# INTRODUCTION TO MULTI-HOP WIRELESS NETWORK ROUTING

## 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.

# 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.

## *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.

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.

### 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.

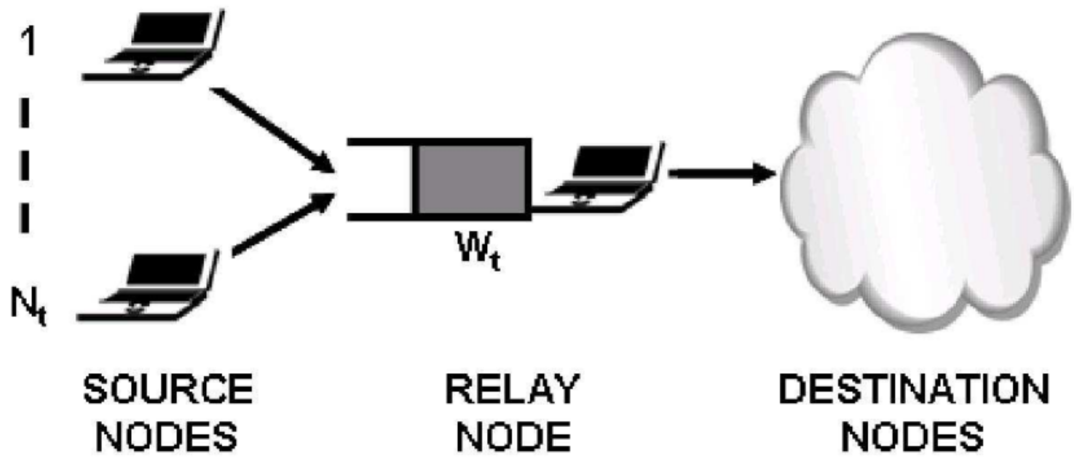


**Fig. 5.1.** Ad-hoc network scenario.

### 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 $m_n C_n/(n+m_n)$. 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.

### 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 $\pi_n$, 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 $\Delta W_{buffer}$ during $D_{source}$. Explicit expression for $W_{buffer}$ and $\Delta W_{buffer}$ 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

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

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 $\tau$ 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 E $D_{overall}$ as we have derived expressions for the means of both parts of (5.1).

