GD-Aggregate: A WAN Virtual Topology Building Tool for Hard Real-Time and Embedded Applications (Appendices)

Qixin Wang[*], Xue Liu[†], Jennifer Hou[*], and Lui Sha[*]
[*] Department of Computer Science, University of Illinois at Urbana-Champaign
Email: {qwang4, jhou, lrs}@uiuc.edu
[†] School of Computer Science, McGill University
Email: xueliu@cs.mcgill.ca

Abstract

The convergence of computer and physical world calls for next generation *Wide Area Network* (WAN) infrastructures for hard real-time and embedded applications. Such networks need virtual topologies to achieve scalability, configurability, and flexibility. Virtual topologies are made of virtual links, for which, the state-of-the-art building tool is *Guaranteed Rate server based aggregates* (GR-aggregates). However, common-practice weight assignment scheme couples GR-aggregate *End-to-End* (E2E) delay bound with aggregate's data throughput inverse proportionally. This is undesirable for many hard real-time embedded sensing/actuating applications, whose traffic has small data throughput but requires short E2E delay. We propose *Guaranteed Delay server based aggregates* (GD-aggregates), which allow assigning weights according to *priorities* instead of data throughput. This decouples E2E delay guarantee from data throughput, hence meets the needs of hard real-time embedded applications. In addition, GD-aggregates can be analyzed with simple closed form formulae, and can be easily planned with optimization tools.

I. Introduction

The trend of next generation networks is to converge computers and the physical world. This demands next generation *Wide Area Network* (WAN) for hard real-time and embedded applications (simplified as *Real-Time and Embedded WAN*, or RTE-WAN in the following). Efforts like the Real-Time and Embedded GENI [1] and the Cyber-Physical Systems [2][3][4][5] are calling for design proposals of such RTE-WANs, which must provide scalability, configurability, flexibility, and hard real-time.

To achieve scalability, configurability, and flexibility, the RTE-WAN needs virtual topologies: reconfigurable *virtual links* shall overlay on top of physical links, and hide physical links from higher level networking activities, such as routing, QoS provisioning, and applications; a virtual link may span one or several sequentially connected physical links, occupy part or all of the physical links' bandwidth, and may be reconfigured.

In addition to scalability, configurability, and flexibility, an RTE-WAN virtual link must also guarantee hard real-time E2E delay. Considering all these goals, Section IV discusses different technological alternatives on how to build RTE-WAN virtual links, and shows *Guaranteed Rate server based aggregates* (GR-aggregates) proposed by Sun and Shin [6][7] is the best technology to start with.

Specifically, a virtual link is embodied by the GR-aggregates passing through it. A GR-aggregate (see Section II-A) is basically an aggregation of flows that start from the virtual link's sender-end node, traverse the virtual link's intermediate nodes, and arrive at the virtual link's receiver-end node. A GR-aggregate aggregates flows of similar characteristics, and provides hard real-time E2E delay guarantee. Fig. 1 illustrates the relationship between GR-aggregates and their corresponding virtual link, where a virtual link over two physical links (AC and CB) consists of three GR-aggregates: $F1$, $F2$, and $F3$.

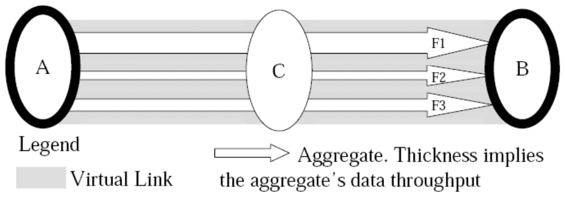

A
C
B
F1
F2
F3
Legend
Virtual Link
Aggregate. Thickness implies the aggregate's data throughput

Fig. 1. A virtual link is embodied by the aggregates passing through it

Originally, the GR-aggregate design is for Internet traffic [6][7]. Though this design fits the Internet context well, when applied to RTE-WAN, GR-aggregates face a new challenge: using common-practice configuration method, a GR-aggregate's E2E delay bound is inverse-proportionally coupled with the aggregate's data throughput. This makes aggregates with small data throughput to have large E2E delay bounds, and vice versa. Such tight coupling is undesirable to many RTE-WAN traffics, especially the sensing/actuating traffic, which usually has small data throughput, but demands short E2E delay guarantee (see Section II-B).

To solve this problem, we propose *Guaranteed Delay server based aggregates* (GD-aggregates). GD-aggregate design decouples (or partially decouples) E2E delay guarantee from data throughput. It also supports priorities, which facilitates real-time system design. GD-aggregates can be analyzed with simple closed form formulae, and can be easily planned with optimization tools.

We evaluate the performance of GD-aggregates in the context of underground mining, a representative RTE-WAN application with typical RTE-WAN traffic. The results show that GD-aggregates support RTE-WAN traffic better than GR-aggregates. However, due to page limits, the performance evaluation is moved to Appendix J of [8].

The rest of the paper is organized as follows: Section II describes the state-of-the-art GR-aggregate design and its coupling problem when using common-practice configuration methods; Section III presents our GD-aggregate solution; Section IV discusses related work; Section V concludes the paper.

II. GR-AGGREGATE AND THE COUPLING PROBLEM

In the following, Section II-A describes the GR-aggregate design [6][7] and its E2E delay bounds; and Section II-B explains GR-aggregate's coupling problem between E2E delay bound and data throughput.

Unless explicitly noted, we assume output queueing [9], and pick symbols consistent with [6][7] whenever possible.

A. GR-aggregate

The GR-aggregate is based on the concept of *Guaranteed Rate* (GR) server proposed by Goyal et al. [10]:

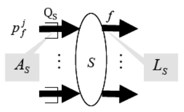


Fig. 2. A Queueing Server

Definition 1 (Guaranteed Rate Server): As shown in Fig. 2, suppose the jth ($j = 1, 2, \ldots$) packet p_f^j of flow f arrives at queue Q_S at time $A_S(p_f^j)$, and leaves server S at time $L_S(p_f^j)$. Suppose the length of p_f^j is ℓ_f^j. We define *Guaranteed Rate Clock* (GRC) of packet p_f^j at server S as

$$\mathrm{GRC}(p_f^j) \overset{def}{=} \max\{A_S(p_f^j), \mathrm{GRC}(p_f^{j-1})\} + \frac{\ell_f^j}{r_f},$$
$$\forall j = 1, 2, \ldots,$$

where constant r_f is called the *Guaranteed Rate,* and $\mathrm{GRC}(p_f^0) \overset{def}{=} 0$. If server S can provide a guaranteed rate r_f for flow f, such that for each packet p_f^j ($j = 1, 2, \ldots$) of f, there is

$$L_S(p_f^j) \leq \mathrm{GRC}(p_f^j) + \alpha, \tag{1}$$

where α is a constant independent of p_f^j, then S is a *Guaranteed Rate* (GR) server. In addition, we call α the *scheduling constant* of GR server S for f. ■

Many existing scheduling servers, such as the *Weighted Fair Queueing* (WFQ) server [11] and the *Worst-Case Fair Weighted Fair Queueing* (WF^{2}Q) server [12], are GR servers [10]. *Without loss of generality, we assume all scheduling servers are WFQ servers*. A WFQ server S has a scheduling constant of $\alpha = \ell^{max}/C$, where ℓ^{max} is the maximum size of packets entering S, and C is the output capacity of S. If a flow f is assigned a WFQ scheduling weight of ϕ_f, its guaranteed rate $r_f = \phi_f C$.

Based on the above notions, the GR-aggregate design runs as follows [6][7]: A GR-aggregate starts by creating an aggregate of flows at a GR server (called "*low-end server*"), and letting the aggregate traverse several GR servers (called "*high-end servers*") as shown in Fig. 3. In the figure, flow f joins other flows at low-end server $S_L^{(1)}$ at Node 1. The output of $S_L^{(1)}$ is an aggregate, denoted as F. High-end server $S_H^{(1)}$ forwards F to Node 2, and $S_H^{(2)}$ forwards F to Node 3, so on and so forth, until F reaches Node K. As the receiver-end of aggregate F, the routing circuit of Node K forwards individual packets of aggregate F according to their original flow headers. Therefore packets of flow f are forwarded to their corresponding output port, where it may join another set of flows at low-end server $S_L^{(K)}$ to create another GR-aggregate F'. We denote $A_{SL}^{(i)}(p)$ and $L_{SL}^{(i)}(p)$ as the time when packet p reaches $Q_{SL}^{(i)}$ and leaves server $S_L^{(i)}$ at Node i respectively; and denote $A_{SH}^{(i)}(p)$ and $L_{SH}^{(i)}(p)$ as the time packet p reaches $Q_{SH}^{(i)}$

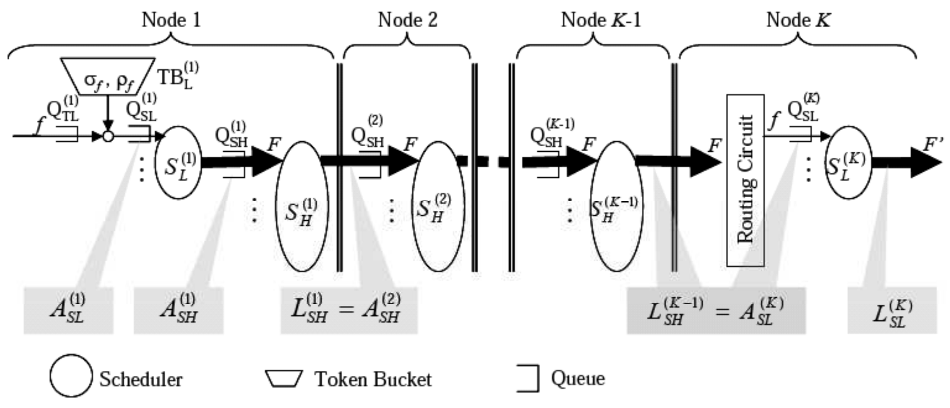


Fig. 3. A GR server based aggregate. Labels in the figure explain the symbols denoting arrival and departure (leaving) time of packets. Routing circuits in Node $1 \sim (K-1)$ are omitted in the figure.

and leaves server $S_H^{(i)}$ at Node i respectively. The *End-to-End* (E2E) delay guarantee of packets traveling through GR-aggregates is restated in Theorem 1 (originally from [6][7]).

Theorem 1 (GR-aggregate E2E Delay): Suppose a flow f joins an GR-aggregate F at $S_L^{(1)}$ as shown in Fig. 3, then the E2E delay d_f^j for any packet p_f^j $(j = 1, 2, \ldots)$ of f satisfies

$$\begin{aligned} d_f^j &\overset{def}{=} A_{SL}^{(K)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \\ &= L_{SH}^{(K-1)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \qquad (2) \\ &\leq [\mathrm{GRC}_{SL}^{(1)}(p_f^j) - A_{SL}^{(1)}(p_f^j)] + (K-2)\frac{\ell_F^{max}}{R_F} \\ &\quad + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{(i)}, \qquad (3) \end{aligned}$$

where ℓ_F^{max} and ℓ_f^{max} are the maximum packet length for aggregate F and flow f respectively; R_F is the guaranteed rate for aggregate F at $S_H^{(1)} \sim S_H^{(K-1)}$; r_f is the guaranteed rate for flow f at $S_L^{(1)}$ and $S_L^{(K)}$; the output capacity of $S_L^{(1)}$ is R_F; the output capacity of $S_H^{(i)}$ $(i = 1, 2, \ldots, (K-1))$ is $C^{(i)}$; $\alpha_L^{(1)}$ is the scheduling constant of GR server $S_L^{(1)}$; and $\alpha_H^{(i)}$ is the scheduling constant of GR server $S_H^{(i)}$. Particularly, if f is constrained by a token bucket (σ_f, ρ_f) before arriving at $Q_{SL}^{(1)}$ at Node 1, where $\rho_f \leq r_f$, then

$$d_f^j \leq \frac{\sigma_f}{r_f} + (K-2)\frac{\ell_F^{max}}{R_F} + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{(i)}. \qquad (4)$$

Proof: See [6][7], particularly Appendix III of [7]. ■

With minor modification to the proof of the above theorem, we can improve the delay bound if packets entering $S_L^{(1)}$ are *Conflict-Free*, which is defined in the following:

Theorem 2 (E2E Delay for Conflict-Free Packet Arrival): In Fig. 3, if packets entering $S_L^{(1)}$ follow *Conflict-Free* pattern, i.e., for any two packets p_1 and p_2 arriving at $S_L^{(1)}$ consecutively, the arrival time $A_{SL}^{(1)}(p_1)$ and $A_{SL}^{(1)}(p_2)$ satisfy $A_{SL}^{(1)}(p_2) \geq A_{SL}^{(1)}(p_1) + \frac{\ell_1}{C_L^{(1)}}$, where ℓ_1 is the length of p_1, $C_L^{(1)}$ is the output capacity of $S_L^{(1)}$, then

$$d_f^j \leq (K-1)\frac{\ell_F^{max}}{R_F} + \sum_{i=1}^{K-1} \alpha_H^{(i)}. \qquad (5)$$

Proof: See Appendix A of [8]. ■

Conflict-Free packet arrival pattern is common to RTE-WAN sensing/actuating end nodes. For example, if an end node polls N local sensors in round robin, then the sensor reading packets can arrive at this node one after another without temporal overlap. Such packet arrival pattern is Conflict-Free.

B. The Coupling Problem under Data Throughput Proportional Weight Assignment (DTPWA)

With the above E2E delay bounds, GR-aggregates effectively meet the needs of Internet, the GR-aggregates' original application context [6][7]. However, when applying GR-aggregates to RTE-WAN, a new challenge emerges:

The GR-aggregate E2E delay bound is inverse-proportionally coupled with the aggregate F and flow f's guaranteed rates R_F and r_f (see Inequality (3) $\sim$ (5)). Though the original GR-aggregate design [6][7] does not specify how to set R_F and r_f, as a common-practice, people assign WFQ scheduling weights ϕ_F and ϕ_f proportional to data throughput to avoid queue overflow. For simplicity, we refer to such common-practice as *Data Throughput Proportional Weight Assignment* (DTPWA).

Since guaranteed rate $R_F = \phi_F C$ and $r_f = \phi_f C$, where C is the server output capacity, DTPWA makes guaranteed rate proportional to aggregate/flow data throughput. Therefore, a GR-aggregate's E2E delay bound becomes inverse-proportionally coupled with the aggregate/flow data throughput. As a consequence, if aggregate F and flow f are of small data throughput, then guaranteed rates R_F and r_f are small, and E2E delay bound d_f^j becomes large; if F and f are of large data throughput, then R_F and r_f are large, and d_f^j becomes small.

This is undesirable for RTE-WAN traffic, which typically includes

1) *hard real-time sensing/actuating traffic, which usually has small data throughput but demands short E2E delay bounds*;

2) hard real-time video streams, which have large data throughput and demand short E2E delay bounds;

3) soft real-time traffic, such as FTP, which may have large data throughput, but only demands bounded E2E delays (does not have to be short).

