Physical Channel Access (PCA): Time and Frequency Access Methods Simulation in NS-2

Nicolas Kuhn[1,2], Olivier Mehani[1], Huyen-Chi Bui[2], Jérôme Lacan[2], José Radzik[2], and Emmanuel Lochin[2]

[1] NICTA, Sydney, Australia
[2] University of Toulouse, ISAE, TeSA, France

Abstract. We present an NS-2 module, Physical Channel Access (PCA), to simulate different access methods on a link shared with Multi-Frequency Time Division Multiple Access (MF-TDMA). This technique is widely used in various network technologies, such as satellite communication. In this context, different access methods at the gateway induce different queuing delays and available capacities, which strongly impact transport layer performance. Depending on QoS requirements, design of new congestion and flow control mechanisms and/or access methods requires evaluation through simulations.
PCA module emulates the delays that packets will experience using the shared link, based on descriptive parameters of lower layers characteristics. Though PCA has been developed with DVB-RCS2 considerations in mind (for which we present a use case), other MF-TDMA-based applications can easily be simulated by adapting input parameters. Moreover, the presented implementation details highlight the main methods that might need modifications to implement more specific functionality or emulate other similar access methods (*e.g.*, OFDMA).

1 Introduction

When the medium needs to be shared between multiple active users, the resource is fairly distributed with multiple access techniques, such as Multi-Frequency Time Division Multiple Access (MF-TDMA) or Orthogonal Frequency-Division Multiple Access (OFDMA). OFMDA is used in 4G/LTE and WiMAX/802.16 whilst MF-TDMA is used in Digital Video Broadcasting (DVB). Considering recent deployments of such multiple access networks, it is important to study the interactions of these access methods (PHY/MAC) with the rest of the network protocols (transport layer) in order to ensure optimal experience for the end user. This can be achieved through large simulations studies.

Several modules simulating OFDMA or MF-TDMA techniques for NS-2 [1] already exist. In the context of OFDMA, WiMAX has seen a lot of interest [2, 3]. The current DVB-S2/RCS specifications have also been implemented with MF-TDMA characteristics [4, 5]. These modules attempt to be as close as possible to the real systems and the layout of their components, which make them unsuitable to assess proposed changes to the DVB-S2/RCS architecture (*e.g.*, access

methods strategies), nor to extend their work to other MF-TDMA networks. Moreover, this level of realism might not be necessary for the study of high layer behaviours. Indeed, Gurtov and Floyd claim that a better trade-off between generality, realism and accurate modeling can be found to improve transport protocol performance evaluation [6].

We therefore propose an NS-2 module, Physical Channel Access (PCA), which emulates packet delays at the access point based on specific access methods parameters. This module allows to integrate channel access considerations within NS-2 and assess their impact on upper layers performance. We follow the idea of [6] by emulating the characteristics of various physical channel access techniques rather than extending MAC/PHY simulation models. This therefore allows to conveniently study a wider range of scenarios than the modules presented in [4, 5]:

- experimental channel access strategies for MF-TDMA based systems (*e.g.*, current drafts);
- generic time-frequency multiplexing architectures;
- adaptive access methods.

PCA is well suited for current needs in the extension of the DVB specifications: it is based on the currently standardized parameters but allows to depart from them for further investigations. For example, it allows to consider experimental access methods and capacity allocation processes which are under discussion to support transmission of home user data on the satellite return channel (RCS), so far reserved to signaling.

The rest of the paper is organized as follows. In Section 2, we present the general concepts behind MF-TDMA and notations relevant to the rest of this paper. We describe the implementation of PCA in Section 3. We document the internal parameters and present a use case in the context of DVB-RCS2 in Section 4. We conclude and discuss future work in Section 5.

2 MF-TDMA networks

On an MF-TDMA link, the capacity is shared at the *Access Point*: it is dynamically distributed on times × frequency blocks (denoted "frame" in this article). The access point (the NS-2 node where the module we present in Section 3 is introduced) forwards traffic from one or more users to one or more receivers over the shared medium, therefore covering both up and down link scenarios.

