Modeling Reliable M2M/IoT Traffic over Random Access Satellite Links in Non-saturated Conditions

Manlio Bacco, Pietro Cassarà, Marco Colucci, Alberto Gotta

***Abstract*—Nowadays, Machine-to-Machine and Internet of Things traffic sources puts the terrestrial networks under great pressure. While 5G is still on its way, satellites are used to deliver a fraction of such an enormous traffic rate. In this work, we investigate the use of the Constrained Application Protocol to reliably deliver Machine-to-Machine and Internet of Things traffic in a *push* fashion, which also implements a Selective Repeat Automatic Repeat reQuest and a sender-based variant of the TCP Friendly Rate Control protocol. We aim at providing an analytical model to evaluate the working point of the system in non-saturated conditions as a function of the MAC parameters in use, when such a closed-loop congestion control mechanism is in use over a random access satellite channel. The proposed analytical model is then validated against simulation results, showing a good precision.**

I. Introduction

According to data traffic forecast reports, more than one trillion of devices will be connected to the Internet in 2019, passing the threshold of 2.0 zettabytes per year. A large fraction of this traffic is generated by Machine-to-Machine (M2M)/Internet of Things (IoT) communications, showing an ever-increasing growth rate. In this context, a large effort is already in place in order to accommodate M2M/IoT communications, which are putting the current infrastructure under strong pressure. 5G communications are expected to relieve some of this pressure, providing large bandwidth, large coverage, scalability, energy efficiency, and addressing privacy and security concerns. But is also expected that 5G will trigger new services and applications to appear, increasing the demand of coverage, bandwidth, and low latency: ultimately, requiring more and more network resources to be deployed. While 5G is still on its way, 4G is struggling to accommodate the needs of current IoT/M2M deployments: in fact, 4G networks can be insufficient to deal with such an enormous traffic rate [1], [2], [3]. In these scenarios, satellites can play a complementary role by offloading a fraction of the traffic from terrestrial networks, and their role is expected to grow [4], [5]. Further to this, satellites naturally provide ubiquitous coverage, making them attractive in areas where cellular communications are not available. In addition, the coverage represents a key feature to enable the so-called IoT massive internetworking. For instance, scenarios involving connected cars and autonomous vehicles may require the use of satellite to be effectively enabled. Nowadays, the large part of satellite-enabled IoT is dominated by narrowband providers, such as L-band, but high-throughput Ku-band and Ka-band satellite connections can open to the volume of opportunity provided by the M2M/IoT sector, in a very near future.

Several architectures and protocols have been proposed to efficiently deliver M2M/IoT traffic, dealing with power efficiency, resource-constrained devices, and scarce bandwidth. In order to really enable IoT communications, *interoperability* must be guaranteed, among other features. The valuable survey in [6] underlines that current M2M/IoT markets are fragmented, because different vertical solutions have been already designed and implemented: still, the effort to identify a common architecture must be a primary concern. Internet Engineering Task Force (IETF) is making large efforts in promoting the adoption of the Constrained Application Protocol (CoAP) protocol [7]. It is an User Datagram Protocol (UDP)-based lightweight application protocol, designed for M2M/IoT communications, intended for resource-constrained devices on constrained Internet Protocol (IP) networks [8]. On the other hand, Transmission Control Protocol (TCP)-based protocols have been proposed as well. One of the most used is the Message Queue Telemetry Transport (MQTT) protocol [9], designed by IBM in 1999 for satellite communications. While CoAP implements a request-response paradigm, MQTT is based on the use of the Publish / Subscribe (PUB/SUB) paradigm: the latter *pushes* fresh data to the consumers, as soon as they are available, while the former periodically *pulls* new data (if any) from the producers. Another difference between CoAP and MQTT is related to the congestion control mechanisms: while the latter is TCP-based, the former is UDP-based. Because of this, MQTT demands to TCP the congestion control. On the contrary, CoAP must implement a proper mechanism at the application layer, in order to avoid congestion phenomena.

When considering IoT/M2M data exchanges via satellite, the use of Random Access (RA) techniques is a necessity, in order to accommodate a large population of devices sending data in a sporadic way. Conversely, the use of Dedicated Access (DA) techniques can prove unpractical in those scenarios. Because of this, we focus our attention on RA techniques. Further than this, we are interested in reliable data exchange, thus using an Automatic Repeat reQuest (ARQ) protocol. A further question is related to the layer where the ARQ algorithm should run, and whether or not a congestion control mechanism must run at the same layer. TCP takes care of ARQ and congestion control at the transport layer, allowing the design of reliable applications without additional effort. In the last years, an opposite tendency is widespreading: i.e., moving ARQ and congestion control mechanisms from the

Institute of Information Science and Technologies (ISTI), National Research Council (CNR), Via Moruzzi 1, 56124 Pisa
correspondant author: Manlio Bacco, e-mail: manlio.bacco@isti.cnr.it

transport to the application layer[1], relying on a lightweight transport protocol simpler than TCP (like, for instance, UDP). RFC 6574 describes this dilemma as follows: *with the work on new application protocols, like CoAP, the question of whether a congestion control mechanism should be used at the underlying transport protocol or by the application protocol itself arises*[2].

In this work, we propose the use of CoAP over RA satellite links in M2M/IoT scenarios, providing reliable data exchanges by means of an ARQ and a congestion control mechanism implemented at the application layer. We provide an analytical characterization of the queuing system, aiming at identifying the working point in non-saturated conditions. The analytical model is then validated against simulation results, in order to show the good match it can provide. To the best of our knowledge, this is one of the few works in the literature dealing, both analytically and empirically, with the use of closed-loop congestion control mechanisms over satellite RA channels, providing a TCP-friendly solution implemented at the application layer.

The rest of this work is organized as follows: Section II provides the necessary background, along with the most relevant works found in the literature. Section III describes the scenario under consideration, providing details on the TCP Friendly Rate Control (TFRC) variant we propose in this work. The analytical system model is then provided in Section IV, showing how the working point can be analytically found in the system under consideration. The validity of the analytical model is validated in Section V against the empirical results provided by simulations. Eventually, the conclusions are drawn in Section VI.

II. Background and Rationale of this work

In this section, we review the most relevant works found in the literature about the analytical characterizations of congestion control algorithms on top of satellite RA schemes. Furthermore, we analyze the congestion control algorithms for CoAP proposed up to now.