Because of the inverse-proportional coupling of E2E delay bound and data throughput under DTPWA, hard real-time sensing/actuating traffic gets very large E2E delay bounds. This problem motivates us to decouple E2E delay bounds from data throughput.

III. GD-AGGREGATE

We propose *Guaranteed Delay server based aggregates* (GD-aggregates), to allow non-DTPWA weight assignments, particularly *priority* based weight assignment, so as to decouple E2E delay bounds from data throughput.

In the following, Section III-A introduces basic building components for GD-aggregates; Section III-B describes the GD-aggregate design and gives its E2E delay bounds; Section III-C elaborates on how to assign weights according GD-aggregates' priorities, so as to decouple E2E delay bounds from data throughput, and shows how to analyze GD-aggregates with simple closed form formulae, and to plan GD-aggregates with optimization tools.

A. Guaranteed Delay Server

To guarantee a bounded E2E delay, it is enough to guarantee a packet transmission time bound at each intermediate node. If this time bound is decoupled from data throughput, the E2E delay is decoupled from data throughput. This observation motivates our proposal of *Guaranteed Delay* (GD) server as a basic building component:

Definition 2 (Guaranteed Delay Server): Same as shown in Fig. 2, suppose the jth ($j = 1, 2, \ldots$) packet p_f^j of flow f arrives at queue Q_S at $A_S(p_f^j)$, and leaves server S at $L_S(p_f^j)$. Suppose the length of p_f^j is ℓ_f^j. We define *Guaranteed Delay Clock* (GDC) of packet p_f^j at server S as

$$\mathrm{GDC}(p_f^j) \overset{def}{=} \max\{A_S(p_f^j), \mathrm{GDC}(p_f^{j-1})\} + \Delta(\ell_f^j),$$
$$\forall j = 1, 2, \ldots,$$

where the *nonnegative monotonically nondecreasing* function $\Delta(\ell)$ is called the *Guaranteed Delay Function*, and $\mathrm{GDC}(\ell_f^0) \overset{def}{=} 0$. If server S can provide a nonnegative monotonically nondecreasing guaranteed delay function $\Delta(\ell)$ for flow f, such that for each packet p_f^j ($j = 1, 2, \ldots$) of f, there is

$$L_S(p_f^j) \le \mathrm{GDC}(p_f^j) + \alpha, \tag{6}$$

where α is a constant independent of p_f^j, then S is a *Guaranteed Delay* (GD) server. In addition, we call α the *scheduling constant* of GD server S for f. ■

We prove that any *Token Bucket Constrained WFQ* (TBC-WFQ) server is a GD server, and give its guaranteed delay function. The analysis uses the concept of *greedy starting*, and the liquid flow model based *Generalized Processor Sharing* (GPS) server [13].

A server S is called *Token Bucket Constrained* (TBC) if each of its input flows is constrained by a token bucket, as shown in Fig. 4. A flow f is called *greedy starting* from time τ, if $\forall t \geq \tau$, the number of bits pass through f's token bucket TB_f during $[\tau, t]$ equals $\sigma(\tau) + (t - \tau)\rho$, where $\sigma(\tau)$ is the number of tokens in TB_f at τ, and ρ is TB_f's token filling rate, which is also f's data throughput.

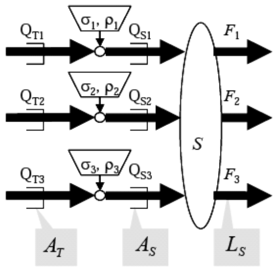


Fig. 4. A Token-Bucket-Constrained GPS (TBC-GPS) or Token-Bucket-Constrained WFQ (TBC-WFQ) Server (depends on whether S is GPS or WFQ)

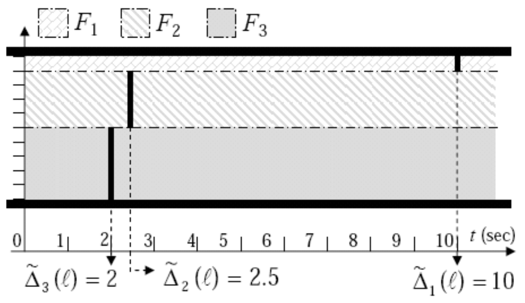
Theorem 3 (Critical Instance for Transmission Time): Suppose under TBC-GPS a chunk p of ℓ continuous bits of flow f starts transmission (i.e. p reaches the head of the queue) at τ, and the transmission completes at $\tau + \tilde{\Delta}$, Then $\tilde{\Delta} \leq \hat{\Delta}(\ell)$, where $\hat{\Delta}(\ell)$ is the transmission time cost if p is the first ℓ bits of flow f to send at time 0, with all flows of the system greedy starting from time 0. Note, without loss of generality, this paper always assume the whole system is initiated at time 0, and all token buckets are full when initiated.

Proof: The proof is based on Lemma 10 of [13]. See Appendix B of [8] for details. ■

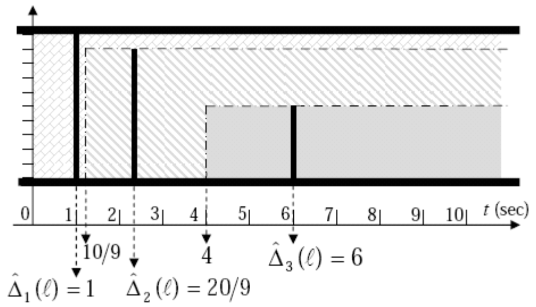
According to the above theorem, function $\hat{\Delta}(\ell)$ gives the transmission time bound for a packet of f with length ℓ. Hence, we call $\hat{\Delta}(\ell)$ the *Packet Transmission Time Bound Function*. $\hat{\Delta}(\ell)$ has the following property:

Property 1: $\hat{\Delta}(\ell)$ is nonnegative monotonically nondecreasing. Particularly, $\forall 0 \leq \ell_1 \leq \ell_2$, $0 \leq \hat{\Delta}(\ell_1) \leq \hat{\Delta}(\ell_2)$, and $\forall 0 \leq \ell \leq \ell^{max}$, $\hat{\Delta}(\ell) \leq \hat{\Delta}(\ell^{max})$, where ℓ^{max} is the maximum packet size possible. ■

Theorem 3 not only specifies how to calculate the packet transmission time bound function $\hat{\Delta}(\ell)$, but also implies this packet transmission time bound can be decoupled from flow's data throughput. The key to the decoupling is to assign proper scheduling weights. This is shown in the following example:



(a) Using common-practice *Data Throughput Proportional Weight Assignment* (DTPWA), packet transmission time bound is coupled (inverse proportional) with data throughput ρ.



(b) Using unconventional weight assignment (in fact, PAWA described in Section III-C), packet transmission time bound still exists due to Theorem 3, and is decoupled from data throughput ρ.

Fig. 5. Theorem 3's implications on decoupling packet transmission time bound from flow data throughput. Thick vertical solid line segments indicate the time when the first packets are transmitted.

Example 1: Consider a TBC-GPS server S with three input flows $F_1 \sim F_3$ as shown in Fig. 4. Suppose S's output capacity is 1, and each flow's data throughput (equivalent to the token bucket's token filling rate) is $\rho_1 = 0.1$, $\rho_2 = 0.4$, and $\rho_3 = 0.5$ respectively. For simplicity, suppose all flows' packet sizes are $\ell = 1$, and all flows' token bucket capacity equals ℓ.

According to DTPWA, $F_1 \sim F_3$ are assigned weight $\phi_1 = 0.1$, $\phi_2 = 0.4$, and $\phi_3 = 0.5$ respectively, resulting in guaranteed rates $R_1 = \rho_1 = 0.1$, $R_2 = \rho_2 = 0.4$, and $R_3 = \rho_3 = 0.5$. With such guaranteed rates, a packet of $F_1 \sim F_3$ has a transmission time cost of $\tilde{\Delta}_1(\ell) = 10$, $\tilde{\Delta}_2(\ell) = 2.5$, and $\tilde{\Delta}_3(\ell) = 2$ respectively, as shown in Fig. 5(a). The per packet transmission time is inverse-proportional to flow's data throughput.

In contrast, suppose we assign weight $\phi_1 = 0.999$, $\phi_2 = 0.000999$, and $\phi_3 = 0.000001$. Then the greedy starting scenario is shown in Fig. 5(b), with $\hat{\Delta}_1(\ell) = 1$, $\hat{\Delta}_2(\ell) = 20/9$, and $\hat{\Delta}_3(\ell) = 6$, which are decoupled from data throughput. According to Theorem 3, $\hat{\Delta}_1(\ell) \sim \hat{\Delta}_3(\ell)$ bounds the packet transmission time for flow $F_1 \sim F_3$ respectively. Therefore, the packet transmission time is decoupled from data throughput. ■

Based on Theorem 3, we prove TBC-WFQ servers are GD servers:

Theorem 4 (TBC-WFQ Servers Are GD Servers): If S is a TBC-GPS or TBC-WFQ server, define

$$\mathrm{GDC}(p_f^j) \overset{def}{=} \max\{A_S(p_f^j), \mathrm{GDC}(p_f^{j-1})\} + \hat{\Delta}(\ell_f^j), \tag{7}$$

where $\mathrm{GDC}(p_f^0) = 0$ and $\hat{\Delta}(\ell)$ is the packet transmission time bound function derived from Theorem 3. Then

$$\begin{aligned} L_S^{GPS}(p_f^j) &\leq \mathrm{GDC}(p_f^j), & (8)\\ \text{and} \quad L_S^{WFQ}(p_f^j) &\leq L_S^{GPS}(p_f^j) + \frac{\ell^{max}}{C} & \\ &\leq \mathrm{GDC}(p_f^j) + \frac{\ell^{max}}{C}, & (9)\end{aligned}$$

where $L_S^{GPS}(p)$ and $L_S^{WFQ}(p)$ are the time when packet p leaves S when S is a GPS and WFQ respectively. That is, a TBC-WFQ server S is a GD server; its guaranteed delay function is its packet transmission time bound function $\hat{\Delta}(\ell)$; and its scheduling constant $\alpha = \frac{\ell^{max}}{C}$, where ℓ^{max} is the maximum size of packets entering S, and C is S's output capacity.

Proof: See Appendix C of [8]. ■

B. The GD-aggregate Design and E2E Delay Bound

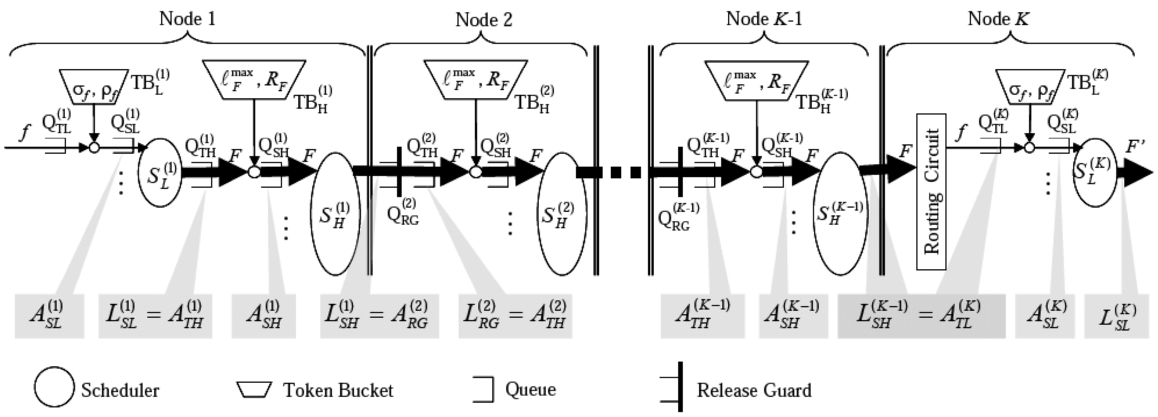


Fig. 6. A Release-Guarded Token-Bucket-Constrained WFQ (TBC-WFQ) based aggregate. Labels in the figure explain the symbols denoting arrival and departure (leaving) time of packets. Routing circuits in Node $1 \sim (K-1)$ are omitted in the figure.

Using TBC-WFQ servers, we build *GD server based aggregates* (denoted as *GD-aggregates*) as shown in Fig. 6. In Fig. 6, a GD-aggregate is created by aggregating several flows with *one GR* (not necessarily GD) server at the sender-end node, forwarded by *several GD* servers along intermediate nodes, and de-aggregated at receiver-end node by forwarding each packet according to its original flow header. We call the GR server that creates a GD-aggregate the "*low-end server*"; while the GD servers that forward the GD-aggregate the "*high-end servers*".

Specifically, a flow f joins other flows at *GR* server $S_L^{(1)}$ at Node 1 to output *GD-aggregate* F. $S_L^{(1)}$ guarantees f a rate of r_f, which is no less than f's data throughput. Particularly, if f complies with token bucket (σ_f, ρ_f) (denoted as $f \sim (\sigma_f, \rho_f)$) before arriving at $Q_{SL}^{(1)}$, then $r_f \geq \rho_f$. *We call* $R_F \overset{def}{=} \sum_{f \in F} r_f$ *and* $\rho_F \overset{def}{=} \sum_{f \in F} \rho_f$

the capacity and the data throughput of GD-aggregate F respectively (naturally $R_F \geq \rho_{_F}$ since $r_f \geq \rho_{_f}$). Note we use R_F as GD-aggregate capacity to maintain symbolic consistancy with [6][7]. We set the output capacity $C_L^{(1)}$ of $S_L^{(1)}$ to R_F. We also exploit the fact that $S_L^{(1)}$ is implemented in software to play a trick called *Time-Of-Scheduling-Equals-Time-Of-Leaving* (TOSETOL) [7]: once $S_L^{(1)}$ schedules a packet p, p is immediately output to $Q_{TH}^{(1)}$; meanwhile, $S_L^{(1)}$ waits another $\ell/C_L^{(1)} = \ell/R_F$ seconds before scheduling next packet, where ℓ is the length of p.

Once a packet p leaves $S_L^{(1)}$, it is regarded as a packet of GD-aggregate F, and is queued at $Q_{TH}^{(1)}$ to enter TBC-WFQ server $S_H^{(1)}$. The corresponding token bucket $TB_H^{(1)}$ has a bucket capacity of ℓ_F^{max} and a token filling rate of R_F, where $\ell_F^{max} \overset{def}{=} \max_{f\in F}\{\ell_f^{max}\}$, and ℓ_f^{max} is the maximum packet size of flow f. For simplicity, we denote $TB_H^{(1)} = (\ell_F^{max}, R_F)$.

