Token Bucket Based Traffic Shaping and Monitoring for WLAN-based Control Systems

Kiran Mathews, Christopher Kramer, Reinhard Gotzhein
Networked Systems Group
University of Kaiserslautern, Germany
Email: {mathews, kramer, gotzhein}@cs.uni-kl.de

***Abstract*—In industrial automation, there is strong interest in using wireless technologies to run production environments, and in particular to build networked control systems. To operate such networks effectively, sufficient control over the communication system is needed, regarding, e.g., bandwidth usage, frame latency, and frame loss. In this paper, we present an approach for traffic shaping that applies a variant of the token bucket algorithm, and an approach for traffic monitoring in order to determine network overload. In particular, we introduce and apply a metric called *unusable wasted bandwidth ratio*, which is derived from the number of *unusable wasted tokens* to detect network overload locally. We have implemented both approaches on a commercial platform with WLAN adapters, and have conducted real experiments to assess their performance.**

I. Introduction

To monitor and control production processes in industrial environments, there is a strong interest in building on wireless communication technologies. With these technologies, it is possible to connect sensors, actuators, and controllers in a flexible and economic way, and even to support mobile objects. These systems are also referred to as wireless networked control systems (WNCS), and are subject of extensive research (see, e.g., [1]).

Control applications of WNCS typically have Quality-of-Service (QoS) requirements regarding bandwidth, transmission latencies, and frame loss rates. To provide sufficient guarantees, two basic approaches can be applied, both establishing control over the communication system. With open communication loop control, there is no dynamic feedback from the communication network. Therefore, to provide guarantees, it is mandatory that the network behavior is highly predictable, which, in turn, requires deterministic communication protocols. A possible approach is to use TDMA (Time Division Multiple Access) with exclusive reservations, which is implemented by special-purpose technologies such as WirelessHART [2] or ISA 100.11a [3]. Closed communication loop control relies on dynamic feedback from the communication network, and adapts network usage to the observed network status in order to sustain stable QoS operation. This requires suitable approaches for traffic monitoring, i.e., sensing of the current network load, and traffic shaping, e.g., delaying or discarding of frames.

978-1-5386-3531-5/17/$31.00 © 2017 IEEE

In our current research, we are exploring, developing, and validating the technical foundations for WNCS to match stochastic real-time requirements based on IEEE 802.11 (WLAN). From a research perspective, the use of this generally available and low-cost digital radio technology is a particular challenge, since it has not been developed for WNCS. In this paper, we present traffic control functionality that can be realized with off-the-shelf WLAN adapters, on top of standard drivers. For traffic shaping, we have devised and implemented a variant of the token bucket algorithm [4], which is specifically adapted for control systems in order to distribute the usable bandwidth while reducing frame collisions. For traffic monitoring, we introduce a new metric called *unusable wasted bandwidth ratio* which is derived from the number of *unusable wasted tokens* to detect network overload locally. Real experiments with and without foreign traffic provide evidence that, similar to *Channel Busy Time* (CBT), these measures can be used for traffic monitoring and network overload detection. Our prototype implementation on off-the-shelf hardware proves the applicability, even on hardware that does not provide channel information like CBT.

The paper is structured as follows: In Sect. II, we discuss related works. In Sect. III, we devise network control functionality for traffic shaping and traffic monitoring. In Sect. IV, we elaborate on the implementation of those functionalities on a commercial hardware platform with WLAN. Sect. V presents results of the experimental assessment of our algorithms, and Sect. VI draws conclusions and outlines future work.

II. Related work

The token bucket algorithm has been used in wireless networks for different purposes. In [5], it is used for traffic management in WiMAX networks. In [6] and [7], it is used for packet scheduling and admission control in IEEE 802.16. It is also used as scheduling algorithm for IEEE 802.11 based sensor networks [8] and for several other purposes. To the best of our knowledge, our approach is the first time that token bucket is used

to detect network overload, i.e. for traffic monitoring. Additionally, all previous uses of token bucket define the token size in bits or bytes, assuming a fixed transmission rate, whereas we define the token size in μs representing the medium occupation, thus taking WLAN's different transmission rates into account.

Conventionally, metrics such as CBT, derived from hardware registers, and *Medium Utilization Time*, calculated by capturing WiFi frames, were used to monitor the medium conditions [9]. In [10], CBT is used in a rate-adaptation algorithm, and in [11], it is used for admission control in mesh networks. In [9], CBT is used to estimate available bandwidth of WLAN links. CBT requires polling the channel survey details from the hardware registers. In order to access the hardware registers, most of the solutions (see [10], [11], [12]) modified the kernel driver. Nowadays, few WLAN adapters along with specific driver versions support it without modifications. *Medium Utilization Time* has the overhead of processing each captured frame. So, the former depends on the hardware support while the latter introduces processing overhead. The backpressure routing approach in [13] uses virtual queues that can be seen as another approach to detect network load.

