A Comparative Study of Congestion Control Algorithms in IPv6 Wireless Sensor Networks

Vasilis Michopoulos, Lin Guan, George Oikonomou, Iain Phillips
Department of Computer Science
Loughborough University Loughborough, UK LE11 3TU
Email: {V.Michopoulos, L.Guan, G.Oikonomou, I.W.Phillips}@lboro.ac.uk

Abstract—**In Wireless Sensor Networks (WSNs), congestion can cause a plethora of malfunctions such as packet loss, lower throughput and energy inefficiency, potentially resulting in reduced deployment lifetime and under-performing applications. This has led to several proposals describing congestion control (CC) mechanisms for sensor networks. Furthermore, the WSN research community has made significant efforts towards power saving MAC protocols with Radio Duty Cycling (RDC). However, careful study of previous work reveals that RDC schemes are often neglected during the design and evaluation of congestion control algorithms. In this paper, we argue that the presence (or lack) of RDC can drastically influence the performance of congestion detection. In addition, most WSN CC mechanisms are evaluated under traditional sensor network topologies and protocols (e.g. trickle data dissemination, tree data collection). The emerging IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) and related standards pose a new requirement: we now need to investigate if previous findings regarding congestion control are still applicable. In this context, this paper contributes a comprehensive evaluation of existing congestion detection mechanisms in a simulated, multi-node 6LoWPAN sensor network. We present results from two sets of experiments, differentiated by the presence or lack of RDC.**

I. Introduction

A wireless sensor deployment usually consists of a number of nodes monitoring their surrounding environment. Collected data traverse the network towards sink nodes in a multi-hop fashion. They are then sent to the end user from the sink node, either directly or through a gateway [1]. IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) and related specifications are becoming increasingly popular. Those technologies promise better scalability in terms of node number, due to the extended addressing scheme. Additionally, the integration between a 6LoWPAN and the Internet now becomes a question of network layer routing and alleviates the need for an application layer gateway [2].

In a WSN, congestion can occur due to simultaneous packet transmission by multiple nodes as a result of an event detection (e.g. sudden temperature rise). Current state-of-the-art congestion control schemes operate in two phases: i) congestion detection (CD) and ii) congestion avoidance (CA). The former employs algorithms aiming to predict if the network is likely to get congested, based on observed network traffic. The latter adjusts each node's transmission rate in order to avoid buffer overflow. Due to the constrained nature of sensor networks, adopting existing congestion control (CC) methods from wired networks is not suitable. For instance, it has been shown that TCP has suboptimal performance as well as high energy consumption when applied in sensor networks. This has led to the development of new CC protocols, often optimized for WSN-specific network topologies and operation (e.g. tree data collection). However, such optimizations can potentially harm the generality of the findings.

It has been shown that idle radio listening is a major source of energy consumption. In order to achieve energy savings and prolong the life cycle of a sensor deployment, research community members have made significant efforts on the development of radio cycling algorithms. With RDC schemes, nodes will turn off their transceiver for long periods of time in order to minimize idle listening. Based on a synchronisation protocol, they will turn on their radios almost simultaneously, exchange data and go back to sleep mode. Crucially, while it can have significant impact on the performance of CC, this trait of a typical WSN is often neglected when comparing various CC algorithms.

With the above observations taken into account, this paper's contribution is two-fold: i) we present results highlighting the behavior of standard congestion detection mechanisms in a multi-hop 6LoWPAN network ii) we demonstrate how RDC mechanisms affect different congestion detection mechanisms.

The rest of the paper is organized as follows. In section II we present existing congestion control schemes, accompanied by a taxonomy and discussion. Section III discusses the details of CD and CA mechanisms used by existing schemes, the comparison framework, while the simulation results and a performance analysis of the chosen CD mechanisms are also presented at the end of the chapter. Our conclusions appear in Section IV.

