On the Potential of IEEE 802.11s Intra-Mesh Congestion Control

Desheng Fu, Barbara Staehle, Rastin Pries, Dirk Staehle
University of Würzburg, Institute of Computer Science
Würzburg, Germany
fu@qsid.com, {bstaehle|pries|dstaehle}@informatik.uni-wuerzburg.de

ABSTRACT

IEEE 802.11s is an emerging IEEE 802.11 amendment, aiming at standardizing wireless mesh networking. As congestion is a major problem in wireless mesh networks, IEEE 802.11s addresses this problem by introducing the intra-mesh congestion control. In order to explore the potential of this mechanism, we describe two different IEEE 802.11s compliant congestion control mechanisms and discuss their respective benefits and limitations. Results from a simulative evaluation demonstrate that intra-mesh congestion control is suitable for avoiding the loss of packets which have already been forwarded over the air interface and, if appropriately implemented, increases the overall network throughput. The results do however also point out the limitations of the proposed IEEE 802.11s congestion notification format.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design—*Wireless Communication*; C.2.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Operations—*Network Management*

General Terms

Algorithms, Performance

Keywords

Wireless Mesh Networks, Congestion Control, IEEE 802.11s

1. INTRODUCTION

Wireless Mesh Networks (WMNs) are a promising opportunity for enhancing the current wireless last mile access. If mesh station are cooperating to connect end users via multi-hop paths, this allows to cover a significantly larger area than if the stations are working as individual access points. The price for the extended coverage is the increased number of severe congestion problems which arise wireless mesh networks built on the contention-based IEEE 802.11 technology. To improve the overall performance of WMNS, the upcoming IEEE 802.11s [5], which introduces IEEE 802.11 amendments for wireless mesh networking, includes thus basic mechanisms for congestion control.

The proposed *intra-mesh congestion control* consists of three building blocks, namely *congestion monitoring and detection*, *congestion control signaling*, and *local rate control* [5]. Congestion monitoring means that each station constantly checks local parameters and detects a congestion if certain thresholds are exceeded. After the detection, a congested station determines which nearby stations should be controlled and sends them a signaling message. Finally, all stations, which received the signaling messages, apply rate control, i.e. adjust their outgoing throughputs until the congestion duration, specified in the signaling message, expires. IEEE 802.11s only specifies the format of the signaling messages and leaves the rest of the congestion control mechanism, especially the rate control, open to the vendor [5]. As the congestion duration is the only information which is contained in the signaling messages, the amount of transmitted control data is minimized. The price for this small overhead is that the congested station can only inform the controlled stations about the congestion, but is neither able to consider the state of the controlled stations nor to specify how to adjust the sending rate.

The primary goal of intra-mesh congestion control is to avoid dropping packets which have already been forwarded over the air interface. If such a packet is dropped, the network throughput is not only decreased by the size of the dropped packet but also by the amount of payload which could not been sent instead. This so called *intra-mesh packet loss* is hence affecting the network throughput more severely than *local packet loss* which denotes the case if a station drops packets coming from its own application layer. In this paper, we present two different ways for realizing the three steps for congestion control proposed by IEEE 802.11s [5]. A simulation study allows to investigate if these schemes, which avoid intra-mesh packet loss at the price of an increased local packet loss are beneficial in terms of increased network throughput.

The *total congestion control* (TCC) and the more fine-grained *link specific congestion control* (LSCC) algorithm realize the first requirement for congestion control in a similar way: Each station monitors the size of its MAC layer queue and considers itself congested if the buffer size exceeds a critical threshold. LSCC additionally analyzes where the intra-mesh packets come from. The second congestion control step, namely the transmission of control packets is realized as multicast to all neighboring stations. LSCC

This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01 BK 0800, GLab). The authors alone are responsible for the content of the paper.

MSWiM'10, October 17–21, 2010, Bodrum, Turkey.

is additionally able to control selected neighbors only, if necessary. The reaction to the signaling is again implemented differently: While TCC blocks all outgoing data traffic of a controlled node, LSCC stops only the packets intended for the congested node.

In this work, we describe and evaluate both mechanisms. We detail on method inherent problems and implementation challenges and compare both mechanisms in terms of throughput benefits. The remainder of this paper is organized as follows. Section 2 gives an overview of previously proposed WMN congestion control mechanisms. The IEEE 802.11s compliant intra-mesh congestion control mechanisms TCC and LSCC are described in Section 3. Section 4 reports on the results of a simulation study allowing to compare the performance of TCC and LSCC. Finally, Section 5 concludes this paper and provides a brief outlook on future work.

2. RELATED WORK

Surveying all congestion control algorithms applicable for WMNs goes beyond the scope of this work, but an exhaustive survey of mobile ad hoc network congestion control algorithms whereof the most are also suitable for WMNs is e.g. given by Lochert et al. [7]. Most of those mechanisms modify the TCP congestion control by changing its behavior through explicit feedback mechanisms. One example for this idea is TCP-F, introduced by Chandran et al. [2], which tries to freeze the TCP state such as the congestion window temporarily when non-congestion related losses or timeout events are detected. ECN introduced in [8] is another end-to-end congestion notification mechanism cooperating with TCP. If congestion is imminent, a station may set a mark in the IP header which will trigger the source of data flows to reduce their sending rate. A similar idea is proposed by Chen and Nahrstedt [3] who require forwarding stations to embed explicit rate information in each flow. The main disadvantage of these approaches is that [2] and [8] build on TCP whereas [3] requires the TCP reliable data control. They are hence not generally applicable for all transport-layer protocols.

