

A Clean-Slate Architecture for Reliable Data Delivery in Wireless Mesh Networks

Sherif M. ElRakabawy and Christoph Lindemann

University of Leipzig
Department of Computer Science
Johannissgasse 26
04103 Leipzig, Germany

Abstract— In this paper, we introduce a clean-slate architecture for improving the delivery of data packets in IEEE 802.11 wireless mesh networks. Opposed to the rigid TCP/IP layer architecture which exhibits serious deficiencies in such networks, we propose a unitary layer approach that combines both routing and transport functionalities in a single layer. The new *Mesh Transmission Layer* (MTL) incorporates cross-interacting routing and transport modules for a reliable data delivery based on the loss probabilities of wireless links. Due to the significant drawbacks of standard TCP over IEEE 802.11, we particularly focus on the transport module, proposing a pure rate-based approach for transmitting data packets according to the current contention in the network. By considering the IEEE 802.11 spatial reuse constraint and employing a novel acknowledgment scheme, the new transport module improves both goodput and fairness in wireless mesh networks. In a comparative performance study, we show that MTL achieves up to 48% more goodput and up to 100% less packet drops than TCP/IP, while maintaining excellent fairness results.

Keywords: Congestion control, IEEE 802.11, TCP, Wireless network protocols

I. INTRODUCTION

Since the emergence of the IEEE 802.11 standard [8], wireless networks have become enormously popular. The freedom of wireless Internet access provided by IEEE 802.11 has been mushrooming the number of wireless hotspots around the globe as a rapidly rising trend. Besides conventional single-cell wireless hotspots, wireless mesh networks ([1], [11]) comprising multiple IEEE 802.11 nodes have gained increased attention in recent years, both in academia and industry. Such networks typically aim to provide cost-efficient Internet access with minimal infrastructure expenditure, which makes it particularly attractive for suburban areas with little or no broadband availability. Mesh nodes, which are typically wireless routers mounted on buildings, form a multihop backbone to forward packets hop-by-hop between the Internet and other mesh nodes. Mesh participants with mesh gateways, which have direct access to the Internet, can share it with other participants.

Within this context, wireless mesh networks possess unique characteristics compared with the wired Internet, raising key challenges which must be addressed for achieving a reliable operation of such networks. Opposed to wired networks, in IEEE 802.11 networks, the wireless channel is a scarce resource shared among nodes within their radio range. Furthermore, channel capture, hidden and exposed terminal

effects, and the IEEE 802.11 medium access control constitute features of wireless mesh networks not present in a wired IP network. Along with the unique characteristics of the wireless channel, mobility-related problems can occur when mobile mesh clients move at a certain speed, resulting in dropped packets at link layer. In such a case, standard TCP interprets the packet loss as a sign of congestion and reduces throughput.

One of the key causes of such deficiencies in wireless mesh networks is the fact that the current TCP/IP protocol stack has been originally designed for wired IP networks. Specifically, the strict layered architecture of the TCP/IP stack has proven to be inflexible due to the exclusive restriction of crucial information about the network on the respective layer. While assigning a special service to each layer reduces the complexity of the network design, other layers are perceived as black boxes, holding critical information about the network state. Such inaccessible information is required by all layers to adapt to the special characteristics of wireless mesh networks such as route breaks and lossy links.

Due to such fundamental deficiencies in the basic design of the classic TCP/IP stack, it seems more reasonable to consider designing a new network stack from scratch rather than improving the existing one incrementally. Such a trend, which is referred to as *clean-slate* network design, has been becoming increasingly popular within the research community ([13], [17], [18]). To this end, we propose a clean-slate architecture for reliable data delivery in wireless mesh networks. Thereby, we focus on the routing and transport functionalities within the network stack in order to avoid modifications of the widely deployed IEEE 802.11 protocol. Opposed to the rigid TCP/IP stack, we design a unitary layer architecture, in which the routing and transport functionalities are both merged in a single layer. The new layer, which is denoted as the *Mesh Transmission Layer* (MTL), incorporates a routing and a transport module which aim to achieve reliable packet delivery using feedback information from the network. We particularly focus on the aspect of congestion control within our newly designed Mesh Transmission Layer. Opposed to classic TCP, the new transport module operates purely rate-based, omitting the congestion window which has been criticized in several works ([4], [9]) for causing unfairness between competing flows. Besides the fairness problem, the new transport module also considers the spatial reuse constraint of IEEE 802.11, which significantly degrades the end-to-end goodput over multiple hops. Similar to [3], the new layer MTL determines the quality of the wireless links based on their loss