II. Survey of Congestion Control Mechanisms

Congestion in wireless sensor networks mainly leads to two events: i) buffer overflows and ii) radio collisions. Collisions can be prevented with medium access control approaches, such as with carrier sense (CSMA), time division (TDMA) and code division (CDMA). While there are several CSMA-based MAC protocols, reducing the frequency of radio collisions does not necessarily mean that the problem has been fully resolved. Congestion can still occur, since the local fairness achieved by CSMA protocols contributes to potential buffer overflows. WSN congestion control schemes either focus on the MAC layer or adopt a cross layer approach, combining MAC layer information with information provided by the network layer.

A. MAC Layer Schemes

Carrier sensing is an important part of MAC layers for WSNs. In [3], the authors study carrier sense performance and demonstrate that it can significantly improve the performance in heavy traffic conditions. However, carrier sense has some limitations, originating from the fact that the sender relies on local information to predict the packet reception probability. This can result in lack of information related to parent nodes, which in turn can cause collisions and thus low channel utilization.

While traditional CSMA based protocols resolve collisions by backing off in time, Power back off *(CSMA/PB)* [4] enhances CSMA by adding a transmission power control component called power back off (back off in space component). The authors argue that backing off in space is more efficient than backing off in time, demonstrating that by reducing transmission range by 50% results in a four fold decrease in contention. Low transmission range leads to a low contention path and thus higher network throughput. Using this method may lead to extra routing message exchanges in order to adjust the routes to the sink, leading to slightly increase network overhead.

Enhanced CSMA [5], attempts to improve the performance of CSMA by lowering the cost of channel state, by adding a learning approach in order to predict the probability of successful reception. In E-CSMA nodes keep state information about all their neighbours. This is acquired by recording the successful reception probability for each neighbor. Before transmitting a packet, a node uses current channel state and the reserved information of the intended receiver as references. Maintaining E-CSMA information results in increased energy consumption and network overhead.

A hybrid collision avoidance method is proposed in [6] . In this study each node operates in two alternative modes, sender initiated *(SI)* and receiver initiated *(RI)*. SI is the default mode, and uses a four-way RTS-CTS, data, ACK handshake. RI is the newly introduced mode in this hybrid mechanism, which operates with a three-way collision avoidance handshake: i) request for request to send *(RRTS)*, ii) multiple access with collision avoidance by introducing *(MCA-IB)* and iii) collision free receiver initiated multiple access *(RIMA)*. A node switches to RI mode only when it does not perform well in SI mode mode. In order to perform receiver initiated handshake, both sender and receiver need to enter RI mode. By adaptively sharing the burden of collision avoidance hand shake between the nodes, better fairness and higher throughput is achieved.

In idle sense [7], each node observes the number of idle slots between two transmission attempts and compares their theoretical estimate. It then adjusts the contention window via an Additive Increase and Multiplicative Decrease *(AIMD)* algorithm. In reality, idle sense is a modification of CSMA/CA; after contention, nodes dynamically converge (in a fully distributed manner) to similar values of their contention window instead of relying on exponential back off. In order to achieve this, a relation between the current state of the network and controlling the contention window is established. Idle sense adjusts the contention window when a collision occurs rather than detecting the collision to control the congestion. Transmission opportunities are allocated to the nodes based on the number of idle slots. This method results in higher throughput and short term fairness.

In [8] a hybrid congestion control protocol *(HCCP)* is suggested. Each node calculates its congestion degree by checking if its packet buffer is likely to overflow in case of a failed packet transmission. Nodes periodically exchange their congestion degree which is used for the calculation of the next periods transmission rate. The number of upstream neighbors is also taken into account in this calculation. The periodical exchange of packets consumes more energy and creates additional overhead to the network, especially when the frequency is very high (HCCP's performance improves as frequency increases).

In [9] an adaptive flow and back-off interval that work's in concert with energy efficient, distributed power control (*DPC*) are proposed. The onset of congestion is detected from buffer occupancies at the nodes as well as the predicted transmitting power. Then the rate selection algorithm is executed at the receivers to determine the appropriate rate. Moreover, weights can be assigned to assign different importance between the flows. In addition, in this scheme the distributed power control (*DPC*) and rate information is exchanged between the nodes. The adaptive rate scheme in this protocol is implemented at each node and acts as a back-pressure signal to minimize the effect of congestion on a hop-by-hop basis. First, the outgoing traffic flow is estimated. Consequently the congestion is alleviated by designing a suitable back off intervals for each node based on channel state and current traffic.