A reference work discussing the use of TCP-like congestion control mechanisms on top of satellite RA schemes can be found in [10]. In the latter, the authors analyze the performance of a simplified TCP on top of a Slotted ALOHA (SA) scheme over an RA satellite channel. In the presence of collisions, which are typical of RA channels, the authors describe how the closed loop congestion control mechanism interacts with the erasures due to collisions, triggering retransmissions of lost TCP segments (like they were lost because of congestion). Lower and upper bounds for the throughput estimation at steady state are provided, but disregarding duplicated Acknowledgments (ACKs) and the TimeOut (TO) when a retransmission occurs. The TFRC protocol [11], which was born to provide congestion control over UDP in a TCP-friendly way, is analyzed in [12]. The authors consider a terrestrial wireless link, developing an analytical framework to evaluate the Quality of Service (QoS) that TFRC can offer, in the presence of an underlying truncated ARQ mechanism. The performance metrics under consideration are the link utilization, the throughput, and the packet loss rate. The analysis shows that the approach proposed in [12] is a viable solution when considering a terrestrial Bandwidth-Delay Product (BDP). Conversely, when considering channels with a large BDP, as for instance the RA satellite link under consideration in this work, careful attention must be paid to the use of ARQ mechanisms, in order to avoid instability effects [13], [14].

The use of MQTT over RA satellite channels has been empirically evaluated in [15], while a preliminary empirical evaluation of CoAP in the same scenario is available in [16]. Anyway, a congestion control mechanism is not in use in the latter work. Instead, if we look at terrestrial scenarios, several works can be found discussing congestion control techniques and in the following we present the main relevant ones, along with the necessary background. At this time, CoAP Simple Congestion Control/Advanced (CoCoA) [17] should be considered as the reference congestion control algorithm for CoAP, at least in the case of terrestrial deployments. CoAP does not specify a congestion control algorithm, but just rate-limiting settings, in order to strongly reduce the probability of congestion events. In fact, it makes use of the *NSTART* parameter: it is defined as the maximum number of simultaneous outstanding interactions. It means that, when an ACK is expected (it is the case of CoAP *confirmable* messages), up to NSTART CoAP packets can be in-flight. NSTART is the (fixed) size of the equivalent CoAP transmission window, contrarily to the TCP congestion window (CWND), which is instead variable in size according to the dynamics of TCP congestion control algorithm. In fact, CoAP works in a rate-based fashion, and not in a window-based one (like the most common terrestrial TCP flavors). In the presence of CoAP *non-confirmable* messages, for which an ACK is not expected, a severe rate-limiting rule applies: a conservative limit is provided by [17][3], but it opens to the implementations of a TFRC-like algorithm, as discussed in Section 5.1 of [17]. Instead, the CoCoA algorithm is designed to achieve a better performance level than CoAP: briefly, the latter just considers a binary exponential backoff before retransmissions, without any Retransmission TimeOut (RTO) estimation, while the former also implements an RTO estimator exploiting RTT samples, so being able to adapt to varying network conditions. The aforementioned rate-limiting rule suggested by CoAP specifications sees NSTART equal to one: it is thought for resource-constrained devices with sporadic traffic in terrestrial networks. The effects of a value of NSTART greater than one is discussed in [18], where a modified version of CoCoA is compared to CoAP and to regular CoCoA, when considering cloud services in terrestrial networks. The authors conclude

[1] In fact, complete network stack implementations in user-space are already available, such as Intel DPDK framework, OpenFastPath framework, Multi-Stack Operating System (OS), and so on.

[2] RFC 6574, *Report from the Smart Object Workshop*, Section 4, paragraph titled *Congestion Control*.

[3] This strong limitation on the maximum outgoing rate is confirmed in RFC 7641: *the server should not send more than one non-confirmable notification per Round-Trip Time (RTT) to a client on average. If the server cannot maintain an RTT estimate for a client, it should not send more than one non-confirmable notification every three seconds and should use an even less aggressive rate when possible.*

that NSTART can be safely increased (a NSTART ≤ 4 is considered), even if the network can be driven into congestion in a faster way.

If we look at the use of CoAP in satellite scenarios, a few works can be found [16], [19]. In [19], the authors assess the performance of CoAP over a satellite channel, by comparing the achievable performance level with that provided by the use of MQTT. They conclude that CoAP, if subject to a fine tuning of the available parameters, can provide an higher performance level than that achievable by the use of MQTT. In [16], the use of CoAP over satellite is discussed where NSTART ≥ 1 and a low traffic rate, compatible with M2M/IoT scenarios, is considered. The main innovation in [16] is the joint use of the *observer* pattern [20] and of the *proxying* functionality [7] (later described in details). So, a fully compliant PUB/SUB-like implementation of CoAP is obtained, providing a reduction of the *completion time* metric. The latter is defined in [16] as the time it takes to successfully deliver a CoAP message from the data producer to the data consumer(s). A guaranteed message delivery depends on the use of an ARQ algorithm. In the case of RA channels, dominated by collisions, load control must be applied in order to avoid the aforementioned instability effects.

In this work, we design and implement a sender-based variant of the TFRC algorithm, namely TFRC-s, implemented as CoAP congestion control algorithm along with an ARQ mechanism, and we assess its performance level, both analytically and empirically, in a satellite Digital Video Broadcasting - Return Channel via Satellite (DVB-RCS2)-based system.

III. Reference Scenario

In this section, we describe the fundamental elements composing our reference scenario: a brief overview of the CoAP protocol is provided in Section III-A, along with some details on the proxy functionality and the observer pattern. TFRC and our TFRC-s implementations are described in Sections III-B and III-C, respectively. Furthermore, the RA scheme in use over a DVB-RCS2 link is described in Section III-D, then the complete reference architecture is described in Section III-E.

A. CoAP protocol

CoAP follows a Representational State Transfer (REST) architectural style, where each server encapsulates a *resource*, identified by a Uniform Resource Identifier (URI). A CoAP client sends a request by means of a confirmable or non-confirmable message, in order to retrieve the resource representation. A CoAP request is performed by means of the HTTP *GET* verb. The message format includes a fixed-size header (4 bytes), a variable-length *Token* field (0-8 bytes), an *options* field, and the payload (up to a suggested size of 1024 bytes). CoAP provides optional reliability at the application layer: if confirmable messages are sent, then the ARQ mechanism in use is a simple Stop-and-Wait protocol (because of the default $NSTART = 1$), employing exponential back-off. Anyway, CoAP specifications open to the implementation of a different ARQ mechanism, as the use of $NSTART > 1$ would require. This would allow to take advantage of a large class of devices supporting more complex mechanisms and benefiting of a larger transmission window. In this work, we suppose a $NSTART \geq 1$ and the use of an Selective-Repeat ARQ (SR-ARQ). In order to reduce the delivery delay of CoAP, which we recall implements a request-response pattern, the mechanisms described in the next two paragraphs can be applied.

