Should We Use the Default Protocol Settings for Networks of Constrained Devices?

August Betzler, Carles Gomez, Ilker Demirkol, Josep Paradells

Should We Use the Default Protocol Settings for Networks of Constrained Devices?

August Betzler, Carles Gomez, Ilker Demirkol, and Josep Paradells
Department of Telematic Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain
i2CAT Foundation, Barcelona, Spain
{august.betzler, carlesgo, ilker.demirkol, teljpa}@entel.upc.edu

***Abstract*—Constrained devices will play a crucial role in the Internet of Things. These devices use communication technologies which often support the formation of wireless multihop topologies. Therefore, end-to-end data transfer involves the collaboration of multiple protocol layers. However, the settings and mechanisms used in a specific layer affect performance of the others. In this paper, we assess a set of crucial network design criteria for potential MAC layer, network layer and transport layer protocols for networks of constrained devices. We evaluate the default and alternative settings and mechanisms for these protocols on a 60-node testbed. The experiments show how performance can be improved significantly by using different settings and mechanisms from the default ones.**

I. Introduction

In this paper, we analyze the effects of several network design criteria on the performance of wireless networks of constrained devices. When talking about constrained devices, we refer to wireless sensor nodes that have a limited amount of resources, inter alia memory capacity, processing power and power supply. In current research topics such as Internet of Things, constrained devices play a central role. Networks of such wireless devices supporting one-to-one data transfers may require the formation of multihop topologies. To achieve this, the collaboration of multiple protocol layers is necessary. However, in this case, the settings and mechanisms used in a specific layer affect performance of the others. A specific research is required to evaluate these phenomena. Ideally, such research should be carried out in a real environment, e.g., by using a testbed. A testbed evaluation does not only supersede the need to develop a complex model for simulations of physical indoor channels, it also enables the accurate observation of network performances. In this work, we use a testbed of constrained devices (sensors) to analyze the effect of different network design criteria from different stack layers on the network performance. To do that, we evaluate default settings defined or implemented for these design criteria in potential protocols of these constrained devices along with the certain alternative settings. The results of the experiments carried out during this investigation shall be a useful guide that presents the possible performance achievements over the default settings of these criteria.

The scope of the investigation extends over three layers of the protocol architecture: The Medium Access Control (MAC) layer, the Network (NWK) layer and the Transport layer. Protocols such as IEEE 802.15.4, AODV and TCP define several mechanisms and parameters for these layers, respectively. Default configurations, however, may not deliver the best performance under specific network conditions. Therefore, the default settings and certain alternative mechanisms are investigated in this work, using network performance metrics such as goodput and successful data transfer rate. The list of the investigated network parameters and their default configurations are given in Table I.

At the MAC layer, the impact of changing the maximum number of retries and altering the backoff behavior is evaluated. By default, the operating system of the nodes used in our evaluation, i.e., TinyOS, employs a backoff mechanism different than the well-known Binary Exponential Backoff (BEB). Moreover, the maximum number of retries is a crucial parameter that can affect overall network performance. At the NWK layer, two different routing metrics are evaluated: the classic Hop Count metric, which is the default metric of the AODV and a Link Quality Indicator (LQI) based metric, called PATH-DR [1]. At the transport layer, the effects of different approaches for Retransmission Timeout (RTO) calculation algorithms and end-to-end ACK-mechanisms are analyzed. Also, it is demonstrated that for constrained networks such as the one used in this work, a Congestion Window Limit (CWL) must be used and tuned to an appropriate value. The evaluation of the aforementioned criteria will be carried out on a testbed, as described in the next section. Combining and evaluating all possible combinations of parameters and mechanisms inside and across the layers would exceed the limits of this paper. Hence, during evaluations, configurations of individual design criteria are singled out and the settings of other criteria are fixed. Unless otherwise noted, the default configurations of each layer, as described in Table I, are used.

