

Performance Evaluation of TCP in an Integrated WPAN and WLAN Environment

Isameldin M. Suliman, Janne Lehtomäki, and Ian Oppermann
Centre for Wireless Communications
P.O. Box 4500, 90014 University of Oulu, Finland
isam@ee.oulu.fi

Abstract—Short-range low power radio frequency systems such as Bluetooth and UWB enable the deployment of wireless personal area networks (WPAN). A WPAN can interface to larger networks to provide broader network access and Internet connectivity. We evaluate the performance of TCP over an integrated WLAN and WPAN system using a real network testbed. The end-to-end throughput is found to increase, when the window size is increased. However, for large window sizes, measurement results revealed that a wide range of round trip times (RTTs) is experienced. With small window sizes, the variability in the RTT is smaller. The measurements showed that in an integrated network long RTT delays and frequent duplicate acknowledgements lead to an increase in the number of packets transmitted unnecessarily. We study the effect of the number of active short-range devices (Bluetooth in this case) in a piconet. The results show that as the number of active slaves present in the WPAN increases, the bandwidth received by slaves exchanging data decreases. This behaviour is attributed to the Bluetooth scheduling mechanism which uses the round robin polling method. The main finding of our experiments is that for Bluetooth to be successful in enabling WPAN, the time slot allocation scheme should be efficient and fair. Finally, the issue of fair bandwidth allocation among multiple TCP streams is also investigated. The results show that for small number of TCP connections, the bandwidth is fairly distributed. However, as the number of simultaneous TCP connections increases, bandwidth distribution seems to become somewhat less fair.

Index Terms—Fairness index, integrated network, short-range communication, WLAN, WPAN.

I. INTRODUCTION

The growing number of mobile computing devices such as laptops, personal digital assistants (PDAs), as well as electronic devices in the home has created a demand for wireless personal area networks (WPANs). A WPAN is a personal area network for interconnecting devices centered around an individual's personal operating space (POS) in which the connections are wireless. The IEEE 802.15 family of standards supports various WPANs, for example, ultra wideband (UWB), Bluetooth, and ZigBee. A WPAN supports communication over relative short distances, typically, the communication range is about 10 metres. Based on this WPAN infrastructure, users can access various data services and computing resources through individual devices. Since WPANs have a limited geographically coverage, they can interface to larger networks to provide access to various computing and networking resources available on and through these networks. A number of recent

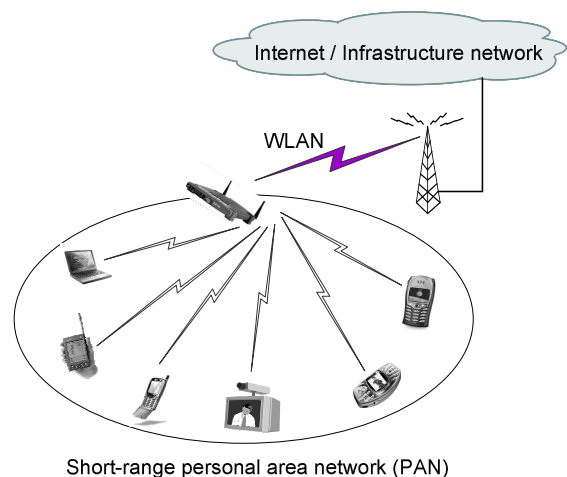


Fig. 1. An integrated WPAN and WLAN network.

efforts focusing on WPAN research describe the coupling of the short-range WPAN networks with large infrastructure networks [1], [2], and in particular considering UWB in the core of personal networks [3]. Figure 1 presents an integrated short-range WPAN and WLAN network.

Bluetooth is one of the widely deployed WPAN technology in home appliances and other devices. The Bluetooth system [4] provides a point-to-point connection when only two Bluetooth units are involved, or a point-to-multipoint connection when the channel is shared among several Bluetooth units forming a piconet. One Bluetooth unit acts as the master (access point) of the piconet, whereas the other units act as slaves. The master controls all communication in the piconet and all data traffic must be routed through the master. UWB is another wireless technology that aims at very low power consumption with some additional functionalities such as positioning and localization, and is expected to provide low cost wireless connectivity in WPAN environment. The UWB technology offers a very wide frequency spectrum of 3 to 10 GHz and can support multiple applications such as radar imaging, health care systems, and ad-hoc networking. Because of its high data rate capabilities, UWB is ideal for high-quality multimedia networking between home electronics and entertainment

devices. The advantages of UWB can be summarized as follows: robustness to multipath fading, spectrum efficiency, low transmitted power, low probability of intercept/detection, and the ability of many users to coexist with little mutual interference. A tutorial overview of UWB radio technology can be found in [5].

