Ubiquitous TV delivery to the masses

Roger Karrer and Thomas Huehn
Deutsche Telekom Laboratories
TU Berlin
{roger,thomas}@net.t-labs.tu-berlin.de

Abstract

The vision of ubiquitous delivery of TV streams to the masses at high quality is tantalizing. However, today, TV streaming is either limited to fixed wired outlets (IPTV), or it is distributed over DVB-x but the required receivers are not built into their devices. Wireless mesh networks have the potential to bridge this gap, as they can receive the TV from either technology and forward it over multiple hops to the user devices that are today equipped with 802.11 cards by default. This paper describes the set up of a TV streaming over the Magnets mesh network deployed in the city of Berlin. Initial performance evaluations on the Magnets backbone and the Magnets indoor mesh show that a combination of careful engineering and multi-card wireless routers can provide sufficient bandwidth to support TV streaming over wireless networks.

1. Introduction

The vision of ubiquitous high-speed access to the Internet is tantalizing. With the trend to all-IP networks and services, users will not only be able to surf the web, but also have access to a plethora of voice (VoIP) and streaming services, such as TV (IPTV). In fact, forecasts for 2012 predict that Mobile TV will be at least successful if not the "`killer application"' for next-generation mobile systems.

For the distribution, various technologies are available today, including WiFi, 3G and Digital Video Broadcast, e.g. via terrestrial antennas (DVB-T). The use and the advantages of the different technologies depend on the availability of the technology, the content and the data access. For example, DVB-x is suited to broadcast the same data to a large user group, but is a one-way data communication that does not allow interactive data access. In the city of Berlin, e.g., three DVB-T antennas are sufficient to broadcast digital TV in a way that within the city area and its near surroundings, TV channels can be accessed indoors and outdoors with a small indoor antenna. Beyond this "`inner circle"', TV programs can be received with outdoor or roof-mounted antennas. As a result, 30 channels are available today at good quality for free (excluding general fees), a net save of tens of dollars compared to TV via cable.

But how does the data that arrives on wired Internet connections via IPTV or via DVB-T a large user group? IPTV is only available at the outlet of the wired connections. DVB-T, in contrast, is widely available, however only few portable devices such as laptops or home entertainment devices are equipped with DVB-T receivers. Even though indoor boxes and, most recently, USB-enabled DVB-T receivers are available, it is unlikely that all devices will be equipped with such receivers given that their costs ranges from several tens to hundreds of dollars. Therefore, in today's environments, a gap exists between the vision of pervasive TV access and its realization

WiFi mesh networks have the potential to bridge the gap between the locations where TV data is available (wired outlets or DVB-x senders) and the end users. We understand a mesh network as a set of infrastructure-based, power-wired access points (APs) that primarily relays the data over multiple hops (APs) from or to a small subset of the APs that are connected to a data source. In the case of TV, this data source is either a wired Internet connection or a DVB-T receiver. Thus, by forwarding data from the data source over multiple hops, the mesh network allows users within its coverage area to receive TV streams.

However, can WiFi mesh networks really come up to the demands of TV streams? It is well known that many deployed WiFi meshes today have serious performance issues. First, the application-layer throughput of links of the MIT Roofnet [1] or the Technology-For-All network in Houston [2] is limited to single-digit numbers, even though the WiFi cards have a raw throughput of 54 Mbps. Taking the protocol overhead of approximately 50% of the different layers into account, the measured throughput is simply just a fraction of the expected application-layer throughput. Second, wireless mesh networks show severe performance degradations for flows that traverse multiple hops. Moreover, they show severe unfairness towards flows that traverse more hops than shorter flows. In particular, studies show that flows may completely starve after 3 hops in a wireless network [3]. Therefore, current mesh networks are not suited to support the bandwidth demands required for TV streaming over mesh networks. But does this shortcoming imply that wireless mesh networks are unable to support TV streams?

This paper describes the set up of a TV streaming over the Magnets mesh network deployed in the city of Berlin [4]. Compared to the above mesh networks, Magnets has two fundamental differences. First, Magnets contains a dedicated, well-engineered high-speed wireless backbone. Initial measurements show that single backbone links are able to support high data rates. However, we have not investigated if the backbone characteristics are sufficient to support TV streams in terms of bandwidth, delay and jitter. Second, attached to the backbone is a WiFi mesh network whose APs are equipped with multiple WiFi cards. Having multiple WiFi cards allows an AP to simultaneously send and/or receive data from and to different APs. In this paper, we provide evidence that the performance degradations found in access points with a single WiFi card only are not visible when multiple cards are used in different frequency bands.

The contributions of this paper are three-fold. First, we describe the design of the *MagNets* network. In particular, we highlight the design of the backbone and the mesh network. We provide details on the network layout and the mesh nodes used for the backbone and the mesh.

Second, we describe the setup to stream TV over the backbone and the mesh network. We provide details on the hardware, the software and the requirements in the mesh and at end systems to support TV streaming over the mesh and the backbone.

Finally, we provide initial measurements on the backbone and in the mesh that show that a combination of careful engineering and multi-card wireless routers can provide sufficient bandwidth to support TV streaming over wireless networks.

This paper is organized as follows. Section 2 provides background information. Section 3 describes the architecture of the Magnets network and the setup of the TV streaming. Section 4 presents measurements from streaming TV and TV-like traffic over Magnets. After discussing related work in Section 5, we conclude in Section 6.

2. Background

This section describes the scenario and provides background on the technical details of DVB-T in general and the setup in Berlin in particular.

