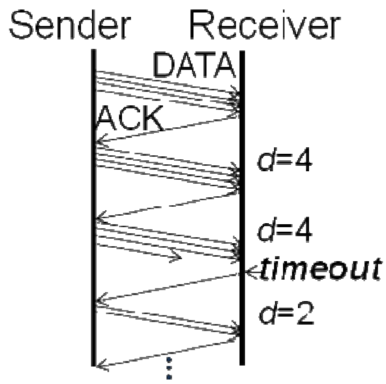

Fig. 5.14. TCP-DAA.

Cutting the delaying window down to one is more conservative and may be proper for highly noisy environments, where considerable improvements are hard to achieve. After a reduction in the delaying window, subsequent timely data packets trigger the delaying window growth toward the maximum size again. As mentioned above, we should handle the start-up phase differently. The delaying window growth is governed by

$$delaying\_window = \begin{cases} delaying\_window + \mu, \\ if \ \max\_delaying\_window = false \\ delaying\_window + 1, otherwise \end{cases}$$

An increase may be fixed at one (packet) or determined by the start-up speed factor μ, with 0 < μ < 1. By properly setting the μ parameter, the algorithm achieved better performance for short-lived flows [Oli05b, TD(05)001]. The reason for this factor is that there might be a shortage of ACKs at the sender, if we would increase the delaying window by 1 during the start-up phase leading to a situation where transmissions are triggered by timeout. The value of max_delaying_window defines whether the start-up phase is over. This occurs when the delaying window reaches its maximum value for the first time.

The mechanisms discussed above make TCP-DAA effective because they actively monitor the channel condition to use the scarce channel bandwidth efficiently. When the channel is facing really poor conditions, TCP-DAA should perform in general as effective as a standard TCP. An improved algorithm TCP-DAAp for very noisy conditions with packet error rates up to 10% is described in [Oli07].

5.5 Management and Channel Assignment in Wireless Mesh Networks

Wireless multi-radio multi-channel mesh networks have the potential to provide ubiquitous and ultra high-speed broadband access in urban and rural areas, to both fixed and mobile users, with low operation and management costs. This section investigates issues related to the configuration and management of wireless mesh networks, and in particular management and performance monitoring of a metropolitan scale wireless mesh network, handover based on application layer information, optimal route selection in heterogeneous infrastructure and ad-hoc networks, and finally self-management and channel assignment in wireless mesh networks.

We first start by discussing the experiences in the design, management and monitoring of a metropolitan scale multi-radio mesh test-bed. Key features in the design of the test-bed were to allow access to the mesh nodes even when the mesh node's main CPU crashes, and to support continuous online monitoring of link performance. To achieve both requirements, an independent management and monitoring network was implemented, and each node was equipped with a remote power switch.

Cellular-assisted management architecture is described, that allows independent routing of signaling and data messages over different communication technologies. The architecture allows nodes to learn the networking capabilities of the actual environment, and select the best end-to-end data path between nodes, which can consist of infrastructure-based and/or ad hoc links.

The next work deals with self-management of wireless mesh networks, and in particular self-healing and self-configuration in the case of configuration errors, faulty software updates, and the addition of new nodes. The architecture avoids the need for a second mesh network for management. Node configuration is achieved autonomously, with each node periodically checking with its neighbours for new configurations and software versions. To ensure network connectivity after reconfiguration, fall-back solutions and checks are implemented.

Finally, we discuss the problem of channel assignment in wireless mesh networks. Interference-aware channel assignment procedures seek to minimize some measure of interference, but such procedures typically do not take into account the traffic generated by each node, nor the routing paths. On the other hand, traffic aware channel assignment algorithms take into account link flow requirements or traffic load information from each mesh node. We describe a heuristic approach to channel assignment that involves the following three steps: 1) the maximum throughput for each link in the communication graph is estimated in the absence of interference, 2) channels are assigned to radios in an attempt to make the pre-computed flow rates estimated in the previous step schedulable, and d) the pre-computed flow rates are adjusted in order to ensure they are schedulable based on the channel assignment in the previous step.

5.5.1 Design, Management, and Monitoring of an Experimental Metropolitan Mesh Network

To investigate issues related to the management and monitoring of a multi-radio mesh network in an actual metropolitan environment, we have deployed an experimental multi-radio mesh network covering an area of approximately 60 km^2 in the city of Heraklion, Crete, Greece [Ang07, TD (07)038]. The long term objective is to use the mesh network as a metropolitan scale test-bed to investigate the performance of a multi-radio mesh network built from commodity components in 1 to 5 km city links; to evaluate channel assignment procedures for efficiently utilizing the wireless spectrum; to investigate MAC and network layer mechanisms for supporting performance guarantees, and to evaluate routing metrics for multi-radio, multi-channel, multi-rate mesh networks.

Other mesh and/or long-distance IEEE 802.11 networks include the 802.11b-based Digital Gangetic Plains rural area test-bed with 1-23 km links [Che06] the WiLDNet network with 50-100 km links [Pat06], the Roofnet network considering single-radio mesh nodes [Bic05], the Quail Ridge wireless mesh network [Wu07], and other city-wide mesh networks.

Each multi-radio mesh node consists of a mini-ITX board (EPIA SP 13000, 1.3 GHz C3 CPU, 512 MB DDR400 memory) and a 40 GB 2.5" HDD. A four slot mini PCI to PCI adapter (MikroTik RouterBOARD 14) holds four 802.11a/g mini PCI adapters (NL-5354 MP PLUS Aries 2, Atheros-based High Power Super A/G dual Band 802.11a/b/g). The mini-ITX runs Gentoo 2006 i686 Linux (2.6.18 kernel) with the MadWiFi driver version 0.9.2. Finally, nodes run the OLSR daemon version 0.4.10 [OLS08], which implements the OLSR protocol. One of the design requirements was to allow remote management, monitoring, and recovery of the mesh nodes, even in situations when a mesh node's mini-ITX board crashes or its wireless interfaces are down. To address this requirement we added to each mesh node an additional 802.11a client (Fig. 5.15 a), which connects to a management and monitoring network that operates in parallel to the experimental mesh network. Additionally, to enable remote recovery of the mesh node's mini-ITX board we added an intelligent remote power switch (Dataprobe iBoot, Fig. 5.15 b).

This allows the power to be switched off and on through a web interface, but also supports timed power reboot based on the results from the power switch pinging other devices (the mini-ITX board or some remote device).

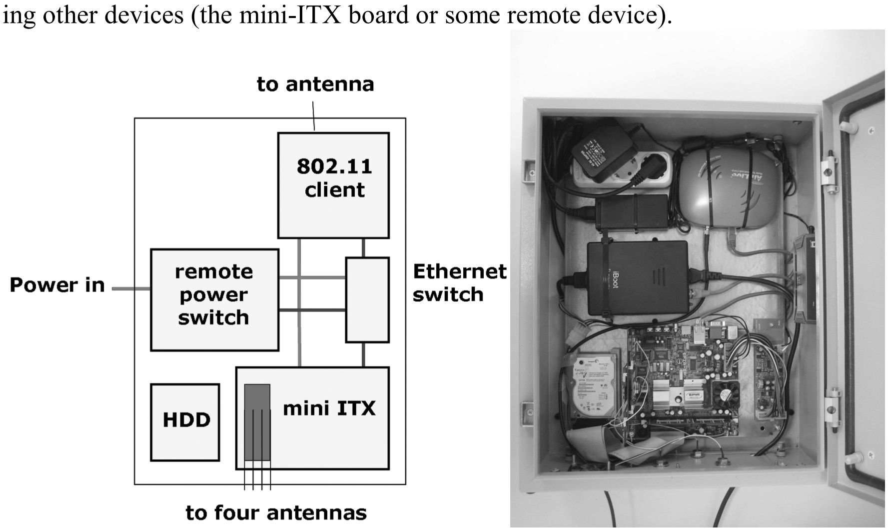

Fig. 5.15. Multi-radio mesh node a) component layout b) actual node.

The metropolitan mesh network covers an area of approximately 60 km^2 and currently contains 14 nodes, among which six are core mesh nodes, whose design was discussed in the previous section. The distance and antennas used for the links between core mesh nodes are shown in Table 5.2. The mesh test-bed is connected to a fixed network through two nodes (FORTH and UoC).

Link	Distance (km)	Antennas
Ekab-Lygerakis	5.0	29 dBi grid – 21 dBi panel
Ekab-Tsakalidis	4.9	29 dBi grid – 21 dBi panel
Lygerakis-Tsakalidis	2.0	21 dBi – 19 dBi panel

UoC-Lygerakis	1.6	21 dBi – 19 dBi panel
UoC-Tsakalidis	3.3	21 dBi – 19 dBi panel

Table 5.2. Links between core mesh nodes.

Table 5.3 shows typical values we have observed for the *Signal-to-Noise Ratio* (SNR) and transmission rate of the core links in an interval of 12 hours. These results show that the links are asymmetric and the link quality varies. Moreover, the variation of the link quality is different for different links. We have developed a monitoring tool that consists of set of scripts that continuously monitor important performance metrics for all links between core nodes, and make them available through a web server using the Round Robin Database Tool (RRDTool), Fig. 5.16. The metrics include the SNR, transmission rate, MAC and physical layer errors [Her08]. Additionally, the monitoring tool can perform active measurements of the round-trip delay and the throughput. The measured values can be displayed at different time granularities, which include daily, weekly, monthly, and yearly. In addition to periodic measurements, through a different interface we can perform on demand measurements of various metrics, such as two-way delay and throughput.

Link	SNR (min, max, avg)	Rate (min, max, avg [Mbps])
Ekab-Tsakalidis	18,29,21	6,54,24
Tsakalidis-Ekab	13,23,18	6,48,24
Lygerakis-Tsakalidis	13,18,15	6,18,7
Tsakalidis-Lygerakis	14,26,19	6,36,8
Lygerakis-UoC	13,19,17	12,54,44
UoC-Lygerakis	22,28,25	12,54,45
Tsakalidis-UoC	20,25,22	12,24,18
UoC-Tsakalidis	20,29,22	12,36,24

Table 5.3. Link SNR and transmission rate.