Before presenting in details our NS-2 module, we provide some definitions of the terms used in this paper:

- Flow: data transfer at the transport layer;
- Datagram: network layer segment of a flow;
- Link Layer Data Unit (LLDU): N_{data} bytes of a fragmented datagram;
- Physical Layer Data Unit (PLDU): LLDU with an optional N_{repair} recovery bytes ($N = N_{\text{data}} + N_{\text{repair}}$);

– Block: PLDUs can be further split into N_{block} blocks if the access method requires;
– Slots: element of a frame where a block can be scheduled.

2.1 Access methods

In the context of MF-TDMA networks, two classes of access methods can be introduced: dedicated and random access methods.

Dedicated access When a new flow arrives at the access point, parts of the channel have to be reserved for it. This induces a delay resulting from the reservation negotiation process. The reservation ensures that capacity is fairly distributed: if there are 40 slots available and 10 users, each user can transmit data on 4 slots.

Random access No reservation is needed and data can be transmitted without additional delay. However several users can unknowingly use the same slot and data risks not to be recovered. Stronger error codes are introduced at the physical layer and each user can transmit a reduced N_{data} useful bytes: N_{repair} redundancy bytes are added to the N_{data} bytes to form a code word of $N = N_{\text{data}} + N_{\text{repair}}$ bytes that are split into N_{block} blocks. N_{ra} slots form a *Random Access* block (RA block) on which erasure codes are introduced. Each transmitter randomly spreads its N_{block} blocks across the N_{ra} slots of the RA block for spectral diversity.

Performance of random access methods can be described by the probability that a receiver decodes its N_{data} useful bytes depending on the number of users that transmit data on the RA block. Table 1 shows a generic example of such a description where $P_{i,j}$ is the probability that a packet cannot be recovered by the receiver when there are $N_U \in [NbUser_j; NbUser_{j+1}]$ users on the RA block and and the signal-to-noise ratio of the channel is $Es/N0_i$.

Table 1. Random access method performance

0	$NbUser_1$	$NbUser_2$	$NbUser_3$	...	$NbUser_{26}$	0
$Es/N0_1$	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	...	$P_{1,26}$	0
$Es/N0_2$	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	...	$P_{2,26}$	0
...	...	...	...	...	...	...
$Es/N0_X$	$P_{X,1}$	$P_{X,2}$	$P_{X,3}$	...	$P_{X,26}$	0

2.2 Frame structure

The capacity is dynamically distributed between the different users on the time × frequency frame which structure is detailed in Figure 1. At the access point,

transmission of a frame is scheduled every T_F. We denote by N_S the number of time slots available per frequency. The frequencies on which data is transmitted can be divided depending on the access method: F_R frequencies are dedicated to the random access methods and F_D are reserved to the dedicated access methods. In total, a frame can carry $N_S \times (F_R + F_D)$ slots.

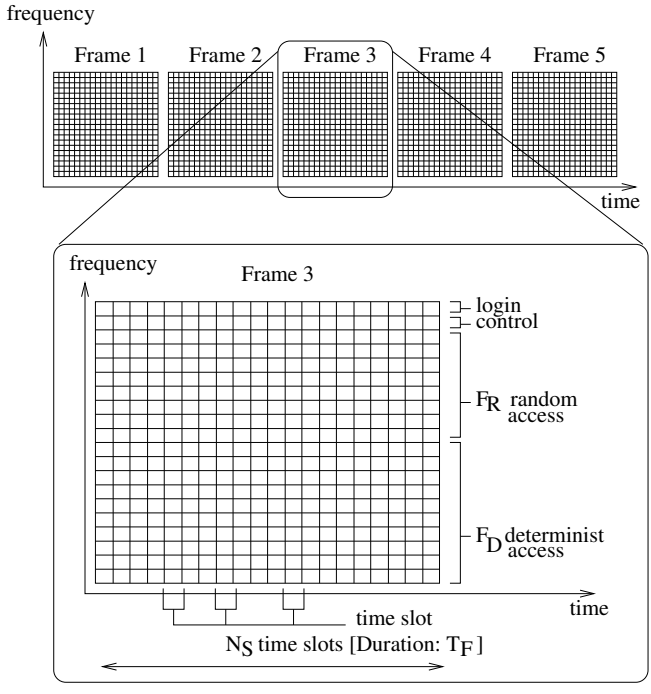

Fig. 1. Times × Frequency block description

2.3 Antenna limitations

It is possible that some transmitters cannot send data on different frequencies at once. This limitation has to be considered when determining the maximum number of slots that a user is allowed to occupy on each frame. This is highly linked to the frame structure, and in this case, a flow can only use N_S slots whatever the number of available frequencies.

