Back Pressure Congestion Control for CoAP/6LoWPAN Networks

Angelo P. Castellani\textsuperscript{a,b,*}, Michele Rossi\textsuperscript{a,b}, Michele Zorzi\textsuperscript{a,b}

\textsuperscript{a}Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131, Padova, Italy
\textsuperscript{b}Consorzio Ferrara Ricerche, via Saragat 1, 44122, Ferrara, Italy

Abstract

In this paper we address the design of network architectures for the Internet of Things by proposing practical algorithms to augment IETF CoAP/6LoWPAN protocol stacks with congestion control functionalities. Our design is inspired by previous theoretical work on back pressure routing and is targeted toward Web-based architectures featuring bidirectional data flows made up of CoAP request/response pairs. Here, we present three different cross-layer and fully decentralized congestion control schemes and compare them against ideal back pressure and current UDP-based protocol stacks. Hence, we discuss results obtained using ns-3 through an extensive simulation campaign for two different scenarios: unidirectional and upstream flows and bidirectional Web-based CoAP flows. Our results confirm that the proposed congestion control algorithms perform satisfactorily in both scenarios for a wide range of values of their configuration parameters, and are amenable to the implementation onto existing protocol stacks for embedded sensor devices.

Keywords: Congestion Control, Back Pressure, Protocol Design, Internet of Things, Web Services, CoAP, 6LoWPAN, Simulation

1. Introduction

In the last few years, we have witnessed considerable advances in terms of protocol design for wireless sensor networking. These have led to a solid understanding of the problems related to channel access, routing and data gathering, delivering efficient protocol stacks and ultimately spurring the standardization of protocols for data collection and addressing.

The work in this paper considers network protocols recently standardized by IETF, namely, CoAP [1] and 6LoWPAN [2], whose combined use permits Web-based bi-directional communications between sensor devices and Internet servers. 6LoWPAN provides header compression and specifies communication profiles that allow the implementation of IPv6 addressing. CoAP is a stateless protocol that is aimed at replacing HTTP for lightweight and resource constrained devices. As such, it implements a reduced set of functionalities with respect to HTTP. While CoAP and 6LoWPAN provide the basis for Web-oriented protocol stacks for embedded devices and natively support UDP traffic, they do not fully address the congestion problem, and only provide some conservative recommendations, as we discuss below in Section 4.

The Internet protocol suite, i.e., TCP/IP, has been designed adopting the “end-to-end argument” [3], which has proven to be very effective in networks of smart terminals operating bulk data transfers. However, TCP congestion control (CC) [4, 5] has been designed with an implicit assumption: data transfers causing congestion are usually long enough to be efficiently controlled through end-to-end CC algorithms. By their own nature, slow start and congestion avoidance are techniques that converge after some time and after a potentially large amount of data has been transferred. However, when the amount of data required to create congestion on the network is very small, these techniques do not provide an efficient solution to the CC problem. In addition to this, TCP is known to be severely impacted by the long delays that are typical of constrained networks.

In this paper, following the research lines identified in [6], we develop practical congestion control algorithms for constrained Internet of Things (IoT) exploiting 6LoWPAN technology. These networks are characterized by very constrained processing, memory and communication capabilities [7], a potentially large number of nodes, and infrequent communication patterns which very much differ from standard Internet flows.

Our present work quantifies the benefits of implementing congestion control at layer 3 by exploiting practical and lightweight algorithms based on the concept of back pressure routing by Tassiulas and Ephremides [8]. Since its conception, Back Pressure (BP) policies have been extensively explored, leading to distributed theoretical algorithms that achieve optimal throughput performance in distributed networks. Practical applications of these schemes have also been studied in several papers such as [9], which applies a similar policy to the queues of wireless sensor nodes to realize an efficient data collection protocol. However, that solution makes strong use of channel...
snooping and poses limitations on the implementation of radio duty-cycling (RDC). In [10] the authors propose CODA, a distributed algorithm that uses explicit messages to detect congestion and therefore can also work in the presence of RDC. An evaluation of CODA in 6LoWPAN networks can be found in [11], where the authors measure the loss probability and the number of delivered packets. While these studies on CODA are of interest for the application of congestion control principles in energy efficient networks, some practical issues still remain open, namely: i) the explicit BP messages are not provided in standard existing protocols, and therefore cannot be used in standard networking stacks, and ii) there is no discussion on some important issues such as the effect of the required number of hop-by-hop retransmissions and of bidirectional CoAP traffic support.

Our main objective in this paper is to systematically compare through detailed simulations different lightweight BP approaches, including existing as well as original algorithms, in order to assess their suitability for the implementation into IoT devices and their benefits in terms of performance gains.

The main contributions of this paper are the following:

- We propose a number of practical and lightweight congestion control algorithms for constrained devices, devising CC policies based on distributed back pressure control, with the objectives of detecting and alleviating network congestion, providing reliability and ultimately controlling the injection of data traffic into the network.

- We present extensive simulation results by comparing the performance of the proposed CC policies with that of ideal back pressure algorithms and showing that layer 3 BP congestion control is feasible on constrained IoT devices, and results in significant performance gains at the expense of minimal additional complexity.

- We present protocols and results for unidirectional and upstream data traffic as well as for bidirectional CoAP flows.

The remainder of this paper is organized as follows. In Section 2 we describe the system model and present our practical BP-based congestion control algorithms for constrained devices. First, in Section 3 we evaluate the performance of these algorithms focusing on unidirectional and upstream data collection. Then, in Section 4 we consider bidirectional communication scenarios such as those arising from CoAP-based Web-services. Finally, we draw our conclusions in Section 5.

2. Back Pressure Congestion Control on 6LoWPAN

In the following, we present some CC designs that are explicitly tailored to constrained networks featuring infrequent communication patterns. Specifically, we propose to perform congestion control actions at the network layer, as this allows the implementation of BP algorithms that work on aggregates of datagrams, i.e., on IP queues. Note that working on aggregates is desirable due to the nature of the traffic found in 6LoWPAN networks, which usually reaches considerable volumes only when the output of multiple nodes is combined. Moreover, this results in a lower complexity in terms of software structure, memory utilization and communication requirements for the control of network queues.

2.1. Node Model

Each node has been modeled according to the Internet Host model [12], which classifies protocols into Link, Network, Transport and Application layers.

Link. 6LoWPAN has been specifically designed for the IEEE 802.15.4 PHY/MAC [13]. Thus, in our model each node is equipped with an IEEE 802.15.4 radio transceiver operating at 2.4 GHz with a nominal available transmission rate of 250 kb/s. Layer 2 operates according to the IEEE 802.15.4 standard and the total number of transmissions per packet is limited to a maximum of 7.

Network. IPv6 and 6LoWPAN belong to this category; our node has been equipped with a standard layer 3 device (L3D) operating as follows. For each IP datagram, received either from the applications residing in the upper layers or from the radio, L3D first understands whether this datagram has to be delivered to the local host.

As a second step, L3D looks in the Internet routing table, extracts the next hop toward which the datagram has to be sent, and places the received datagram into the layer 3 queue for outbound traffic. This queue is managed according to a First-In First-Out Drop Tail (FIFO-DT) discipline. Note that we account for a single IP queue at layer 3, which is a realistic limitation and is typical of constrained devices.