Later, TBC-WFQ server $S_H^{(1)}$ will forward p via the physical output link to Node 2. At Node 2, p is first queued at *release guard* $Q_{RG}^{(2)}$ [14][15]. A release guard is a special kind of token bucket that *only* allows backlogged packets consuming tokens when the bucket is full, one packet at a time. For $Q_{RG}^{(2)}$, its bucket capacity and token filling rate are also ℓ_F^{max} and R_F respectively. For simplicity, we denote $Q_{RG}^{(2)} = (\ell_F^{max}, R_F)_{_{RG}}$. Once p leaves release guard $Q_{RG}^{(2)}$, it is queued at token bucket $TB_H^{(2)} = (\ell_F^{max}, R_F)$ to enter TBC-WFQ server $S_H^{(2)}$. Later, $S_H^{(2)}$ will forward p to Node 3, which also has $Q_{RG}^{(3)} = (\ell_F^{max}, R_F)_{_{RG}}$, $TB_H^{(3)} = (\ell_F^{max}, R_F)$, and so on and so forth, until p reaches Node K. At Node K, GD-aggregate F is de-aggregated, i.e., p is routed according to its original flow header to its corresponding output interface, where it may again join another set of flows at GR server $S_L^{(K)}$ to create another aggregate F'.

The above GD-aggregate architecture in Fig. 6 can be further simplified due to the following lemma:

Lemma 1 (All TB_Hs Can Be Ignored): The queueing delay at $Q_{TH}^{(i)}$ $(i = 1, 2, \ldots, (K-1))$ are all 0.

Proof: See proof of Lemma 3 in Appendix D of [8]. ■

This means all $Q_{TH}^{(i)}$ and $TB_H^{(i)}$ $(i = 1, 2, \ldots, (K-1))$ in Fig. 6 can be removed.

Based on Lemma 1, we derive the following theorem for GD-aggregate E2E delay bound:

Theorem 5 (E2E Delay′ with Prerequisites): Suppose, as shown in Fig. 6, flow f joins GD-aggregate F from GR server $S_L^{(1)}$ at Node 1, traverses release guarded GD server $S_H^{(1)} \sim S_H^{(K-1)}$, and finally reaches Node K to be de-aggregated. Suppose the guaranteed delay function for aggregate F at $S_H^{(i)}$ is $\Delta_F^{(i)}(\ell)$ $(i = 1, 2, \ldots, (K-1))$; and for any valid packet length $\ell \in [\ell_F^{min}, \ell_F^{max}]$, $\Delta_F^{(i)}(\ell) \leq \frac{\ell}{R_F}$, where R_F is F's capacity. Then for jth $(j = 1, 2, \ldots)$ packet of f, p_f^j, the E2E delay ${d_f^j}'$ (we use ${d_f^j}'$ instead of d_f^j to make symbols consistent with [6][7]) satisfies:

$$\begin{aligned} {d_f^j}' &\overset{def}{=} L_{SH}^{(K-1)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \qquad (10) \\ &\leq [\mathrm{GRC}_{SL}^{(1)}(p_f^j) - A_{SL}^{(1)}(p_f^j)] + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_F^{max}) \\ &\quad + \alpha_{_L}^{(1)} + \sum_{i=1}^{K-1} \alpha_{_H}^{(i)}, \qquad (11) \end{aligned}$$

where $A_{SL}^{(1)}(p)$ is when packet p arrives at $Q_{SL}^{(1)}$, $L_{SH}^{(K-1)}(p)$ is when packet p leaves $S_H^{(K-1)}$, $\alpha_{_L}^{(1)}$ is the scheduling constant of GR server $S_L^{(1)}$ for flow f, and $\alpha_{_H}^{(i)}$ is the scheduling constant of GD server $S_H^{(i)}$ for GD-aggregate F $(i = 1, 2, \ldots, (K-1))$. In addition, if packets arrive at $S_L^{(1)}$ in Conflict-Free pattern (defined in Theorem 2), then

$${d_f^j}' \leq \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_F^{max}) + \sum_{i=1}^{K-1} \alpha_{_H}^{(i)}. \qquad (12)$$

Proof: See Appendix D in [8]. ■

Corollary 1: If flow f conforms to token bucket $TB_L^{(1)} = (\sigma_{_f}, \rho_{_f})$ as shown in Fig. 6, and $S_L^{(1)}$ guarantees rate $r_f \geq \rho_{_f}$, then Inequality (11) becomes:

$${d_f^j}' \leq \frac{\sigma_{_f}}{r_f} + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_F^{max}) + \alpha_{_L}^{(1)} + \sum_{i=1}^{K-1} \alpha_{_H}^{(i)}. \qquad (13)$$

Proof: Similar to the derivation of Inequality (43) of [10], we have $\text{GRC}_{SL}^{(1)}(p_f^j) \le \frac{\sigma_f}{r_f} + A_{SL}^{(1)}(p_f^j)$. ■

Corollary 2: If, as shown in Fig. 6, flow f joins another GD-aggregate F' at Node K at GR server $S_L^{(K)}$, token bucket $TB_L^{(1)} = TB_L^{(K)} = (\sigma_f, \rho_f)$, and all other conditions are the same as those of Corollary 1, then E2E delay

$$d_f^j \overset{def}{=} A_{SL}^{(K)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \tag{14}$$

$$\le \quad \frac{\sigma_f}{r_f} + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_F^{max}) + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{(i)}. \tag{15}$$

Proof: See Appendix E of [8]. ■

We have two observations on the E2E delay bounds in the above theorems and corollaries:

1) GD-aggregate E2E delay bound is decoupled (or partially decoupled) from data throughput: Due to Theorem 3, and as shown in Example 1, by assigning proper scheduling weight, $\Delta_F^{(i)}(\ell_F^{max})$ in Inequality (11) $\sim$ (15) can be decoupled from data throughput ρ_F. Therefore GD-aggregate E2E delay bound is decoupled from data throughput. Note if the GD-aggregate transports RTE-WAN hard real-time sensing/actuating traffic, packets can easily arrive in Conflict-Free pattern, therefore E2E delay bound is calculated with Inequality (12), which is completely decoupled from data throughput. If the GD-aggregate transports RTE-WAN hard real-time video traffic or soft real-time traffic, the E2E delay bound is partially decoupled from data throughput. But the improvement is still significant, with all $(K-2)\frac{\ell_F^{max}}{R_F}$ terms removed. What is more, the coupling problem is not a prominent defect for RTE-WAN hard real-time video traffic or soft real-time traffic anyway.

2) GD-aggregate is a generalization of GR-aggregate (with a bounded error): In Fig. 6, if we stick to DTPWA, that is, assigning WFQ weight proportional to input flow/aggregate's data throughput (ρ_f and ρ_F), then $\Delta_F^{(i)}(\ell_F^{max}) = \hat{\Delta}_F^{(i)}(\ell_F^{max}) \le \ell_F^{max}/R_F$, and Inequality (11) implies Inequality (3), with only a maximal possible error of ℓ_F^{max}/R_F.

However, there are still three unsettled problems in order to use Inequality (11) $\sim$ (15):

1) How to assign proper scheduling weight so that the guaranteed delay function $\Delta_F^{(i)}(\ell)$ is decoupled from data throughput?

2) Given the proper scheduling weight, how to calculate the guaranteed delay function $\Delta_F^{(i)}(\ell)$?

3) A GD-aggregate must satisfy precondition $\forall \ell \in [\ell_F^{min}, \ell_F^{max}]$, $\Delta_F^{(i)}(\ell) \le \ell/R_F$ ($i = 1 \sim (K-1)$) to use Inequality (11) $\sim$ (15). How to remove this precondition?

The next sub-section addresses these problems.

C. Priority Approximating Weight Assignment (PAWA) Scheme

To address the three problems proposed in the end of last sub-section, we propose the *Priority Approximating Weight Assignment* (PAWA) scheme, with following features:

1) Introduces priorities into GD-aggregates. A GD-aggregate F's priority decides F's scheduling weight, and hence decides the guaranteed delay function $\Delta_F^{(i)}(\ell)$. Particularly, $\Delta_F^{(i)}(\ell)$ is decoupled from data throughput, and a higher priority corresponds to shorter $\Delta_F^{(i)}(\ell_F^{max})$.

2) Guaranteed delay function $\Delta_F^{(i)}(\ell)$ can be calculated with a closed-form linear formula.

3) The precondition of Inequality (11) $\sim$ (15) can be removed (at the cost of a larger E2E delay bound).

In addition, PAWA scheme allows planning with classic optimization tools, and enables simple admission tests.

The details are as follows:

The Scheme

Under PAWA scheme, a high-end[1] TBC-WFQ server S supports a set of priorities: $1, 2, \ldots, \Pi$, where smaller number means higher priority. Each priority π ($\pi < \Pi$) corresponds to three parameters: packet transmission time bound $\Delta_\pi^\star$, total aggregates' capacity $R_\pi^\star$, and total maximum packet size $\ell_\pi^\star$. During configuration time, system

[1]As mentioned in Section III-B, we call the GR server that creates a GD-aggregate the "*low-end server*"; while the GD servers that forward the GD-aggregate the "*high-end servers*".

administrator can set these three parameters to any real numbers as long as they satisfy the following constraints:

$$\Delta_0^\star \overset{def}{=} 0, \quad 0 < \Delta_1^\star < \Delta_2^\star < \ldots < \Delta_{\Pi-1}^\star; \tag{16}$$

$$R_\pi^\star \geq R_\pi^{\star min} > 0, \qquad \pi = 1 \sim (\Pi - 1); \tag{17}$$

$$\ell_\pi^\star \geq \ell_\pi^{\star min} > 0, \qquad \pi = 1 \sim (\Pi - 1); \tag{18}$$

$$\textstyle\sum_{\pi=1}^{\Pi-1} R_\pi^\star < C, R_\Pi^\star \overset{def}{=} C - \sum_{\pi=1}^{\Pi-1} R_\pi^\star; \tag{19}$$

$$C_0^\star \overset{def}{=} 0, \quad C_1^\star \overset{def}{=} C; \tag{20}$$

$$C_\pi^\star \overset{def}{=} C_{\pi-1}^\star - R_{\pi-1}^\star = C - \textstyle\sum_{i=1}^{\pi-1} R_i^\star, \quad \pi = 2 \sim \Pi; \tag{21}$$

$$\ell_1^\star = \Delta_1^\star C_1^\star; \tag{22}$$

$$\ell_\pi^\star = \Delta_\pi^\star C_\pi^\star - \Delta_{\pi-1}^\star C_{\pi-1}^\star, \pi = 2 \sim (\Pi - 1); \tag{23}$$

where C is the output capacity of S; $R_\pi^{\star min}$ and $\ell_\pi^{\star min}$ are minimum limits for $R_\pi^\star$ and $\ell_\pi^\star$ set by administrator.

Each GD-aggregate entering S must pick one priority. Denote $\mathcal{F}_\pi$ as the set of GD-aggregates entering S with priority π. To simplify our analysis, the system will add a dummy GD-aggregate $\bar{F}_\pi$ to each $\mathcal{F}_\pi$, where $\bar{F}_\pi$ is constrained by a token bucket of $(\ell_{\bar{F}_\pi}^{max}, R_{\bar{F}_\pi})$. For any $\pi < \Pi$, $\bar{F}_\pi$'s token bucket capacity $\ell_{\bar{F}_\pi}^{max} = \ell_\pi^\star - \sum_{F\in\mathcal{F}_\pi, F\neq\bar{F}_\pi} \ell_F^{max}$; for $\pi = \Pi$, $\ell_{\bar{F}_\pi}^{max}$ is set to an arbitrary constant that can be used as packet length. The token filling rate $R_{\bar{F}_\pi} = R_\pi^\star - \sum_{F\in\mathcal{F}_\pi, F\neq\bar{F}_\pi} R_F$. To insure the feasibility of setting $\ell_{\bar{F}_\pi}^{max}$ and $R_{\bar{F}_\pi}$, we require

$$\textstyle\sum_{F\in\mathcal{F}_\pi, F\neq\bar{F}_\pi} \ell_F^{max} \leq \ell_\pi^\star \quad (\forall \pi < \Pi); \tag{24}$$

$$\text{and } \textstyle\sum_{F\in\mathcal{F}_\pi, F\neq\bar{F}_\pi} R_F \leq R_\pi^\star \quad (\forall \pi = 1 \sim \Pi). \tag{25}$$

A new aggregate F is not admitted to $\mathcal{F}_\pi$ if its admission will violate Formulae (24) or (25).

The weight assignment rules run as follows:

Each $\mathcal{F}_\pi$ is assigned a total weight of ψ_π, such that

$$\psi_\pi / \psi_{\pi+1} = \Psi >> 1, \qquad \pi = 1 \sim \Pi - 1 \quad \text{and} \quad \textstyle\sum_{\pi=1}^{\Pi} \psi_\pi = 1, \tag{26}$$

where Ψ is a sufficiently large constant. For each GD-aggregate $F \in \mathcal{F}_\pi$ (including $\bar{F}_\pi$), its weight ϕ_F is

$$\phi_F = \begin{cases} \psi_\pi \ell_F^{max} / \ell_\pi^\star, & \text{when } \pi < \Pi; \\ \psi_\pi R_F / R_\pi^\star, & \text{when } \pi = \Pi. \end{cases} \tag{27}$$

Based on above rules, PAWA provides many desirable properties. First, it results in closed-form linear guaranteed delay functions:

Theorem 6 (PAWA Guaranteed Delay Function): If TBC-WFQ server S complies with PAWA scheme, then $\forall F \in \mathcal{F}_\pi$ and $\forall \ell_F^{min} \leq \ell \leq \ell_F^{max}$, the PAWA GD server S provides F a guaranteed delay function

$$\Delta_F^{(S)}(\ell) = \hat{\Delta}_F^{(S)}(\ell) = \begin{cases} \Delta_{\pi-1}^\star \frac{C_{\pi-1}^\star}{C_\pi^\star} + \frac{\ell}{\ell_F^{max}}(\Delta_\pi^\star - \Delta_{\pi-1}^\star \frac{C_{\pi-1}^\star}{C_\pi^\star}), \\ \qquad \text{when } \pi < \Pi; \\ \Delta_{\Pi-1}^\star \frac{C_{\Pi-1}^\star}{C_\Pi^\star} + \frac{\ell}{R_F}, \quad \text{when } \pi = \Pi; \end{cases} \tag{28}$$

where $\hat{\Delta}_F^{(S)}(\ell)$ is the packet transmission time bound function mentioned in Theorem 3. Particularly, Equation (28) implies when $\pi < \Pi$, $\forall F \in \mathcal{F}_\pi$,

$$\Delta_F^{(S)}(\ell_F^{max}) = \hat{\Delta}_F^{(S)}(\ell_F^{max}) = \Delta_\pi^\star, \tag{29}$$

which is why we call $\Delta_\pi^\star$ the "packet transmission time bound" parameter[2].

[2]According to Theorem 3, every packet's transmission time under GPS is no more than $\hat{\Delta}_F^{(S)}(\ell_F^{max})$.

