Model validation through experimental testbed: the fluid flow behavior example

Adel Aziz
EPFL
School of Computer and Communication Sciences
Lausanne, Switzerland
adel.aziz@epfl.ch

Thomas Huehn
Deutsche Telekom Laboratories
TU Berlin
10587 Berlin (Germany)
thomas@net.t-labs.tu-berlin.de

Roger Karrer
Deutsche Telekom Laboratories
TU Berlin
10587 Berlin (Germany)
roger.karrer@telekom.de

Patrick Thiran
EPFL
School of Computer and Communication Sciences
Lausanne, Switzerland
patrick.thiran@epfl.ch

ABSTRACT

This testbed practice paper presents our efforts to validate an analytical model for fluid flow behavior in wireless mesh networks with an experimental evaluation. We have developed a fluid model for multihop communication in wireless mesh networks and analyzed it with simulations. Now, we describe our efforts to reproduce the modeled and simulated network with an indoor WiFi mesh network and to measure flow parameters that allow us to verify that the underlying assumptions and the flow behavior can be matched in real networks. Our experiences emphasize the need to gap the bridge between simulations and experimental validation as well as the lack of tools to efficiently validate results. These findings are particularly true in wireless mesh networks where interference is beyond the control of the experiment and where nodes are distributed such that an easy coordination and monitoring of the nodes is not possible.

1. INTRODUCTION

Wireless mesh networks have the potential to revolutionize the way people experience the Internet. Entire cities and communities have already realized or are about to deploy wireless mesh networks to enable ubiquitous high-speed Internet access using off-the-shelf hardware ([1, 2, 3, 4]).

Unfortunately, the traffic characteristics of WiFi mesh networks are still ill understood. While models are available that describe the access of a single access point [6], the characterization of multihop wireless traffic is still in its infancy. The traffic characterization over multihop wireless is particularly important because the backhaul (cfg. Figure 2) of a mesh transports data over multiple wireless hops. The efficiency of the backhaul is crucial for the performance of the entire WMN because the backhaul aggregates the traffic of multiple users in the mesh and transports it from to user to the Internet and back.

TRIDENTCOM 2008, 17th – 20th Mar 2008, Innsbruck, Austria.

DOI 10.4108/icst.tridentcom.2008.10411

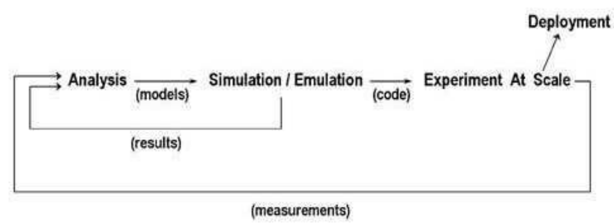

Figure 1: Progress requires a full cycle

The theme of our work is to enhance our understanding of wireless mesh networks by developing a novel model for multihop flow behavior and by validating the model in an experimental testbed. We hereby emphasize the need to complete a full cycle of progress, consisting of analysis, simulation and experimental evaluation as depicted in Figure 1 to make true progress towards better understand WMNs. For the experimental evaluation, we are relying on the indoor mesh that is part of the Magnets testbed [12].

The goal of this paper is to describe our experiences with validating a previously developed model for flow behavior in multihop wireless networks in a testbed. In particular, we have developed an analytical model that captures the flow behavior of multihop wireless communication using a fluid model approach and validated the model with simulations [5]. Thus, a testbed implementation and evaluation is needed to complete the progress cycle. The main challenge thereby is to build a testbed to validate the model. The design questions include: what topology is suited; which hard- and software should be used; what monitoring infrastructure is needed in terms of devices and software tool chain to process the monitored data and to verify the results? We anticipate that the experience and the lessons learned have influenced the deployment of the indoor testbed.

