Multi-timescale fairness for heterogeneous broadband traffic in access-aggregation networks*

Szilveszter Nádas[1], Balázs Varga[1], Illés Horváth[2][0000−0003−2724−1851], András Mészáros[3], and Miklós Telek[3][0000−0001−9600−6084]

[1] Ericsson Research, Hungary {szilveszter.nadas,balazs.a.varga}@ericsson.com
[2] MTA-BME Information Systems Research Group
horvath.illes.antal@gmail.com
[3] Dept. of Networked Systems and Services, Budapest University of Technology and Economics, Hungary
{meszarosa, telek}@hit.bme.hu

Abstract. We propose a throughput value function (TVF) based solution for providing multi time-scale (MTS) fairness for broadband traffic in access-aggregation networks. The primary goal of MTS fairness is a dynamic control of resource sharing that considers the usage history of the broadband connection. We present a flow level description of the multi time-scale throughput value function (MTS-TVF) based resource sharing. We provide dimensioning guidelines in traffic aggregation scenarios and present its simulation-based performance analysis.
In the performance analysis, our focus is on overloaded systems involving both low load users (with temporally active traffic) and high load users (with heavy traffic, e.g. continuous multiple downloads). We find that the Quality of Experience (QoE) of low load users significantly increases when using MTS-TVF and it becomes similar to that of a lightly loaded system, while the change in the QoE is minimal for high load users.

Keywords: fairness, multiple timescales, core stateless, resource sharing, throughput value function, QoS, fluid model

1 Introduction

Resource sharing among traffic flows has remained an area of interest in networking research. Fairness is usually interpreted as equal (or weighted) throughput [1] experienced by flows. By definition, throughput is a measure derived from total packet transmission during a time interval, the length of which is called *timescale*. With the introduction of 5G for mobile and Fiber-To-The-Home (FTTH) for fixed Internet access, the capacity of the last mile has significantly been increased, resulting in much higher load on the access-aggregation networks than before, thus moving bottlenecks from the edge to routers in the aggregation.

* Partially supported by the OTKA K123914 and the TUDFO/51757/2019-ITM grants.

In such a network, the congestion controls used by the flows and the (propagation) round trip times (RTTs) are much more heterogeneous than in data centers and other closed enterprise networks. To handle the increased load and to serve these high-speed bottlenecks, a new node functionality is needed, where controlling resource sharing is an important design goal.

Most current resource sharing control methods are based on throughput measured only on a short timescale (e.g. RTT). For bursty traffic, throughput measured on multiple timescales (e.g. RTT, 1 s, 10 s, session duration) usually results in different values. From the end-user perspective, network performance is better described by throughput during the active periods of a source as opposed to the general case when active and inactive periods are both considered. Taking the history of inactivity into account is advantageous for short transmissions like web downloads or initial buffering of adaptive video streaming. A comprehensive recent survey on fairness [1] states that "getting a scheme to instantly serve web flows for improved performance while maintaining fairness between other persistent traffic remains an open and significant design problem to be investigated." For elastic flows, [2] argues that "highly unequal flow rates have led to flow completion times considerably better than with equal flow rates, indeed nearly as good as they were before the contending long-running flow was introduced".

The literature on these two main concepts, resource sharing control methods based on multi-timescale (MTS) throughput measurement and throughput value function (TVF) is limited. A solution for providing MTS fairness, referred to as Multi-Timescale Bandwidth Profile (MTS-BWP), was introduced in [3]. It defines and implements multi-timescale fairness for a network scenario with few sources with well-defined traffic behaviour. MTS-BWP applies several token buckets per Drop Precedence representing increasing timescales of throughput measurements. MTS-BWP implementation complexity increases with the number of drop precedences, which may out-weigh its advantages when fine grained control is needed. The concept of using TVFs for fine-grained resource sharing based on short timescale throughput measure was introduced in [4], and the MTS extension of the TVF is discussed in [5]. The packet level behaviour of multi time-scale throughput value function (MTS-TVF) is considered and evaluated in [5], using a packet level simulation tool. Due to the inherent complexity of the packet level behaviour, the applicability of the packet level analysis is restricted to rather simple scenarios, much smaller than the aggregation scenario considered in this paper. Practically, only the evaluation of the initial transient of a small network scenario is feasible with the packet level simulator, which is a single jump in the fluid simulator. A contribution of the paper is the introduction of the fluid model of MTS-TVF resource sharing and an associated fluid simulation tool, which makes it possible to evaluate the performance of MTS-TVF based resource sharing in real networking aggregation scenarios.

To really utilize the advantages of MTS-TVF, flexible but explicit dimensioning guidelines are needed. The main contribution of this paper is a design approach for MTS-TVF resource sharing, that provides such dimensioning guidelines to achieve MTS fairness goals for heterogeneous broadband traffic in access-aggregation network. The benefit of using the proposed MTS-TVF resource sharing is evaluated by comparing its behaviour with the single timescale TVF (STS-TVF) based one, and the TCP fairness based one.

The rest of the paper is organized as follows. Section 2 gives an overview of TVF based resource sharing. Section 3 introduces multi-timescale fairness. Section 4 describes a fluid model of the proposed resource sharing method that will be used for dimensioning. Section 5 provides dimensioning guidelines for specific goals. Section 6 provides an approximate analysis of the system based on analytic calculations. Section 7 provides numerical results, and Section 8 concludes the work.