1) Observer pattern: CoAP specifications open to the implementation of the so-called *observer* pattern [20], similar to the PUB/SUB one. A CoAP client performs a registration to the server(s), indicating the URIs it is interested into. After that, the server pushes new data to the registered client as soon as they are available. Anyway, there is still a difference with respect to the PUB/SUB paradigm: the decoupling between data consumers and producers cannot be provided by the observer pattern. The absence of an intermediate entity leaves the server(s) in charge of keeping a list of the registered clients. The next paragraph explains how the use of a CoAP proxy can solve the aforementioned issue.

2) CoAP proxying: The use of the *proxying* functionality, as described in [7], adds the missing element that can provide PUB/SUB-like data exchanges. A proxy is defined as a CoAP endpoint that can be delegated by clients to perform requests on their behalf. Thus, a CoAP proxy is an intermediate entity, decoupling the clients from the servers.

By implementing both proxying functionality and observer pattern, as proposed in [16], the number of interactions between clients and servers can be successfully reduced. In fact, in satellite systems, where network resources are expensive, this provides several advantages (such as a lower completion time).

B. TCP Friendly Rate Control (TFRC)

TFRC [11] is a congestion control mechanism designed for unicast flows, ensuring reasonable fairness with competing TCP flows. TFRC exploits a feedback mechanism: the measured loss rate at the receiver is sent back at the sender, in order to adjust the sending rate accordingly. The latter is estimated by relying on the throughput equation in [21] and here reported in (1), which requires as inputs the experienced RTT, the loss rate p, the initial RTO value TO, and the b value (*delayed ACKs* feature of TCP). It has the following form:

$$T = \frac{1}{RTT\sqrt{2bp/3} + TO\,\left(3\sqrt{3bp/8}\,p\,(1+32p^2)\right)}, \qquad (1)$$

and provides the throughput estimation T [packets/second]. By exploiting feedback reports, an RTT estimation is performed at the sender. Conversely, p is estimated at the receiver and sent back at the sender via feedback reports. By relying on (1), the sender can adjust its sending rate in order to match the perceived channel statistics and interact fairly with competing TCP flows. TFRC estimates the loss rate by relying on a filter, which weights the most recent l loss intervals ($l = 8$ is recommended in [11]): a *loss interval* is the number of packets sent by the receiver up to the detection of a loss (excluded). When a loss is detected, a new loss interval begins. In order to have smooth variations of the loss rate, the most recent loss intervals weight more than the older ones.

C. TFRC-s: a sender-side only variant of TFRC

We designed and implemented a sender-based variant of the TFRC protocol, namely *TFRC-s*, which is used as CoAP congestion control mechanism. TFRC estimates p at the receiver side, then it is transmitted back to the sender. In TFRC-s, the loss rate p is estimated at the sender by exploiting CoAP ACKs: thus, TFRC-s does not need feedback reports[4] because CoAP ACKs are used for both ARQ and congestion control mechanisms, similarly to TCP. We underline that only the sender implements TFRC-s, while the CoAP receiver does not: the latter only acknowledges the received CoAP packets. Thus, differently from TFRC, our variant does not require that the receiver implements TFRC-s, as shown in Figure 1. Further than this, the sending rate is updated more frequently by TFRC-s than by TFRC, by exploiting each ACK. In more words, a receiver employing TFRC should send at least one feedback report per RTT in order to adapt to varying congestion conditions. In the case of high delay links, an additional delay introduced by such a mechanism may reduce the reactiveness of the system, causing under-utilization of the resources, or, conversely, load peaks with overwhelming collision rates. Therefore, in order to track as precisely as possible the instantaneous value of the collision rate, a TFRC-s sender updates the sending rate at every ACK (thus, more than once per RTT). The latter is estimated by using (1), where we assume a b value equal to two[5]. TFRC-s sets $l = 8$. In our scenario, we assume an in-order packet reception, thus the reception of an out-of-order ACK is interpreted as the symptom of a packet loss. If so, the lost packet is retransmitted at the expiration of the current TO, whose value is doubled (up to 64 seconds) after the retransmission.

D. Random access scheme in use

We consider a DVB-RCS2-compliant satellite system and a finite number n of ground stations, sending data via a satellite RA return link towards remote stations. The Media Access Control (MAC) protocol in use is Contention Resolution Diversity Slotted ALOHA (CRDSA) [22], employing 3 replicas. Time is slotted and each frame, of duration T_F seconds, is composed of M time-slots. Each ground station can use up to one time-slot per frame with probability equal to one. A RA channel has an optimal load level: looking at the curve load vs. throughput in [22], the so-called G^* point can be identified, defined as the load level that provides the maximum achievable throughput. The working point of the system should be as close as possible to G^*, but it should not exceed the latter, in order to avoid an overwhelming collision rate. The use of an ARQ mechanism, needed for reliability, forces to use a control mechanism, in order to avoid the aforementioned instability effects. We recall that the DVB-RCS2 standard foresees a normative load control mechanism, but also that it is too simple to effectively counteract an increasing loss rate, and it needs a not trivial tuning per different scenarios, as proved in [23]. Because of the latter, it is not used in this work. We assume that each time-slot carries exactly a single packet coming from Layer III, which encapsulates a single CoAP message. We firstly consider the case of power balancing among the stations. Power-balancing is the worst condition that the ground stations can face, because power unbalancing effects would improve the benefits provided by the capture effect, improving the Successive Interference Cancellation (SIC) performance [24]. Because of this, the empirical results provided in Section V-A represent a lower bound for the achievable system throughput. In addition to the aforementioned scenario, we also explore the effect of having a fraction of terminals experiencing severe fading conditions in Section V-B, in order to assess how the system working point shifts.

E. Reference architecture

We assume that a large number of CoAP servers are connected to a local CoAP proxy, which sends data to remote CoAP clients over a RA satellite link, as shown in Figure 1. In a general way, also the clients can be downstream a proxy. Furthermore, we assume that CoAP ACKs are always correctly received, and that proxies are not resource-limited. TFRC-s is implemented as congestion control mechanism of the CoAP protocol, and we assume a very large NSTART value, in order to have a sending rate only depending on TFRC-s. If NSTART were set to a value lower than the BDP [packets/second] of the satellite link, it would limit the maximum number of in-flight packets (similarly to the TCP *advertised window*) so that, in turns, the achievable sending rate would be limited as well.