III. Network control functionality

In this section, we introduce our token bucket method for bandwidth management, traffic shaping, and traffic monitoring. Let $G = (V, E)$ be a wireless communication network, consisting of a set of nodes V and a set of links $E \subseteq V \times V$. The set V is the union of disjoint sets V^i and V^e, consisting of internal and external nodes, respectively. Internal nodes are the nodes applying the token bucket method and performing traffic shaping and monitoring. External nodes belong to foreign WLAN networks that share the medium with the internal nodes and produce unpredictable additional traffic.

A. Bandwidth reservation and traffic shaping

To operate IEEE 802.11 based WNCS networks effectively with respect to bandwidth usage, latency, and loss, we introduce functionality for bandwidth reservation and traffic shaping.

1) Bandwidth reservation: In our current design, bandwidth reservation is performed dynamically by a centralized bandwidth manager (CBM). Other than decentralized approaches, this solution gives accurate control over assigned bandwidth. In this paper, the term *bandwidth* denotes the maximum amount of time a node is allowed to use the medium for transmission or *maximum medium occupancy duration*. To operate the network effectively, we only use a certain ratio $bwr^G_{avail} \in [0, 1]$ of the total bandwidth of the wireless network G, e.g. 20%. This ratio is to be chosen such that medium contention is kept sufficiently small, and depends, among other things, on the influence of external traffic. Similarly, $bwr^v_{avail} \in [0, 1]$ denotes available local bandwidth for node v.

We assume that each node $v \in V^i$ has a relative bandwidth requirement $bwr^v_{required} \in [0, 1]$, which it requests from the CBM. Based on bwr^G_{avail} and the currently assigned relative bandwidth bwr^G_{assgn}, CBM decides whether the request can be granted. If so, CBM returns $bwr^v_{required}$ as bwr^v_{assgn}, otherwise, it returns 0. More specifically, the following constraint must be enforced by CBM:

$$\sum_{v \in V^i} bwr^v_{assgn} = bwr^G_{assgn} \leq bwr^G_{avail}$$

The protocol for bandwidth reservation is straightforward and therefore not detailed further. In a multihop network, it builds on a protocol for automatic link detection, providing topology information used for routing purposes.

2) Traffic shaping: The purpose of traffic shaping is to monitor whether local bandwidth usage bwr^v_{used} of a node v conforms to the assigned relative bandwidth bwr^v_{assgn}, to delay or drop frames exceeding it, and to distribute the traffic uniformly in order to avoid global peaks and therefore high contention situations.

For traffic shaping, we have devised a modified token bucket algorithm (see Fig. 1). As in the original token bucket algorithm, tokens are placed into the bucket with a constant refill interval. In the original token bucket algorithm, each token represents the right to send a certain number of bytes. On WLAN, frames can be sent at many different transmissions rates, thus the time to transmit a certain number of bytes can vary significantly. To take this into account, we use time slices of equal length as tokens. Each time slice represents an access right to the medium for a short period of time. To schedule a frame for transmission, tokens sufficient to cover the frame transmission time including overhead have to be collected in the bucket. These tokens are removed from the bucket when the frame is moved from the arrival queue to the send queue. When the bucket is already filled, further refill attempts lead to an overflow, i.e., tokens are lost.

The token bucket of a node $v \in V^i$ is characterized by the bandwidth profile, bp^v. To determine bp^v, CBM needs the maximum payload size, in addition to $bwr^v_{required}$. The bandwidth profile for a node v is a tuple bp^v:

$bp^v = (d^v_{txMax}, d^v_{refill}, bwr^v_{assgn})$, where

- $d^v_{txMax} \in \mathbb{N}_0$ [μs] is the *maximum medium occupancy duration* for a node *v* or the maximum burst size. To minimize possible bursts and to guarantee that all frames can be sent, we set it to the duration required to transmit the largest possible frame at the lowest transmission rate:

$$d^v_{txMax} = \frac{b_{payloadMax} + b_{MLO}}{r_{txMin}} + d_{PLO} \quad (1)$$

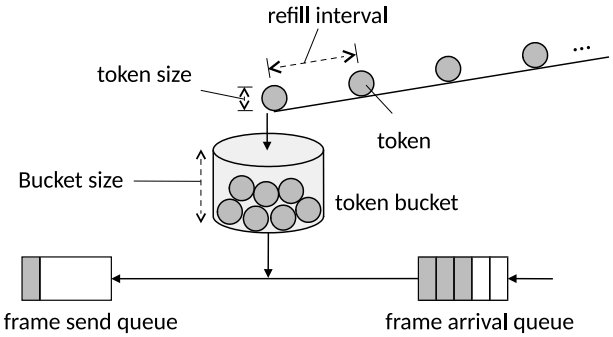

Fig. 1. Bandwidth management with token bucket

Here, b_{MLO} denotes the MAC layer overhead in bytes, and d_{PLO} is the average of PHY layer overhead in micro seconds, covering interframe spacing, average backoff interval, and PHY preamble.