2 Overview of STS-TVF resource sharing

In a very high level view, TVF determines how the resources are shared between users with different bandwidths. The STS-TVF resource sharing [4] extends the idea of core stateless resource sharing solutions like [6,7] by marking each packet with a continuous value called Packet Value (PV). The main goal in a network element is to maximize the total aggregate PV of delivered packets. The resource sharing procedure is composed of two phases: 1) Packet marking at network edge; 2) Packet scheduling and dropping based on the PV in the middle of the network.

1) The goal of packet marking is to assign a PV to each packet based on the operator policy and the traffic rate R of the traffic source node (represents e.g. a subscriber and referred to as *node* in the sequel). The PV represents the potential of the packet to get through the network, but the transmission probability also depends on the congestion level of the network. If the network is highly congested packet with high PVs might be dropped, while in case of moderate congestion even packets with low PV get through.

To achieve this goal, packets are marked at the edge of the network by using the resource sharing policy of the operator described by a TVF (denoted by $TVF(.)$). The marker assigns *random* PVs to packets from a proper TVF and bandwidth dependent distribution, such that the rate of packets of the given node with PV larger than x is $TVF(x)$.

The packet marker is implemented as follows. Generated traffic is measured on a single time scale: when the measured rate is R, the assigned PV is $TVF(x)$, where x is a uniformly distributed sample in $[0, R]$. The same packet marking algorithm is applied in all nodes.

We note that operators might have different TVFs for different user classes (e.g. Gold, Silver, Background, Voice), but in this paper we restrict our attention to a single user class.

2) Resource nodes in the middle of the network treat packets solely relying on the carried PVs. Each such node aims at maximizing the total amount of PV transmitted over the shared bottleneck. To this end, scheduling algorithms of different complexity can be used, including algorithms that drop the packet with the smallest PV (even from the middle of the buffer) when the buffer length is too long [4] or using proportional integral controllers (PI-controllers) to determine a PV threshold for packet dropping [8].

Accordingly, at high congestion only packets with high PVs are transmitted, more precisely packets with PV above a given Congestion Threshold Value (CTV) that reflects the actual congestion level. Note that the amount of high and low PV packets in different flows determines the resource share between them. As a result, flows with larger share of high PV packets receive more throughput.

3 Multi-timescale Fairness

For bandwidth profiling, bitrate is typically measured on a short timescale in the order of RTT. It expresses the instantaneous resource usage and it can even capture short bursts. STS-TVF resource sharing uses only this short timescale bitrate to share bandwidth. In other words, the history of nodes is not considered in STS-TVF resource sharing. When our goal is to ensure long-term fairness (or network usage service level agreement) among flows with largely different profiles, bitrates on longer timescales are far more expressive. We assume n timescales $(TS_1, \ldots, TS_n)$ with different lengths: $TS_1 \approx \mathrm{RTT} < TS_2 < \ldots < TS_n$ (e.g. RTT, 1 s, 10 s, session duration). For a flow with an equally spaced, stable traffic, after the time associated with the largest timescale has elapsed, we expect all those rate measurements to be the same, i.e., $R_i \approx R$, $\forall TS_i$, where R_i is the measured rate on timescale TS_i. However, in transient situations, e.g. when transmission starts for a previously silent flow, we expect small rate measurements of long timescales, while R_1 (of $TS_1 \approx \mathrm{RTT}$) may be high (i.e. $R_1 > R_2 > \ldots > R_n$). Rate measurements at shorter timescales react faster to the changes of network conditions, while at longer timescales temporal changes may remain invisible. Similar behavior with the opposite ordering can be seen for a case when a flow stops transmission (or its rate decreases) after a long active period. Implementation of rate measurement algorithms is detailed in [5].

3.1 Multi-Timescale throughput value functions (MTS-TVF)

The goal of MTS-TVF [5] is to control the resource sharing between users with different rates on different timescales. E.g. such that after an inactive period of a subscriber it gets an advantage for its new session compared to subscribers with long time transmission.

Fig. 1 depicts an example with four TVFs: $TVF_4() \ldots TVF_1()$ (disregard the other notations for the time being). The actual throughput value of the packet is derived from the four TVFs based on the actual $R_4, \ldots, R_1$ throughput measurements as follows.

3.2 MTS Rate Measurement-based Marker

The packet marking based on MTS-TVF is a two steps procedure.

- First, a composite TVF (CTVF) is computed based on the actual R_i measurements and the $TVF_i(), i \in \{1, \ldots, n\}$ functions.

Algorithm 1 $\texttt{CTVF}(n, R_i, TVF_i(), i \in \{1, \ldots, n\})$

```
$R'_n = R_n$,
**for** $i = n - 1, i > 0, i = i - 1$ **do**
    $R'_i = \max(R'_{i+1}, R_i)$,        $//$largest of $R_j$, $j \geqslant i$
    $PV_i = TVF_{i+1}\left(R'_{i+1} + \sum_{j=i+1}^{n-1} \Delta_j\right)$,
    $\Delta_i = TVF_i^{-1}(PV_i) - \left(R'_{i+1} + \sum_{j=i+1}^{n-1} \Delta_j\right)$,
**end for**
$CTVF =$
    $\begin{cases} TVF_n(x) & \text{if } x < R'_n, \\ TVF_{n-1}(x + \Delta_{n-1}) & \text{if } R'_n \leqslant x < R'_{n-1}, \\ \vdots & \vdots \\ TVF_1\left(x + \sum_{j=i}^{n-1} \Delta_j\right) & \text{if } R'_2 \leqslant x, \end{cases}$
Return($CTVF$),
```

– In the second step, the computed CTVF function is used as the (single) TVF in STS-TVF resource sharing and the PV is randomly assigned as follows. When the measured rates are $R_1, \ldots, R_n$, the assigned PV is $CTVF(x)$, where x is a uniformly distributed random sample in [0, R_1].