The rest of the paper is organized as follows. Section II introduces the testbed and defines the test scenarios. The evaluation results are given in Section III, pointing out the benefits and drawbacks of using certain settings and analyzing under which conditions they allow a performance improvement. Finally, Section IV concludes the paper.

II. Testbed Design

In this section, the features of the testbed used are presented. Additionally, the configuration of the testbed nodes is explained and the two test scenarios for the experiments are defined.

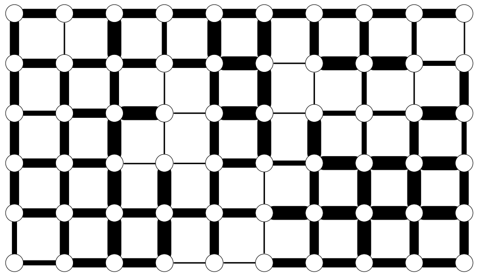


Fig. 1: A snapshot of the connectivity among neighboring grid points in the testbed. Line thickness is proportional to the LQI value of the link.

A. Testbed setup

For the experiments, a testbed of 60 TelosB nodes [2] is used. The testbed is located indoors in an office environment. The nodes are attached on the bottom of a rectangular wooden grid that hangs 0.5 meters below the ceiling. The grid has a length of 8.1 meters and a width of 5.4 meters. The nodes in the grid are separated equidistantly from each other, building a mesh with 6 rows and 10 columns (6x10). Fig. 1 represents this grid along with the link qualities observed at a specific time instant for each neighboring grid points. On the upper part of the wooden structure, a USB cable-tree connects all nodes to a central computer. The nodes are powered by this connection, at the same time allowing it to program them and to communicate with them over the serial interface. Highly complex signal propagation is experienced in the cluttered environment of the testbed, as it is typical for indoor networks. Adding to the complexity of the network conditions, during daytime, active 802.11 networks have been detected in the same environment. In contrary during nighttime there could not be observed any traffic. To avoid strong interference from 802.11 networks, the nodes use channel 26 to communicate with each other, since none of the 802.11 devices operate in the corresponding frequency range. To assure similar conditions for all experiments, they are carried out during nighttime, where the general activity inside the building is the lowest.

The TelosB nodes are programmed with TinyOS and equipped with a TI MSP430 microcontroller, 48 kByte Flash, 10 kByte RAM and a TI CC2420 RF transceiver. The radio transceiver operates in the 2.4 GHz frequency band with a maximum data transmission rate of 250 kbit/s. The CC2420 radio allows adjusting the transmission power in 30 steps, the lowest value corresponding to an output power of less than -25 dBm and the highest to 0 dBm. For all tests, the transmission power of the nodes is reduced to the minimum to achieve the longest route lengths possible.

B. Protocols Implemented

TinyOS provides a default MAC layer for TelosB nodes, to which several modifications are investigated in the experiments. At the network layer, the lightweight nst-AODV [3] is used as the routing protocol. Protocols based on AODV are used in the area of constrained networks, e.g., in ZigBee [4] and the one-to-one mechanism developed for RPL [5], to establish arbitrary one-to-one connections. nst-AODV's working principles and advantages over the standard AODV implementation are presented in [3]. To reduce complexity, local route repair is disabled. Above the network layer, a basic version of the Transmission Control Protocol (TCP) based on RFC 793 with the slow start algorithm is used for a reliable transmission of data and flow control. All core mechanisms are implemented, with certain minor variations, adapting them to the specific test conditions. Accordingly, the granularity of 1 ms is used, since TinyOS provides millisecond timers. Minimum and initial RTO values are set to 500 ms and 1 s, respectively. The default values defined by the investigated protocols and the operating system are shown in Table I for the network design criteria evaluated.