In [10] a mitigating congestion control protocol is recommended. This protocol uses three congestion mitigation mechanisms: i) hop-by-hop flow control, ii) source-rate limiting and iii) prioritized medium access layer that allows congestion to drain quickly at local nodes. In rate limiting mechanisms, nodes must continuously monitor their parents actions in order to generate tokens. This passive continuous listening consumes more energy and network resources. In addition, this protocol requires a tree routing structure to work correctly.

The authors of [11] propose a low-overhead congestion sharing mechanism called Interference-aware Fair Rate Control *(IFRC)*. In IFRC each node adaptively converges to a fair and efficient rate for the flows; with the use of a distributed rate adaptation technique. This is achieved by accurately sharing congestion information with potential interferes (*two nodes are potential interferes when the flow originating from one node uses a link that interferes with the link between the other node and it's parent*). IFRC consists of three inter-related components; one that measures the level of congestion at a node, another that shares congestion information with potential interferes, and a third that adapts rate using an AIMD control law. In order to measure the level of congestion at a node, a single queue for all the flows passing through that node is maintained. When the queue size exceeds an upper threshold (*the upper threshold is dynamically adjusted according to average queue size exchanged between nodes*), the node is said to be congested, and the node reduces its rate according to an AIMD rate adaptation scheme. In order to successfully achieve sharing the average queue size between nodes, the information about the current transmission rate and the average queue size in each node is attached in the header of each outgoing packet.

B. Cross Layer Schemes

Recently more and more schemes try to solve the buffer overflow problem through a cross-layer approach. Although these schemes use and combine the parameters in a different way than the MAC schemes, the basic idea is the same. These schemes support the idea that MAC level and routing level information has to be combined in order to solve the congestion problem.

Congestion detection and avoidance *(CODA)* [12] is an energy efficient, congestion control scheme for WSN. In this protocol two basic mechanism have been used in order to confront congestion: Open-loop hop-by-hop backpressure and Closed-loop multi-source regulation. With the former, a backpressure message is generated and broadcast to all one-hop downstream nodes when congestion is detected. When a node receives the backpressure message it decides based on its local network conditions if the message will be further propagated. In Closed-loop multi-source regulation when a source event rate is less than the maximum theoretical throughput of the channel the source regulates itself. On the other hand if the source rate is higher than the maximum theoretical throughput of the channel, closed-loop congestion control is triggered. In that case a source requires constant, slow time-scale feedback (*e.g. ACK*) from the sink in order to maintain its rate. As long as there is no failure to receive ACK sources can maintain their rates, otherwise rate reduction is forced. The mechanisms proposed in this protocol can successfully reduce congestion but they don't eliminate it.

Event to Sink Reliable Transport *(ESRT)* is a transport solution developed to achieve reliable event detection with minimum energy expenditure and congestion resolution functionality [13]. With ESRT, a congestion notification bit is set to the packets header when the buffer is likely to overflow during the next period. The sink based in the reliability measurement, the congestion notification bits and the previous reporting rate will compute the next new reporting rate. In this protocol there is no congestion control mechanism at the intermediate nodes; the sink is responsible upon all rate adjustments in the network. The weakness of this protocol is the need of a powerful sink node in order to broadcast the rate updates even to the most remote nodes in the network. Furthermore fairness is not considered in this protocol since in case of congestion, rate reduction is applied to all network nodes even when their traffic path is not congested.

Chen et al [14] propose an aggregate fairness model which defines end-to-end and implements it with a localized algorithm. In order to avoid congestion each node piggybacks its current buffer state in the frame header. When a child node overhears a message it caches the buffer state of it parent node. Child nodes forward packets to parent nodes only if the buffer is not full. Moreover, an aggregate fairness algorithm is used for rate reduction. When a node receives more packets than it can forward, the sensor will calculate and allocate the data rates of child nodes by a weighted fairness function. This means the actual rate from an upstream link should be proportional to the link's aggregate flow weight.

