

A Testbed Analysis of the Effects of IEEE 802.11s Power Save on Mesh Link Performance

Marco Porsch, Thomas Bauschert

Chemnitz University of Technology,
Reichenhainer Str. 70, 09126 Chemnitz, Germany
{marco.porsch, thomas.bauschert}@etit.tu-chemnitz.de

Abstract. Recently, the IEEE published the 802.11s standard amendment for wireless mesh networks using IEEE 802.11 hardware. Since most of the new techniques may be implemented in software, this allows to bring the benefits of mesh networking to a broad market of about one billion devices. The standard amendment describes new power save schemes which allow efficient energy saving for all mesh nodes. On the downside, the power save schemes will also degrade the network performance in terms of delay and throughput. In this paper we present a testbed analysis of the effects of 802.11s power save on mesh link performance. We analyze the effects on round-trip time and throughput in detail for different configurations and network parameters.

Keywords: wireless mesh networks, IEEE 802.11s, power save, green networking, energy efficiency

1 Introduction

Wireless mesh networks are a class of networks which perform backbone packet forwarding on air. Thus, they allow high-performance network coverage in areas where it would be difficult to install cables. Additionally, they employ techniques for self-configuration, organization and healing. In contrast to traditional ad-hoc networks, wireless mesh networks also integrate various other types of networks by employing mesh gateways.

IEEE 802.11s [1] is a mesh networking amendment to the IEEE 802.11 standard for wireless LAN [2]. It has been finished in late 2011 and will be included into the new comprehensive standard IEEE 802.11-2012 [3]. As a member of the 802.11 family, 802.11s relies on a basis of about one billion devices used in home networking, business and public places [4]. Many of these devices require only software modifications to be able to join in to mesh networks. A software protocol stack for IEEE 802.11s mesh networking has been included into the GNU/Linux kernel since July 2008 [5] and may be used on Linux devices with a supported wireless LAN chipset.

Due to the large amount and the often mobile nature of these devices, power save is a significant issue. For the traditional infrastructure and ad-hoc wireless LANs power save modes have been standardized and established. Unfortunately, the most

commonly used infrastructure mode allows power saving only for clients – not for access points. The authors of [6] highlight the wasted energy due to idle – but always-on – access points for which no power save scheme has been designed. IEEE 802.11s defines new power save schemes which are available for all mesh nodes and which are specifically designed for the wireless mesh environments.

This paper deals with an analysis of the 802.11s power save schemes. For that we implemented these schemes in a real mesh network testbed and examined their effects on mesh link performance. A detailed analysis for different configuration and network parameters is provided. Section 2 covers the related work on this topic. The 802.11s power save schemes are explained in section 3. In section 4 an outline of the power save testbed implementation is given and effects of 802.11s power save on network performance are illustrated and explained. In section 5 we summarize our current achievements and give an outlook on future work.

2 Related work

A wide overview on the general issues and approaches of power saving in communication systems is given in [7]. The authors of [8] give a good introduction on the different 802.11 infrastructure power save schemes before they discuss the challenges of providing wireless mesh access point coverage with solar-powered mesh nodes. An in-detail description of the 802.11 power save schemes is given in the respective standards IEEE 802.11-2007 [2] and the IEEE 802.11s amendment [1].

Jardosh et al. [6] raise awareness for the problem of missing access point power save and propose a resource-on-demand management strategy, which monitors the network and powers off idle nodes based on location and volume of user demand. Similarly, [9] focuses on wireless backhaul mesh networks and provides an optimization framework that advises which nodes to shut down under dynamic load conditions without impacting the quality of service. In [10] the power management problem in mesh networks is formulated as a game theoretic problem with respect to throughput. However, in [10] energy saving is achieved by controlling the transmission power, which is not widely used in 802.11 networks and does not reduce the power consumption of idle nodes. The author of [11] addresses the joint problem of routing and channel assignment for multi-channel, multi-radio nodes with the objective of saving energy by suspending the maximum number of radios.