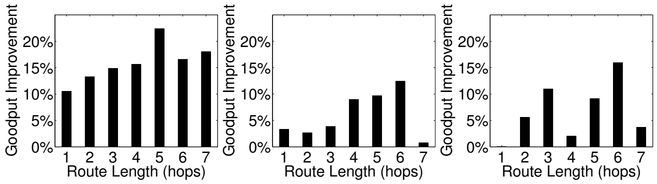
TABLE I: Default Settings of the Investigated Design Criteria

Layer	Criterion	Default Setting
MAC	Backoff method	TinyOS backoff
	Max retries	3
Routing	Routing metric	Min. hop count
Transport	CWL	8
	RTT calculation	Karn method
	ACK method	Cumulative ACK

C. Test Scenarios Defined

To test the alternative configurations, a data transfer application is defined, which consists in transmitting a fixed amount of 44 kByte from a single data source to a destination node. This amount of data may represent different types of content, including logged events or a binary node image. The transfer of this data is considered successful, if all data frames are delivered successfully at the destination node. This setup also represents a real world application of over the air programming (also known as OTA) of the nodes. During the transmission process, detailed information such as the packet Round Trip Time (RTT) measured by the transport layer and the amount of failed MAC layer transmissions is logged.

Two different route setups are defined for the evaluations. In the first setup, static routes are used to make evaluations, where these static routes are built by running the nst-AODV a priori and using these routes throughout the test. Experiments with static routes are repeated 10 times, corresponding to the transmission of approximately 4700 frames. In the second route setup, nst-AODV is fully enabled, allowing route changes during the data transfer and the participation of all nodes of the grid in the transmission. To extract an average behavior, 31 random pairs are chosen as endpoints of the transfers. Although the dynamic route setup is more adaptive and realistic, the static route evaluations are also crucial. The reason is that they enable a more comparable setup, since the dynamic routes might change the route lengths and the values of other metrics significantly. The experiments with dynamic routing are repeated 5 times, corresponding to the transmission of approximately 2350 frames for each pair of nodes. In both setups, all experiments are repeated with different CWLs for the Transport layer, since observations show that this parame-



(a) CWL=1 (b) CWL=4 (c) CWL=8

Fig. 2: The percent improvement in goodput when using exponential backoffs compared to the default TinyOS backoff for different CWL values.

ter has an essential influence on the outcome of experiments. The maximum CWL is set to be 8 and 5 for static and dynamic routing scenarios, respectively. For all tests, the nodes' buffers were chosen big enough to avoid dropped packets due to buffer overflows. Using smaller buffers and multiple flows is left out for future work. The investigated performance metrics are the goodput, calculated as the useful amount of data over time received by the destination node, the RTT between the two endpoints of the transmission and the percentage of successful transfers from all the transfers of a test run. Only successful transfers are used to evaluate measured values as goodput and RTT.

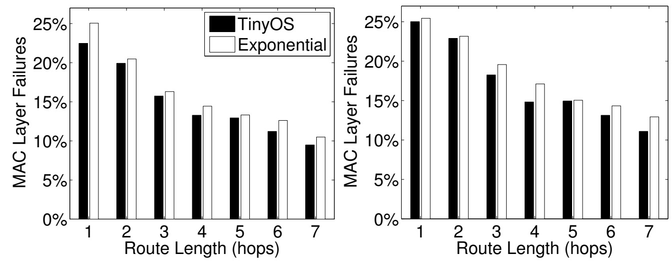
III. Evaluation Results

The first part of the evaluation focuses on modifications at the MAC layer. Subsequently, the findings for the routing layer are presented and ultimately design criteria for the transport layer are analyzed.