Our L3Ds implement hop-by-hop layer 3 retransmissions and different BP control algorithms, as specified in Sections 2.2 and 4.1.

Transport. The UDP transport protocol is adopted. UDP only performs a checksum check for every received datagram, without any further processing or buffering operations.

Application. We have considered two usage scenarios:

S1) Unidirectional flows (Section 3): for the study of unidirectional upstream data traffic, we have adopted the Iperf [14] protocol. It permits to evaluate at the receiver the number of lost packets, the number of out-of-order deliveries, the multi-hop delay and the per-packet jitter. Data sources emit UDP traffic at a constant bit rate (CBR), except for the cases where the local layer 3 queue is full. In these cases, the emission of the datagram is delayed until a layer 3 queue slot becomes available.

S2) Bidirectional flows (Section 4): to evaluate the effectiveness of the proposed congestion control algorithms for bidirectional traffic, we have used the Constrained Application Protocol (CoAP) [1] to transport Iperf messages. CoAP implements a lightweight bidirectional exchange targeted to client-server architectures. In this case clients are placed outside the constrained IoT domain and emit
CoAP requests at a constant bit rate. These requests are sent to a border gateway and from there to the IoT nodes. Upon receiving these requests, IoT nodes reply with CoAP responses that flow in the opposite direction.

### 2.2. Layer 3 Device Types

We advocate the implementation of congestion control through the use of practical back pressure techniques, which are embedded into the layer 3 device of each sensor node. Next, these augmented L3Ds are presented in detail, whereas their performance evaluation is carried out in Sections 3 and 4, where we respectively look at unidirectional and bidirectional flows.

**Static.** This is the L3D described in Section 2.1, which does not account for any congestion control mechanism. It is a baseline scheme considered here to gauge the advantages offered by the following BP schemes.

**IdealBP.** This L3D refrains from transmitting as long as the queue length at the next hop is higher than that of the local queue. This behavior mimics the ideal BP policies devised by Tassiulas and Ephremides [8]. Note that in actual implementations nodes can only know the queue length at the next hop through the exchange of proper control signals. For IdealBP, in our simulations this information is made available to any node through a genie. Although IdealBP is impractical, we have considered it here to validate the BP approach and also see how much its performance deviates from that of the practical algorithms that we propose next.

According to IdealBP’s BP policy, the datagram at the current node is transmitted to the next hop whenever their queue differential is positive and the remote queue length at layer 3 is smaller than a pre-determined threshold $Q_{thr} > 0$. This threshold is required because it could happen that multiple devices concurrently send their datagrams (one per device) to the same next hop. In this case, the queue at the next hop could overflow even though the preceding queue differential was positive.

**Griping.** This device uses an explicit BP signal on congestion and is similar to the CODA BP policy [10], that has also been evaluated in [11]. Differently from [10] and [11], in Griping subsequent BP control messages must be transmitted at least $K$ seconds apart, where $K$ is a tunable parameter. In fact, we noticed that close transmissions of BP control messages toward the same source lead to inefficiencies in terms of transmission overhead. Griping works as follows.

- **Receiver:** whenever a Griping L3D receiver gets a new layer 3 datagram and its layer 3 queue length $Q$ is larger than a threshold $Q_{thr} > 0$, it transmits a unicast BP control message back to the source of that datagram. Subsequent BP control messages transmitted by the same source must be spaced by at least $K$ seconds. The transmission of a new BP is canceled when this condition is not met.

- **Transmitter:** at any time, each Griping transmitter sends its own datagrams at a rate that is updated according to an Additive Increase and Multiplicative Decrease (AIMD) approach. Specifically, the Griping transmitter works considering reference “time slots” of $T$ seconds each. Within each time slot it can transmit up to $W$ datagrams. Every time the Griping transmitter receives a BP message, $W$ is halved, thus effectively halving the transmission rate. If no BP message is received in $T$ seconds, then $W$ is increased by one unit. In our simulations, the parameters $K$ and $T$ have been tuned and subsequently set to 100 ms and 750 ms, respectively.

**Remarks.** Due to its simplicity, Griping is amenable to the implementation on constrained nodes. Moreover, we note that this technique does not require any interaction with the PHY and MAC layers and therefore does not rely on their specific implementation. This makes it possible to implement Griping with radio duty cycling, which is a critical feature for wireless sensor networks. Layer 3 losses in Griping occur in two cases.

C1) **Receiver side:** a packet is correctly received at layer 2 and is passed to layer 3, where the network queue is full. The packet is thus discarded and a layer 3 queue overflow occurs.

C2) **Sender side:** a packet is discarded when none of the allowed retransmissions at layer 3 has led to its successful reception.

**Deaf.** This is an alternative approach that aims at removing the complexity associated with the transmission of BP control messages. Deaf is implemented as follows.

- **Receiver:** a Deaf receiver stops sending layer 2 acknowledgments whenever the layer 3 queue length $Q$ is larger than a predefined threshold $Q_{thr} > 0$. The stopped acknowledgment flow is perceived by the Deaf transmitter as an implicit BP notification.

- **Transmitter:** a Deaf transmitter handles layer 3 retransmissions as follows. After a new layer 3 datagram transmission, a back off timer is initialized for this datagram to $T_{wait}$ seconds, where $T_{wait}$ is drawn from a random variable uniformly distributed in $[0, 2^n T]$, where $T$ is the duration of a layer 3 time slot and $n$ is a function of the total number of transmissions performed for this datagram. Thus, the retransmission of this datagram at layer 3 will occur no earlier than $T_{wait}$ seconds from its current transmission attempt. $n$ is updated as follows: after each retransmission of the datagram, a transmission counter, $n_{tx}$, is incremented by one and the length of the previous interval for $T_{wait}$ is adapted by picking a new value of $n$ as: $n = \min(n_{tx}, n_{max})$, with $n_{max} = 4$. $T$ has been set to 0.1 s and for the first transmission of the datagram $n_{tx}$ is initialized to zero. Note that $T_{wait}$ is doubled at each failure of the same datagram, which is equivalent to halving the associated transmission rate. The underlying layer 2 notifies layer 3 about the transmission status (either failed or

---

1Note that the concurrent transmissions occur at layer 3; lower layers will multiplex these transmissions so as to avoid collisions, by possibly retransmitting collided packets.
successful) of every upper layer datagram. We say that a layer 3 datagram failure event occurs whenever the maximum number of retransmissions is reached for this packet, which is still unsuccessfully delivered.

Remarks. Note that this technique does require some cross-layer interaction between layer 3 and layer 2. In fact, a given layer 2 frame is not acknowledged by *Deaf* whenever the overlying layer 3 communicates to the lower layer a failure for that packet. However, this does not require the processing of further PHY- or MAC-layer metrics. For this reason, we argue that *Deaf* does not interfere with radio duty-cycling as others cross-layer approaches usually do. Note that congestion control in this case is enforced by spacing apart the retransmissions of the same packet at layer 3. This amounts to decreasing the actual layer 3 transmission rate through implicit notifications from the *Deaf* receiver. Also, there is a semantic violation in that retransmissions are performed at layer 3, which is in contrast with standard congestion control that is implemented in the transport protocol at layer 4. Also, *Deaf* never acknowledges layer 2 packets whenever the layer 3 queue is full. Thus, event C1 above never occurs and packets can only be discarded due to event C2.