Proof: See Appendix F of [8]. ■

Second, PAWA guarantees E2E delay without the $\Delta_F^{(i)}(\ell) \le \frac{\ell}{R_F}$ prerequisite in Theorem 5. This is described in the following by Theorem 7 and 8:

Theorem 7 (PAWA TBC-WFQ Server is also GR): Without loss of generality, suppose in Node i (e.g. $i = 1$) of Fig. 6, $Q_{TH}^{(i)}$, $TB_H^{(i)}$, $Q_{SH}^{(i)}$, and $S_H^{(i)}$ make up a PAWA TBC-WFQ server. Then for each $F \in \mathcal{F}_\pi$ ($\pi = 1 \sim \Pi$), $S_H^{(i)}$ is also a GR server with guaranteed rate $R_{_F}$ and scheduling constant

$$\alpha_{_H}^{'(i)} = \begin{cases} \Delta_\pi^\star C_\pi^\star / C_{\pi+1}^\star + \frac{\ell_{SH}^{(i)max}}{C}, & \text{if } \pi < \Pi, \\ \Delta_{\Pi-1}^\star C_{\Pi-1}^\star / C_\Pi^\star + \frac{\ell_{SH}^{(i)max}}{C}, & \text{if } \pi = \Pi, \end{cases} \tag{30}$$

where $\ell_{SH}^{(i)max}$ is the maximum packet length of all aggregates/flows entering $S_H^{(i)}$. That is, if define $\mathrm{GRC}_{SH}^{(i)}(p_F^j) \overset{def}{=} \max\{A_{SH}^{(i)}(p_F^j), \mathrm{GRC}_{SH}^{(i)}(p_F^{j-1})\} + \ell_F^j / R_{_F}$, then $L_{SH}^{(i)}(p_F^j) \le \mathrm{GRC}_{SH}^{(i)}(p_F^j) + \alpha_{_H}^{'(i)}$, where $L_{SH}^{(i)}(p)$ is the time when p leaves WFQ server $S_H^{(i)}$. Note $\mathrm{GRC}_{SH}^{(i)}(p_F^0) \overset{def}{=} 0$.

Proof: See Appendix G of [8]. ■

Theorem 8 (E2E Delay′ without Prerequisites): Suppose flow f joins GD-aggregate F at $S_L^{(1)}$ and traverses $S_H^{(1)}$, $S_H^{(2)}$, …, $S_H^{(K-1)}$, and $S_L^{(K)}$ as shown in Fig. 6. Suppose each TBC-WFQ server $S_H^{(i)}$ ($i = 1, 2, \dots, K-1$) enforces PAWA scheme. According to Theorem 7, $S_H^{(i)}$ is also a GR server for F with a GR scheduling constant $\alpha_{_H}^{'(i)}$ (see Formula (30)). Then for packet p_f^j, the E2E delay $d_f^{j\,'}$ satisfies:

$$\begin{aligned} d_f^{j\,'} &\overset{def}{=} L_{SH}^{(K-1)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \\ &\le [\mathrm{GRC}_{SL}^{(1)}(p_f^j) - A_{SL}^{(1)}(p_f^j)] + (K-1)\frac{\ell_F^{max}}{R_{_F}} \\ &\quad + \alpha_{_L}^{(1)} + \sum_{i=1}^{K-1} \alpha_{_H}^{'(i)}, \end{aligned} \tag{31}$$

where $A_{SL}^{(1)}(p)$ is when packet p arrives at $Q_{SL}^{(1)}$; $L_{SH}^{(K-1)}(p)$ is when packet p leaves $S_H^{(K-1)}$; $\alpha_{_L}^{(1)}$ is the GR scheduling constant at server $S_L^{(1)}$, and $\alpha_{_H}^{'(i)}$ is the GR scheduling constant at server $S_H^{(i)}$. In addition, if packets arrive at $S_L^{(1)}$ in Conflict-Free pattern (defined in Theorem 2), then

$$d_f^{j\,'} \le (K-1)\frac{\ell_F^{max}}{R_{_F}} + \sum_{i=1}^{K-1} \alpha_{_H}^{'(i)}. \tag{32}$$

Proof: See Appendix H of [8]. ■

Corollary 3: If flow f conforms to token bucket $TB_L^{(1)} = (\sigma_{_f}, \rho_{_f})$ as shown in Fig. 6, and $r_f \ge \rho_{_f}$, then Inequality (31) becomes:

$$d_f^{j\,'} \le \frac{\sigma_{_f}}{r_f} + (K-1)\frac{\ell_F^{max}}{R_{_F}} + \alpha_{_L}^{(1)} + \sum_{i=1}^{K-1} \alpha_{_H}^{'(i)}. \tag{33}$$

Proof: Similar to the derivation of Inequality (43) in Goyal et al. [10], we have $\mathrm{GRC}_{SL}^{(1)}(p_f^j) \le \frac{\sigma_f}{r_f} + A_{SL}^{(1)}(p_f^j)$. ■

Corollary 4: If, as shown in Fig. 6, flow f joins another GD-aggregate F' at Node K at GR server $S_L^{(K)}$, token bucket $TB_L^{(1)} = TB_L^{(K)} = (\sigma_{_f}, \rho_{_f})$, $r_f \ge \rho_{_f}$, and all other conditions are the same as those of Corollary 3, then E2E delay

$$\begin{aligned} d_f^j &\overset{def}{=} A_{SL}^{(K)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \\ &\le \frac{\sigma_{_f}}{r_f} + (K-1)\frac{\ell_F^{max}}{R_{_F}} + \alpha_{_L}^{(1)} + \sum_{i=1}^{K-1} \alpha_{_H}^{'(i)}. \end{aligned} \tag{34}$$

Proof: See Appendix I of [8]. ■

PAWA Parameter Planning

During configuration time, a PAWA TBC-WFQ server administrator can plan the $\{\Delta_\pi^\star\}$, $\{R_\pi^\star\}$, and $\{\ell_\pi^\star\}$ parameters with classic optimization tools. Just to give an example:

Given C, $\{\Delta_\pi^\star\}$, desired total aggregates' capacity $\{\tilde{R}_\pi^\star\}$, desired total max packet size $\{\tilde{\ell}_\pi^\star\}$, weight (importance) w_π of getting $R_\pi^\star$ close to $\tilde{R}_\pi^\star$, and weight ϖ_π of getting $\ell_\pi^\star$ close to $\tilde{\ell}_\pi^\star$, derive optimal settings of $\{R_\pi^\star\}$ and $\{\ell_\pi^\star\}$.

The problem corresponds to the following convex optimization problem:

$$\min \textstyle\sum_{\pi=1}^{\Pi-1} \left(w_\pi (R_\pi^\star - \tilde{R}_\pi^\star)^2 + \varpi_\pi (\ell_\pi^\star - \tilde{\ell}_\pi^\star)^2 \right),$$

with convex linear constraint set (16) $\sim$ (23).

GD-aggregate Admission Test

To add a GD-aggregate F with priority π at PAWA TBC-WFQ server S, F only needs to pass the following three tests:

Test 1:

$$\ell_F^{max} + \sum_{f \in \mathcal{F}_\pi, f \neq \bar{F}_\pi} \ell_f^{max} \le \ell_\pi^\star, \qquad \text{if } \pi < \Pi; \tag{35}$$

Test 2:

$$R_F + \sum_{f \in \mathcal{F}_\pi, f \neq \bar{F}_\pi} R_f \le R_\pi^\star; \tag{36}$$

Test 3 (Theorem 5 Prerequisite): If $\forall \ell \in [\ell_F^{min}, \ell_F^{max}]$, $\Delta_F^{(S)}(\ell) \le \ell / R_F$ ($\Delta_F^{(S)}(\ell)$ is derived from Equation (28)), then use Theorem 5, Corollary 1, or Corollary 2 to calculate E2E delay. Otherwise, use Theorem 8, Corollary 3, or Corollary 4 to calculate E2E delay.

Usually, we should assign RTE-WAN hard real-time sensing/actuating GD-aggregates with the highest priority. Because such GD-aggregates' maximum packet lengthes are small, such priority assignment will empirically always satisfy the Theorem 5 prerequisite. RTE-WAN hard real-time video GD-aggregates shall take lower priorities, which may or may not satisfy the Theorem 5 prerequisite. But the E2E delay bounds will still be satisfactory, because RTE-WAN hard real-time video traffic has large data throughput. RTE-WAN soft real-time GD-aggregates shall take lowest priorities, and will still get bounded E2E delay. All of these are illustrated by the underground mining case study described in Appendix J of [8].

IV. Related Work

There are other candidate technologies for WAN virtual topologies (virtual links): Overlay network [16] also discusses virtual links. However, they are not hard real-time virtual links. DiffServ [17][6][7][18] is similar to aggregates: flows with similar QoS requirements are transmitted as one group. However, DiffServ uses FIFO scheduling, which is hard to guarantee hard real-time E2E delay when traffic is bursty. As pointed out by Wang et al. [18], even when token bucket ratio $\frac{\sigma}{\rho}$ is as low as 1.28, the maximal schedulable link utilization drops below 5%. Real-time virtual machines [19][20][21][22][23][24][25] can be a good candidate to support hard real-time virtual links. However, to our best knowledge, mutual exclusion, which is essential for packet scheduling, is still an open problem for hierarchical real-time virtual machines: efficient system architecture and simple closed-form schedulability formulae are yet to be developed, especially for hierarchies with more than two levels. In comparison, the GR-aggregate [6][7] scheme better meets the needs of RTE-WAN applications: it guarantees hard real-time E2E delay, assumes packetized (mutually exclusive) traffic model, supports hierarchical aggregation of arbitrary number of levels, provides closed-form analytical formulae, and can easily achieve 100% link utilization. Therefore, it is good to start RTE-WAN virtual link design on top of GR-aggregates.

Our GD-aggregate design decouples E2E delay bound from data throughput. There are other efforts on decoupling E2E delay bound from data throughput. Particularly, Geogiadis et al. [26] also discover that the combination of per node traffic shapers (token buckets) and fair queueing weights can decouple E2E delay bound from flow data throughput. The idea is similar to ours, but Geogiadis et al. assume fluid model, and do not talk about aggregation. It requires nontrivial additional work to adapt their theory to packetized aggregates, and to derive simple closed-form planning and admission test formulae. Goyal et al. [27] generalize the GR server notion to cases where guaranteed rates may differ between packets of the same flow. However, they do not talk about aggregation, and they assume the per packet guaranteed rates are either given a priori, or referring to the smallest instantaneous rates during the packets' transmission. In comparison, our E2E delay bound analysis is decoupled from any fixed packet transmission rate, and therefore enhances feasibility, flexibility, and accuracy.

As for underground mining, Mangharam et al. also discuss it in [28]. Their work focuses on multi-hop wireless sensor networks for voice traffic. Our work focuses on wireline WAN for critical hard real-time remote control traffic and large data throughput (such as video) traffic.

V. Conclusion

The convergence of computer and physical world calls for next generation WAN infrastructures for hard real-time and embedded applications. Such networks need virtual topologies to achieve scalability, configurability, and flexibility. Virtual topologies are made of virtual links, for which, the state-of-the-art building tool is *Guaranteed Rate server based aggregates* (GR-aggregates) [6][7]. However, common-practice weight assignment scheme couples GR-aggregate *End-to-End* (E2E) delay bound with aggregate's data throughput inverse proportionally. This is undesirable for many hard real-time embedded sensing/actuating applications, whose traffic has small data throughput but requires short E2E delay. We propose *Guaranteed Delay server based aggregates* (GD-aggregates) design, which allows assigning weight according to *priority* instead of data throughput. This decouples E2E delay guarantee from data throughput, hence meets the needs of hard real-time embedded applications. In addition, GD-aggregates can be analyzed with simple closed form formulae, and can be easily planned with optimization tools. Performance evaluation in the context of underground mining, a representative RTE-WAN application with typical RTE-WAN traffic (see Appendix J of [8]), shows GD-aggregates better serve the needs of hard real-time embedded applications.

Acknowledgement

This work is supported in part by NSF CCR 03-25716, NSF CNS 06-49885 SGER, by ONR N00014-05-0739, and by a grant from Lockheed Martin and a grant from Rockwell Collins. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of sponsors.

The authors thank anonymous reviewers for their advice on improving this paper.

Appendix A
Proof of Theorem 2

If packets arrive at $S_L^{(1)}$ in Conflict-Free pattern, then in Appendix III of [7], the first formula becomes

$$\begin{aligned} &\mathrm{GRC}_{SH}^{(1)}(p_f^j) = A_{SH}^{(1)}(p_f^j) + \frac{\ell_f^j}{R_F} \\ = \;& A_{SL}^{(1)}(p_f^j) + \frac{\ell_f^j}{R_F} \quad \text{(Due to TOSETOL trick)} \\ \leq \;& A_{SL}^{(1)}(p_f^j) + \frac{\ell_F^{max}}{R_F}. \end{aligned}$$

The rest follows the same proof in Appendix III of [7]. ■

Appendix B
Proof of Theorem 3

We first restate Lemma 10 of Parekh et al. [13]:

Lemma 2 (Lemma 10 of [13]): Suppose the GPS server output capacity is bounded, then for any arrival function[3] $\Lambda \sim (\sigma, \rho)$, if interval $[\tau, t]$ is within[4] a flow busy period of flow f, then $\hat{W}_f(0, t-\tau) \leq W_f(\tau, t)$, where $W_f(\tau, t)$ is the bits of f transmitted during $[\tau, t]$ under Λ, and $\hat{W}_f(0, t-\tau)$ is the bits of f transmitted during $[0, t-\tau]$ when every flow is greedy starting from time 0. Note the whole system is initiated at time 0.

According to Lemma 2, $\hat{W}_f(0, \tilde{\Delta}) \leq W_f(\tau, \tau + \tilde{\Delta}) = \ell$. Therefore, if p is the first ℓ bits of flow f to send at time 0, and all flows of the systems are greedy starting at time 0, the transmission time needed $\hat{\Delta}(\ell)$ must be no less than $\tilde{\Delta}$. That is $\hat{\Delta}(\ell) \geq \tilde{\Delta}$. ■

Appendix C
Proof of Theorem 4

First prove Inequality (8) by induction.

Step 1: When $j = 1$, p_f^1 is scheduled by S at $A_S(p_f^1)$. Due to Theorem 3, $L_S^{GPS}(p_f^1) \leq A_S(p_f^1) + \hat{\Delta}(\ell_f^1) \leq \max\{A_S(p_f^1), \mathrm{GDC}(p_f^0)\} + \hat{\Delta}(\ell_f^1) = \mathrm{GDC}(p_f^1)$.