Hop-by-hop congestion control mechanisms in contrast mainly work on the data-link layer. Only neighboring routers communicate and each router has to locally decide how to resolve congestion. One example of such a protocol is wGPD, introduced by Akyol et al. [1], which is a combined congestion control and scheduling algorithm. Its goal is to bound the network queues while maximizing the network utility which is achieved by maintaining so called urgency weights and a separate queue for each potential destination. NRED, introduced by Xu et al. [9] in contrast, realizes a totally different concept: all stations sharing the channel access time maintain a common queue from which packets are sent according to topology and traffic dependent characteristics. The downside of these and similar ideas is however that information on the urgency weights and local queue sizes have to be exchanged between neighboring stations. While this is e.g. achievable by modifying RTS/CTS frames, this and other previously discussed behaviors are not foreseen by IEEE 802.11s which provides a very limited congestion control notification packet format only. In the remainder of this work we will therefore present two IEEE 802.11s compliant mechanisms which are additionally independent from the transport layer and thus works for both, TCP and UDP traffic.

3. IEEE 802.11S COMPLIANT INTRA-MESH CONGESTION CONTROL

In this section, we describe two simple but IEEE 802.11s compliant congestion control mechanisms, the total and the link selective congestion control. As outlined earlier, IEEE 802.11s intra-mesh congestion control consists of congestion monitoring and detection, signaling, and rate control. In the following, we describe which parts are identically implemented for both TCC and LSCC before we detail on the implementation differences and discuss the respective advantages and disadvantages.

3.1 Common Features of TCC and LSCC

What is common to both congestion control algorithm is the buffer management algorithm which addresses the following problem: If a network uses UDP or another transport layer protocol without congestion control, the sending rate is not adapted to the network condition. It may hence easily happen that a station's MAC layer queue is filled with packets coming from its own application layer, as the channel access time is not sufficient for transmitting them fast enough. If the buffer is filled by local packets, this means that packets the station should forward for other nodes can not be buffered and have to be dropped. This phenomenon is what we call intra-mesh packet loss and decreases the network throughput more strongly than if the station would have dropped the packets from its own application layer, causing local packet loss. It is hence more important to avoid intra-mesh packet loss and consequently, each station reserves 80% of its MAC layer queue for packets which it has to forward on behalf of other nodes. Only the remaining 20% are used for local packets originating at the station. Our experiments showed that this configuration is not affecting the total network throughput, but is important for the mechanisms we describe in the following. The performance evaluation of other buffer partitions is the scope of our future studies.

How each station could monitor and detect congestion is not specified by [5]. This could e.g. be achieved by monitoring the receiving and the transmitting rate, and determine a congestion if the receiving rate is significantly larger than the transmitting rate. Another possibility is to detect a congestion if the MAC layer queue size exceeds a certain threshold. The latter option allows to detect an imminent packet dropping and is therefore a good option for achieving the main goal of intra-mesh congestion control, namely to avoid intra-mesh packet loss. The buffer fill level is easy to quantify which makes the buffer monitoring moreover simple to implement. Consequently, we choose this way for implementing the local congestion monitoring and detecting step.

After detecting congestion, a congested station transmits congestion control notification packets. Note that it may be necessary to send these control packets both to the source of the congestion and to other uninvolved neighboring station, but the payload carried by those packets, the congestion notification element, is the same. The only information an IEEE 802.11s compliant congestion notification element carries is the congestion notification expiration timer for each of four different access classes. This expiration timer is defined as the time, the station expects the congestion to last. We consider only one access class, the congestion notification expiration timer, which can take values between 0 and roughly 6.5 sec, is hence the only information a congested station can transmit.

How a station computes the value of the timer is again left open by [5], but our experiments showed that using a constant value for the congestion notification expiration timer is not suitable for increasing the WMN performance. In order to make the congestion control adaptive to the environment, each station maintains a congestion duration variable which is used for each congestion control notification packet. This variable is initialized network wide with the same value and adapted by two locally working mechanisms which are started as soon as a station has sent out the first congestion control notification. The first mechanism lets a station check its buffer each time shortly before the last used congestion notification expiration timer expires. If the buffer is still too full, this means that the congestion control duration has been too short. Consequently, the congestion duration variable is slightly increased and used to send a new congestion control notification packet. The second mechanism lets the station check the buffer fill level each time it has transmitted a data packet. If the buffer level is below a certain threshold, this means that the last congestion control duration has been too long. In this case, the congestion duration variable is slightly decreased. If the congestion control expiration timer is still active, a congestion control notification packet with the expiration timer set to 0 is transmitted. This ensures that all stations which limited their outgoing traffic stop this limitation.

The congestion signaling and the rate control mechanisms are implemented differently and discussed separately for TCC and LSCC.

3.2 Total Congestion Control

An implementation of the congestion signaling has first to choose the destinations of a congestion control notification packet. At first glance, two obvious possibilities for this problem exist. One possibility is to broadcast such a frame. This ensures that all stations which share the channel access time with the congested station limit their transmission rate and thereby allow the congested station to access the medium. The disadvantage of this method is clearly that a large number of stations are blocked, including stations that are neither forwarding nor receiving packets from the congested station. We therefore limit the range of congestion control packets to all neighbors in the routing tree. This ensures that nearby station not involved in the congestion may still forward their traffic, while it prevents a congested station from receiving new packets.