- $d^v_{refill} \in \mathbb{N}_0$ [μs] is the refill interval in which the medium occupancy duration collected in the bucket of node v is incremented. As heuristics, the refill interval should be small enough to avoid unnecessary delays, which on average is half a token per frame.
- $bwr^v_{assgn} \in [0,1] : bwr^v_{requested}$, if granted; 0 otherwise. It is the bandwidth assigned to node v relative to the network bandwidth.

Compared to the generic token bucket [4], d^v_{txMax} and d^v_{refill} denote the bucket size and the token refill interval, respectively.

From bp^v, further bucket parameters of node v can be determined as follows:

- d^v_{token} [μs] denotes the size of a token.

$$d^v_{token} = d^v_{refill} \cdot bwr^v_{assgn} \tag{2}$$

- $n^v_{fillings}$ denotes the total number of tokens required for completely filling the bucket

$$n^v_{fillings} = \left\lceil \frac{d^v_{txMax}}{d^v_{token}} \right\rceil \tag{3}$$

- d^v_{bucket} [μs] denotes the minimum time required to completely fill an empty bucket.

$$d^v_{bucket} = n^v_{fillings} \cdot d^v_{refill} \tag{4}$$

Next, we discuss an example for calculating the bucket parameters based on the application requirement. Consider a periodic application *appX* running on node v with following values:

$$\begin{aligned} d^v_{refill} &= 1,000 \mu s \\ b^{appX}_{payload} &= 300\,B \\ d^{appX}_{period} &= 10^7\,\mu s \\ n^{appX}_{msg} &= 5 \end{aligned}$$

where,

- $b^{appX}_{payload} \in \mathbb{N}_0$ is the payload size in bytes.
- $d^{appX}_{period} \in \mathbb{N}_0$ [μs] is the application period length.
- $n^{appX}_{msg} \in \mathbb{N}_0$ is the number of messages to send inside a period for a periodic application.

Using equation Eqn. (1) and assuming a transmission rate of 1 Mbps, we get

$$d^v_{txMax} = \frac{300B + 52B}{1Mbps} + 288.5\,\mu s = 3,104.5\,\mu s \tag{5}$$

The remaining bucket parameters are set as follows:

$$\begin{aligned} bwr^v_{assgn} &= \frac{d^v_{txMax} \cdot n^{appX}_{msg}}{d^{appX}_{period}} \\ &= \frac{3,104.5\,\mu s \cdot 5}{10^7\,\mu s} \approx\ 0.15\% \end{aligned}$$

Using equations Eqn. (2), Eqn. (3), and Eqn. (4), we get

$$\begin{aligned} d^v_{token} &= 1,000\,\mu s \cdot 0.15\% = 1.5\,\mu s \\ n^v_{fillings} &= \left\lceil \frac{3,104.5\,\mu s}{1.5\,\mu s} \right\rceil = 2,070 \\ d^v_{bucket} &= 2,070 \cdot 1,000\,\mu s = 2,070,000\,\mu s \end{aligned}$$

Fig. 1 illustrates the operation of the modified token bucket algorithm. Frames to be sent are placed into the *frame arrival queue*. If the bucket is sufficiently filled and the *frame send queue* is empty, the next frame can be scheduled for transmission, by placing it into the send queue and consuming the required number of tokens from the bucket. If the bucket does not contain enough tokens or the *frame send queue* still contains a frame, the next frame is delayed until both conditions hold. Tokens are placed into the bucket as determined by the refill interval.

Note that by limiting the size of the *frame send queue* to 1 and by keeping the bucket size small, transmission bursts are limited. When a frame is scheduled for transmission, the next frame is delayed. Thereby, the used bandwidth ratio bwr^v_{used} of a node v is limited to bwr^v_{assgn}.

