

# Improving TCP Performance in IEEE 802.11 Networks

A. Statkus<sup>1</sup>, S. Paulikas<sup>1</sup>

<sup>1</sup>*Department of Telecommunication Engineering,  
Vilnius Gediminas Technical University,  
Naugarduko St. 41–211, LT-03227 Vilnius, Lithuania*  
sarunas.paulikas@vgtu.lt

**Abstract**—With growing number of wireless devices in the field, the main and sometime the only way to access the Internet is wireless networks based on IEEE 802.11 standard. In most cases these wireless Internet access links are bottlenecks. A fair and efficient utilization of these links is becoming more complicated and to deal with that have higher layer protocols. Today the main such protocol is transport control protocol (TCP). However, due to big protocol overhead of IEEE 802.11 and TCP, performance degradation of the wireless link utilization is observed. In this paper the acknowledgment frames (ACKs) filtering solution for reducing TCP protocol overhead in IEEE 802.11 wireless networks is presented. The results of performed experiments not only show the increase of TCP throughput but also the reduce of the wireless network saturation without any considerable influence to TCP functionality.

**Index Terms**—ACK filtering, IEEE 802.11 ACK, TCP performance.

## I. INTRODUCTION

Today rapidly developing Internet significantly relies on mobile or wireless (IEEE 802.11a/b/g) technologies, which allow easily and low cost expansion of the Internet coverage. According to IHISiSupply research, during the next year the Wi-Fi chip-sets shipment should exceed 1 billion units [1].

With growing number of wireless devices in the field a fair and efficient link utilization is becoming more complicated [2]. The solution of this problem is mostly left for higher layer protocols. The today's main protocol that has to deal with it is Transport Control Protocol (TCP). TCP is a reliable connection-oriented protocol that delivers the flow control of data packets in various types of networks by utilising a sliding window and acknowledgment technique together with congestion avoidance. TCP is extensively used by many network applications and thus delivers high interoperability.

During the years many TCP congestion avoidance mechanisms were developed and implemented in different TCP versions. Most of them, e.g. High Speed TCP (HSTCP) (documented in RFC 3649), Cubic, Fast (documented in RFC 3782) were targeted to high speed networks. Mentioned TCP implementations use acknowledgment

frames (ACKs) and adjustment of sliding window size [3]. Sliding window size is adjusted in real time according to received ACK. Thus, the peers are allowed to increase the sliding window if positive ACK is received. This leads to an increase of data transfer rate at TCP level. That is why, the undisturbed exchange of ACKs for efficient flow control is desired. Any disturbances of ACKs flow influence the efficiency of TCP data transfer [4], [5].

In opposite to TCP, which tries to find the maximum throughput of link, the IEEE 802.11 wireless standard is more concentrated on successful data delivery between wireless nodes. By implementing the layer 2 acknowledgment mechanism the IEEE 802.11 standard is basically trying to hide the wireless link problems and packet drops from user application and higher layer protocols. This not only can reduce the recovery time after packet drop, but also allows sustaining stable and guaranteed throughput of wireless links.

Similar to TCP, the IEEE 802.11 acknowledgment mechanism is based on small ACK informational frames, which are sent after receiving each data frame. The only difference is that TCP ACKs are sent not for every TCP data frames and they are cumulative, the newest ACK obsoletes the older one. Like TCP ACK the IEEE 802.11 ACK frames have a big protocol overhead and in some cases can dramatically reduce wireless link capacity.

The poor performance of short frame traffic for TCP as for IEEE 802.11 is well known and analysed in detail [6], and [7]. The degradation of throughput occurs due to technological expenditures of wireless link [7]. It is clearly observable on high throughput links [8]. Therefore, with increasing rate of TCP data segments, the rate of TCP ACK as well as rate of IEEE 802.11 ACK is increasing rapidly, too. Here, in addition to increasing protocol overhead, radio environments saturation is also increasing due to big number of IEEE 802.11 ACK messages and that leads to network performance degradation [10].

One of the ways to increase TCP performance in IEEE 802.11 wireless links is to reduce the number of TCP ACK packets. This can be achieved using ACK filtering technique [9], [10] that is already implemented and employed in asymmetric networks such as wireless networks [9], however, still considered to be inconsistent with TCP traffic and in some cases could lead to performance degradation.

The second and less analysed choice, which has less influence on TCP, is to disable the IEEE 802.11 ACK messages for TCP ACK packets, by implying selective ACK filtering on wireless interfaces.

The aim of current paper is to analyse how this theoretical selective ACK filtering mechanism works in practice, does it influence the TCP functionality, and what TCP performance gain can be achieved in IEEE 802.11 networks.