Since transmission control protocol (TCP) is the most commonly used transport protocol that provides reliable data transfer for Internet applications, evaluating its performance in emerging wireless networks is invaluable for a variety of purposes including identifying protocol limitations and problems so that solutions can be proposed for improving the protocol. TCP was not originally designed for wireless communications, where the communication is restricted by the mobility of nodes, temporary disconnection, coverage area, high bit error rates, and on limitations on bandwidth. These restrictions make operation of TCP protocol on wireless networks a challenging task and lead to poor performance of the TCP protocol. TCP responds to all packet losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance. Various techniques for improving TCP in wireless networks have been considered [6]. A number of studies have investigated the performance of TCP on these heterogenous networks including Bluetooth and WLAN [7], [8]. Since Bluetooth and WLAN operate in the same frequency band, radio interference can limit performance. These issues have been investigated in a work presented by [9].

This paper evaluates the performance of TCP protocol in an integrated WLAN and WPAN network. The paper is organized as follows. In Section II, we present the integrated network testbed, which uses 802.11b and Bluetooth technologies. The Bluetooth WPAN was chosen as an example of short range low power technologies, though the experiment measurements described are also applicable to most short radio systems (for instance, UWB). Experiment measurements and TCP Performance evaluations are presented in Section III. Finally, conclusions and direction for future work are presented in Section IV.

II. TESTBED NETWORK

To perform our experiments, we have set up a network testbed consisting of two wireless technologies: Bluetooth and IEEE802.11b. The network topology of our target measurements is taken from the typical real world wireless networks. A simple example of such a usage scenario is a Bluetooth-based WPAN connected to a large infrastructure network using WLAN network. In this network scenario, a Bluetooth piconet offers wireless connectivity to several Bluetooth devices scattered over small area. This scenario is shown in Figure 2 which represent an integrated WPAN and WLAN network testbed used for conducting the experiments. The testbed consists of 9 Fujitsu Siemens Pentium Celeron

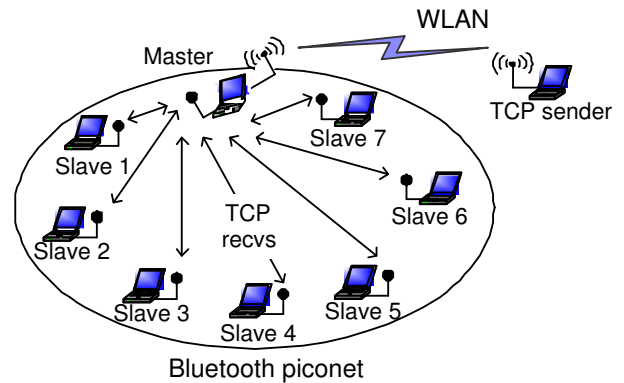


Fig. 2. Network testbed.

laptops running Linux Red Hat operating system. The Linux Bluetooth Bluez¹ stack is used for configuring Bluetooth cards while HostAP driver is used for configuring the WLAN cards. One laptop has WLAN interface for communication in the WLAN network and a Bluetooth interface that acts as a master for the Bluetooth piconet. This machine is considered as the gateway for the piconet. Seven laptops (S1, through S7) are configured as slaves in the piconet. The second laptop in the WLAN network is used for generating TCP traffic to be sent to the slaves in the piconet. The traffic generator *iperf*² has been used for generating TCP traffic and the *tcpdump*³ has been used for logging the TCP data at TCP receivers. The TCP version used in the measurements supports the TCP SACK extension [10].

III. PERFORMANCE EVALUATION

In this section, we discuss the results of a number of TCP experiments conducted over the integrated WLAN and WPAN network. Each run involved a TCP transfer lasting 30 seconds. All experiments were repeated five times, and their average were reported.

A. Effect of the window and packet sizes

The flow control mechanism prevents senders from overloading the capacity of receivers [11] by dictating the rate of the data transmitted by a TCP connection. The most fundamental tuning parameter for TCP flow control is the TCP window size, which controls how much data can be in the network without waiting for the corresponding acknowledgement. If it is too small, the sender window will never fully open up and the connection will be idle at times leading to poor performance. If the receiver buffer is too large, TCP flow control breaks and the sender can overload the receiver, which will cause the TCP window to shut down. The bandwidth delay product is typically used to estimate the "best" window size.