In *FACC* [15] a rate-based fairness-aware congestion control is proposed. In this protocol the intermediate relaying sensor nodes are categorized into near-sink and near-source nodes. Near-source nodes maintain a per-flow state and allocate an approximately fair rate to each passing flow by comparing the incoming rate of each flow and the fair bandwidth share. On the other hand, near-sink nodes do not need to maintain a per-flow state and use lightweight probabilistic dropping algorithm based on queue occupancy and hit frequency. This categorization allows an appropriate rate to be assigned to the near-source nodes, while energy saving and congestion avoidance is secured to the near-sink nodes by a simple algorithm. This protocol uses both routing information to characterize near sink or source nodes in combination with the busyness ratio of the 802.11 MAC to detect congestion.

C. Discussion on Congestion Detection and Avoidance

Studying the above congestion control schemes, CD mechanisms can be broken down into sub-categories: (1) monitoring the buffer state. (2) checking the local channel occupancy *(this method is also referred as intelligent CD)*. Based on how the buffer state is monitored by the different mechanisms, the first category can be further divided in: (a) buffer threshold *(Threshold can be dynamic or static)* [9] [11], (b) periodic buffer [13] [8]. Some mechanisms use both (1) and (2) for congestion detection [10] [12] [15].

The CA mechanisms implemented by the previously mentioned congestion control schemes are: (1) adjust the transmission rates locally *(when a node is congested only its child nodes adjust their rates)* [10] [9] [12] [15] [14]. (2) adjust the transmission rates of all nodes in the network *ESRT HCCP IFRC*. (3) drop a packet *(used by most cc mechanisms)*.

Since the main task of WSN is to collect and transfer data, layered architectures such as IP is not compulsory. Unified cross layer architectures, usually provide more options in saving energy and storage. On the other hand unified architectures lack in evolvability, scale, diversity of applications, interoperability and standardization.

III. Comparisons and Performance Evaluation

As discussed in section II, congestion control use different mechanisms for CD and different for CA. The two most used CD mechanisms are: (1) measuring buffer occupancy and (2) intelligent CD, calculating the congestion degree by dividing the packet service time with packet inter arrival time

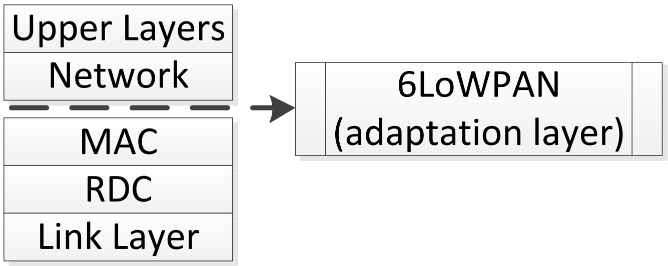

Fig. 1. The Contiki uIPv6 Network Stack

($CD = P_s/P_a$). Buffer occupancy can be measured either with a buffer threshold *(when packets in the buffer exceed the threshold, congestion is detected)* or with a periodic buffer *(the node calculates if the buffer would overflow, if it were to receive the same number of packets in the next period P)*. For CA many different methods have been used, but all of them share the same concept *(reduce rates when congestion is detected and then increase them periodically with an AIMD like scheme)*.