This paper is organized as follows. Section 2 provides a brief overview of the traffic engineering challenges. Next, Sections 2.2

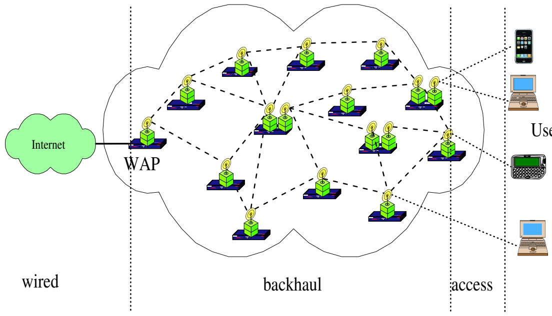

Figure 2: A wireless mesh network consists of a backhaul and an access part

and 2.3 describe our multihop traffic model to address the above challenges and simulation results. Then, Section 3 describes the testbed experiences. After discussing related work in Section 5, Section 6 concludes our paper.

2. BACKGROUND

This section provides a brief overview of wireless mesh networks, with the focus on the efficiency of the multihop wireless backhaul. Then, we present the previously developed model and the simulation results to improve efficiency in the backhaul. Details on the model and the simulation results can be found here [5].

2.1 Problem statement

Wireless mesh networks consist of two parts as depicted in Figure 2: an *access part* that provides connectivity between a mesh node and a user, and a *backhaul network* that transports data over multiple wireless mesh nodes from the access node to a wireless mesh node that is equipped with a fixed network line and that we term Wired Access Point (WAP). The remaining mesh nodes are without wired connection to the network and we denote them as Transit Access Points (TAPs). Using modern mesh nodes that support multiple WiFi cards, these two network parts typically run on orthogonal frequencies to avoid extensive interferences.

The logical topology of a backhaul is typically arranged as a tree, with the wired mesh node at its root. If there are multiple wired mesh nodes in the network, the logical topology can be split into different parts, each with its separate trees. For simplicity reasons, we consider only one branch of a single tree in our model, as depicted in Figure 3, assuming that the different branches are sufficiently far apart from each other that interference among the branches is negligible. This simplification may not be perfectly achievable in practice, but by using different channels for each branch, the assumption can be sufficiently approximated.

The main challenge of the backhaul network is to transport data as efficiently as possible from a user to the wired mesh node and back. A number of challenges have to be addressed, including fairness and performance [13]. For example, Gambiroza et al. [11] have shown that multihop networks exhibit severe unfairness as the throughput of a user drastically degrades with the distance from the wired mesh node. In this paper, we focus on maximizing end-to-end throughput, minimizing delays and packet loss as performance metrics.

2.2 Model

To efficiently meet the backhaul objectives, we argue that the nature of the packet flow through the network plays a key role. In particular, we advocate that the flow behavior should follow fluid

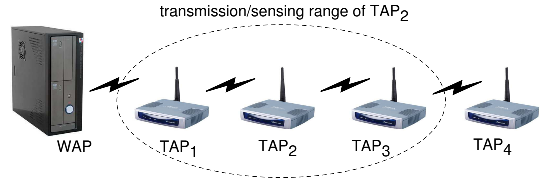

Figure 3: Simplified topology of the backhaul

physics terminology in that flows through the backhaul should be *laminar*. A laminar flow is characterized by a smooth propagation of packets through the network, where every packet only spends a negligible time in the mesh node buffers. To motivate this argument, consider the analogy of vehicular traffic: along a street with multiple traffic lights (which correspond to the mesh nodes), the total "throughput", "delay" and "loss" is most efficient if cars can cruise at constant speed, i.e. when the traffic lights are arranged in sequence such that a single car rides on a green wave. The same argument hold for wireless mesh networks: laminar flows will improve the stability of TCP-based flows as well as the quality of delay-sensitive applications such as VoIP and multimedia applications.

In contrast, we find that the current 802.11 protocol with its random access and backoff mechanism leads to *turbulent* flow behavior. We refer to turbulent flows when packets spend a significant and arbitrary time in the mesh nodes. There are several drawbacks when flows are turbulent. First, buffer management is difficult: buffer sizes should be increased to avoid packet loss and therefore retransmissions, but large buffers also undesirably increase delay, especially when the delay of multiple buffers is accumulated. As a result, end-to-end protocols such as TCP that rely on delay measurements fail to perform efficiently, and VoIP flows inherit undesirable delay variations.

