Demonstrating QoE-aware 5G Network Slicing Emulated with HTB in OMNeT++

Marija Gajić[‡*], Marcin Bosk[†*], Susanna Schwarzmann[§], Stanislav Lange[‡], Thomas Zinner[‡]
[†]Technical University of Munich, Department of Informatics, Chair of Connected Mobility
[‡]NTNU - Norwegian University of Science and Technology, Department of Information Security and Communication Technology
[§]Technische Universität Berlin, Chair of Internet Network Architectures
[‡]{marija.gajic|stanislav.lange|thomas.zinner}@ntnu.no, [†]bosk@in.tum.de, [§]susanna@inet.tu-berlin.de

***Abstract*—Today's networks support a great variety of services with different bandwidth and latency requirements. To maintain high user satisfaction and efficient resource utilization, providers employ traffic shaping. One such mechanism is the Hierarchical Token Bucket (HTB), allowing for two-level flow bitrate guarantees and aggregation. In this demo, we present *HTBQueue* - our OMNeT++ realization of the HTB, and show how the module can be used for mimicking 5G network slicing and analyzing its effect on network services.**

I. Introduction

Nowadays networks provide support for a multitude of heterogeneous services. Network slicing promises to address the challenges of heterogeneity in terms of scalability, performance, and resource efficiency by providing coarse-grained differentiation, homogeneity within slices, and multiplexing gain. In this context, slices are provided as virtual networks with appropriate levels of resource isolation and sharing. As the process of standardization of network slicing is still ongoing, related hardware capabilities, implementation details, and resulting performance are not entirely apparent yet. Hence, simulation is a necessary tool for assessing the technology's potential under various assumptions regarding its behaviour, degree of isolation, and comparison against other mechanisms.

The OMNeT++/INET discrete event simulation framework is particularly popular since it offers packet-level granularity and many building blocks for queuing functionality, as well as different application and device types. With recent additions [1], it now supports advanced traffic shaping mechanisms such as the Hierarchical Token Bucket (HTB).

The HTB constitutes a popular traffic shaping mechanism for several reasons. First, its two-level bit rate limiting capabilities allow setting an *assured* (guaranteed, R_a) bit rate as well as a *ceiling* (maximum, R_c) bit rate for each flow. It aligns well with modern networking architectures in which the general notion of two-level traffic shaping is prevalent. For instance, the 3GPP [2] standard for the Next Generation Radio Access Network (NG-RAN) in 5G defines an R_a and an R_c for each flow. Second, the HTB enables flexible Quality of Service (QoS) differentiation and can therefore be used to enforce per-flow QoS policies [3]. This differentiation can happen at various granularity levels, e.g., based on service type, IP addresses, or more complex matching criteria. Third, the core structure of HTB is a hierarchical tree built with three node types: root, inner, and leaf representing the entire link, traffic groups, and hardware queues, respectively. This structure allows "borrowing" of excess capacity on the margin between guaranteed and maximum bit rates [1].

Combined with its high rate conformance [4], the outlined characteristics make the HTB suitable for emulating network slicing [5]. In particular, inner nodes in the HTB tree can correspond to slices with appropriate bit rate settings. Depending on the desired trade-off between isolation and resource sharing, HTB can also enable borrowing between various slices.

In this demo paper, we present our *HTBQueue* module for the INET Framework[1] that provides all discussed features and is based on the Linux HTB implementation[2]. Our demonstration shows the outlined slicing use case in a multi-application context. Specifically, we highlight how the *HTBQueue* module can be used to mimic 5G network slicing and illustrate potential benefits of employing slicing by comparing the achieved bitrate and Quality of Experience (*QoE*), shown as Mean Opinion Score (*MOS*), with a baseline without slicing. We extend the OMNeT++/INET simulation framework with capabilities for (a) live monitoring of flow bitrates and their *QoE* using MySQL and Grafana and (b) interacting with the flow behavior via the monitoring GUI.

II. Demonstration Scenarios and Presentation

To demonstrate the functionality of the *HTBQueue*, we use a simple topology consisting of one host and one server that are directly connected via a 10 Mbit/s link. Low capacity of the link (and thereafter low number of active flows) was chosen for the sake of better ability to analyze individual flow behavior. The approach is well-suited also for larger-scale scenarios.

The host runs three different applications: Video on Demand *(VoD)* as well as TCP- and UDP- based file download *(FD)* applications. The HTB is applied on both sides of the link. All experiments include nine flows - three flows per application type. Each flow corresponds to a leaf in the HTB tree. Table I shows the R_a and R_c of the flows as well as their application

[*]Marcin Bosk and Marija Gajić contributed equally to this paper.

[1]https://inet.omnetpp.org/Introduction.html, Accessed 11.01.2022
[2]http://luxik.cdi.cz/~devik/qos/htb/, Accessed 11.01.2022

TABLE I
R_a AND R_c CONFIGURATION SETTINGS IN MBPS

Scenario	*VoD* Flows 0-2		TCP *FD* Flows 3-5		UDP *FD* Flows 6-8		*VoD* Slice		TCP *FD* Slice		UDP *FD* Slice		Root	
	R_a	R_c	R_a	R_c	R_a	R_c	R_a	R_c	R_a	R_c	R_a	R_c	R_a	R_c
S1	0.5	3.5	0.5	1.5	0	3	/	/	/	/	/	/	10	10
S2	0.5	3.5	0.5	1.5	0	3	5	10	4	4	0	3	10	10

mapping for all scenarios. The rates were chosen so that *MOS* is never below 3.5, corresponding to a ”fair” service quality.