/Fuse*. *Fuse* combines the BP actions of *Griping* and *Deaf*. As we shall see below, this combined action effectively reaps the benefits of both *Deaf* and *Griping* BP policies. Specifically:

- **Receiver**: the *Fuse* receiver behaves as *Griping* as long as its queue length is smaller than the maximum queue size $Q_{\text{max}}$. When the layer 3 queue is full, *Fuse* combines the BP actions of *Griping* and *Deaf*, i.e., it stops sending layer 2 acknowledgments, as *Deaf* does, but also continues to send explicit congestion notification messages, as *Griping* does.

- **Transmitter**: the *Fuse* transmitter implements the AIMD rate control policy, as discussed above for the *Griping* transmitter.

### 3. Unidirectional Upstream Data Traffic

In this section we analyze the performance of the back pressure algorithms of Section 2.2 when applied to 6LoWPAN networks adopting RPL [15] and transmitting data packets over unidirectional and upstream flows, i.e., from some source sensor nodes to the border router which interconnects the constrained sensor network to the unconstrained Internet, see the first case study of Fig. 1.

#### 3.1. System Parameters

The following parameters have been chosen to evaluate their impact on the performance.

**Offered Traffic Load.** $\lambda_{\text{tx}}$ defines the rate at which each source emits UDP datagrams at layer 3 and is measured in packets per second.

#### 3.2. Performance Metrics

To compare the performance of the various L3Ds, the following metrics have been considered.

**Reception rate.** $\lambda_{\text{rx}}$ defines the average rate at which layer 3 packets are correctly received by the destination, and is measured in packets per second.

**Multihop delay.** $D$ refers to the time taken by a packet to be correctly received by the border router (BR, see Fig. 1) from its transmission instant at the source (one of the 9 leaf nodes of the routing tree of Fig. 1). $D$ is obtained averaging the packet delay over all packets that are correctly received by the border router.

**Loss probability.** $P_{\text{loss}}$ represents the probability that an emitted layer 3 datagram is lost either due to buffer overrun or because the maximum number of retransmissions has been reached. $P_{\text{loss}}$ is computed as the ratio of the total number of datagrams lost in the network to the total number of datagrams emitted by all sources.
Rejection rate. $R$ defines the average rate at which packets from the application are rejected by the network layer due to a full layer 3 queue. In this case, application layer packets are not lost at layer 3 but their insertion into the layer 3 queue is denied and an error message is sent to the application layer. Upon receiving this error message, the application slows down and multihop delay $D$ vs. the offered traffic load $\lambda$.

Transmission Overhead. $(N_{ax})$ Represents the average total number of layer 2 packets that are transmitted in the network for the successful end-to-end (from a leaf node to the border router) delivery of a single layer 3 datagram. This metric accounts for the number of layer 2 packets that are sent to carry layer 3 data messages as well as layer 3 BP control messages, such as those sent by Griping and Fuse for the explicit signaling of a congestion event.

3.3. Results for Upstream Unidirectional Traffic

Simulation setup. Simulations have been run in ns-3 [16], using IEEE 802.15.4 at the PHY and MAC layers. At layer 2 packets have a fixed size of 127 bytes, including 12 bytes for the layer 2 headers. Layer 3 datagrams have a fixed size of 115 bytes, which means that a layer 3 datagram fits into a layer 2 packet and fragmentation is not needed. A tree topology has been built, see Fig. 1, containing 9 leaf nodes, 4 routers (R) and a border router (BR).

This topology is representative of a typical routing scenario for 6LoWPAN networks adopting RPL [15], a recently standardized routing protocol for low-power lossy networks.

In the following results, errors due to wireless transmissions have not been considered as neglecting channel impairments makes it possible to isolate and characterize the effects that are solely due to the considered congestion control algorithms, which is the main purpose of our study in this paper.

The simulation duration is set to 200 seconds, and $Q_{\text{max}}$ is set to 31 packets to resemble typical limitations of IPv6/6LoWPAN stacks on constrained hardware platforms (see [17]). Each of the points in the following graphs is obtained averaging 180 independent simulation runs.

Impact of varying the transmission rate $\lambda_{tx}$. In Fig. 2 we show $\lambda_{tx}$ and $D$ for each L3D device as a function of $\lambda_{tx} \in [1, 30]$ pkt/s. The remaining system parameters have been set to: $N_{\text{retx}} = 15$, $Q_{\text{max}} = 31$ packets, and $Q_{\text{thr}} = 15$ packets.

Considering the Static device, as long as the input rate $\lambda_{tx}$ remains smaller than a certain saturation threshold $\lambda_{sat}$ (of about 9 pkt/s in Fig. 2), we have that the reception rate $\lambda_{tx}^{\text{sat}}$ is approximately equal to $\lambda_{ax}^{\text{Static}}$ and $D^{\text{Static}}$ is stable and small. This means that the network can effectively serve the injected data traffic. Here, the packet delay is typically dominated by the transmission and propagation delays over the multi-hop paths from the sources to the BR, whereas the queueing delay is negligible. As $\lambda_{ax}^{\text{Static}}$ grows larger than $\lambda_{sat}$, $\lambda_{ax}^{\text{Static}}$ saturates reaching the so called saturation throughput. At this point, $D^{\text{Static}}$ grows abruptly and this is due to the queueing component of the delay, that considerably increases as $\lambda_{ax}$ becomes higher than the actual layer 2 service rate. Similar performance tradeoffs are observed for all L3Ds.

While there are no substantial differences between Static and the other L3Ds in terms of $\lambda_{tx}$, we note that all the other devices obtain an average delay $D$ increased by a factor of about 4 during congestion if compared with Static. This is due to the fact that these devices put off the transmission of new layer 3 packets when the network is congested, whereas Static keeps transmitting at a fixed rate, irrespective of the congestion status of the network. Also, Griping and Fuse account for the longest delay, and this is due to their explicit transmission of back pressure messages. From this first figure we observe that back pressure tends to increase the delay but is able to retain most of the throughput performance of the greedy Static transmission policy.

The rejection rate $R$ has been plots in Fig. 3 for the same simulation parameters. For BP devices, flow congestion actions are taken as soon as $\lambda_{ax}$ becomes equal to $\lambda_{sat}$ and are enforced as long as $\lambda_{tx} \geq \lambda_{sat}$. These actions correspond to increasing the layer 3 rejection rate $R$. We note that IdealBP has the lowest rejection rate and the highest reception rate among all BP schemes, and thus, as expected, it is the best performing algorithm, i.e., the one able to fully exploit the benefits of back pressure.

$R$ of Deaf, Griping and Fuse is very similar and close to that of IdealBP. Moreover, their back pressure policy becomes effective when $\lambda_{tx} \geq \lambda_{sat}$, which is testified by the prompt increase in $R$ when the network operates beyond the saturation point.