### 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.

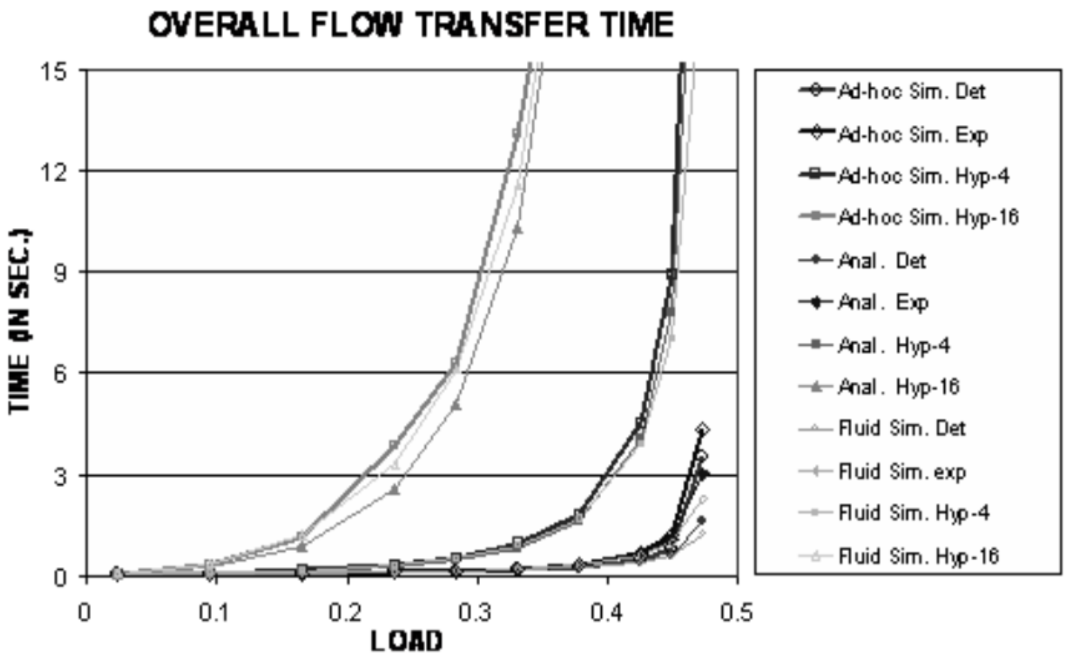


**Fig. 5.2.** Overall flow transfer time in equal resource-sharing bottleneck model.

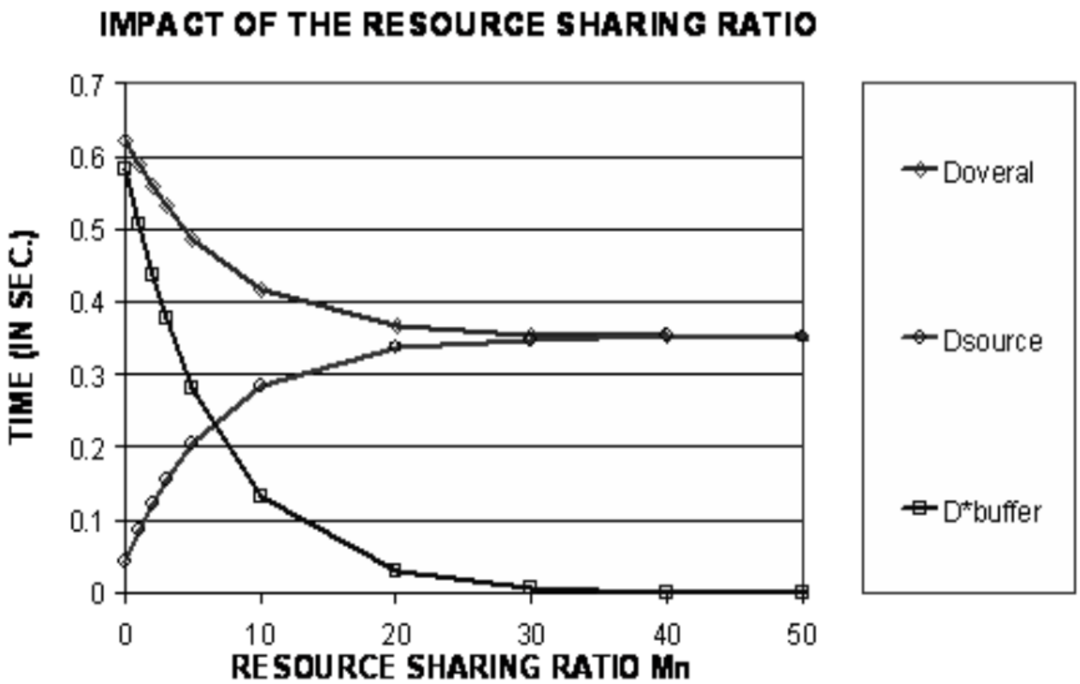


**Fig. 5.3.** Impact of the resource sharing ratio m for load 0.43.

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].