Algorithm 1 (from [5]) implements the marking procedure, where the **for** loop goes downward and the $\sum_{j=i+1}^{n-1}$ summation is idle for $i = n - 1$. In a high level description of the procedure, the first step is to compile a single TVF referred to as CTVF, which is sensitive to the R_i rates and the second step is to apply the "single time scale" packet marking approach from [4]. Fig. 1 and 2 demonstrate the composition of the CTVF using a 4 TS example. The algorithm constructs the CTVF by properly shifting sections from each of the $TVF_i()$ functions to form a single monotone decreasing function. Intuitively, the idea is that a given $TVF_i()$ determines the resource share when the instantaneous rate is between R_{i+1} and R_i (when $R_{i+1} < R_i$). For a detailed explanation of Algorithm 1, we refer to [5], while here we provide some further remarks.

The algorithm does not utilize R_1 (it is only used for computing a PV). When $R_{i+1} < R_i$ holds for $i \in \{1, \ldots, n-1\}$, as it is in Fig. 1, $R'_i = R_i$ and the **for** cycle computes Δ_i values and PV_i values for $i \in \{1, \ldots, n-1\}$.

The composition of CTVF is demonstrated in Fig. 2 (where this shifting changes the appearance of the function in log-log scale). Essentially, the R_i values determine which part of $TVF_i()$ plays role in the CTVF. The higher $R_i - R_{i+1}$ is, the more significant the role of $TVF_i()$ in the CTVF is. When the $R_{i+1} < R_i$ relation is violated for some i, the procedure compiles the CTVF without using the $TVF_i()$ function.

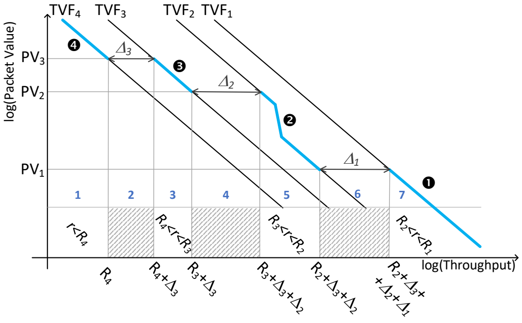

Fig. 1. Example of TVFs for 4-timescales

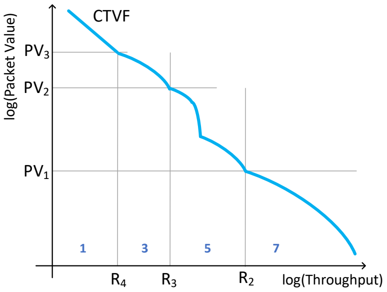

Fig. 2. CTVF composed from the Example in Fig. 1

4 Fluid simulation of MTS-TVF resource sharing

With MTS-TVF we can implement fine grained resource sharing policies. The appropriate evaluation and validation are crucial elements of developing such policies. However, these are not trivial tasks for MTS-TVS. In [5], the behaviour of the MTS-TVF resource sharing is investigated with a packet level simulator, unfortunately this approach, while precise, restricts the analysis to simple scenarios over a rather short time period. In order to gain dimensioning level information, in this section, we introduce a fluid model [9] of the MTS-TVF resource sharing method. Our fluid model assumes idealized resource sharing characteristics, namely instantaneous bandwidth adaptation and no bottleneck buffer: RTT is equal to 0, there is no packet loss, and packets are infinitesimally small. These assumptions correspond to a fluid model where the throughput of each flow adapts instantly to varying conditions. Although at packet level, congestion results in packets lost and re-sent, for dimensioning purposes there is no need for such level of detail and fluid models work properly.

4.1 Fluid model of packet marking and forwarding

In the fluid model the rate measurements are maintained on all timescales ($R_1 \dots R_n$) and based on that the CTVF is computed for all nodes using Algorithm 1.

In the fluid model the PV computation of the nodes and the associated packet dropping at bottleneck link is replaced by the calculation of the ideal resource sharing using the concept of Congestion Threshold Value (CTV), which is computed from

$$\sum_{u \in \{\text{all nodes}\}} CTVF_u^{-1}(CTV) = C, \qquad (1)$$

where C is the capacity of the bottleneck link. Intuitively, (1) states that the instantaneous bandwidth allocated to node u is $CTVF_u^{-1}(CTV)$, and the allocated bandwidth sums up to C. (1) is an implicit equation for the unknown CTV, whose solution is unique due to the strict monotonicity of $CTVF_u(x)$, the simulator computes the solution of (1) by binary search.

4.2 Fluid simulator

In our model, a node can generate multiple flows with different characteristics (e.g. web download, video).

Our fluid simulator keeps track of the state of the system:

- the arrival time and finishing time of each flow;
- the list of all active flows at all nodes along with the remaining flow size;
- the bitrate history of each node (from which $R_1, \ldots, R_n$ is known).

Based on the above information, the simulator calculates the CTV according to (1) and the bandwidth rate allocation for each node and for each flow in the system. The simulator recalculates all information at regular small time intervals Δt, and whenever a flow arrives or leaves the system.

5 Dimensioning guidelines

MTS-TVF is a powerful tool to control resource sharing. However, to fully utilize its capabilities, properly founded dimensioning rules are needed. In this section we propose resource sharing guidelines for providing MTS fairness for heterogeneous broadband traffic in an access-aggregation network. In the dimensioning we only consider the resource sharing for congested system states, because throughput goals are more critical in these cases. Specifically, we consider the following scenario: There are two kinds of nodes, high load nodes (HLNs) and low load nodes (LLNs), competing for the bandwidth (C) of a common bottleneck link. The numbers of HLNs, LLNs and all nodes are N_H, N_L, and $N = N_H + N_L$, respectively. The N_H HLNs are constantly active, resulting in a fully utilized bottleneck link. Consequently, the traffic history of HLNs is the same with relatively high measured throughput on the largest timescale and the load of a single LLN is low enough that a newly active LLN has negligible measured throughput on the largest timescale.