The authors of [14] analyze the differences in synchronized and unsynchronized beaconing for 802.11s mesh networks and come to the conclusion that synchronized beaconing, as used in 802.11 ad-hoc networks, would have been a better choice for mesh networks. Similar to our work, the authors of [12] take a look at the details of the power save schemes defined by IEEE 802.11s. They examine the performance using an analytical model and packet-based simulations with ns-2.

Many of the mentioned schemes are not or only partially interoperable with IEEE 802.11s. Adhering to the strictly defined power save schemes of IEEE 802.11s is beneficial, as it allows interoperability of mesh networks with heterogeneous hardware. But still, the schemes of IEEE 802.11s may be improved without losing the

interoperability by using the well-defined extensible protocol architecture, e.g. for synchronization. But first, the characteristics and weaknesses of the 802.11s standard power save scheme should be analyzed before designing suitable improvements. To our best knowledge no testbed study of the effects of 802.11s power save on network performance has been published yet.

3 IEEE 802.11s power save

In all of the 802.11 family power save schemes, energy saving is achieved by suspending the radio. In this low power doze state the radio power consumption may be reduced by more than two orders of magnitude compared to the awake state [8]. As stated in [7], because of the complex baseband processing, the power consumption of a radio while receiving is similar to the power consumption for transmitting. Thus, compared to pure transmission power control, these schemes are also beneficial for idle nodes. Additionally, since fewer interrupts occur, also the CPU may save power by switching to a sleep state. The radio will be reactivated for a scheduled event, e.g. for the receipt of a neighbor's beacon. The resulting deafness effects are taken into account differently by 802.11 infrastructure, ad-hoc and mesh power save schemes.

The 802.11s power save schemes are similar to that of the ad-hoc mode, except for two major differences. In ad-hoc synchronization gradually shifts the target beacon transmission time (TBTT) of all nodes to a common time value, which leads to a fully synchronized beaconing. Additionally, nodes contend for sending the beacon in CSMA/CA fashion: a random backoff is employed and a node that already heard a beacon will cancel its own transmission until the next TBTT [13]. This scheme results in a distributed beaconing with a common TBTT. In IEEE 802.11s fundamental differences are that there is no synchronized beaconing and that there are different distinctive power modes. Here synchronization does not shift the TBTT to a common time value. Instead, it focuses on keeping the offsets between the TBTT of neighboring nodes at a constant value. Furthermore, since the beacon contents are highly dynamic compared to ad-hoc mode [14], each node has to send its own beacon. While infrastructure and ad-hoc WLAN employ one single power mode towards all neighbors, in IEEE 802.11s a different power mode can be used for each individual neighbor link of a mesh node. The three defined power modes are active, light sleep and deep sleep mode. For a link in active mode the receiver has to be always active. Thus, no special treatment is needed, but saving energy by suspending the radio is not possible. For a link in light sleep mode the radio can be suspended, but the node has to wake up periodically to receive the beacons of the corresponding neighbor. In case of a link in deep sleep mode a node does not have to wake up for its neighbor's beacons. Instead, it only sends its own beacons and is expected to somehow maintain synchronization with the mesh. The link-specific power mode is indicated in unicast frames towards the corresponding neighbor. If any of the links of a node is in power save mode, this power mode is also indicated in broadcast frames. The procedures to determine which power mode to use for a link are not defined by IEEE 802.11s. Possible options would be a centralized power mode control via a mesh management entity or a distributed control for each individual mesh node by a heuristic algorithm.