Static keeps sending packets at the maximum possible rate, irrespective of the queue status at the relays. This moves to the right the value of $\lambda$ for which layer 3 queues are filled up and packets start to be rejected (the increase of $R$ becomes apparent for $\lambda_{tx} \geq 20$ pkt/s in Fig. 3). However, as we shall see below the drawback of this aggressive transmission behavior is that layer 3 queues are subject to higher loss rates.

As we show shortly, Griping, Deaf and Fuse have a substantially smaller $P_{\text{loss}}$ than Static as they reject only the data traffic.
that the network cannot sustain, mimicking IdealBP’s behavior. Note that layer 3 rejection does not imply discarding packets but rather slowing down the packet generation rate at the application.\(^2\)

In Fig. 4 we show the transmission overhead \(N_{tx}\) as a function of the offered traffic load \(\lambda_{tx}\). As expected, IdealBP has the best performance among all schemes as it applies BP control by leveraging at no cost the exact and instantaneous knowledge of all network queues, which is provided in the simulations through a genie. As \(\lambda_{tx}\) increases beyond \(\lambda_{sat}\), all the remaining schemes show a degraded performance in terms of \(N_{tx}\). Deaf is the scheme that leads to the highest transmission overhead and this is inherent in its design, as this scheme tends to hit the maximum number of retransmission attempts while handling congestion control. Static is the second-worst as in this case congestion is emphasized through the careless injection of data traffic. Griping and Fuse both perform very close to IdealBP as the corresponding BP policies explicitly send congestion notifications to the senders and this has the effect of timely slowing down the volume of data that is injected into the network, alleviating the congestion.

**Impact of varying \(N_{retx}\).** Fig. 5 shows the loss probability \(P_{loss}\) as a function of \(N_{retx}\). The remaining system parameters have been set to: \(\lambda_{tx} = 20\) pkt/s, \(Q_{max} = 31\) packets, and \(Q_{nce} = 15\) packets. Note that a transmission rate \(\lambda_{tx} > \lambda_{sat}\) has been chosen so as to measure the ability of the different L3Ds to handle network congestion.

From Fig. 5 we observe the expected result that \(P_{loss}\) generally decreases as \(N_{retx}\) grows. This decrease is faster for Griping, Fuse and IdealBP as these algorithms use explicit signaling to detect congestion. The initial \(P_{loss}\) decrease is slower for Deaf which therefore shows worse \(P_{loss}\) performance for small values of \(N_{retx}\), say, \(N_{retx} \leq 7\). As expected, Static has the worst reliability performance as retransmissions are disabled for this scheme.

Also, Griping has a floor at \(P_{loss} \approx 0.02\), which is due to the inherent delay incurred in the explicit BP notification from the relay nodes. In fact, between the instant when a BP message is issued by a relay node and the instant when the controller at the corresponding source node enforces some back pressure action, the transmission rate remains equal to the one that has caused the congestion and, in turn, layer 3 losses are possible at the receiver node due to the overflow of its buffer. Thus, a vulnerable period exists between the instant when congestion is detected at the relays (that is, when their queue size increases beyond \(Q_{sat}\)) and the instant when the layer 3 flow is effectively slowed down at the sources. During this vulnerable period, losses due to buffer overflow are likely to occur. For Deaf, losses are still present due to the exhaustion of the overall number of retransmissions per packet per hop (at both layer 2 and layer 3) and for this reason its \(P_{loss}\) monotonically decreases with an increasing \(N_{retx}\).

Fuse has the best \(P_{loss}\) performance and the reason for this is the combined effect of Griping and Deaf. In particular, the
explicit signaling of Griping allows for a prompter reaction to congestion events, which substantially decreases the probability that Fuse reaches the maximum number of retransmissions. Moreover, the vulnerable period issue is solved as, whenever the receiver’s queue is filled up, Def’s BP control is invoked and packets that overflow from this queue are subsequently retransmitted by the corresponding sender (due to the stopped acknowledgement flow).

For what concerns previously shown performance metrics, all of them stabilize for small values of $N_{\text{retx}}$ to the values shown in Figs. 2, 3, and 4.

Furthermore, when $N_{\text{retx}} = 0$, network congestion goes undetected and back pressure algorithms are never activated. In fact, in this case packets are transmitted but never retained in local queues, which are therefore filled up at a much slower pace. Thus, increasing $N_{\text{retx}}$ allows the fill-up of layer 3 queues and, in turn, the detection of congestion events: in fact, the rejection rate $R$ is positive for $N_{\text{retx}} > 0$. $N_{\text{retx}} = 0$ leads to poor performance on all metrics for all BP schemes.

Even $N_{\text{retx}} = 1$ leads to substantial throughput improvements in terms of $\lambda_0$. Setting $N_{\text{retx}} = 2$, which means up to 7 layer 2 and just 1 layer 3 retransmission, grants a throughput that is very close to the maximum achievable for the given network setup. The throughput increase is always accompanied by a corresponding increase in the delay performance, which also stabilizes for small values of $N_{\text{retx}}$. Counterintuitively, $N_{\text{retx}}$ remains stable for all BP devices when $N_{\text{retx}} > 1$.

Impact of varying $Q_{\text{thr}}$. Fig. 6 shows the impact of $Q_{\text{thr}}$ for $N_{\text{retx}} = 15$, $\lambda_0 = 20$ pkt/s, $Q_{\text{max}} = 31$ packets. Here, we only show the plot for $N_{\text{retx}}$, the other metrics are just discussed as their behavior is similar to what observed above. Static is represented as a horizontal line in the plot, since its behavior does not depend upon $Q_{\text{thr}}$.

As expected, $\lambda_0$ grows for increasing $Q_{\text{thr}}$, as a larger threshold lowers the probability of having buffer under-runs, thus leading to higher throughput efficiencies. For the average delay $D$, an increasing $Q_{\text{thr}}$ puts off the enforcement of back pressure control actions. Correspondingly, the number of packets stored in layer 3 queues and their average delay both increase. On the other hand, very low values of $Q_{\text{thr}}$ lead to long delays too as in this case back pressure control is almost always active, i.e., transmission rates are often slowed down and this implies longer L3D service times.

As expected, $R$ decreases monotonically with $Q_{\text{thr}}$ for all BP schemes as the rate of back pressure control actions is lowered for increasing values of $Q_{\text{thr}}$.

The behavior of $P_{\text{loss}}$ differs among the considered layer 3 devices. These results are just commented but not plotted for the sake of space. IdealBP shows no losses at all as its control action is deterministic and immediate. Lowering $Q_{\text{thr}}$ for Griping implies an earlier enforcement of back pressure policies, which leads to a larger buffer space to compensate for incoming packets during its vulnerable period. Thus, an increasing $Q_{\text{thr}}$ implies larger buffer overflow probabilities (i.e., larger $P_{\text{loss}}$). Conversely, for Def, $P_{\text{loss}}$ decreases with increasing $Q_{\text{thr}}$, ranging between 0.2% and 0.01%. This is because using the Def device packet losses only occur whenever a source reaches the maximum number of allowed retransmissions for a datagram (see C2 above). This event for Def is more likely to occur when $Q_{\text{thr}}$ is small, because in this case BP congestion control is activated more often, which means that layer 2 packets are acknowledged less frequently and this leads to more layer 2 failures. Fuse has the best performance among all L3Ds, with a $P_{\text{loss}}$ that is smaller than $10^{-4}$ for $Q_{\text{thr}} \leq 26$, whereas its $P_{\text{loss}}$ converges to that of Def as $Q_{\text{thr}}$ approaches $Q_{\text{max}}$. This good performance is due to the combined effect of Griping and Def BP control.