In this scenario we aim to achieve the following dimensioning goals (DGs):

DG1: We want each HLN to achieve at least BW_1 throughput in long-term average.

DG2: If a LLN with inactive history becomes active we aim to allocate it approximately ρ times as much bandwidth as HLNs get, and this allocated bandwidth has to be high enough so that the LLN is able to download *ibs* Mbit in t_1 seconds.

DG3: To avoid extreme fluctuations in the bandwidth allocated, ρ set to be the lowest value satisfying DG2.

DG2 can correspond to, e.g. downloading a web page in t_1 time. In the simulations in Section 7 we consider video downloads. The video starts only when a buffer is filled in the video player (hence the name *ibs*, initial buffer size).

5.1 The proposed MTS-TVF

For DG1 to hold, it is necessary that

$$N_H BW_1 + \ell_L \cdot C < C, \tag{2}$$

where ℓ_L is the total load of LLNs relative to C (i.e. $\ell_L \cdot C$ is the total load of LLNs). To satisfy the DGs we propose to use two timescales with the following TVFs (shown in Figure 3):

$$TVF_1(x) = \begin{cases} 1/x, & \text{if } x < BW_1, \\ \frac{(BW_2 - x)}{(BW_2 - BW_1)BW_1} + \frac{\rho(x - BW_1)}{(BW_2 - BW_1)\rho BW_2}, & \text{if } BW_1 \leqslant x < BW_2, \\ \rho/x, & \text{otherwise,} \end{cases} \tag{3}$$

$$TVF_2(x) = 1/x, \tag{4}$$

where BW_2 is set to $BW_2 = \rho(BW_1 + \epsilon)$, thus $TVF_1(x)$ is strictly monotone decreasing (i.e., invertible, which is needed for the Algorithm 1 to work) in the (BW_1, BW_2) interval, where ϵ is a small positive value ($0 < \epsilon << BW_1$) and $TVF_1(x)$ is linear between BW_1 and BW_2. Cf. Algorithm 1, as the number of timescales is two

$$\Delta_1 = TVF_1^{-1}(TVF_2(R_2)) - R_2. \tag{5}$$

The first timescale is the RTT. The second timescale and ρ are set to satisfy DG2 and DG3:

$$TS_2 = \frac{ibs}{BW_1} \quad \text{and} \quad \rho = \frac{ibs}{BW_1 t_1}. \tag{6}$$

5.2 Intuitive behaviour of the proposed MTS-TVF

For $x < BW_1$, $TVF_1(x) = TVF_2(x)$, and consequently $\Delta_1 = 0$ (see (5) and also Fig. 3). For any node with $R_2 \leqslant BW_1$ (where R_2 is its bitrate on TS_2) the CTVF (purple/dashed curve in Fig. 4) is

$$\widetilde{CTVF}(x) = TVF_1(x). \tag{7}$$

For any node with $R_2 > BW_1$, we assume that ϵ is a rather small value to make TVF_1 invertible (with the given numerical precision) thus we avoid the discussion of the case when $BW_1 < R_2 < BW_1 + \epsilon$. In the case when $R_2 > BW_1 + \epsilon$, $\Delta_1 = (\rho - 1)R_2$ and the CTVF is (brown/solid curve in Fig. 4)

$$\widehat{CTVF}(x) = \begin{cases} 1/x & \text{if } x \leqslant R_2 \\ 1/(x + \Delta_1)\rho & \text{otherwise.} \end{cases} \tag{8}$$