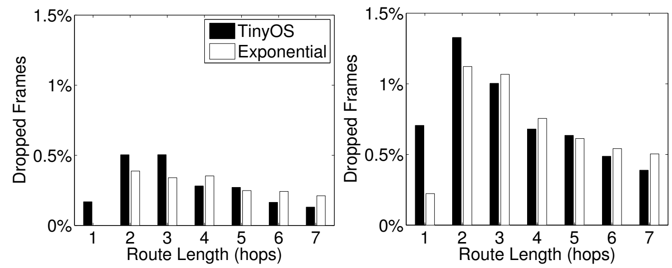
A. MAC layer

1) MAC layer backoff: An important design criterion at the MAC layer is the type of backoff mechanism used to delay a transmission if the channel is found to be busy. The TinyOS uses its own backoff mechanism as described in [6], whereas it does not employ the well-known Binary Exponential Backoff (BEB) mechanism used in IEEE 802.15.4 [7] or in IEEE 802.11. IEEE 802.15.4's default backoff starts with a small interval of [0,140] symbol periods (SP) and if the channel is busy, it keeps on doubling the upper bound of this interval up to a predefined threshold, until the channel is found to be free or all MAC layer retries are spent. TinyOS backoff starts with a large interval of [20,640] SP and falls back to a smaller interval of [20,140] SP, and uses this small interval for the subsequent retries. It lies in the nature of BEB to increase the chances for the last winner of a channel contention to capture the channel again. On the other hand, the TinyOS backoff is meant to deliver a higher degree of fairness, since the probability for a winning node to capture the channel again is lower.

The performances of both backoff mechanisms are evaluated in the static routing scenario, and are depicted in Figs. 2 and 3. While the goodput increases clearly with a smaller CWL, the benefits are less predictable for larger CWL. With a window size of 1, the benefit of using the IEEE 802.15.4 backoff increases with the number of hops. Each node on the route uses the shorter initial backoff, decreasing the average transmission delay at each hop, which results in shorter RTTs and therefore increases the goodput directly. As the CWL, and hence the contention increases, the exponential backoff leads to slightly more collisions and the percentage of dropped frames increases as shown in Fig. 3. This is a drawback, since the loss of packets decreases the performance of the transport layer. However, the benefit of shorter hop delays outweighs the performance loss induced by dropped packets and the higher amount of MAC layer retries.



(a) CWL=4 (b) CWL=8

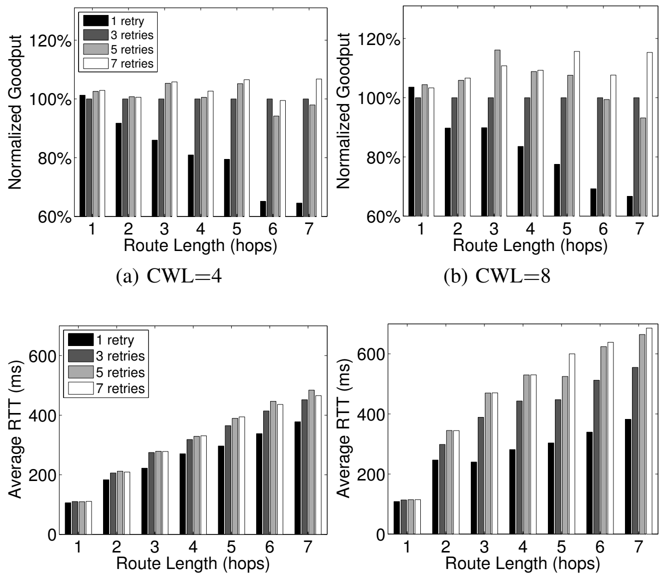


(c) CWL=4 (d) CWL=8

Fig. 3: Effect of backoff mechanism choice on percentage of failed MAC layer transmissions and overall frame drop rates for different route lengths and CWL values. The frame drops occur after the maximum number of unsuccessful MAC layer retries is exceeded.

2) MAC layer retries: The MAC layer provides one-hop reliability, where a common parameter is the maximum number of retries for a single packet transmission. TinyOS does not define a default value for this parameter. Hence, we evaluated IEEE 802.15.4's default parameter value of 3 retries [7] along with the values of 1, 5, and 7 in static scenarios.