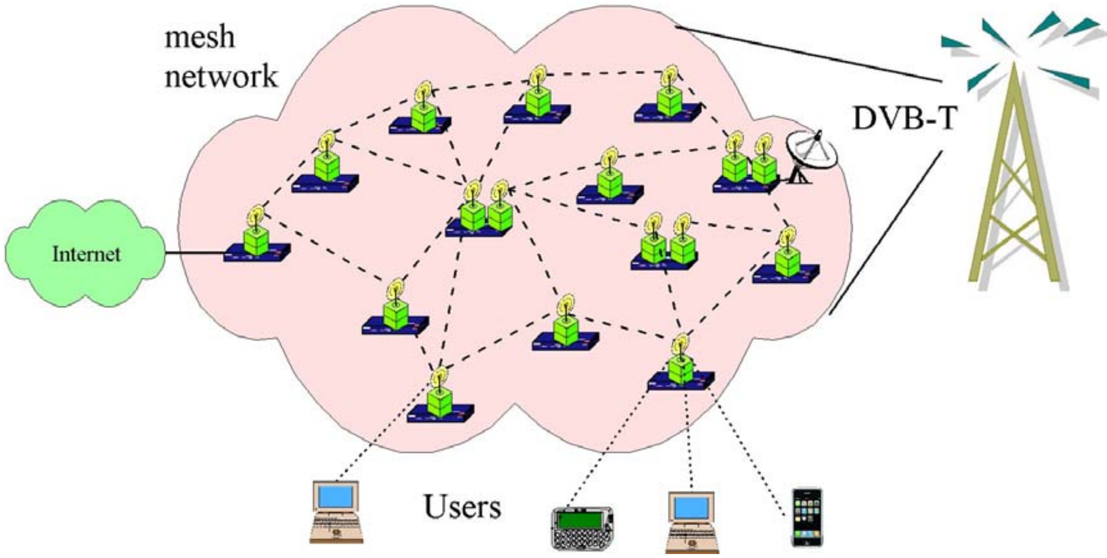

Figure 1: TV streaming over a wireless mesh network

Figure 1 depicts the scenario we are considering: users (mobile nodes, MN) with laptops or any other device that is able to display TV streams and that is equipped with a WiFi card are connected to an access point of the mesh cloud. Other APs of the same mesh are connected to a TV streaming source: either a fixed Internet connection where the TV is streamed via IPTV over the core, or a mesh router or server that contains a DVB-T receiver. The mesh forwards the data over multiple hops from this source to the different users.

DVB-T stands for Digital Video Broadcasting - Terrestrial and it is the DVB European consortium standard for the broadcast transmission of digital terrestrial television. DVB-T transmits a compressed digital audio/video stream, using OFDM modulation with concatenated channel coding (i.e. COFDM). Currently, a second-generation specification (DVB-T2) is under discussion.

DVB-T uses MPEG-2 for source coding. The compressed video, compressed audio, and data streams are multiplexed into PSs (Programme Streams). One or more PSs are joined together into an MPEG-2 TS (MPEG-2 Transport Stream). The TS is the basic digital stream which is being transmitted and received by home Set Top Boxes or by DVB-T USB devices. Most recently, H.264 is also used as an alternative to H.264.

The bitrates for the transported data depend on a number of coding and modulation parameters and can range from about 5 to about 32 Mbps. In Berlin, the DVB-T senders are set to 16-QAM 2/3, which result in bitrates between 14 and 18 Mbps [5].

Given the scenario and the requirements from the DVB-T streaming, the main challenge we are interested in is: are mesh networks able to support the required bandwidth, latency and jitter for TV streaming? Commodity hardware today achieves a raw data throughput of 5 Mbps or even 108 Mbps via proprietary modes. With the overhead of the different protocols in the Internet stack that accounts for roughly 50% of the raw throughput, the expected application-layer throughput can be estimated to roughly 27 Mbps. However, experimental mesh networks deployed, such as the MIT Roofnet [1] or the Technology-For-All in Houston [2], reach only single-digit throughputs and are far from achieving the rates required to stream TV.

3. Architecture

This section first provides information about the Magnets testbed, in particular the WiFi backbone and the WiFi mesh in isolation. Then we describe the setup to stream TV over Magnets.

3.1. Magnets backbone

The *MagNets* project aims at deploying a next-generation wireless access network architecture. Within this network, the high-speed WiFi backbone connects 5 high-rise buildings in the heart of Berlin. The backbone is composed of 6 PC based routers and 12 Access Points (AP) (10 indoor and 2 outdoor). The APs consist of Intel IXP420@266 MHz (indoor) and IXP425@533 MHz (outdoor) programmable network processors (NP) as CPU, and Atheros 5213/5112 chipset for their WLAN interfaces, and run a proprietary operating system called LC.OS. More information on the topology can be found in [4].

In previous work, we have shown that the Magnets backbone achieves high link speeds over single and multiple hops by three means [6]. First, each link can be activated individually to avoid multi-hop throughput degradations known from mesh networks [3]. To achieve independent link transmissions, each link is operated via a dedicated AP. The APs are interconnected via a Linux PC with multiple network interface cards that acts as a router. Second, directional antennas ensure a high signal level to bridge the distances but also reduce the interference with other links.