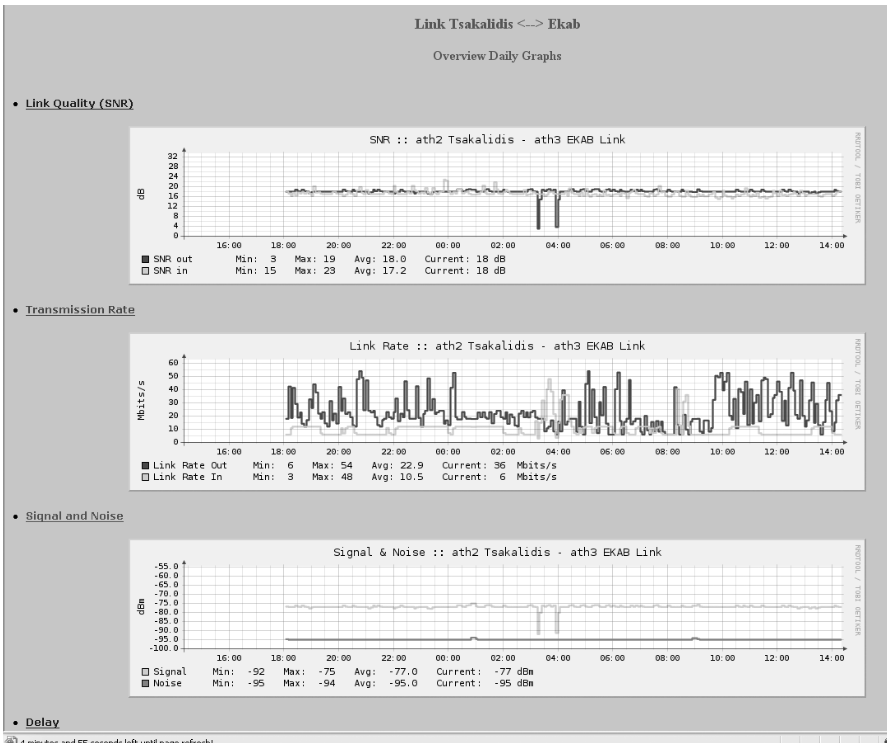

Fig. 5.16. Snapshot of mesh network monitoring tool. The top graph gives the link quality (SNR) and the bottom the transmission rates for the two directions of one link.

5.5.2 Cellular-Assisted Management

A single wireless network technology cannot support the different and often changing application requirements. While cellular networks offer large coverage with rather low speeds, wireless LAN based networks offer the contrary. To optimally meet the applications need, different communication technologies have to be integrated. The motivation for this is further discussed in [Gus03], where the vision of being “always best connected” is introduced. Dynamically assigning the most appropriate networking resources to each node depending on its actual needs and capabilities may increase user satisfaction.

The middleware presented in [Cal04, Bha04] is able to gather context information of the application layer to optimize handover decisions. The concept of MIRAI [Ino04] addresses this problem by defining dynamically one channel to be used for signalling information and negotiation of handover decisions. The authors propose an agent based platform that provides location-based information on available access networks through a so-called basic access signalling, which is assumed to have a larger coverage than all other access networks. This concept is very beneficial, especially if the basic access signalling channel is a low power channel. Such a signalling channel does not have to provide high data rates. MIRAI focuses on infrastructure-based access networks only. However, communicating nodes can come close enough to establish direct links based on short range and infrastructure-less communication technologies. These links might provide larger data rates than infrastructure-based networks. Especially in scenarios, where nodes are moving in groups and the probability of being within the range of direct communication is high, the average data rates can be considerably increased. This is the case for public transportation, battlefield scenarios, but also in smaller campus networks.

Depending on the available networks, the optimal end-to-end data path between nodes can consist of infrastructure-based and/or ad-hoc links. To enable such heterogeneous networking and being always best connected, a novel protocol architecture called *Cellular Assisted Heterogeneous Networking* (CAHN) has been implemented and presented in [Dan03, Dan06a, Dan05, Dan06b, TD(06)037]. CAHN allows independent routing of signalling and data messages over different communication technologies. For example, signalling traffic between two end systems can be transferred reliably and securely via cellular networks, while data are transferred via high-speed direct WLAN links, cf. Fig. 5.17.

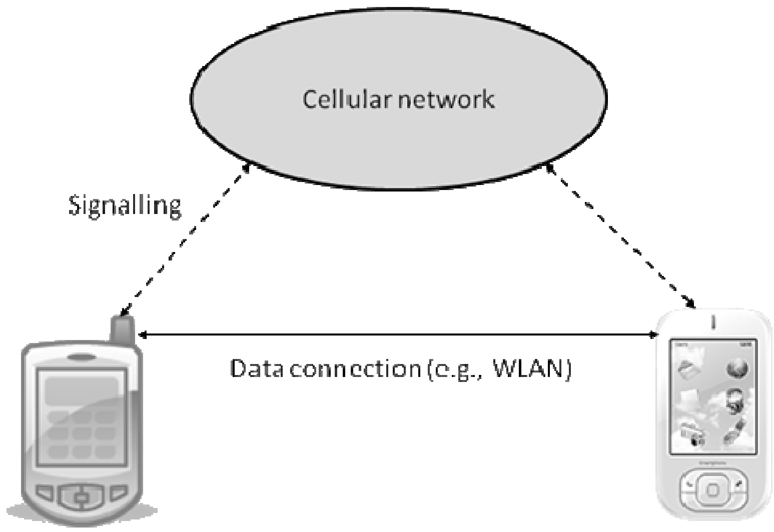

Fig. 5.17. Cellular Assisted Heterogeneous Networking.

During data session establishment nodes can learn about the networking capabilities and actual environment, such as the currently available networks at the peer's location. In particular, the use of existing cellular mobile networks for the primary signalling plane has several advantages. The well established and power optimized location, paging, security, and mobility management services can implicitly be shared for other communication technologies and networks, which are lacking security, power saving functionality, and the possibility to establish direct links.

The security relation between the cellular subscriber and the network operator together with the roaming relations between the operators offer a safe communication channel, which can be used to exchange the signalling protocol messages. Especially, if the establishment of infrastructure-less communication channels is considered, the reliable exchange of configuration and security related parameters is absolutely mandatory to securely bootstrap such channels. Having established a safe initial communication between the communicating entities, all other parameters required to establish further communication channels can securely be negotiated. CAHN provides the missing part to securely extend the scope of heterogeneous networking also to ad-hoc links.

The ability to reach any node through the low power cellular network allows switching power demanding broadband data channels and interfaces to sleep mode, if no data session is going on. Waking up data channel interfaces only on-demand can considerably increase the power efficiency of the mobile devices. This on-demand mode also has the potential to increase the efficiency of allocated networking resources because the nodes are only attached to the high-data rate network when data has to be transferred.

The ability to securely bootstrap infrastructure-less communication between nodes enables the system to support direct node-to-node links when evaluating the most appropriate connection. Based on exchanged signalling messages, two wireless nodes can detect that they can reach other and then establish a direct (high) bandwidth connection.

CAHN has been evaluated by simulations [Dan06b, TD(06)037]. The results show that CAHN enables the efficient use of available network resources. In certain scenarios the network load can be decreased by about 50%, which results in lower session drop and blocking rates. Furthermore, the throughput can in almost be doubled in some scenarios, depending on the number of nodes. Furthermore, in the investigated scenarios approximately 20% of energy could be saved by on-demand mode and another 20% by using fast direct links that allow shorter data transmissions. These results motivate the tight collaboration of various communication networks, including both, infrastructure-based and ad-hoc network technologies.

5.5.3 Self-Management

Wireless mesh networks are evolving to an important access technology for broadband services. They have various aspects of self-management. The network should be able to deal with link or node failures by selection of alternative routes, what is usually done by a routing protocol. In case of a multi-channel network the node should automatically select an appropriate communication channel. This is done by channel allocation mechanisms inside the MAC layer.

Other important self-management issues are self-healing and self-configuration in case of configuration errors, faulty software updates, and integration of new nodes. The erroneous configurations and software updates should be automatically recovered to guarantee best possible network connectivity. New nodes are automatically integrated in the network. A newly integrated node adopts parameters and the current software version used in the network automatically.

There exist multiple deployments of WMNs related to research, e.g., Massachusetts Institute of Technology (MIT) Roofnet [Bic05, Agu04] Orbit project [Ray05], Microsoft Research [Dra04a, Dra04b]. Further, several cities are establishing metropolitan area wireless mesh networks [Kar03]. All these existing wireless mesh networks share one important characteristic – they are distributed over large areas. Physical access to certain nodes can be difficult and time-consuming. For example organizational and technical issues may complicate access to nodes on rooftops. During the lifetime of a wireless mesh network it is necessary to process reconfigurations and software updates to match new requirements or simply to fix errors. The reconfiguration and update process is a possible point of failure in each network. Configuration errors or faulty software updates may disconnect the network. Changes of radio communication parameters can affect the physical topology of the network as well as cut off nodes from the network. Without an automated reconfiguration, which supports the user in case of defective configuration or errors, physical access to specific nodes may be required.

As experimental research is an important step in the design of wireless research, safe reconfiguration and update mechanisms are important and save time. UCSB's ATMA [Ho04] framework offers a management framework for experimental wireless networks. However, its concept of deploying a second additional wireless mesh network for management is not applicable for productive networks. Safe reconfiguration and update mechanisms are important for experimental and productive WMNs. The developed secure management architecture for WMNs [Sta07b, TD(07)003] provides a management solution that works without any additional infrastructure, e.g., wired or wireless backhaul networks. It guarantees the availability of the network in case of configuration errors or erroneous software updates and therefore avoids costly on-site reconfigurations. It provides fallback solutions for configuration errors and kernel panics.

Moreover, network and node configurations as well as software updates are distributed autonomously. Each node periodically checks whether its neighbours have newer configuration and software versions. If there are updates available, the node downloads them to its exchange directory. The other neighbours will recognize these new versions and download them as well. The updates are propagated inside the network. The new configurations and updates are activated after a predefined time.

If a node was down during the distribution of the updates it fetches them as soon as it is up again. If the update contains changes in any critical parameters like wireless channel, so that the awaken node has no longer any connection to its former neighbours, it will fall back to its initial configuration and try to join the network as a new node.

In order to guarantee network connectivity after reconfiguration, fallback solutions and checks have been implemented. If for example transmission power is decreased and afterwards the node has no connection anymore, the transmission power is, automatically, step-wise increased until the former connectivity is reached. Other disruptive changes are considered as well.

The architecture further provides a safe way to upgrade the node's operating system. After integrity checks have been performed, the updated kernel and file system are loaded into the update storage and the node is instructed to load the operating system from there. On the next reboot the node automatically loads the operating system from the default storage. If the software update succeeds and the node is up with the new operating system, the update can be made permanent by copying the updates to the default storage. In case there occurs any problem while booting the operating system, e.g., a kernel panic, the node will be rebooted and load the old operating system version.

The architecture has been implemented on the *Wireless Router Application Platform* (WRAP.2E) from PCEngines [Pce08] using an embedded Linux operating system. The configuration and update distribution has been built by enhancing cfengine [Bur05] with custom scripts for the fail-over cases.

5.5.4 Channel Assignment

We focus on wireless mesh networks in which mesh routers are equipped with multiple radios. The design of multi-radio nodes is becoming practical given the availability of cost-effective wireless network interface cards. Also, the availability of multiple radios enables a node to simultaneously transmit/receive on different channels and thus to avoid the need for channel switching. Recently, researchers showed the substantial increase in network throughput with respect to single-radio WMNs [Ran05, Ran04].