B. Traffic Monitoring

Traffic monitoring is performed by monitoring the token bucket. When the assigned bandwidth is not exploited, the token bucket will have overflows, resulting in *wasted tokens*. This can have two reasons:

1) The assigned bandwidth exceeds the used bandwidth, i.e. $bwr^v_{assgn} \geq bwr^v_{used}$, e.g. because frames are created only sporadically or have a variable size. The tokens wasted here are called *usable wasted tokens*, because the application could have used them, it just did not need to.
2) Network is overloaded, i.e. $bwr^v_{assgn} \geq bwr^v_{avail}$. Because of high contention, frames stay longer in the *sending queue*, thus frames in the *frame arrival queue* cannot use the refilled tokens which leads to overflows. The tokens wasted this way

are called *unusable wasted tokens*, as too much medium contention prohibits their use.

In order to monitor usable and unusable wasted tokens, we introduce two counters:

1) The counter $n^v_{WTusable}(t)$ denotes the usable tokens wasted until time t. It is incremented each time a token is generated, if the token bucket is filled, and both frame send queue and frame arrival queue are empty.
2) The counter $n^v_{WTunusable}(t)$ denotes the unusable tokens wasted until time t. It is incremented each time a token is generated, if the token bucket is filled, and frame send queue or frame arrival queue are not empty.

With these counters and based on monitored transmission activities, the following characteristic parameters can be derived:

- $d^v_{tx}(t)$: Total amount of time a node v used the medium for transmitting frames until time t from the start of an experiment.

$$d^v_{tx}(t) = \sum_{\forall f \in F^v(t)} \left(\frac{b^f_{payload} + b_{MLO}}{r^f_{tx}} + d_{PLO} \right) \quad (6)$$

  Here $F^v(t)$ denotes frames sent by node v until time t, and $b^f_{payload}$ denotes the payload size of the individual frame and r^f_{tx} the rate at which it is sent.
- bwr^v_{used}: Ratio of used assigned bandwidth by a node v until time t since the node start time. It is calculated as,

$$bwr^v_{used}(t) = \frac{d^v_{tx}(t)}{(t - t_{start}) \cdot bwr^v_{assgn}} \quad (7)$$

- $bwr^v_{WBwUnusable}$: Ratio of unusable wasted assigned bandwidth of node v at time t since the start.

$$bwr^v_{WBwUnusable}(t) = \frac{n^v_{WTunusable}(t) \cdot d^v_{token}}{(t - t_{start}) \cdot bwr^v_{assgn}} \quad (8)$$

- $bwr^v_{WBwUsable}(t)$: Ratio of usable wasted assigned bandwidth of node v at time t since the start.

$$bwr^v_{WBwUsable}(t) = \frac{n^v_{WTusable}(t) \cdot d^v_{token}}{(t - t_{start}) \cdot bwr^v_{assgn}} \quad (9)$$

- $bwr^v_{WBw}(t)$: Ratio of wasted assigned bandwidth by the node v at time t since the start.

$$\begin{aligned} &bwr^v_{WBw}(t) \\ &\quad = bwr^v_{WBwUnusable}(t) + bwr^v_{WBwUsable}(t) \\ &\quad = 1 - bwr^v_{used}(t) \end{aligned} \quad (10)$$

- $d^v_{CBT}(t)$: The value *Channel Busy Time* (CBT) from the hardware register of node v at time t.
- $tr^v_{CBT}(t)$: Ratio of the time the medium was observed busy by node v until time t since the start.

$$tr^v_{CBT}(t) = \frac{d^v_{CBT}(t) - d^v_{CBT}(t_{start})}{t - t_{start}} \quad (11)$$

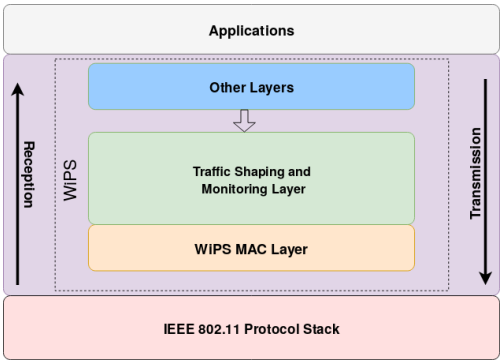

Fig. 2. Architecture of WiPS framework

- $d^v_{MES}(t)$: For node v, $d^v_{CBT}(t)$ includes the duration where a node sensed energy on the medium plus the time taken for transmitting the frames [14], [15]. $d^v_{MES}(t)$ defines the amount of time the node sensed medium energy excluding its transmission duration until time t from the start.

$$d^v_{MES}(t) = (d^v_{CBT}(t) - d^v_{CBT}(t_{start})) - d^v_{tx}(t) \quad (12)$$

- $tr^v_{MES}(t)$: The ratio of the time, the node sensed medium energy excluding its transmission duration until time t from the start.

$$tr^v_{MES}(t) = \frac{d^v_{MES}(t)}{t - t_{start}} \quad (13)$$

We propose $bwr^v_{WBwUnusable}(t)$ as a metric to detect network overload and experimentally assess it against the well-known metric $tr^v_{CBT}(t)$ in Sect. V.