The layer-III packets pending for transmission are stored in the MAC buffer (or queue). We define the arrival process that feeds the buffer as $A(t)$. Each packet in the buffer is sent according to the service process, namely $S(t)$. A packet sent by a CoAP proxy can be lost with probability p because of the following: with probability p_{drop}, because of buffer overflow; with probability p_{coll}, if a collision occurs in the channel; with probability p_{loss}, if the packet is corrupted due to channel conditions. In this work, we assume $p_{drop} = 0$: in fact, we set a MAC buffer size B large enough to avoid buffer overflow phenomena.

IV. System Model

In this section, we provide the analytical characterization of the considered system. The aim of this part is in deriving the working point of the system when a closed loop congestion control is applied, in terms of system throughput at steady state. The working point is a function of both the number n of competing terminals (CoAP proxies) and the MAC parameters, i.e. the number of timeslots M and the number of the CRDSA replicas N_{rep}, which determine the loss rate p. In order to do this, we analyze the queuing process at each terminal.

[4] We remark that, if TFRC is used in combination with an application protocol providing reliability, both application-level ACKs and feedback reports at transport layer are sent from the receiver to the sender.

[5] A larger b implies a reduced traffic on the satellite forward link, but also that a less reactive behavior must be expected because of the fewer received ACKs. In this work, we assume a typical $b = 2$ value, which represents a reasonable trade-off.

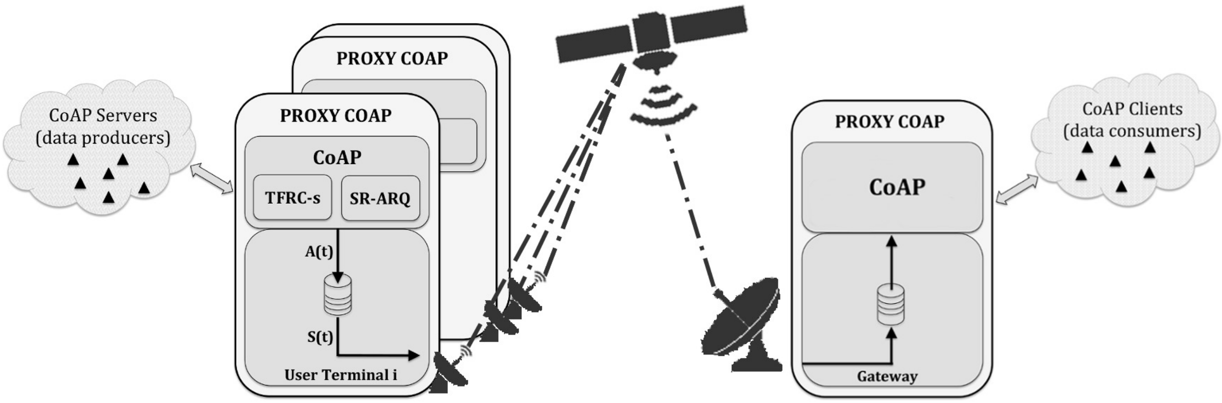


Fig. 1: The reference scenario under consideration in this work, where TFRC-s is used as CoAP congestion control in conjunction with SR-ARQ in the sender-side CoAP proxy.

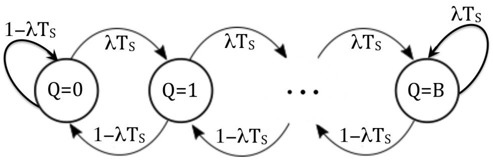


Fig. 2: DTMC of the queuing process.

We observe the arrival process $A(t)$, driven by TFRC-s, which feeds each queue (or buffer) with rate λ. By observing $A(t)$ at multiples of the average service time T_S, the arrival probability is λT_S, where λ is given by (1) divided by the bottleneck rate T_F^{-1}. Note that, per ground station, a single packet is served in the upcoming RA frame with rate T_F^{-1}, thus the average service time is equal to the frame duration. Figure 2 shows the Discrete Time Markov Chain (DTMC) that describes the evolution of the queuing process as a function of λ, which depends on p, RTT, TO, and b. Let $\boldsymbol{\pi}$ be the stationary probability vector of the buffer length, and $\mathbf{P}$ the transition matrix of the DTMC, such that:

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}; \tag{2}$$

$$\boldsymbol{\pi} = [\pi_0 \cdots \pi_B];$$

$$\mathbf{P} \in \mathbb{R}^{BXB} \ ; \ \boldsymbol{\pi} \in \mathbb{R}^B$$

$$\mathbf{P} = \begin{bmatrix} 1-\lambda T_S & \lambda T_S & 0 & \cdots & 0 \\ 1-\lambda T_S & 0 & \lambda T_S & \cdots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & 1-\lambda T_S & 0 & \lambda T_S \\ 0 & \cdots & 0 & 1-\lambda T_S & \lambda T_S \end{bmatrix}.$$

The solution of (2) can be numerically evaluated, or can be found by relying on the $\mathcal{Z}$-transform (as in [25], sec. 2.3). Note that π_0 is used to evaluate the *transmission probability* p_{tx}, defined as the probability that a transmission occurs, given that the buffer is not empty, according to the following expression:

$$p_{tx} = 1 - \pi_0. \tag{3}$$

π_0 is defined as:

$$\pi_0 = \frac{\alpha^B}{1 + \alpha\left(1 + \alpha\,\frac{\alpha^{B-1}-1}{\alpha-1}\right)} \tag{4}$$

with $\alpha = \frac{1-\lambda T_S}{\lambda T_S}$. The proof is in Appendix A.

We recall that we choose B large enough to prevent any buffer overflow. It means that, under this assumption, the queue size can be seen as infinite: more precisely, it means that (4) can be approximated to (proof in Appendix B):

$$\pi_0 = \frac{\alpha - 1}{\alpha}. \tag{5}$$

A. Uniform Case

In this section, we consider the case of all terminals experiencing the same loss rate p (*uniform case*). The actual load of the system depends on the combined dynamics of two processes: those depending on the n concurrent queues that feed the system, and those depending on CRDSA (i.e., the number of replicas and the SIC process). The number of concurrent transmissions in an RA frame is given by a binomial random variable $\mathcal{B}(n, p_{tx})$ [23], while the average normalized system load G is defined as follows:

$$G = \frac{E\left[\mathcal{B}(n, p_{tx})\right]}{M} = \frac{n}{M}\,p_{tx} = \frac{n}{M}\,(1 - \pi_0), \tag{6}$$

where we define the normalized nominal load as $G_n = n/M$. By combining (1), (4), and (6), we obtain G as a function of n and p.