Finally, from Fig. 6 we observe that IdealBP has the best overall performance, whereas Def has the worst. Griping performs very close to IdealBP for all values of $Q_{\text{thr}}$. Fuse performs very close to both IdealBP and Griping for $Q_{\text{thr}}$ smaller than $Q_{\text{max}}/2$, whereas as $Q_{\text{thr}}$ increases toward $Q_{\text{max}}$, the transmission overhead of Fuse converges to that of Def. This is representative of the fact that the overhead in the latter case is dominated by the retransmissions due to the stopped acknowledgement flow.

4. Back Pressure Congestion Control for CoAP

Current trends in IoT networking involve the use of Web services on constrained IoT devices. These entail the bi-directional exchange of messages according to a request/response paradigm, see Fig. 1, as per the REST architectural style [18]. The Constrained Application Protocol (CoAP) [1] defines a simple, efficient, and flexible protocol to allow REST architectures to scale down to smart objects, by preserving interoperability with HTTP [19].

In this section, we apply back pressure congestion control to bidirectional CoAP traffic. In this case, typical communication patterns amount to the transmission of CoAP control messages from the outside Internet network to the constrained IoT nodes and their subsequent CoAP responses. CoAP makes it possible for IoT resources to be accessible as Web services, and in particular makes them available on the Internet as HTTP Web services (through CoAP to/from HTTP mapping, see [19]).
Referring to the second case study of Fig. 1, CoAP requests are sent over the constrained network as IPv6 datagrams flowing from the 6LoWPAN border router down to the leaf nodes. Upon receiving these requests, leaf nodes reply with IPv6 datagrams carrying the corresponding CoAP responses; the latter datagrams flow from the leaf nodes to the border router.

Note that the congestion problem is only marginally handled by the CoAP specification, which recommends a fixed congestion window of 1 packet at the CoAP senders. However, this static window may result in underutilized transmission resources when the network has some residual transport capacity and is as well inefficient when even this small window value suffices to create congestion. Differently, our approach is to prevent network queues from overrunning and as well to avoid the injection by the border router of an excessive number of requests into the constrained network. The latter objective is accomplished at the border router through the rejection of requests into the constrained network. The former objective is achieved by the activation of BP congestion before the layer 3 queue is filled with packets and this leaves some room to accommodate the network queues is still valid measure of network congestion.

4.1. L3 Devices for Bidirectional Back Pressure

Differently from traditional BP, when bidirectional traffic is taken into account some additional mechanisms need to be added to the congestion control policies. In fact, BP should not slow down response traffic, because any dropped CoAP response would mean a network loss from the client’s perspective.

Taking into account the fact that every CoAP request solicits a CoAP response flowing along the reverse path, the length of the network queues is still a valid measure of network congestion. However, this measure alone is not entirely representative of the number of outstanding CoAP requests that are still waiting for a corresponding CoAP response message. Note that these responses may still cause buffer overruns as they are transmitted over the constrained network. Ideally, one would need to track the number of outstanding CoAP requests, so as to gauge the expected future load due to CoAP responses and shape the data traffic accordingly. However, such a task is generally too complex for resource constrained IoT devices. Aiming at a lightweight design, in the following we modify the L3Ds of Section 2.2 with the objective of pushing back CoAP requests based on the queue length metric alone. While suboptimal, this solution entails little changes on current CoAP stacks and incurs low communication overhead. The goal of this section is to check whether, in spite of its simplicity, our queue-length-based control can provide satisfactory performance and also check which are the most important parameters that have to be tuned for its successful utilization.

Hence, the L3D devices of Section 2.2 have been modified as follows:

**Static.** This device does not apply any congestion control algorithm and is unchanged with respect to that of Section 2.2.

**IdealBP.** This device only applies its queue-length-based differential BP to the CoAP request traffic flowing from the border router to the leaf IoT nodes (in coming CoAP requests).

**Griping.** This device emits its explicit back pressure messages only upon the reception of CoAP requests (inbound CoAP traffic), whereas congestion control is not applied to CoAP responses (outbound CoAP traffic).

**Def.** This device implements the backoff policy of Section 2.2 and refrains from transmitting layer 2 acknowledgements when the corresponding layer 3 datagrams are CoAP requests.

**Fuse.** This device extends the Fuse BP policy of Section 2.2 adding a further threshold \( Q_{\text{thr}2} \) such that \( Q_{\text{thr}} < Q_{\text{thr}2} < Q_{\text{max}} \). The Def BP policy is activated when the queue length grows beyond the new threshold \( Q_{\text{thr}2} \), whereas the behavior of the Griping BP policy remains unchanged. This second threshold allows the activation of BP congestion before the layer 3 queue is filled with packets and this leaves some room to accommodate the CoAP reverse traffic.3 As for Griping and Def, BP is only applied to CoAP requests.

For the border router, whenever its layer 3 queue becomes full, it rejects any further incoming CoAP request by issuing a 503 Service Unavailable error message.

4.2. System Parameters

**Offered request load.** \( \lambda_{\text{tx}} \) defines the rate at which each CoAP client, placed in the external Internet network, sends CoAP requests toward a specific CoAP server placed within the constrained IoT network. Note that a server corresponds to an IoT leaf node in our simulation scenario, see Fig. 1.

The definition of the remaining system parameters \( Q_{\text{thr}}, Q_{\text{max}}, N_{\text{max}} \) remains the same as that of Section 3.1, the new threshold \( Q_{\text{thr}2} \) has been set to \( Q_{\text{thr}} + 5 \) packets.

3In other terms, the difference \( Q_{\text{max}} - Q_{\text{thr}} \) is our best a priori estimate of the impact of the CoAP requests that will follow the CoAP requests that are currently admitted to the network.
of the 9 leaf nodes; the border router hosts a CoAP proxy, which
gogy of Fig. 1, where a CoAP server has been deployed on each
Simulations have been run over the topol-
sent by responses as well as layer 3 BP control messages, such as those
of layer 2 packets that are sent to carry CoAP requests and re-
successful end-to-end bidirectional exchange (from the border
router to a leaf node and back to the border router) of a sin-
Transmission Overhead. \( N_{tx} \) represents the average number of
layer 2 packets that are transmitted in the network for the successful end-to-end bidirectional exchange (from the border router to a leaf node and back to the border router) of a single CoAP request and response pair. This metric accounts for the layer 2 packets that are sent to carry CoAP requests and responses as well as layer 3 BP control messages, such as those sent by Gripping and Fuse for the explicit signaling of a congestion event.

4.3. Performance Metrics

To compare the performance of the proposed L3Ds, the following performance metrics have been considered.

Received response rate. \( \lambda_{tx} \) defines the average per server (running on a leaf node) rate of CoAP responses that are correctly received by the border router and is measured in correctly received CoAP responses per second per server.