Third, the APs feature two proprietary protocols to enhance the throughput beyond the 54 Mbps supported by 802.11a/g termed *Turbo Mode* and *Burst Mode* that can be enabled optionally. *Turbo Mode* doubles the channel from 20 MHz to 40 MHz. While, using *Burst Mode*, the sender only waits for the shorter SIFS (Short Inter-Frame Space) after a successful data exchange instead of the longer Distributed Inter-Frame Space (DIFS) specified in 802.11. The modes should result in a performance enhancement of 10 Mbps for *Burst Mode* and a throughput doubling for *Turbo Mode*. These modes are expected to boost the backbone performance without negative impact due to the independent link scheduling and the use of directional antennas. For general (mesh) networks, however, *Burst Mode* can lead to severe unfairness and *Turbo Mode* interferes with all other channels in the 2.4 GHz spectrum because it must be centered around channel 6 to stay within the allotted frequency band.

3.2. Magnets mesh

The *MagNets* mesh network aims at investigating the limitations in terms of capacity and delay, but with off-the-shelf hardware only. Our main interest is to assess how a mesh network can scale to high capacity and thus to high end-to-end throughput if the mesh nodes are equipped with multiple WiFi cards. Towards that goals, we have decided to acquire 20 RouterBoard 532 [7] as an all-in-one integrated communication platform. Such a board features a MIPS32 CPU running at up to 400 MHz and a 32-bit PCI controller at 66 MHz. For networking, the board provides up to 3 Ethernet ports and 2 MiniPCI slots on board. Daughterboards can additionally be attached via on-board connectors. The RouterBoard 564, e.g., is a daughterboard that provides 6 Ethernet ports and 4 MiniPCI slots. Using Atheros 802.11a/g WiFi cards that offer 54 Mbps in their standard mode and 108 Mbps with SuperAG enhanced technology, the theoretical throughput of a routerboard reaches up to 648 Mbps.

We deployed the boards on 5 adjacent floors of our office building. Due to the concrete and steal construction of the building, the connectivity among the nodes shows an interesting behavior. On each floor, the boards are placed at the four corners of the building. While each board "`sees"' its neighbors along the wall, the connectivity along the diagonal is very low. That is, the connectivity of the 4 mesh nodes forms a rectangle. Vertically, connectivity exists between adjacent floors only, however, there is no connectivity between nodes that are separated by two or more floors.

Given this connectivity, we have several options to configure the routing and thus modify the logical topology of the mesh network. For our study, we use static routing for two reasons: to avoid effects on streaming by route changes and build the different logical topologies. In an extreme case, we can build a linear topology of 20 mesh nodes. To avoid any undesirable interference among the nodes, the channels of the different WiFi cards are separated as far apart as possible. That is, we use the entire free spectrum at 2.4 GHz and in the upper and lower band of the 5 GHz range. The detailed assignment varies with the routing and thus the logical topology.

3.3. TV streaming over Magnets

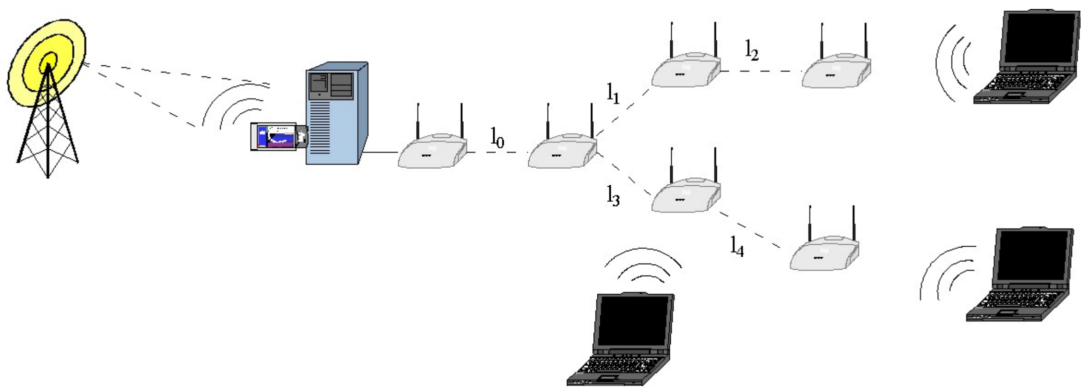

Figure 2: Indoor testbed setup for TV streaming

To set up TV streaming over Magnets, only a few components had to be acquired and configured, as sketched in Figure 2. To receive the data stream sent out from the antennas, we connected a USB DVB-T receiver to a Linux server. This server acts as the ingress point of the TV stream into Magnets. The hardware on the USB DVB-T receiver captures the TV stream and forwards it to the TV capturing software running on the server. For this purpose, we used the *Kaffeine* player that already supports the handling of DVB-T devices, and we installed the *XviD* library to interpret the data stream.

The *Kaffeine* player also supports the broadcasting of the incoming data stream to a destination. For our purpose, we connected an AP via a wire to the Linux host and configured that the data stream is directly forwarded to the AP. From there, the data stream is forwarded over multiple APs towards the WiFi-enabled laptops that act as sinks. With respect to the APs, we used both the Magnets backbone and the indoor mesh to forward the data. We always used fixed routes to forward the data because at this stage we wanted to investigate if the underlying infrastructure is already able to support the data rates required by the TV stream. A dynamic routing protocol would have mixed the link behavior with the routing behavior and would like probably have resulted in effects that are not easily understood. Therefore, we decided to stick to static routes only.

4. Evaluation

This section describes our measurement strategy and discusses our experimental results.

4.1. Measurement strategy

To assess the ability to support TV streaming over Magnets, we perform two sets of measurements over the backbone and the mesh independently. First, we use *iperf*, a well-known traffic generator tool, to generate constant stream of UDP packets at a rate of 16 Mbps with packets of 1kB size. This rate corresponds to the streaming rate of DVB-T in Berlin. Second, we attach the DVB-T stick to the server and forward the real TV stream over the mesh.