Fig. 4 depicts the goodput and RTT results for static routes and for CWL values of 4 and 8. The evaluations showed that for a CWL of 1, the performance differences between each configuration are negligible, and hence are not shown. This is due to the good average quality of the chosen route links and due to the fact that there is a low contention in case of CWL of 1. As seen in Fig. 4, the goodput increases with the number of allowed retries, even though the average RTT measured by the transport layer increases due to the delay that additional retries cause at the MAC layer. Moreover, additional retries ensure the one-hop transmission of packets, reaching a successful transfer ratio of 99% in the case of 7 retries, as seen in Fig. 5.



(a) CWL=4 (b) CWL=8

(c) CWL=4 (d) CWL=8

Fig. 4: Effect of maximum MAC layer retry values on normalized goodput (normalized by the goodput of 3 retries), and on the average RTT durations for different route lengths and CWL values.

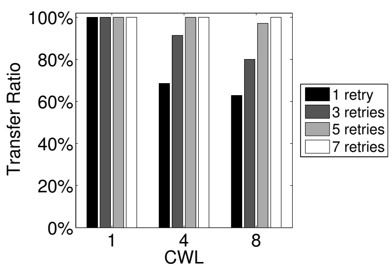


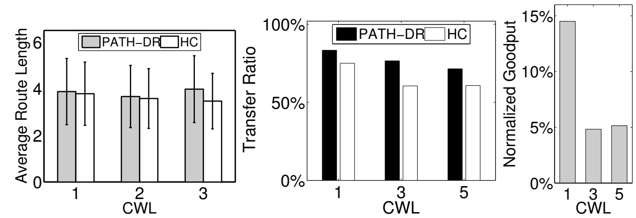
Fig. 5: The average successful transfer ratio over all route lengths for different maximum MAC layer retry and CWL values.

The great RTT increase from 3 retries to 5 (and 7) is not only caused by the additional one-hop delays, but more importantly by the transport layer behavior. With fewer retries, the number of dropped packets increases. A packet loss will most likely provoke a transport layer RTO which resets the window size to 1. That is a reason for the smaller average RTT values observed for smaller CWL values.

B. NWK Layer

At the routing layer, the performances of using PATH-DR [1] and hop count (HC) metrics in dynamic routing scenarios are compared. The former estimates the packet delivery ratio of a route based on the LQIs measured on each link traversed. The route with the highest PATH-DR is chosen for the data transmission. When using the HC metric, the shortest route is chosen, without taking the link qualities into consideration.

Dynamic routing tests are carried out with different CWL values for each routing metric. The average route length and its



(a) (b) (c)

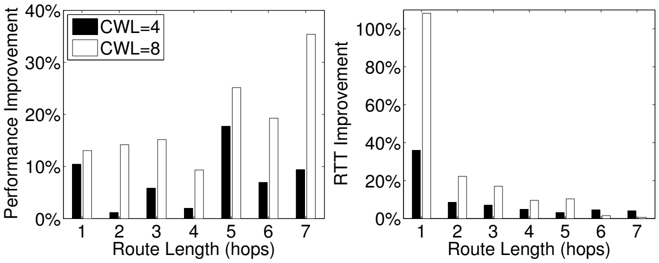
Fig. 6: Comparison of using PATH-DR versus using HC as the routing metric for (a) the average route lengths, (b) successful file transfer ratio, and (c) the percent improvement of the goodput of the PATH-DR metric compared to the HC metric.

standard deviation using both metrics are depicted in Fig. 6a. Routes chosen with a HC metric are shorter and their length has a smaller standard deviation than the routes chosen by the PATH-DR metric. In terms of goodput, the PATH-DR metric delivers better results for all CWLs evaluated (Fig. 6c). More importantly, the successful transfer rate is higher, as shown in Fig. 6b. The reason for both a higher goodput and a higher percentage of successful transfers lies in the fact that the PATH-DR metric chooses routes with high LQIs on the links, resulting in a low ratio of erroneous packets. Consequently, for subsequent measurements in dynamic routing scenarios, the PATH-DR metric is used to select the routes.