Round trip-time. \( D \) defines the average lapse of time (seconds) spent at the border router waiting for a CoAP response to an accepted CoAP request.

Loss probability. \( P_{loss} \) defines the percentage of CoAP responses that are not received by the border router, although the corresponding CoAP requests have been accepted into the constrained network.

Rate of rejects. \( R \) defines the average per client rate of CoAP requests that are not accepted into the network by the border router (issuing an HTTP 503 status code, as per our discussion above) and is measured in terms of rejected CoAP requests per second per client.

Transmission Overhead. \( N_{tx} \) accepts CoAP requests from 9 CoAP clients placed in the external Internet network. Each CoAP client emits Non-Confirmable (NON)\(^4\) requests at a constant rate \( \lambda_{tx} \) toward a CoAP server running on a leaf node. CoAP requests and responses have a fixed layer 3 size of 12 bytes and 115 bytes, respectively, including 6LoWPAN/UDP headers. The duration for each simulation run is 500 seconds, and the queue size of all nodes is 31 packets. The simulation points on the following graphs have been obtained averaging over 180 independent simulation runs.

Impact of varying \( \lambda_{tx} \). As a first result, Fig. 7 shows \( \lambda_{tx} \) and \( D \) as a function of \( \lambda_{tx} \in \{1, \ldots, 15\} \) req/client/s. The remaining simulation parameters are \( Q_{max} = 31 \) packets, \( Q_{dur} = 20 \) packets, \( Q_{thr} = 25 \) packets and \( N_{req} = 15 \).

For Fig. 7, we note that the general behavior of all metrics is similar to that observed for unidirectional traffic, see Fig. 2. The main difference is that in this case \( \lambda_{sat} \) is nearly halved due to the presence of CoAP bidirectional exchanges, whereby two packets (CoAP request and response) must be handled by the network for each accepted CoAP request. In fact, although CoAP requests and responses differ in size, their cost in terms of overall time spent, including retransmissions, is nearly the same and this is due to the dominating effect of MAC layer tasks such as the time required to gain access to the channel, back off times, etc., which do not depend on the data frame size.

Also, we note that \( \lambda_{Static} \) equals \( \lambda_{Static} \) up to about \( \lambda_{Sat} = 5 \) req/client/s, beyond which the response rate starts decreasing to a floor of about 3 req/client/s. This behavior is due to the so called congestion collapse event, similar to that observed in the early days of the Internet (see \([4, 20]\)). The congestion collapse is caused by the border router accepting more requests than those that can be served by the network, which is given by \( \lambda_{Static} \). For the delay, we note that \( D \) grows until \( \lambda_{tx} \) stabilizes.

Notably, IdealBP, Gripping, Defa and Fuse are not subject to the congestion collapse of Static but their throughput performance stabilizes as soon as \( \lambda_{tx} \) increases beyond \( \lambda_{Sat} \). Moreover, their delay remains stable even with \( \lambda_{tx} \) larger than \( \lambda_{Sat} \). This occurs because the border router acts as a proxy by rejecting traffic as soon as its outbound queue toward the constrained network becomes full.

For the rejection rate \( R \), similarly to what observed for Fig. 3 (unidirectional flows), Static starts rejecting packets when \( \lambda_{Static} \) is approximately 10 req/client/s, which is about twice \( \lambda_{Sat} \). The remaining L3Ds react to an increasing \( \lambda_{tx} \) by rejecting packets as soon as the offered traffic increases beyond \( \lambda_{sat} \), with \( \lambda_{Sat} \) halved with respect to that of Fig. 3. As for the unidirectional traffic scenario, IdealBP shows no losses, \( P_{loss}^{Gripping} \) converges to about 0.5%, \( P_{loss}^{Defa} \) and \( P_{loss}^{Fuse} \) both stabilize around \( 10^{-5} \).

Overall, it is worth noting that Fuse obtains nearly the same throughput as Gripping but has the same \( P_{loss} \) performance as Defa. This is due to the combined effect of Gripping’s explicit

\(^{4}\) NON requests do not have application-layer retransmissions; we chose to use this kind of requests, since our objective is the layer 3 evaluation of the congestion control performance.
signaling, which effectively limits the send rate within the network, and the fact that all requests are deterministically rejected by the border router when its queue length increases beyond $Q_{thr}$, which helps preventing congestion events.

In Fig. 8 we look at the transmission overhead $N_{tx}$. As for the unidirectional traffic scenario, $IdealBP$ shows a smaller transmission overhead than the other schemes for $\lambda_{tx} \leq \lambda_{sat}$. $Static$ presents the highest $N_{tx}$, which increases for increasing $\lambda_{tx}$ until it hits a maximum, which occurs at around $2\cdot\lambda_{sat}$, when the corresponding $R$ starts increasing. The remaining back pressure schemes effectively limit the maximum overhead and in particular we note that $Defa$ performs quite well here, in contrast to its unsatisfactory overhead performance for unidirectional traffic. The reason for this is that in this case the border router is the only source of data traffic and sends its packets directly over the bottleneck link of the network. In this case, $Defa$'s exponential backoff mechanism effectively keeps the overhead at a small value. This is in contrast to what happens for the unidirectional scenario where: i) there are multiple sources competing for the channel (multiple leaf nodes), ii) the considered tree topology is such that these multiple sources all insist onto the same routers and the data traffic is ultimately conveyed to a single border router (from many nodes to one), leading to an increasing congestion status as the data gets closer to the border router. Thus, in the unidirectional upstream case these facts result in a much more congested network and $Defa$'s exponential backoff alone is ineffective.

The good performance of $Defa$ for bidirectional CoAP flows makes it suitable to add BP functionalities to current CoAP/6LoWPAN protocol stacks, without requiring the definition of further BP messages. In fact, in spite of its simplicity this scheme effectively avoids the network collapse and also leads to a reasonably small traffic overhead.

**Impact of varying $N_{tx}$.** Fig. 9 shows $P_{loss}$ by varying $N_{tx}$ in $[0, 1, \ldots, 15]$. The remaining simulation parameters are $\lambda_{tx} = 20$ req/client/s (the system is congested), $Q_{max} = 31$ packets, $Q_{thr} = 20$ packets and $Q_{thr/2} = 25$ packets.

As observed in Section 2, BP requires an adequate number of hop-by-hop retransmissions to work. In fact, $IdealBP$ obtains no substantial advantage over $Static$ when no retransmissions are allowed, whereas even a very small number of retransmissions ($N_{ret} \leq 3$ for the considered setup) is sufficient for it to effectively relieve network congestion (see the sudden drop of $P_{loss}$ as $N_{ret}$ grows).

$Griping$ and $Defa$ require as well an adequate number of retransmissions to effectively work. $P_{loss}$ drops quickly and then stabilizes to a floor of about 0.2%. $P_{loss}$ monotonically decreases for increasing $N_{ret}$ for both $Defa$ and $Fuse$, although the latter requires a higher number of retransmissions due to the delayed BP control implied by the new threshold $Q_{thr}$ vs. $Q_{thr}$.