A. Methodology and Simulation Settings

For this study, we implemented the two different ways of detecting congestion based on the buffer occupancy as well as the mechanism calculating the congestion degree based on $CD = P_s/P_a$. In order to evaluate the different ways to detect congestion, we tested them under the same AIMD mechanism. When a node receives a congestion notification message, it reduces its rate to 50%. Furthermore each node increases its rate by reducing the inter packet transmission time. Let t_i be the inter packet transmission time, then the the transmission rates will be increased by reducing the t_i by t_i/δ every t_i seconds. Hence we have $t_{i+1} = t_i - \frac{t_i}{\delta}$. It is trivial to show that the transmission rate r_i will be a linear function with slope δ. The choice of δ dictates the intensity of additive increase and thus choosing a suitable value is of great significance. Let t_{min} be the lowest value for inter transmission time *(highest rate)* and t_{max} the highest after a rate reduction *($t_{max} = t_{min}/2$)*. In order to avoid t_{max} to jump to t_{min} in a single step we require $t_{max}/\delta << t_{max}$. In order to achieve these requirements we set δ to $\alpha/\sqrt{t_{max}}$ where α is a positive number greater than one *($\alpha > 1$)*. The final equation will be: $t_{i+1} = t_i - \frac{t_i}{\alpha * \sqrt{t_{max}}}$.

The comparison of the above mechanisms has been conducted with COOJA [16], a cross layer simulator part of the Contiki embedded operating system *(the IPv6 - 6LoWPAN layer structure of contiki os can be seen in Figure 1)*. The radio medium used for the experiments is called unit disk graph medium with distance loss UDGM *(allows modification of radio transmission ranges ny the user)*. In order to perform our evaluation, we implemented the mechanisms as MAC layer extensions. The buffer size used in MAC layer is 10 packets *(total size 10 * 128 bytes)*. Furthermore, there is no buffer for incoming packets *(only outgoing packets are stored in the buffer)*. In the RDC layer, both sicslowmac and contikimac have been used for the experiments. Sicslowmac is a simple protocol which creates the 802.15.4 frames and forwards them to the next node, without ever turning the radio off (no RDC). On the other hand contikimac makes use of 802.15.4 link layer acknowledgment (ACK) frames and turns the radio off for energy saving mode *(with RDC)*. Routing in our experiments was handled by RPL *(IPv6 Routing Protocol for Low power and Lossy Networks)*, a new highly modular routing protocol, specifically designed for long lived networks. The sources were producing CBR traffic of 4 packets/s, and each packet had a 32-byte payload. The motes emulated in COOJA for our experiments are sky motes and the total duration of each simulation was 50 seconds.

TABLE I
NUMBER OF SUCCESSFULLY RECEIVED PACKETS AT THE SOURCE

Throughput	*Threshold*	*Period 0.5*	*Period 0.25*	*Service/Arrival*	*No CC*
Sicslowmac	699	742	747	165	264
Contikimac	123	115	106	81	69

In our experiments, we had a total of 20 nodes placed randomly in a grid *(multiple experiments with random topologies had been made with similar results)*. Among them them we had 6 sources and 1 sink, while the remaining nodes were intermediates. For the evaluation we used two metrics: i) total packets received by the source and ii) packet loss.

B. Total Packets Received and Loss Comparison

Fig. 2 displays the total received packets/second while Fig. 4 shows the loss for each of the selected CD mechanisms *(see section III-A)*, operating with the Sicslowmac RDC. We can observe that mechanisms using buffer occupancy performed much better than intelligent CD. We also observe that intelligent CD was less successful in receiving packets than in scenarios without congestion control. Studying Fig. 2 and Table I, mechanisms that detect congestion based on buffer state had significantly higher number of successfully received packets than both no congestion control and intelligent CD. Both buffer threshold and periodic buffer performed similarly in terms of total packets received by the sink, with periodic buffer demonstrating a slight increase in received packets. On the contrary, buffer monitoring based on threshold had lower loss than periodic buffer. Furthermore it is noteworthy that with different period times, periodic buffer had significant differences in loss *(the smallest the period the lower the loss)*. More detailed results about the different performance of periodic buffer when different time periods used, can be found in [8]. Without congestion control, network performance was worse than under the presence of most of the congestion control mechanisms, both in terms of total packets received as well as in terms of loss rate.