IV. Implementation using IEEE 802.11(WLAN)

In this section, we give a survey of our framework, testbed and hardware platforms used for experiments. Experiments were conducted in two testbeds: The *static testbed* is located inside a university building where there is interference from other wireless networks. The *mobile testbed* is used for experiments in the absence of external traffic in the basement of the building. Each testbed consists of six Raspberry Pi nodes running Arch Linux with Linux kernel version 4.6.5-v7+. A LogiLink WL0084B USB WiFi adapter with Ralink Chipset using rt2800 drivers is attached to each node. In the mobile testbed, all nodes were placed very close to each other, whereas in the static testbed, the nodes were placed in different rooms. Nevertheless, both testbeds are single-hop, i.e. each node can directly communicate with each other node.

To simplify the development of WiFi based protocols, experiments and applications, we developed a framework called WiPS (WiFiProtocolStack). The extensible layer architecture of WiPS is sketched in Fig. 2. It runs on Linux and supports different platforms and WiFi hardware by encapsulating hardware access using

libpcap [16]. The WiFi adapters are operated in monitor mode and the payload is pushed into the IEEE 802.11 protocol stack through libpcap along with Radiotap headers [17], which allow us to define the transmission rate for each frame. This is important because the number of tokens required to send a frame is determined based on its medium access time, which can only be calculated if the transmission rate is known. The IEEE 802.11 MAC-layer is not used for acknowledgements, addressing or retransmissions. Instead, the framework always pushes IEEE 802.11 broadcast frames into the IEEE 802.11 stack. An additional *WiPS MAC layer* is added on top of IEEE 802.11, providing support for addressing, broadcasts, acknowledgements, retransmissions, sequence numbers as well as duplicate filtering. This has the advantage that we have full control over these functionalities.

The *Traffic Shaping and Monitoring Layer* (TSML) performs the functionality of traffic shaping and monitoring. Traffic shaping includes the implementation of the redefined token bucket mechanism. It calculates the bandwidth required for frame transmissions, manages the bucket refilling process and reports wasted tokens for monitoring purpose. As mentioned in Sect. III-B, when the application requires higher network load than assigned, frames are queued in the *frame arrival queue*. The TSML only pushes one frame into the IEEE 802.11 protocol stack at a time, delaying other frames until it receives a successful kernel acknowledgment for the previous transmission. Monitoring contains the logging of transmission and reception details, and periodically polling the registers of the WiFi adapter to store the *Channel Survey Details*. It also keeps track of the wasted tokens and logs the number of *usable wasted tokens* and *unusable wasted tokens*. The logs produced by this layer are used in the assessment of the experiments.

V. Experimental assessment

In this section, we present the results of selected experiments conducted to analyze the quality of traffic shaping and traffic monitoring using metrics based on *unusable wasted tokens*. To evaluate the effects of external traffic, the experiments were run both on the mobile testbed without external traffic and on the static testbed with several external WiFi Access Points and devices nearby. In the experiments, an application is running on each node, sending broadcast frames of size 480 bytes[1] at a constant transmission rate of 1 Mbps. All experiments were conducted on the 2.4 GHz ISM band.

In the first experiment, we compare the *unusable wasted tokens* and *unusable wasted bandwidth ratio* metrics with *channel busy time ratio* and *medium energy sensed ratio*. In this experiment, the *frame arrival queue* is always filled, i.e. whenever the bucket is full, there will be a frame waiting to consume the tokens. This means that the assigned bandwidth is used if possible, i.e. $bwr^v_{used} \geq bwr^v_{assgn}$. In other words, the experiments are designed in such a way to avoid the occurrence of *usable wasted tokens*. For different bwr^v_{assgn} (i.e. network load bwr^G_{assgn}), the metrics mentioned above are compared.

[1]excluding the IEEE 802.11 MAC overhead

BW profile	d^v_{txMax}	d^v_{refill}	bwr^v_{assgn}
$bp_{0.5}$	4545 μs	100 μs	0.5 %
$bp_{1.0}$	4545 μs	100 μs	1 %

TABLE I
Bandwidth profiles used for the experiments

Tab. I shows the bandwidth profiles and their respective bucket parameters used in the experiment. As we first want to test the sensitivity of our metrics, the network load is configured really low. Tab. II shows the results of an experiment conducted in the absence of external traffic with $bp_{1.0}$, i.e. 1% bandwidth assigned to each internal node. No tokens were wasted, so all nodes had enough bandwidth available to satisfy the assigned bandwidth, i.e. each node satisfies the condition $bwr^v_{assgn} \leq bwr^v_{avail}$.