probabilities. Such loss probabilities are then adopted for both the routing and transport modules. While the routing module relies on such probabilities to determine the best route to a destination, the transport module integrates them for determining the optimum transmission rate of data packets. In a comparative performance study, we show that MTL achieves up to 48% more goodput and up to 100% less packet drops than TCP/IP while maintaining excellent fairness results.

The remainder of this paper is organized as follows. Section II summarizes related work on challenges in wireless mesh networks, specifically considering the default TCP/IP stack. In Section III we introduce our clean-slate architecture for reliable data delivery, while providing a comprehensive performance study of MTL versus the standard TCP/IP stack in Section IV. Finally, concluding remarks are given.

II. RELATED WORK

Although most of the research in the networking field still builds on the classic TCP/IP architecture, the trend is increasingly moving towards the clean-slate approach. In [13], Feldmann gave an overview on the challenges that must be addressed for a reliable future Internet, and discussed the clean-slate approach as a possible solution. Scott et al. [15] addressed the problem of the current TCP/IP architecture in infrastructure-less mobile environments. The authors introduced a non-layered mobile architecture in pocket switched networks to support reliable communication in mobile environments. In [9], the authors proposed ATP, a transport protocol specially tailored for multihop wireless networks. ATP employs pure rate-based transmission of packets, where the transmission rate is determined using feedback from intermediate nodes along the path. In contrast to [15], we focus on the reliable data delivery through effective congestion control in static wireless mesh networks. Opposed to [9], our approach considers a unitary layer architecture rather than an autonomous transport protocol.

Several congestion control mechanisms have been proposed for wireless mesh networks. In [6], Fu et al. pointed out the hidden terminal problem in wireless multihop networks and experimentally showed that for a chain topology the optimal windows size, for which TCP achieves best throughput, is roughly given by 1/4 of the hop count of the path. Gambiroza et al. [7] studied TCP performance and fairness in wireless mesh networks comprising numerous wireless relay nodes (there called Transit Access Points, TAPs) and a connection to the wired Internet. They introduced TAP-fairness to characterize the idealized goodput and fairness objective for such networks, and proposed a distributed link layer algorithm for achieving TAP-fairness among active TCP flows. Shi et al. [12] addressed the unfairness problem which occurs between one-hop flows contending with two-hop flows for gateway access. The authors proposed a solution based on a contention window policy in IEEE 802.11e. Chen et al. [14] proposed a cross-layer approach for congestion control, routing, and scheduling in multihop wireless networks. The authors proposed an algorithm which is based on a utility maximization problem with predefined rate and scheduling constraints. Our work builds on findings in [4], in which TCP with Adaptive Pacing has been introduced. The results showed that adaptive pacing yields significant performance improvement with

respect to standard TCP. Opposed to previous approaches for congestion control, we propose a novel unitary layer architecture in which we combine routing and rate-based congestion control rather than incrementally building on the existing TCP congestion control algorithm.

Path metrics based on link qualities have been studied in the context of routing protocols for wireless mesh networks ([3], [16]). In [3], the authors introduced the Expected Transmission Count (ETX) as a new metric for multihop wireless networks. The metric considers the packet delivery probability at each link and determines the path with the minimum loss rate and highest throughput. Draves et al. [16] introduced the Expected Transmission Time (ETT) metric, which considers the bandwidth of a link along with its ETX weight. In this paper, we incorporate the ETX metric in our Mesh Transmission Layer. Specifically, we use ETX measurements both for routing decision as well as for adjusting the transmission rate of data packets according to the current contention on the path.