Our analysis shows that the random access mechanism is not suited for multihop backhaul networks. In fact, random access should provide a fair resource access to all competing nodes within range. However, due to the particular multihop topology of the backhaul network, the mesh nodes experience a severe imbalance in their ability to access the medium. As a result, some mesh nodes start to rapidly build up their queue and eventually drop packets while other nodes remain empty for most of their time.

We have identified that the exponential backoff is the main culprit of the unfairness. If a mesh node repeatedly fails to get access to the medium, two things happen. First, because of the access failure, the node assumes that the medium is busy and exponentially increases its congestion window. As a consequence, the node is forbidden to try the medium for longer and longer intervals. During these silent intervals, the other nodes in range find the medium idle. Thus they successfully transmit, which fills up the buffer of the waiting node, and they do not increase their contention window because the transmission was successful.

Our solution to address the above problem is 2 modifications to 802.11. First, the exponential backoff mechanism is *disabled*. Instead, we set the congestion window to a fixed value, i.e. each node randomly picks its backoff in a fixed time interval after an unsuccessful attempt to access the medium. Second, we increase the "short retry limit" that defines the maximum number of retransmissions before a packet is dropped at the MAC layer. This increase avoids that packets are dropped early and thus must be retransmitted on an end-to-end base. To provide evidence, we performed a set of simulations that are described next.

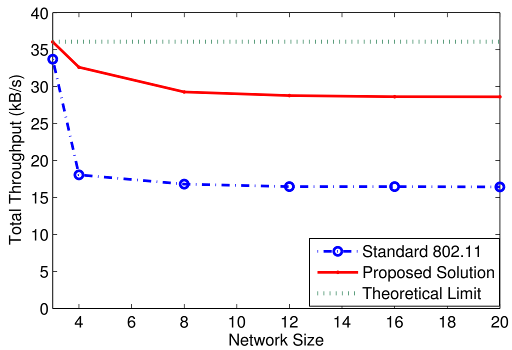

Figure 4: Backhaul throughput

2.3 Simulations

We have implemented the proposed modifications of the 802.11 protocol in ns-2. To verify our model, we define a topology of n mesh nodes in a linear node setting with n varying from 4 to 20. The nodes are separated by a distance of 200m and the transmission range is set to 250m in order to enable communication between direct neighbors. The RTS/CTS handshake mechanism is enabled at the same bitrate as the data packets to fix the sensing range at the same value as the transmission range. We set the link capacity at the minimal value of 1 Mbps and use packets of 1500 Bytes. Using these values and the spatial reuse factor from [10], we have calculated that the upper bound on the backhaul application layer throughput is 288 kbps [5].

Figure 4 shows the total end-to-end throughput of the backhaul as a function of the number of nodes on the x–axis. The theoretical limit of 288 kbps is independent of the network size for $n > 2$. The throughput with standard 802.11 quickly drops below 50% of the theoretical limit for $n > 3$ nodes. With our proposed solution, the throughput remains as high as 91% for $n = 4$ nodes and 79% for $n = 20$ nodes.

3. TESTBED VALIDATION

This section describes our efforts to reproduce the simulation results in the Magnets indoor testbed. First, we describe the indoor testbed setup. Then, we present our experiences with the validation and the lessons learned from it.

3.1 Testbed setup

The indoor testbed of the Magnets project [12] consists of 15 nodes that are deployed over 3 floors of the T-Labs building. Figure 5 shows that 5 mesh nodes are deployed on opposite sides of the T-Labs building and one node at the center. From the wired gateway (WAP), the data is forwarded over pre-configured routes along the linear topology over a configurable number of hops. Using Laptops, we measured that the WAP and TAP_4 are out of range of each other because the elevator shafts are impenetrable for the mesh nodes' signal. Similarly, we measured that the distance is approximately 50m and glass between TAP_1 and TAP_3 prevent them from seeing each other. Therefore, we use TAP_2 to connect as a relay node between the others. Thus, the interference model matches the simulation setup. In addition to the mesh nodes, 2 Linux-based PCs are connected via wired links to the WAP and the last TAP respectively to run as traffic generator and traffic sink.