This two-staged strategy is motivated by three key facts. The first is that the DVB-T stream does not generate packets at a precise regular interval. Inside the server, the data transfer from the USB card to the Kaffeine player and from there to the AP can interfere as the transfers cross the same bus. In contrast, *iperf* is the only application running on the server, the mesh and the client and is therefore more precise, in particular to measure jitter and latencies.

Second, we set up *iperf* to have its packets carry a sequence number to detect packet loss. With DVB-T, the identification of packet loss is not straightforward.

On the other hand, DVB-T has the advantages that we can assess the video quality. We modified the client viewer to calculate the number of correctly received frames in time for displaying. That is, packets that are not received in time are not displayed and are therefore not included in these measurements. Moreover, since MPEG gives different priorities to its frames within a Group of Pictures (GOP), the loss of a single packet may have an influence of multiple other packets or frames. Finally, DVB-T measurements take into account that frames may not be displayed if they arrive correctly but too late at the client. Again, this metric is not considered with the *iperf* measurements.

Therefore, DVB-T allows us to measure application-layer metrics whereas *iperf* relates to network-layer metrics.

When streaming data over the backbone, we consider a two-hop topology from the node at T-Labs via TC to HHI (see Figure 1 in [4]). We chose this topology because links from and to ETF are not reliable enough and we excluded the direct links from T-Labs to HHI to avoid mutual interference among the nodes at HHI that are all in the 2.4 GHz band.

For the mesh, we set up the routing on the mesh nodes to form a linear topology. The experiments we show in this paper compare the performance seen by a client that is between 1 and 4 hops away from the IPTV server.

All experiments were conducted over 2 minutes time and repeated 20 times.

4.2. Backbone measurements

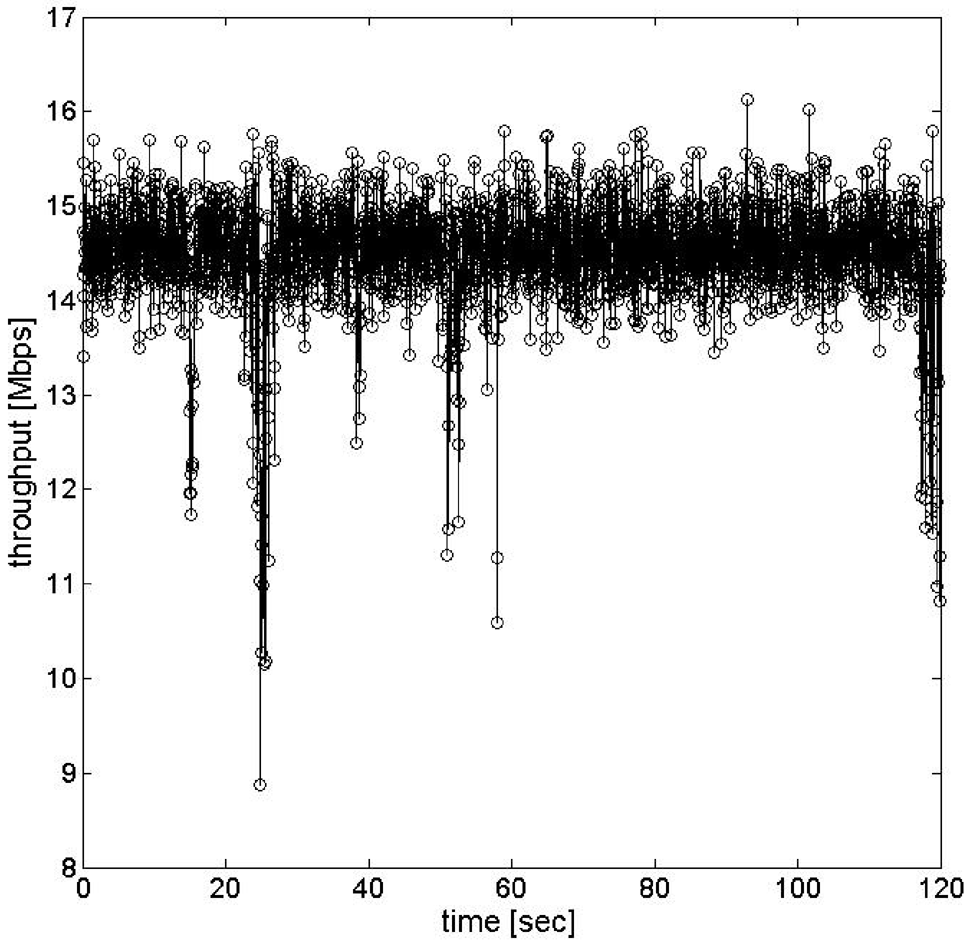

Figure 3: Backbone throughput

Figure 3 depicts the throughput at the receiving node when traffic is injected at 16 Mbps using *iperf*. The x-axis denotes the time of the experiment; the y-axis denotes the throughput in Mbps. The figure shows a randomly selected trace. We anticipate that all traces have similar characteristics in terms of average throughput and standard deviation, so that only the details of when which dip occurs changes among the traces.

The figure shows that the Magnets backbone is able to support the traffic injected via *iperf*. In fact, the mean throughput is slightly above 15 Mbps and therefore only minimally below the injected traffic rate. The standard deviation lies at 1.3 Mbps, which emphasizes the stability of the links. The small dips in the performance, however, show that the stability is treacherous, while in fact the medium is still air and therefore sudden drops in the performance may occur for whatever reason.