## II. TCP ACKNOWLEDGMENT MECHANISM

The TCP packet acknowledgement mechanism is in the core of TCP and depends on the demand of the receiver to communicate with the transmitter by means of sending back an ACK packet when it successfully receives data. TCP employs a cumulative ACK mechanism, documented in RFC 813. The number field in a TCP packet header received by a transmitter says that all previous data packets with sequence numbers less than indicated have been successfully received at the receiver.

Usually, after receiving data TCP doesn't send an ACK instantly. According to RFC 2581 it delays ACK, expecting that there are more data packets sent in the same direction. A typical TCP stack uses timer set up to 200 ms and replays with ACK for next data packet.

The TCP transmitter uses ACK to find out how many packets can be in the transmission in the network without them being dropped in queues of routers. For this task it uses a sliding congestion window, which length function  $W(t)$  depends on the received ACK. After the TCP session initialization, transmitter is permitted to increase the  $W(t)$  for each successfully received ACK.

The TCP data packet retransmission can happen, because of two main reasons: when, according to RFC 813, retransmission timer expires and when three or more duplicated ACK are received.

The first case, the expiration of retransmission timer, occurs after no new packets are acknowledged for a predefined period of time, according to RFC 2581, 200 ms. The retransmission timeout  $t_{rto}$  is treated as indication of a packet loss, and it evokes retransmission of the unacknowledged data packets. The time interval during which the confirming ACK packet must be received can be calculated as

$$t_{rto} = \overline{t_{rtt}} + 4 \cdot \sigma_{rtt}, \quad (1)$$

where  $\overline{t_{rtt}}$  – the mean time of successful TCP packet transmission,  $\sigma_{rtt}$  – the standard deviation of  $t_{rtt}$ .

If during  $t_{rto}$  the ACK is not received, the data packets loss will be detected and according to RFC 2988 the TCP transmitter will set sliding window length  $W(t)$  to 1 packet. Thus  $t_{rto}$  indicates that channel throughput has decreased drastically [10].

The second case occurs when three or more duplicated ACK have been received. After duplicated ACK is detected, the transmitter stops sending packets until  $t_{rto}$  to expire. The RFC clearly states that current ACK technique must retain two very important features: prevent the retransmission of data packets, and as soon as possible send ACK to allow further transmission of data packets.

In addition to that requirement the following must be taken into account: the ACK packets are very important at the initial phase of data transmission and the rate of ACKs, which relies on data packets rate, cannot be less than  $1/t_{rto}$  [11].

## III. IEEE 802.11 ACKNOWLEDGMENT MECHANISM

The IEEE 802.11 MAC protocol provides a fair access to the shared wireless medium through two different access mechanisms: polling-based protocol, called the Point Coordination Function (PCF) and contention-based access protocol, called the Distributed Coordination Function (DCF).

The DCF is more frequently implemented, as compared with PCF and it is used for automatic medium sharing between compatible stations. This is done by means of the CSMA/CA mechanism. Before a station can send data, it must sense the wireless medium to determine if it is free. If it is so, the transmission may proceed, otherwise the station will wait until the end of the in-progress transmission [12]. The CSMA/CA requires a minimum specified space gap between frame transmissions to ensure that the medium has been idle for the specified time period.

To ensure that the DCF uses two basic rules: the medium must be free more than for DIFS (Distributed Inter-Frame Space time) interval, and to reduce the probability of two stations sending data at the same time, the stations must wait a random back of period –  $T_b$  [12]

$$T_b = \text{Random}(CW) \cdot T_{\text{slot}}. \quad (2)$$

The  $\text{Random}(CW)$  is a pseudo-random integer value selected from the interval  $[0, CW]$  and the  $T_{\text{slot}}$  value corresponds to the Slot Time which is a physical characteristic of IEEE 802.11 standard.

Like in TCP the IEEE 802.11 MAC uses ACK messages upon successful reception of a data frame. This ACK frame is sent before other data frames, the wireless client wait small time gap SIFS (Short Inter-Frame Space) before transmitting the ACK frame. This frame exchange sequence prevents other stations from sending data by requiring waiting for the medium to be idle for a longer time than SIFS period before attempting using the medium. This allows to successfully complete the frame exchange sequence.

If no ACK frame is received within a SIFS interval due possibly to an error in reception of the preceding data frame, the transmitter will contend again for the medium to retransmit the frame after an ACK timeout Fig. 1.