Node	$n^{src}_{WTunusable}(t)$	$bwr^{src}_{WBwUnusable}(t)$	$tr^{src}_{CBT}(t)$	$tr^{src}_{MES}(t)$
38	0	0.00 %	5.70 %	4.79 %
39	0	0.00 %	7.80 %	6.88 %
40	0	0.00 %	8.14 %	7.21 %
41	0	0.00 %	5.80 %	4.87 %
42	0	0.00 %	5.74 %	4.81 %
43	0	0.00 %	6.36 %	5.44 %

TABLE II
Experiment without external traffic using bandwidth profile $bp_{1.0}$ on 6 nodes resulting in 6 % internal network load

Node	$n^{src}_{WTunusable}(t)$	$bwr^{src}_{WBwUnusable}(t)$	tr^{src}_{CBT}	$tr^{src}_{MES}(t)$
7	293,358	0.89 %	24.80 %	23.83 %
8	232,686	0.71 %	19.87 %	18.89 %
22	13,686	0.04 %	16.90 %	15.91 %
23	249,952	0.76 %	24.57 %	23.59 %
28	42,648	0.13 %	17.90 %	16.92 %
29	11,152	0.03 %	14.32 %	13.32 %

TABLE III
Experiment with external traffic using bandwidth profile $bp_{1.0}$ on 6 nodes resulting in 6 % internal network load

Tab. III shows the results of the experiment conducted in the presence of external traffic with the same bandwidth profile. The d^v_{token} (calculated using Eqn. (2)) for the nodes while using the bandwidth profile $bp_{1.0}$ is 1 μs. As shown in Tab. III, node 7 lost 293,358 *unusable wasted tokens*, i.e. it was not able to use 293,358 μs $(293,358 \cdot 1\mu s)$ of its assigned bandwidth time. This

Node	$n^{src}_{WTunusable}(t)$	$bwr^{src}_{WBwUnusable}(t)$	tr^{src}_{CBT}	$tr^{src}_{MES}(t)$
7	8,164	0.02 %	14.80 %	14.31 %
8	4,726	0.01 %	10.25 %	9.76 %
22	0	0.00 %	11.86 %	11.37 %
23	8,527	0.03 %	13.73 %	13.24 %
28	0	0.00 %	12.74 %	12.24 %
29	0	0.00 %	8.39 %	7.89 %

TABLE IV
EXPERIMENT WITH EXTERNAL TRAFFIC USING BANDWIDTH PROFILE $bp_{0.5}$ ON 6 NODES RESULTING IN 3 % INTERNAL NETWORK LOAD

experiment was conducted for 55 minutes and nodes were assigned a bandwidth of 1%, i.e. 1% of 55 minutes is 33,000,000 μs. Out of allocated 33,000,000 μs, node 7 was not able to use 293,358 μs due to delays caused by medium contention. Thus, as stated in Tab. III, node 7 lost 0.89 % of its assigned bandwidth.

Further looking into the Tab. III, nodes 7, 8 and 23 lost a considerably higher number of tokens compared to the others. Also, on the same nodes, the WiFi adapter sensed a higher amount of medium energy, i.e. tr^v_{CBT} and tr^v_{MES}. This shows that the node which sensed a comparatively high duration of medium energy lost more tokens.

Next, we reduce the bandwidth to half and analyze the same metrics. Tab. IV shows the results of the experiment in the presence of external traffic, with 0.5 % assigned bandwidth to each node (bandwidth profile $bp_{0.5}$). The results show that some nodes had enough bandwidth to run the application while the nodes that lost most tokens in the previous experiment, lost a smaller number of tokens. Similarly, the nodes which lost more tokens sensed a comparatively longer period of medium energy. This shows that operating the network at bandwidth profile $bp_{1.0}$ generates some contention, which is reduced by assigning the reduced bandwidth profile $bp_{0.5}$.

Another observation is the differences in $tr^v_{CBT}(t)$ and $tr^v_{MES}(t)$ in Tab. III and Tab. IV. A difference of 3% in network load results in a variation of $tr^v_{CBT}(t)$ and $tr^v_{MES}(t)$ by 10%. Increasing the assigned bandwidth increases the network load from our network. This increase might affect the external APs and devices using the same channel by increasing their chance of frame loss. This leads to an increased number of retransmissions by external nodes which increase the duration of medium energy sensed, and metric $tr^v_{CBT}(t)$.