As an example, if $N_S = 40$ slots and:

- $F_R = 0$ & $F_D > 1$ (dedicated access): a unique user can exploit $N_S = 40$ slots;
- $F_R = 1$ & $F_D = 0$ (random access), $N_{\text{block}} = 3$, $N_{\text{ra}} = 40$: a unique user can exploit $\lfloor N_S / N_{\text{block}} \rfloor = 13$ slots.

3 Implementation details

In order to control delays, PCA is implemented as a queueing delay. We inherit from the `DropTail` queue management scheme, of which our PCA sub-class redefines the methods used to process the packets. Each node uses the `enque()` and `deque()` methods to add and remove packets from the queue.

In Figure 2, we compare the `enque()` and `deque()` methods of `DropTail` and `DropTail/PCA`. With `DropTail`, when the `enque()` method adds packet P_{N+1}, it is added at the end of the sending buffer and transmitted when $P_1, \ldots, P_N$ have been transmitted with the `deque()` method. With `DropTail/PCA`, when a packet is `enque()`ed, it is also added to the sending buffer. However, depending on the access method introduced, only a subset of the datagram is considered sent with each frame. When the last byte of a datagram has been transmitted, `deque()`, which is called every T_F, removes the packets from the sending buffer and passes it along.

The `DropTail/PCA` queuing policy is implemented in two files (`pca.cc` and `pca.h`) located in the `queue/` sub-directory of the NS-2 source. We detail the content of these files here.

3.1 Data structures

Our module implements linked-lists in order to store information about the current flows and their packets.

Packets list The packet list contains information about the different packets that have reached the access point node but have not been fully transmitted yet. Each packet is defined by:

- `appl_id`: identifier of the flow;
- `pkt_seqno`: sequence number in the flow;
- `frame_in`: emulated frame number after which the data of the packet start to be transmitted;
- `frame_out`: frame number at which the last bit of the packet must be transmitted;
- `bool_first_frame`: boolean to specify if the connection needs to be established (first packet of the current application);
- `bool_lost`: boolean specifying whether the packet is lost;
- `bool_rand`: boolean specified if the access method is random (`bool_rand`=1) or dedicated (`bool_rand`=0);
- `bits_to_send`: actual number of bits of the datagram that have not been sent yet;
- `bits_next_frame`: number of bits that will be sent at the next frame;
- `remaining_slot_frame_appl_det`: number of dedicated slots that remain for this packet's flow;
- `remaining_slot_frame_appl_rnd`: number of random slots that remains for the flow of the packet;