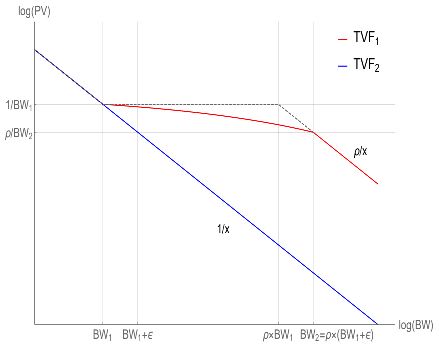

Fig. 3. The proposed TVFs

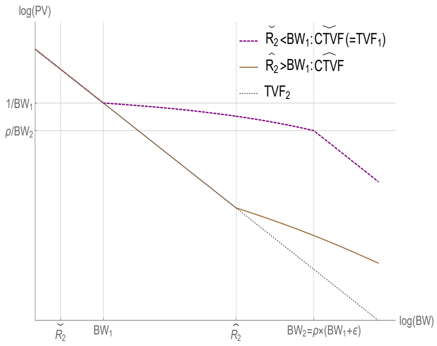

Fig. 4. CTVFs for $R_2 < BW_1$ and $R_2 > BW_1$

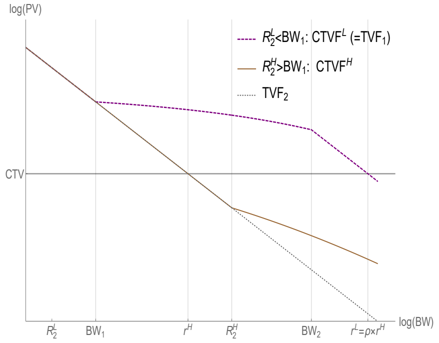

Fig. 5. CTV with $R_2^{(L)} < BW_1$ and $R_2^{(H)} > BW_1$

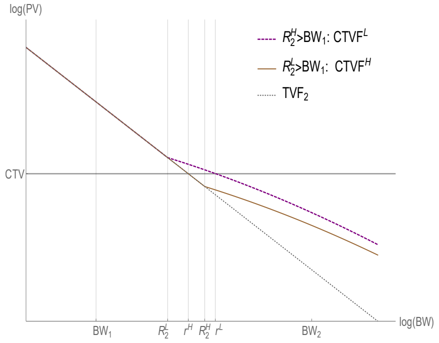

Fig. 6. CTV with $R_2^{(L)} > BW_1$ and $R_2^{(H)} > BW_1$

Let $R_2^{(H)}$ denote the bitrate of HLNs measured on TS_2, $R_2^{(L)}$ denote the bitrate of a chosen active LLN measured on TS_2 and $r^{(H)}$ and $r^{(L)}$ denote the instantaneous bitrates of HLNs and active LLNs, respectively.

Assuming that the system is always close its stationary behaviour and the number of active LLNs is relatively stable, $r^{(H)}$ is close to constant, consequently $R_2^{(H)} \approx r^{(H)}$, and due to (2), $R_2^{(H)} \approx r^{(H)} > BW_1$ the associated CTVF at $r^{(H)}$ is $\widehat{CTVF}(r^{(H)}) \approx 1/r^{(H)} \approx 1/R_2^{(H)}$.

Furthermore, assuming that HLNs with identical history compete for the bandwidth remaining for HLNs, $(1 - \ell_L)C$, the CTV is obtained from (1) as follows

$$\begin{gathered}\textstyle\sum_{u\in\{\text{HLNs}\}} CTVF_u^{-1}(CTV) = (1-\ell_L)C,\\ \Downarrow\\ \widehat{CTVF}^{-1}(CTV) = (1-\ell_L)C/N_H = R_2^{(H)} > BW_1,\\ \Downarrow\\ CTV = \widehat{CTVF}(R_2^{(H)}) = 1/R_2^{(H)} < 1/BW_1.\end{gathered}$$

where $R_2^{(H)} > BW_1$ comes from (2).

Let us assume that a formerly inactive LLN becomes active at a given point in time. Then $R_2^{(L)} = 0$, and the CTVF of the node is $\widetilde{CTVF}() = TVF_1()$. As this LLN is added to the competition for the bandwidth, the CTV increases a bit and the bandwidth of the HLNs decreases a bit, but the CTV remains below $1/BW_1$ and the bandwidth allocated to LLNs is $r^{(L)} = \widetilde{CTVF}^{-1}(CTV) = \rho/CTV$ (from the "otherwise" option of $TVF_1()$), while the bandwidth allocated to HLNs is $r^{(H)} = 1/CTV$ (as it is exemplified in Fig. 5). From this point $R_2^{(L)}$ starts increasing monotonically such that and $r^{(L)} \approx \rho r^{(H)}$ for as long as $R_2^{(L)} < BW_1$. If ibs megabits are downloaded in time $t_{ibs} \leqslant TS_2$ then at t_{ibs} after the LLN becomes active

$$R_2^{(L)} \geqslant \frac{ibs}{TS_2} = BW_1, \tag{9}$$

where we used TS_2 from (6) in the second step. According to (9) and (6), ibs megabits are downloaded with rate

$$r^{(L)} \approx \rho r^{(H)} > \rho BW_1 = \frac{ibs}{t_1}, \tag{10}$$

therefore DG2 will be fulfilled and ibs megabits will be downloaded in less than t_1 seconds. At the limit of the inequality (2), $r^{(L)} = \rho BW_1 = ibs/t_1$, which is just enough to download ibs megabits in t_1 seconds, therefore we set ρ according to (6), which is the lowest ρ to satisfy DG3.

When, due to the high throughput ($r^{(L)} \approx \rho r^{(H)}$) at the beginning of the active period of the LLN, $R_2^{(L)}$ increases above BW_1 the associated CTVF becomes $\widehat{CTVF}(x)$ and the bandwidth allocation modifies according to Fig. 6.

6 Approximate analysis of the stationary behaviour

In general, we expect the system to converge to some stationary behaviour, but calculating the stationary distribution explicitly is infeasible. Instead, we focus on the mean number of active LLNs in the stationary distribution. In this setting, all HLNs are active, and one of the dimensioning parameter is the number of active LLNs.

In this section, we present an approximate calculation based on intuitive assumptions which allow to compute the stationary behaviour and later we evaluate the accuracy of the approximation. The approximate analysis is based on the assumption that all active LLNs have a single flow which started from a perfect node history. Note that even apart from this assumption, the calculations only provide an approximation due to the non-linearity and long memory of the system.

The following calculations are specific to the TVF designed in Section 5, which sharply distinguish the nodes with high and low measured R_i rates, referred to as good and bad history. The active LLNs are divided into the following categories:

- $N_L^{(1)}$ is the mean number of active LLNs with a web flow with good history;
- $N_L^{(2)}$ is the mean number of active LLNs with a web flow with bad history;
- $N_L^{(3)}$ is the mean number of active LLNs with a video flow with good history;
- $N_L^{(4)}$ is the mean number of active LLNs with a video flow with bad history.