The second set of experiments was conducted with different bandwidth profiles with and without external traffic. The motivation behind the experiments is to evaluate the traffic monitoring using *unusable wasted tokens* in different medium conditions. For this, the experiments in the presence of external traffic were conducted on two channels of 2.4 GHz ISM band with different APs running on each channel and in different

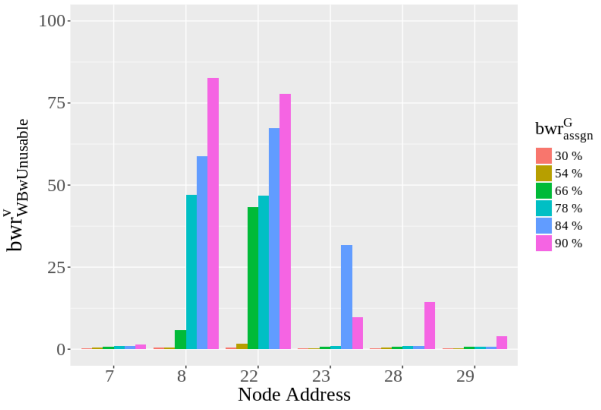

Fig. 3. Experiment conducted in the *static testbed* with external traffic on Channel 6. x-axis represents the *NodeAddress*, y-axis denotes the *unusable wasted bandwidth ratio* in percent.

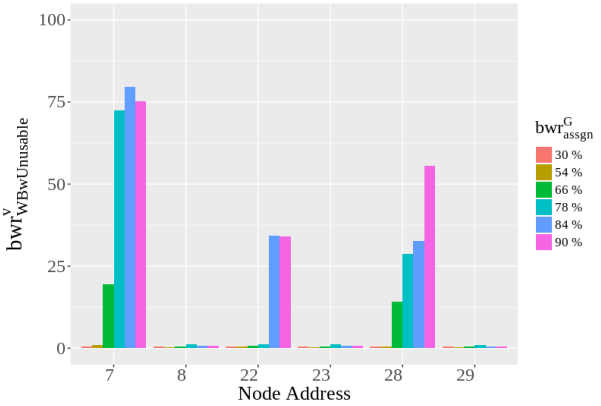

Fig. 4. Experiment conducted in the *static testbed* with external traffic on Channel 1

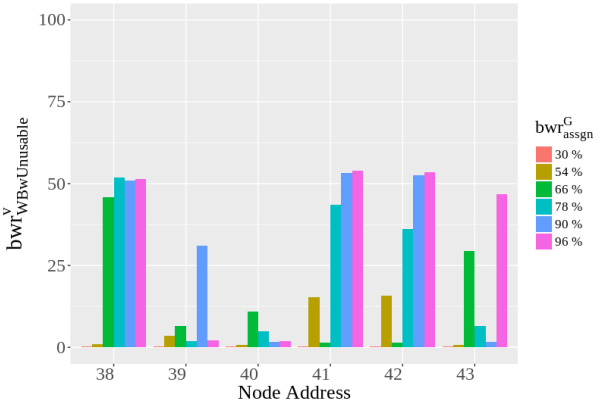

Fig. 5. Experiment conducted in the *mobile testbed* without of external traffic

locations. In these experiments, higher network loads from 30% to 96% were used. Again, the network load bwr^G_{assgn} is equally split among 6 nodes. For example, a network load of bwr^G_{assgn} = 30% split among 6 node results in bwr^v_{assgn} = 5 % for each node. Each node runs

the same application mentioned earlier in this section for 20 minutes. Fig. 3, Fig. 4, and Fig. 5 show results with higher assigned bandwidth ratios for each experiment.

Fig. 3 shows that increased network load initially affects nodes 8 and 22. This experiment was conducted on channel 6 of the 2.4 GHz ISM band. Fig. 4 shows the results for the experiment conducted on channel 1 with the same bandwidth configurations. Here, nodes 7 and 28 lost most of the assigned bandwidth. Both experiments were conducted during working hours, with APs from different working groups located in the same building. For example, an AP of our working group operating on channel 1 is located near to node 7. Experiments conducted on channel 1 always affected node 7, and on channel 6 affect nodes 8 and 22. The experiments were repeated on both channels and similar results were obtained, i.e. it seems to depend on the channel and thus external traffic, which nodes get affected most.