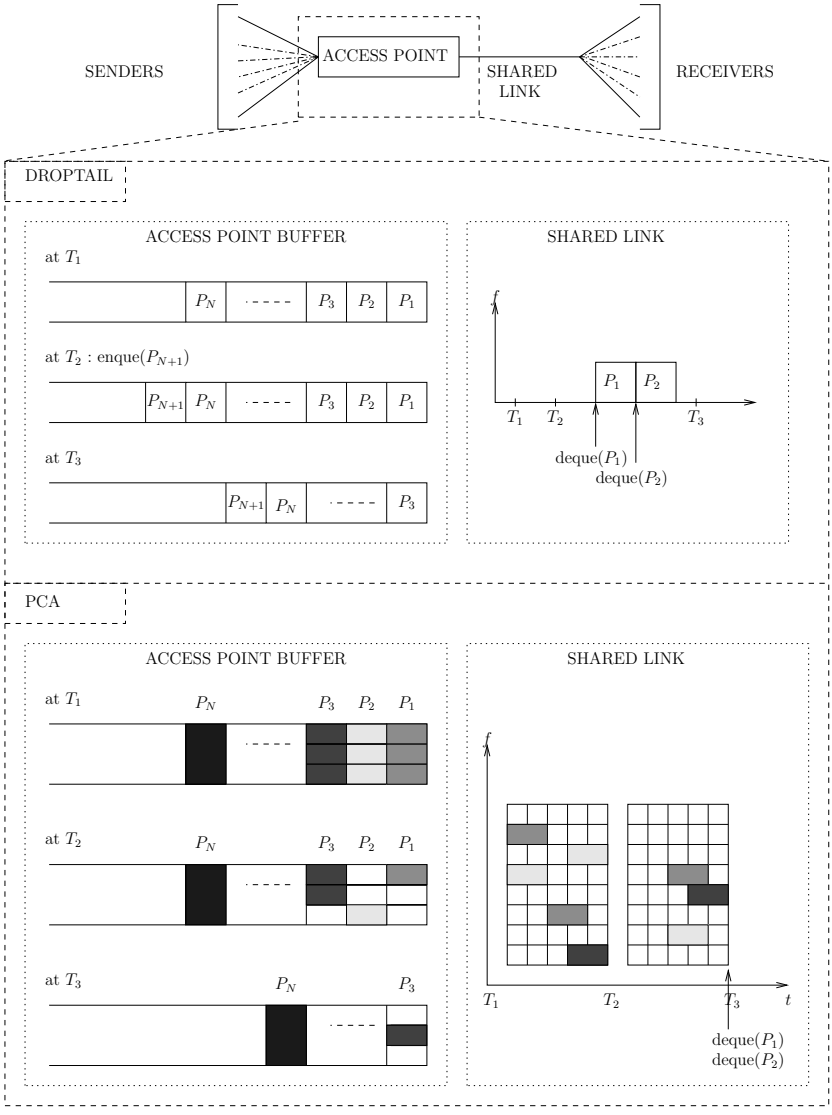

Fig. 2. Capacity allocation: `enque()` and `deque()`

– `used_slot_frame_appl_rnd`: number of slots that the packet's flow will use in the next frame.

Applications list This linked list is used to collect information relative to the currently active applications. It tracks all the open connections and maintains information about the last transmitted datagrams.

– `appl_id`: identifier of the application;
– `pkt_seq`: sequence number of the last packet transmitted;

– `last_time_out`: time when the last packet of the given application has been sent.

3.2 `enque(Packet *p)` method

The `enque()` method is called when the network layer passes a packet down to PCA. It registers the packet and its attributes for consideration in the capacity distribution process. Figure 3 summarises the operation of this function.

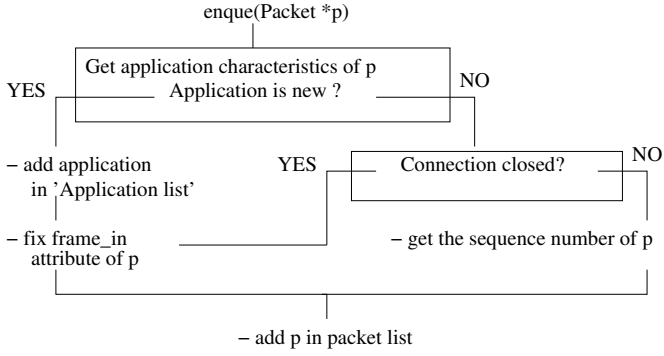

Fig. 3. `enque()` method flowchart

Before enqueuing a packet, it verifies whether the connection needs to be established and the application added in the applications list. If the application is not new, it checks whether the connection is still open. It then adds the packet with its characteristics to the packets list. If `pkt_seqno`=0, it sets the `frame_in` attribute of the packet (first frame where data from this packet starts being transmitted) depending on the access method. `frame_out` is initially set to ∞, and later updated by the `adaptBitNextFrame()` method described in the next section.

3.3 `deque()` method

The `deque()` method emulates arrival of a new frame at the receiver. It is called by a timer every T_F. It loops over the packet list, forwards the packet for which `frame_out` is the current frame to the receiving node and updates the transmission progress of the other datagrams by adjusting their attributes. Figure 4 details how the number of bits to transmit on the next frame is calculated in the `adaptBitNextFrame()` function. This process has been introduced to take care of:

– fair distribution of the capacity with dedicated access methods;
– determination of erasure probability with random access methods (depending on the load of the link and methods performance as detailed in Table 1);

– adaptation of the packet transmission to ensure that flows send their packets in the order they have been received.