If a station receives a congestion control notification, it has to apply local rate control, i.e. to decide how to limit its outgoing traffic. This reaction is neither specified by [5] nor does the congestion control packet provide information about the maximum permissible throughput during the limitation. A controlled station must therefore autonomously decide what to do. It is imaginable to implement iterative reductions of the outgoing traffic, but the most simple solution is to use an on-off scheme, i.e. to limit the outgoing traffic to zero. In other words, with the exception of control and signaling packets, no packets are transmitted any more once a congestion control packet has arrived. We chose this solution for the TCC rate control as it is easy to implement and furthermore maximally increases the channel access time of the congested station.

The implementation simplicity comes at the price of two problems which are discussed in the following. The effect which we call *waste of bandwidth* occurs if a station is forwarding packets to more than one neighbor and is blocked due to one of them. If e.g. station A is forwarding packets to nodes B, C, and D, whereof B signals a congestion, A will not forward packets to node B any more which reduces the congestion at this station. During the control phase, A however also stops sending packets to C, and D. This may be useful, if A and B are interfering and B may increase its transmission rate as A is not sending any more. In some cases this total traffic limitation does however not help the congested station as the limitation does not solve the cause of the problem. This problem is clarified by the example depicted in Figure 1. Station D forwards flow 1 to station C and flow 2 to station A. In this example, the link rate of (D, C) is larger than the one of (C, B) and C can not forward the packets to B as fast as it receives them from D. Consequently, C's buffer level is soon above the critical threshold and it sends a congestion control message to D and B. This causes D to stop sending any packets and flow 2 will also starve. This does however not improve the situation of station C since the reason for its congestion is the capacity of the link (C, B) and not a reduced channel access probability due to D's activity. TCC is hence able to prevent intra-mesh packet loss, but at the price of an increased local packet loss at D and a starvation of flow 2 and thereby a network throughput decrease.

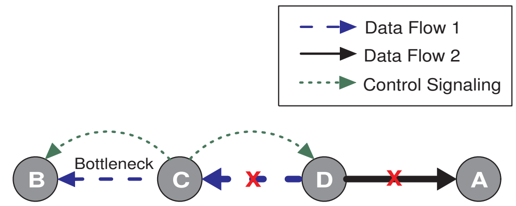


Figure 1: Waste of Bandwidth

The other negative effect, called *congestion control collision*, is visualized in Figure 2(a). It occurs if two neighboring stations are congested and try to control each other. Consequently, no traffic between the two stations is allowed during the limitation. Sometimes, this might resolve the congestion problem, but in some cases, this is not improving, but worsening the situation. In the exemplary network topology, station B and station C can not forward the packets of data flow 1 and 2 to stations A and D respectively, fast enough. Both hence detect a full buffer and send out congestion control notification packets thereby trying to limit each other. Thus, neither station B nor station C is allowed to transmit packets. As a result, the buffer levels of B and C do not decrease, as no packet can be relayed from B to A and from C to D. TCC does hence not solve the problem as the buffers of both stations are still full after the congestion control phase is over.

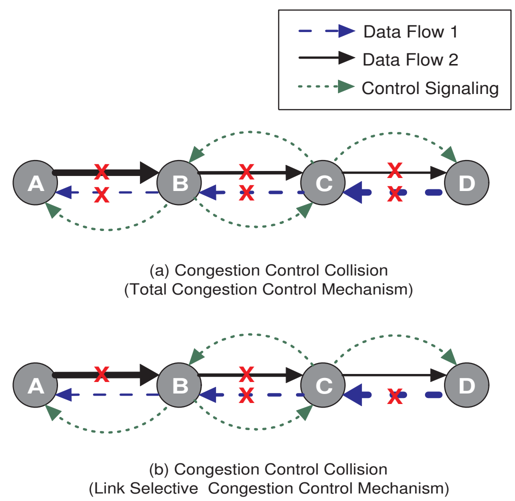


Figure 2: Congestion Control Collision

The reason why the two previously discussed ideas occur if TCC is applied is the multiple relationships between neighboring stations, i.e. it is rather common that a node is both receiving from and forwarding packets to another node. As WMNs are typically used as Internet access networks and bidirectional flows, caused e.g. by simultaneous video downloads and web surfing activities, are prevailing, the probability of such a mutual relationship is rather high. If now one of those nodes (in the example of Figure 2(a) station B) is congested and signals this to the other station (in the example station C), the receiver of this packet can not determine its role in the problem. It could be the cause of the congestion since it sends too much packets to the congested station. The appropriate reaction in this case would be a transmission stop, which would be the right choice for station C. It could however also be the solution of the congestion since it receives packets from the overflowed station and relays them, thereby alleviating the load of the congested station. In this case, it should ignore the signaling. This would also be the right choice for C, as otherwise no packets of flow 2 would be forwarded to D any more. LSCC, which we discuss in the next section, therefore uses more sophisticated rate control and signaling solutions to cope with this problem.

3.3 Link Selective Congestion Control

As outlined earlier, TCC implements the local rate control required by [5] by stopping the transmission of all packets. LSCC in contrast, lets the station only stop the transmission on the link to the congested station. The advantage of this more sophisticated concept is illustrated in Figure 2(b). Congestion control collision might still occur if station B and C detect a congestion at the same time. However, only the traffic between station B and C is stopped while B can still send packets to A and C to D. Thereby, the buffers of stations B and C are cleared and the congestion is solved.