III. CLEAN-SLATE DATA DELIVERY

A. Unitary Layer Architecture

As discussed in [13] and [17], the classic TCP/IP layer model faces significant challenges coupled with future Internet technologies such as wireless mesh networks. In particular, the TCP/IP layer architecture is inflexible due to the restriction of useful information on certain layers, and does not consider the unique characteristics of IEEE 802.11 such as the spatial reuse constraint [6] and unfairness problems [10]. Specifically the congestion control algorithm of TCP at layer 4 possesses serious deficiencies over IEEE 802.11 mesh networks, since it considers each network as a black box and thus provokes packet loss to identify spare bandwidth. Thus, in this paper, we focus on congestion control and introduce a novel transport module in Section III.C that considers the unique challenges of IEEE 802.11.

Due to such deficiencies associated with the classic TCP/IP architecture we introduce a new unitary layer design, in which we merge the transport and routing functionalities into a single *Mesh Transmission Layer (MTL)*. As shown in Figure 1, routing and transport are embedded as modules within MTL. Such unitary architecture enables embedded exchange of feedback information of the network between both modules rather than considering the network as a black box. The background of such a merge is the fact that the performance of both routing and transport over IEEE 802.11 mesh networks depend on the same key factor, which is the quality of the wireless links. By sharing information between routing and transport on the current condition of wireless links, a more reliable end-to-end data delivery can be achieved. Similar to

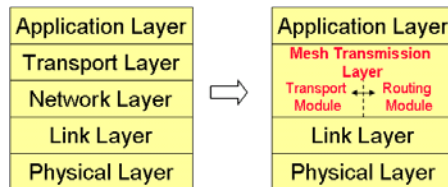


Figure 1. The Mesh Transmission Layer comprising a transport and a routing module

[3], link quality is determined by measuring the packet loss probability at each link. The routing module uses information on link quality for route determination and simultaneously shares such information with the transport module, which adjusts its packet transmission rate accordingly. On the other hand, the transport module provides the routing module with information on end-to-end connections to maintain information on accessibility of nodes in the routing table.

Subsequently, we discuss the detailed operation of the routing and transport modules and how they interact to achieve reliability. Thereby, we focus on the novel congestion control algorithm employed in the transport module since the standard TCP congestion control accounts for the most significant performance deficiencies in IEEE 802.11 mesh networks ([4], [7]).

B. Routing Module

The routing module implements proactive route determination similar to the Optimized Link State Routing Protocol (OLSR) [2]. It uses HELLO and Topology Control (TC) messages to proactively discover and disseminate link state information throughout the mesh network. HELLO messages are broadcasted to determine 1-hop and 2-hop neighbors and add them to the routing table. TC messages are propagated throughout the network to disseminate such neighbor information. Besides determining and maintaining the best routes between mesh nodes in a network, the routing module within the Mesh Transmission Layer is also responsible for providing the transport module with crucial information on the quality of the wireless links.

The quality of the wireless links, which is used for route maintenance and shared with the transport module, is determined by computing the packet delivery probability at each link. Such probability is derived by calculating the expected transmission count (ETX) [3] at each link:

$$ETX = \frac{1}{d_f \cdot d_r}, \quad (1)$$

where d_f and d_r denote the forward and reverse packet delivery ratios, respectively. The forward delivery ratio describes the probability that a packet is successfully forwarded over a link from sender to receiver, whereas the reverse delivery ratio describes the probability that this packet is correctly acknowledged. Such delivery ratios are determined using probe packets of a fixed size that are broadcasted by each node in fixed time intervals. Each node remembers the number of probe packets it correctly receives at each link within the last x seconds, and calculates the ratio between the received and the total number of probe packets sent accordingly. The ETX of a route comprises the sum of the ETX values of all links on the route.

Along with the ETX metric, the routing module also relies on information from the transport module on end-to-end connections for quick updates of the routing table. Specifically, as the transport module is responsible for maintaining end-to-end transport connections between end hosts, it notifies the routing module about the accessibility of current connected peers. Hence, in case a peer is not responding, the routing module is notified, and the corresponding peer is temporarily labeled as inaccessible for data exchange. In case TC messages validate that the peer is not part of the network, it is deleted

from the routing table. Such an inaccessibility label is helpful to inform the user/application about the exact status of other peers.