In basic DCF mode of IEEE 802.11 MAC standard the throughput of wireless link can be calculated based on following function

$$\text{Throughput} = 8 \cdot L_{\text{data}} / T_{\text{dcf}}, \quad (3)$$

where  $L_{\text{data}}$  – the length of payload size of the IEEE 802.11 data frame;  $T_{\text{dcf}}$  – the total time needed to make the successful full frame exchange sequence. The  $T_{\text{dcf}}$  time consists of sum of times needed to make frame exchange in

wireless link

$$T_{dcf} = T_{difs} + T_b + T_{data} + T_{sifs} + T_{ack}. \quad (4)$$

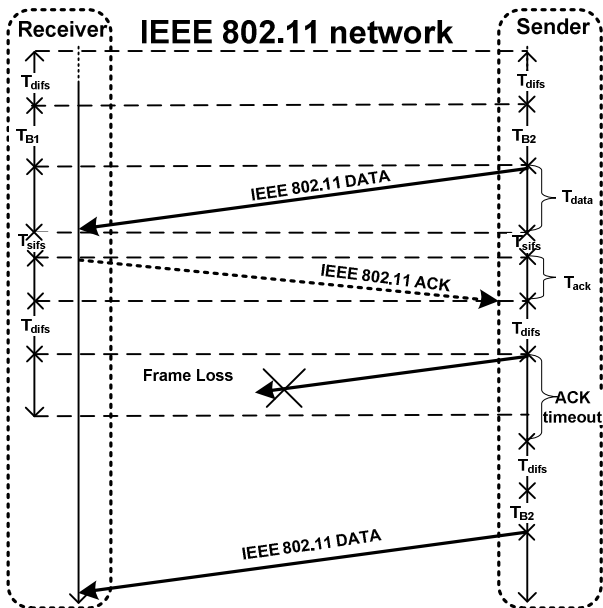


Fig. 1. IEEE 802.11 frame exchange.

The  $T_{pre}$ ,  $T_{phy}$ ,  $T_{difs}$ ,  $T_{sifs}$  variables are constants and depend on different IEEE 802.11 standards. The  $T_{data}$  and  $T_{ack}$  time frames are calculated based on payload size, ceiling function and sending data rate of physical wireless media or from  $T_{ofdm}$  – the transmission time of one OFDM symbol and  $N_{dbps}$  – the number of data bits per OFDM symbol:

$$T_{data} = T_{pre} + T_{phy} + T_{ofdm} \cdot \left\lceil \frac{22 + 8 \cdot (L_{mac} + L_{data})}{N_{dbps}} \right\rceil, \quad (5)$$

$$T_{ack} = T_{pre} + T_{phy} + T_{ofdm} \cdot \left\lceil \frac{22 + 8 \cdot L_{ack}}{N_{dbps}} \right\rceil. \quad (6)$$

In our research we used IEEE 802.11a standard (with lowest saturation nowadays) operating in the 5 GHz frequency band. The time variables for IEEE 802.11a are as follow:  $T_{pre} = 16 \mu s$ ,  $T_{phy} = 4 \mu s$ ,  $T_{difs} = 34 \mu s$ ,  $T_{sifs} = 16 \mu s$ , the  $N_{dbps}$  are selected based on physical rate except the control response frame (CTS or ACK), which must be transmitted using the mandatory rates or rate belonging to Basic Rate Set [11]. In our research a 24 Mbit/s physical data rate of IEEE 802.11a media was used.

#### IV. IEEE 802.11 ACK FILTERING TECHNIQUE

The concept of IEEE 802.11 ACK filtering can be used not only to reduce wireless media saturation and increase TCP throughput, but also to apply intelligent QoS by filtering data flows based on upper layer criteria. Due to simplicity of the IEEE 802.11 mechanism this can be done by changing the frame type from data to management or changing the Ethernet frame destination field from Unicast to Broadcast or Multicast before it enters the IEEE 802.11

network interface. By doing it we not only eliminate the IEEE 802.11 ACK frames, and change the sending frame type but also remove unneeded timers (SIFS, back off EIFS), that additionally increase wireless links capacity.

The theoretical throughput of the IEEE 802.11 link without ACK can be calculated based on (8)

$$\text{Throughput} = \frac{8 \cdot L_{data}}{T_{difs} + T_b + T_{data}}. \quad (8)$$

In Fig. 2 a theoretical throughput of the IEEE 802.11a wireless link is shown based on (9). With very small frames the throughput can be increased up to 30 % compared with the default behaviour but only if the wireless link has very few or no errors.