¹<http://www.bluez.org/>

²<http://dast.nlanr.net/Projects/Iperf/>

³<http://www.tcpdump.org/>

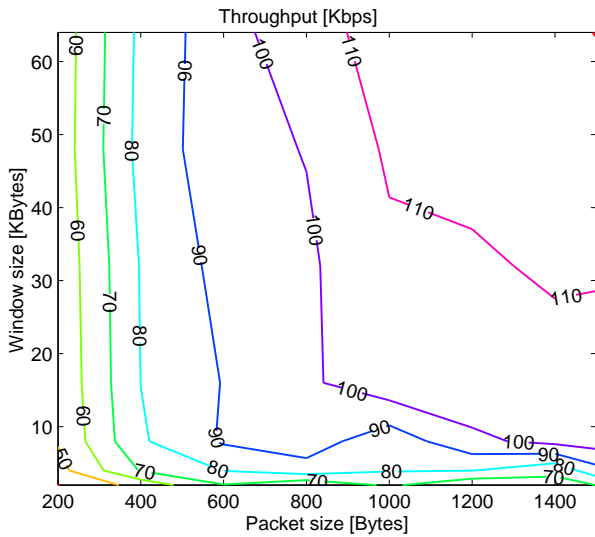


Fig. 3. Measured TCP throughput as a function of window and packet sizes when one slave is receiving data, seven active slaves are present in the piconet.

The relationship between the bandwidth-delay-product (BDP) and the window size is

$$\text{WindowSize} = \text{Bandwidth} * \text{RTT}, \quad (1)$$

where RTT is the path round trip time. Figure 3 shows the TCP throughput as a function of packet and window sizes when one slave is actually receiving data with other six active slaves in the piconet. We observed that when the window sizes are small, TCP performance is bad. As the window size increases, the performance of TCP improves and the best results were achieved with window size 64 Kbytes and packet size equal 1500 bytes. Regarding the dependence of the throughput on packet sizes, Figure 3 indicates an increase in the packet size leads to an increase in the throughput. One reason for the performance degradation with small packet size is the fact that small values of packets limit the amount of traffic that can be sent.

B. Effect of the no of active slaves in the piconet on the throughput

This set of experiments examined the throughput of a single TCP connection in the presence of several active slaves in the piconet. The sender TCP window size was set to 64 Kbytes and the packet size was 1500 bytes, which were found to provide the best TCP performance. The tests were mainly performed to investigate the performance of the Bluetooth polling scheme when only one slave is exchanging data while other slaves in the piconet are active but do not exchange any data. As shown in Figure 4, the bandwidth obtained by the TCP connection depends on the number of active slaves in the piconet. As the number of active slaves increases, TCP throughput decreases. This Bluetooth behavior is attributed to the scheduling mechanism which is based on pure round robin polling. With round robin approach, the master serves each slave according to a fixed cyclic order. To overcome

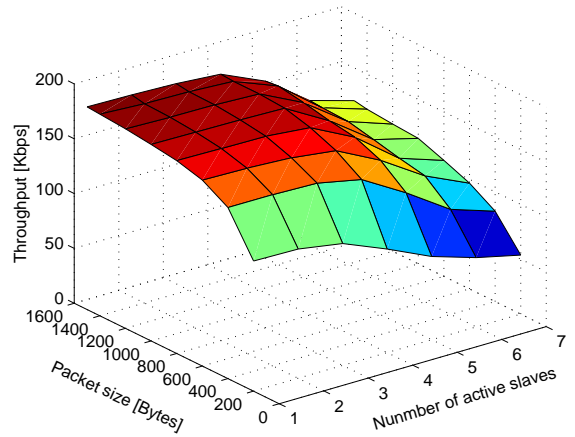


Fig. 4. Effect of the no. of active slaves in the piconet on TCP throughput when only one slave is actually receiving data.