C. Transport Module

The transport module within the Mesh Transmission layer is responsible for a reliable data transport between end-to-end hosts in the mesh network. The new module overcomes the well-known deficiencies of classic TCP in IEEE 802.11 multihop wireless networks, which arise from TCP's congestion control algorithm. First, TCP's window-based congestion control leads to packet bursts when received acknowledgments trigger the transmission of several data packets, e.g., when receiving a cumulative ACK. Due to the spatial reuse constraint of the wireless channel in IEEE 802.11 multihop wireless networks [6], neighboring nodes cannot transmit simultaneously. Thus, packet bursts result in increased contention on the wireless channel. This link layer contention may lead to packet drops due to the hidden and exposed terminal problems [6]. Second, TCP's congestion control algorithm relies on packet losses as indication of congestion and, thus, provokes losses in order to identify spare bandwidth. In IEEE 802.11 multihop wireless networks, this behavior results in increased congestion, causing significant performance degradation for TCP [10].

To overcome the deficiencies stated above, we propose a purely rate-based congestion control algorithm rather than employing window-based mechanisms. The adaptive transmission rate accounts for the spatial reuse constraint of IEEE 802.11 and proactively identifies incipient congestion, i.e. before congestion-related losses actually occur. The novel transport module is packet-based and uses sequence numbers to maintain in-order packet delivery. Since it operates purely rate-based, neither a congestion window nor related window-adjustment algorithms such as slow start and congestion avoidance are required. *Explicit Gap Notifications* are employed in the new transport algorithm to achieve reliability and save bandwidth.

1) *Explicit Gap Notifications*

A key feature of the new transport module is that it uses *Explicit Gap Notifications* (EGNs) rather than conventional acknowledgments to achieve reliability. EGNs are packets transmitted from receiver to sender in order to explicitly indicate gaps within the data packet flow rather than sending a positive acknowledgment for each correctly received packet. A single EGN packet is cumulative since it can report more than one undelivered data packet. This saves a fair amount of costly bandwidth and improves end-to-end goodput.

Using the EGN mechanism, the sender maintains a sending buffer of a fixed size, 200 kbytes by default, in which it saves the packets already transmitted. The reason for such a buffer is to be able to retransmit dropped packets upon receiving one or more EGNs. In case the sending buffer gets full while no EGN has been received within the lifetime of the current buffer, the sender requests an explicit EGN by purposely holding back a data packet. Once the requested EGN has been received, the sender transmits the held packet and empties the entire buffer. The purpose of such explicit request is to make sure that the receiver is still alive and replying, and that all previous data packets have been successfully received.

2) Rate-based Congestion Control

a) Considering Link Contention

The classic TCP congestion control algorithm saturates the link by increasing the load issued into the network until a packet loss is detected, where such packet loss identifies congestion. Upon congestion, the transmission rate is throttled to empty overfilled queues on the routers and is then increased again until a new packet loss is detected and so forth. Considering the characteristics of IEEE 802.11 multihop networks, it becomes obvious that a transport protocol which actually provokes packet drops to get network feedback has to suffer from poor performance. Thus, our congestion control algorithm identifies high contention on the network path of the transport connection and proactively throttles the transmission rate before losses occur.

In order to identify contention on the path between sender and receiver, we consider the packet loss probability at each link on the path. Such loss probability is directly correlated to the level of contention at the wireless links and is thus a reliable measure. First we consider the loss probability of a wireless link, p_{link} , as a function of the forward and reverse loss probabilities p_f and p_r , respectively:

$$p_{link} = p_f + (1 - p_f)p_r \quad (2)$$

In other words, a packet is considered lost in case either the packet itself is dropped on the forward path (i.e. p_f), or the packet is successfully transmitted but the corresponding link layer acknowledgment is dropped on the reverse path (i.e. $(1 - p_f)p_r$). To prevent a redundant computation overhead, we derive p_{link} from ETX:

$$\begin{aligned} p_{link} &= p_f + (1 - p_f)p_r \\ &= (1 - d_f) + (1 - (1 - d_f))(1 - d_r) \\ &= 1 - d_f d_r \\ &= 1 - \frac{1}{ETX} \end{aligned} \quad (3)$$