Step 2: Suppse when $j = m$, there is

$$L_S^{GPS}(p_f^m) \leq \mathrm{GDC}(p_f^m). \tag{37}$$

Step 3: Then when $j = m+1$, we have:

Case 1: If $A_S(p_f^{m+1}) \geq L_S^{GPS}(p_f^m)$, then

$$\begin{aligned} &L_S^{GPS}(p_f^{m+1}) \\ \leq \;& A_S(p_f^{m+1}) + \hat{\Delta}(\ell_f^{m+1}) \qquad \text{(Due to Theorem 3)} \\ \leq \;& \max\{A_S(p_f^{m+1}), \mathrm{GDC}(p_f^m)\} + \hat{\Delta}(\ell_f^{m+1}) \\ = \;& \mathrm{GDC}(p_f^{m+1}). \end{aligned}$$

Case 2: If $A_S(p_f^{m+1}) < L_S^{GPS}(p_f^m)$ then

$$\begin{aligned} &L_S^{GPS}(p_f^{m+1}) \\ \leq \;& L_S^{GPS}(p_f^m) + \hat{\Delta}(\ell_f^{m+1}) \qquad \text{(Due to Theorem 3)} \\ \leq \;& \mathrm{GDC}(p_f^m) + \hat{\Delta}(\ell_f^{m+1}) \qquad \text{(Due to Inequality (37))} \\ \leq \;& \max\{A_S(p_f^{m+1}), \mathrm{GDC}(p_f^m)\} + \hat{\Delta}(\ell_f^{m+1}) \\ = \;& \mathrm{GDC}(p_f^{m+1}) \end{aligned}$$

[3]See [13] for the definition of "arrival function".

[4]In the orignal text of [13], τ refers to the beginning of a flow busy period (see [13] for the definition of "flow busy period") of f. But the proof actually sustains as long as $[\tau, t]$ is *within* one of f's flow busy periods.

Combining Case 1 and 2, Inequality (8) holds for $j = m+1$.

Based on Step 1 $\sim$ 3, Inequality (8) sustains.

Inequality (9) hence also sustains, due to WFQ property [13]. ■

APPENDIX D
PROOF OF THEOREM 5

Lemma 3 (Extended Version of Lemma 1): The queueing delay at $Q_{TH}^{(i)}$ ($i = 1, 2, 3, \ldots, K-1$) are all 0. In addition,

$$\begin{aligned} & A_{SH}^{(i)}(p_F^{j+1}) = A_{TH}^{(i)}(p_F^{j+1}) \\ \geq \quad & A_{TH}^{(i)}(p_F^j) + \frac{\ell_F^j}{R_F} = A_{SH}^{(i)}(p_F^j) + \frac{\ell_F^j}{R_F}, \\ & j = 1, 2, \ldots, \end{aligned} \tag{38}$$

where $A_{SH}^{(i)}(p)$ is the time when packet p arrives at high-end server $S_H^{(i)}$ at Node i (more specifically, arrives at $Q_{SH}^{(i)}$); and $A_{TH}^{(i)}$ is the time when p arrives at token-bucket $TB_H^{(i)}$ at Node i (more specifically, arrives at $Q_{TH}^{(i)}$).

Proof: For $i = 1$, it is due to the fact that $S_L^{(1)}$'s output capacity is R_F and also due to the TOSETOL trick.

For $i = 2, 3, \ldots, K-1$, it is due to the definition of release guard and due to the placement of $Q_{RG}^{(i)}$. ■

Lemma 4 (One Hop Release-Guard Delay Bound): For a GD-aggregate as shown in Fig. 6, if $\forall \ell_F^{min} \leq \ell \leq \ell_F^{max}$, $\Delta_F(\ell) \leq \frac{\ell}{R_F}$, then $\forall i = 1, 2, \ldots, K-1$, and $\forall j = 1, 2, \ldots$,

$$\begin{cases} L_{SH}^{(i)}(p_F^j) \leq A_{SH}^{(i)}(p_F^j) + \Delta_F(\ell_F^j) + \alpha_{_H}^{(i)}, \\ \mathrm{GDC}_{SH}^{(i)}(p_F^j) = A_{SH}^{(i)}(p_F^j) + \Delta_F(\ell_F^j). \end{cases} \tag{39}$$

And

$$\begin{aligned} & L_{RG}^{(i+1)}(p_F^j) \leq A_{SH}^{(i)}(p_F^j) + \Delta_F(\ell_F^{max}) + \alpha_{_H}^{(i)}, \\ & \qquad i = 1, 2, \ldots, K-2, \end{aligned} \tag{40}$$

where $\Delta_F(\ell)$ and $\alpha_{_H}^{(i)}$ are GD server $S_H^{(i)}$'s guaranteed delay function for aggregate F and GD server scheduling constant respectively.

Proof: Due to GD server definition,

$$\begin{aligned} & L_{SH}^{(i)}(p_F^j) \\ \leq \quad & \mathrm{GDC}_{SH}^{(i)}(p_F^j) + \alpha_{_H}^{(i)}. \end{aligned} \tag{41}$$

In the following, we prove Formulae (39) by induction.

Step 1: When $j = 1$, $\mathrm{GDC}_{SH}^{(i)}(p_F^0) \overset{def}{=} 0$, therefore

$$\begin{aligned} & \mathrm{GDC}_{SH}^{(i)}(p_F^1) \\ \overset{def}{=} \quad & \max\{A_{SH}^{(i)}(p_F^1), \mathrm{GDC}_{SH}^{(i)}(p_F^0)\} + \Delta_F(\ell_F^1) \\ = \quad & A_{SH}^{(i)}(p_F^1) + \Delta_F(\ell_F^1), \\ \text{and} \quad & (41) \\ \Rightarrow \quad & L_{SH}^{(i)}(p_F^1) \leq A_{SH}^{(i)}(p_F^1) + \Delta_F(\ell_F^1) + \alpha_{_H}^{(i)}. \end{aligned}$$

Step 2: Suppose when $j = m$ ($m \geq 1$),

$$\begin{cases} L_{SH}^{(i)}(p_F^m) \leq A_{SH}^{(i)}(p_F^m) + \Delta_F(\ell_F^m) + \alpha_{_H}^{(i)}, \\ \mathrm{GDC}_{SH}^{(i)}(p_F^m) = A_{SH}^{(i)}(p_F^m) + \Delta_F(\ell_F^m). \end{cases}$$

Step 3: When $j = m + 1$, we have

$$\begin{aligned}
& \mathrm{GDC}^{(i)}_{SH}(p^m_F) \\
= \;& A^{(i)}_{SH}(p^m_F) + \Delta_F(\ell^m_F) \\
\leq \;& A^{(i)}_{SH}(p^m_F) + \frac{\ell^m_F}{R_F} \\
& \quad (\forall \ell^{min}_F \leq \ell \leq \ell^{max}_F, \Delta_F(\ell) \leq \frac{\ell}{R_F}) \\
\leq \;& A^{(i)}_{SH}(p^{m+1}_F). \qquad \text{(Due to (38))}
\end{aligned}$$

$$\begin{aligned}
\therefore \;& \mathrm{GDC}^{(i)}_{SH}(p^{m+1}_F) \\
\overset{def}{=} \;& \max\{A^{(i)}_{SH}(p^{m+1}_F), \mathrm{GDC}^{(i)}_{SH}(p^m_F)\} + \Delta_F(\ell^{m+1}_F) \\
= \;& A^{(i)}_{SH}(p^{m+1}_F) + \Delta_F(\ell^{m+1}_F).
\end{aligned}$$

$$\begin{aligned}
\therefore \;& L^{(i)}_{SH}(p^{m+1}_F) \\
\leq \;& \mathrm{GDC}^{(i)}_{SH}(p^{m+1}_F) + \alpha^{(i)}_{H} \qquad (S^{(i)}_H \text{ is GD server}) \\
= \;& A^{(i)}_{SH}(p^{m+1}_F) + \Delta_F(\ell^{m+1}_F) + \alpha^{(i)}_{H}.
\end{aligned}$$

From Step 1 $\sim$ 3, Formulae (39) sustain.

Specifically, we have $\forall i = 1, 2, \ldots K-2$ and $\forall j = 1, 2, \ldots,$

$$\begin{aligned}
(39) \;\Rightarrow\;& A^{(i+1)}_{RG}(p^j_F) = L^{(i)}_{SH}(p^j_F) \\
\leq \;& A^{(i)}_{SH}(p^j_F) + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H}. \\
& \text{(Due to Property 1)}
\end{aligned} \tag{42}$$

In the following, we prove by induction that $\forall i = 1, 2, \ldots, K-2$, and $\forall j = 1, 2, \ldots$, Inequality (40) sustains.

Step 1: When $j = 1$, we have $L^{(i+1)}_{RG}(p^1_F) = A^{(i+1)}_{RG}(p^1_F) = L^{(i)}_{SH}(p^1_F) \leq A^{(i)}_{SH}(p^1_F) + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H}$.

Step 2: Suppose when $j = m$ ($m \geq 1$),

$$L^{(i+1)}_{RG}(p^m_F) \leq A^{(i)}_{SH}(p^m_F) + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H}. \tag{43}$$

Step 3: When $j = m+1$, Inequality (43) says that $RG^{(i+1)}$ releases p^m_F no later than $A^{(i)}_{SH}(p^m_F)+\Delta_F(\ell^{max}_F)+\alpha^{(i)}_{H}$, therefore at any time $t \geq A^{(i)}_{SH}(p^m_F)+\frac{\ell^m_F}{R_F}+\Delta_F(\ell^{max}_F)+\alpha^{(i)}_{H}$, $RG^{(i+1)}$ should not block p^{m+1}_F. Meanwhile, according to Inequality (42), p^{m+1}_F arrives at $RG^{(i+1)}$ by $A^{(i)}_{SH}(p^{m+1}_F) + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H}$. According to Inequality (38), $A^{(i)}_{SH}(p^{m+1}_F) + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H} \geq A^{(i)}_{SH}(p^m_F) + \frac{\ell^m_F}{R_F} + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H}$. Therefore $L^{(i+1)}_{RG}(p^{m+1}_F) \leq A^{(i)}_{SH}(p^{m+1}_F) + \Delta_F(\ell^{max}_F) + \alpha^{(i)}_{H}$.

From Step 1 $\sim$ 3, Inequality (40) sustains. ∎

Proof of Theorem 5: Due to Lemma 3 and the fact that $S^{(1)}_L$ is a GR server, we have

$$A^{(1)}_{SH}(p^j_f) \leq \mathrm{GRC}^{(1)}_{SL}(p^j_f) + \alpha^{(1)}_{L}. \tag{44}$$

According to Lemma 3 and Lemma 4,

$$\begin{aligned}
& A^{(2)}_{SH}(p^j_f) = L^{(2)}_{RG}(p^j_f) \\
\leq \;& A^{(1)}_{SH}(p^j_f) + \Delta^{(1)}_F(\ell^{max}_F) + \alpha^{(1)}_{H},
\end{aligned} \tag{45}$$

$$\vdots$$

$$\begin{aligned}
& A^{(K-1)}_{SH}(p^j_f) = L^{(K-1)}_{RG}(p^j_f) \\
\leq \;& A^{(K-2)}_{SH}(p^j_f) + \Delta^{(K-2)}_F(\ell^{max}_F) + \alpha^{(K-2)}_{H}.
\end{aligned} \tag{46}$$

Adding Inequality (44) to (46) together, we get

$$\begin{aligned} A_{SH}^{(K-1)}(p_f^j) &\leq \mathrm{GRC}_{SL}^{(1)}(p_f^j) + \alpha_L^{(1)} \\ &\quad + \sum_{i=1}^{K-2} \Delta_F^{(i)}(\ell_F^{max}) + \sum_{i=1}^{K-2} \alpha_{\scriptscriptstyle H}^{(i)}. \end{aligned} \tag{47}$$

Meanwhile, since $S_H^{(K-1)}$ is a GD server, we have

$$\begin{aligned} & L_{SH}^{(K-1)}(p_f^j) \\ \leq\ & \mathrm{GDC}_H^{(K-1)}(p_f^j) + \alpha_{\scriptscriptstyle H}^{(K-1)} \\ =\ & A_{SH}^{(K-1)}(p_f^j) + \Delta_F^{(K-1)}(\ell_f^j) + \alpha_{\scriptscriptstyle H}^{(K-1)} \\ & \quad (\text{Due to } (39)) \\ \leq\ & A_{SH}^{(K-1)}(p_f^j) + \Delta_F^{(K-1)}(\ell_F^{max}) + \alpha_{\scriptscriptstyle H}^{(K-1)}. \\ & \quad (\text{Due to Property } 1) \end{aligned} \tag{48}$$

Therefore

$$\begin{aligned} & (47)(48) \\ \Rightarrow\ & L_{SH}^{(K-1)}(p_f^j) \\ \leq\ & \mathrm{GRC}_{SL}^{(1)}(p_f^j) + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_F^{max}) \\ & + \sum_{i=1}^{K-1} \alpha_{\scriptscriptstyle H}^{(i)} \\ \Rightarrow\ & d_f^{j\prime} \overset{def}{=} L_{SH}^{(K-1)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \\ \leq\ & [\mathrm{GRC}_{SL}^{(1)}(p_f^j) - A_{SL}^{(1)}(p_f^j)] + \alpha_L^{(1)} \\ & + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_F^{max}) + \sum_{i=1}^{K-1} \alpha_{\scriptscriptstyle H}^{(i)}. \end{aligned}$$

In addition, if packets arrive at $S_L^{(1)}$ in Conflict-Free pattern, then (44) becomes $A_{SH}^{(1)}(p_f^j) = A_{SL}^{(1)}(p_f^j)$ due to the TOSETOL trick. With same approach we can prove Formula (12).

■

Appendix E
Proof of Corollary 2

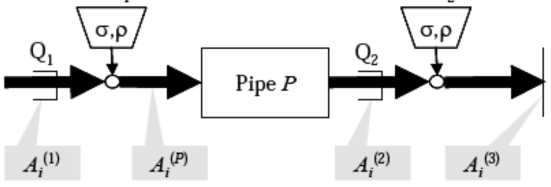


Fig. 7. Neighboring Token Buckets Connected by a Pipe

Lemma 5: Suppose at time t_0 there are x tokens in a token bucket $TB = (\sigma, \rho)$. Suppose no token is ever consumed (by traffic). Denote $\mathrm{accept}(x, t)$ as the tokens accepted by TB during interval $[t_0, t_0+t]$ (tokens overflown are *not* accepted). Then $\forall 0 \le x_1 \le x_2 \le \sigma$, and $\forall t \ge 0$,

$$x_1 + \mathrm{accept}(x_1, t) \le x_2 + \mathrm{accept}(x_2, t), \tag{49}$$

$$\mathrm{accept}(x_1, t) \ge \mathrm{accept}(x_2, t). \tag{50}$$

Proof: Trivial according to the definition of token bucket. ■

Lemma 6 (Neighboring Token Buckets Delay): Suppose flow f passes through two token buckets $TB_1 = (\sigma, \rho)$ and $TB_2 = (\sigma, \rho)$ connected by a pipe P, as shown in Fig. 7. P forwards f's packets in a first-in-first-out order, and with a forwarding delay of $d_i \overset{def}{=} A_i^{(2)} - A_i^{(P)}$ ($d_i \ge 0$) for the ith ($i = 1, 2, \ldots$) packet p_i, where $A_i^{(2)}$ and $A_i^{(P)}$ are the time when p_i arrives at Q_2 and P respectively. d_is may vary for different packets. Denote $D \overset{def}{=} \max\{d_i\}$; also denote $A_i^{(1)}$ and $A_i^{(3)}$ as the time when packet p_i arrives at Q_1 and Destination; and denote $\delta_i^{(P)} \overset{def}{=} A_{i+1}^{(P)} - A_i^{(P)}$. Then

$$\begin{aligned} A_i^{(3)} &\le A_1^{(P)} + \sum_{k=1}^{i-1} \delta_k^{(P)} + D \\ &= A_i^{(P)} + D \overset{def}{=} \hat{A}_i^{(3)}. \end{aligned} \tag{51}$$

In addition, if after p_i, no more packets ever arrive at Q_2, and denote the remaining tokens in TB_2 at time t under such case as $\sigma_2'(t)$, then

$$\sigma_2'(\hat{A}_i^{(3)}+) \ge \sigma_1(A_i^{(P)}+), \tag{52}$$

where $\sigma_1(t)$ is the remaining number of tokens in token bucket TB_1 at time t; and "$t+$", "$t-$" represent "$t + \varepsilon$", "$t - \varepsilon$" respectively, where ε is a sufficiently small positive real number.