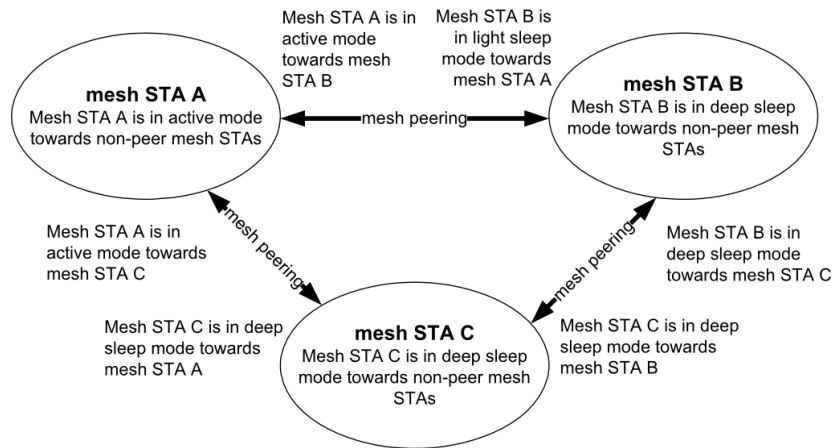


Fig. 1. Illustration of the link-specific power modes of IEEE 802.11s (taken from [1])

Fig. 1 shows an example to illustrate the neighbor-specific power modes. There are three mesh stations (STA): A is in active mode towards both of its neighbors, C is in deep sleep mode towards both of its neighbors, while B is in light sleep mode towards A and in deep sleep mode towards C. In more complex topologies neighbors may also be distinguished into mesh peers and non-peers, which are handled differently.

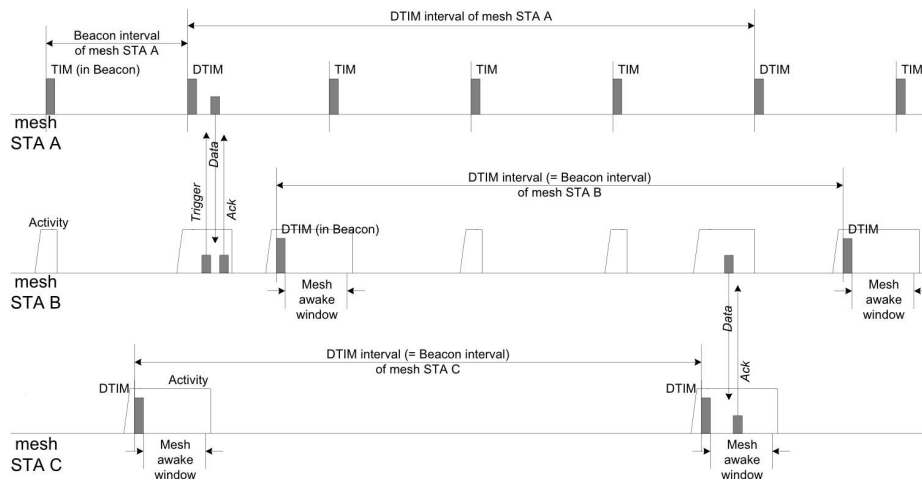


Fig. 2. Example scenario related to the network of Fig. 1 (taken from [1])

Fig. 2 illustrates the behavior of active, light and deep sleep nodes according to the example topology given in Fig. 1. Since A is in active mode towards at least one neighbor, it cannot doze. STA B, which is in light sleep mode towards A, wakes up regularly for receiving A's beacons and for sending its own beacon. STA C in deep sleep does not have to receive neighbors' beacons and only wakes up for its own

transmissions. For unicast frame exchanges Mesh Awake Windows are used. After sending its beacon, a node will keep awake for this specific duration, which is advertised in an Information Element within the beacon. In this time interval neighbors may direct frames to the node. This is illustrated in Fig. 2: as B has data to send to C, it will wake up at the corresponding TBTT and will transmit the data frame within the following awake window. This is possible since deep sleep neighbors are still expected to maintain synchronization. Similar to infrastructure WLAN, beacons contain a TIM element (traffic indication map) which announces data buffered for a neighbor. This neighbor may then poll the buffered frames. This is visible in Fig. 2 where B triggers a frame transmission from A after receiving a beacon. Periodically nodes send a beacon of type DTIM (delivery traffic indication message), in which pending multicast frames are indicated and transmitted immediately afterwards.