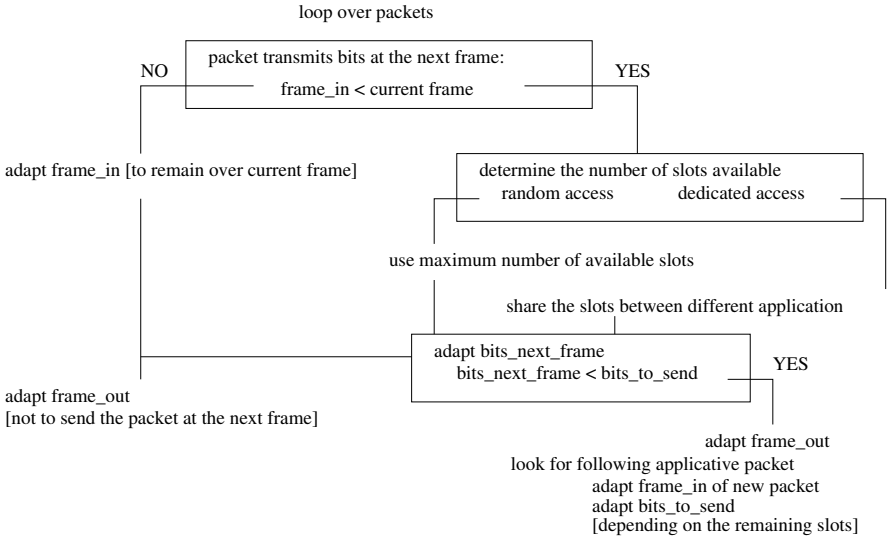

Fig. 4. `adaptBitNextFrame()` method flowchart

To do so, `adaptBitNextFrame()` updates the values of `bits_next_frame`, `used_slot_frame_appl_rnd` and `frame_out` as follows. At frame F, for each packet where `frame_in`$< F$, we compute $B_{\text{remaining}}$=`bits_to_send`-`bits_next_frame`, which corresponds to the data which remains to be transmitted. If $B_{\text{remaining}} > 0$, the packet is left in the queue; `bits_next_frame`, which corresponds to the data that will be transmitted at frame $F+1$, is determined depending on the access method (as well as the number of slots which it will use, `used_slot_frame_appl_rnd`) and `bits_to_send` is set to $B_{\text{remaining}}$. If $B_{\text{remaining}} \leq 0$, `frame_out` is set to $F+1$; the next packet for that application is then found in the packet list, its `frame_in` is set to F and $B_{\text{remaining}}$ is subtracted from its `bits_to_send`.

3.4 Limitations and extensions

PCA can be used to conduct large studies on (MF-)TDMA schemes, however it has some limitations. First, the performance of random access methods depends on the signal-to-noise ratio of the specific link between one receiver and the access point. It is currently assumed that this value is the same for all receivers, but this can be easily lifted by adapting the receiver-to-SNR mapping code. Second, PCA does not consider prioritization between flows. Nonetheless, this could be achieved by flagging packets at higher layers and inspecting these flags in `deque()` and `adaptBitNextFrame()` functions.

The development of this module has been driven by (MF-)TDMA specifications. However, it can easily be extended for other similar access methods (time and/or frequency multiplexing) by adapting the `adaptBitNextFrame()` method to reflect the specific data scheduling scheme of the desired technique. Also, in the current implementation, it was considered that one flow could only send a limited amount of data per RA block. This quantity can be adjusted in the simulation parameters (through `sizeSlotRandom_` and `nbSlotRndFreqGroup_`).

4 Use case example

In this section, we detail the principal parameters of the `DropTail/PCA` queuing policy and illustrate them with an example in the context of DVB-RCS2.