While LSCC improves upon TCC, its implementation is more difficult. The reason for this is that a selective traffic control requires a more advanced buffer management system than a simple FIFO buffer. This is similar to the idea introduce in [1] and can be imagined as a FIFO queue where the packets forwarded to the same station have the same color. In the normal case, i.e. if the congestion control is not active, the colors are ignored and the packets in the queue are sent according to their time of arrival. If however a neighboring station has signaled congestion, no packets to this station are sent, i.e. no packets with this color, but only packets with other colors are transmitted. To realize this principle, each station creates one virtual FIFO buffer per neighboring station in its physical buffer space. The length of each virtual buffer is only limited by the total buffer space which means that a packet can be inserted into every virtual buffer as long as there is enough physical buffer space. If the congestion control is not active, and after a congestion control phase has ended, the virtual buffers are ignored and the packets are simply sent according their time of arrival. Upon the arrival of a congestion signaling packet from a neighboring station, no packets out of the corresponding virtual buffer are chosen.

Virtual buffers and link selective rate control avoid the waste of bandwidth problem inherent to TCC. However, LSCC also has an inherent inefficiency which leads to bandwidth reductions. We call this the *buffer blocking* effect and illustrate it in Figure 3. In Figure 3(a) the situation is depicted where, due to the congestion control mechanism, some packets are blocked and may not be sent towards their destination. Consequently they must be hold in the buffer. Recall that the total buffer size is *not* equally divided among the virtual buffers, but that each virtual buffer could in theory occupy the entire buffer space. The advantage of this solution is that no storage resources are wasted if one virtual buffer fills up quickly while others are empty. The disadvantage is that if a station is not allowed to forward packets to one of its neighbors, but continuously receives packets for this neighbor, it could happen that the station's buffer entirely fills up with blocked packets. Without congestion control, the worst case depicted in Figure 3(b) would occur and the buffer is entirely filled up with blocked packets. To prevent intra-mesh packet loss, the station would send out control messages and consequently cause all of its children, to limit the sending rate. As the children whereof the station still could forward the packets do also stop transmitting, the results is again a waste of bandwidth.

To avoid buffer blocking, the part of the buffer where the local

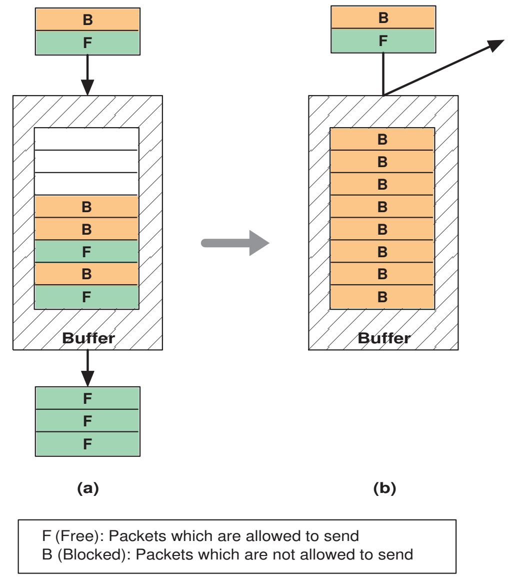
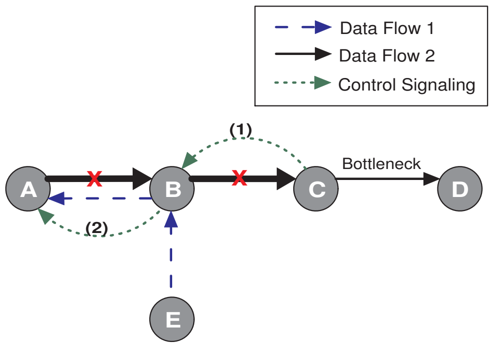


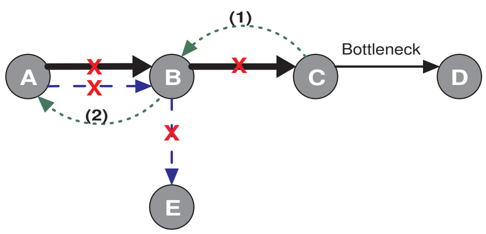
Figure 3: Buffer Blocking

packets and where the packets to relay are hold have to be handled differently. The local buffer blocking effect is simply resolved by limiting the total buffer usage of blocked local packets to 50%, a value which worked best during our experiments. This means that no more than 10% of the total buffer are occupied by blocked packets, the other 10% available to local packets can still be used for packets which may be sent. This method is however not applicable for the buffer where the packets to forward are hold since it would lead to intra-mesh packet loss which is opposed to the intuition of intra-mesh congestion control. The perfect solution for this problem would be to limit only the traffic, which causes this effect. Such solutions might be available for some congestion control mechanisms but not for an IEEE 802.11s compliant intra-mesh congestion control, where the duration of the congestion control phase is the only information which may be transmitted by a congested station. A congested station is however able to choose which station should be limited and can realize a link selective signaling mechanism. This mechanism is more complex to implement but required for situations like the one depicted in Figure 4(a).