When multiple unicast frames should be transmitted, a node may start a Peer Service Period (PSP). A PSP is a variable duration period which is initiated by an acknowledged frame transmission. The PSP re-uses schemes from U-APSD – an advanced power save scheme of infrastructure WLAN [15]. After the initiation the pair stays awake and exchange frames according to the general 802.11 channel access rules until the period is ended. The PSP is started by the setting of RSPI and EOSP flags in the QoS Control field of the frame header. According to the setting of these flags a node stays awake, sends its buffered frames, or does both. The corresponding combinations are shown in Table 1. After the period is ended by an acknowledged frame containing the EOSP flag, both nodes may sleep again.

Table 1. RSPI and EOSP field combinations (taken from [1])

| RSPI | EOSP | Mesh peer service period triggering |
|------|------|---|
| 0 | 0 | One mesh peer service period is initiated. The transmitter of the trigger frame is the owner in the mesh peer service period. |
| 0 | 1 | No mesh peer service period is initiated. |
| 1 | 0 | Two mesh peer service periods are initiated. Both mesh STAs are owners in a mesh peer service period. |
| 1 | 1 | One mesh peer service period is initiated. The receiver of the trigger frame is the owner in the mesh peer service period. |

With this set of rules, a node is allowed to doze when:

- it is in light/deep sleep mode towards all of its neighbors
- it is not within its awake window
- there is no neighbor beacon receipt scheduled
- there is no PSP ongoing
- there is no AP interface active (in case of a mesh gateway towards an infrastructure mode WLAN)

Of course the energy saving comes with a price: higher delay and decreased throughput are the drawbacks of saving energy. Additional to the usual delays caused by CSMA/CA channel access and processing time, power save will introduce a new delay component caused by frame buffering. All unicast frames towards a power save neighbor must be buffered until they may be transmitted in the neighbor's awake

window or during a subsequent PSP. For multicasts the delay may be even higher depending on the DTIM period parameter, since multicasts are transmitted only after the DTIM beacon. Multicasts are often important frames, such as mesh path discovery frames or ARP requests. In these cases multicast buffering will cause an additional delay on the depending unicast frames.

With respect to the PSP implementation, a further problem is that the link delay depends on the link load. Since PSPs are ended when the power save frame buffer is empty, PSPs are variable in duration. For a low link load PSPs will be short compared to the inactive doze times. On the other hand, if the link load is high, PSPs will become dominant until – when the link capacity is fully utilized – there remains no doze time at all. Because of this, the throughput is expected not to be considerably reduced due to power save. However, the throughput might be limited because of two factors. First, since the queue size of the power save frame buffer may be limited for each mesh point, frames may be dropped. The queue size depends on the individual implementation and may be different in heterogeneous mesh networks. For UDP traffic this will cause frequent packet losses depending on the data rate of the sender application, while TCP will adjust the sender data rate. As the buffer queue size determines the maximum number of transmittable packets per beacon interval, additionally, the beacon configuration influences the throughput. The second factor limiting the throughput is the delay caused by power save frame buffering. TCP may interpret high delays as frame losses and may react by throttling the sender data rate.

4 Testbed implementation and results

In our testbed implementation we use off-the-shelf wireless routers. These have been re-programmed with OpenWrt, a Linux distribution for embedded devices [16]. The Linux kernel module for wireless LAN – “mac80211” [17] – has been extended with the 802.11s power save techniques. mac80211 is a “SoftMAC” framework module, which resides above the hardware-specific driver module in the kernel protocol stack. In this module most of the MAC layer management functions may be performed in software. For the case of 802.11s mesh networks this includes the Peer Management as well as the routing protocol HWMP and – with our extensions – also the power save and mesh synchronization routines. The mesh power save implementation has been started by members of the Open Source and Linux Lab [18], who implemented power save state management and indication [19]. We contributed and extended the implementation with the routines for synchronization, frame buffering and PSPs. Yet to-be implemented are the routines for deep sleep mode and the actual doze state (currently the radios are not actually suspended).