The following equation provides the expression of p as a function of G and the number of replicas N_{rep} according to the law that describes the SIC process Γ:

$$p = \Gamma(G, N_{rep}). \tag{7}$$

We refer the reader to [24] for further details. The equation system composed of (6) and (7) is a nonlinear system in the two variables p and p_{tx}, which can be numerically solved. In fact, by inverting (7), it results that $p_{tx} = M\Gamma^{-1}(p)/n$ is a continuous and monotone increasing function in the range $p \in [0, 1]$. Thus, the system has a unique solution.

Name	Value
RA scheme	CRDSA
Number of replicas N_{rep}	3
Frame duration T_f	13 [ms]
Time-slots per RA frame M	64
Nominal RTT	520 [ms]
Nominal transmission power	1.26 [W]

TABLE I: SNS3-based simulator parameters.

n	G_n	p_{tx}	G	p	T
32	0.5	0.90	0.450	0.0016	0.449
38	0.6	0.81	0.486	0.0018	0.479
45	0.7	0.69	0.483	0.0022	0.482
51	0.8	0.61	0.488	0.0025	0.487
58	0.9	0.55	0.495	0.0029	0.494
64	1.0	0.50	0.500	0.0032	0.498

TABLE II: Simulation results for the homogeneous case.

B. Heterogeneous Case

In this section, contrarily to Section IV-A, we consider the case of n terminals each experiencing a different loss rate $p^{(i)}$ for $i = 1, \cdots, n$. In addition, a different transmission probability $p_{tx}^{(i)}$ is considered as well, then extending the framework to a *heterogeneous case*. Therefore, the transmission probability of the i-th terminal in (3) must be redefined as:

$$p_{tx}^{(i)} = 1 - \pi_0^{(i)}. \tag{8}$$

In the same way, the loss rate of the i-th terminal in (7) must be redefined as:

$$p^{(i)} = \Gamma\left(\frac{p_{tx}^{(i)}}{M}, N_{rep}\right). \tag{9}$$

Because of (8) and (9), G becomes:

$$G = \frac{1}{M}\sum_{i=1}^{n} p_{tx}^{(i)}. \tag{10}$$

Hence, the solution of the equation system given by (6) and (7) in Section IV-A can be extended to a system of n pairs of equations, which can be numerically solved.

V. Performance Evaluation

The scenario presented in Section III is here empirically evaluated: the proposed analytical framework in the two cases presented in Sections IV-A and IV-B is validated in Sections V-A and V-B, respectively. We use the S-NS3 simulator [26], based on NS3, and the results here presented are obtained through an extensive simulation campaign. Table I reports the main parameters of the simulator. We adopt CRDSA with $N_{rep} = 3$ because it outperforms both CRDSA with two replicas and Irregular Repetition Slotted ALOHA (IRSA) [27].

A. Uniform case - simulation results

In this section, the channel is assumed as ideal, i.e. the system does not undergo fading phenomena. Therefore, p is only characterized by MAC collisions due to the use of a RA channel. The results are shown in Figure 3: the loss curve of 3-CRDSA (continuous line) is estimated according to [24], by assuming that, if a collision occurs, the two (or more) packets in the colliding timeslot are erased. This loss curve has been further validated through extensive simulations. The dashed line in Figure 3 is obtained by exploiting the analytical framework we propose in Section IV, and it represents the load offered by each terminal, i.e. G/n, where G can be read in (6). As explained in Section IV, the working point of the system at steady state can be analytically obtained by solving the equation system composed by (6) and (7), whose solution is given by the interception of the curves in Figure 3. The red squares represent the working points obtained by the simulations, confirming the validity of the proposed model, which shows a reduced error if $G_n > 0.5$. Note that system loads $G_n < 0.5$ are not under consideration in this work, because they would be the case of a system working under queue saturation conditions (i.e., $p_{tx} \approx 1$), where the interactions between the underlying RA scheme and the TFRC-s at the application layer become negligible. Table II provides the working point marked in red in Figure 3 as (p,p_{tx}) at each load G_n under consideration, and it also shows the aggregated normalized load G. In the last column, the aggregated normalized throughput T can be read.

In the following, we empirically evaluate the TCP friendliness of TFRC-s. As already said, TFRC-s uses (1) to estimate the sending rate. We can estimate the equivalent sending rate of TCP as follows:

$$\lambda_{TCP}(t) = W_{TCP}(t)/RTT, \tag{11}$$

where $W_{TCP}(t)$ is the CWND, distributed as in [28]. Figure 4 compares the probability mass functions of λ and of λ_{TCP}. TFRC-s preserves the reasonable fairness with competing TCP flows, because the average values of both the distributions are comparable. On the other side, the distributions show different variances, showing how TCP is more aggressive in the bandwidth probing than TFRC-s (and TFRC).

B. Heterogeneous case - simulation results

In this section, we analyze the proposed system in non-ideal conditions: a fraction $n_{out} \leq n$ of the ground stations experience severe fading conditions with rate p_{out}, i.e. a Signal to Noise plus Interference Ratio (SNIR) below the decoding threshold, while the remaining $n - n_{out}$ stations see a channel in clear sky conditions. Note that the stations in severe fading conditions experience a negligible p_{coll} since $p_{out} >> p_{coll}$. In the literature, for instance as in [24], the energy per symbol to noise power spectral density ratio E_S/N_0 of every terminal is i.i.d., with mean μ and standard deviation σ, leading to a uniform power unbalancing among terminals, contrarily to the approach proposed in this section. The impact of power unbalancing is here analyzed at queue level, by differentiating the study of the transmission probability p_{tx} among terminals under severe fading conditions and terminals in clear sky conditions. Figure 5 presents the simulation results in this case, where a subset $n_{out} \in \{21, 42\}$ out of $n = 64$ terminals experience a 3dB loss in the transmission power (reported in Table I) with $p_{out} = [2\%, 5\%]$ w.r.t. the terminals