In order to account to the loss probability of a given path between a sender A and a receiver B , we consider the loss probability at the bottleneck-link of the path, i.e. the link with the highest loss probability. We do so to adjust the transmission rate at the sender according to the most congested link on the path in order to avoid packet drops and degraded goodput. We define the maximum loss probability $P_{<A,B>}^{\max}$ on a path of i links as:

$$P_{<A,B>}^{\max} = \max_i \left(1 - \frac{1}{ETX_i} \right), \quad (4)$$

where ETX_i denotes the ETX value at link i . In Subsection C.2.C we show how $P_{<A,B>}^{\max}$ is considered for the computation of the final packet transmission rate.

b) Considering the Spatial Reuse Constraint

Besides the measure of contention on the network path, the derivation of an appropriate transmission rate should also account for the spatial reuse constraint of IEEE 802.11 multihop wireless networks [6]. That is, due to the hidden terminal effect, in a chain topology where the transmission range of each node is about 250m and the interference and

carrier sensing ranges are 550m, a TCP sender at node i can only transmit a packet successfully as soon as node $(i+3)$ has finished its transmission in order to avoid collisions. We refer to the time elapsed between transmitting a TCP packet by node i and receiving the packet at node $(i+4)$ as the *4-hop propagation delay* (FHD). Such hidden terminal effects depend mainly on the characteristics of the network interface as well as the adopted routing protocol. First, the network interface determines the ratio between the transmission range and the interference range. Due to the settings of the network interface considered in this paper, hidden terminals along the path can be avoided if a transmitting node delays the transmission of a data packet until the previously sent packet is forwarded 4 times. Thus, we consider FHD for the calculation of the transmission rate.

If we assume a network with perfect scheduling at link layer, the maximum spatial reuse with minimum collisions would be achieved with a transmission rate $R_{max} = 1/FHD$. Following [6], an upper bound for the capacity of a path with h hops in an IEEE 802.11 multihop wireless network is given by $h/4$ packets. Let $T_{one-way}$ denote the time a packet traverses from the sender to the receiver. This quantity can be computed as $T_{one-way} = FHD \cdot h/4$. Subsequently, the number of packets in flight on the way from sender to receiver with a sender's transmission rate of R_{max} , is given by:

$$\# \text{ packets in flight} = R_{max} \cdot T_{one-way} = \frac{1}{4}h \quad (5)$$

Thus, the number of packets in flight transmitted with the maximum transmission rate R_{max} reflects the maximum capacity of the network path. Note that for network paths with $h < 4$, R_{max} is computed using the h -hop propagation delay instead of the 4-hop propagation delay. Without loss of generality and for ease of exposition, we consider network paths with $h \geq 4$ in the subsequent discussion.

In order to use R_{max} as an upper bound for the transmission rate, we need to measure the 4-hop propagation delay FHD of the data packets. To prevent extra control traffic overhead, we estimate FHD based on end-to-end round-trip-times (RTT) measurements at the sender. The RTT is composed of the sum of the delay experienced by the data packet on the way from the sender to the receiver and the delay experienced by the Explicit Gap Notification (EGN) packet sent from the receiver to the sender. Each of these delays comprise the time to forward the packet over h hops, where each forwarding requires a queuing delay t_q and transmission delays t_{data} , and t_{EGN} , respectively. The link layer retransmissions and backoff are implicitly considered in the queuing delay t_q . Using the measured RTT, we get:

$$RTT = h(t_q + t_{data}) + h(t_q + t_{EGN}) \quad (6)$$

Solving for t_q while using $t_{data} = s_{data}/b$ and $t_{EGN} = s_{EGN}/b$ for a bandwidth b and data/EGN packet sizes s_{data} and s_{EGN} , we derive the average queuing delay as:

$$t_q = \frac{1}{2} \left(\frac{RTT}{h} - \frac{s_{data} + s_{EGN}}{b} \right) \quad (7)$$

Subsequently, we can estimate the 4-hop propagation delay of the transport data packet:

$$FHD = 4 \left(t_q + \frac{S_{data}}{b} \right) = 2 \left(\frac{RTT}{h} + \frac{S_{data} - S_{EGN}}{b} \right) \quad (8)$$