Beyond the topology on a single flow is to set up a multihop topology over the different floors. The main difference between experiments on one floor and inter-floor experiments is that the

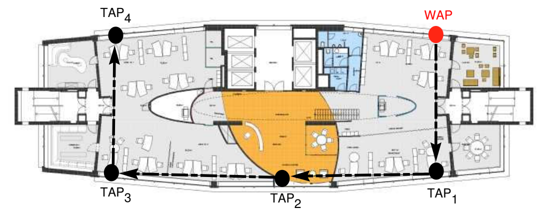

Figure 5: Illustration of our initial testbed topology.

environmental changes are more pronounced on a floor: people are moving around, windows and doors are opened and closed, elevators pass. In contrast, the connectivity among the floors is likely to remain stable as the nodes are placed along the walls. Moreover, we had seen that the inter-floor material allowed transmissions over neighboring floors whereas communication over 2 floors was not possible. We confirmed these communication and sensing range properties of our multi-floors topology by checking the independence between link l_1 ($TAP_1 \rightarrow WAP$) and l_2 ($TAP_3 \rightarrow TAP_4$). Independence was confirmed as the individual throughput of link l_1 and l_2 was similar when the links were transmitting individually or simultaneously.

The specific equipment present in our first deployment consist of 5 Routerboards 532 equipped with 1 single Atheros-based WiFi card running on 802.11a/b/g. The cards are connected to a 3dB indoor omni-directional antenna to amplify the signal. We operated the cards in 802.11a mode only because at least 3 (sometimes up to 10) interfering networks could be detected in the office in the 2.4 GHz range. Since these networks are beyond the control of the experiment, their influence would have been disturbing our results. In the 5GHz range, in contrast, no interfering network was detected.

The software running on the boards is the Kamikaze version of OpenWRT 2.6.21.5 with the MadWifi driver 0.9.3.1 release. Static routing is used at the network layer due to the relatively fixed nature of WMNs to avoid that routing changes interfere with the experiments. UDP traffic is injected at a rate of 30 Mbps using iperf [19] with a fixed packet size of 1500 bytes. A single experiment runs for at least 600 seconds to ensure that stationary state results are obtained.

Thus, in summary, the setup has been verified to correspond to the simulation topology and the hard- and software are sufficiently fast and customizable to verify the simulation results.

3.2 Challenges

To verify the simulation results with the testbed, the following questions have to be addressed. First, the driver must be modified to change the congestion window. Second, the measurements must be performed. Finally, the experimental results must be compared to the simulation results.

The modification of the congestion window is a challenging engineering problem because the documentation of the MadWifi is not always accurate. In particular, the MadWifi Manual [18] states that the value can be changed via the *iwpriv*. The command takes as parameter one of the four class of service defined in 802.11e: Best Effort (BE), Background (BK), Video (VI) or Voice (VO) to modify the value. Each of these classes of service is implemented in the mesh node as different queues with dedicated CW_{min} and CW_{max} parameters that can be set independently. We therefore concluded that the congestion window can be modified with a simple *SET* command.

The second challenge is to monitor the congestion window during the experiment in each node. Our approach was to log the congestion window inside the kernel and write it to flash disk where it can be read after the experiment.

The third problem is eventually to evaluate the results. Initially, we believed that it is sufficient to perform end-to-end measurements with tools such as iperf. Even though we were aware of the fact that a real testbed is likely not to produce perfectly steady results, we believed that we could at least see significant differences in the throughput results, as the differences between standard 802.11 and the proposed solution in the simulations were almost a factor of 2.

After these modifications, we performed several sets of measurements. Against our predictions, we were not able to verify the simulations for several reasons.

3.3 Experience

The first problem lied in the modification of the congestion window. After setting a congestion window with *SET*, the *GET* calls showed that the value was not changed. What is wrong: was the value not changed, was the value changed but the *GET* command does not work? To solve this limitation, we modified the MadWifi source code by removing in the file "*ieee80211_proto.c*" the part of the code locking the contention window parameters to their standard values for BE traffic (lines 864-867). Nevertheless, doubts remain whether the value is really changed or not.