Multi-radio WMNs evidently cause the channel assignment problem, i.e., the problem to assign one of the available channels to each radio of every mesh router. We assume that the number of available radio interfaces per node is smaller than the number of available channels. Since transmissions over the same channel interfere with each other and cannot take place simultaneously, channels should be assigned to minimize interference. Moreover, a channel assignment strategy is constrained by the need to preserve connectivity, as two neighbour nodes can communicate with each other only if their radio interfaces share a common channel. We classify a channel assignment algorithm as *interference-aware*, if it aims to minimize some measure of interference.

Interference-aware channel assignment algorithms do not consider the amount of traffic generated at each mesh router and how it is routed through the mesh network. Solving such a routing problem associates each wireless link with a flow rate representing the amount of traffic that is expected to cross that wireless link. The channel assignment algorithm may exploit the knowledge of the computed flow rates to assign channels in such a way that the computed flow rates can be actually achieved. Indeed, the bandwidth available on a wireless link is determined by the channel assignment, because due to interference the channel capacity must be shared among all the links in a neighbourhood using the same channel. However, the routing problem can only be solved if the bandwidth available on each link is known. Channel assignment and routing are not independent problems and must be jointly solved. We call channel assignment algorithms as *traffic-aware* if they have been developed in the context of a joint channel assignment and routing problem. Unfortunately, the joint channel assignment and routing problem is NP-complete, thus only approximate solutions have been proposed. An approximate algorithm for the joint channel assignment and routing problem typically consists of the following three steps:

1. Determine the pre-computed flow rates: A pre-computed flow rate is determined for every link based on the given optimization objective.
2. Determine the channel assignment: Channels are assigned to radios in the attempt to make the pre-computed flow rates resulting from the previous step schedulable, i.e., actually achievable considering the interference among transmissions over the same channel.
3. Adjust the pre-computed flow rates: The pre-computed flow rates resulting from the first step may be adjusted in order to obtain a set of schedulable flow rates given the computed channel assignment.

A heuristic algorithm [Ava07, TD(07)052] has been developed for the joint channel assignment and routing problem with the objective to maximize the overall network throughput. The proposal follows the steps listed above and is described in the following.

5.5.4.1 Determining Pre-computed Flow Rates

We propose a method to compute the flow rates having the objective to maximize the achievable network throughput. We underline that the proposed method does not need the knowledge of the (expected) traffic demands. Since the flow rates are computed when a channel assignment is not known yet, the wireless mesh network is represented by the potential communication graph. Thus, to determine the flow rates used as input to the channel assignment algorithm, we may compute the maximum achievable throughput of the potential communication graph under the protocol interference model. However, the problem to determine such throughput is NP-complete. Also, the maximum throughput computed in such a way is smaller than the actual capacity of the WMN, as simultaneous transmissions can take place over potentially interfering links that have been assigned different channels. Instead, if we compute the maximum throughput of the potential communication graph assuming that interference does not arise, then we clearly overestimate the capacity of the WMN. However, it is not important to exactly evaluate the maximum throughput, as the ultimate purpose of the flow rates is to give an indication about which links are most critical in carrying traffic. Since the effect of the interference is basically to prevent simultaneous transmissions over interfering links, we decided to base the flow rate values on the computation of the maximum throughput of the potential communication graph in the absence of interference.

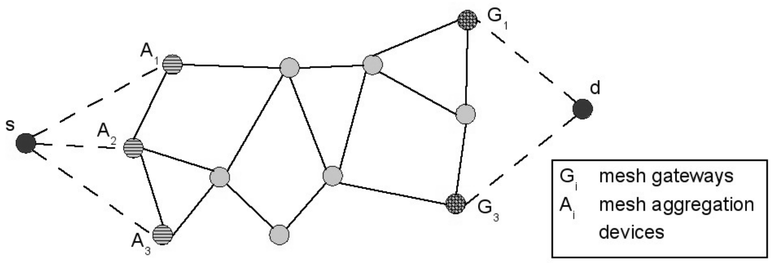

Fig. 5.18. Reducing the potential communication graph to a single-source single-sink topology.

Even disregarding the interference, the problem of finding the capacity of the potential communication graph, being an instance of a maximum multi-commodity flow problem, is NP-complete. However, we can turn to the simpler single commodity flow problem by means of the following key observation: in a WMN, mesh routers have to forward packets towards the wired network, regardless of which particular gateway is used. Mesh aggregation devices collecting user traffic do not have to forward each packet to a specific mesh gateway, but can direct it to *any* of the mesh gateways. The problem of maximizing the flow from any source to any sink is a single commodity flow problem with multiple sources and sinks. There is a standard approach to reduce this more general version to the case with a single source and a single sink, which consists in adding two extra nodes connected respectively to the mesh aggregation devices and the mesh gateways by links of infinite capacity (cf. Fig 5.18).

The maximum throughput on the graph including the extra nodes, in the absence of interference, can then be computed as the maximum network flow between the two extra nodes. The maximum flow computation associates each link with the flow it must carry. We use this amount of flow as the link flow rate to be used by the channel assignment algorithm.

5.5.4.2 Channel Assignment Algorithm

A sufficient condition for a set of pre-computed flow rates to be schedulable given a channel assignment is derived in [Ali06]. The goal of the channel assignment algorithm is therefore to assign channels such that the sufficient condition for schedulability is satisfied for the set of pre-computed flow rates determined by the first stage of the approximate solution to the joint channel assignment and routing problem.

The developed channel assignment algorithm is named MCAR (Maxflow-based Channel Assignment and Routing). The computed channel assignment is subject to some constraints. For instance, it must ensure the connectivity of the induced graph, as two nodes can communicate with each other only if their radios share a common channel. Also, in case channels are assigned to links, it must ensure that the number of channels assigned to a node does not exceed the number of available radios.

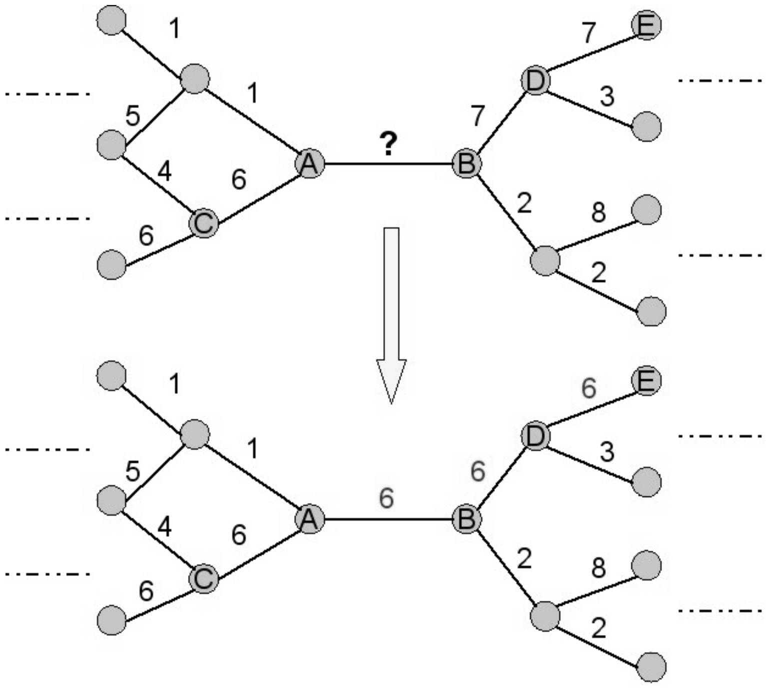

Fig. 5.19. Recursive replacements of channels.

Other solutions [Ran05, Tan05] may require recursive adjustments to previous channel assignments (cf. Fig. 5.19). The proposed channel assignment algorithms are based on the same concept: links are visited in some particular order and a common channel is assigned to the interfaces of both end nodes. If all interfaces of the end nodes on a link have already an assigned channel and they do not share any common channel, then it is necessary to replace one of these channel assignments. Due to the limited number of radio interfaces per node, this replacement may trigger a chain reaction and must be performed recursively.

Our algorithm, instead, avoids this problem while ensuring connectivity and feasibility by splitting the channel assignment solution in two stages. In the first stage, links are grouped based on the flows they carry (Fig. 5.20). A group may contain links from many different nodes. For each node, the first stage assures that the number of different groups assigned to its links does not exceed the number of radio interfaces. The second stage selects a channel for each group and assigns the selected channel to all links of the group. An attempt is made to assign different channels to groups containing interfering links.

```
for each node u
    if the feasibility constraint is violated
        then repeat
                merge two groups associated with links of u
            until the feasibility constraint is satisfied
    h=1; t=# of u's neighbors
    while h<=t
        do if the element pointed to by h is unassociated
                then assign it to a (new) group
            while h<t and the group does not exceed its share
                do assign the group of the element h to t
                    increment t
            increment h
```

Fig. 5.20. Pseudo-code of the link-group binding stage.

Our approach ensures connectivity by assigning a common channel at both end nodes of every link. After the second stage of our algorithm, the number of channels assigned to a node does not exceed the number of radio interfaces, because the first stage returns a number of groups per node not greater than the number of radio interfaces. Thus, the constraint on the number of radio interfaces per node is obeyed and no replacement of previous channel assignments is required.

Finally, we note that splitting the algorithm in two stages allows selecting channels based on information on the whole network. Indeed, the first stage partitions the set of links into groups enabling the second stage to predict the impact of selecting a channel for a group on the whole network.

5.5.4.3 Adjusting Pre-computed Flow Rates

The problem to find a channel assignment which makes a given set of pre-computed flow rates schedulable is also NP-complete. Thus, the proposed channel assignment algorithm is just a heuristic and does not ensure that the set of pre-computed flow rates is schedulable given the computed channel assignment. Hence, the last stage of our approximate solution to the joint channel assignment and routing algorithm applies a scaling factor to the pre-computed flow rates in order to obtain a schedulable set of flow rates.

5.6 Power Saving in Wireless Multi-hop Networks

Wireless sensor networks are another popular example of wireless multi-hop networks. WSNs have reached high interest in the research community but also many practical application examples and scenarios have been developed such as environmental monitoring. Energy consumption is the main concern in wireless sensor networks. In particular, communication (transmitting and receiving) is rather expensive in terms of energy consumption. Therefore, wireless sensor network communication protocols try to put the transceiver or even the whole node into a sleep state.

This section first discusses power saving concepts and proposes a scheme based on unsynchronized, periodic sleeping of sensor nodes. It further describes how an existing MAC protocol can be further improved by applying the proposed scheme.

For performance evaluation of wireless sensor network protocols and mechanisms, accurate modelling of energy consumption is extremely important. The importance of the right choice for energy modelling is further discussed in this section. The selected model must be as close as possible to the target hardware to be used for implementation. Otherwise, wrong protocol design choices might be the consequence.