4.1 Parameters

The parameters are set following the standard NS-2 fashion:

```
Queue/DropTail/PCA set <PARAMETER> <VALUE>
```

The following parameters have to be specified prior to starting a simulation:

- `cutConnect_`: time after which the connection between the gateway and the user is closed (in seconds);
- `esN0_`: signal-to-noise ratio of the channel in dB (for random access methods performance);
- `switchAleaDet_`: sequence number at which the access method switches from random to dedicated;
- `frameDuration_` (T_F): duration of a frame;
- `nbSlotPerFreq_` (N_S): number of time slots per frequency;
- `sizeSlotRandom_` (N_{data}): useful number of bits that can be sent on one RA block (*i.e.*, where random access methods are introduced);
- `sizeSlotDeter_` (N_{data}): useful number of bits for each time slots where dedicated access methods are introduced;
- `rtt_`: two-way link delay (in seconds);
- `freqRandom_` (F_R): number of frequencies used for random access;
- `nbFreqPerRand_` ($(F_R \times N_S)/N_{\text{ra}}$): number of frequencies comprised in an RA block;
- `freqDeter_` (F_D): number of frequencies used for dedicated access;
- `maxThroughtput_`: maximum authorized throughput for one given flow (in Mbps);
- `nbSlotRndFreqGroup_` (N_{block}): number of blocks a PLDU is split into for distribution in one RA block;
- `boolAntennaLimit_`: boolean whether one transmitter has one or $F_R + F_D$ antennas.

In order to introduce PCA, the link between two nodes N1 and N2 (N1 is the access point node) can then be defined as:

```
$ns simplex-link $N1 $N2 $bandwidth [$rtt_ / 2]  \
DropTail/PCA $random_access_file_performance
```

where `$random_access_file_performance` is the name of the file containing information about random access performance laid out as in Table 1.

4.2 Use case in the context of DVB-RCS2

We illustrate the results obtained with this PCA module in the context of DVB-RCS2. DVB-RCS2 is the return link on which the home user transmits data to the satellite gateway: the satellite link is shared between the different users. There is a recent interest in enabling the home user to transmit data (*e.g.* for web browsing or email exchange) through this channel. However, there is contention about which access method is best suited for this use. PCA allows to provide insight on this question by evaluating transport layer performance with various access method proposals.

We consider three different cases: (1) a dedicated access method and random access methods ((2) CRDSA [7] and (3) MuSCA [8]). We use input for each random access method that follows the format illustrated in Section 2. We base the choice of parameters on specifications defined in [9] and present them in Table 2.

Table 2. Use case simulation parameters

Parameters	Access method		
	Dedicated	Random (CRDSA)	Random (MUSCA)
`cutConnect_`	3	3	3
`esN0_`	5	5	5
`switchAleaDet_`	0	∞	∞
`frameDuration_`	0.045	0.045	0.045
`nbSlotPerFreq_`	40	40	40
`sizeSlotRandom_`	xx	613	680
`sizeSlotDeter_`	920	xx	xx
`rtt_`	0.5	0.5	0.5
`freqRandom_`	0	100	100
`nbFreqPerRand_`	2.5	2.5	2.5
`freqDeter_`	100	0	0
`maxThroughtput_`	1 Mbps	1 Mbps	1 Mbps
`nbSlotRndFreqGroup_`	xx	3	3
`boolAntennaLimit_`	1	1	1

We consider two nodes in NS-2. The first node transmits a various number of FTP flows to the second node. This allows to study the transport layer performance (no data-starved sender). The size of IP packets is 1500 bytes, and the

queue at the sender is large enough not to be overflowed. We use the Linux implementation of TCP Reno, with SACK options. The simulation time is 20 seconds.

In Figure 5, we show the total achievable throughput measured on the shared link level. This shows that dedicated access methods support more load on the network whereas random access methods, which PCA allows us to study, experience lower throughput with increasing loads.