In the following we focus on the very basis of each mesh network: an individual mesh link between two mesh peers A and B. Concerning power save, we examine four cases for this mesh link: both peers are active, either A or B is in light sleep mode or both nodes are in light sleep mode.

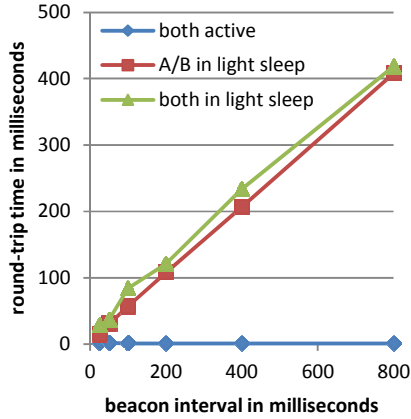


Fig. 3. Average round-trip time of probe packets for different beacon interval values (10 probes/s, 30s averages, no additional load)

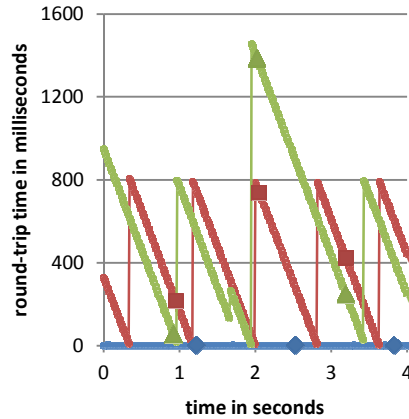


Fig. 4. Round-trip time of individual probe packets (100 probes/s, 800ms beacon interval, no additional load)

Fig. 3 shows the average round-trip time (RTT) on a mesh link of high signal quality for different beacon interval values. The RTT is measured with ICMP echo requests and replies with a rate of 10 requests per second. In this measurement setup the link load is caused only by the probe packets themselves, leading to a very low load. It is visible that the beacon interval has no considerable effect on active mode performance. Here – without power save – the RTT is constantly at about 1ms. This is achieved with the radio being always active, listening for incoming packets. If a single peer is in power save mode, the average RTT increases linearly with the beacon interval. That is because with the beacon interval also the time between PSPs – the time of inactivity – increases. Thus, the mean buffering delay of packets is longer. For both peers in power save mode the RTT is slightly higher as explained below.

The exact behavior of the RTT is illustrated in Fig. 4 for a beacon interval of 800ms. In active mode the RTT is almost constant for the individual probes. In light sleep mode, power save causes delay in form of a saw tooth curve. The falling slope is formed by frames being queued in the power save frame buffer, where they are delayed until the next TBTT of the power-saving node. The rising edge slope marks a PSP in which all buffered frames are released. A frame which is queued right before the PSP will experience the minimum delay, while a frame that is buffered right afterwards will suffer the maximum delay. The saw tooth lower value of about 10ms is caused by the delay of in-order transmission of the frames buffered during the inactive time. The saw tooth upper value and width are determined by the beacon interval. Thus, for low link load the RTT on a single power save mesh link is linearly dependent on the configured beacon interval and the mean value is approximately half of the beacon interval. The high RTT variation, i.e. the saw tooth spikes, may be harmful for overlaying protocols, as we will show later. If both nodes are in power save mode, there are two sources of delay, as frames will be buffered at node A and at node B. But currently, in our implementation the PSPs are always started bi-directionally (with $RSPI=1$ and $EOSP=0$). This means that both A and B may transmit

their buffered frames when either one starts a PSP. Thus, there are two opportunities for frame release and the overall RTT is not generally increased. The reason for the slightly higher RTT, if both nodes are in light sleep, is seen in the exceptionally high spike of Fig. 4. These occur in situations when the first node has already ended the PSP, while the second has not. In this case the second node will not initialize a new PSP and thus the first node misses one opportunity to transmit. The TBTT offset of both nodes is visible in the additional peak height.