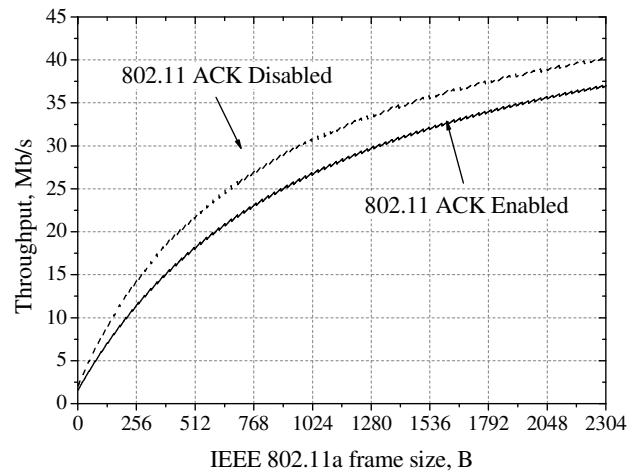


Fig. 2. Throughput increase with IEEE 802.11 ACK filtering.

#### V. PERFORMANCE EVALUATION EXPERIMENTS

In order to find out what real influence of the TCP and IEEE 802.11 ACK filtering gives to the TCP performance and how it influences the functionality of the whole wireless network, a few experiments were carried out. For these experiments, a wireless network made of two access points (AP) connected to two PCs via 100 Mbps Ethernet links (100BaseTx) was used. These PCs were running Linux OS, (2.6.32 kernel) with enabled TCP Cubic congestions avoidance mechanism. The PC2 was configured to act as FTP client that sends TCP data packets to PC1. The PC1 was connected to Wi-Fi AP via 100 Mbps Ethernet link and configured as TCP server that receives TCP data packets and sends back ACKs. Between these nodes the TCP session was established.

The IEEE 802.11 ACK filtering or modification of Ethernet destination field from Unicast to Multicast was performed on Wi-Fi AP, by using Linux `<ebtables>` application, which filtered and changed TCP ACK frames from Unicast to Multicast. Due to this the Wireless adapter driver treated the frames as informational and did not wait for acknowledgment frames. After the frame reached the wireless client the frames were changed back to Unicast, using the same application, so that the PC1 did not notice any distortions in received Ethernet frames.

In Fig. 3 we can see the default behaviour of the TCP and wireless link, when no ACK filtering was used (Filtering

OFF). It is clearly seen that wireless link overhead is very high in IEEE 802.11 ACK frames. These frames are sent not only for TCP data frames but also for TCP ACK frames. This generates almost half of all transmitted IEEE 802.11 frames. In total the IEEE 802.11 layer 2 frame rate exceeds 600 frames per second.

During the default test scenario (Filtering OFF) the average TCP throughput did not exceed 26 Mbit/s, that is considered the maximum practical speed for TCP throughput in IEEE 802.11a wireless networks [13].

By applying wireless link IEEE 802.11 ACK filtering on TCP ACK frames, we not only reduced the wireless frame rate more than 16 % but also increased the TCP good put more than 1.5 Mbit/s, up to 27.5 Mbit/s (see Fig. 4).

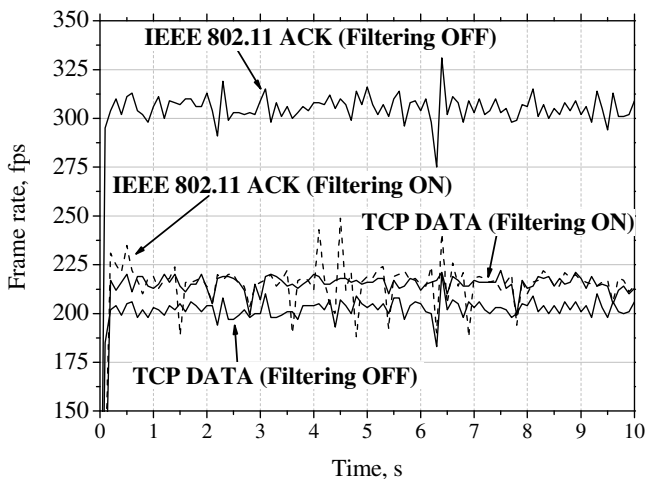


Fig. 3. Default TCP message rate in IEEE 802.11a.

This is almost the maximum theoretical throughput of IEEE 802.11a link without ACK for TCP protocol calculated according to equation (9). But most importantly, the used wireless filtering mechanism did not impact the TCP good put and stability (Fig. 4) of the TCP session during all transmission period, and also instability of the TCP session recovery after packet drop was not observed.