As in the unidirectional scenario, a high number of retransmissions does not negatively impact $N_{tx}$, which remains stable for all devices with $N_{ret} > 3$.

**Impact of varying $Q_{max}$.** Memory requirements have a strong relevance for constrained devices. In particular, the available memory and its management limit the queue length in actual implementations, e.g., see [17].

Figs. 10 and 11 show $\lambda_{tx}$ and $P_{loss}$ by varying $Q_{max}$ in $[3, 6, \ldots, 60]$. The remaining simulation parameters are: $Q_{thr} = \lceil (2/3)Q_{max} \rceil$ packets, $Q_{thr/2} = Q_{thr} + 5$ packets, $N_{ret} = 15$ and $\lambda_{tx} = 20$ req/client/s.

The throughput $\lambda_{tx}$ of $Static$ remains stable around 3 res/client/s and is only marginally affected by $Q_{max}$. For $IdealBP$, $Griping$ and $Fuse$, $\lambda_{tx}$ converges to about 5.2 res/client/s for $Q_{max} \geq 15$ packets. Thus, besides improving reliability, BP control also makes it possible to roughly double the throughput performance. We also observe that $Defa$ has a throughput performance that is roughly from 5 to 10% worse than that of the other BP schemes. The reason for this is inherent in how $Defa$ reacts to congestion events. In fact, $Defa$ detects congestion by stopping the transmission of the layer 2 acknowledgments associated with layer 3 CoAP requests. This has the twofold effect of slowing down the send rate of CoAP requests, while occupying the channel with their retransmissions. However, these retransmissions prevent the senders from exploiting the channel for other useful traffic such as the transmission of...
CoAP responses, whose correct delivery would contribute to a higher throughput performance. The performance gap between *Fuse* and the other BP schemes decreases for increasing $Q_{\text{max}}$, as $Q_{\text{thr}}$ also increases, leading to a less frequent activation of BP control policies and of the just discussed inefficiencies in terms of channel utilization (waste of channel resources).

For $P_{\text{loss}}$ from Fig. 11 we see that *Static* is unaffected by $Q_{\text{max}}$, whereas the reliability performance improves for all other L3Ds for increasing $Q_{\text{max}}$. This occurs because a larger $Q_{\text{max}}$ implies that network queues have more room to absorb traffic bursts and this makes buffer overruns less likely to occur. We also observe that *Deaf* has a slightly smaller $P_{\text{loss}}$ than *Fuse* as its smaller BP threshold $Q_{\text{thr}} < Q_{\text{thr}2}$ implies a prompter reaction to congestion events.

4.5. Results for Bidirectional CoAP Traffic in the Presence of Asymmetric Topologies and Cross-Traffic

**Asymmetric node deployment.** To assess the validity of the discussed congestion control techniques for asymmetric node deployments, the reference topology of Fig. 1 has been modified by moving 2 nodes from subnet S1 to subnet S3. The new node count for each subnet is as follows: S1 has a single node, S2 has three nodes, and S3 has five nodes.

Fig. 12(a) shows the obtained $P_{\text{loss}}$ for each of the subnets for the symmetric topology of Fig. 1, whereas Fig. 12(b) shows the performance metric obtained for the asymmetric topology discussed above. The simulation parameters are $\lambda_{\text{tx}} = 15 \text{ req/client/s}$, $Q_{\text{max}} = 31 \text{ packets}$, $Q_{\text{thr}} = 20 \text{ packets}$, $Q_{\text{thr}2} = 25 \text{ packets}$, and $N_{\text{tx}} = 15$.

As expected, the reliability performance for asymmetric topologies is slightly impacted, especially for those subnets containing a larger number of nodes (in our example scenario S2 and especially S3). For medium size subnets, such as S2, the performance of *Deaf* is close to that of *Fuse* and both achieve small error rates. However, as the number of nodes increases (S3) the *Fuse* device is to be preferred due to its higher robustness: here the addition of *Griping* ’s explicit BP messages leads to extra benefits in terms of $P_{\text{loss}}$. The remaining performance metrics $\lambda_{\text{tx}}$, $R$, $D$, and $N_{\text{tx}}$ are almost unaffected.

**Layer 3 devices for cross-traffic.** In what follows, we study the impact of cross-traffic due to the simultaneous presence of multiple border routers. Note that this is an unlikely scenario for 6LoWPAN networks adopting RPL [15], as in this standard multiple and distinct RPL sub-trees will be created for each of the border routers so as to minimize the interference among them. Nevertheless, some residual interference is still possible in practice and in the remainder we evaluate its impact.

For the networking scenario we still consider the network topology of Fig. 1, where a first border router BR1 (node BR in the figure) sends CoAP requests to nodes 1, 5, 6, 7 and 8; the CoAP request rate from BR1 is indicated as $\lambda_{\text{tx}}^1$. In addition, node 9 has been replaced with a second border router, BR2, which sends CoAP requests to nodes 2, 3 and 4; the CoAP request rate from BR2 is indicated as $\lambda_{\text{tx}}^2$. CoAP clients and servers are configured as specified in Section 4.4. Further, we define three groups of nodes as follows: $G_1 = \{1, 5, 6\}$, $G_2 = \{7, 8\}$ and $G_3 = \{2, 3, 4\}$ where G1 and G2 contain the nodes that send their CoAP responses to BR1, whereas the nodes in G3 send CoAP responses to BR2.

For this scenario, routers R1 and R4 deal with CoAP requests flowing in opposite directions and the devices defined in Section 4.1 deliver poor performance in this case. Next, we redefine our practical layer 3 BP devices to effectively handle this situation.

**Static.** This device remains unchanged with respect to that of Section 2.2.

**Griping.** This device follows the same rules explained in Section 2.2 and further refined in Section 4.1. Thus, flow control is only applied to CoAP requests, following the algorithm of Section 2.2. However, in order to effectively address the presence of CoAP requests coming from different border routers, the *Griping* device at the receiver additionally implements the following rule. Whenever, upon the reception of a CoAP request, the local layer 3 queue occupancy $Q$ is larger than the pre-defined threshold $Q_{\text{thr}}$, the *Griping* device at the receiver acts as follows: 1) it calculates the fraction $\eta \in [0, 1]$ of packets in the local layer 3 queue that are directed toward the same next hop (same layer 3 address) as that of the current CoAP request, and 2) an explicit BP message is sent back to the originator of the CoAP request only if $\eta$ is larger than or equal to a pre-defined threshold $\eta_{\text{thr}}$. $\eta_{\text{thr}} = 0.15$ has been selected for the results in this section.

**Remark.** This additional rule is implemented at the *Griping* receiver to avoid that one or more CoAP flows directed toward a certain set of border routers will take all the available bandwidth, whereas the remaining flows will starve. This rule effectively achieves this, as those flows which are given limited bandwidth (roughly less than $\eta_{\text{thr}}$) are not further slowed down through explicit BP messages. Flow control, through the reduction of the transmission rate is instead implemented for those flows that get the largest portion of the link capacity.
Deaf. This device follows the same rules explained in Section 2.2 and further refined in Section 4.1. However, the following probabilistic rule at the receiver has been implemented to handle cross-traffic to multiple WSN sinks. The acknowledgement flow at layer 2 is stopped upon the reception of a CoAP request when the following conditions are verified: C1) the layer 3 queue occupancy $Q$ is larger than $Q_{thr}$, as defined in Section 4.1, C2) if C1 is verified, the transmission of the returning layer 2 ACK for the layer 2 packet associated with the current CoAP request is canceled with probability $p = \eta^{1/N_{req}}$, where $N_{req}$ is the number of retransmissions allowed at layer 2 and $\eta$ is the fraction of packets in the local layer 3 queue that are directed toward the same next hop (same layer 3 address) as that of the just received CoAP request.