For the second challenge, we found that logging is not a viable solution. The effects of the congestion window are most prominent when the throughput is high. However, a high throughput also implies a high load on the mesh node. With the slow CPU and the limited memory on the node, the logging is likely to interfere with the experiment. Therefore, it is impossible to clearly separate the effects of the congestion window from the logging impact.

For the third challenge, the effect on the end-to-end throughput, we found that the differences are not sufficient to verify the model. While the average throughput showed improvements by a factor of 1.5 with the proposed solution over standard 802.11, we found that the deviations were in the order of 50% of the average throughput. Are the differences now caused by our proposed solution or by environmental changes, such as people moving? Moreover, we found significant differences when repeating the experiments at different days, with average throughput variations by more than a factor of 3.

To assess the influence of the environment, we performed single hop measurements between the different nodes. We found that e.g. a simple displacement of a mesh node by as little as a fraction of a cm leads to significant changes in the connectivity, from high SNRs to basically no connection at all. Similar experiences have been reported by [17]. Moreover, people passing by, windows and doors being opened led to drastic changes in the performance. The sensitivity of our experiments was large in part because we were operating in the 5GHz range to avoid the interference of other networks. Even worse, we found contradicting results whether the TAPs are really within range or not: sometimes neighboring TAPs could not detect each other, sometimes two-hop neighbors could detect each other.

Thus, to perform the validation, it would have been necessary to pin down how much their interference level *during* the measurement. Unfortunately, as stated above, the low CPU speed and the limited memory do not allow for online measurements and for collecting environmental data. Moreover, we realized that we would have been lacking tools that allow us to easily correlate measurements of lower layer parameters with e.g. the end-to-end throughput.

Thus, using this setup, we were able to roughly confirm the results at a macroscopic level, but not to verify them to a sufficient degree. That is, we saw differences in the end-to-end throughput as a function of the congestion window. The results looked initially consistent, however, over time, the above drawbacks and variations led to throughput variations. The results could be considered and interpreted as "consistent" only with a large degree of goodwill. Thus, we were far from able to verify the results at a microscopic level, e.g. by verifying that the buffer built-up was indeed responsible for the throughput differences.

4. SMALL-SCALE TESTBED FOR MICROSCOPIC BEHAVIOR ANALYSIS

Due to the insufficient monitoring capabilities in the above setup, we decided to build a small-scale testbed with monitoring capabilities for the microscopic behavior as follows. 5 mesh nodes are placed within 1 m^2, as illustrated in Figure 6. The nodes are arranged a metallic box that blocks the radio waves to achieve the desired linear topology. The external antennas were removed from the nodes to decrease the sensing and communication range. Moreover, an additional 6^{th} node is deployed at a high elevation where all communication can be monitored. This node runs in monitoring mode and therefore acts as a centralized monitor of the entire wireless medium of the testbed. If desired, the monitoring node can also be positioned close to a given TAP to analyze the collision analysis of a specific node in detail.

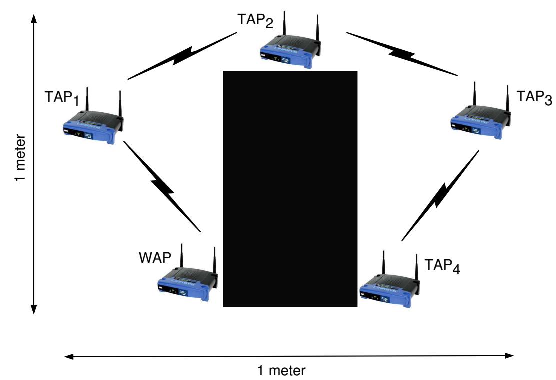

Figure 6: Illustration of our final mini-testbed.

All nodes run the firmware and tools from the OpenWRT-based Freifunk firmware [3]. This firmware contains several monitoring software packages, including the RRD-toolset called "freifunk-statistics-en 1.5.32" to log various parameters and represent them in a graphical-backend. Besides the CPU-aware self-monitoring based on the RRD-toolset, Kismet-2007-10-R1 is installed as a packet sniffer on the monitoring node. Kismet logs all sniffed packets passively and saves them in a tcpdump/wireshark compatible file format.