In this case, station C is congested, and sends a signaling (1) to station B. If the duration of the congestion control is too long, the buffer of station B fills up with packets which should be sent to C and B is also congested. It is however still able to forward traffic from E to A. By analyzing where the packets in its buffer come from, station B is aware of this problem, and therefore only sends the congestion control notification to A and not to E which would happen under TCC. Choosing the recipient of congestion control notification frames is actually not a trivial task. This is mainly because all neighboring stations should be blocked if the buffer is almost full, but those stations, whereof the packets occupy too much buffer space, should be limited before reaching this threshold. Thus, a look-ahead congestion detection and signaling algorithm is applied which complements the normal signaling algorithm. The look ahead strategy causes a station to receive a congestion control packet not only if it is forwarding packets to a node whereof the buffer is almost full, but also if a buffer blocking effect due to packets from this station is imminent. Compared to the previously introduced signaling algorithm, where packets go always



(a) Solvable Situation for Intra-Mesh Congestion Control



(b) Unsolvable Situation for Intra-Mesh Congestion Control

Figure 4: Solvable and Unsolvable Buffer Blocking Situations

to all neighbors, this additional signaling algorithm limits only the stations, whose packets occupy the most buffer space.

The limitations of the link selective signaling and the look-ahead strategy are illustrated by the example depicted in Figure 4(b). In this case, B could still forward the packets of flow 1, as there is no bottleneck situation. Station A has only to stop sending packets of flow 2 in order to end the congestion. Due to the congestion control message format given by IEEE 802.11s, B is however unable to communicate to A that only the packets belonging to flow 2 have to be limited. A flow selective congestion control protocol required for such situations is thus not feasible without a modification of the congestion control format proposed by IEEE 802.11s.

4. PERFORMANCE EVALUATION

In this section, we evaluate the impact of TCC and LSCC on the WMN performance. For this purpose, we use our own simulation framework which is implemented in C++ and is publicly available[1]. Before we analyze the performance of the congestion control mechanisms, we shortly review the most important simulation settings.

4.1 Simulation Setup

The throughput achievable in a WMN as well as the arising congestion problems depend on the network topology and the distribution of data flows. In order to analyze the WMN performance and the effect of congestion control mechanisms under varying circumstances, we use 50 randomly generated WMN snapshots. Each network snapshot consists of 40 stations which are randomly placed on a grid with length 20 m in a 500 $\times$ 500 meter square with one gateway in each square corner. An example instance is shown in Figure 5. Mesh stations are represented by diamonds, gateways by squares. The lines between the stations visualize the routing structure, whereof we explain the creation later in this section.

[1]http://www.useproject.com/, last accessed 08/2010

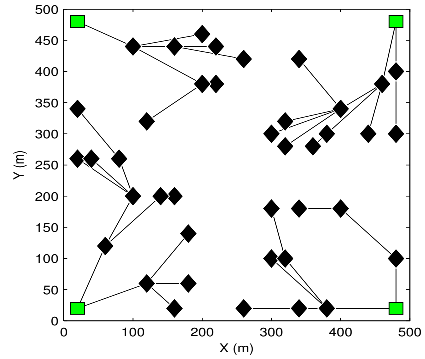


Figure 5: Exemplary Randomly Generated Network Snapshot

We consider only static environments with non-mobile stations which we assume furthermore to use the same channel. The path gain is considered to be symmetric and the background noise $P_n = -93.5$ dBm to be constant in the whole network. During the simulation, the strength of the received power, P_r, can hence be calculated as suggested in [4]:

$$P_r = P_t + K - 10\gamma \log_{10}(\frac{d}{d_0}) - \psi_{dB}. \qquad (1)$$

P_t is the transmission power which we assume to be 20 dBm for all nodes and d is the transmission distance. $K = -140.046$ dBm, $\gamma = 4$ and $d_0 = 1000$ m describe a typical mesh networking environment. The impact of shadow fading is modeled to be constant for each pair of stations and quantified by the normally distributed random variable ψ with mean zero and variance $\sigma_\psi^2 = 3.65$ dB.

The effect of adaptive modulation and coding is captured by choosing for each link the modulation and coding scheme (MCS) with the maximum data rate which guarantees a frame error rate (FER) of less than 1%. In the simulation, 8 different MCS are used, enabling data rates between 6 and 54 Mbps which are available for the IEEE 802.11-2007 OFDM PHY in the 2.45 GHz band [6]. The signal to interference and noise ratio (SINR) requirements and the maximal feasible transmission distances which allow to meet a FER of 1% when an IP packet with 1.5 kB payload is transmitted over an AWGN channel of bandwidth W = 20 MHz are obtained by link level simulations. The results from this link level simulation are also used for computing the FER for a transmitted packet in dependence of the packet length, the average SINR during the packet transmission and the used MCS.

We do not incorporate the impact of the hybrid wireless mesh routing protocol proposed by IEEE 802.11s [5] or another self-organizing routing protocol, but assume that all stations route packets according to predefined routing tables. A setting where the WMN is used as access network is assumed, i.e. there are no traffic flows between the mesh nodes, but only between the mesh nodes and the gateways which serve as interface to the Internet. The resulting routing structure consists thus of trees rooted in the gateways and is created by a routing metric which aims at maximizing the used link rates while keeping the hop count small.

The layers of the networking stack are abstracted as follows. The PHY and MAC layer are implemented according to the specifications of IEEE 802.11-2007 for 2.45 GHz OFDM. Furthermore, the distributed coordination function with CSMA/CA and RTS/CTS is used. Each station has a MAC layer buffer of 256 kB, the buffer size of the gateways is 2 MB. Mesh routing is implemented on MAC layer, the networking layer is hence empty. On transport layer, UDP and TCP New Reno are used. The application layer is abstracted by assigning a constant bit rate downlink (from the gateway to the station) data flow of 300 kbps and a constant bit rate uplink (from the station to the gateway) data flow of 100 kbps to each station if not mentioned otherwise.