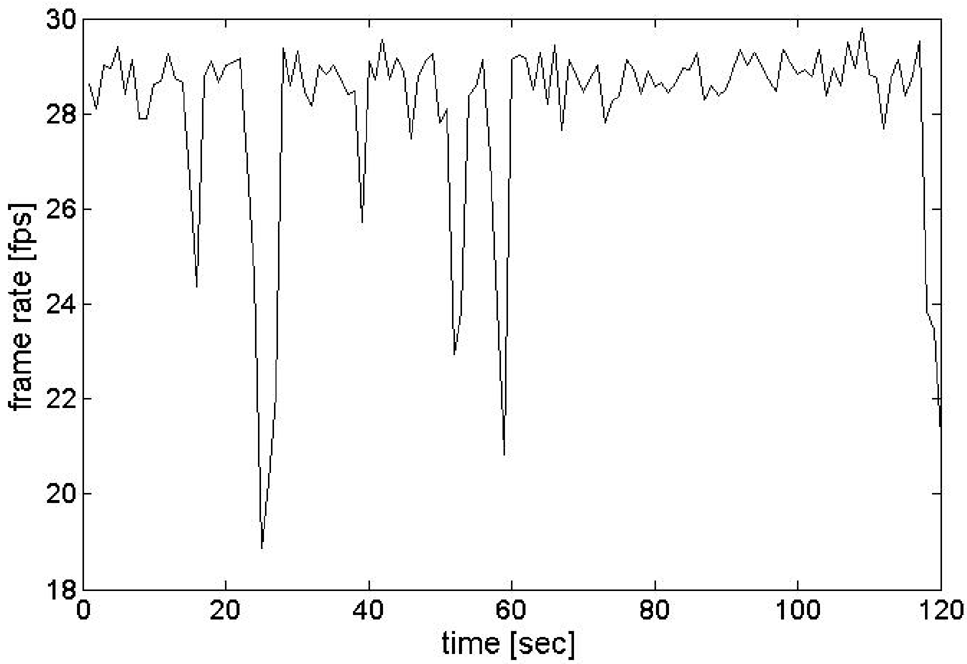

Figure 4: Backbone framerate

Next, with the same setup, but using the DVB-T receiver instead of *iperf*, we measure the frame rate that the video player sees at the destination. The video player logs the number of frames it can display per second. This metric is shown on the y-axis of Figure 4 as a function of time on the x-axis for a random experiment again. Note that the number of displayable frames *excludes* those frames that arrived at the receiver but can not be displayed because a frame with higher priority was not received correctly. For example, if a P-frame is missing in the stream, B-frames that depend on the P-frame can not be displayed either even though they may have been received correctly. These frames are not accounted for because the player can not display them.

The figure shows that the backbone is able to maintain an almost reliable frame rate. In fact, the average frame rate is 28 frames per second, out of 30 transmitted, with a standard deviation of 2 frames per second. These rates clearly lead to an acceptable if not excellent viewing experience by a human user.

Thus we conclude that the Magnets backbone is able to provide the necessary support to stream TV from a source over multiple hops to a destination.

4.2. Mesh measurements

This section repeats the above measurements, except that the data is now streamed over the indoor mesh rather than over the outdoor backbone. There are two significant differences in the measurement setup that we expect will influence the results. First, the mesh is indoor. Therefore, we expect that the performance of the links is more stable than in the outdoor environment. Second, the mesh network uses omni-directional antennas whereas the backbone used directional antennas. Due to the differences in the environment, the antennas do not have a significant impact on the throughput, but with omni-directional antennas we are able to compare the streaming performance when the APs have one or two WiFi cards enabled.

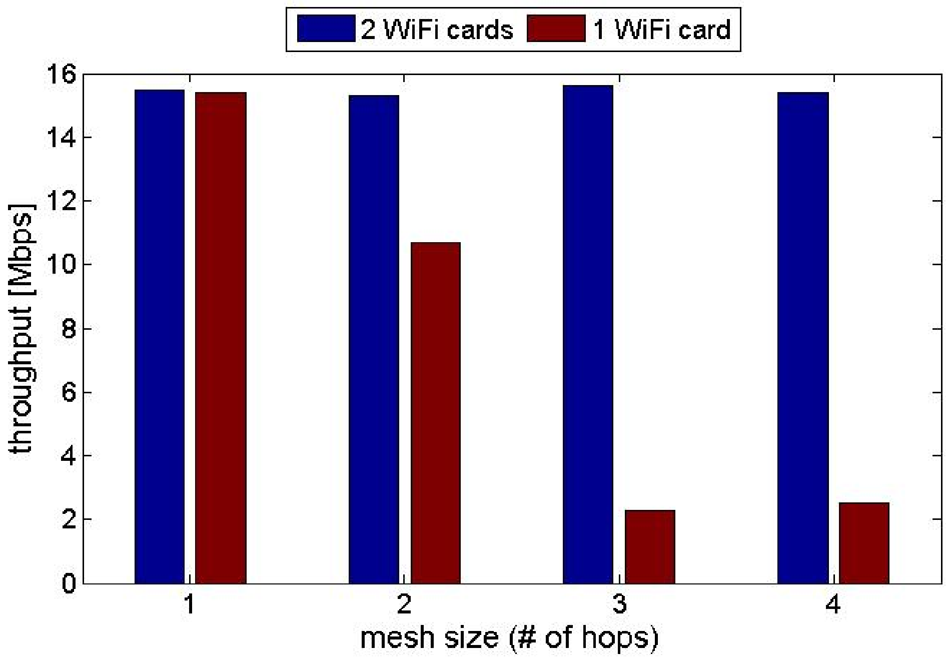

Figure 5: Mesh throughput

First, we repeat the above experiment using *iperf* over the mesh, with an traffic load of 16 Mbps. Figure 5 shows the average throughput as a function of the size of the mesh, which corresponds to the number of hops due to the linear topology on the x-axis. The two bar groups show when the APs have 1 or 2 WiFi cards enabled. The throughput is first averaged per experiment and then averaged over the 20 repetitions.