5.6.1 Unsynchronized Power Saving in Wireless Multi-hop and Sensor Networks

In wireless mobile ad-hoc and sensor networks, efficient power saving mechanisms can drastically increase network lifetime. However, reasonable connectivity properties are nevertheless necessary for proper operation of wireless networks. Many power saving mechanisms introduce central or distributed synchronization and periodic switching between sleep and awake states. Since synchronization causes overhead, several variants of unsynchronized power saving mechanisms have been proposed.

In the IEEE 802.11 power saving mode all nodes ideally wake up at the beginning of a beacon interval, remain awake during the *Asynchronous Traffic Indication Map* (ATIM) window to exchange traffic announcements in case of pending traffic, and fall into sleep again, if there is none. A quorum based approach [Tse02] proposes that each node organizes its beacon intervals into groups of n consecutive intervals. *Multi Hop TIM* (MTIM) messages are sent by the nodes at the beginning of an interval. A group of n intervals is organized as $\sqrt{n} * \sqrt{n}$ array with $\sqrt{n}$ columns and $\sqrt{n}$ rows. Each node selects a column and a row and will be awake during the selected $2\sqrt{n}-1$ intervals. Even if two hosts are not synchronized (i.e. they select the starting point of an interval in an asynchronous way), a node will receive a MTIM message not later than after n intervals. However, the n may be quite large and may result in a very high delay. The wake ratio of $(2\sqrt{n} - 1)/n$ is rather high (e.g., n = 25: wake ratio = 9/25 = 36%). Another approach [Fee02] proposes that nodes are active for 50% + ε of the time. ATIM messages are sent at the beginning and at the end of each wake interval. This ensures that wake intervals of any two nodes overlap by at least ε allowing exchanging traffic announcements in a predictable way.

The mechanism proposed in [Bra05] and analyzed in a static multi-hop wireless ad hoc network environment [Hur06, TD(06)040] defines one wake and two sleep periods during one basic cycle duration T, as depicted in Fig. 5.21.

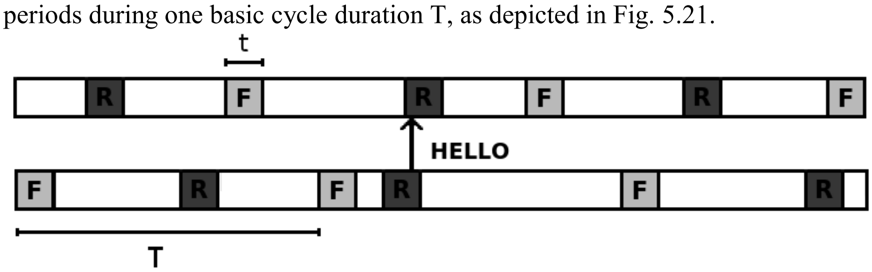

Fig. 5.21. Unsynchronized with Fixed and Random Intervals.

Nodes strictly alternate between a fixed (F) and a random wake period (R). Each of the wake periods shall be of the same duration t. The start of the random wake period is uniformly distributed between the end of a fixed wake period and the start of the next one. All nodes operate with the same basic cycle duration T, although remaining unsynchronized, and switch between wake and sleep states with their individual wake-up pattern. Nodes operate with the same wake ratio W = 2t/T. If two nodes have intersecting fixed intervals they can mutually learn about their presence and contact each other based on the knowledge of the periodically occurring fixed intervals. However, if there is no intersection between the fixed wake periods of two nodes, they may never learn about their presence. This motivates the choice for the random wake period, which ensures that two nodes with disjoint wake-up pattern will sooner or later be awake at the same time and therefore be able to exchange announcements about their fixed wake periods. As examined in [Hur06, TD(06)040], this wake-up scheme can easily be applied to multi-hop wireless ad hoc networks and reactive routing schemes.

A node intending to broadcast a message can figure out the best instant to forward the message. The best instant when the largest subset of the neighbouring nodes is awake during the next basic cycle T. Fig. 5.22 depicts this approach. The node calculates the best instant for broadcasting a message to be within Δx. This calculation is based on the gained knowledge about the neighbours' fixed wake intervals. The aim of the broadcast is not to reach all neighbours, but only the largest possible number of neighbours. Similar as in probabilistic broadcasting schemes not every neighbour may be reached, but the scheme further alleviates the broadcast storm problem. Using this technique, and taking the two best instants for rebroadcasting a route request of an on-demand routing protocol, the success ratio reached 97% even for the very low wake ratio of 4% [Hur06, TD(06)040]. The scheme has been further developed in [Hur07, TD(07)034] by integrating it with the broadcast mechanism of WiseMAC [ElH04], a MAC protocol for wireless sensor networks.

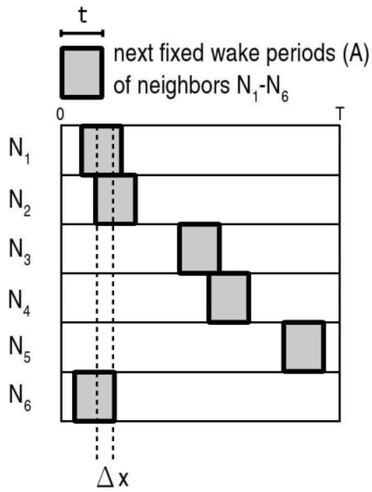

Fig. 5.22. Announced fixed wake periods in an intersection table.

Moreover, the scheme has been further improved by replacing random intervals by moving intervals. Fig. 5.23 shows the possible combinations of fixed and moving (backwards or forwards) intervals. The results proved that moving intervals are feasible and even perform slightly better than a combination of fixed/random intervals. In this case we propose two alternating intervals: The first one is always moving forward, while the second one is always moving backwards, see case IV in Fig. 5.23.

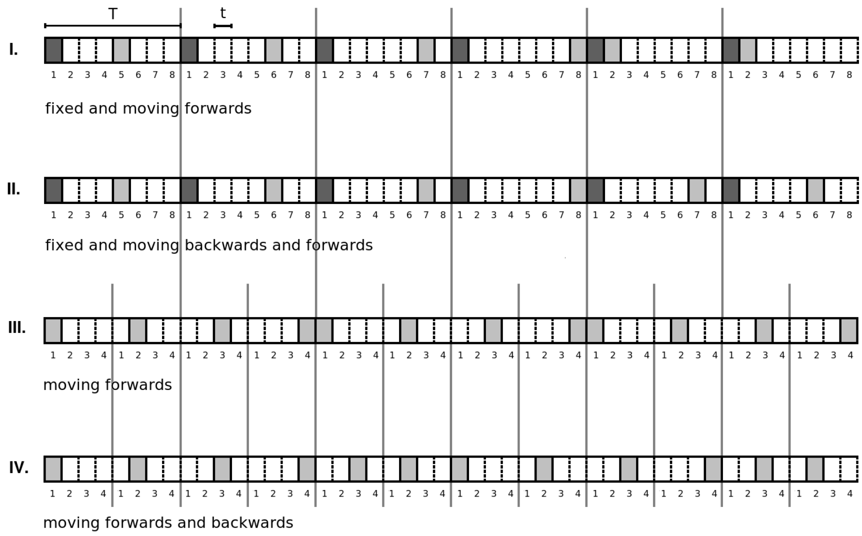

Fig. 5.23. Fixed and moving intervals.

3.6.1.1 On Sensor Network Energy Consumption Modelling

To maximize the network survivability and hence the quality and quantity of the assembled data, the wireless sensor network has thus to be designed in the most energy efficient manner as possible. Analytical approaches for predicting the energy consumption in wireless sensor networks have thus to find adequate abstractions for the sensor node behaviour.

The most widely used energy model for analyzing radio operations in sensor networks, e.g., used by [Cha04, Chi04, Mel05], has been proposed by [Hei02]. The amount of energy necessary for a transmission over a distance d is modelled as the sum of a constant and the required transmission power. The latter scales with a power of d, accounting for the path loss, the first represents the energy consumptions of the transmitter and receiver electronics. [Wan06] introduced a more hardware oriented approach and consider the drain efficiency of the power amplifier of the sensor node radio chip, i.e. the ratio of transmission output power and the consumed DC input power, which is smaller than 10% and increasing with the output power for contemporary hardware.

We use these two different metrics to identify the critical communication cost parameters which have to be modelled with special care. For this purpose, we establish a unifying framework which allows us to compare the different models and express the energy required for transmitting one bit over a distance d as

$$E(d) = E_0 + \frac{T(c,d)}{\eta(c,d)} + E_{rx}\,. \tag{5.5}$$

In the formula, E_0 and E_{rx} represent the constant power consumptions in the transmitter's and receiver's signal processing and front-end circuits. T(c,d), the necessary transmission output power, is increasing with distance and depends on the channel characteristics. These are represented by c and capture the radio propagation model and the receiver sensitivity. While this notation for the transmission power seems somewhat artificial, it allows comparing different channel modelling approaches. η(c,d) denotes the drain efficiency of the transmitter power amplifier, which is the ratio of transmission output power to DC input power. For most transceivers it is increasing with the output power, depends thus also on c and d.

We determine the transmission costs according to a *Theoretical Model* (TM) which is based on [Hei02] and a *Hardware oriented Model* (HM), which is inspired by [Wan06]. In Fig. 5.24 we illustrate the resulting costs for transmitting a bit, if all parameters are set according to the pure model (labelled "HM" and "TM" respectively). We furthermore examine what influence it has, if one parameter is set according to the other metric. "TM, c HM" stands for the resulting transmission costs, if all parameters are set according to TM but the channel model is taken from HM.

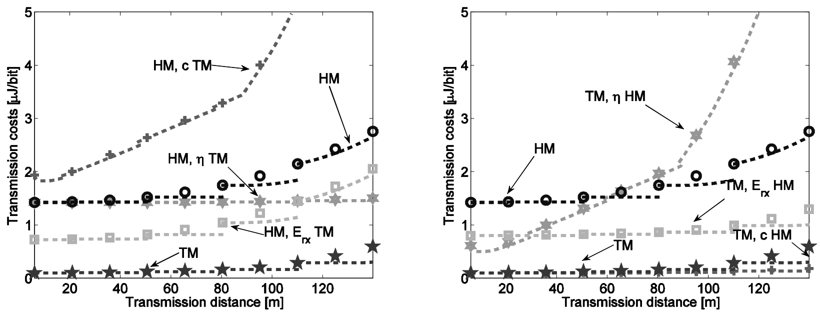

Fig. 5.24. Transmission costs resulting from different energy consumption models.