This estimation requires that the sender knows about the number of hops on the network path to the receiver and the bandwidth of the wireless network interface. Since this information is available from the routing module and the link layer, no extra overhead is required.

c) Computing the Transmission Rate

Since the computation of the adaptive transmission rate should account for both the current contention on the network path and the spatial reuse constraint, we incorporate the maximum path loss probability $P_{<A,B>}^{\max}$ and FHD in the transmission rate formula. Recall that a rate of $R_{max}=1/FHD$ specifies an upper bound for the achievable goodput under optimal conditions, i.e. with theoretically perfect scheduling and no contention. In order to adaptively throttle the transmission rate R according to the current degree of contention, we use $P_{<A,B>}^{\max}$ as an additional decay factor:

$$R = \frac{1}{\overline{FHD} \cdot (1 + P_{<A,B>}^{\max})} \quad (9)$$

In favor of a stable transmission rate, we average the measured 4-hop propagation delay samples using the exponentially weighted moving average (EWMA) with averaging weight α :

$$\overline{FHD} = \alpha \cdot \overline{FHD}_{old} + (1 - \alpha) \cdot FHD \quad (10)$$

As validated by our simulations, a suitable value for α has proven to be 0.7.

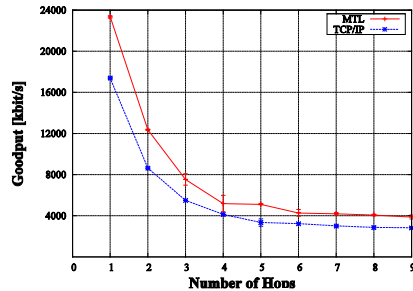


Figure 2. Chain topology: Goodput vs. number of hops

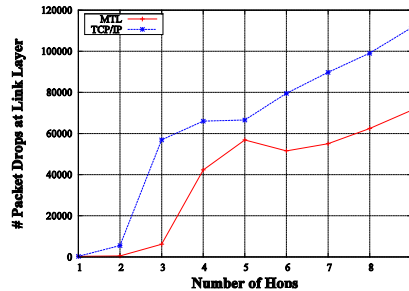


Figure 3. Chain topology: Number of packet drops at link layer vs. number of hops

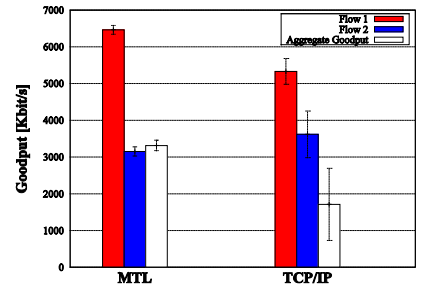


Figure 4. Parallel chains topology: Goodput and fairness

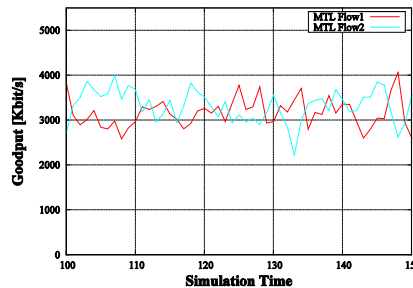


Figure 5. Parallel chains topology: Goodput vs. simulation time for MTL

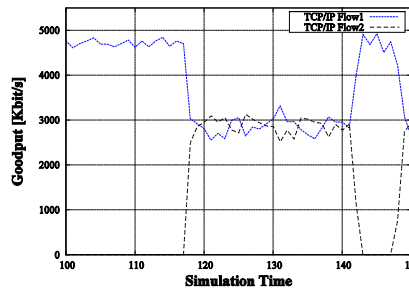


Figure 6. Parallel chains topology: Goodput vs. simulation time for TCP/IP

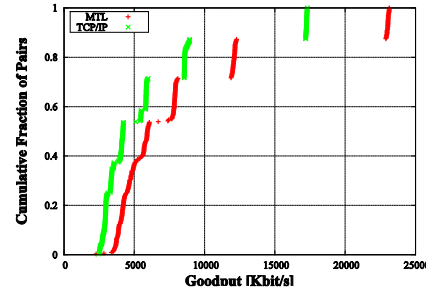


Figure 7. Grid topology: Cumulative distribution function (CDF) of the goodput between each pair in the network (Average MTL: 8,755 kbit/s; Average TCP/IP: 6,386 kbit/s)