Remark. With this new probabilistic rule, whenever a CoAP request flow directed toward a certain layer 3 destination takes a fraction $\eta \in [0, 1]$ of the local layer 3 queue, the probability that $N_{req}$ subsequent layer 2 ACKs for a CoAP request belonging to this flow are denied is $\eta = \eta^{1/N_{req}}$. This implies that the probability that the layer 2 flow is stopped for a certain CoAP request is $\eta$. Note that a stopped layer 2 ACK flow is perceived by the Deaf transmitter as an implicit BP indication. The rationale behind this is that flows should be penalized according to the fraction of link capacity assigned to them.

Fuse. Since it is defined as the combination of Deaf and Gripping, its behavior arises from the combination of the previously redefined devices.

Results for CoAP cross-traffic. Fig. 13 shows $P_{loss}$ in the presence of CoAP cross-traffic from BR2 for the following simulation parameters: $\lambda_{tx}^1 = 15$ req/client/s, $\lambda_{tx}^2 \in [5, 15]$ req/client/s, $Q_{max} = 31$ packets, $Q_{thr} = 20$ packets, $Q_{thr2} = 25$ packets and $N_{req} = 15$. The case where the cross-traffic rate is moderate, i.e., $\lambda_{tx}^2 \leq 5$ req/client/s (see Fig. 13(a)), is reasonable in actual 6LoWPAN/RPL networks where this type of traffic may be due to, e.g., the presence of peer-to-peer traffic. The extreme case $\lambda_{tx}^2 = \lambda_{tx}^1$ (Fig. 13(b)) is instead undesirable and can be avoided by maintaining distinct RPL trees.

In general, $P_{loss}$ is impacted with respect to that of the previous networking scenarios. From Fig. 13(a), we note that the Deaf and Fuse devices can guarantee an error rate smaller than 1%, which may be adequate in most practical cases. Notably, Fuse shows robustness with respect to an increasing $\lambda_{tx}^1$, being almost unaffected as $\lambda_{tx}^1$ goes from 5 to 15 req/client/s, see Fig. 13(b).

5. Conclusions

In this paper, we have proposed several congestion control techniques for IoT networks. These algorithms have been conceived to add congestion control capabilities to IETF CoAP/6LoWPAN-based protocol stacks and their benefits have been quantified for unidirectional as well as bidirectional CoAP flows, which are typical of Web architectures. Overall, our schemes lend themselves to implementation in existing solutions and perform satisfactorily under a wide range of parameter settings. Among them, Deaf looks particularly suitable for implementation purposes as this scheme requires minimal modifications to existing protocol stacks and does not involve the definition of additional messages. Also, in spite of its simplicity, the throughput performance of this algorithm is only 5 – 10% smaller than that of the best performing scheme, while leading to excellent results in terms of reliability. The best performing scheme is Fuse, which combines Deaf with explicit control messages to detect congestion events. This scheme performs well for all the considered metrics and appears as a valid solution for unidirectional and bidirectional traffic. Our future work is devoted to reducing the impact of congestion control on the end-to-end delay, to further evaluating the impact of different topologies on the performance of BP congestion control algorithms, and to the investigation of mechanisms for the self-tuning of key system parameters (i.e., $Q_{thr}$ and $Q_{thr2}$), so that the proposed congestion control algorithms can automatically
adapt to any topology, network and traffic flow model. Analytical and experimental evaluations are also interesting future research directions.

**Acknowledgments**

The authors would like to thank Nicola Bui for some useful early discussions on this topic, which motivated us to conduct this study. The research leading to these results has received funding from the Seventh Framework Programme under grant agreement n. 257521 [21].

**References**


[14] Iperf Project. URL http://src.net/project/iperf/


[16] NS-3 Project. URL http://www.nsnam.org/


Angelo P. Castellani was born in Rome, Italy, on July 7th, 1981. He received the Ph.D. in Information Engineering in 2012 from the University of Padova, Italy, where he is currently a postdoctoral research fellow. He obtained the BE and ME, summa cum laude, both from the University of Rome “Sapienza” (Italy) in 2004 and 2006, respectively. His research interests are in the fields of sensor networks and the Internet of Things (IoT), including protocol design and experimentation. He is active in standardization work within the IETF and is currently involved in the EU-funded IoT-A project, both activities are centered around IoT networking.
research.

Michele Rossi received the Laurea degree in Electrical Engineering and the Ph.D. degree in Information Engineering from the University of Ferrara in 2000 and 2004, respectively. From March 2000 to October 2005 he has been a Research Fellow at the Department of Engineering of the University of Ferrara. During 2003 he was on leave at the Center for Wireless Communications (CWC) at the University of California San Diego (UCSD), where he performed research on wireless sensor networks. In November 2005 he joined the Department of Information Engineering of the University of Padova, Italy, where he is an Assistant Professor. Dr. Rossi is currently involved in the EU-funded IoT-A and SWAP projects, both on wireless sensor networking. His research interests are centered around the dissemination of data in distributed ad hoc and wireless sensor networks, including protocol design through stochastic optimization, integrated MAC/routing schemes, compression techniques for wireless sensor networks and cooperative routing policies for ad hoc networks. Dr. Rossi currently serves on the Editorial Board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Michele Zorzi was born in Venice, Italy, on December 6th, 1966. He received the Laurea and the PhD degrees in Electrical Engineering from the University of Padova, Italy, in 1990 and 1994, respectively. During the Academic Year 1992/93, he was on leave at the University of California, San Diego (UCSD) as a visiting PhD student, working on multiple access in mobile radio networks. In 1993, he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, in 1998 he joined the School of Engineering of the University of Ferrara, Italy, where he became a Professor in 2000. Since November 2003, he has been on the faculty at the Information Engineering Department of the University of Padova. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks, energy constrained communications protocols, broadband wireless access and underwater acoustic communications and networking. Dr. Zorzi was the Editor-In-Chief of the IEEE WIRELESS COMMUNICATIONS MAGAZINE from 2003 to 2005 and the Editor-In-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2011, and currently serves on the Editorial Board of the WILEY JOURNAL OF WIRELESS COMMUNICATIONS AND MOBILE COMPUTING. He was also guest editor for special issues in the IEEE PERSONAL COMMUNICATIONS MAGAZINE (Energy Management in Personal Communications Systems) IEEE WIRELESS COMMUNICATIONS MAGAZINE (Cognitive Wireless Networks) and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Multi-media Network Radios, and Underwater Wireless Communications Networks). He served as a Member-at-large of the Board of Governors of the IEEE Communications Society from 2009 to 2011, and is a Fellow of the IEEE.