In [Sta07a, TD(07)040] we analyze these differences in detail and use the different transmission costs to establish energy efficient routing topologies. We found that the resulting routing topologies vary strongly. Especially the abstractions of channel characteristics and the mapping of transmission output power to required DC input power influence the analysis of transmission energy consumptions heavily. We furthermore estimated the daily radio related energy consumptions of a sensor node in a given topology and saw that these estimations vary very strongly with the efficiency of the MAC protocol, i.e., the number of undesired transmissions a sensor node is forced to overhear. This often neglected factor is also responsible for a possibly wrong estimation of energy consumptions: if an ideal MAC protocol is assumed, i.e. no node is forced to overhear foreign transmissions, topologies that favour a few long hops over several short ones, are rated to be by far more energy efficient than topologies which contain more short hops. This statement does not hold any more if the effect of overhearing is considered, as for larger transmission ranges, more energy may be consumed for discarding unwanted messages. Thus, for any statement concerning per node energy consumptions, the consideration of the MAC layer and the structure of the routing topology are essential.

All in all, our findings illustrate that energy models used for the design and analysis of real world sensor network deployments have to be chosen with care and with respect of the used hardware, as a bad design choice may lead to incorrect routing decisions.

5.7 Context and Service Discovery

For multi-hop wireless networks, and especially for ad-hoc networks, the discovery of the environment of a device is a very important capability that makes the device useful. Contrary to many wired networks, a multi-hop wireless ad-hoc network will not be installed and configured by a (non-)professional network administrator. Rather, devices move into each other's reach, and engage in communication spontaneously. To enable sensible joint communication behaviour of devices, it is of paramount importance that devices can discover context information and services available in their communication neighbourhood automatically. AHOY [Liu06, TD(06)033, Liu07b, Haa07, Goe07, TD(06)020, Liu07a, Goe06] an approach to perform context or service discovery in ad-hoc networks based on the use of attenuated Bloom filters is described in this section. A Bloom filter is an efficient space-saving data structure to represent context or service presence information. Attenuated Bloom filters are used to advertise the availability of context information or services multiple hops away, and to guide queries to discover them. In the remainder of this section, first the AHOY discovery protocol including its validation is described. Thereafter, alternative discovery protocols are discussed.

5.7.1 Service Discovery Using Attenuated Bloom Filters

A Bloom code can represent a set of services of context information types [Blo70]. Each service will be coded by using b independent hash functions over the range <0, w-1>, where w is the width of the filter. The default value for each bit in the Bloom code is 0. The bits of positions associated with the hashes will be set to 1. Our approach uses attenuated Bloom filters, each of which consists of a few layers of basic Bloom filters. The first layer of the filter contains the services for the current node, while the second layer contains the services for the nodes one hop away, and so on. In other words, a node can find the services *i* hops away in the i^{th} layer. When querying for a certain type of service, the same hash functions are performed. If all positions in a Bloom filter indicated by one of the hashes contain a 1, the presence of the queried service is likely (but not guaranteed). Otherwise the service is not present. The use of these attenuated Bloom filters introduces the possibility of having false positives, which will be resolved during a later stage of the discovery process. By using attenuated Bloom filters consisting of multiple layers, services at more than one hop distance can be discovered, while avoiding saturation of the Bloom filter by attenuating (shifting out) bits set by sources further away.

For example, let us assume a 6-bit Bloom filter with b equal to 2 (Fig. 5.25). If a printer service is hashed into {1, 3} and temperature information is mapped into {2, 5}, we obtain the filter shown in Fig. 1. The filter will give a positive answer to queries for printer or temperature information. It definitely does not contain a camera service, which is hashed into {0, 3}. Nodes may also think a loudspeaker service {1, 5} is contained in this filter, but actually it is not. This situation is referred to as false positive.

0	1	2	3	4	5
0	1	1	1	0	1

Fig. 5.25. A simple 6-bit Bloom filter.

Context aggregation can be simply implemented by attenuated Bloom filters. When a node A receives incoming Bloom filters $filter_B$ and $filter_C$ from neighbours B and C respectively, it shifts all the contents of $filter_B$ and $filter_C$ one layer down and discards the last layer. An OR operation will be done to those new filters, $filter_B'$ and $filter_C'$, and the first layer will filled by the A's own Bloom filter, $filter_A$, representing the services A has to offer. Consequently, the attenuated Bloom filter of node A is constructed such that the first layer represents the local information from node A; the second layer contains the information from neighbour B and C; the third layer covers the information two hops away which can be reached via B or C. Fig. 5.26 shows the process of context aggregation in a node.

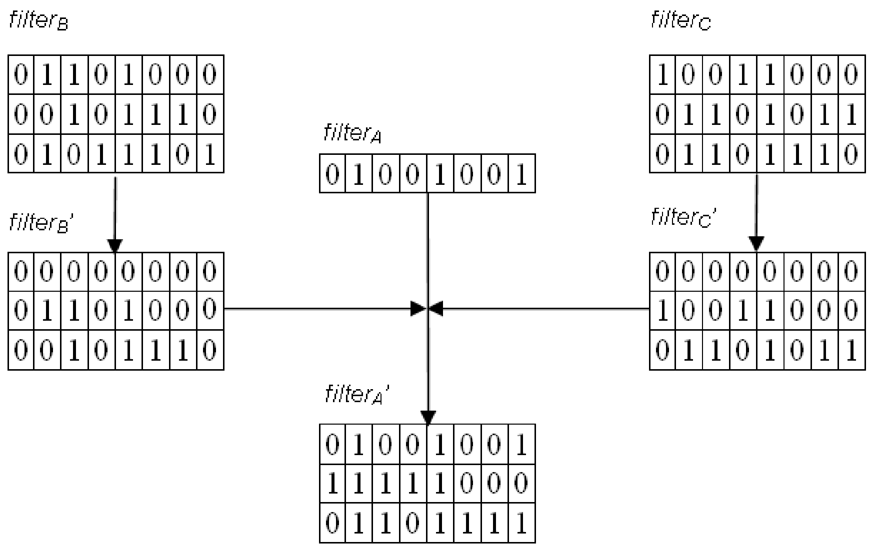

Fig. 5.26. Aggregation of attenuated Bloom filters.

5.7.1.1 Protocol Description

When designing a discovery protocol for multi-hop ad-hoc networks one can think of two extreme approaches. The first one is complete flooding of service advertisements, so that all nodes have complete knowledge about available services. The second one is complete flooding of a query message, whereas no advertisements are used, so that all nodes in the network are queried for the availability of a service once it is needed. Each of these approaches can be optimal in a very specific situation, i.e. when there are only very few services to be advertised respectively queried. Attenuated Bloom filters can be used to design a compromise between these two extremes that is in most practical situations much more efficient than the two extremes, while still maintaining the fully distributed operation of the protocol that is advantageous in ad-hoc networks. In the AHOY protocol, nodes in the ad-hoc network broadcast an attenuated Bloom filter summarizing the available services up-to d hops away to their neighbours. The neighbours store the Bloom filters and use them to update and afterwards broadcast their own attenuated Bloom filters. Thus, all neighbours obtain information about services available in their neighbourhood in an iterative way.

An attenuated Bloom filter is only broadcasted if it has been changed compared to the last broadcast. In order to detect new neighbours, and out-of-date information, nodes periodically broadcast keep-alive messages indicating the address of the sender and a sequence number of the latest advertised attenuated Bloom filter.

Nodes receiving this keep-alive message that do not have the latest advertisement from the sender will specifically request for it.

When an application in a node is requesting for a certain service, the service name will be hashed, and the hash result will be compared with all layers in all stored Bloom filters. In this way, the node will learn where in the network, i.e., in the direction of which neighbour, the service can be found, and how many hops away it can be found. A query message will be constructed and sent to the neighbour(s), for which a match with the Bloom filter occurred. Upon receipt of the query message, the neighbours will repeat this procedure, until the query message arrives at a node that has provides the service. This node will send a response message to the originator of the query. Responses can be routed back to the query originated using an external routing protocol, or using path information stored in intermediate nodes during the querying, i.e., the discovery protocol can be tightly integrated with routing. Note that different strategies can be followed concerning the forwarding of queries in case of multiple matches. All matches can be explored in parallel, using a query id to avoid loops or duplication of queries. Alternatively, additional matches can be explored sequentially, where the path with the lowest hop-count is explored first.

5.7.1.2 Protocol Analysis and Validation

AHOY has been analyzed and validated in a number of ways. In [Liu06, TD(06)033, Liu07b], an analytical model is presented in which the load of advertisements and excess queries due to false positives is determined for a stationary ad-hoc network. Fig. 5.27 shows the results obtained if the ad-hoc nodes are positioned in a grid structure, so that each node has exactly four neighbours, each advertising one service plus the services of its neighbours up to four hops away. In the figure, the transmission cost of advertisement and query messages (in bits) per advertisement period ($1/\mu$) is displayed as a function of the number of initiated queries (λ) per advertisement period. This is done for AHOY (BF cost), for a protocol where all services are completely broadcasted (Complete AD), and a protocol where no advertisements are sent, and queries are flooded if needed (No AD). If we note the logarithmic scales, we can see that AHOY outperforms the others over a wide range of parameter values. If the rate at which queries are initiated and the rate at which services change are more or less balanced, the reduction of the message load is a full order of magnitude.

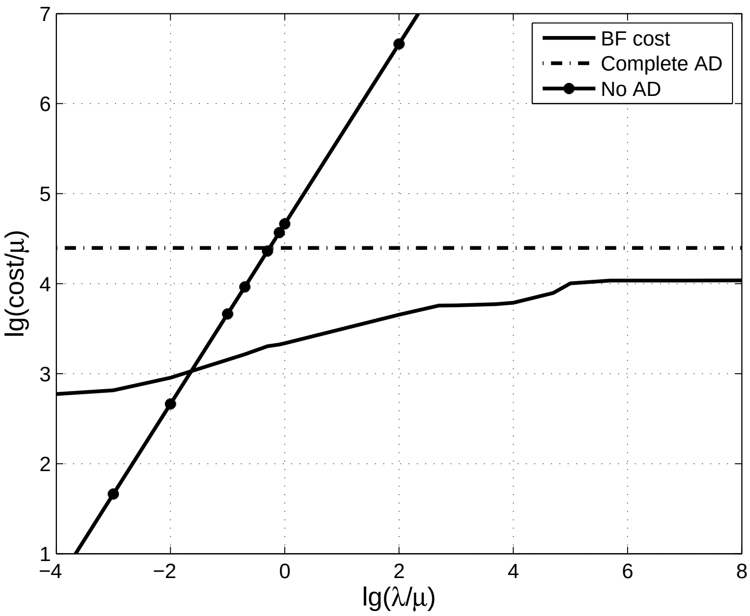

Fig. 5.27. Advertisement and excess query load compared to alternatives [Liu06].