An exhaustive parameter study for TCC and LSCC is the scope of our future works. For the initial performance evaluation, we use the previously introduced values for buffer management and congestion detection and the following further parameters which have shown the best performance during our studies. A congestion is detected if more than 60% of the total buffer space is occupied. During the congestion control phase, no packets should be sent. The congestion control duration is therefore considered too short if after the expiration of the timer still 20% of the buffer is occupied and considered too long, if only 10% of the buffer space is occupied by packets the station has to forward. In the first case, the congestion duration, which has been intialized to 0.1 sec is multiplied by 1.2, in the seccond case it is multiplied by 0.8. The look-ahead strategy is applied as soon as more than 40% of the buffer is occupied by packets the station has to forward. In this case, a congestion control signaling is sent to all nodes which are responsible for more than 2% of the total buffer content.

4.2 Performance of a Congested WMN

To underline the importance of intra-mesh congestion control and to illustrate how congestion wastes network resources, we examine the performance of the considered WMN topologies without congestion control is active. For this purpose, UDP is used which has no built-in congestion avoidance mechanism. As the routing topology is fixed and all data flows start at the beginning of the simulation, there is no initial transient phase and the total network throughput, i.e. the sum of all 40 uplink and downlink data flows, averaged over the simulation duration of 180 s is a suitable measure for the network performance. In addition, we use the *weighted intra-mesh packet loss* as metric. It is obtained by weighting the number of deleted bits per second by the number of transmitted hops before the deletion. If during one second, only one packet with 1.5 kB payload is dropped, which has been forwarded over two hops, this corresponds to a weighted intra-mesh packet loss of 3 kBps. All results are obtained as averages over 3 simulation runs. As we assume a static routing structure and a nearly constant traffic pattern, the degree of randomness in our simulation setup is very small and these runs are sufficient to produce statistically significant results. We therefore omit the confidence intervals which are too small to be visible.

The results of this experiment are depicted in Figure 6. On the x-axis, the 50 different considered network snapshots are sorted by increasing weighted intra-mesh packet loss. On the y-axis the total network throughput and the weighted intra-mesh packet loss are shown. Observe that the achieved throughput varies between 6.4 Mbps and 15.6 Mbps. In relation to the maximal achievable throughput of 16 Mbps (40 stations, each station has a 100 kbps uplink data flow and a 300 kbps downlink data flow), this means that in the worst case, only 40% of the traffic is transported while in the best case, about 98% of the traffic is transported. The reason for this is represented by the weighted intra-mesh data loss which ranges from zero to 8.5 Mbps.

The large variance of the intra-mesh packet loss, being an indicator for congestion, illustrates that the network snapshots are very heterogeneous. If the stations are distributed appropriately, the network throughput is large as there is nearly no congestion. In contrast to this, if the stations are distributed unsuitably, the congestion is rather large and consequently the bandwidth for most data flows

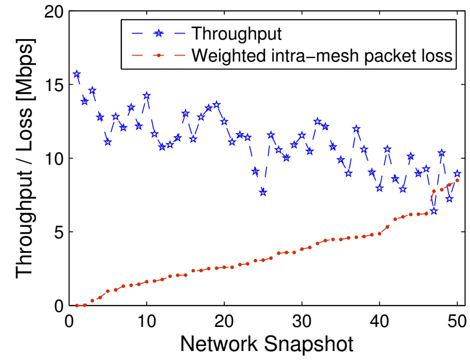


Figure 6: Throughput and Weighted Intra-Mesh Packet Loss

is not sufficient. Observe however also the zig-zag shape of the throughput curve which visualizes that the intra-mesh packet loss is not the only reason for a decreased throughput. The throughput of the two network snapshots shown at the very left, where nearly no intra-mesh packet loss occurs differs for example by nearly 2 Mbps. The reason for this is local data loss which may for instance occur if a node is connected by a very slow link only and can thus not forward the packets coming from the application layer fast enough. Note that this local packet loss phenomenon is not positively, but negatively influenced by intra-mesh congestion control as intra-mesh packet loss is avoided at the price of an increased local packet loss. The varying local packet loss is moreover another indicator for the heterogeneity of the considered topologies and responsible for the topology dependent throughput increase caused by intra-mesh congestion control. The results we show in the remainder of this section will thus illustrate, that not all congestion control strategies are always beneficial in terms of increased throughput.

4.3 Intra-Mesh Congestion Control and UDP

In this subsection, we evaluate the performance of the intra-mesh congestion control mechanisms in a WMN where UDP is used as transport layer protocol. As mentioned earlier, both TCC and LSCC, are able to avoid intra-mesh packet loss at the price for of an increased local packet loss. To find out if this price is acceptable, we compare the throughputs achieved in the 50 considered network snapshots if no congestion control, TCC, or LSCC are used. Figure 7 visualizes this comparison. On the x-axis the considered network snapshots are sorted by the throughput achieved without congestion control, the network throughput is shown on the y-axis. Observe that for the two network instances shown at the right, the throughputs achieved with or without congestion control are the same. These network snapshots are the two which can be found at the left in Figure 6, as there is no or nearly no intra-mesh packet loss. Consequently, the throughput can not be increased by intra-mesh congestion control.