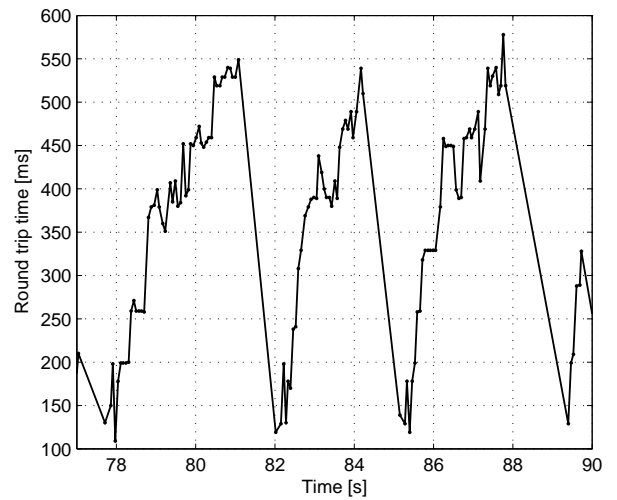


Fig. 5. Round trip time measured when the window size equal 64 Kbytes and the packet size equal 1500 bytes.

this problem, an efficient Bluetooth scheduling algorithm that provide fair radio resource allocation based on traffic load in each master-slave pairs is needed. Recently, the authors of [12] proposed an intra-piconet scheduling in Bluetooth piconets that allocates the time to each slave according to its current traffic, while trying to limit the maximum piconet cycle time.

C. Impact of the round trip time (RTT)

Round trip times play an important role in the operation of TCP protocol. The RTT is defined as the time between when a packet is sent and when its acknowledgement is received back. Lower values of the RTT indicate better performance. Figure 5 presents a sample of RTT per packet which were reported by the *tcptrace*⁴. The measurements indicate that the round trip time varies over time, mainly, because of the queueing at the Bluetooth bottleneck. As the queue build-up

⁴<http://www.tcptrace.org/>

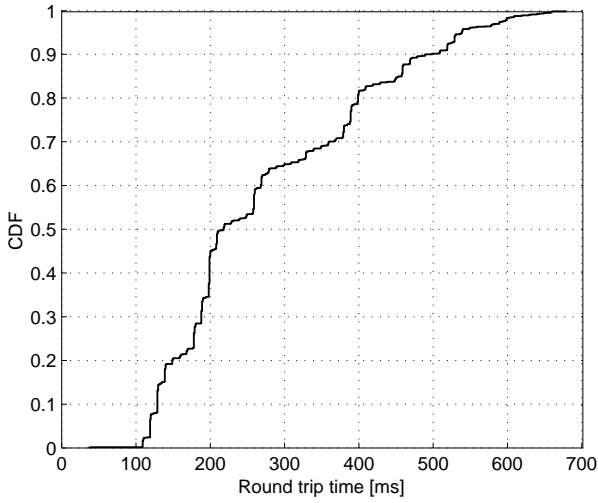


Fig. 6. Round trip time distribution.

continues, the RTT delay become large. Figure 6 shows the cumulative distribution (CDF) of the RTT samples for the whole experiment duration. The following observations can be made:

- 1) The observed RTTs range from 37 *ms* to 680 *ms*. The average is 277.7 *ms* and the standard deviation is 138.6 *ms*.
- 2) We note that we observed very few (less than 1%) RTT samples under 100 *ms*, but 50% of the RTT samples have values smaller than 210 *ms*.

These observations indicate that the range of RTTs experienced by TCP segments is extremely large. The constant the increase in RTT indicates that queues build-up continues until the bottleneck capacity is saturated due to the bandwidth mismatch between WLAN and Bluetooth technologies. Long RTT delay deteriorates the TCP performance by triggering retransmission timeouts (RTO) even though data segments are not lost. This behaviour has a negative effect on the end-to-end throughput and results in a sharp drop to the TCP throughput.

D. TCP throughput and congestion window evolution over time

According to [13], the TCP data sender only retransmits a packet after a retransmit timeout has occurred, or after three duplicate acknowledgements (ACKs) have arrived triggering the Fast Retransmit algorithm. TCP takes retransmission timeout as a loss indication. When the RTO timer timeouts, the TCP sender retransmits the first segment that has not been acknowledged. The RTO is updated dynamically based on two state variables, smoothed round-trip time (SRTT) and round-trip time variation (RTTVAR). According to [14], the RTO can be computed as follows:

$$\begin{aligned}
 SRTT_{i+1} &= \alpha * SRTT_i + (1 - \alpha) * RTT \\
 RTTVAR_{i+1} &= \beta * |SRTT - RTT_i| + \\
 &\quad (1 - \beta) * RTTVAR_i \\
 RTO_{i+1} &= \max(SRTT_{i+1} + 4 * RTTVAR_{i+1}, 1),
 \end{aligned} \tag{2}$$