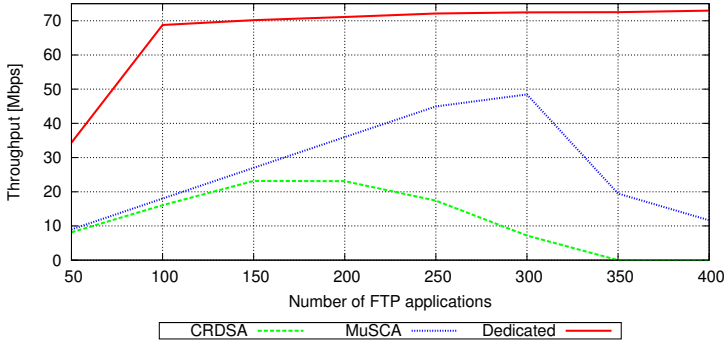

Fig. 5. Throughput depending on the load of the network

Figure 6 illustrates the evolution of the packet sequence numbers of one given FTP application during the first seconds of the simulation. We only show the results with MuSCA as the random access method; the performance with CRDSA is qualitatively similar. Contrary to the previous metrics, random access methods seem to perform better. Indeed, thanks to a faster connection establishment, random access methods transmit the first packets faster than dedicated access methods. However, with dedicated access, the time needed to effectively transmit one packet is smaller: as detailed in Table 2, more bits can be sent on one slot (*i.e.*, `sizeSlotDeter_>sizeSlotRandom_`).

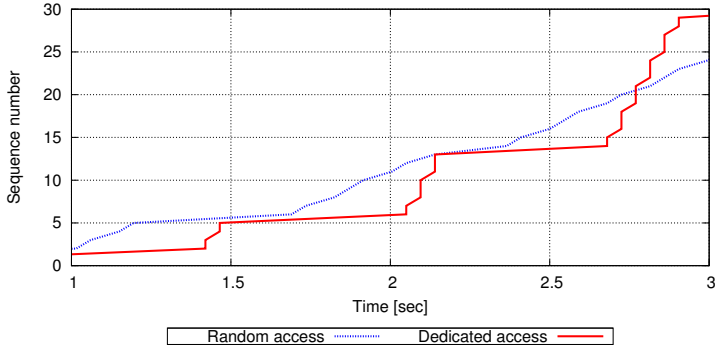

Fig. 6. Sequence number evolution

These preliminary simulations present a first evaluations of the impact of access methods on TCP performance in the context of DVB-RCS2. We expect to explore and analyze this use case further.

5 Conclusion and future work

In this article, we presented PCA, a module for NS-2 that enables to emulate channel access methods for the evaluation of the interaction with transport protocol mechanisms. This module considers time and/or frequency multiplexing access methods and can be used in contexts where the capacity of the channel is shared among multiple users. We detailed the main components and their operation as well as the parameters required to configure the module.

PCA is useful to assess the impact of medium access strategies on transport performance, especially for satellite links. It was initially developed with MF-TDMA specifications in mind, however we also indicated where extensions should be made to accommodate other similar technologies (*e.g.*, OFDMA).

We are currently using this module to investigate random access performance in the context of DVB-RCS2. Also, we plan to release the module as open source soon on ISAE webpage.

References

1. K. Fall, K. Varadhan. The ns Manual (formerly ns Notes and Documentation). VINT Project. 2009
2. The Network Simulator NS-2 — NIST add-on — IEEE 802.16 model (MAC/PHY). Technical Report. NIST. 2009
3. P. Wu, T. Tsai, Y. Kao, J. Hwang, C. Lee. An NS2 Simulation Module for Multicast and OFDMA of IEEE 802.16e Mobile WiMAX. See: `http://edith.cse.nsysu.edu.tw/wimax/wimax.htm`.
4. T. Gayraud, L. Bertaux, P. Berthou. A NS-2 Simulation model of DVB-S2/RCS Satellite network. 15th Ka Band Conference. 2009.
5. R. Secchi. DVB-RCS(2) for ns-2. `http://homepages.abdn.ac.uk/r.secchi/pages/dvbrcs_ns2.htm`. Last accessed 12 November 2012.
6. A. Gurtov, S. Floyd. Modeling Wireless Links for Transport Protocols. ACM Computer Communication Review (CCR). 34(2):85-96, 2004.
7. E. Casini, R. De Gaudenzi, Od.R. Herrero. Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks. IEEE Transactions on Wireless Communications. 2007.
8. H.-C. Bui, J. Lacan, M.-L. Boucheret. An Enhanced Multiple Random Access Scheme for Satellite Communications. Wireless Telecommunications Symposium (WTS 2012). 2012.
9. Digital Video Broadcasting (DVB); Second Generation DVB Interactive Satellite System; Part 2: Lower Layers for Satellite standard. Draft ETSI EN 101 545-2 V1.1.1. 2011.