Proof: Prove by induction.

Step 1: When $i = 1$, it is trivial to see that Formulae (51) and (52) both sustain.

Step 2: Suppose when $i = m$ ($m \ge 1$), there are

$$\begin{aligned} A_m^{(3)} &\le A_1^{(P)} + \sum_{k=1}^{m-1} \delta_k^{(P)} + D \\ &\overset{def}{=} \hat{A}_m^{(3)}, \end{aligned} \tag{53}$$

$$\text{and} \quad \sigma_2'(\hat{A}_m^{(3)}+) \ge \sigma_1(A_m^{(P)}+), \tag{54}$$

where $\sigma_2'(t)$ refers to the number of tokens in TB_2 at t if no more packets ever arrive at Q_2 after p_m.

Step 3: When $i = m + 1$, we have

$$\sigma_1(A_{m+1}^{(P)}-) = \sigma_1(A_m^{(P)}+) + \mathrm{accept}\left(\sigma_1(A_m^{(P)}+), \delta_m^{(P)}\right). \tag{55}$$

If no more packets ever arrive at Q_2 after the mth packet p_m, because of Inequality (54), (49), and the fact that TB_1 and TB_2 are both token bucket of (σ, ρ), we have

$$\begin{aligned} & \sigma_2'(\hat{A}_m^{(3)}+) + \mathrm{accept}\left(\sigma_2'(\hat{A}_m^{(3)}+), \delta_m^{(P)}\right) \\ \ge\ & \sigma_1(A_m^{(P)}+) + \mathrm{accept}\left(\sigma_1(A_m^{(P)}+), \delta_m^{(P)}\right). \\ & \text{(Due to (54) and (49))} \end{aligned} \tag{56}$$

Therefore

$$\begin{aligned} & \sigma_2'(\hat{A}_{m+1}^{(3)}) \\ =\ & \sigma_2'(\hat{A}_m^{(3)}+) + \mathrm{accept}\left(\sigma_2'(\hat{A}_m^{(3)}+), \delta_m^{(P)}\right) \qquad (57) \\ \ge\ & \sigma_1(A_m^{(P)}+) + \mathrm{accept}\left(\sigma_1(A_m^{(P)}+), \delta_m^{(P)}\right) \\ =\ & \sigma_1(A_{m+1}^{(P)}-). \qquad \text{(Due to (55))} \end{aligned} \tag{58}$$

Hence, if no more packets arrive at Q_2 after the mth packet, then $\exists t \in [A_m^{(3)}+, \hat{A}_{m+1}^{(3)}]$ (note Inequality (53) implies $A_m^{(3)}+ \leq \hat{A}_{m+1}^{(3)}$), such that $\forall \tau \geq t$ and $\tau \in [A_m^{(3)}+, \hat{A}_{m+1}^{(3)}]$, $\sigma_2'(\tau) \geq \sigma_1(A_{m+1}^{(P)}-)$. Suppose for all such t, $T = \min\{t\}$.

Note if the $(m+1)$th packet p_{m+1} *does* arrive at Q_2, p_{m+1} will not be blocked (by TB_2 at Q_2) during $[T, \hat{A}_{m+1}^{(3)}]$. Because as long as TB_2 has no less than $\sigma_1(A_{m+1}^{(P)}-)$ tokens, p_{m+1} shall pass, which is what happened at TB_1 at $A_{m+1}^{(P)}-$.

On the other hand, since D is the maximum delay possible, $A_{m+1}^{(2)} \leq A_{m+1}^{(P)} + D = \hat{A}_{m+1}^{(3)}$. Therefore, p_{m+1} must be able to leave TB_2 at a time instance no later than $\hat{A}_{m+1}^{(3)}$ (since p_{m+1} will not be blocked at Q_2 during $[T, \hat{A}_{m+1}^{(3)}]$). That is, $A_{m+1}^{(3)} \leq \hat{A}_{m+1}^{(3)}$, *Inequality (51) holds for* $i = m+1$.

Next, we prove Inequality (52) also holds for $i = m+1$.

Denote $\sigma_2''(t)$ as the number of tokens in TB_2 if no more packets arrive at Q_2 after p_{m+1}. Denote ℓ_{m+1} as the length of packet p_{m+1}.

Case 1: $A_{m+1}^{(3)} \leq \hat{A}_m^{(3)}$.

Under such case, we have

$$\begin{aligned}
\sigma_2'(A_{m+1}^{(3)}+) &\geq \sigma_2''(A_{m+1}^{(3)}+), && (59)\\
\sigma_2''(A_{m+1}^{(3)}+) &= \sigma_2'(A_{m+1}^{(3)}+) - \ell_{m+1}, && (60)\\
\sigma_2'(\hat{A}_m^{(3)}) &= \sigma_2'(A_{m+1}^{(3)}+) + \text{accept}\Big(\sigma_2'(A_{m+1}^{(3)}+),\\
&\quad \hat{A}_m^{(3)} - A_{m+1}^{(3)}\Big), && (61)\\
\text{and} \quad \sigma_2''(\hat{A}_m^{(3)}) &= \sigma_2''(A_{m+1}^{(3)}+) + \text{accept}\Big(\sigma_2''(A_{m+1}^{(3)}+),\\
&\quad \hat{A}_m^{(3)} - A_{m+1}^{(3)}\Big). && (62)
\end{aligned}$$

Due to Inequality (49) of Lemma 5,

$$(59), (61), \text{ and } (62) \Rightarrow \sigma_2'(\hat{A}_m^{(3)}) \geq \sigma_2''(\hat{A}_m^{(3)}). \qquad (63)$$

Due to Inequality (50) of Lemma 5,

$$\begin{aligned}
& (59)\\
\Rightarrow\ & \text{accept}\Big(\sigma_2''(A_{m+1}^{(3)}+), \hat{A}_m^{(3)} - A_{m+1}^{(3)}\Big)\\
\geq\ & \text{accept}\Big(\sigma_2'(A_{m+1}^{(3)}+), \hat{A}_m^{(3)} - A_{m+1}^{(3)}\Big)\\
\Rightarrow\ & \sigma_2''(A_{m+1}^{(3)}+) + \text{accept}\Big(\sigma_2''(A_{m+1}^{(3)}+),\\
& \hat{A}_m^{(3)} - A_{m+1}^{(3)}\Big)\\
\geq\ & \sigma_2'(A_{m+1}^{(3)}+) - \ell_{m+1} + \text{accept}\Big(\sigma_2'(A_{m+1}^{(3)}+),\\
& \hat{A}_m^{(3)} - A_{m+1}^{(3)}\Big) \qquad \text{(Due to (60))}\\
\Rightarrow\ & \sigma_2''(\hat{A}_m^{(3)}) \geq \sigma_2'(\hat{A}_m^{(3)}) - \ell_{m+1}. \qquad (64)
\end{aligned}$$

Meanwhile,

$$\begin{aligned}
& \text{(50) and (63)} \\
\Rightarrow\quad & \text{accept}\left(\sigma_2''(\hat{A}_m^{(3)}+), \delta_m^{(P)}\right) \\
\geq\quad & \text{accept}\left(\sigma_2'(\hat{A}_m^{(3)}+), \delta_m^{(P)}\right). \qquad (65)
\end{aligned}$$

$$\begin{aligned}
& \text{(64) and (65)} \\
\Rightarrow\quad & \sigma_2''(\hat{A}_{m+1}^{(3)}+) \\
=\quad & \sigma_2''(\hat{A}_m^{(3)}+) + \text{accept}\left(\sigma_2''(\hat{A}_m^{(3)}+), \delta_m^{(P)}\right) \\
\geq\quad & \sigma_2'(\hat{A}_m^{(3)}+) - \ell_{m+1} + \text{accept}\left(\sigma_2'(\hat{A}_m^{(3)}+), \delta_m^{(P)}\right) \\
\geq\quad & \sigma_1(A_{m+1}^{(P)}-) - \ell_{m+1} \qquad \text{(Due to (57) } \sim \text{ (58))} \\
=\quad & \sigma_1(A_{m+1}^{(P)}+). \qquad (66)
\end{aligned}$$

Case 2: $A_{m+1}^{(3)} > \hat{A}_m^{(3)}$ (note $A_m^{(3)} \leq \hat{A}_m^{(3)}$, as proven in Step 2).
In such case, $\forall \tau \in [A_m^{(3)}+, A_{m+1}^{(3)}-]$ (note $A_m^{(3)} \leq \hat{A}_m^{(3)}$), there is $\sigma_2''(\tau) = \sigma_2'(\tau)$. Particularly,

$$\sigma_2''(\hat{A}_m^{(3)}+) = \sigma_2'(\hat{A}_m^{(3)}+). \qquad (67)$$

Meanwhile

$$\sigma_2''(A_{m+1}^{(3)}+) = \sigma_2'(A_{m+1}^{(3)}) - \ell_{m+1}. \qquad (68)$$

Therefore

$$\begin{aligned}
& \sigma_2''(\hat{A}_{m+1}^{(3)}+) \\
=\quad & \sigma_2''(\hat{A}_m^{(3)}+) + \text{accept}\left(\sigma_2''(\hat{A}_m^{(3)}+), A_{m+1}^{(3)} - \hat{A}_m^{(3)}\right) \\
& -\ell_{m+1} + \text{accept}\left(\sigma_2''(A_{m+1}^{(3)}+), \hat{A}_{m+1}^{(3)} - A_{m+1}^{(3)}\right) \\
& \text{((51) holds for } i = m+1 \text{ implies } A_{m+1}^{(3)} \leq \hat{A}_{m+1}^{(3)}) \\
=\quad & \sigma_2'(\hat{A}_m^{(3)}+) + \text{accept}\left(\sigma_2'(\hat{A}_m^{(3)}+), A_{m+1}^{(3)} - \hat{A}_m^{(3)}\right) \\
& -\ell_{m+1} + \text{accept}\left(\sigma_2''(A_{m+1}^{(3)}+), \hat{A}_{m+1}^{(3)} - A_{m+1}^{(3)}\right) \\
& \qquad \text{(Due to (67))} \\
=\quad & \sigma_2'(A_{m+1}^{(3)}) + \text{accept}\left(\sigma_2''(A_{m+1}^{(3)}+), \hat{A}_{m+1}^{(3)} - A_{m+1}^{(3)}\right) \\
& -\ell_{m+1} \\
=\quad & \sigma_2'(A_{m+1}^{(3)}) + \text{accept}\left(\sigma_2'(A_{m+1}^{(3)}+) - \ell_{m+1},\right. \\
& \left.\hat{A}_{m+1}^{(3)} - A_{m+1}^{(3)}\right) - \ell_{m+1} \qquad \text{(Due to (68))} \\
\geq\quad & \sigma_2'(A_{m+1}^{(3)}) + \text{accept}\left(\sigma_2'(A_{m+1}^{(3)}+), \hat{A}_{m+1}^{(3)} - A_{m+1}^{(3)}\right) \\
& -\ell_{m+1} \qquad \text{(Due to (50))} \\
=\quad & \sigma_2'(\hat{A}_{m+1}^{(3)}) - \ell_{m+1} \\
\geq\quad & \sigma_1(A_{m+1}^{(P)}-) - \ell_{m+1} \qquad \text{(Due to (57) } \sim \text{ (58))} \\
=\quad & \sigma_1(A_{m+1}^{(P)}+).
\end{aligned}$$

Combining Case 1 and 2, Inequality (52) also holds for $i = m+1$.
From Step 1 $\sim$ 3, Lemma 6 sustains. ■

Proof of Corollary 2: According to Lemma 6,

$$\begin{aligned} & A_{SL}^{(1)}(p_f^j) + d_f^j \\ = \;& A_{SL}^{(K)}(p_f^j) \qquad \text{(Definition of } d_f^j\text{)} \\ \le \;& A_{SL}^{(1)}(p_f^j) + \max\{d_f^{k'}\} \qquad \text{(Lemma 6)} \\ \le \;& A_{SL}^{(1)}(p_f^j) + \frac{\sigma_f}{r_f} + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_f^{max}) + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{(i)} \\ & \quad \text{(Corollary 1)} \\ \therefore \;& d_f^j \le \frac{\sigma_f}{r_f} + \sum_{i=1}^{K-1} \Delta_F^{(i)}(\ell_f^{max}) + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{(i)}. \end{aligned}$$

■

APPENDIX F
PROOF OF THEOREM 6

Lemma 7 (Preemption Approximation): Without loss of generality, suppose in Node i (e.g. $i = 1$) of Fig. 6, $Q_{TH}^{(i)}$, $TB_H^{(i)}$, $Q_{SH}^{(i)}$, and $S_H^{(i)}$ make up a PAWA TBC-GPS server (i.e. $S_H^{(i)}$ is a GPS server). If $\exists F \in \mathcal{F}_\pi$ has backlog, then $\forall F' \in \mathcal{F}'_{\pi'}$ receives (approximately) 0 service rate, where $\pi < \pi'$.

Proof: Trivial due to (26). ■

Lemma 8: $\forall \pi = 1 \sim \Pi - 1$, $C_\pi^\star > R_\pi^\star$, and $C_\Pi^\star = R_\Pi^\star$.

Proof: Trivial due to (17) and (19) $\sim$ (21). ■

Lemma 9: Without loss of generality, suppose in Node i (e.g. $i = 1$) of Fig. 6, $Q_{TH}^{(i)}$, $TB_H^{(i)}$, $Q_{SH}^{(i)}$, and $S_H^{(i)}$ make up a PAWA TBC-GPS server (i.e. $S_H^{(i)}$ is a GPS server). If all flows greedy start from time 0, then $\forall \pi \in \{1, 2, \ldots, \Pi - 1\}$, $\forall F \in \mathcal{F}_\pi$ completes sending its first ℓ_F^{max} bits at $t_\pi^0 = \Delta_\pi^\star$; at $t_\pi^1 = \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star}$, for the first time all flows $F \in \mathcal{F}_\pi$ deplete their backlogs; and after t_π^1, $\forall F \in \mathcal{F}_\pi$ maintains a service rate of R_F without any more backlog.