Fig. 5 shows the result in absence of external traffic in the *mobile testbed*. It shows that the increased network load affects the nodes randomly depending on the contention window and random backoff. Here, nodes are very close to each other and no external traffic affects the contention. For example, at network load of 54%, nodes 41 and 42 lost most of the assigned bandwidth comparing to a higher network load of 66%. On the other hand, at a network load of 66%, nodes 38 and 43 lost most of the assigned bandwidth. This randomness is expected since there is no influence of external traffic (from fixed APs), and the contention for the medium is between the six internal nodes which have equal influence on each other.

The results of the experiments show that the metrics derived from *unusable wasted tokens* and CBT can both be used for traffic monitoring. While CBT greatly depends on the hardware used and only some drivers allow to poll the required register values, wasted tokens can be monitored independently of hardware and drivers. Moreover, wasted tokens can be used to directly detect the contention faced by a node for transmitting. Therefore, $bwr^{v}_{WBwUnusable}(t)$ seems to be a suitable metric for our CBM to decide whether to grant or deny the $bwr^{v}_{requested}$ by a node v, thus performing the admission control.

VI. Conclusion

We have proposed a new metric called *unusable wasted bandwidth ratio* based on token bucket traffic shaping for the detection of network overload. Our experiments show that, similar to CBT, this metric can be used to detect network load locally. Especially, our experiments show that even in a single-hop network, external traffic can affect nodes differently, and both metrics can be used to detect which nodes are affected most. While CBT is highly dependent on hardware and often not available in practice, our metric is hardware-independent.

Currently, CBT has been proposed as a metric for many different functionalities like rate-adaption, admission control and routing. In future work, we will further explore the use of our metric in these different functionalities. Especially, we will implement a bandwidth manager that uses the metric for dynamic admission control.

References

[1] J. Lunze and L. Grüne, "Introduction to Networked Control Systems," in *Control Theory of Digitally Networked Dynamic Systems*. Springer, 2014, pp. 1–30.

[2] International Electrotechnical Commission, "Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART (IEC 62591 ed 1.0)," Geneva, Switzerland, April 2010.

[3] ——, "Industrial Communication Networks — Wireless Communication Network and Communication Profiles - ISA 100.11a (IEC 62734 ed 1.0)," Geneva, Switzerland, March 2012.

[4] A. S. Tanenbaum, *Computer Networks, Third edition*. Reading, Massachusetts: Prentice Hall PTR, 2003, Pages 381-384.

[5] S. Ghazal, J. B. Othman, and J.-P. Claudé, "Traffic Management based on Token Bucket Mechanism for WiMAX Networks," *Cluster Computing*, vol. 15, no. 4, pp. 391–400, 2012.

[6] K. Wongthavarawat and A. Ganz, "Packet Scheduling for QoS support in ieee 802.16 Broadband Wireless Access Systems," *International Journal of Communication Systems*, vol. 16, no. 1, pp. 81–96, 2003.

[7] C.-H. Jiang and T.-C. Tsai, "Token bucket based CAC and packet scheduling for IEEE 802.16 Broadband Wireless Access Networks," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 1. IEEE, 2006, pp. 183–187.

[8] J. L. Valenzuela, A. Monleon, I. San Esteban, M. Portoles, and O. Sallent, "A hierarchical Token Bucket algorithm to enhance QoS in IEEE 802.11: proposal, implementation and evaluation," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 4. IEEE, 2004, pp. 2659–2662.

[9] P. Dely, A. J. Kassler, and D. Sivchenko, "Theoretical and experimental analysis of the Channel Busy Fraction in IEEE 802.11," in *Future Network and Mobile Summit, 2010*. IEEE, 2010, pp. 1–9.

[10] P. A. K. Acharya, A. Sharma, E. M. Belding, K. C. Almeroth, and K. Papagiannaki, "Congestion-aware rate adaptation in wireless networks: A measurement-driven approach," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*. IEEE, 2008, pp. 1–9.

[11] I. Sheriff, P. A. K. Acharya, and E. M. Belding, "Measurement-driven Admission Control on Wireless Backhaul Networks," *Computer Communications*, vol. 31, no. 7, pp. 1354–1371, 2008.

[12] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in IEEE 802.11 b Wireless Networks," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association, 2005, pp. 25–25.

[13] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 279–290.

[14] http://elixir.free-electrons.com/linux/v4.6.5/source/include/uapi/linux/nl80211.h lines 2951 - 2974.

[15] http://elixir.free-electrons.com/linux/v4.6.5/source/include/net/cfg80211.h lines 579 - 597.

[16] V. Jacobson and S. McCanne, "libpcap: Packet capture library," *Lawrence Berkeley Laboratory, Berkeley, CA*, 2009.

[17] http://www.radiotap.org/.