## *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).

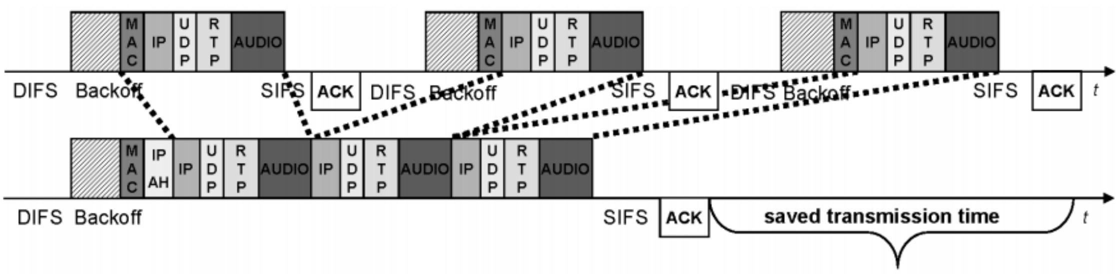


**Fig. 5.4.** Packet Aggregation saves transmission time and reduces overhead

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.

## 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.

### 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].

## 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.

## *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].

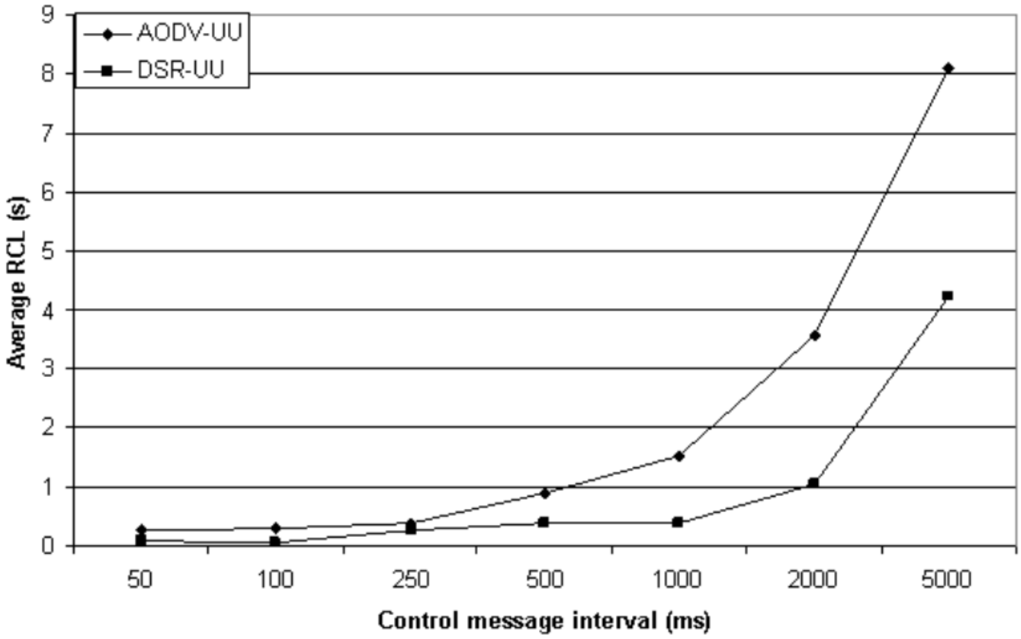


**Fig. 5.5.** Measured RCL in a two-branch, two-hop path test-bed using AODV and DSR.

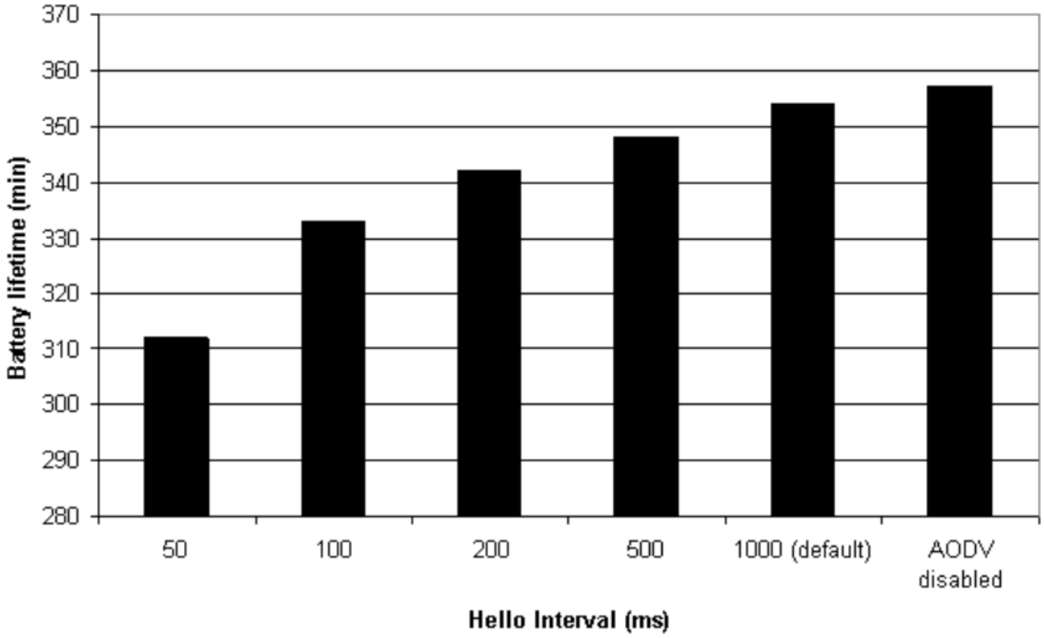


**Fig. 5.6.** Impact of AODV Hello Interval on laptop battery lifetime [Gom05a].

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.

## *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.

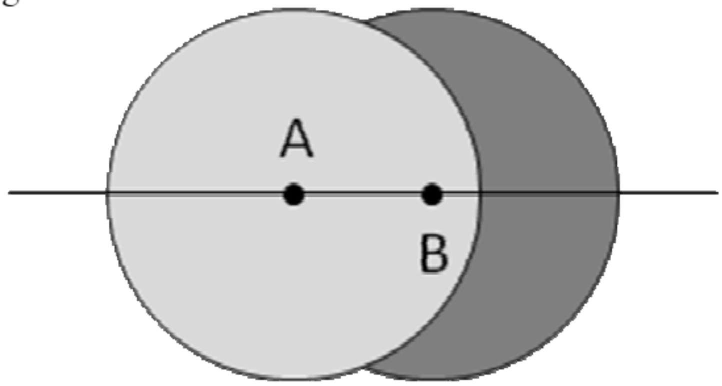


**Fig. 5.7.** Additionally Covered Area (AC).

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.

## *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.

## 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.

REFERENCE