Proof: Prove by induction:

Step 1: When $\pi = 1$, before $t_1^0 = \Delta_1^\star$, $\forall F \in \mathcal{F}_1$ gets a service rate of

$$\begin{aligned} \gamma_F &= \frac{\phi_F}{\psi_1} C_1^\star \quad \text{(Due to Lemma 7)} \\ &= \frac{\ell_F^{max}}{\ell_1^\star} C_1^\star \quad \text{(Due to (27))} \\ &= \frac{\ell_F^{max}}{C_1^\star \Delta_1^\star} C_1^\star \quad \text{(Due to (22))} \\ &= \frac{\ell_F^{max}}{\Delta_1^\star}. \end{aligned}$$

$\therefore F$ completes sending its first ℓ_F^{max} bits at $t_1^0 = \Delta_1^\star$.

Since t_1^1 is the first time when all $F \in \mathcal{F}_1$ deplete their backlogs,

$$\begin{aligned} & \ell_1^\star + R_1^\star t_1^1 = C_1^\star t_1^1 \\ \Rightarrow \;& t_1^1 = \frac{\ell_1^\star}{C_1^\star - R_1^\star} = \frac{\Delta_1^\star C_1^\star}{C_2^\star}. \quad \text{(Due to (22) and (21))} \end{aligned}$$

After t_1^1, according to GPS definition (see [13] Formula (1)) and Formulae (26), (27), $\forall F \in \mathcal{F}_1$ shall maintain a service rate of R_F without any more backlog.

Step 2: Suppose when $\pi = m$, $\forall F \in \mathcal{F}_m$ completes sending its first ℓ_F^{max} bits at $t_m^0 = \Delta_m^\star$; at $t_m^1 = \Delta_m^\star \frac{C_m^\star}{C_{m+1}^\star}$, for the first time all $F \in \mathcal{F}_m$ deplete their backlogs; and after t_m^1, $\forall F \in \mathcal{F}_m$ maintains a service rate of R_F without any more backlog.

Step 3: When $\pi = m+1$, $\forall F \in \mathcal{F}_{m+1}$ starts receiving service at t_m^1 with a service rate of $\gamma_{_F} = \frac{\phi_F}{\psi_{m+1}} C_{m+1}^\star$. Therefore it completes sending first ℓ_F^{max} bits at

$$\begin{aligned}
& t_{m+1}^0 = t_m^1 + \frac{\psi_{_F} \ell_F^{max}}{\phi_{_F} C_{m+1}^\star} \\
= \ & t_m^1 + \frac{\ell_{m+1}^\star \ell_F^{max}}{\ell_F^{max} C_{m+1}^\star} = \Delta_m^\star \frac{C_m^\star}{C_{m+1}^\star} + \frac{\ell_{m+1}^\star}{C_{m+1}^\star} \\
= \ & \frac{\Delta_m^\star C_m^\star + \Delta_{m+1}^\star C_{m+1}^\star - \Delta_m^\star C_m^\star}{C_{m+1}^\star} \quad \text{(Due to (23))} \\
= \ & \Delta_{m+1}^\star.
\end{aligned} \tag{69}$$

Since t_{m+1}^1 is the first time when all $F \in \mathcal{F}_{m+1}$ depletes their backlogs,

$$\begin{aligned}
& \ell_{m+1}^\star + R_{m+1}^\star t_{m+1}^1 \\
= \ & C_{m+1}^\star (t_{m+1}^1 - t_m^1) \quad \text{(Due to Lemma 7)} \\
\Rightarrow \ & t_{m+1}^1 = \frac{\ell_{m+1}^\star + C_{m+1}^\star t_m^1}{C_{m+1}^\star - R_{m+1}^\star} = \frac{\ell_{m+1}^\star + C_m^\star \Delta_m^\star}{C_{m+2}^\star} \\
& = \frac{\Delta_{m+1}^\star C_{m+1}^\star - \Delta_m^\star C_m^\star + \Delta_m^\star C_m^\star}{C_{m+2}^\star} \quad \text{(Due to (23))} \\
& = \Delta_{m+1}^\star \frac{C_{m+1}^\star}{C_{m+2}^\star}.
\end{aligned}$$

After t_{m+1}^1, according to GPS definition (see [13] Formula (1)) and Formulae (26), (27), $\forall F \in \mathcal{F}_{m+1}$ shall maintain a service rate of R_F without any more backlog.

Based on Step 1 $\sim$ 3, the lemma sustains. ■

Lemma 10: In addition to what claimed Lemma 9, under greedy starting, when $\pi = \Pi$, $\forall F \in \mathcal{F}_\Pi$ receives 0 service rate until $\Delta_{\Pi-1}^\star \frac{C_{\Pi-1}^\star}{C_\Pi^\star}$, then maintains a service rate of R_F.

Proof: Trivial due to Lemma 7, 9, and Formula (27). ■

Proof of Theorem 6: When $\pi = 1, 2, \ldots, \Pi - 1$, $\forall F \in \mathcal{F}_\pi$ starts receiving service at $t_{\pi-1}^1 = \Delta_{\pi-1}^\star \frac{C_{\pi-1}^\star}{C_\pi^\star}$, and finishes its first ℓ_F^{max} bits at $t_\pi^0 = \Delta_\pi^\star$. Between $t_{\pi-1}^1$ and t_π^0, F receives a constant service rate. Therefore Equation (28) naturally holds.

When $\pi = \Pi$, Equation (28) naturally holds due to Lemma 10. ■

APPENDIX G
PROOF OF THEOREM 7

Lemma 11: Without loss of generality, suppose in Node i (e.g. $i = 1$) of Fig. 6, $Q_{TH}^{(i)}$, $TB_H^{(i)}$, $Q_{SH}^{(i)}$, and $S_H^{(i)}$ make up a PAWA TBC-GPS/WFQ server (i.e. $S_H^{(i)}$ is a GPS or WFQ server). Then $\forall 0 < k < j$, there is:

$$\begin{aligned}
& A_{SH}^{(i)}(p_F^k) + \sum_{x=k}^{j} \frac{\ell_F^x}{R_{_F}} \\
\leq \ & \mathrm{GRC}_{SH}^{(i)}(p_F^j) \\
= \ & \max\{A_{SH}^{(i)}(p_F^j), \mathrm{GRC}_{SH}^{(i)}(p_F^{j-1})\} + \frac{\ell_F^j}{R_{_F}},
\end{aligned} \tag{70}$$

where $A_{SH}^{(i)}(p)$ is the time when packet p arrives at $Q_{SH}^{(i)}$, $\mathrm{GRC}_{SH}^{(i)}(p_F^j) \overset{def}{=} \max\{A_{SH}^{(i)}(p_F^j), \mathrm{GRC}_{SH}^{(i)}(p_F^{j-1})\} + \ell_F^j / R_{_F}$, $\mathrm{GRC}_{SH}^{(i)}(p_F^0) \overset{def}{=} 0$, and $F \in \mathcal{F}_\pi$.

Proof: Prove by induction:

Step 1: When $j = k + 1$,

$$\begin{aligned} & A_{SH}^{(i)}(p_F^k) + \frac{\ell_F^k}{R_F} \\ \leq\ & \max\{A_{SH}^{(i)}(p_F^k), \mathrm{GRC}_{SH}^{(i)}(p_F^{k-1})\} + \frac{\ell_F^k}{R_F} \\ =\ & \mathrm{GRC}_{SH}^{(i)}(p_F^k) \\ \leq\ & \max\{A_{SH}^{(i)}(p_F^{k+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^k)\}. \\ \therefore\ & A_{SH}^{(i)}(p_F^k) + \frac{\ell_F^k}{R_F} + \frac{\ell_F^{k+1}}{R_F} \\ \leq\ & \max\{A_{SH}^{(i)}(p_F^{k+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^k)\} + \frac{\ell_F^{k+1}}{R_F}. \end{aligned}$$

That is, Inequality (70) holds for $j = k + 1$.

Step 2: Suppose Inequality (70) holds for $j = m > k$, that is,

$$\begin{aligned} & A_{SH}^{(i)}(p_F^k) + \sum_{x=k}^{m} \frac{\ell_F^x}{R_F} \\ \leq\ & \mathrm{GRC}_{SH}^{(i)}(p_F^m) \\ =\ & \max\{A_{SH}^{(i)}(p_F^m), \mathrm{GRC}_{SH}^{(i)}(p_F^{m-1})\} + \frac{\ell_F^m}{R_F}. \end{aligned} \tag{71}$$

Step 3: When $j = m + 1$,

$$\begin{aligned} (71) \Rightarrow\ & A_{SH}^{(i)}(p_F^k) + \sum_{x=k}^{m} \frac{\ell_F^x}{R_F} \\ \leq\ & \max\{A_{SH}^{(i)}(p_F^m), \mathrm{GRC}_{SH}^{(i)}(p_F^{m-1})\} + \frac{\ell_F^m}{R_F} \\ =\ & \mathrm{GRC}_{SH}^{(i)}(p_F^m) \\ \leq\ & \max\{A_{SH}^{(i)}(p_F^{m+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^m)\} & (72) \\ (72) \Rightarrow\ & A_{SH}^{(i)}(p_F^k) + \sum_{x=k}^{m+1} \frac{\ell_F^x}{R_F} \\ \leq\ & \max\{A_{SH}^{(i)}(p_F^{m+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^m)\} + \frac{\ell_F^{m+1}}{R_F} \\ =\ & \mathrm{GRC}_{SH}^{(i)}(p_F^{m+1}). & (73) \end{aligned}$$

■

Lemma 12: Without loss of generality, suppose in Node i (e.g. $i = 1$) of Fig. 6, $Q_{TH}^{(i)}$, $TB_H^{(i)}$, $Q_{SH}^{(i)}$, and $S_H^{(i)}$ make up a PAWA TBC-GPS server (i.e. $S_H^{(i)}$ is a GPS server). Suppose $F \in \mathcal{F}_\pi$, where $\pi < \Pi$, then for $j = 1, 2, \ldots$,

$$L_{SH}^{(i)GPS}(p_F^j) \leq \mathrm{GRC}_{SH}^{(i)}(p_F^j) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star}, \tag{74}$$

where $L_{SH}^{(i)GPS}(p)$ is the time when packet p leaves GPS server $S_H^{(i)}$, $\mathrm{GRC}_{SH}^{(i)}(p_F^j) \overset{def}{=} \max\{A_{SH}^{(i)}(p_F^j), \mathrm{GRC}_{SH}^{(i)}(p_F^{j-1})\} + \ell_F^j/R_F$, $A_{SH}^{(i)}(p)$ is the time when packet p reaches $Q_{SH}^{(i)}$, and $\mathrm{GRC}_{SH}^{(i)}(p_F^0) \overset{def}{=} 0$.

Proof: Prove by induction,

Step 1: When $j = 1$, p_F^1 starts transmission at $A_{SH}^{(i)}(p_F^1)$ (though may receive nearly 0 service rate). Due to Theorem 3 and Lemma 9, $L_{SH}^{(i)GPS}(p_F^1) \leq A_{SH}^{(i)}(p_F^1) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star} + \frac{\ell_F^1}{R_F}$. Meanwhile, since $\mathrm{GRC}_{SH}^{(i)}(p_F^1) = \max\{A_{SH}^{(i)}(p_F^1), 0\} + \frac{\ell_F^1}{R_F} = A_{SH}^{(i)}(p_F^1) + \frac{\ell_F^1}{R_F}$, we have $L_{SH}^{(i)GPS}(p_F^1) \leq \mathrm{GRC}_{SH}^{(i)}(p_F^1) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star}$.

Step 2: Suppose when $j = m$ ($m \geq 1$), there is $L_{SH}^{(i)GPS}(p_F^m) \leq \mathrm{GRC}_{SH}^{(i)}(p_F^m) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star}$.

Step 3: When $j = m + 1$, we have two cases:

Case 1: If $A_{SH}^{(i)}(p_F^{m+1}) \geq L_{SH}^{(i)GPS}(p_F^m)$, that is, $Q_{SH}^{(i)}$ is empty when p_F^{m+1} arrives. Then due to Theorem 3 and Lemma 9,

$$\begin{aligned}
& L_{SH}^{(i)GPS}(p_F^{m+1}) \\
\leq\ & A_{SH}^{(i)}(p_F^{m+1}) + \frac{\ell_F^{m+1}}{R_F} + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star} \\
\leq\ & \max\{A_{SH}^{(i)}(p_F^{m+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^m)\} + \frac{\ell_F^{m+1}}{R_F} \\
& + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star} \\
=\ & \mathrm{GRC}_{SH}^{(i)}(p_F^{m+1}) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star}.
\end{aligned}$$

Case 2: If $A_{SH}^{(i)}(p_F^{m+1}) < L_{SH}^{(i)GPS}(p_F^m)$, that is, $Q_{SH}^{(i)}$ is backlogged when p_F^{m+1} arrives. Then suppose the current backlog at $Q_{SH}^{(i)}$ starts from packet p_F^k ($1 \leq k < m + 1$). That is, $Q_{SH}^{(i)}$ is empty when p_F^k arrives, and $Q_{SH}^{(i)}$ is continuously backlogged till at least $L_{SH}^{(i)GPS}(p_F^{m+1})$. Due to Theorem 3 and Lemma 9, we have

$$\begin{aligned}
& L_{SH}^{(i)GPS}(p_F^{m+1}) \\
\leq\ & A_{SH}^{(i)}(p_F^k) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star} + \sum_{x=k}^{m+1} \frac{\ell_F^x}{R_F} \\
\leq\ & \max\{A_{SH}^{(i)}(p_F^{m+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^m)\} + \frac{\ell_F^{m+1}}{R_F} \\
& + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star} \qquad \text{(Due to Lemma 11)} \\
=\ & \mathrm{GRC}_{SH}^{(i)}(p_F^{m+1}) + \Delta_\pi^\star \frac{C_\pi^\star}{C_{\pi+1}^\star}.
\end{aligned}$$

From Step 1 $\sim$ 3, Inequality (74) sustains. ∎

With similar approach, we can also prove:

Lemma 13: In addition to Lemma 12, when $F \in \mathcal{F}_\Pi$, for $j = 1, 2, \ldots$,

$$L_{SH}^{(i)GPS}(p_F^j) \leq \mathrm{GRC}_{SH}^{(i)}(p_F^j) + \Delta_{\Pi-1}^\star \frac{C_{\Pi-1}^\star}{C_\Pi^\star}, \tag{75}$$

Proof of Theorem 7: According to WFQ property,

$$\begin{aligned}
& L_{SH}^{(i)}(p_F^j) \\
\leq\ & L_{SH}^{(i)GPS}(p_F^j) + \frac{\ell_{SH}^{(i)max}}{C} \\
\leq\ & \mathrm{GRC}_{SH}^{(i)}(p_F^j) + \alpha_{H}^{'(i)}. \\
& \qquad \text{(Due to Lemma 12, 13)}
\end{aligned}$$

∎

Appendix H
Proof of Theorem 8

Lemma 14 (1-Hop GR Server Release-Guard Delay): In Fig. 6, $\forall i = 1 \sim (K-1)$, $\forall 0 \leq \ell \leq \ell_F^{max}$, and $\forall j = 1, 2, \ldots$,

$$\begin{cases} L_{SH}^{(i)}(p_F^j) \leq A_{SH}^{(i)}(p_F^j) + \frac{\ell_F^j}{R_{_F}} + \alpha_{_H}^{'(i)}, \\ \mathrm{GRC}_{SH}^{(i)}(p_F^j) = A_{SH}^{(i)}(p_F^j) + \frac{\ell_F^j}{R_{_F}}. \end{cases} \tag{76}$$

And $\forall i = 1 \sim (K-2)$,

$$L_{RG}^{(i+1)}(p_F^j) \leq A_{SH}^{(i)}(p_F^j) + \frac{\ell_F^{max}}{R_{_F}} + \alpha_{_H}^{'(i)}.$$

where $A_{SH}^{(i)}(p)$, $L_{SH}^{(i)}(p)$, and $L_{RG}^{(i)}(p)$ are the time when packet p arrives at $Q_{SH}^{(i)}$, leaves $S_H^{(i)}$, and leaves release guard $Q_{RG}^{(i)}$ respectively; $\alpha_{_H}^{'(i)}$ is the GR scheduling constant defined in Formula (30) of Theorem 7.