C. Transport layer

Here, we focus on the design criteria of the transport layer. TCP offers a reliable transmission of data packets, variable window sizes and a dynamic buffer allocation to support multiple data flows. Additionally, in our implementation it has access to cross-layer information provided by the underlying nst-AODV, such as the route length and the RTT measured during the session initialization. This information may be used for the calculation of Retransmission Timeouts or the calculation of an upper bound for the CWL. Before focusing on these aspects of the transport layer, an evaluation of two end-to-end reliability mechanisms is done.

1) Acknowledgement methods: In this subsection, the performances of two acknowledgement mechanisms are compared. These are i) Cumulative ACKs, where a single ACK can confirm the reception of several packets at once. The ACK packet is updated before its transmission, if there are new data packet arrivals; and ii) Positive ACKs, where each received packet is confirmed by a separate ACK. Since the link quality based PATH-DR metric is used to choose the routes in these experiments, the resulting PDR is expected to be high, and hence, the cumulative ACKs are expected to yield a better performance. Experiments on static routes are carried out to confirm this assumption. Fig. 7 shows how the average goodput and RTT for cumulative ACKs compares to positive ACKs for different CWLs. The results for CWL of 1 are similar for positive and cumulative ACK methods, since in this case they result in the same behavior, and hence are skipped. As seen in the figure, cumulative ACKs always



(a) Goodput comparison (b) RTT comparison

Fig. 7: Performance improvement of cumulative ACKs on positive ACKs in terms of (a) average goodput, and (b) average RTT.

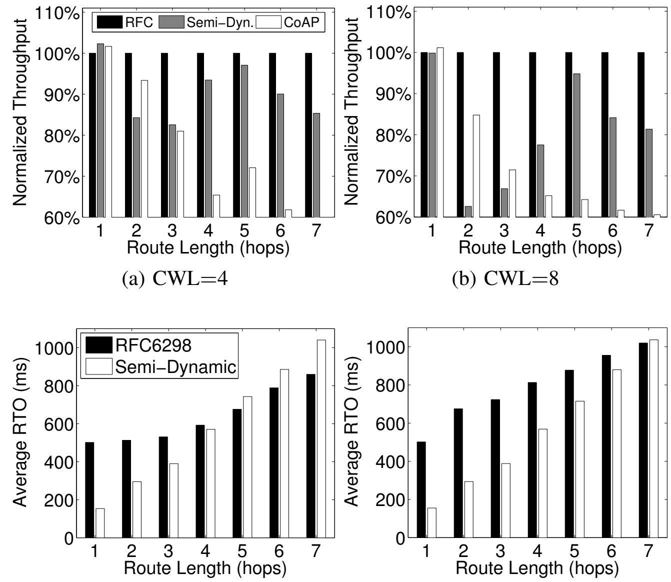
perform similar or better than the positive ACKs for both goodput and RTT. Moreover, higher CWL value results in a better improvement in these values. We observed a linear increase in the improvement of the goodput values with an increase in CWL value. The main cause for such increase is that less ACKs are sent in order to confirm the reception of data in the case of cumulative ACKs. With positive ACKs, the number of packets on both the upstream and the downstream contending for the channel increases with the ACK packets. This leads to an increment of the one way delays in both directions of the end to end transmission, resulting in a higher RTT.

2) Retransmission Timeout (RTO) calculation: The RTO for a packet determines the duration after which a packet transmission is considered as failed, and the packet needs a retransmission. There are two basic RTO events affecting the performance of the Transport layer. The loss of data or ACK packets causes the RTO-timer to fire. If the RTO is too large compared to the actual RTT, the packet loss is noticed with a large delay. This effect can decrease the goodput considerably. On the other hand, if the RTO is too short, packets are retransmitted even though a valid ACK is about to arrive. This also can decrease the performance noticeably, depending on how the protocol reacts to RTOs. Since, by default, TCP assumes congestion, the common reaction of the transport layer is to reduce its window size to 1.