Not shown in the graph is the effect of the DTIM period parameter. This value has an impact on all multicast frames and unicasts which depend on them. For example, when HWMP broadcasts a Path Request to resolve a path to a destination, all frames addressed to that destination will be delayed until the path is resolved. Only after the frames are addressed according to the resolved path, they will be sent – or queued in the power save unicast buffer if the neighbor is in power save mode. Since multicast frames are sent only after DTIM beacons, these and depending frames may be delayed by a maximum of the beacon interval multiplied by the DTIM period.

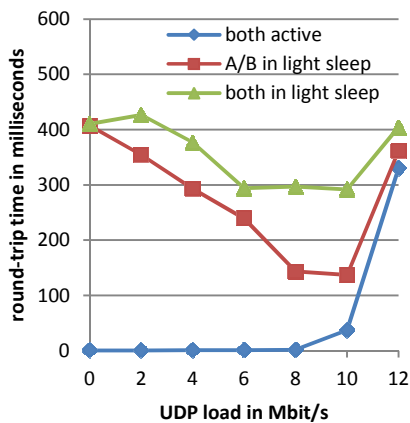


Fig. 5. Average round-trip time of probe packets for different UDP load values (10 probes/s, 30s averages, 800ms beacon interval, symmetric UDP load)

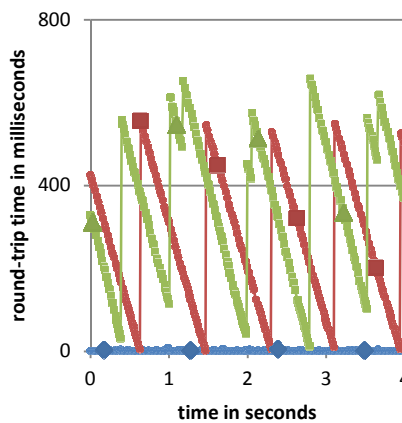


Fig. 6. Round-trip time of individual probe packets (100 probes/s, 800ms beacon interval, 4Mbit/s symmetric UDP load)

Additional to the beacon interval also the link load and the resulting queue fill level influence the RTT of a mesh link in power save mode. This can be seen in Fig. 5 where the RTT for a beacon interval of 800ms and variable UDP load is given. The UDP traffic is generated symmetrically in both directions. In our setup the routers are close to each other and usually apply a PHY layer transmission rate of 54Mbit/s. The RTT is again measured with ICMP echo requests and replies. As a reference, in active mode the load measurably affects RTT only when the incoming load is similar to or higher than the current link capacity. If a single peer is in power save mode, the maximum RTT is experienced when the load is very low. With increasing load the RTT is reduced linearly until the link capacity is reached. Similarly, if both peers are in power save mode, the RTT decreases with increasing load.

Fig. 6 shows an in-detail analysis of the RTT for a load of 4Mbit/s. Compared to Fig. 4 it can be seen that the spike height of the saw tooth curve is reduced. This is due to the dynamic duration of the PSPs: PSPs are ended when there are no further frames buffered in the power save queues. If the interarrival rate of to-be sent packets is high, the duration of PSPs is increased. This reduces the duration of the inactivity periods which determines the maximum RTT. For further increasing load PSPs may even span over multiple beacon intervals. If both nodes are in power save mode, it often happens that some probe packets miss one PSP at either node. That is because under real-life conditions the duration of the PSPs are not equal for both nodes and not constant over time. The effect of this behavior is visible in the two-peak saw tooth of Fig. 6, where the second peak is formed by packets which missed one PSP that has been ended prematurely. This is also the reason for the increase of RTT that is visible at 2Mbit/s load in Fig. 5. It should be noted that, if both nodes are in power save mode, the RTT is also influenced by the TBTT offset of both nodes. This value is determined by the arbitrary startup time of the nodes and is kept constant by the synchronization algorithm.