AHOY has also been implemented in the OPNET simulator. Simulation experiments, where the protocol runs on an IEEE 802.11-based ad-hoc network confirm the analytical results described above. Further, in the simulator it was possible to evaluate the performance of the protocol in an environment where all nodes are moving. In these experiments, 25 nodes with a wireless communication range of 300 m are moving in a simulation area of 1500 m × 1500 m. The nodes are all moving according to a random waypoint pattern. The starting positions of the nodes are uniformly spread over the simulation area and the nodes start moving as soon as the simulation starts. For all nodes a destination is chosen distributed uniformly over the simulation area. Nodes move towards their destination with a random speed and pause at their destination for a random amount of time. After the pause a new destination is chosen for the node with a new random speed. To prevent nodes getting trapped at low speeds we use a minimum speed of 0.1 m/s. The maximum speed varies from 1 to 20 m/s and the wait time is uniformly distributed between 0 and 30 seconds. During the experiments, nodes regularly initiate queries for a service that is available on some other node. Fig. 5.28 displays results for the percentage of successful queries, depending on the maximum speed of the nodes. This percentage depends on the depth of the attenuated Bloom filter, *d*. If d is low, there is quite a high probability that a service cannot be found within *d* hops from the querying node, given the density of nodes. The percentage further depends on the maximum speed, as we have intentionally limited the speed at which changes in service availability (advertised attenuated Bloom filters) are propagated through the network, to reduce contention in the 802.11 *Carrier Sense Multiple Access / Collision Avoidance* (CSMA/CA) mechanism. As a result, at higher maximum speed, performance is somewhat degraded. Additional experiments have shown that at high speeds (20 m/s), more than 90% of the services available within *d* hops are found, whereas at low speeds, close to 100% of the services are found.

A final validation of the protocol is by means of a prototype implementation of the protocol [Haa07]. AHOY has been implemented in Ruby, and tested on Debian Linux, OpenBSD and Mac OS X. Several protocol alternatives have been explored and the final prototype performed as expected.

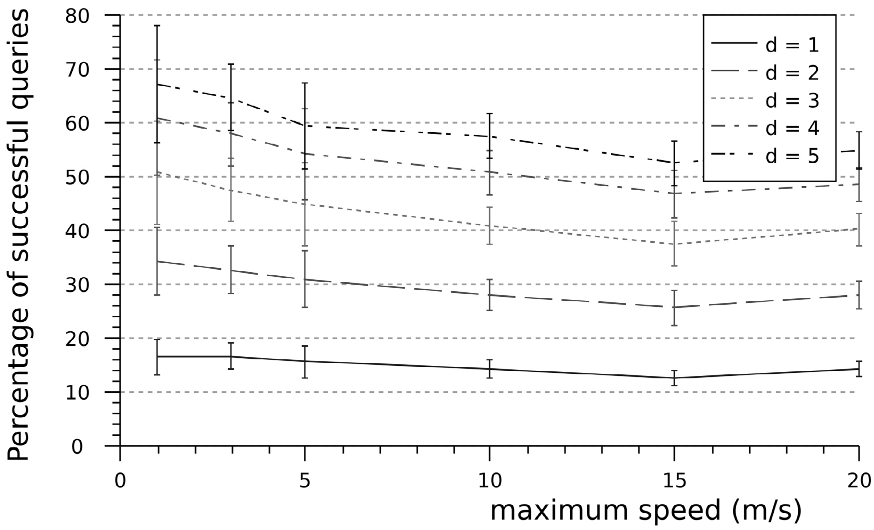

Fig. 5.28. Simulation results for the percentage of successful queries [Goe07].

5.7.2 Alternative Discovery Mechanisms

Several alternatives for AHOY can be found in the literature. ZeroConf [Zer06] defines a suite of simple protocols for zero configuration networking. One of these protocols is *Domain Name System Service Discovery* (DNS-SD) [DNS06], which allows service discovery, and is implemented as part of Apple's Bonjour [Bon06] framework (used extensively by various applications for Mac OS X), as well as supported by the *K Desktop Environment* (KDE) [KDE06] and GNOME [Gno08] desktop environments. The zero configuration service discovery mechanism is built on top of multicast DNS [Mul08]. This is a decentralized and lightweight solution. However, queries must reach all nodes on the network, which does not scale well to larger networks.

In Scalable Service Discovery for MANET [Sai05], a service discovery protocol based on Bloom filters and directory agents is proposed. Service presence information only has to be exchanged among directory nodes, and queries only have to be made from the client node to the nearest directory node, making this protocol particularly efficient in terms of network traffic. Furthermore, Bloom filters are used for the exchange of service availability information among directory agents. Directories are automatically set up using an election algorithm. This protocol differs from AHOY in a number of important aspects. Most importantly, Scalable Service Discovery allows services in the whole network to be discovered, whereas AHOY limits service discovery to a preconfigured number of hops. Secondly, where AHOY services are described by simple strings, Scalable Service Discovery describes services using *DARPA Agent Markup Language* (DAML) [DAM06] and allows queries to refer to attributes and values. Thus, Scalable Service Discovery is more powerful than AHOY, but also much more complex. Also, Scalable Service Discovery relies on directories, which could be problematic in mobile ad-hoc networks, where connectivity changes frequently. By contrast, AHOY is fully decentralized.

GSD [Cha02] is a service discovery protocol designed specifically for mobile ad-hoc networks. It is based on "peer-to-peer caching of service advertisements and group-based intelligent forwarding of service requests". Like Scalable Service Discovery, GSD uses DAML to describe services, thus allowing rich queries to be made. Furthermore, it uses the class/subclass hierarchy described by DAML to selectively forward queries (i.e. queries are not flooded to all nodes). In this sense, DAML descriptions are used in a similar fashion to how Bloom filters are used in Ahoy and Scalable Service Discovery. Although GSD does not use Bloom filters, it does share a number of traits with Ahoy. Like Ahoy, GSD limits service advertisements to a preconfigured number of hops. Also, in GSD as well as in AHOY, all nodes have an equal role, and all nodes cache the information from all service announcements they receive. Finally, both GSD and AHOY resolve service names by means of query messages that are limited to a certain number of hops and are selectively forwarded to neighbours likely to have information about the requested service. Unlike AHOY nodes, which announce all services they know of in a single attenuated Bloom filter, the GSD nodes advertise each service separately, with a large amount of information contained in the advertisements. It can thus be expected that announcing services is much more expensive in GSD than it is in AHOY.

5.8 References

[Agu04] D. Aguayo, Link-level measurements from an 802.11b mesh network, ACM SIGCOMM Computer Communication Review, 34, 2004.
[Aky05] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks and ISDN Systems, Vol. 47, No. 4, 2005.
[Ali06] M. Alicherry, R. Bhatia, E. Li, Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Network, IEEE Journal on Selected Areas in Communications, Vol. 24, No. 11, 2006.
[All98] M. Allman, On the Generation and Use of TCP Acknowledgements, ACM Computer Communications Review, 1998.
[All99] M. Allman, V. Paxson, TCP congestion control, Internet RFC 2581, 1999.
[All01] M. Allman, H. Balakrishman, S. Floyd, Enhancing TCP's Loss Recovery Using Limited Transmit, Internet RFC 3042, 2001.
[Ang07] V. Angelakis, M. Genetzakis, Heraklion MESH: An Experimental Metropolitan Multi-Radio Mesh Network., 2nd ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH), 2007.
[Ava07] S. Avallone, I. Akyildiz, A Channel Assignment Algorithm for Multi-Radio Wireless Mesh Networks, 1st IEEE International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN), 2007.
[Ber06] H. van den Berg, M. Mandjes, F. Roijers, Performance Modeling and Analysis of a Bottleneck Node in an IEEE 802.11 Ad-hoc Network, AdHoc-Now, 2006.
[Bha04] B. Bhargava, Integrating heterogeneous wireless technologies: a cellular aided mobile ad hoc network (CAMA), Mobile Networks and Applicaions, August 2004.
[Bia98] S. Biaz, N. Vaidya, Distinguishing Congestion Losses from Wireless Transmission Losses: A Negative Result, 7th International Conference on Computer Communication and Networks, 1998.
[Bic05] J. Bicket, Architecture and Evaluation of an Unplanned 802.11b Mesh Network, ACM Mobicom, 2005.
[Blo70] B. Bloom, Space/Time Trade-offs in Hash Coding with Allowable Errors, Communications of the ACM, Vol. 13, No. 7, 1970.
[Bon06] Bonjour, http://developer.apple.com/networking/bonjour/index.html, 2006.
[Bom98] E. Bommaiah et al., AMRoute: Ad hoc Multicast Routing Protocol, Internet Draft, work in progress, 1998.
[Bra89] R. Braden, Requirements for Internet Hosts - Communication Layers, Internet RFC 1122, 1989.
[Bra99] L. Brakmo, L. Peterson, TCP Vegas: End to End Congestion Avoidance on a Global Internet, IEEE Journal on Selected Areas in Communications, 1999, Vol. 13, No. 8.
[Bra05] T. Braun, L. Feeney, Power Saving in Wireless Ad hoc Networks without Synchronization, 5th Scandinavian Workshop on Wireless Ad-hoc Networks, 2005.
[Bur05] M. Burgess, A tiny overview of cfengine: Convergent maintenance agent, 1st International Workshop on Multi-Agent and Robotic Systems MARS/ICINCO, 2005.
[Cal04] M. Calisti, T. Lozza, D. Greenwood, An agent-based middleware for adaptive roaming in wireless networks, AAMAS Workshop on agents for ubiquitous computing, 2004.
[Car03] J. Cartigny, D. Simplot, Border node retransmission based probabilistic broadcast protocols in ad-hoc networks, Telecommunication Systems, April 2003.
[Cen07] U.S. Census Bureau Geography Web Page. http://www.census.gov/geo/www, 2007.
[Cha98] K. Chandran, A Feedback Based Scheme for Improving TCP Performance in Ad- Hoc Wireless Networks, 18th International Conference on Distributed Computing Systems, 1998.
[Cha02] D. Chakraborty et al., GSD: A Novel Group-based Service Discovery Protocol for MANETs, 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN), 2002.

[Cha04] J. Chang, L. Tassiulas, Maximum Lifetime Routing in Wireless Sensor Networks, IEEE/ACM Transactions on Networking, 2004.

[Che04a] K. Chen et al., Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks, Elsevier Computer Communications, Vol. 27, No. 10, 2004.

[Che04b] K. Chen, K. Nahrstedt, Limitations of Equation-Based Congestion Control in Mobile Ad Hoc Networks, 24th International Conference on Distributed Computing Systems, 2004.

[Che06] K. Chebrolu, B. Raman, S. Sen, Long-Distance 802.11b Links: Performance Measurements and Experience, ACM Mobicom 2006.

[Chi04] C. Chiasserini, M. Garetto, Modeling the Performance of Wireless Sensor Networks, IEEE Infocom, 2004.

[Cla03] T. Clausen, P. Jacquet, Optimized link state routing protocol (OLSR), Internet RFC 3626, 2003.