Using the notations also from Table 1, we approximate the system behaviour with the following equations:

$$C = (N_L^{(1)} + N_L^{(3)})\rho R_{\mathrm{st}} + (N_L^{(2)} + N_L^{(4)} + N_H) R_{\mathrm{st}}, \tag{11}$$

$$\ell_L \cdot C = (N_L^{(1)} + N_L^{(3)})\rho R_{\mathrm{st}} + (N_L^{(2)} + N_L^{(4)}) R_{\mathrm{st}}, \tag{12}$$

$$\frac{N_L^{(1)}}{N_L^{(2)}} = \frac{ibs_2}{(fs_1 - ibs_2)\rho}, \tag{13}$$

$$\frac{N_L^{(3)}}{N_L^{(4)}} = \frac{ibs_2}{(fs_2 - ibs_2)\rho}, \tag{14}$$

$$\frac{20\%}{80\%} = \frac{\rho N_L^{(1)} + N_L^{(2)}}{\rho N_L^{(3)} + N_L^{(4)}}, \tag{15}$$

where R_{st} is the mean bandwidth allocated to a node with bad history (identical to all nodes with bad history).

(11) corresponds to the fact that the system is always used at full capacity (due to $\ell_L + \ell_H > 1$). According to (12), the entire load of the LLNs is serviced (no discarding at LLNs). (13) and (14) set the ratio of *time* spent in good/bad node history for LLNs, taking into account that the node history changes from good to bad after downloading initial buffer size ibs_2. Finally, (15) sets the web/video ratio of the incoming data.

(11)–(15) leads to a system of linear equations describing the mean stationary behaviour of the system, which can then be compared with actual simulations, done in the next section.

C	1000 Mbps	total capacity
N_L	900	number of LLNs
N_H	100	number of HLNs
	80%/20%	LLNs' video/web data ratio
fs_1	5 MB	web download file size
fs_2	18.75 MB	video download file size
ibs_2	3.125 MB	initial video buffer size
t_1	2 s	initial buffer download time
t_2	30 s	video download time
	30	number of HLN flows per node
	10	maximal number of flows at a LLN
BW_1	5 Mbps	guaranteed throughput of HLNs
ℓ_L	$0.1, 0.2, 0.3, 0.4$	load of LLNs proportional to C

Table 1. Model parameters

	$\ell_L = 0.2$		$\ell_L = 0.4$	
	approx.	sim.	approx.	sim.
slow LLN ($N_L^{(2)} + N_L^{(4)}$)	19.5	17.3	49.4	51.4
fast LLN ($N_L^{(1)} + N_L^{(3)}$)	2.6	3.9	6.9	6.6

Table 2. Number of active LLNs according to simulation and approximate calculation

7 Simulation results

7.1 Simulation setup

The parameters of the considered heterogeneous Broadband traffic scenario of the Access-Aggregation Network is summarized in Table 1. The HLNs have 30 continuously active flows with data to transmit and the LLNs initiate web and video flows according to Poisson arrival processes, whose arrival rate can be obtained from ℓ_L, fs_1, fs_2 and the video/web data ratio of LLNs. The number of flows at a LLN is at most 10. Flows arriving when this limit is reached are dropped.

Based on the simulation runs we check the following requirements:

- if the long-term average throughput of HLNs is larger than BW_1 (DG1);
- if the video flows can fill up the initial buffer of size ibs_2 in time t_1.
- if the full video of size fs_2 is downloaded in time t_2. (The throughput required for this is fs_2/t_2, which is equal to BW_1, which is provided for the nodes even with bad history (see Table 1). So as long as a LLN has exactly one active flow which is a video download, then it is guaranteed to finish downloading in t_2 time).

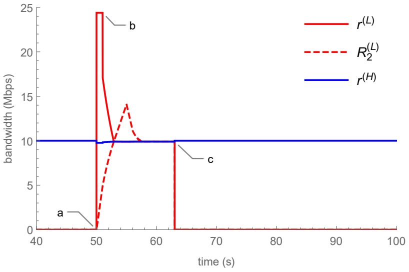

Fig. 7. Node bandwidth time series with 100 bad history nodes

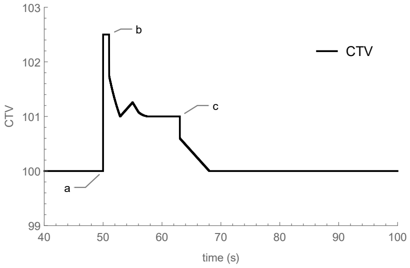

Fig. 8. CTV time series with 100 bad history nodes

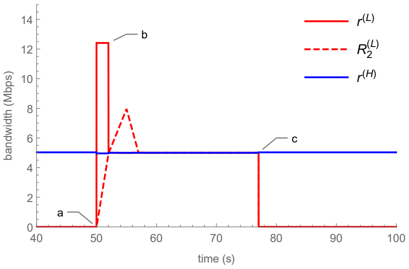

Fig. 9. Node bandwidth time series with 199 bad history nodes

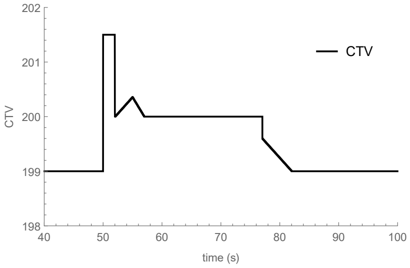

Fig. 10. CTV time series with 199 bad history nodes

We note that web downloads have been included for a more realistic traffic model, but no criteria or dimensioning is included for web downloads in the present paper.

7.2 Numerical analysis of the mean stationary behaviour

To validate the approximate calculations of Section 6, the number of active LLNs was also evaluated by simulation. Table 2 displays the results, grouped according to fast nodes (good history, $N_L^{(1)} + N_L^{(3)}$ with the notation of Section 6) and slow nodes (bad history, $N_L^{(2)} + N_L^{(4)}$). The relative load of LLNs is $\ell_L = 0.2$ or 0.4.