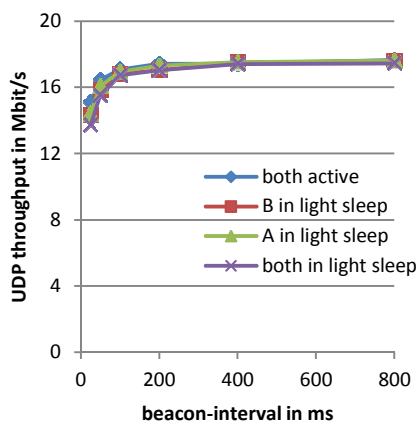


Fig. 7. Average UDP throughput for different beacon interval values (30s averages, asymmetric load)

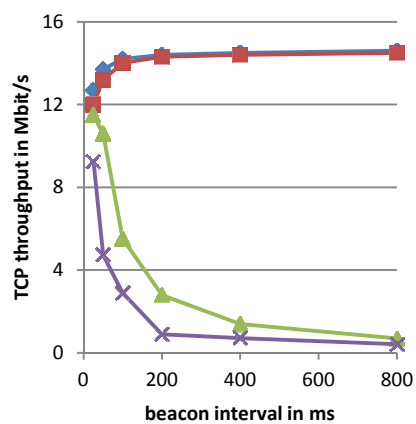


Fig. 8. Average TCP throughput for different beacon interval values (30s averages, asymmetric load)

Fig. 7 shows the maximum UDP throughput of a mesh link in active and power save mode for variable beacon interval values. In active mode the throughput degrades for unusually low beacon interval values, where the high beacons overhead lowers the achievable throughput. It is visible that in power save mode, due to the dynamic PSP length adjustment, the available throughput remains quite constant. A loss of throughput would happen if the size of the power save frame buffer is set too low. In the Linux kernel usually a maximum queue size of 64 packets per neighbor STA and access category is predefined to prevent memory shortage. The oldest buffered frames are dropped if the maximum size of the buffer is exceeded. Thus, the buffer size determines the number of packets that may be transmitted per beacon interval towards

a power saving node. Since mesh routers are expected not to be limited in memory, we set the buffer size to a value of 2048 frames to avoid these cut-off effects.

A different behavior can be seen in Fig. 8 where TCP traffic passes a link in power save mode. Although TCP transports data bi-directionally, typically the load in forward and backward direction is not symmetric. For example, in a file download only the TCP ACKs will traverse the backward route. As we have previously seen in Fig. 5 and Fig. 6, the RTT and RTT variation of a 802.11s mesh link in power save mode depend on the link load. Thus, for a TCP file transmission the delay spike height of the high-load forward route from A to B will be low, while it will be considerably higher for the low-load backward route from B to A. The effects on TCP (Reno) are visible in Fig. 8: if the node generating the traffic (i.e. sending the file) is in power save mode, the throughput is reduced drastically. That is because TCP interprets the sharp jumps of delay on the backward route (i.e. the spikes of Fig. 4) as losses and reduces the forward sender data rate – although the link is well capable of carrying the load as seen in Fig. 7. On the other hand, if only the destination is in power save mode, the TCP throughput is unaffected. This is because on the high-load forward route the delay spike height is low due to the long PSPs, while on the backward route the delay variation is also low, as in our implementation a node in power save mode will briefly interrupt its inactive state when it has data to send – in this case the TCP ACKs. If both nodes are in power save mode, the TCP throughput is further decreased, because then the average RTT and its spike height are at their maximum values. Additionally, from Fig. 8 it can be seen that, if only the traffic source or both source and destination are in power save mode, the throughput is inversely proportional to the beacon interval. That is because with higher beacon interval also the harmful RTT spike height is increased.