IV. PERFORMANCE EVALUATION

To evaluate the performance of our Mesh Transmission Layer (MTL), we conduct a performance study using ns-2 [5], in which we compare the performance of MTL with the classic TCP/IP stack. As a transport protocol for the TCP/IP stack we deploy the widely used TCP NewReno, whereas for routing the standard OLSR protocol with ETX support is adopted. In all experiments, except for experiments showing transient behavior, we conduct steady-state simulations starting with an initially idle system. In each run, we utilize FTP connections until 55,000 packets are successfully transmitted, and split the output of the experiment in 11 batches, each 5,000 packets in size. The first batch is discarded as initial transient. The considered performance measures are derived from the remaining 10 batches with 95% confidence intervals by the batch means method.

In ns-2, all link-layer parameters of IEEE 802.11 are configured to provide a transmission range of 250m and a carrier sensing range as well as an interference range of 550m. The RTS/CTS handshake is disabled and we consider a channel bandwidth of 54 Mbit/s according to IEEE 802.11g while setting the size of transport data packets (both for TCP and MTL) to 1,460 bytes.

A. Chain Topology

The first topology we consider is an equally spaced chain comprising $h+1$ nodes (h hops) with a single FTP flow and a 200m inter-node distance. TCP packets traverse along the chain from the leftmost node (i.e., the source) to the rightmost node (i.e., the destination). Nodes in the chain are positioned such that only direct neighbors can communicate with each other over one hop. Figures 2 and 3 show the goodput as well as the number of packets dropped at link layer versus number of hops for MTL and TCP/IP, respectively. Figure 2 shows that MTL achieves up to 48% more goodput than TCP/IP (e.g. at 2 hops).

The reason for such goodput improvement can be explained by considering the number of packet drops at link layer in Figure 3. There we see that MTL causes up to 100% less packet drops than TCP/IP (e.g. at 2 hops) due to its less aggressive packet transmission which adapts to the network load and decreases hidden terminal effects.

B. Parallel Chains Topology

To evaluate the fairness of MTL versus TCP/IP we consider a parallel chains topology comprising two parallel chains, each consisting of 6 nodes with a 200m inter-node distance. The distance between both chains is 400m, which means that they lie within each other's interference range. An FTP flow runs on each of the chains from the leftmost node (i.e., the source) to the rightmost node (i.e., the destination). In this experiment, we evaluate the goodput and fairness between flow 1 running on the upper chain and flow 2 running on the lower chain for MTL and TCP/IP, respectively.

Figure 4 shows the goodput achieved by each flow as well as the aggregate goodput over both. We can observe that while MTL shares the available bandwidth equally between both flows, TCP/IP favors flow 1 which achieves over 100% more goodput than flow 2. Besides the nearly-optimal fairness, MTL also achieves around 23% more aggregate goodput than TCP/IP. In figures 5 and 6 we evaluate the transient goodput of the FTP flows between the 100th and the 150th second of the simulation for MTL and TCP/IP, respectively. We choose such an interval in the middle of the simulation to ensure that both FTP flows have already reached their steady-states. While MTL lets both flows share the available bandwidth equally (Figure 5), the TCP/IP flows fluctuate over time, resulting in periods where flow 1 acquires all of the available bandwidth at cost of flow 2 (Figure 6). This validates that the adaptive transmission rate of MTL, which accounts for the current contention in the network, is superior to the aggressive window strategy of standard TCP.

C. Grid Topology

As a more complex topology, we consider a 6x6 grid (36 nodes) with multiple FTP flows. Specifically, we utilize an FTP connection between each pair in the network. This simulates a scenario where multiple users perform several downloads back to back. Figure 7 shows the cumulative distribution function (CDF) of the corresponding individual goodput. As we can observe, nodes deploying MTL achieve higher goodput values than nodes deploying TCP/IP. Especially for high goodput values does the performance gap between MTL and TCP/IP increase. In particular, while MTL achieves an average goodput of 8,755 kbit/s, TCP/IP only achieves 6,386 kbit/s. This is an improvement of around 37% in favor of MTL.