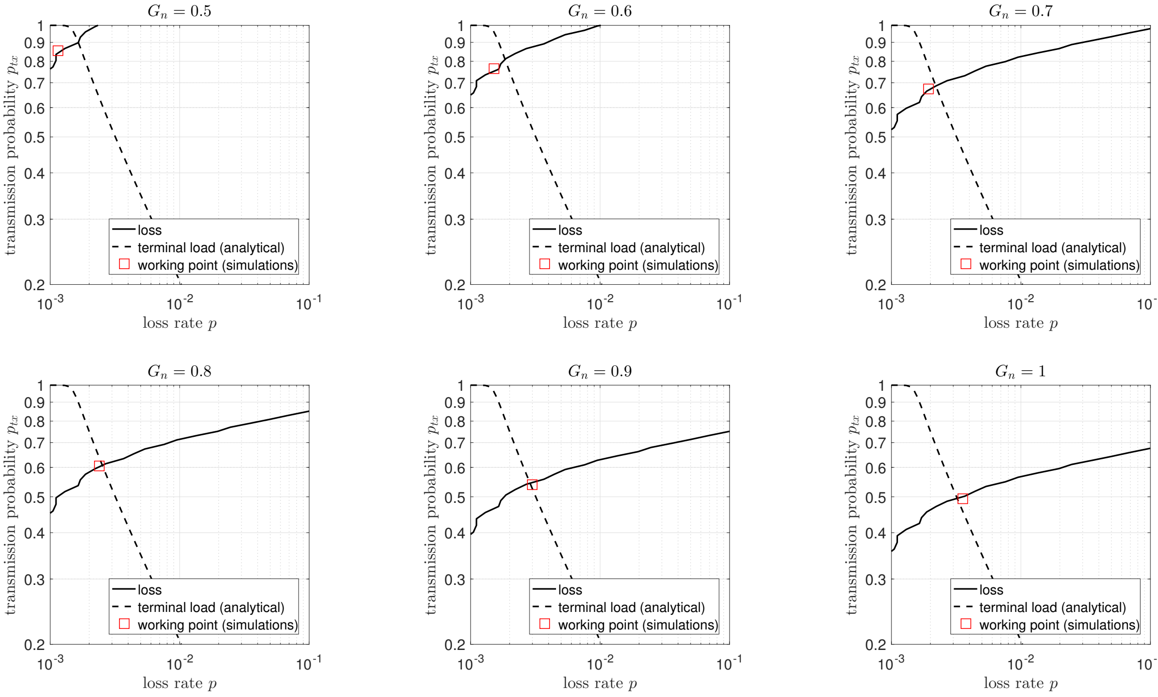


Fig. 3: Transmission probability p_{tx} vs. loss rate p for an increasing number of ground terminals $n \in [32, 64]$, equivalent to $G_n \in [0.5, 1]$. The working point (in red) is the empirical result at different nominal loads, while the analytical working point can be read at the intersection of the loss curve and of the terminal load curve.

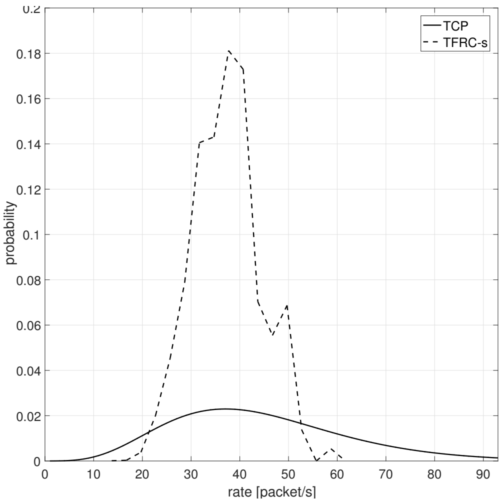


Fig. 4: Distribution of the sending rate λ of TFRC-s and the sending rate λ_{TCP} of TCP for $G_n = 1$.

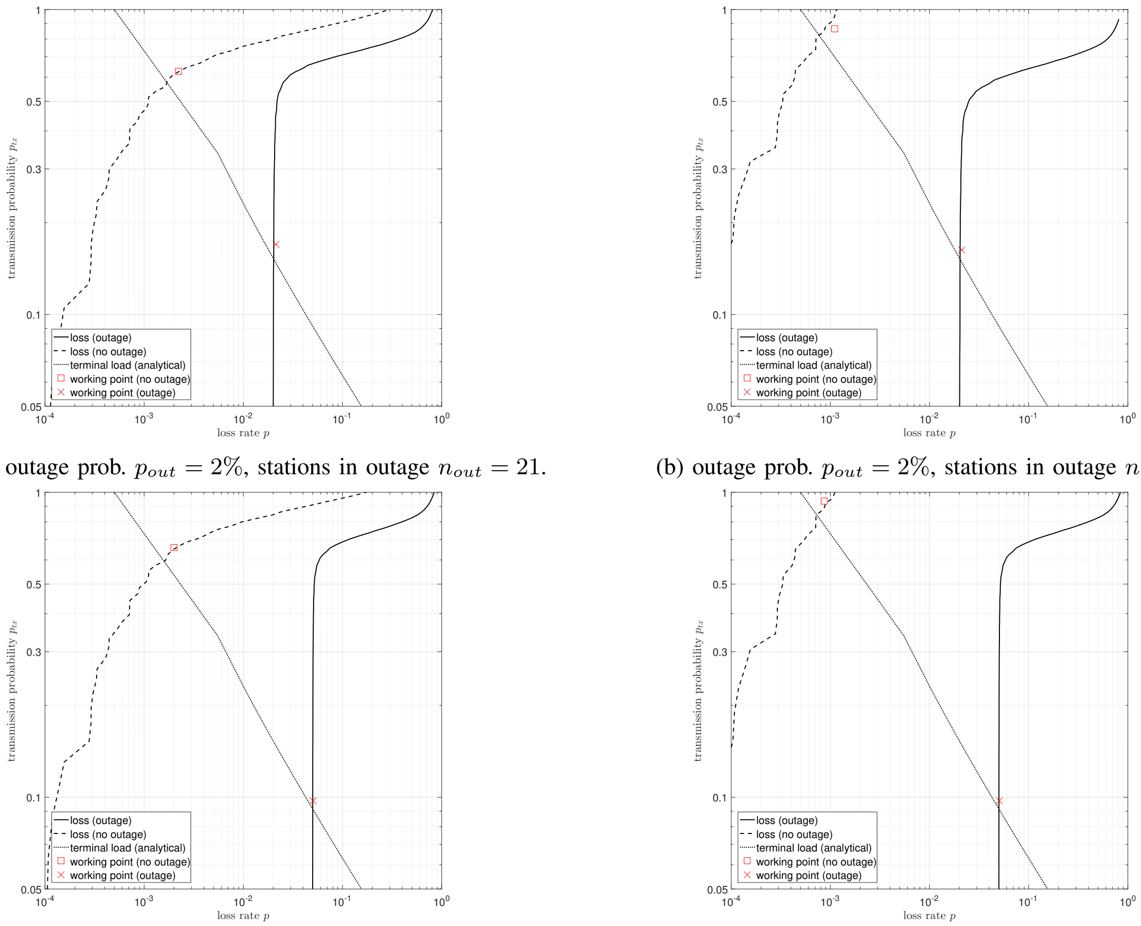
experiencing clear sky conditions. The loss curve, in the case of outage, is calculated as $p_{out}(1-p_{coll})+p_{coll}$. Two working points are presented in Figure 5: a red cross marker for the n_{out} terminals experiencing the outage events, and a red squared marker for the terminals in clear sky conditions. The system load can be calculated with (10). The system loss rate is calculated with (7), using (10) for the expression of G. Table III reports the numerical values of G, T and p_{tx} for $n_{out} \in \{21, 42, 64\}$. Finally, it must be noted that the terminals in clear sky conditions have a working point that may lead to the queue saturation because of the negligible collision rate that they experience. In fact, the latter is reduced, at system level, because n_{out} stations do not actually interfere with the remaining $n - n_{out}$ ones.