The RTO may be calculated in many different ways, ranging from using static values, to advanced formulas that include several variables. In the following, the effect of three different algorithms on the Transport layer performance is compared:

i. A static RTO, where the RTO is chosen from an interval, as it is proposed in CoAP [8]. The initial interval lies between 2 and 3 seconds.
ii. A semi-dynamic RTO, where the RTO is calculated once from the RTT that is measured during the session establishment according to RTO = K x RTT, where K is set to 4. During the session, the value of RTO does not change.
iii. A dynamic RTO calculation, according to RFC6298 [9].

Fig. 8 compares the performance of the CoAP and semi-dynamic algorithms relative to RFC6298's algorithm for static routes. With a CWL of 1, neither packet losses, nor spurious



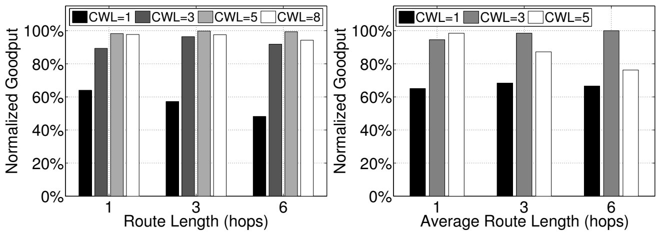
(a) CWL=4 (b) CWL=8

(c) CWL=4 (d) CWL=8

Fig. 8: Effect of RTO algorithm choice in terms of normalized goodput (normalized by the goodput of RFC6298's algorithm) and RTOs for different CWLs.

retransmissions can be observed, since the nodes observe very low contention for the channel. Consequently the performance is identical for all algorithms, and are not displayed. By increasing CWL, however, differences in the performance of the three algorithms can be observed that depend highly on the number of hops. Figs. 8a and 8b show that the fully-dynamic RFC6298 algorithm outperforms the other two algorithms in routes with multiple hops. The CoAP algorithm does not adapt in any way to the actual average RTT of a transmission. The semi-dynamic RTO is only calculated once for the whole transmission, thus the overall performance of this algorithm depends strongly on the RTT value initially measured. Experiments show that, on average, the dynamic algorithm performs better than the semi-dynamic algorithm for different route lengths. It can also be observed that the differences get smaller, when the average RTOs of both algorithms are close to each other. In one-hop scenarios, no packet losses could be observed, therefore the CoAP and semi-dynamic algorithms perform quite well.

3) Congestion Window Management: An important issue that needs to be taken into account when designing a transport layer protocol is the congestion window management. This is especially important for constrained devices, since internal buffer space and available radio bandwidth are strongly limited. Bandwidth calculation algorithms are proposed for IEEE 802.11 networks, among which some are integrated in TCP [10]. These algorithms try to estimate the available bandwidth to adjust their congestion window size correspondingly. However, it is not clear if the processing and memory costs of these algorithms justify an implementation for constrained devices. Basic TCP uses channel probing to find an appropriate



(a) Static scenarios

(b) Dynamic scenarios

Fig. 9: The comparison of normalized goodput (normalized by the goodput of the recommended CWL) for different CWLs in (a) static and (b) dynamic scenarios.

congestion window size. The window size is increased, until the traffic created overshoots the available bandwidth and an RTO happens, signaling congestion. In the following, a simple bandwidth estimation algorithm that tries to prevent the congestion window from overshooting is analyzed by defining an upper limit. The main two motives to implement an upper limit for the congestion window are the following: First, a large CWL results in a higher buffer usage at intermediate nodes, increasing the probability of dropped packets. Second, a higher degree of contention is to be expected, since more data-packets and ACKs travel on the forward and backward path of a route. On the other side, using a small, static CWL may under-utilize the available bandwidth. Experiments are done, to see the effect of the CWL value on the performance achieved on static and dynamic routes.