Table 2 shows that the approximate calculation holds up nicely, even for $\ell_L = 0.4$. The maximal value of ℓ_L for which (2) holds is 0.5, as this limit is approached the assumption on the single active flow per LLN is violated with higher and higher probability (hence the larger error for $\ell_L = 0.4$).

7.3 Time series examples

We show two sample realizations. The first system assumes the parameters from Table 1. The second system differs in the number of HLNs and LLNs. In the example shown in Figures 7–8, the system has 100 active nodes, all with bad history. A video flow arrives at 50 s (point (a)) at a LLN with perfect history.

In the setup of Section 7.1, the 100 active nodes with bad history correspond to the 100 HLNs, and the single active LLN corresponds to $\ell_L = 0.01$ as calculated from (11)–(15).

Fig. 7 displays the instant bitrate of the LLN ($r^{(L)}$), its bitrate on the 5 s timescale ($R_2^{(L)}$), and the instant bitrate of a node with bad history ($r^{(H)}$), while Fig. 8 displays the associated CTV of the system. The TVF related reasons for this evolution of the bandwidth sharing and the CTV are discussed in relation with Fig. 5 and 6 in Section 4.

We note that DG1 (or, equivalently, (2)) allows 200 active nodes at most, so the 100 + 1 active nodes is well below this limit, and as a result, all nodes are allocated a relatively high bandwidth. Nodes with bad history get 10 Mbps (instead of the required $BW_1 = 5$ Mbps), and due to the dimensioning of the TVF, the single LLN gets $2.5 \cdot 10 = 25$ Mbps until the initial buffer is filled up (point (b)), which takes less than the required $t_1 = 2$ seconds, and the total download also takes much less time than the required 30 s.

On the other hand, in the example shown in Figures 9–10, the system has 199 active nodes with bad history when a video flow arrives (marked with (a) in the figures) at a LLN with perfect history. Fig. 9 displays the instant bitrate of the LLN ($r^{(L)}$), its bitrate on the 5 second timescale ($R_2^{(L)}$), and the instant bitrate of a node with bad history ($r^{(H)}$), while 10 displays the CTV.

In the setup of Section 7.1, the 200 active nodes correspond to 100 HLNs, 99 LLNs with bad history and 1 LLN with good history corresponding to $\ell_L = 0.5$ as calculated from (11)–(15).

This system is critical in the sense that DG1 and (2) hold with equality now. From (6),

$$\rho = \frac{ibs_2}{BW_1 t_1} = \frac{25\text{Mbit}}{5\text{Mbps} \cdot 2\text{s}} = 2.5,$$

and the single LLN has bandwidth $\rho BW_1 = 12.5$Mbps allocated until the initial buffer size is reached at point (b) (exactly $t_1 = 2$ seconds after (a)); after that, its history reverts back to bad and its bandwidth allocation drops to $BW_1 = 5$ Mbps until the video is finished at point (c) (27 s after (a)).

7.4 Statistical results

In this section, we make a statistical comparison of three congestion control principles: the MTS-TVF is compared with TCP-fair and node-fair. For TCP-fair, each active node is allocated bandwidth proportional to its number of active flows, while for node-fair, each active node is allocated equal bandwidth. Actually, node-fair can be realized by using STS-TVF, see also [8].

Based on the simulator output, we compute the following statistics:

- the node throughput for active periods (periods when there is no traffic at the respective node are excluded) for LLNs and HLNs;

– the flow throughput for video and web download flows at LLNs;
– ratio of flows where the time-to-play (TTP) criterion and total download time criterion is satisfied.

Fig. 11 compares the node throughput of both LLNs and HLNs for the three congestion controls and various load setups according to Section 7.1, with the total low load varying. The figure depicts the node throughput average with a × symbol and the 10% best – 10% worst interval with bars. The main advantage of MTS-TVF is that it offers better performance for LLNs without hurting the long term performance of HLNs. TCP-fair provides flow count proportional throughput, resulting in very poor performance for LLNs. Node-fair and MTS-PPV provide proper prioritization for LLNs at no cost in the performance of HLNs.

Fig. 12 and 13 display video throughput and web download throughput at LLNs. The significantly better throughput provided to web downloads by MTS-TVF is due to prioritizing nodes with good history, which applies to LLNs as the load of an individual LLN is so small that rare arrivals occur mostly at good history. The initial buffer size of video downloads is 3.125 MB = 25 Mbit, and the applied TVF is dimensioned so that the first 25 Mbit of any flow at a node with good history is allocated a high bandwidth (2.5 times larger than for HLNs). The effect on the overall flow throughput is more pronounced for web downloads (5 MB = 40 Mbit) than video downloads (18.75 MB = 150 Mbit) as a relatively larger portion of the flow is downloaded at a high bandwidth.