Fig. 3 displays the total packets received while Fig. 5 the loss for each of the selected CD mechanisms, operating with RDC *(contikimac)*. With RDC all congestion control mechanisms had better performance than no congestion control, for both total received packets and loss. We can also observe that, similarly to the previous experiment, mechanisms detecting congestion based on the buffer state had significantly better

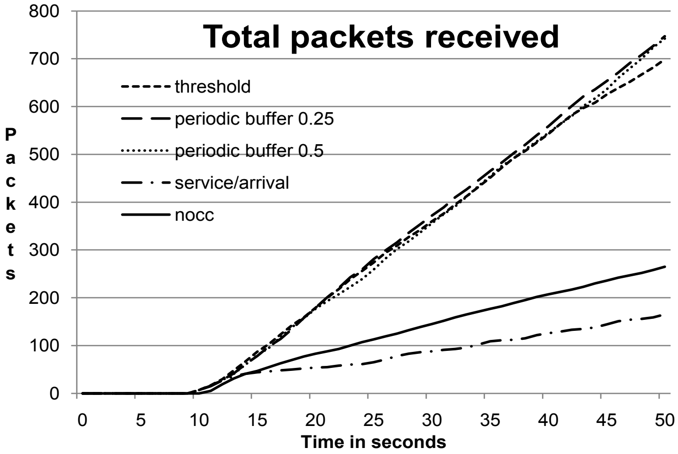

Fig. 2. Successfully received packets without RDC (sicslowmac)

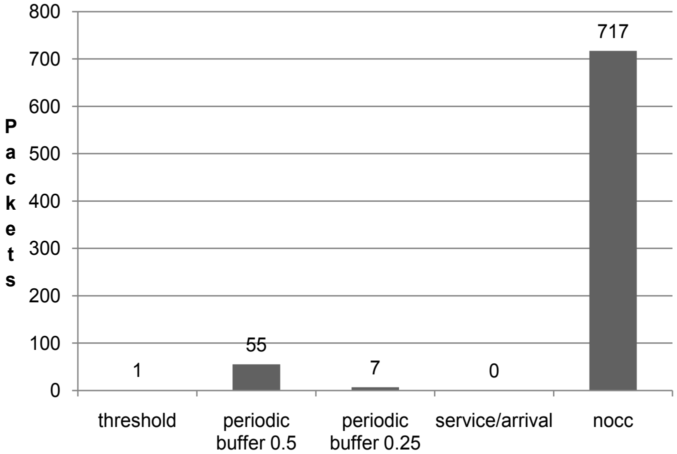

Fig. 4. Packet Loss without RDC (sicslowmac - always on)

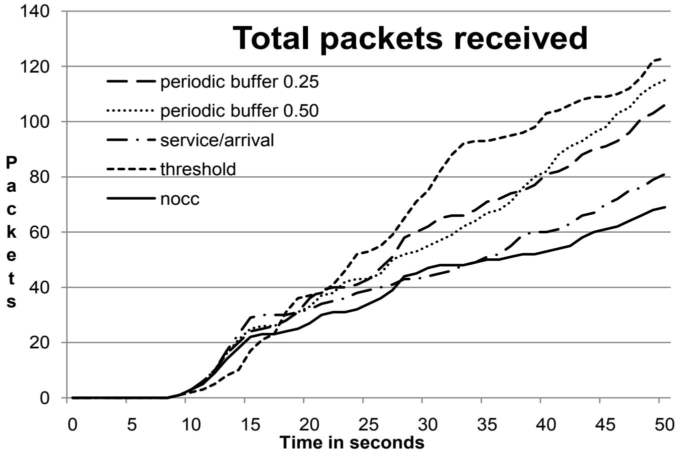

Fig. 3. Successfully received packets with RDC (contikimac)