Unfortunately, the Freifunk release 1.5.32 currently does not support the RouterBoards hardware. Therefore, we replaced the routers by 6 Broadcom-based Linksys WRT54GL v1.0 routers. Compared to the RouterBoards, the Linksys have an even slower CPU, are limited to the 2.4 GHz modes b/g and the closed source wireless drivers from Broadcom. To reduce interference, we run the nodes in g-mode on channel 14. Moreover, because of the CPU limitation, the monitoring node only runs the Kismet drone and is connected via a wired connection to a Kismet server that processes the packets.

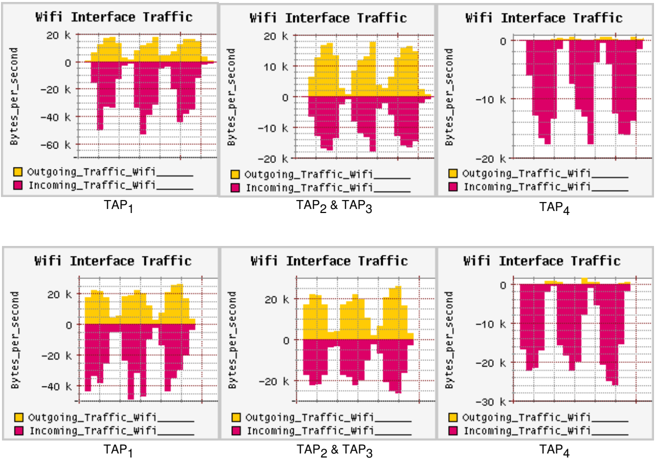

Figure 7: I/O statistics on the different TAPs.

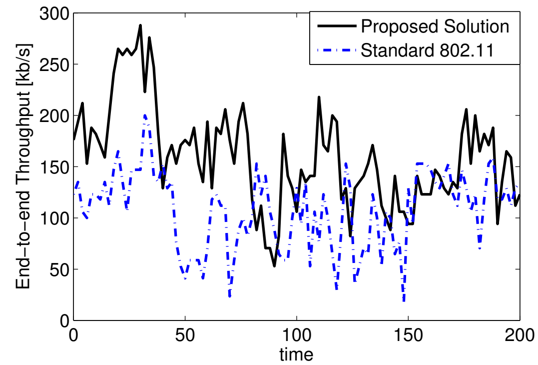

Figure 8: Experimental end-to-end UDP throughput

4.1 Experience

With this setup, we are able to perform per-packet analysis and thus validate the simulation results. Figure 7 shows the incoming and outgoing traffic on the WiFi interfaces as seen by the self-monitoring tool. The statistics of TAP_2 and TAP_3 are similar so that we show only the statistics of TAP_2. All TAPs should ideally have a balanced input/output rate. We note, though, that TAP_1 is able to forward only roughly half of the incoming packets. Thus, the traffic repartition is uneven between incoming and outgoing traffic and TAP_1 is therefore the bottleneck of the system. An analysis of the tcpdump traces on the monitoring node confirm that TAP_1 has a large number of unsuccessful transmissions that lead to an exponential backoff increase, as predicted by our analytical study [5]. This exponential increase of TAP_1's contention window therefore brings an unfair competition between the $WAP - TAP_1$ and $TAP_1 - TAP_2$ links, where the last link is significantly prejudiced in favor of the former one. Unfortunately, this inefficient over-use of the link between WAP and TAP_1 is prejudicial to the remaining competing links and therefore lowers the achieved end-to-end throughput. In contrast, with the modified backoff, the interface statistics on TAP_1 are more balanced, leading to a higher end-to-end as shown in Figure 8 where our proposed modifications achieve an average throughput of 160 kbps on the entire experiment, whereas standard 802.11 only achieves 108 kbps. Thus, the offline analysis of the monitoring information provided us with the MAC layer data necessary to confirm that the exponential backoff policy of 802.11 is the source of inefficient packet forwarding in a multihop backhaul. Indeed, this small-scale testbed enables us to highlight the benefits of our simple modifications by displaying a reduced unfairness between incoming and outgoing traffic at TAP_1 together with an improved end-to-end throughput (TAP_4).