When the APs have 2 WiFi cards enabled, the average throughput lies above 15 Mbps, independent of the size of the mesh. It can be expected that the throughput would remain at a similar level even if the mesh network contained more nodes. In contrast, with only 1 WiFi card per AP, the throughput degrades significantly with the size of the mesh. For a 2 -hop topology, the performance is already reduced to 10 Mbps, and for a 3 - and 4 -hop topology, it is as low as 2 Mbps. Two factors contribute to the performance degradation. First, each relaying node can only send or receive at a given point in time. Therefore, the "`raw capacity"' of a node is actually halved, which has a visible impact on the application-layer throughput. Second, all WiFi cards must be set to the same frequency to ensure that two adjacent nodes see each other. As a result, some interference is caused also to those nodes that are not a direct neighbor. Fortunately, due to the building, the mutual interference is limited and thus this parameter is not affecting the throughput as much as the former.

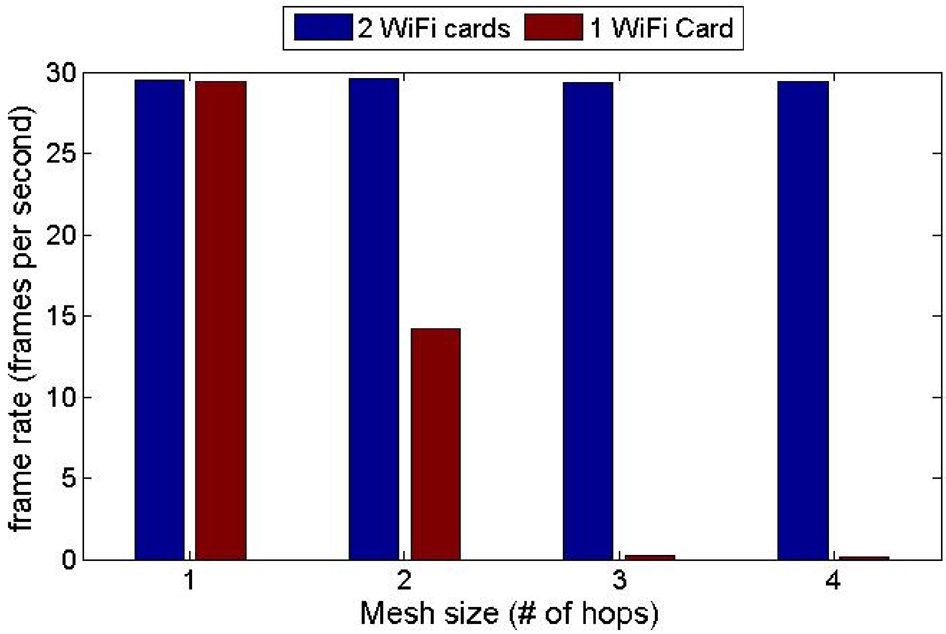

Figure 6: Mesh throughput

Finally, Figure 6 depicts the displayed frame rate at the receiver as a function of the mesh size and the number of WiFi cards. The figure shows a similar trend as in the previous experiment: the frame rate is almost optimal and independent of the mesh size, whereas it degrades when only 1 WiFi card is used. In contrast to Figure 5, however, the frame rate almost reaches zero for a topology of 3 nodes or more. The combination of low throughput and high jitter (not shown in the figures) leads to the situation that the video player hardly receives a correct set of frames in the time needed.

Thus, based on the above results, we conclude that the Magnets backbone and the mesh are able to bridge the gap between TV stream availability and its distribution to a broad user community. In particular, the design of the network allows Magnets to obtain sufficient bandwidth to stream TV to the users.

5. Interactions with related protocols

The previous considerations and evaluations have considered DVB-T as the only protocol and application running on the mesh. However, in a real mesh, a variety of different protocols with different, potentially competing requirements will have to be supported concurrently: users that surf the Web via HTTP/TCP, file downloads (TCP), voice applications (VoIP), etc. equipped with DVB-T receivers to actually profit from the broadcast medium. For a mesh, it is vital to efficiently support the different requirements to maximize the user-perceived quality of experience.

In consequence, two challenges arise for the traffic engineering within the mesh network. First, the relative priorities of the concurrent flows must be reflected in the traffic forwarding on each node. That is, delay-sensitive traffic must given strict or relative priority over bulk data transfer and video. Second, depending on the network topology and the flows, flows must be managed at the ingress to the mesh. While the first challenge can be addressed by selecting appropriate traffic classes for 802.11e, the second question is currently not resolved.