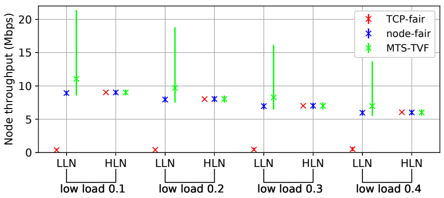

Fig. 11. Node throughput statistics

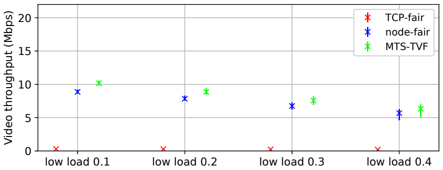

Fig. 12. Video throughput statistics

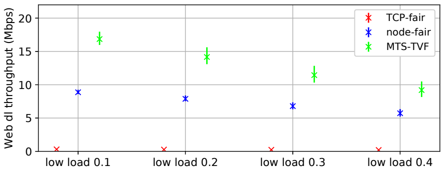

Fig. 13. Web download throughput statistics

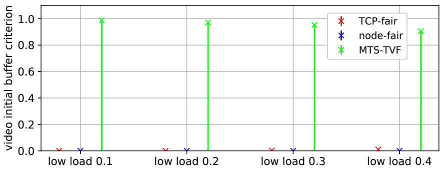

Fig. 14. Video initial buffer criterion at LLNs

Fig. 14 displays the ratio of video download flows which meet the initial buffer criterion of downloading 25 Mbit in 2 s. Note that this criteria is included in the dimensioning guidelines of the MTS-TVF, and accordingly, for MTS-TVF, over 90% of all video download flows meet this criterion even for low load $\ell_L = 0.4$, while for TCP-fair and node-fair, the ratio of flows meeting this criterion is practically zero. This is one of the major advantages of using a properly dimensioned MTS-TVF. The flows that do not meet the criterion for MTS-TVF are due to a flow arriving shortly after another flow at a LLN, with the node history still bad when the second flow arrives. As the number of LLNs increases, the probability of this goes to 0; for 900 LLNs, it still occurs with a small probability (also depending on the total load of LLNs).

Fig. 15 displays the throughput statistics for the initial buffer of video downloads at LLNs, that is, the throughput the flow until the initial buffer is full. Again, MTS-TVF vastly outperforms the other two resource sharing methods. Fig. 16 displays the ratio of video download flows which meet the total download criterion. MTS-TVF was dimensioned so that this criterion is met, and accordingly, for MTS-TVF, it is met for the vast majority of flows. For TCP-fair, this criterion is failed entirely, while for node-fair, it is met for a smaller but still relatively high portion of the flows.

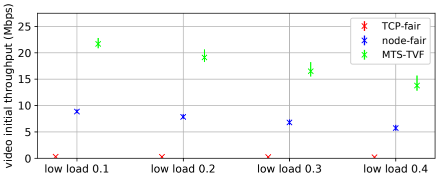

Fig. 15. Video initial buffer throughput at LLNs

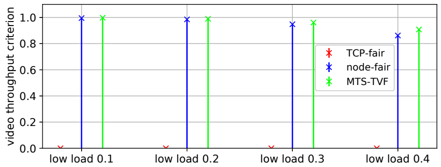

Fig. 16. Throughput criterion for the whole video at LLNs

8 Conclusion

The MTS-TVF based resource sharing introduced in [5], extends the advantages of Multi-Timescale Bandwidth Profile to a wide range of traffic scenarios from only a well defined scenario. It formalizes Multi-Timescale fairness and describes ideal time-series behaviour of resource sharing. However, to utilize the potential benefits of MTS-TVF resource sharing, we need flexible, but explicit dimensioning rules.

In this work we provided a dimensioning method for an access-aggregation network scenario and illustrated the advantages of MTS-TVF using heterogeneous broadband traffic model in access-aggregation network. Using an idealized fluid system model, we showed the time-series behaviour for the working point (CTV) and we showed how the system behaves for several dynamic workloads.

In the studied system, the QoE (assumed based on experienced bandwidth) of low load users significantly increased when using MTS-TVF, effectively making the QoE similar to that of a lightly loaded system, while the effect on the QoE of high load users was minimal. MTS-TVF uses the same policy for both heavy and light loaded users, does not require service identification, and uses well defined policies, therefore it is ideal from a net neutrality perspective.

As an example we showed how the proposed MTS-TVF optimizes the video QoE of moderate loaded users. The current dimensioning concept can be extended for several QoE requirements. Also the same concept can be used for other traffic aggregates, e.g. services, network slices. Additionally, the concept can be combined with the multi-layer virtualization concept, when MTS-TVF is applied for different traffic aggregates simultaneously, e.g. for services, subscribers and network slices at the same time.

References

1. G. Abbas, Z. Halim, and Z. H. Abbas, "Fairness-driven queue management: A survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 324–367, 2016.
2. B. Briscoe, "Per-flow scheduling and the end-to-end argument," Tech. Rep., 2019. [Online]. Available: http://bobbriscoe.net/projects/latency/per-flow_tr.pdf
3. S. Nádas, B. Varga, I. Horváth, A. Mészáros, and M. Telek, "Multi timescale bandwidth profile and its application for burst-aware fairness," in *IEEE Wireless Communications and Networking Conference (WCNC '20)*, 2020.
4. S. Nadas, Z. R. Turanyi, and S. Racz, "Per packet value: A practical concept for network resource sharing," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.
5. S. Nádas, G. Gombos, F. Fejes, and S. Laki, "Towards core-stateless fairness on multiple timescales," in *Proc. of the Applied Networking Research Workshop*, 2019, pp. 30–36.
6. Z. Cao, E. Zegura, and Z. Wang, "Rainbow fair queueing: theory and applications," *Computer Networks*, vol. 47, no. 3, pp. 367 – 392, 2005.
7. I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: A scalable architecture to approximate fair bandwidth allocations in high-speed networks," *IEEE Transaction on Networking*, vol. 11, no. 1, pp. 33–46, 2003.
8. S. Laki, G. Gombos, S. Nádas, and Z. Turányi, "Take your own share of the pie," in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '17. ACM, 2017, pp. 27–32.
9. D. Figueiredo, B. Liu, Y. Guo, J. Kurose, and D. Towsley, "On the efficiency of fluid simulation of networks," *Computer Networks*, vol. 50, pp. 1974–1994, 2006.