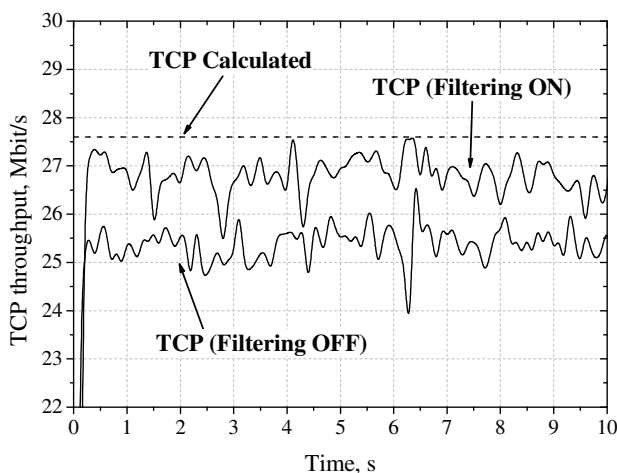


Fig. 4. TCP throughput with IEEE 802.11 filtering ON and OFF.

This was done based on the cumulative TCP ACK technique and threshold time during which the confirming ACK message must be received (1).

Of course the wireless filtering could be implied to reduce the unneeded technical overhead of TCP data packages, however it could be useful only in wireless link conditions with no or very few package drops.

## VI. CONCLUSIONS

In this paper the investigation of TCP throughput and impact of the TCP ACK IEEE 802.11 filtering was presented. The experiments were performed to find out how the filtering of IEEE 802.11 ACKs influences the normal wireless conditions and what impact it has on TCP functionality and performance. We conclude that:

- The presented IEEE 802.11 ACK filtering mechanism has no impact to TCP session stability and good put under normal wireless network conditions.
- Using IEEE 802.11 ACK filtering in TCP session the frame rate over wireless link can be reduced up to 16 % (reduce radio environments saturation) and good put increases up to 1.5 Mbit/s without impacting the TCP session stability (see Fig. 4).
- By using IEEE 802.11 ACK filtering on upper layer of Ethernet protocol (TCP, UDP, ICMP, etc.) an optimized QoS mechanism could be used to increase the wireless link throughput or reduce RF saturation.

## REFERENCES

- [1] R. Braverman, "Wi-Fi Chipset Shipments to Double in 2011", 2011. [Online]. Available: <http://www.isuppli.com/Mobile-and-Wireless-Communications/MarketWatch/Pages/Wi-Fi-Chipset-Shipments-to-Double-in-2011.aspx>
- [2] A. Kajackas, A. Šaltis, "User Access Link Impact on Web Browsing Quality". *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, no.4, pp. 59–64, 2010.
- [3] V. Jacobson, "Congestion avoidance and control". in *Proc. of the Conference IEEE (SIGCOMM)*, 1988, vol. 1, pp 314–329.
- [4] A Balachandran, G. M. Voelker, P. Bahl, R. P. Venkat, "Characterizing user behavior and network performance in a public wireless LAN". in *Proc. of ACM Sigmetrics. Marina Del Rey*, pp. 195–205, 2002.
- [5] L. Pavilanskas, "Analysis of TCP algorithms in the reliable IEEE 802.11 b LINK". in *Proc. of ASMTA*, pp. 25–30, 2005.
- [6] K. Jain, J. Padhye, V. Padmanabhan, L Qiu, "Impact of interference on multi-hop wireless network performance". in *Proc. of the MobiCom*, pp. 66–80, 2003.
- [7] R. Sharma, G. Singh, R. Agnihotri. "Comparison of performance analysis of 802.11a, 802.11b and 802.11g standard". *International Journal on Computer Science and Engineering*, no. 6(2), pp. 66–80, 2010.
- [8] Li Tianji, Ni Qiang, D. Malone, D. Leith, Y. Xiao, T. Turtletti, "Aggregation With Fragment Retransmission for Very High-Speed WLANs". *IEEE/ACM Transactions on Networking*, vol. 17, pp. 591–604, 2009. [Online]. Available: <http://dx.doi.org/10.1109/TNET.2009.2014654>
- [9] H. Wu, J. Wu, Sh. Cheng, J. Ma, "ACK Filtering on Bandwidth Asymmetry Networks". in *Proc. of IEEE: APCC/OECC*, pp. 175–178, 1999.
- [10] P. Karn, "Dropping TCP ACK's", *Mailing list*, 1996, [Online]. Available: <ftp://ftp.isi.edu>
- [11] L. Pavilanskas, A. Statkus, "Evaluation of TCP Acknowledgment Mechanism Influence on Router Performance". *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, no. 7, pp. 95–100, 2010.
- [12] *IEEE std 802.11-1999 Part 11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec.*, 1999.
- [13] M. Gast, *802.11 Wireless Networks: The Definitive Guide, 2<sup>nd</sup> ed.*, O'Reilly Media, 2005.