[Cof70] E. Coffman, R. Munz, H. Trotter, Waiting time distributions forprocessor-sharing systems, Journal of the ACM, Vol. 17, 1970.

[Coh79] J. Cohen, The multiple phase service network with generalized processor sharing, Acta informatica, Vol. 12, No. 12, 1979.

[Cor07] CoRe Group Implementation Portal, http://core.it.uu.se/core, 2007.

[Dan03] M. Danzeisen, T. Braun, D. Rodellar, S. Winiker, Heterogeneous network establishment assisted by celluar operators, 5th IFIP TC6 International Conference on Mobile and Wireless Communications Networks (MWCN), 2003.

[Dan05] M. Danzeisen, T. Braun, S. Winiker, D. Rodellar, Implementation of a cellular framework for Spontaneous Network Establishment, IEEE Wireless Communications and Networking Conference (WCNC), 2005.

[Dan06a] M. Danzeisen, T. Braun, D. Rodellar, S. Winiker, Heterogeneous Communications Enabled by Cellular Operators, IEEE Vehicular Technology Magazine, Vol. 1, No. 1, 2006.

[Dan06b] M. Danzeisen, T. Braun, I. Steiner, M. Heissenbüttel, On the Potential of Heterogeneous Networks, 4th International Conference on Wired/Wireless Internet Communications (WWIC), Springer LNCS 3970, 2006.

[DAM06] DARPA Markup Language (DAML+OIL), http://www.daml.org, 2006.

[DNS06] DNS service discovery (DNS-SD), http://www.dns-sd.org.

[Dra04a] R. Draves, J. Padhye, B.Zill, Routing in multi-radio, multi-hop wireless mesh networks, ACM Mobicom, 2004.

[Dra04b] R. Draves, J. Padhye, B.Zill, Comparison of routing metrics for static multi-hop wireless networks, ACM Sigcomm, 2004.

[ElH04] A. El-Hoiydi, J. Decotignie, WiseMAC: An Ultra Low Power MAC Protocol forMultihop Wireless Sensor Networks, Alogsensors, 2004.

[ElR05] S. ElRakabawy, A. Klemm, C. Lindemann, TCP with Adaptive Pacing for Multihop Wireless Networks, 6th ACM International symposium on Mobile Ad Hoc Networking and Computing, 2005.

[Est07] D. Estrin et al., Advances in Network Simulation, IEEE Computer, Vol. 33, No. 5, 2000.

[Fee02] L. Feeney, A QoS aware Power Save Protocol For Wireless Ad Hoc Networks, 1st Mediterranean Workshop on Ad Hoc Networks, 2003.

[Flo99] S. Floyd, T. Henderson, The NewReno Modification to TCP's Fast Recovery Algorithm, Internet RFC 2582, 1999.

[Fu02a] Z. Fu, X. Meng, S. Lu, How Bad TCP Can Perform in Mobile Ad Hoc Networks, IEEE Symposium on Computers and Communications, 2002.

[Fu02b] Z. Fu, Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks, IEEE International Conference on Network Protocols, 2002.

[Fu05] Z. Fu et al., The Impact of Multihop Wireless Channel on TCP Performance, IEEE Transactions on Mobile Computing, Vol. 4, No. 2, 2005.

[Gar99] J. Garcia-Luna-Aceves, E. Madruga, The Core-Assisted Mesh Protocol. IEEE Journal on Selected Areas in Communications, Vol. 17, 1999.

[Goe06] P. Goering, G. Heijenk, Service discovery using Bloom filters, 12th Annual conference of the Advanced School for Computing and Imaging, 2006.
[Goe07] P. Goering et al., The effect of mobility on local service discovery in the Ahoy ad-hoc network system, European Performance Engineering Workshop (EPEW), 2007.
[Gom05a] C. Gomez, Evaluating performance of real ad-hoc networks using AODV with hello message mechanism for maintaining local connectivity, 16th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005), Berlin 2005 .
[Gom05b] C. Gomez, D. Garcia, J. Paradells, Improving performance of a real ad-hoc network by tuning OLSR parameters, 10th IEEE Symposium on Computers and Communications (ISCC), 2005.
[Gom06] C. Gomez et al., Adapting AODV for IEEE 802.15.4 Mesh Sensor Networks: Theoretical Discussion and Performance Evaluation in a Real Environment, 7th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM), 2006.
[Gno08] GNOME: The free software desktop project, http://www.gnome.org, 2008.
[Gur03] A. Gurtov, R. Ludwig, Responding to Spurious Timeouts in TCP, IEEE Infocom, 2003.
[Gus03] Gustafsson, E. and Jonsson, A. Always Best Connected. IEEE Wireless Communications Magazine, February 2003.
[Haa02] Z. Haas, J. Halpern, L. Li, Gossip-based ad hoc routing, IEEE Infocom, 2002.
[Haa07] R. Haarman, Ahoy: A Proximity-Based Discovery Protocol, M.Sc. thesis, University of Twente, 2007.
[Ham07] E. Hamadani, V. Rakocevic,TCP Contention Control: A Cross Layer Approach to Improve TCP Performance in Multihop Ad hoc Networks, WWIC 2007.
[Hei02] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, 2002.
[Hei06] M. Heissenbüttel, T. Braun, M. Wälchli, T. Bernoulli, Optimized Stateless Broadcasting in Wireless Multi-hop Networks, IEEE Infocom 2006.
[Hei07] M. Heissenbüttel, T. Braun, M. Wälchli, T. Bernoulli, Evaluating the Limitations of and Alternatives in Beaconing, Ad-Hoc Networks, Elseiver, December 2007.
[Her08] Heraklion MESH, http://www.ics.forth.gr/HMESH, 2008.
[Ho04] C. Ho et al., A scalable framework for wireless network monitoring, 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots (WMASH), 2004.
[Hol99] G. Holland , N. Vaidya, Analysis of TCP Performance over Mobile Ad Hoc Networks, ACM Mobicom, 1999
[Hur06] P. Hurni, T. Braun, L. Feeney, Simulation and Evaluation of Unsynchronized Power Saving Mechanisms in Wireless Ad Hoc Networks, Springer LNCS 3970, 2006.
[Hur07] P. Hurni, T. Braun, Improving Unsynchronized MAC Mechanisms in Wireless Sensor Networks, 1st ERCIM Workshop on eMobility, 2007.
[Ino04] M. Inoue et al., Novel out-of-band signaling for seamless interworking betweem heterogeneous networks, IEEE Wireless Communications, April 2004.
[Jet01a] J. Jetcheva et al., A simple protocol for multicast and broadcast in mobile ad hoc networks, Internet Draft, work in progress, 2001.
[Jet01b] J. Jetcheva, D. Johnson, Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks, ACM MobiHoc, 2001.
[Ji98] L. Ji, S. Corson, A Lightweight Adaptive Multicast Algorithm, IEEE Globecom, 1998.
[Ji01] L. Ji, M. Corson, Differential Destination Multicast: A MANET Multicast Routing Protocol of Small Groups, IEEE Infocom, 2001.
[Jim03] T. Jimenez, E. Altman, Novel Delayed ACK Techniques for Improving TCP Performance in Multihop Wireless Networks, Personal Wireless Communciations (PWC), 2003.

[Joh95] S. Johnson, Increasing TCP Throughput by Using an Extended Acknowledgment Interval, M.Sc. Thesis, Ohio Unversity, 1995.

[Joh07] D. Johnson, Y. Hu, D. Maltz, The dynamic source routing protocol (DSR) for mobile ad hoc networks, Internet RFC 4728, 2007.

[Jos06] H. Joshi, Distributed Robust Geocast: A Multicast Protocol for Inter-Vehicle Communicaton, M.Sc. Thesis, North Carolina State University, 2006.

[Kai02] X. Kaixin et al., TCP Behavior across Multihop Wireless Networks and the Wired Internet, WoWMoM, 2002.

[Kas07] A. Kassler, M. Castro, P. Dely, VoIP Packet Aggregation based on Link Quality Metric for Multihop Wireless Mesh Networks, Future Telecommunication Conference (FTC), 2007.

[Kar03] R. Karrer, A. Sabharwal, E. Knightly, Enabling large-scale wireless broadband: The case for taps, 2nd Workshop on Hot Topics in Networks (Hot-Nets II), 2003.

[Kar07] J. Karlsson, A. Battle, A. Kassler, TCP Performance in Mobile Ad Hoc Networks Connected to the Internet, IWT 2007.

[KDE06] K Desktop Environment, http://www.kde.org, 2006.

[Kih07] M. Kihl et al., Reliable Geographical Multicast Routing in Vehicular Ad-hoc Networks, Lecture Notes in Computer Science 4517, Springer, 2007.

[Lal03] D. Lal et al., Measurement and Characterization of Link Quality Metrics in Energy Constrained Wireless Sensor Networks, IEEE Globecom, 2003.

[Lao01] A. Laouiti, A. Qayyum, L. Viennot, Multipoint relaying: An efficient technique for flooding in mobile wireless networks, 34th Annual Hawaii International Conference on System Sciences (HICSS), 2001.

[Lee00] S. Lee, M. Gerla, AODV-BR: backup routing in ad hoc networks, IEEE Wireless Communications and Networking Conference (WCNC), 2000.

[Lee01] S. Lee, M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, IEEE International Conference on Communications (ICC), 2001.

[Lee02] S. Lee, W. Su, M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, ACM/Kluwer Mobile Networks and Applications, 2002.

[Leu01] R. Leung et al., MP-DSR: a qos-aware multi-path dynamic source routing protocol for wireless ad-hoc networks, 26th Annual IEEE Conference on Local Computer Networks (LCN), 2001.

[Lim00] H. Lim, C. Kim, Multicast tree construction and flooding in wireless ad hoc networks, 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, 2000.

[Lin06] Y. Lin, V. Wong, Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs, IEEE Global Telecommunications Conference, San Francisco, 2006.

[Liu01] J. Liu, S. Singh ATCP: TCP for Mobile Ad Hoc Networks, IEEE Journal on Selected Areas in Communcations, July 2001.

[Liu03] J. Liu, I. Matta, M. Crovella, End-to-End Inference of Loss Nature in a Hybrid Wired/Wireless Environment, Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2003.

[Liu06] F. Liu, G. Heijenk, Context discovery using attenuated Bloom filters in ad-hoc networks, 4th International Conference on Wired/Wireless Internet Communications (WWIC), 2006.

[Liu07a] F. Liu, P. Goering, G. Heijenk Modeling service discovery in ad-hoc networks, 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PEWASUN), 2007.

[Liu07b] F. Liu, G. Heijenk, Context discovery using attenuated Bloom filters in ad-hoc networks, Journal of Internet Engineering, Vol. 1, No. 1, 2007.

[Lud03] R. Ludwig, M. Meyer, The Eifel Detection Algorithm for TCP, Internet RFC 3522, 2003.