4.2 Implications

We experienced two main set of problems that will influence our future work. First, the replication of a simulation environment in a real testbed is far from easy. We hereby refer less to the interference and other environmental factors that influence the tests, but to the fact that we lack tools and methodologies to monitor and categorize them and assess their impact. The problem starts already with the drivers. Many drivers today are still closed source and do not reveal sufficient information and monitoring capabilities for researchers. Open source drivers, on the other hand, are frequently ill documented and provide incomplete or inconsistent functionality. Finally, also the equipment does not always behave according to standards [7]. We conclude that it is vital to even plan the verification of the basic functionality into the testbed. Without the verification of the basic functionality, the assessment of the behavior of all higher layer protocols may lead to faulty conclusions. Unfortunately, we also learned that current mesh nodes have limited capabilities to perform data forwarding and monitoring functionality at the same time. The slow CPU speed and the limited memory severely restrict the verification on the device functionality itself. But how can we address the problem of *observing without perturbing*? The second problem is the lack of an efficient monitoring and data analysis toolchain. In particular, it is necessary to combine information from different layers. Related to the previous problem is the need to ensure that conflicting information is not ignored but identified and resolved. Next, different hardware and software (drivers) may not report the same set of parameters. Therefore, if not all necessary parameters are actively reported, the missing parameters must be estimated, e.g. via observation on alternative nodes. Finally, the challenge increases when information from multiple distributed mesh nodes must be combined. The distribution of information may differ in timing as well as information quality. One of the few toolschains available is Jigsaw [9]; however the tool is limited to Atheros cards and does not readily interoperate with other tools, e.g. the Freifunk firmware. Therefore, we emphasize the need for further developments of tool chains in general, but in particular also tool chains that derive information that is not directly provided from the driver.

These lessons learned had a direct impact on our testbed. First, we decided to maintain the small testbed with the microscopic analysis capabilities also for future tests. Even though of limited dimension, we found this testbed very useful to understand the microscopic behavior. Second, we decided to take down the entire indoor mesh and add an additional board to each case. Thus, the original board in the mesh node will continue to act as a data forwarding device, whereas the second board will be used for monitoring only. This setup has the advantage that the observation on the monitoring board does not perturb the experiment itself. The drawback is that the information may be slightly less accurate than the actual information. For example, data transmissions are captured by the monitoring node only after the packet leaves the antenna, whereas the data node itself could report the packet before it leaves the driver. Such differences must be taken into account when combining the information from the different layers and nodes.

5. RELATED WORK

The MAC 802.11 protocol has been initially designed for single-hop communications and its behavior and performance boundaries have been thoroughly cover for this case in recent studies such as [14] that extends the seminal Bianchi model [6]. Unfortunately this protocol without modifications is unsuited for the multihop traffic occurring in WMNs or general ad-hoc networks.

Therefore, understanding the causes of the inefficiency of 802.11-based multihop networks was the motivation behind our fluidity model and the deployment of our experimental testbed. A large body of research has studied similar problems by focusing on the unfair inter-flow competition leading to throughput degradation and flow starvation. In particular, the negative impact of the *parking-lot scenario* occurring when flows of different length transmit concurrently has been modeled and studied through simulations in [11]. Furthermore, recent work ([15, 16]) identified the hidden node situation as the source of the unfair inter-flow competition due to rerouting instability and asymmetrical unfairness. Source-rate limiting among other mechanisms is presented as a solution to reduce this problem and simulation results support their analysis. Our approach therefore differs as we base our model on fluid physic theory and focus on the intra-flow link competition that perturbs the traffic fluidity. Furthermore, we support our analysis by a model, simulations and real testbed measurements.