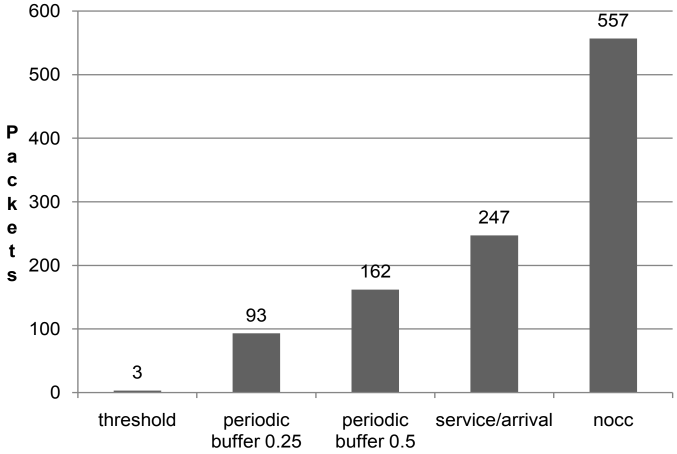

Fig. 5. Packet Loss with Contikimac

results than intelligent CD and no congestion control. In contrast to simulations without RDC, buffer threshold based CD had higher number of received packets than periodic buffers. In addition buffer based CD had the lowest loss. With RDC big differences are noticeable at the performance of intelligent CD. In this experiment, intelligent CD performed better than no congestion control in terms of total successfully received packets. On the contrary intelligent CD had the highest loss in comparison to the rest of the CD mechanisms. Similarly to the previous experiment, without congestion control the network's performance was worse than in any other case.

C. Performance Analysis

When a mechanism has more successfully received packets within the same time interval, it is natural to conclude that this mechanism has higher throughput. Based on the previously discussed simulation results, we observe that intelligent CD performed poorly. In comparison to buffer state CD mechanisms, intelligent CD had over 200% fewer received packets in the first experiment *(no RDC)*. This poor performance can be justified if we take under consideration RPL's control messages. When a node simultaneously receives multiple broadcasts *(RPL control messages)*, intelligent CD assumes congestion. Consequently, the rate reduction algorithm is trigered, which in turn explains why this mechanism resulted in the lowest number of received packets. On the other hand, low transmission rates can explain why this mechanism had no loss. Based on the experiments, we observe that without RDC, CD based on periodic buffer had over all more received packets and thus the highest throughput but higher loss than CD based on buffer threshold and intelligent CD. The reason behind the higher throughput of this mechanism is the higher buffer utilization *(CD based on buffer threshold will always trigger back off when the buffer capacity reaches the threshold therefore only a percentage of the total capacity of the buffer is used. On the other hand periodic buffers can utilize buffer capacity more efficiently and avoid rate reduction messages, even when buffer occupancy levels are very high)*.

Further analysis of the above experiments reveals that CD (and thus congestion control) can be directly affected by RDC. Without RDC, all CD mechanisms detected and confronted congestion successfully. On the other hand, with RDC the experiment results were different, especially in the case of intelligent CD. Without RDC, it achieves zero packet loss but suffers from low numbers of received packets and thus throughput. In contrast, when RDC is used the performance of this method is closer to the other CD mechanisms in terms of received packets and throughput, but at the cost of a very high packet loss rate. With RDC, nodes synchronize before transmissions and frequently turn the radio off *(radio is also turned off in case of collision).* Furthermore, since the RDC layer underlies MAC, RDC synchronization message exchange happens transparently and is not taken into account by the MAC layer CD algorithm. These characteristics of Contikimac *(with RDC)* reduce the rate at which packets are forwarded to the MAC layer and thus intelligent CD does not detect congestion as frequently as in the previous experiment. In contrast, when RDC is enabled this mechanism fails to successfully detect congestion, hence packets are lost. Even though CD schemes based on buffer state confronted congestion successfully in both experiments, the results were diverse. With RDC, buffer based on threshold CD resulted in approximately 20% higher amount of received packets than periodic buffers. On top of that, CD based on buffer threshold maintained insignificant packet loss. In contrast to this CD based on periodic buffer had high packet loss. RDC causes radio to ocasionaly turn off. When the radio turns on again previous nodes with already built-up buffers can transmit multiple packets in sequence. When congestion is detected periodicaly *(the mechanism is checking if the buffer is going to overflow if the same number of packets stored in the buffer during the next period)*, such traffic paterns can erroneously assume that congestion is imminent.