[Lud05] R. Ludwig, A. Gurtov, The Eifel Response Algorithm for TCP, Internet RFC 4014, 2005.
[Mai04] C. Maihöfer, A survey of geocast routing protocols, IEEE Communications Surveys & Tutorials, Vol. 6, No. 2, 2004.
[Mar95] M. Marathe, Simple heuristics for unit disk graphs, Networks. 1995.
[Mar02] M. Marina, S. Das, Ad hoc on-demand multipath distance vector routing, ACM SIGMOBILE Mobile Computing and Communications Review, 2002.
[Mel05] T. Melodia, D. Pompili, I. Akyildiz, On the Interdependence of Distributed Topology Control and Geographical Routing in Ad Hoc and Sensor Networks, IEEE Journal on Selected Areas in Communications, 2005.
[Mue04] S. Mueller, D. Ghosal, Multipath routing in mobile ad hoc networks: Issues and challenges, Lecture Notes in Computer Science, 2965, 2004.
[Mul08] Multicast DNS, http://www.multicastdns.org, 2008.
[Nah05] K. Nahm, A. Helmy, J. Kuo, TCP over 802.11 Multihop Networks: Issues and Performance Enhancement, International Symposium on Mobile Ad Hoc Networking & Computing, 2005.
[Nan00] Nandagopal et al., Achieving MAC Layer Fairness in Wireless Packet Networks, International Conference on Mobile Computing and Networking, 2000.
[Ni99] S. Ni, The broadcast storm problem in a mobile ad hoc network, 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 1999
[Non99] J. Nonnenmacher, E. Biersack, Scalable feedback for large groups, IEEE/ACM Transactions on Networking, June 1999.
[Nsn07] Main Page Nsnam, http://nsnam.isi.edu/nsnam, 2007.
[Oli05a] R. de Oliveira, Addressing the Challenges for TCP over Multihop Wireless Networks, Ph.D. Thesis, Universität Bern, 2005.
[Oli05b] R. de Oliveira, T. Braun, A Dynamic Adaptive Acknowledgment Strategy for TCP over Multihop Wireless Networks, IEEE Infocom, 2005.
[Oli07] R. de Oliveira, T. Braun, A Smart TCP Acknowledgment Approach for Multihop Wireless Networks, IEEE Transactions on Mobile Computing, February 2007.
[OLS08] olsrd – an adoc wireless mesh routing daemon, http://www.olsr.org, 2008.
[Pat06] R. Patra et al. WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks, USENIX NSDI, 2006.
[Pce08] Pcengines, Wireless Router Application Platform (WRAP), http://www.pcengines.ch, 2008
[Per03] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On Demand Distance Vector Routing (AODV), Internet RFC 3561, 2003.
[Roy99] E. Royer, C. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, ACM/IEEE Mobicom, 1999.
[Ran04] A. Raniwala, K. Gopalan, T. Chiueh, Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks, ACM Mobile Computing and Communications Review, Vol. 8, No. 2, 2004.
[Ran05] A. Raniwala, T. Chiueh, Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network, IEEE Infocom, 2005.
[Ray05] D. Raychaudhuri et al., Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols, IEEE Wireless Communications and Networking Conference, 2005.
[Roi07] F. Roijers, H. van den Berg, M. Mandjes, Fluid flow performance modeling of a relay node in an IEEE 802.11 ad-hoc network, ITC20, 2007.
[Sai05] F. Sailhan, V. Issarny, Scalable service discovery for manet, 3rd IEEE International Conference on Pervasive Computing and Communications (PERCOM), 2005.
[Sar03] P. Sarolahti, M. Kojo, K. Raatikainen, F-RTO: An Enhanced Recovery Algorithm for TCP Retransmission Timeouts, ACM Computer Communications Review, 2003.

[Sar05] P. Sarolahti, M. Kojo, Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol (SCTP), Internet RFC 4138, 2005.

[Sic07] M. Sichitiu, M. Kihl, Inter-Vehicle Communication Systems – a Survey, IEEE Communications Surveys & Tutorials, 2007.

[Sin99] P. Sinha, R. Sivakumar, V. Bharghavan, MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing, IEEE Wireless Communications and Networking Conference (WCNC), 1999.

[Sou06] M. Souryal et al., Link Assessment in an Indoor 802.11 Network, Wireless Communications and Networking Conference, 2006.

[Sta07a] B. Staehle, D. Staehle, Impact of Energy Models on Energy Efficient Sensor Network Routing, MASS 2007.

[Sta07b] T. Staub, D. Balsiger, M. Lustenberger, T. Braun, „Secure Remote Management and Software Distribution for Wireless Mesh Networks, 7th International Workshop on Applications and Services in Wireless Networks (ASWN), 2007.

[Sto02] I. Stojmenovic, M. Seddigh, J. Zunic, Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, IEEE Transactions on Parallel and Distributed Systems, January 2002.

[Sun03] H. Sun, H. Hughes, Adaptive multi-path routing scheme for qos support in mobile ad-hoc networks, International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2003.

[Sun05] K. Sundaresan et al., ATP: A Reliable Transport Protocol for Ad-Hoc Networks, IEEE Transactions on Mobile Computing, December 2005.

[Sus00] J. Susec, I. Marsic, An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks, Center for Advanced Information Processing (CAIP), Technical Report TR-248, Rutgers University, 2000.

[Tan05] J. Tang, G. Xue, W. Zhang, Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks, ACM MobiHoc, 2005.

[Teb04] H. Tebbe, A. Kassler, P. Ruiz, QoS-aware Mesh Construction to Enhance Multicast Routing in Mobile Ad Hoc Networks, Intersense 2006.

[Tse02] Y. Tseng, C. Hsu, T. Hsieh, Power Saving Protocols for IEEE 802.11-based Multi-hop Ad hoc Networks, IEEE Infocom 2002.

[Tse03] Y. Tseng, S. Ni, E. Shih, Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network, IEEE Transactions on Computers, 2003.

[TSB06] Arlington: Telecommunications Industry Association, TSB-116-A - Voice Quality Recommendations for IP Telephony, 2006.

[Val03] A. Valera, W. Seah, S. Rao, Cooperative packet caching and shortest multipath routing in mobile ad hoc networks, IEEE Infocom, 2003.

[Wan01] L. Wang et al., Adaptive multipath source routing in ad hoc networks, IEEE International Conference on Communications (ICC), 2001.

[Wan06] Q. Wang, M. Hempstead, W. Yang, A Realistic Power Consumption Model for Wireless Sensor Network Devices, IEEE SECON, 2006.

[Wie00] J. Wieselthier, G. Nguyen, A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, IEEE Infocom, 2000.

[Wil02] B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, 3rd ACM International Symposium on Mobile and Ad Hoc Networking and Computing (Mobi-Hoc '02), 2002.

[Wu99] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), 1999.

[Wu07] D. Wu, D. Gupta, P. Mohapatra, Quail Ridge Reserve Wireless Mesh Network: Experiences, Challenges and Findings, TRIDENTCOM, 2007.

[Xiu04] W. Xiuchao, Simulate 802.11b channel within ns2. Singapore, Technical Report, National University of Singapore, 2004.

[Xu01] S. Xu, T. Saadawi, Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks ?, IEEE Communications Magazine, June 2001.

[Xu02] K. Xu et al., TCP Behavior Across Multihop Wireless Networks and the Wired Internet, 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM), 2002.

[Xu05] W. Xu, P. Yan, D. Xia, Similar node-disjoint multi-paths routing in wireless ad hoc networks, International Conference on Wireless Communications, Networking and Mobile Computing (WiMob), 2005.

[Ye03] Z. Ye, S. Krishnamurthy, S. Tripathi,A framework for reliable routing in mobile ad hoc networks, IEEE Infocom, 2003.

[Yua05] Y. Yuan et al., Romer: Resilient opportunistic mesh routing for wireless mesh networks, 1st IEEE Workshop on Wireless Mesh Networks (WiMesh), Santa Clara, 2005.

[Zer06] Zero configuration networking (ZeroConf), http://www.zeroconf.org, 2006.

The COST 290 documents can be downloaded from www.cost290.org. These documents referenced in this Chapter are listed below:

[TD(05)001] R. de Oliveira, T. Braun, A Dynamic Adaptive Acknowledgment Strategy for TCP over Multihop Wireless Networks.

[TD(06)003] H. van den Berg, M. Mandjes, F. Roijers, Performance Modeling and Analysis of a Bottleneck Node in an IEEE 802.11 Ad-hoc Network.

[TD(06)016] M. Heissenbüttel, T. Braun, M. Wälchli, Th. Bernoulli, Optimized Stateless Broadcasting in Wireless Multi-hop Networks.

[TD(06)020] F. Liu, G. Heijenk, Modeling of Service Discovery in Ad Hoc Networks.

[TD(06)029] A. Kassler, QAMNet: Providing Quality of Service for Ad-hoc Multicast Enabled Network.

[TD(06)033] F. Liu, G. Heijenk, Context Discovery Using Attenuated Bloom Filters in Ad-hoc Networks.

[TD(06)037] M. Danzeisen, T. Braun, I. Steiner, M. Heissenbüttel, On the Potential of Heterogeneous Networks.

[TD(06)040] P. Hurni, T. Braun, L. Feeney, Simulation and Evaluation of Unsynchronized Power Saving Mechanisms in Wireless Ad hoc Networks.

[TD(06)049] M. Kihl, User-perceived performance in VANETs.

[TD(06)051] T. Staub, Multipath Routing in Wireless Mesh Networks.

[TD(07)003] T. Braun, Secure Remote Management and Software Distribution for Wireless Mesh Networks.

[TD(07)016] H. van den Berg, Fluid Flow Performance Modeling of a Relay Node in an IEEE 802.11 Ad-hoc Network.

[TD(07)020] A. Kassler, On Packet Aggregation for VoIP in Wireless Meshed Networks.

[TD(07)026] V. Rakocevic, Cross-Layer Traffic Control in Mesh Networks.

[TD07(032)] S. Aprin, Evaluation of an Intelligent Route Guidance System based on Inter-Vehicle Communication.

[TD(07)034] T. Braun, Power Saving in Wireless Multi-hop Networks.

[TD(07)038] V. Siris, Heraklion MESH: An Experimental Metropolitan Multi-Radio Mesh Network.

[TD(07)040] B. Staehle, Impact of Energy Models on Energy Efficient Sensor Network Routing.

[TD(07)051] A. Kassler, Karlstad University, Sweden): TCP performance across internet connected MANETs.

[TD(07)052] S. Avallone, A Channel Assignment and Routing Algorithms for Multi-Radio Wireless Mesh Networks.
[TD(07)053] C. Gomez, N. Vesselinova, J. Paradells, F. Barcelo, Routing in MANETs: protocol settings, tradeoffs and energy consumption.