n_{out}	p_{out}	p_{tx}^{out}	$p_{tx}^{no\ out}$	**G**	**T**
21	2%	0.17	0.63	0.480	0.4791
42	2%	0.17	0.87	0.404	0.4035
64	2%	0.17	0.00	0.170	0.1667
21	5%	0.10	0.66	0.476	0.4752
42	5%	0.10	0.94	0.389	0.3886
64	5%	0.10	0.00	0.100	0.0950

TABLE III: Simulation results for the heterogeneous case ($G_n = 1$).

VI. Conclusions

In this work, we investigated the use of CoAP on top of a RA satellite medium to reliably deliver M2M/IoT data. In order to achieve reliability as may be required by many services/applications, and also provide a congestion control mechanism TFRC-like, we propose the use of the CoAP protocol in a PUB/SUB-like fashion, which also implements an SR-ARQ and a sender-based variant of TFRC. We underline again that the use of a TFRC-like solution, like the one we design, implement and validate in this work, is strongly recommended by CoAP specifications. The analytical framework in this work allows to evaluate the working point of the system as a function of the CRDSA parameters and of the TFRC-s process feeding the queue, in non-saturated conditions. The framework also allows for predicting the queue utilization factor, by calculating the transmission probability



(a) outage prob. $p_{out} = 2\%$, stations in outage $n_{out} = 21$.

(b) outage prob. $p_{out} = 2\%$, stations in outage $n_{out} = 42$.

(c) outage prob. $p_{out} = 5\%$, stations in outage $n_{out} = 21$.

(d) outage prob. $p_{out} = 5\%$, stations in outage $n_{out} = 42$.

Fig. 5: Transmission Probability p_{tx} vs. loss rate p_{out} for $G_n = 1$. Two working points are marked in red: the square is for the stations in clear sky conditions, the cross is for stations experience severe fading conditions.

at steady state over an RA satellite channel. To the best of our knowledge, this is the first work that analyzes a closed-loop congestion control on an RA satellite channel employing CRDSA. The analytical model is compared against simulation results in two different cases: an homogeneous one, in the absence of fading phenomena, and a heterogeneous one, where a subset of ground stations experiences a temporary outage event.

Acknowledgments

This work has been partially supported by SatNEx (Satellite Network of Experts) programme, IV phase. We would like to thank Dr. Nedo Celandroni who has provided us his valuable support during the writing of this work.

References

[1] T. Pötsch, S. N. K. K. Marwat, Y. Zaki, and C. Gorg, "Influence of future M2M communication on the LTE system," in *6th Wireless and Mobile Networking Conference (WMNC), Joint IFIP*. IEEE, 2013, pp. 1–4.

[2] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 4–16, 2014.

[3] C. Pereira and A. Aguiar, "Towards efficient mobile M2M communications: survey and open challenges," *Sensors*, vol. 14, no. 10, pp. 19 582–19 608, 2014.

[4] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, I. Godor, P. Kersch, A. Kälvemark, G. Lemne, and P. Lindberg, "Ericsson mobility report," *Ericsson [Online]. Available: www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf*, June 2015.

[5] IDATE and UMTS, "Forum report 44," *Mobile Traffic Forecasts*, vol. 2020, 2010.

[6] J. Kim, J. Lee, J. Kim, and J. Yun, "M2M service platforms: survey, issues, and enabling technologies," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 61–76, 2014.

[7] C. Bormann, K. Hartke, and Z. Shelby, "The Constrained Application Protocol (CoAP)," RFC 7252. [Online]. Available: https://rfc-editor.org/rfc/rfc7252.txt

[8] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks," RFC 7228. [Online]. Available: https://tools.ietf.org/html/rfc7228

[9] D. Locke, "MQTT v3.1 protocol specification," *IBM developerWorks Technical Library*, 2010.

[10] C. Liu and E. Modiano, "An analysis of TCP over random access satellite links," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 4. IEEE, 2004, pp. 2033–2040.

[11] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," RFC 5348. [Online]. Available: https://tools.ietf.org/html/rfc5348

[12] H. Shen, L. Cai, and X. Shen, "Performance analysis of TFRC over wireless link with truncated link-level ARQ," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1479–1487, 2006.

[13] C. Kissling, "On the stability of Contention Resolution Diversity Slotted Aloha (CRDSA)," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[14] A. Meloni and M. Murroni, "CRDSA, CRDSA++ and IRSA: Stability and performance evaluation," in *Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012 6th*. IEEE, 2012, pp. 220–225.

[15] M. Bacco, T. De Cola, and A. Gotta, "TCP New Reno over DVB-RCS2 Random Access Links: Performance Analysis and Throughput Estimation," in *Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[16] M. Bacco, M. Colucci, and A. Gotta, "Application Protocols enabling Internet of Remote Things via Random Access Satellite Channels," in *International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6. [Online]. Available: https://arxiv.org/abs/1706.09787

[17] C. Bormann, A. Betzler, C. Gomez, and I. Demirkol, "CoAP Simple Congestion Control/Advanced," draft-ietf-core-cocoa-01. [Online]. Available: https://tools.ietf.org/html/draft-ietf-core-cocoa-01

[18] A. Betzler, C. Gomez, I. Demirkol, and M. Kovatsch, "Congestion control for CoAP cloud services," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*. IEEE, 2014, pp. 1–6.

[19] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of Things application layer protocol analysis over error and delay prone links," in *Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th*. IEEE, 2014, pp. 398–404.

[20] K. Hartke, "Observing Resources in the Constrained Application Protocol (CoAP)," RFC 7641. [Online]. Available: https://rfc-editor.org/rfc/rfc7641.txt

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 303–314, 1998.

[22] E. Casini, R. De Gaudenzi, and O. R. Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 4, pp. 1408–1419, 2007.