Link	l_0	l_1	l_2	l_3	l_4
Throughput [Mbps]	18	18	18	6	6
Bandwidth (clients 1,2,3)	16,16,2	16	0,16,2	6	6
Desirable allocation	6,6,12	6	0,6,12	6	6

Table 1: sample link throughput and client bandwidth (cfg Figure 2)

To illustrate the problem, consider the topology in Figure 2, where 3 clients are attached to the mesh. Assume that the clients on the right share a TV session whereas the client at the bottom surfs the web or has a voice conversation running. Now assume that links l_0 to l_2 have a throughput of 18 Mbps, whereas l_3 and l_4 (to the TV clients) only have a throughput of 6 Mbps, e.g. due to high interference, as indicated in Table 1. Without traffic engineering, the bandwidth usage of the three clients on link l_0 is therefore 16, 16 and 2 Mbps. However, out of the 16 Mbps, only 6 Mbps eventually reach the clients. Therefore, due to the *downstream bottleneck* effect, the upstream bandwidth is not efficiently allocated. The desirable allocation that would ensure the most efficient resource usage would be to limit the video stream already at the ingress to 6 Mbps, leaving 12 Mbps to client 3. With this allocation, the quality of experience of clients 1 and 2 is maintained while the bandwidth to client 3 is significantly increased. While the detailed numbers of this example are randomly chosen, we emphasize that downstream bottlenecks can occur in any real deployed mesh at any time for a plethora of reasons, including interference and user distribution.

Our solution to address this problem is as follows. Each mesh node dynamically monitors the usage of the outgoing links and periodically reports the bandwidth to the upstream node. The ingress node then compares the link bandwidth to the aggregated bandwidth of all flows traversing that link. If the aggregated bandwidth at the ingress exceeds the downstream bandwidth, the ingress node throttles the flows to the bottleneck bandwidth. In the above example, the mesh nodes before l_3 and l_4 would report the bandwidth of 6 Mbps to the upstream nodes, and the ingress node would limit the video to 6 Mbps. Then, client 3 can increase its bandwidth to the 12 Mbps.

We implemented this mechanism in our mesh. In particular, each mesh node monitors and reports the bandwidth every 5 seconds to the upstream node. The ingress node manages a set of queues, one for each mesh node. Upon receiving bandwidth information from a mesh node, it limits the throughput to this node using the *tc* framework in the Linux kernel.

To evaluate our solution, we set up the topology depicted in Figure 2 and limited the throughput of links l_3 and l_4 to 6 Mbps by choosing the according modulation. Similarly, we set the modulation of link l_0 to 18 Mbps. We then inject a TV stream at 16 Mbps and generate additional traffic for client 3 using iperf.

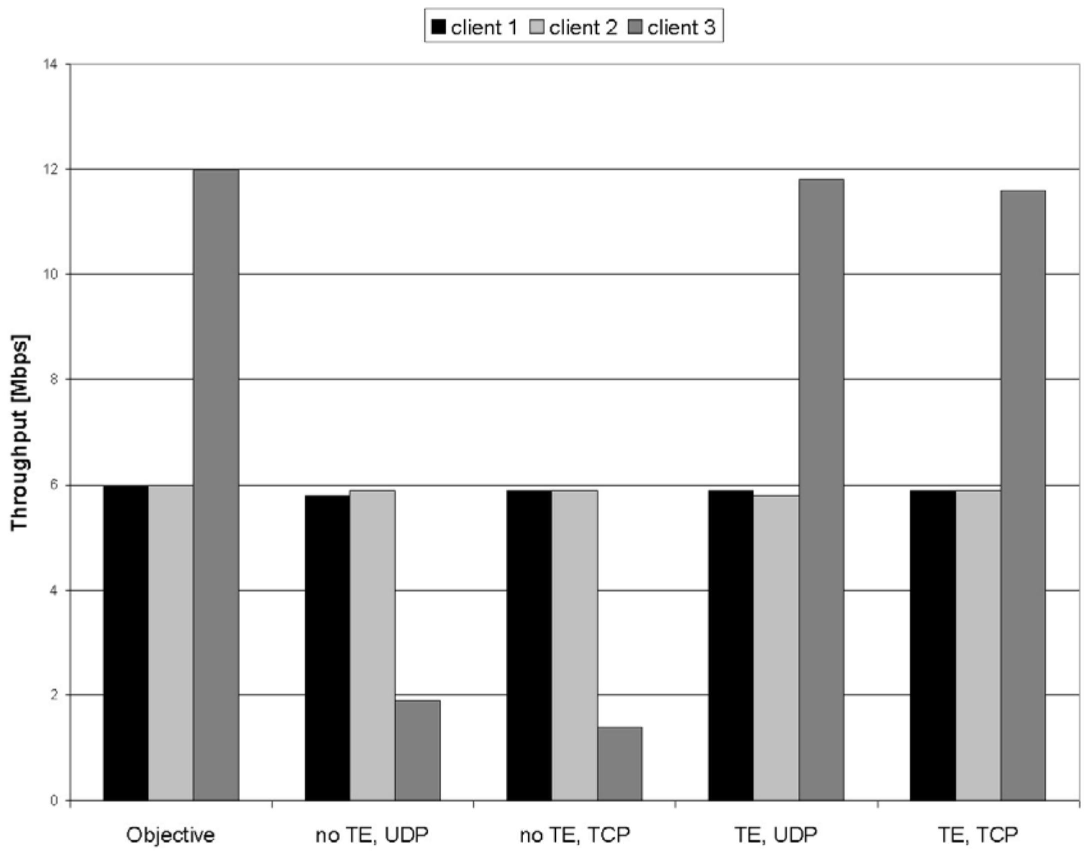

Figure 7: effects of traffic engineering on client throughput

Figure 7 shows the impact of the traffic engineering on the throughput the client experiences. The first set of bars shows the target objective, where client 3 is able to achieve 12 Mbps. The next two sets of bar show the measured throughput at the clients without traffic engineering, as a function of the protocol used for client 3 (UDP or TCP). We note that clients 1 and 2 achieve approximately the 6Mbps, but the throughput of client 3 is a dismal 1.9 Mbps for UDP and 1.4 for TCP. Thus, we see that the achieved throughput for TCP is even far below the expected 2 Mbps for client 3. Finally, the last two sets of bars show the effects of the traffic engineering, where client 3 achieves 11.8 and 11.6 Mbps respectively.