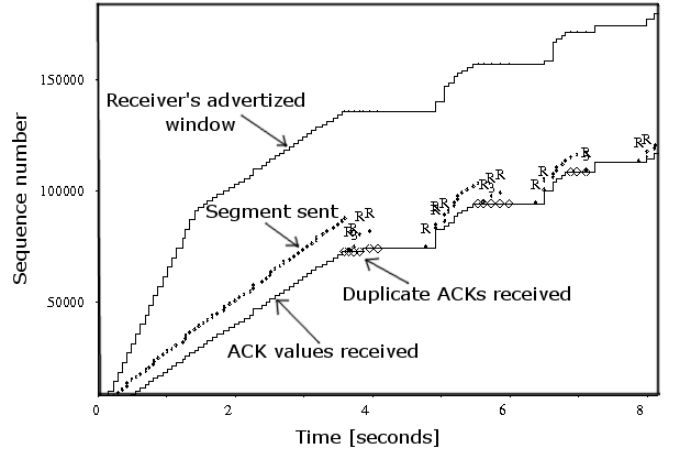


Fig. 7. A sample trace of time sequence number graph showing the activity of TCP connection.

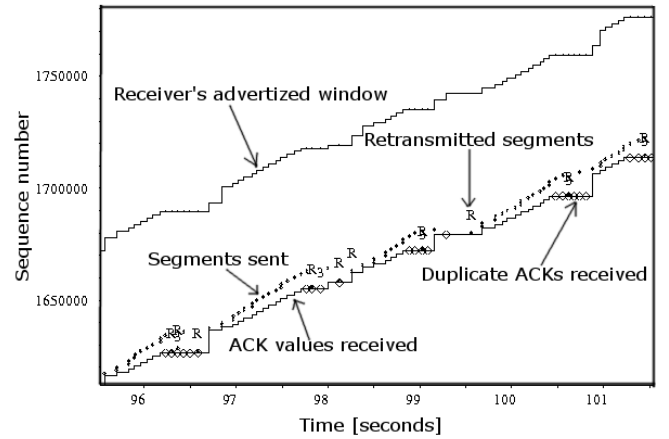


Fig. 8. Time sequence number graph trace.

where RTT is the measured round trip time, α and β are constants between 0 and 1 that controls how rapidly SRTT and RTTVAR adapt to change.

Figure 7 shows the trace of a TCP connection obtained from a scenario where one active slave receiving data is present in the piconet. The packet size is 1500 bytes and the TCP window size is 64 Kbytes. The figure indicates that the first duplicate ACKs arrives at about 3.6 s. The segment sent at 3.59 s is the last segment sent before congestion is detected after the sender receives the third duplicate ACK. After the third duplicate ACKs, TCP sender retransmits the missing packet without waiting for RTO. The figure indicates that TCP sending rate progresses until the duplicate ACK arrives. It also shows that TCP sending window never reaches its maximum size due bandwidth limitation of the Bluetooth bottleneck. A TCP Bluetooth booster has been suggested [15] to overcome the potential problems (such as high RTTs) caused by Bluetooth. Figure 9 shows the sender

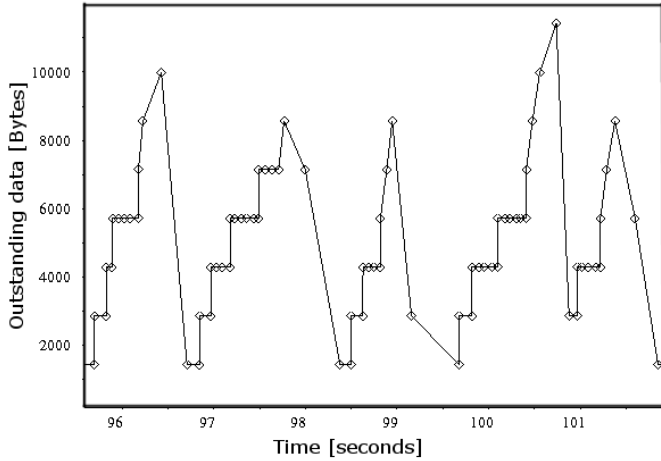


Fig. 9. TCP congestion window evolution.