Let's start the discussion of the results for the remaining topologies by analyzing the throughput achievable by using TCC. Observe that due to the method-inherent waste of bandwidth and collision control problems, this mechanism is only in some cases beneficial for the total network throughput. During our experiments we however found that for the case where the downlink is dominating which is e.g. the case for a WMN where the traffic is mainly caused by video streaming from the Internet, TCC is always suitable for increasing the network throughput. In situations like the one under study, with bidirectional traffic patterns, TCC is not always suitable for increasing the system performance. Note that the zig-zag shape of the curve representing the throughput achieved with TCC is due to this topology dependent benefits and not an indicator for instability.

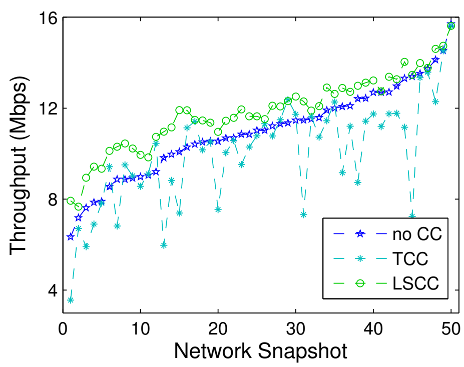


Figure 7: Effects of TCC and LSCC on UDP Traffic

As discussed earlier, LSCC has a higher degree of efficiency, the curve representing the throughput achieved with LSCC consequently shows a more smooth behavior. A comparison to the case without congestion control shows moreover that this mechanism is nearly always suitable for increasing the network throughput. A closer analysis of LSCC's effect on the individual flows reveals that it is beneficial for all uplink flows, but not for all downlink flows. Figure 8 illustrates this flow level effect. For this purpose, we show box plots representing the 40 different uplink and downlink flow throughputs resulting in the example topology shown in Figure 5 when no congestion control, TCC, and LSCC are activated. For each of the three cases, a box depicts the inter quartile range of the throughputs, a horizontal line represents the median. The whiskers are 1.5 times longer than the interquartile range. Values beyond this range are shown by crosses.

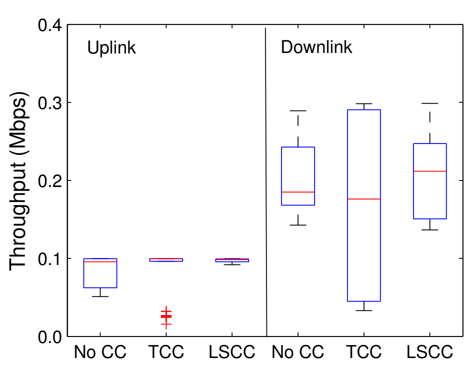


Figure 8: Flow Level Effects of TCC and LSCC and UDP

The representation for the uplink allows to see that with activated congestion control, nearly all flows achieve the desired throughput of 100 kbps and that the median of the uplink throughput flows under TCC is slightly higher than under LSCC which in turn is higher than without congestion control. The price for this is however the starvation of 8 flows under TCC, whereas LSCC achieves a more fair throughput distribution. A similar phenomenon can be observed for the downlink flows. Both congestion control mechanisms are suitable for increasing the maximal downlink throughput even if nearly no flow reaches the desired 300 kbps. Observe however that only LSCC increases the median throughput while it is decreased if TCC is active. The reason for this less beneficial effect on the downlink is that downlink packets have a link starting from the gateways as first hop. Such a link shares the channel access time which a large number of stations wanting to transmit their uplink packets to the gateway. The probability that a gateway can grab the channel for send a downlink packet on its first hop is hence significantly smaller than that a station in the middle of the WMN can send an uplink packet on the first hop. Congestion control now avoids dropping packets which have already been forwarded and makes it even more difficult for the gateways to transmit their packets as more uplink packets are sent to the gateways. Consequently even more downlink packets are blocked. This negative effect on the throughput is stronger for the case of TCC, causing the large flow throughput variance but also results in some downlink flow starvations if LSCC is active. A balanced congestion control solution integrating fairness mechanisms will hence be the topic of our future works.

4.4 Intra-Mesh Congestion Control and TCP

The previously presented results were created for UDP in order to make the effect of our mechanisms more clear. The majority of the Internet traffic is however using TCP, that is why we investigate the performance of TCC and LSCC in combination with TCP as well. During our simulation studies we found that for a scenario like the one considered in the previous section with only one uplink and one downlink TCP flow per station, the TCP congestion control is able to avoid intra-mesh packet loss and additional congestion control algorithms are hence neither required nor improving the system performance. In order to examine, if intra-mesh congestion control is reasonable in the presence of TCP, we consider a more fluctuating traffic which is more likely to occur in reality. We use the same network topologies as for the previous experiments, but divide each data flow into two new data flows, one using TCP, the other one using UDP. Additionally, the UDP data is not generated as CBR traffic, but burstily. Consequently, there is a higher degree of competition and without intra-mesh congestion control, intra-mesh packet loss occurs.