5 Conclusion

In this paper we analyze the effects of IEEE 802.11s power save on mesh link performance. We show that the average traffic RTT linearly increases with the beacon interval of the power saving nodes. The RTT of individual packets follows a saw tooth curve determined by the nodes' beacon interval. Furthermore, the RTT depends on the current link load as nodes prolong their PSPs under high traffic. Although this dynamic PSP length adjustment mitigates the degradation of UDP throughput, TCP throughput is throttled because of the high delay spikes on the low-load backward route. Our analysis shows that effects on RTT and throughput depend on the mesh configuration and the network parameters. Moreover, the mesh link throughput is influenced by the 802.11s implementation – in our case the maximum queue size of the Linux kernel and the implemented type of PSPs.

In our future work we intend to perform energy consumption measurements in our testbed, which will allow us to relate the negative effects of power save to the actual energy savings. Furthermore, we will examine the effects of power saving on actual mesh topologies. We are also working on integrating our code into the mainline Linux kernel.

References

- [1] IEEE Std. 802.11™, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 10: Mesh Networking”, IEEE Computer Society, Sept. 2011
- [2] IEEE Std. 802.11™-2007, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Computer Society, Jun. 2007
- [3] IEEE Std. 802.11™-2012, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Computer Society, work in progress, <http://standards.ieee.org/findstds/standard/802.11-2012.html>
- [4] In-Stat, “Wi-Fi Market Overview: Connectivity Becoming Ubiquitous”, Nov. 2010
- [5] Linux Kernel changelog, http://kernelnewbies.org/Linux_2_6_26
- [6] Jardosh, A., Papagiannaki, K., Belding, E., Almeroth, K., Iannaccone, G., Vinnakota, B.: “Green WLANs: On-Demand WLAN Infrastructures”, *Mobile Networks and Applications*, Vol. 14, No. 6., Dec. 1, 2009, pp. 798–814
- [7] Karl, H.: “An overview of energy-efficiency techniques for mobile communication systems”, Report of AG Mobikom WG7, Oct. 19, 2003
- [8] Todd, T.D., Sayegh, A.A., Smadi, M.N., Dongmei Zhao: “The need for access point power saving in solar powered WLAN mesh networks,” *Network*, IEEE , vol.22, no.3, pp.4–10, May-June 2008
- [9] Capone, A., Malandra, F., Sansò, B.: “Energy Savings in Wireless Mesh Networks in a Time-Variable Context”, *ACM/Springer Mobile Networks and Applications*, pp. 1–14, 2011
- [10] Esam, M., Ashour, M.: “Cooperative Game Strategy for IEEE 802.11s Mesh WLAN Power Management”, ICC 2011 IEEE International Conference on Communications, 5-9 June 2011, pp. 1--5
- [11] Avallone, S.: “Turning off radios to save power in multi-radio wireless mesh networks”, *MobiHoc '11 Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2011
- [12] Alam, M.N., Jäntti, R.: “Performance Analysis of the IEEE 802.11s Power Save Mode for One Peer Link Operation”, AWA Seminar, May 25, 2011, <http://www.netlab.tkk.fi/tutkimus/awa/seminar-25052011.shtml>
- [13] Yu-Chee Tseng, Chih-Shun Hsu, Ten-Yueng Hsieh: “Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks”, *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 200–209
- [14] Safonov, A., Lyakhov A., Sharov, S.: “Synchronization and Beaconing in IEEE 802.11s Mesh Networks”, *ICT 2008. International Conference on Telecommunications 2008*, pp.1--6, 16-19 June 2008
- [15] Kneckt, J., Marin, J., Kasslin, M.: “Some Responses to questions regarding mesh power management”, IEEE P802.11 Wireless LANs, document IEEE 802.11-09/146r0, Jan. 21, 2009
- [16] OpenWrt GNU/Linux distribution for embedded devices, <https://openwrt.org/>
- [17] mac80211 – GNU/Linux wifi driver framework, <http://linuxwireless.org/en/developers/Documentation/mac80211>
- [18] Open Source and Linux Lab, <http://osll.spb.ru/projects/meshpower/>
- [19] Yuranov, D.: “IEEE 802.11s Power Save Modes Support Implementation for Linux”, FRUCT 2011. 2nd Finnish-Russian Mobile Linux Summit, Oct. 5, 2011