We define the CWL value after which the goodput does not increase more than 1% by increasing the CWL as the *recommended CWL*. The goodput results for a set of CWL values are compared to the recommended CWLs in Fig. 9 for given (average) route lengths. The results for static routes show that independent of the route length, after reaching the recommended CWL, the goodput flattens out. Through Fig. 9a, it can be observed that the CWL at which the curves flatten out varies with the number of hops. While having a higher CWL value than the recommended CWL does not change the goodput considerably for static scenarios, it causes a higher degree of congestion along the route. This reflects in a greater amount of MAC retries and dropped packets observed in the experiments. These effects would especially be important for multiple flows scenarios.

The evaluation of the dynamic scenarios shows that a more conservative CWL policy leads to better results. Long routes require the source node to reduce the window size considerably. Fig. 9b shows that the goodput does no longer flatten out with increasing CWLs. Instead, a larger window size than the recommended CWL may cause the goodput to drop for routes with multiple hops. The main reason for this behavior is the mechanism with which nst-AODV reacts to failed MAC layer transmissions. A MAC layer transmission failure is interpreted as a link break, causing a route repair. This stops the transmission temporarily and requires the exchange of control messages throughout the network to find a new route between the two endpoints. Since the probability for link-breaks gets higher with the increase in CWL and with it the degree of contention increases, the amount of route repairs also increases. As the length of the route is known by the nst-AODV, a potential performance improvement can be achieved in existing congestion window based transport layer protocols, by applying a cross-layer interaction between the transport and the routing layers to adjust the CWL.

IV. Conclusion

In this paper, the effects of different network design criteria on the performance of networks of constrained devices are investigated through testbed evaluations. The investigated network criteria are chosen from potential protocols to be used for such constrained networks. The evaluations of several single layer mechanisms show that default settings may not deliver the best performance for the protocols and scenarios considered in this work, and that generally there is room to improve the settings. The effects of changing one of these settings can have a significant influence on the network metrics, such as goodput or successful transfer rate. The results also show that due to synergetic effects between the layers, changing a parameter at one layer may have benefits or drawbacks for other layers. The findings of this paper provide valuable insights for networks of similar real world environments. As future work, multiple flow scenarios and different use cases will be added to the repertoire of experiments, as well as an in-depth analysis of potential cross-layer improvements.

Acknowledgment

This research has been funded through Spanish Government CICYT-TEC2009-11453.

References

[1] C. Gomez, A. Boix, and J. Paradells, “Impact of LQI-based routing metrics on the performance of a one-to-one routing protocol for IEEE 802.15.4 multihop networks,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, pp. 6:1–6:20, February 2010.

[2] Crossbow, “TelosB mote platform.” [Online]. Available: http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=252

[3] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, “Adapting AODV for IEEE 802.15.4 mesh sensor networks: Theoretical discussion and performance evaluation in a real environment,” in *Proc. of WOWMOM’06*, Washington, DC, USA, 2006, pp. 159–170.

[4] F. Cuomo, S. Della Luna, U. Monaco, and F. Melodia, “Routing in ZigBee: Benefits from exploiting the IEEE 802.15.4 association tree,” in *Proc. of ICC ’07*, june 2007, pp. 3271 –3276.

[5] “Reactive discovery of point-to-point routes in low power and lossy networks,” Jan. 2012. [Online]. Available: http://tools.ietf.org/html/draft-ietf-roll-p2p-rpl-07

[6] D. Moss, J. Hui, P. Levis, and J. Il Choi, “CC2420 radio stack,” Jun 2007. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep126.html

[7] *IEEE 802.15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Computer Society Std., September 2006.

[8] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, “Constrained application protocol (CoAP),” CoRE Working Group, Nov. 2011. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-core-coap/

[9] IETF, “RFC 6298: Computing TCP’s retransmission timer,” June 2011. [Online]. Available: http://tools.ietf.org/html/rfc6298

[10] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCP westwood: Bandwidth estimation for enhanced transport over wireless links,” in *Proc. of Mobicom’01*, 2001, pp. 287–297.