In such a case, the TCP congestion control throttles the throughput of the TCP flows in order to avoid packet loss. Consequently, nearly no UDP packets are dropped and nearly all UDP data payload may be sent which accumulates to roughly 8 Mbps. The intra-mesh congestion control can hence only avoid the loss of TCP packets. We therefore concentrate on the intra-mesh congestion control effect on the TCP flows in the remainder of this section. In Figure 9 the resulting TCP throughput without congestion control, with TCC and LSCC are shown. We use the same representation as before and sort the considered network snapshots by the throughput achievable without congestion control. As previously mentioned, the bandwidth is split between TCP and UDP and the maximal achievable throughput would be 8 Mbps. Observe first that in contrast to the experiment described in the previous section, no network achieves a throughput which is close to the feasible maximum. This illustrates that the competition is more fierce in this scenario. Again, both congestion control algorithms are able to avoid intra-mesh packet loss. If we however now compare the overall throughputs, we see that TCC is only in one of the considered scenarios suitable for increasing the throughput. LSCC in contrast leads always to an increased throughput, but its benefit is more network dependent and thus less homogeneous than in the

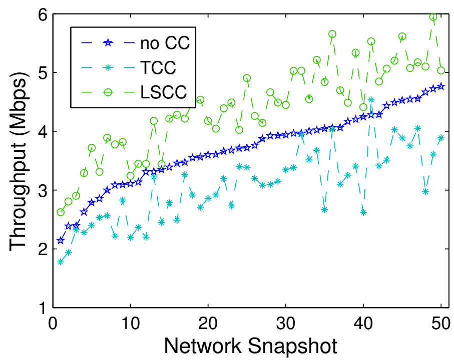


Figure 9: Effects of TCC and LSCC on TCP Traffic

case of UDP only. Additionally, the increases are smaller than in the case where UDP only is active which is of course due to the smaller absolute possible increase as the TCP flows are only half as large, but also due to the TCP inherent congestion control mechanisms.

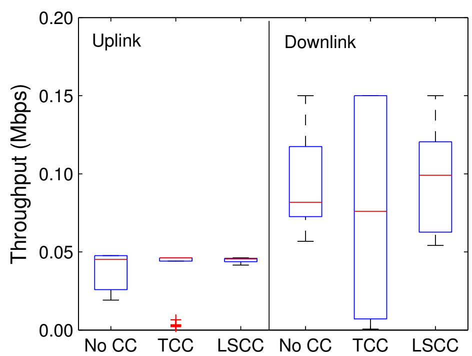


Figure 10: Flow Level Effects of TCC and LSCC and TCP

The absolute smaller effects on the throughput are also illustrated in Figure 10 where we use the same methodology as for Figure 8. It contains a flow level analysis of the intra-mesh congestion control on the TCP flows created in the topology shown in Figure 5. Observe first that similar effects as for UDP traffic occur: while the median capacity of the uplink flows is increased by both TCC and LSCC, only the latter mechanisms is advantageous for the median throughput of downlink TCP flows. LSCC results again in a higher median uplink throughput at the price of 8 starving flows. The throughput variance on the downlink is however even larger than in the UDP case which is also the explanation why the overall effect of TCC is more negative in this scenario than in the case where we do have UDP flows only.

5. CONCLUSION AND OUTLOOK

In this paper, two IEEE 802.11s compliant intra-mesh congestion control mechanisms have been introduced and evaluated. Our goal was to implement congestion control solutions following the principles of IEEE 802.11s in order to understand the opportunities and limitations of this framework. For this purpose, we designed and implemented the necessary virtual buffer management scheme, buffer monitoring, signaling and rate control mechanisms. An extensive simulation study showed that both TCC and LSCC are suitable for avoiding intra-mesh packet loss in all considered network snapshots and under different traffic patterns. In networks with more downlink data flows, both algorithms increase the network throughput. For networks with bidirectional traffic patterns, however, only the more sophisticated LSCC algorithm is beneficial in terms of increased network throughput.

We conclude that, if appropriately implemented, the IEEE 802.11s intra-mesh congestion control is suitable for increasing the WMN performance. An extensive parameter study is therefore the topic of our future works. Our studies however also showed that the positive effects of intra-mesh congestion control are heavily dependent on the network topology, the distribution of data flows and the used transport layer protocol. We moreover found that it is necessary to include fairness aspects in congestion controlling in order to prevent flow starvation. Our results also demonstrate that some congestion situations can only be solved if neighboring stations are able to exchange information going beyond a simple timer by e.g. allowing flow-selective limitations. Our future works will therefore be dedicated to analyzing the challenges and opportunities related to realizing these extensions within the framework of IEEE 802.11s congestion control and to defining appropriate extensions.

Acknowledgment

The authors would like to thank Phuoc Tran-Gia for his support and valuable comments.

6. REFERENCES

[1] U. Akyol, M. Andrews, P. Gupta, J. D. Hobby, I. Saniee, and A. L. Stolyar. Joint Scheduling and Congestion Control in Mobile Ad-Hoc Networks. *INFOCOM*, 2008.

[2] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless Networks. *ICDCS*, 1998.

[3] K. Chen and K. Nahrstedt. The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks. *WCNC*, 2004.

[4] A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[5] IEEE 802.11s/D3.02. Draft amendment to standard IEEE 802.11: ESS Mesh Networking, IEEE, May 2009.

[6] IEEE 802.11-2007. Wireless LAN Medium Access Control and Physical Layer Specifications, IEEE, June 2007.

[7] C. Lochert, B. Scheuermann, and M. Mauve. A Survey on Congestion Control for Mobile Ad Hoc Networks. *Wireless Communications and Mobile Computing*, 7(5), June 2007.

[8] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, September 2001.

[9] K. Xu, M. Gerla, L. Qi, and Y. Shu. TCP Unfairness in Ad Hoc Wireless Networks and a Neighborhood RED solution. *Wireless Networks*, 11(4), 2005.