We introduce two demonstration scenarios: (a) with QoS Flows and no slicing (S1) and (b) with both QoS Flows and slicing (S2). In S1, we only have leaves directly connected to the root with no inner nodes. In S2, we keep the HTB tree structure from S1, but add three inner nodes each considered as a network slice corresponding to one application type. Each slice has (at most) three active flows of the same application type. TCP *FD* slice has its R_c equal to the R_a, meaning that its network resources are isolated. UDP *FD* is considered as *best-effort*, therefore it has an R_a of 0 Mbps. The *VoD* slice has an R_a of 5 Mbps and R_c of 10 Mbps, meaning that video flows can borrow unused resources. Therefore, the excess bandwidth is shared between the *VoD* and UDP *FD* flows.

We connect the simulation output to a database and use Grafana to monitor bitrates and *QoE* indicators in real-time for all active flows. Figure 1 shows the setup and demonstration elements. We start two OMNeT++ simulations simultaneously, with only *VoD* flows, and then gradually include TCP and UDP *FD* clients. We add (and remove) the flows using the interactive GUI which also allows for real-time changes of the UDP *FD* flows' bitrates. This allows us to (a) compare scenarios with and without slicing, (b) analyze the impact that slicing has on the *QoE* of *VoD* and *TCP FD*, and (c) demonstrate the high rate conformance guaranteed in general by the HTB.

With the demonstration scenarios, we show how HTB can be used to emulate network slicing and analyze its effects on the quality of the services. In our case, the inner nodes (slices) were the only difference between the scenarios, and resulted in significant improvement of *MOS* for *VoD*, while preserving *MOS* for TCP *FD*. We did not analyze the *QoE* of UDP traffic, since it was considered as *best-effort* (no guarantees apply).

In addition to showing how HTB can be used to emulate and analyze the effects of slicing on the quality of the delivered services, we also show the high rate conformance of the HTB. The latter is confirmed by checking that none of the flows exceeds their R_c, their R_a are satisfied, and that excess bandwidth is shared fairly between the eligible flows.

Applicability of the *HTBQueue* is not limited to the scenarios depicted here. It can be extended to enable emulation of various, complex slicing approaches (e.g. uni-directional inter-slice borrowing and network slicing with flow prioritization).

III. CONCLUSION / DISCUSSION

Due to a wide range of capabilities and various building blocks, the OMNeT++/INET framework is one of the most commonly used simulation frameworks in the networking

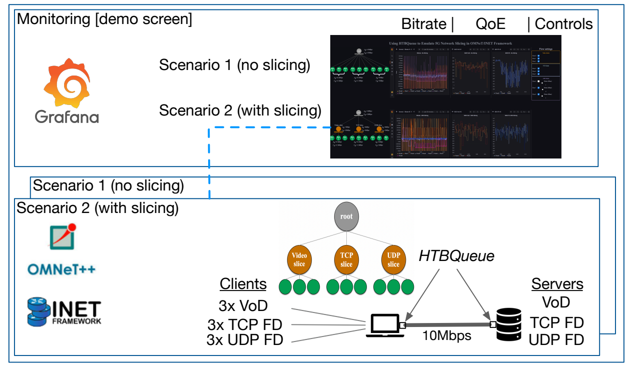

Fig. 1. Setup for live monitoring and demonstration. Results from OMNeT++ simulations are imported to Grafana and shown within an interactive GUI.

domain. With the recent additions, the framework can also be used to mimic the behavior of network slices.

This demo paper discusses potential use cases of the *HTBQueue* module, together with demonstration scenarios featuring three different applications sharing a physical link. In order to illustrate the principles of *HTBQueue* in detail and provoke discussions, a GUI is presented allowing for real-time interaction with flows, and visualization of the effects these changes have on the HTB dynamics. Real-time monitoring of flow bit rates and *MOS* for *VoD* and TCP *FD* is realized with the Grafana tool. We compare flows' behaviour between two scenarios, showcasing HTB's slicing capabilities using inner nodes and a baseline without slicing. We show how inner nodes can be used to emulate network slices, and how slicing can contribute to improving *MOS* of the services of interest. From the results, we also confirm the expected high rate conformance of the HTB in our implementation. Finally, we make the presented work[3] and *HTBQueue* implementation[4] publicly available to the research community.

ACKNOWLEDGMENT

This work has been funded by the German BMBF Software Campus Grant “BigQoE” (01IS17052) and was supported by EC H2020 TeraFlow (101015857).

REFERENCES

[1] M. Bosk, M. Gajić, S. Schwarzmann, S. Lange, and T. Zinner, “HTBQueue: A Hierarchical Token Bucket Implementation for the OMNeT++/INET Framework,” in *OMNeT++ Community Summit*, 2021.

[2] 3rd Generation Partnership Project (3GPP), “NR; NR and NG-RAN Overall description; Stage-2,” Technical Specification (TS) 38.300, Mar. 2021, release 16.5.0.

[3] J. Yan and D. Jin, “Vt-mininet: Virtual-time-enabled mininet for scalable and accurate software-define network emulation,” in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, 2015, pp. 1–7.

[4] A. Saeed, N. Dukkipati, V. Valancius, T. Lam, C. Contavalli, and A. Vahdat, “Carousel: Scalable traffic shaping at end-hosts,” in *ACM SIGCOMM 2017*, 2017.

[5] M. Bosk, M. Gajić, S. Schwarzmann, S. Lange, R. Trivisonno, C. Marquezan, and T. Zinner, “Using 5G QoS Mechanisms to Achieve QoE-Aware Resource Allocation,” in *International Conference on Network and Service Management (CNSM)*. IEEE, 2021.

[3]Code available at: https://github.com/m-bosk/htb-demo
[4]Implementation available at: https://github.com/m-bosk/inet/tree/topic/htb