1. Wireless Networking in the Developing World: A practical guide to planning and building low-cost telecommunications infrastructure (PDF) (2nd ed.). Hacker Friendly LLC. 2007. p. 425.

2. Pahlavan, Kaveh; Levesque, Allen H (1995). Wireless Information Networks. John Wiley & Sons. ISBN 0-471-10607-0.

3. Geier, Jim (2001). Wireless LANs. Sams. ISBN 0-672-32058-4.

4. Goldsmith, Andrea (2005). Wireless Communications. Cambridge University Press. ISBN 0-521-83716-2.

5. Molisch, Andreas (2005). Wireless Communications. Wiley-IEEE Press. ISBN 0-470-84888-X.

6. Pahlavan, Kaveh; Krishnamurthy, Prashant (2002). Principles of Wireless Networks – a Unified Approach. Prentice Hall. ISBN 0-13-093003-2.

7. Rappaport, Theodore (2002). Wireless Communications: Principles and Practice. Prentice Hall. ISBN 0-13-042232-0.

8. Rhoton, John (2001). The Wireless Internet Explained. Digital Press. ISBN 1-55558-257-5.

9. Tse, David; Viswanath, Pramod (2005). Fundamentals of Wireless Communication. Cambridge University Press. ISBN 0-521-84527-0.

10. Pahlavan, Kaveh; Krishnamurthy, Prashant (2009). Networking Fundamentals – Wide, Local and Personal Area Communications. Wiley. ISBN 978-0-470-99290-6.