Over all, CD based on buffer threshold is the most efficient method for congestion detection with RDC since it achieved more successfully received packets, higher throughput and lower loss than the other mechanisms. Furthermore, as observed in our experiments, intelligent CD is an inferior CD method for IPv6 sensor networks *(the performance of this mechanism can be different in non IP sensor networks or in combination with different rate reduction schemes).*

Based on Figures 2 and 3, it is notable that without RDC and packet ACKs *(sicslowmac does not use any packet ACKs)*, our network demonstrated a six-fold throughput increase. On the other hand it is not very realistic to assume a wireless sensor network operating without RDC, since efficient RDC can extend a deployment's lifetime expectancy by orders of magnitude, up to years.

Lastly, the decision of which mechanism is better totally depends in the demands of the WSN. If the sensor network requires high throughput *(no RDC)* and can afford a slightly higher packet loss, periodic buffer can be the best option, while in the remaining cases CD based on buffer threshold can successfully detect and confront congestion.

IV. Conclusion and future work

In this work, we conducted a simulation-based comparative study concerning the performance of multiple CD mechanisms in 6LoWPAN sensor networks. We implemented the mechanisms for the Contiki operating system and tested them under different network conditions and protocols. Our main conclusion of interest is that existing congestion detection methods *(packet service time / packet arrival time)* perform poorly in 6LoWPAN sensor networks. Moreover, it is shown that CD algorithm performance is directly influenced by the radio duty cycling mechanisms. This study is a step towards understanding the performance of various CD mechanisms in a 6LoWPAN environment and can serve as a basis for future network designers, when facing the decision of which CD mechanism to adopt for their deployment. Our future plans include porting those mechanisms to our contiki fork for Sensinode/cc2430 hardware and evaluating them in real, large scale field deployments.

References

[1] L. Yong-Min and L. Wu-Yi, "Improved fairness using dccp in wireless sensor networks," in *Networking and Digital Society, 2009. ICNDS '09. International Conference on*, vol. 1, May 2009, pp. 141 –144.

[2] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP*. Morgan Kaufmann, 2010.

[3] K. Jamieson, B. Hull, A. Miu, and H. Balakrishnan, "Understanding the real-world performance of carrier sense," in *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, ser. E-WIND '05. New York, NY, USA: ACM, 2005, pp. 52–57.

[4] C. J. Colbourn, M. Cui, E. L. Lloyd, and V. R. Syrotiuk, "A carrier sense multiple access protocol with power backoff (csma/pb)," *Ad Hoc Netw.*, vol. 5, pp. 1233–1250, November 2007.

[5] S. Eisenman and A. Campbell, "E-csma: Supporting enhanced csma performance in experimental sensor networks using per-neighbor transmission probability thresholds," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 1208 –1216.

[6] Y. Wang and J. J. Garcia-Luna-Aceves, "A hybrid collision avoidance scheme for ad hoc networks," *Wirel. Netw.*, vol. 10, pp. 439–446, July 2004.

[7] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '05. New York, NY, USA: ACM, 2005, pp. 121–132.

[8] J.-P. Sheu, L.-J. Chang, and W.-K. Hu, "Hybrid congestion control protocol in wireless sensor networks," in *Proc. Vehicular Technology Conference*, 2008.

[9] M. Zawodniok and S. Jagannathan, "Predictive congestion control protocol for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, Nov. 2007.

[10] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *Proc. 2nd international conference on Embedded networked sensor systems* , 2004.

[11] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," in *Proc. ACM SIGCOMM Computer Communication Review*, 2006.

[12] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, " CODA: congestion detection and avoidance in sensor networks," in *Proc. 1st international conference on Embedded networked sensor systems* , 2003.

[13] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, " ESRT: EventtoSink Reliable Transport in Wireless Sensor Networks," in *Proc. 4th ACM international symposium on Mobile ad hoc networking & computing* , 2003.

[14] S. Chen and Z. Zhang, "Localized algorithm for aggregate fairness in wireless sensor networks," in *Proc. International Conference on Mobile Computing and Networking Proceedings*, 2006.

[15] X. Yin, X. Zhou, R. Huang, and Y. Fang, "A fairness-aware congestion control scheme in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, Nov. 2009.

[16] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level simulation in cooja," in *European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*, 2007.