[23] A. Munari, G. Acar, C. Kissling, M. Berioli, and H. P. Lexow, "Multiple access in DVB-RCS2 user uplinks," *International Journal of Satellite Communications and Networking*, vol. 32, no. 5, pp. 359–376, 2014.

[24] O. Del Rio Herrero and R. De Gaudenzi, "Generalized Analytical Framework for the Performance Assessment of Slotted Random Access Protocols," *IEEE Transactions on Wireless Communications*, no. 99, pp. 1–13, 2014.

[25] L. Kleinrock, *Queueing Systems: Volume I Theory*. Wiley, 1975.

[26] V. Hytönen, B. Herman, J. Puttonen, S. Rantanen, and J. Kurjenniemi, "Satellite Network Emulation with Network Simulator 3," in *Ka and Broadband Communications, Navigation and Earth Observation Conference*, 2014.

[27] N. Celandroni, E. Ferro, and A. Gotta, "RA and DA satellite access schemes: a survey and some research results and challenges," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2670–2690, 2014.

[28] S. Bohacek and K. Shah, "TCP Throughput and Timeout-Steady State and Time-Varying Dynamics," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, vol. 3. IEEE, 2004, pp. 1334–1340.

Appendix A

The probability of having an empty queue, as in (4) and (5), can be evaluated by applying the $\mathcal{Z}$-Transform to (2). More precisely, the state distribution of the DTMC at the step k is given by the equation:

$$\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi}^{(k-1)}\mathbf{P} = \boldsymbol{\pi}^{(0)}\mathbf{P}^k \; ; \; k \in [1 \cdots \infty[\tag{12}$$

where $\mathbf{P}$ and $\boldsymbol{\pi}^{(k)}$ are a matrix in $\mathbb{R}^{BxB}$ and a vector in $\mathbb{R}^B$, respectively. Given $\boldsymbol{\pi}^{(0)} = [1, 0, \cdots, 0] \in \mathbb{R}^B$, the solution of (12) can be evaluated by applying the $\mathcal{Z}$-Transform to both members of the equation:

$$\sum_{k=1}^{+\infty} \boldsymbol{\pi}^{(k)} z^k = \sum_{k=1}^{+\infty} \boldsymbol{\pi}^{(k-1)}\mathbf{P} z^k. \tag{13}$$

For $\boldsymbol{\pi}^{(k)} \xrightarrow{\mathcal{Z}} \mathbf{\Pi}(z)$, (13) becomes:

$$\mathbf{\Pi}(z) - \boldsymbol{\pi}^{(0)} = z \left(\sum_{k=1}^{+\infty} \boldsymbol{\pi}^{(k-1)} z^{(k-1)} \right) \mathbf{P} = z\mathbf{\Pi}(z)\mathbf{P}, \tag{14}$$

and, by grouping terms in (14):

$$\mathbf{\Pi}(z) = \boldsymbol{\pi}^{(0)} \left[\mathbf{I} - z\mathbf{P}\right]^{-1}. \tag{15}$$

If the inverse matrix of (14) exists, then it is unique. Through the memberwise comparison between equations (12) and (14), we can define the following pair of transformations: $\mathbf{P}^k \xrightarrow{\mathcal{Z}} \left[\mathbf{I} - z\mathbf{P}\right]^{-1}$ and $\left[\mathbf{I} - z\mathbf{P}\right]^{-1} \xrightarrow{\mathcal{Z}^{-1}} \mathbf{P}^k$. By relying on the latter, we can evaluate the transient solution of the distribution probability of the DTMC state. The stationary distribution $\boldsymbol{\pi}$ can be evaluated by calculating $\lim_{k \to +\infty} \boldsymbol{\pi}^{(0)}\mathbf{P}^k = \boldsymbol{\pi}$.

In our case, $\mathbf{P}$ has most of its entries equal to zero, so that the partial fraction decomposition can be found for $\left[\mathbf{I} - z\mathbf{P}\right]^{-1}$ in a numerical way. Then, we obtain a $\mathcal{Z}$-transform that can be inverted. The solution of $\mathbf{P}^{(k)}$ has the following form:

$$a_0\mathbf{A_0} + a_1\, f_1(\rho^k)\mathbf{A_1} + a_2\, f_2(\rho^k)\mathbf{A_2} \cdots \tag{16}$$

where $\mathbf{A}_i \in \mathbb{R}^{BxB}$, $f_i(\rho^n)$ is a function that decreases with power law for $k > 0$; a_i and ρ are constants in the real interval $[0, 1]$. By numerically solving the limit for $k \to +\infty$ of $\mathbf{P}^k$, the terms $f_i, i \geq 1$ become negligible, so that the entries of the stationary probability $\boldsymbol{\pi}$ become:

$$\pi_b = (-1)^b \frac{(1 - \lambda T_S)^{B-b} \, (\lambda T_S)^{-(B-b-1)}}{1 + \displaystyle\sum_{i=0}^{B-2} (1 - \lambda T_S)^{B-i} \, (\lambda T_S)^{-(B-i-1)}} \tag{17}$$
$$b = 0, 1, \cdots, B.$$

The summation at the denominator in (17) can be written as a geometric series, which leads to:

$$\pi_b = (-1)^b \frac{\alpha^{B-b}}{1 + \alpha \left(1 + \alpha \, \frac{\alpha^{B-1} - 1}{\alpha - 1}\right)}, \tag{18}$$

with $\alpha = (1 - \lambda T_S)/(\lambda T_S)$. From (18), we obtain the probability of an empty queue:

$$\pi_0 = \frac{\alpha^B}{1 + \alpha \left(1 + \alpha \, \frac{\alpha^{B-1} - 1}{\alpha - 1}\right)}. \tag{19}$$

Appendix B

Equation (18) provides the solution of the distribution probability of the queue length for finite values of B. When B is close to infinite, π_b can be calculated as:

$$\lim_{B \to \infty} \pi_b = (-1)^b \alpha^{-b} \lim_{B \to \infty} \frac{1}{\frac{1}{\alpha^B} + \frac{\alpha}{\alpha^B} + \frac{\alpha^{B-1} - 1}{\alpha^{B-1}} \frac{\alpha}{\alpha - 1}} \tag{20}$$

This limit is finite if $\alpha > 1 \Rightarrow \lambda T_S < \frac{1}{2}$, so that follows:

$$\pi_b = (-1)^b \frac{\alpha^{-b}}{\frac{\alpha}{\alpha - 1}} = (-1)^b \, \frac{\alpha - 1}{\alpha^{b+1}}. \tag{21}$$

Eventually, we obtain the empty queue probability as $\pi_0 = (\alpha - 1)/\alpha$.