TCP congestion window evolution, corresponding to the time sequence-number trace in Figure 8 obtained from the same scenario. The outstanding unacknowledged data is used to estimate the congestion window at the sender. The figure illustrates better what actually happens to the congestion window size. The vertical axis represents the outstanding data (congestion window size) in bytes and the horizontal axis represents time at which the data was sent. It can be observed that the evolution of the congestion window is affected by the large round trip delays imposed by Bluetooth, which has a negative impact as it causes the congestion window to grow slowly and hence degrade the end-to-end throughput performance. It also shows that the TCP connection seems to be suffering from a lot of lost and retransmitted data as indicated by the many retransmission attempts.

Because of the packets being queued for transmission at the Bluetooth bottleneck, which causes RTT to increase as more packets are sent, the *cwnd* keeps increasing until duplicate ACKs are received from the TCP destination. At this time the *cwnd* drops to one maximum segment size (MSS), which is 1460 bytes in this experiment. Frequent reception of duplicate ACKs causes the congestion window to drop frequently resulting in an extremely low throughput. The selective acknowledgments (SACK) [10] option has been proposed to recover from multiple segment losses by avoiding waiting for a RTO every time a segment is lost.

E. Multiple TCP Connections

1) Effect of multiple TCP streams on TCP throughput:

These tests were carried out to examine the performance of TCP when there are multiple TCP streams sharing the same bottleneck. It represents the case scenario where several slaves in a piconet are downloading data simultaneously. We did experiments to measure the impact of the number of TCP connections on throughput received by each slave and to examine how the bandwidth is distributed among them.

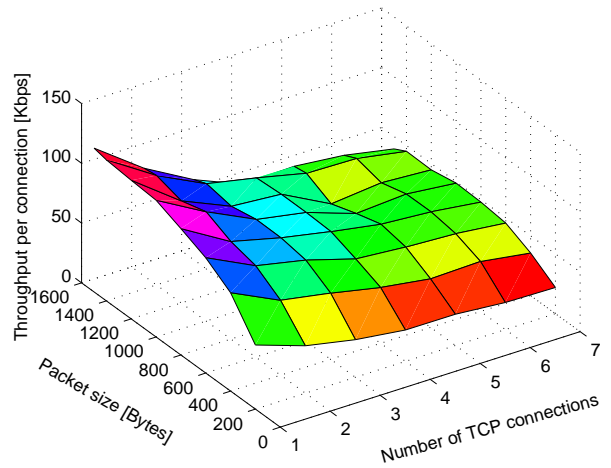


Fig. 10. Effect of number of TCP connections on the throughput received by slaves, seven active slaves are present in the piconet.

TCP sender initiates TCP connections to slaves at the same time with varied packet sizes. Figure 10 shows the average throughput received by slaves as a function of packet size and the number of TCP streams. We see that as number of TCP connections increases, the average TCP throughput received by each slave degrades. However, the aggregate throughput increases with number of TCP flows.

2) *Bandwidth distribution*: Fairness is an important criterion in a system where bandwidth resource is shared between multiple TCP streams. Multiple TCP streams sharing the same link should each receive a fair share of bandwidth. Several measures of fairness have been proposed in literature. We will use the fairness metric proposed by [16] to evaluate the bandwidth distribution among multiple TCP streams. The author proposes a function to measure the fairness of a system. Assuming a system which allocates resources to n contending users such that the i^{th} user receives allocation x_i , then according to [16], the proposed index for the system is

$$f(x_1, x_2, \dots, x_n) = \frac{\left(\sum_{i=1}^n x_i\right)^2}{n \sum_{i=1}^n x_i^2}. \quad (3)$$

The fairness index always lies between 0 and 1. These bounds help understanding the fairness index. For example, a distribution algorithm with a fairness of 0.10 means that it is unfair to 90% of the users. Bandwidth sharing among multiple TCP connections studied in [17] demonstrated that it is not an easy task to realize fair bandwidth allocation among multiple TCP connections. This is due to the distributed control nature of the TCP flow control, such that each TCP connection regulates its sending packet without exchanging information with other TCP connections. To give a more accurate idea of bandwidth distribution, for packet size equal 1500 bytes, with 2 TCP connections, slaves have received