Our work describes the practical challenges encountered in order to deploy an experimental testbed to achieve model validation. Taking our fluid flow model as example, we describe the stepwise methodology used to overcome the difficulties and highlight the need to deploy dedicated tools and methodologies to achieve traffic monitoring. Papers such as ([9, 8, 20]) focus on the difficulties arising in the monitoring of large networks and present tools to obtain a global view through trace merging and trace synchronization.

6. CONCLUSIONS

This paper presents the practical challenges occurring in the deployment of experimental testbed to achieve model validation. The practical validation phase through testbed measurements is of utmost importance in enhancing our understanding of real systems in general and of wireless mesh networks in particular. Using our fluid flow model as example, we present the stepwise methodology we used to overcome the practical challenges. Even though some factors may be specific to our problem and our environment, we argue that the methodology and the approach can be fruitful for related projects. These experiences are particularly important because a lot of work is still only simulation-based and lacks experimental verification in spite of the fact that we know that simulators are only able to represent a small fraction of the problems seen in reality. Therefore, it is vital that the models and simulations are verified with experiments.

Stability and repeatability of the wireless measurements are necessary for enhancing our knowledge of current protocols and testing improvements. In future work, we will extend our mini-testbed of Section 4 by capturing the effect of obstacles present in non-openspace topologies and blocking the wave propagation, and providing reliable measurement in order to support our study on intra-flow behavior.

7. REFERENCES

[1] *The Cloud*. availabe at http://www.thecloud.net/.

[2] *EarthLink*. availabe at http://www.earthlink.net/.

[3] *Freifunk*. availabe at http://freifunk.net/.

[4] *MIT Roofnet*. availabe at http://pdos.csail.mit.edu/roofnet/.

[5] A. Aziz, R. Karrer, and P. Thiran. Effect of 802.11 adaptive exponential backoffs on the fluidity of downlink flows in mesh networks. In *Proc. of WinMee*, Berlin, Germany, Mar. 2008.

[6] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar. 2000.

[7] G. Bianchi, A. D. Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello. Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards. In *Proc. of INFOCOM'07*, Anchorage, Alaska, May 2007.

[8] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benko, J. Chiang, A. Snoeren, S. Savage, and G. Voelker. Automating cross-layer diagnosis of enterprise wireless networks. In *Proc. of ACM Sigcomm*, Kyoto, Japan, Aug. 2007.

[9] Y.-C. Cheng, J. Bellardo, P. Benko, A. Snoeren, and G. V. nd S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *Proc. of ACM Sigcomm*, Pisa, Italy, Sept. 2006.

[10] M. Durvy and P. Thiran. A packing approach to compare slotted and non-slotted medium access control. In *Proc. of INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[11] V. Gambiroza, B. Sadeghi, and E. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *Proceedings of ACM MobiCom*, Philadelphia, PA, Sept. 2004.

[12] R. Karrer, I. Matyasovszki, A. Botta, and A. Pescape. Magnets - experiences from deploying a joint research-operational next-generation wireless access network testbed. In *Proceedings of IEEE Tridentcom'07*, Orlando, FL, May 2007.

[13] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: the case for taps. In *Proc. of HotNets-II*, Cambridge, MA, Nov. 2003.

[14] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs. *IEEE/ACM Transactions on Networking*, 15(3):588–601, June 2007.

[15] L. Li and P. A. S. Ward. Structural unfairness in 802.11-based wireless mesh networks. In *Proc. of CNSR '07*, May 2007.

[16] P. C. Ng and S. C. Liew. Throughput analysis of IEEE 802.11 multi-hop ad hoc networks. *IEEE/ACM Transactions on Networking*, 15(2):309–322, Apr. 2007.

[17] K. Papagiannaki, M. Yarvis, and W. S. Conner. Experimental characterization of home wireless networks and design implications. In *Proc. of INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[18] Protocols Group. *Madwifi/AtherosWireless Linux Driver Users Guide*, Nov. 2005.

[19] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. *Iperf: The TCP/UDP bandwidth measurement tool*. availabe at http://dast.nlanr.net/Projects/Iperf/.

[20] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless lan monitoring and its applications. In *Proc of WiSe '04*, Oct. 2004.