The presented solution provides a simple solution to a complex traffic engineering problem. While the results show that the overall goal is achieved, a number of questions are left for future work. A first question is how to design the ingress filtering in detail. In the above setup, each client receives exactly one traffic class. Therefore, the number of queues at the ingress node scales linearly with the number of mesh nodes. However, if the ingress node should also schedule the traffic based on the traffic class, the number of queues to be managed increases quickly. It is not clear yet whether the distinction is necessary or not at the ingress node. To answer this question, a large-scale study with real users would be needed to assess the impact of the per-class queuing onto the user-perceived quality. Second, we deliberately set the update interval to 5 seconds. Is this interval suited for updates? Obviously, selecting a good update interval depends on a number of factors, such as the variation of the channel over time, the update delay and the tolerable overhead. Therefore, we argue that a detailed study is again needed to isolate the parameters that influence the overall traffic engineering performance and then assess their impact on the performance. Given the potential benefit of addressing the traffic engineering, we will address these questions in future work.

6. Related Work

The vision of using DVB-T as a broadcast medium is shared by other projects, e.g. Daidalos [8]. And, of course, the deployment of DVB-T stations that already broadcast digital TV channels in several cities in Europe. However, it is still unclear how many devices will be equipped with DVB-T receivers to actually profit from the broadcast medium.

Mesh networks are increasingly deployed world wide. The most prominent mesh networks are probably the MIT roofnet [1] and the TfA network in Houston, Texas [2]. However, many cities today deploy mesh networks, including San Francisco, London, Taipeh, etc. Technically, the key distinguishing factor of Magnets is the combination of a high-speed wireless backbone with a wireless mesh network with multiple WiFi cards per mesh node. A key problem that many commercial mesh networks face is that the business cases are still at odds and that applications are missing to attract subscribers.

Therefore, we argue that the combination and integration of DVB-T with a high-speed wireless mesh network that even supports TV streaming is novel. The results highlight the feasibility of streaming TV over a mesh and provide therefore one example of an application that may make mesh networks attractive to users.

7. Conclusions

This paper presents an architecture to bring digital TV to a large user population. The architecture consists of a wireless mesh network that forwards data from a single source that is connected to a TV streaming source to users that are connected to the mesh access points. The streaming source can be a simple wired Internet connection that receives IPTV data, or a DVB-T receiver attached to a wireless mesh node via USB that transforms DVB-T signals into IP packets to forward over the mesh.

An experimental performance evaluation on the Magnets backbone and the indoor mesh shows that Magnets is able to support the data rates required for IPTV. The use of multiple WiFi cards per AP is a key requirement to achieve a sustainable throughput in a mesh network. The comparison of results with 1 or 2 WiFi cards emphasize the need to build mesh networks with multiple WiFi cards. On the other hand, as promising as these results may look, a number of factors may also speak against such a deployment. For example, the free spectrum available today is scarce. What happens if multiple mesh networks, each with multiple WiFi cards, start to interfere? How much free spectrum will we eventually need? Shedding light on these and other questions is a key goal of the Magnets project.

In future work, we will perform more measurements over more complex topologies and even over an outdoor mesh. Moreover, we will look at the control functions within the different layers to understand the limiting factors of the mesh network. Because the presented measurements show the potential of mesh networks to realize the vision of pervasive communication.

References

[1] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of the MIT Roofnet mesh network. In Proc. Of Mobicom, Augt 2005.

[2] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement driven deployment of a two-tier urban mesh access network. In Proc. of ACM Mobisys, Jun 2006

[3] V. Gambiroza, B. Sadeghi, and E. Knightly. End-to-end performance and fairness in Multihop wireless backhaul networks. In Proc. of ACM Mobicom, Sep 2004.

[4] R. Karrer, I. Matyasovszki, A. Botta, and A. Pescape. MagNets: experiences from deploying a joint research-operational next-generation wireless access network testbed. In Proc. TridentCom, May 2007.

[5] http://pda.etsi.org/pda/queryform.asp.

[6] R. Karrer, I. Matyasovszki, A. Botta, and A. Pescape. Experimental Evaluation and Characterization of the Magnets Wireless Backbone. In Proc. ACM WinTech, Sept 2006.

[7] http://www.routerboards.com

[8] R. Mullions, F. Mahon, C. Kuhmuench, M. Crotty, J. Mitic, and T. Pfeifer. Daidalos: a platform for facilitating pervasive services. *Advances in Pervasive Computing*, May 2006.

Authors

Dr. Roger Karrer received his Diploma and PhD in Computer Science from ETH Zurich, Switzerland. He is currently a Senior Research Scientist at the Deutsche Telekom Laboratories in Berlin, Germany. His interests are in Internet architectures and protocols in general. A particular focus is the deployment and evaluation of wireless mesh networks as well as their economical and social impact.

Thomas Huehn received his Diploma in Industrial Engineering and Business Administration from TU-Ilmenau, Germany. He is currently a Research Scientist at the Technical University of Berlin, Intelligent Networks Research Group from Anja Feldmann, Germany. His interests are in distributed architectures and trust based approaches in general. A particular focus is the deployment and evaluation of wireless mesh networks as well as their economical and social impact.