a comparable bandwidth with fairness index of 0.997. As the number of TCP connections increases, the fairness index decreases to 0.875. The packet sizes seem not to have much effect on the fairness index. This result suggests that the TCP congestion control algorithm seems to allocate roughly fair bandwidth share to competing connections when the number of TCP stream is small.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we evaluated the performance of TCP over an integrated WPAN (based on Bluetooth in this example) and WLAN network. Our experimental results confirm that the time slots allocated to Bluetooth slaves in a piconet depends on the number of active slaves in a piconet regardless of their data exchange activities. This is a direct consequence of the round robin scheduling approach. Measurements performed with different window sizes, have revealed that the variation in round trip times increases with the increase in the window size. We have seen that long round trip time delays have a negative impact on the TCP end-to-end throughput performance. Our experimental results suggest that the TCP congestion control algorithm seems to allocate fair bandwidth share to competing connections when the number of TCP streams is small. For future work, it would be interesting to focus on the effects of long RTTs and RTT variations on the behaviour of TCP and to investigate methods for mitigating these effect. Investigations on efficient and fair Bluetooth polling approaches are also necessary for successful Bluetooth-based WPANs.

V. ACKNOWLEDGEMENTS

This work was partially supported by the European Commission PULSERS Project (IST 506897).

REFERENCES

- [1] C. Cordeiro, S. Abhyankar, R. Toshiwal, and D. Agrawal, "BlueStar: Enabling efficient integration between Bluetooth WPANs and IEEE 802.11 WLANs," *Mobile Networks and Applications*, vol. 9, no. 4, pp. 409–422, Aug. 2004.
- [2] D. Famolari and P. Agrawal, "Architecture and performance of an embedded IP Bluetooth personal area network," in *Proceedings of the IEEE International Conference on Personal Wireless Communications*, Hyderabad, India, Dec. 2000, pp. 75–79.
- [3] J. F. M. Gerrits and J. R. F. J. R. Long, "UWB considerations for "my personal global adaptive network" (MAGNET) systems," in *Proceedings of the 30th European Solid-State Circuits Conference (ESSCIRC)*, Leuven, Belgium, Sept. 2004, pp. 45–56.
- [4] Bluetooth Special Interest Group (SIG), "Bluetooth Core Version 1.2," 2003. [Online]. Available: <http://www.bluetooth.com/spec>
- [5] S. Roy, J. R. Foerster, V. S. Somayazulu, and D. G. Leeper, "Ultrawideband radio design: the promise of high-speed, short-range wireless connectivity," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 295–311, Feb. 2004.
- [6] N. M. Chaskar, T. V. Lakshman, and U. Madhow, "TCP over wireless with link level error control: analysis and design methodology," *IEEE Transactions on Networking*, vol. 7, no. 5, pp. 605–615, Oct. 1999.
- [7] I. M. Suliman, T. Hautala, T. Saarinen, C. M. Wayne, and T. Bräysy, "Performance measurements of TCP on a heterogeneous wireless multi-hop network," in *International Workshop on Wireless Ad-hoc Networks (IWWAN04)*, Oulu, Finland, May 2004.

- [8] N. Golmie and O. Rebala, "Techniques to improve the performance of TCP in a mixed Bluetooth and WLAN environment," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, Anchorage, AK, USA, May 2003, pp. 1181–1185.
- [9] K. Matheus and S. Zurbes, "Co-existence of Bluetooth and IEEE 802.11b WLANs: results from a radio network testbed," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, sep 2002, pp. 151–155.
- [10] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options," RFC 2018, Oct. 1996.
- [11] L. L. Peterson and B. S. Davie, *Computer Networks - A System Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2000.
- [12] V. Mistic, E. Ko, and J. Mistic, "Load and QoS adaptive cycle-limited scheduling scheme for Bluetooth piconets," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA, Jan. 2004, pp. 294–301.
- [13] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," RFC 2581, Apr. 1999.
- [14] V. Paxson and M. Allman, "Computing TCP's retransmission timer," RFC 2988, Nov. 2000.
- [15] D. Melpignano and D. Siorpaes, "Bluetooth TCP booster," in *Proceedings of the 53rd IEEE Vehicular Technology Conference*, vol. 3, May 2001, pp. 2137–2141.
- [16] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," DEC Technical Report DEC-TR-301, Sept 1984.
- [17] K. Takagaki, H. Ohsaki, and M. Murata, "Analysis of a window-based flow control mechanism based on TCP Vegas in heterogeneous network environment," in *Proceedings of IEEE ICC'01*, Helsinki, Finland, June 2001.