V. CONCLUSION

We introduced the Mesh Transmission Layer (MTL), a novel clean-slate architecture for reliable data delivery in wireless mesh networks. The new layer merges both the routing and transport functionalities and considers the special characteristics of IEEE 802.11 through effective cross-feedback from the network and adaptive rate-based packet

transmission. We particularly focused on the transport module of MTL, implementing a window-free algorithm that adapts the packet transmission rate according to the current loss probability as well as the spatial reuse constraint of a given path. Through an explicit notification scheme for indicating lost packets, end-to-end reliability is achieved with minimal overhead.

A comparative performance study with MTL versus the classic TCP/IP architecture using several network topologies showed that MTL achieves up to 48% more goodput and up to 100% less packet drops. Furthermore, while TCP/IP exhibits serious fairness drawbacks, MTL shares the available bandwidth equally among active flows contenting for the wireless channel. For future work, we are integrating MTL into Linux in order to evaluate its performance in a large-scale mesh network.

REFERENCES

- [1] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, Architecture and Evaluation of an Unplanned 802.11b Mesh Network, *Proc. ACM MOBICOM*, Cologne, Germany, 2005.
- [2] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol, RFC 3626, <http://www.ietf.org/rfc/rfc3626.txt>, October 2003.
- [3] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, A High-Throughput Path Metric for Multi-Hop Wireless Routing, *Proc. ACM MOBICOM*, San Diego, CA, 2003.
- [4] S. ElRakabawy, A. Klemm, and C. Lindemann, TCP with Adaptive Pacing for Multihop Wireless Networks, *Proc. ACM MobiHoc*, Urbana-Champaign, IL, 2005.
- [5] K. Fall and K. Varadhan (Ed.), The ns-2 Manual, Technical Report, The VINT Project, UC Berkeley, LBL, and Xerox PARC, 2007.
- [6] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, The Impact of Multihop Wireless Channel on TCP Performance, *IEEE Transactions on Mobile Computing*, Vol. 4, Issue 2, March 2005.
- [7] V. Gambiroza, B. Sadeghi, and E. Knightly, End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks, *Proc. ACM MOBICOM*, Philadelphia, PA, 2004.
- [8] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11, August 1999.
- [9] K. Sundaresan, V. Anantharaman, H-Y. Hsieh, R. Sivakumar, ATP: A Reliable Transport Protocol for Ad Hoc Networks, *Transactions on Mobile Computing*, Vol. 4, Issue 6, November 2005.
- [10] K. Xu, M. Gerla, L. Qi, and Y. Shu, TCP Unfairness in Ad Hoc Wireless Networks and a Neighborhood RED Solution, *Wireless Networks*, Vol. 11, Issue 4, 2005.
- [11] I. Akyildiz, X. Wang, and W. Wang, Wireless mesh networks: a survey, *Elsevier Computer Networks*, 47, 2005.
- [12] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. Knightly, Measurement and Modeling of the Origins of Starvation in Congestion Controlled Mesh Networks, *Proc. IEEE INFOCOM*, Phoenix, AZ, 2008.
- [13] A. Feldmann, Internet Clean-Slate Design: What and Why, *ACM SIGCOMM Computer Communication Review*, 37, 3, 2007.
- [14] L. Chen, S. Low, M. Chiang, and J. Doyle, Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks, *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006.
- [15] J. Scott, P. Hui, J. Crowcroft, and C. Diot, Huggle: A Networking Architecture Designed Around Mobile Users, *Proc. IFIP WONS*, Les Menuires, France, 2006.
- [16] R. Draves, J. Padhye, and B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, *Proc. ACM MOBICOM*, Philadelphia, PA, 2004.
- [17] The Global Environment For Network Innovations Project (GENI), <http://www.geni.net/>.
- [18] Future INternet Design (FIND), <http://find.isi.edu/>.