Proof: Due to Theorem 7,

$$\begin{aligned} & L_{SH}^{(i)}(p_F^j) \\ \leq\ & \mathrm{GRC}_{SH}^{(i)}(p_F^j) + \alpha_{_H}^{'(i)} \\ =\ & \max\{A_{SH}^{(i)}(p_F^j), \mathrm{GRC}_{SH}^{(i)}(p_F^{j-1})\} \\ & + \frac{\ell_F^j}{R_{_F}} + \alpha_{_H}^{'(i)}. \end{aligned} \tag{77}$$

In the following, we prove Formulae (76) by induction.

Step 1: When $j = 1$, $\mathrm{GRC}_{SH}^{(i)}(p_F^0) \overset{def}{=} 0$, therefore

$$\begin{aligned} & \mathrm{GRC}_{SH}^{(i)}(p_F^1) \\ \overset{def}{=}\ & \max\{A_{SH}^{(i)}(p_F^1), \mathrm{GRC}_{SH}^{(i)}(p_F^0)\} + \frac{\ell_F^1}{R_{_F}} \\ =\ & A_{SH}^{(i)}(p_F^1) + \frac{\ell_F^1}{R_{_F}}, \\ \text{and}\quad & (77) \\ \Rightarrow\ & L_{SH}^{(i)}(p_F^1) \\ \leq\ & A_{SH}^{(i)}(p_F^1) + \frac{\ell_F^1}{R_{_F}} + \alpha_{_H}^{'(i)}. \end{aligned}$$

Step 2: Suppose when $j = m$ ($m \geq 1$),

$$\begin{cases} L_{SH}^{(i)}(p_F^m) \leq A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^m}{R_{_F}} + \alpha_{_H}^{'(i)}, \\ \mathrm{GRC}_{SH}^{(i)}(p_F^m) = A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^m}{R_{_F}}. \end{cases}$$

Step 3: When $j = m+1$, we have

$$\begin{aligned}
& \mathrm{GRC}_{SH}^{(i)}(p_F^m) \\
= \; & A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^m}{R_F} \\
\le \; & A_{SH}^{(i)}(p_F^{m+1}). \qquad \text{(Due to (38))}
\end{aligned}$$

$$\begin{aligned}
\therefore \; & \mathrm{GRC}_{SH}^{(i)}(p_F^{m+1}) \\
\overset{def}{=} \; & \max\{A_{SH}^{(i)}(p_F^{m+1}), \mathrm{GRC}_{SH}^{(i)}(p_F^m)\} + \frac{\ell_F^{m+1}}{R_F} \\
= \; & A_{SH}^{(i)}(p_F^{m+1}) + \frac{\ell_F^{m+1}}{R_F}.
\end{aligned}$$

$$\begin{aligned}
\therefore \; & L_{SH}^{(i)}(p_F^{m+1}) \\
\le \; & \mathrm{GRC}_{SH}^{(i)}(p_F^{m+1}) + \alpha_H^{'(i)} \\
& \quad \text{(Due to Theorem 7)} \\
= \; & A_{SH}^{(i)}(p_F^{m+1}) + \frac{\ell_F^{m+1}}{R_F} + \alpha_H^{'(i)}.
\end{aligned}$$

From Step 1 $\sim$ 3, Formulae (76) sustain.
Specifically, we have $\forall i = 1, 2, \dots K-2$ and $\forall j = 1, 2, \dots$,

$$\begin{aligned}
& (76) \\
\Rightarrow \; & A_{RG}^{(i+1)}(p_F^j) \\
= \; & L_{SH}^{(i)}(p_F^j) \\
\le \; & A_{SH}^{(i)}(p_F^j) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}.
\end{aligned} \tag{78}$$

In the following, we prove by induction that $\forall i = 1, 2, \dots, K-2$, and $\forall j = 1, 2, \dots$, Inequality (77) sustains.

Step 1: When $j = 1$, we have $L_{RG}^{(i+1)}(p_F^0) \overset{def}{=} 0$ and $\ell_F^0 \overset{def}{=} 0$. Therefore $L_{RG}^{(i+1)}(p_F^1) = A_{RG}^{(i+1)}(p_F^1) = L_{SH}^{(i)}(p_F^1) \le A_{SH}^{(i)}(p_F^1) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}$.

Step 2: Suppose when $j = m$ ($m \ge 1$),

$$L_{RG}^{(i+1)}(p_F^m) \le A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}. \tag{79}$$

Step 3: When $j = m+1$, Inequality (79) says that $RG^{(i+1)}$ releases p_F^m no later than $A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}$, therefore for any time $t \ge A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^m}{R_F} + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}$, $RG^{(i+1)}$ should release p_F^{m+1} if p_F^{m+1} ever arrives. Meanwhile, according to Inequality (78), p_F^{m+1} arrives at $RG^{(i+1)}$ by $A_{SH}^{(i)}(p_F^{m+1}) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}$. According to Inequality (38), $A_{SH}^{(i)}(p_F^{m+1}) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)} \ge A_{SH}^{(i)}(p_F^m) + \frac{\ell_F^m}{R_F} + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}$. Therefore $L_{RG}^{(i+1)}(p_F^{m+1}) \le A_{SH}^{(i)}(p_F^{m+1}) + \frac{\ell_F^{max}}{R_F} + \alpha_H^{'(i)}$.

From Step 1 $\sim$ 3, Inequality (77) sustains. ■

Proof of Theorem 8: Due to Lemma 3 and the fact that $S_L^{(1)}$ is a GR server, we have

$$A_{SH}^{(1)}(p_f^j) \le \mathrm{GRC}_{SL}^{(1)}(p_f^j) + \alpha_L^{(1)}. \tag{80}$$

According to Lemma 3 and Lemma 14,

$$\begin{aligned} & A_{SH}^{(2)}(p_f^j) = L_{RG}^{(2)}(p_f^j) \\ \leq\ & A_{SH}^{(1)}(p_f^j) + \frac{\ell_F^{max}}{R_{_F}} + \alpha_{_H}^{'(1)}, \qquad (81) \\ & \vdots \\ & A_{SH}^{(K-1)}(p_f^j) = L_{RG}^{(K-1)}(p_f^j) \\ \leq\ & A_{SH}^{(K-2)}(p_f^j) + \frac{\ell_F^{max}}{R_{_F}} + \alpha_{_H}^{'(K-2)}. \qquad (82) \end{aligned}$$

Adding Inequality (80) to (82) together, we get

$$\begin{aligned} & A_{SH}^{(K-1)}(p_f^j) \\ \leq\ & \mathrm{GRC}_{SL}^{(1)}(p_f^j) + \alpha_L^{(1)} + (K-2)\frac{\ell_F^{max}}{R_{_F}} \\ & + \sum_{i=1}^{K-2} \alpha_{_H}^{'(i)}. \qquad (83) \end{aligned}$$

Meanwhile, since $S_H^{(K-1)}$ is a GR server, we have

$$\begin{aligned} & L_{SH}^{(K-1)}(p_f^j) \\ \leq\ & \mathrm{GRC}_H^{(K-1)}(p_f^j) + \alpha_{_H}^{'(K-1)} \\ =\ & A_{SH}^{(K-1)}(p_f^j) + \frac{\ell_f^j}{R_{_F}} + \alpha_{_H}^{'(K-1)} \\ & \quad \text{(Due to Formulae (76))} \\ \leq\ & A_{SH}^{(K-1)}(p_f^j) + \frac{\ell_F^{max}}{R_{_F}} + \alpha_{_H}^{'(K-1)}. \qquad (84) \end{aligned}$$

Therefore

$$\begin{aligned} & (83)(84) \\ \Rightarrow\ & L_{SH}^{(K-1)}(p_f^j) \\ \leq\ & \mathrm{GRC}_{SL}^{(1)}(p_f^j) + \alpha_L^{(1)} + (K-1)\frac{\ell_F^{max}}{R_{_F}} + \sum_{i=1}^{K-1} \alpha_{_H}^{'(i)} \\ \Rightarrow\ & d_f^{j\,\prime} \overset{def}{=} L_{SH}^{(K-1)}(p_f^j) - A_{SL}^{(1)}(p_f^j) \\ \leq\ & [\mathrm{GRC}_{SL}^{(1)}(p_f^j) - A_{SL}^{(1)}(p_f^j)] + \alpha_L^{(1)} + (K-1)\frac{\ell_F^{max}}{R_{_F}} \\ & + \sum_{i=1}^{K-1} \alpha_{_H}^{'(i)}. \end{aligned}$$

In addition, if packets arrive at $S_L^{(1)}$ in Conflict-Free pattern, then (80) becomes $A_{SH}^{(1)}(p_f^j) = A_{SL}^{(1)}(p_f^j)$ due to the TOSETOL trick. With same approach we can prove Formula (32). ■

Appendix I
Proof of Corollary 4

According to Lemma 6,

$$\begin{aligned}
& A_{SL}^{(1)}(p_f^j) + d_f^j \\
= \;& A_{SL}^{(K)}(p_f^j) \qquad \text{(Definition of } d_f^j\text{)} \\
\le \;& A_{SL}^{(1)}(p_f^j) + \max\{d_f^{k'}\} \qquad \text{(Lemma 6)} \\
\le \;& A_{SL}^{(1)}(p_f^j) + \frac{\sigma_f}{r_f} + (K-1)\frac{\ell_f^{max}}{R_F} + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{\prime(i)} \\
& \quad \text{(Corollary 3)} \\
\therefore \;\; & d_f^j \le \frac{\sigma_f}{r_f} + (K-1)\frac{\ell_f^{max}}{R_F} + \alpha_L^{(1)} + \sum_{i=1}^{K-1} \alpha_H^{\prime(i)}
\end{aligned} \tag{85}$$

■

Appendix J
Performance Evaluation

To evaluate the performance of GD-aggregates, we carry out a case study on a typical RTE-WAN application: underground mining. Section J-A describes the underground mining scenario and explains why it is a representative RTE-WAN application; Section J-B compares E2E delay guarantees between GD-aggregate and GR-aggregate in the context of underground mining RTE-WAN.

A. Application Scenario

Due to safety concerns, underground mining is pushing for tele-robotics and remote surveillance, so that people can operate underground mining robots and monitor the mine conditions from above the ground [29][30]. Such demand makes underground mining a representative RTE-WAN application, which is explained by Fig. 8.

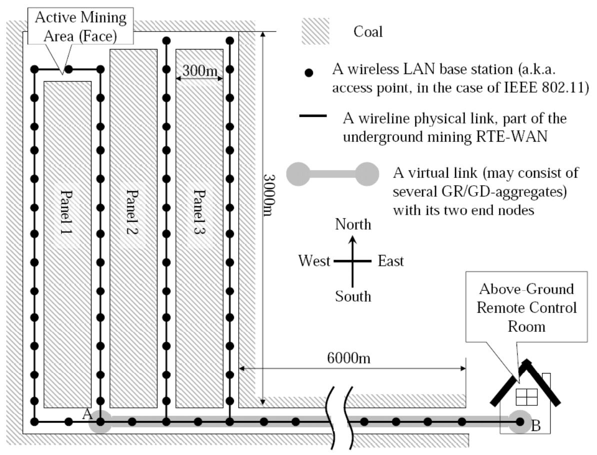

Active Mining Area (Face)
300m
Panel 1
Panel 2
Panel 3
3000m
6000m
A
B
Coal
A wireless LAN base station (a.k.a. access point, in the case of IEEE 802.11)
A wireline physical link, part of the underground mining RTE-WAN
A virtual link (may consist of several GR/GD-aggregates) with its two end nodes
North
West
East
South
Above-Ground Remote Control Room

Fig. 8. A typical underground mine and its RTE-WAN

Fig. 8 plots a longwall coal mine. Longwall is a major underground mining method (other mining methods, such as room-and-pillar and stop-and-pillar, bear similar demand and constraints on RTE-WAN) [31]. Usually, longwall mines extend several to tens of square kilometers. Tunnels in the longwall mine divide coal bodies into rectangular blocks called panels, typically of 300m wide, 3000m long, and $0.9 \sim 4.5$m in height. The mining is carried out at the width edge of a panel (called "face"). A longwall mine may have tens to even hundreds of panels, but the panels being actively mined is determined by production need, and can change over time. Fig. 8 only shows three of all panels, and only Panel 1 is being actively mined.

To allow mobile robots and vehicles, wireless base stations [32][33] (a.k.a. "access points" in IEEE 802.11) shall be deployed every $50 \sim 200$m along each tunnel. To support multi-hop critical hard real-time and large data throughput connections, these wireless base stations must be linked with wireline RTE-WAN backbone[5].

Fig. 8 reveals several features generic to underground mining RTE-WANs:

First, an underground mining RTE-WAN involves all three typical RTE-WAN traffics [31][34][35][36]:

1) hard real-time tele-robotic sensing/actuation traffics with small data throughputs and short E2E delay requirements;

2) hard real-time tele-robotic video traffics with large data throughputs and short E2E delay requirements;

3) soft real-time traffics, such as surveillance video and FTP, which may have large data throughputs, but only demand bounded E2E delays (not necessarily short).

Second, an underground mining RTE-WAN needs virtual topology. There are three reasons:

1) The scale of underground mining RTE-WAN is large: a typical underground mine extends several square kilometers to tens of square kilometers [31].

2) The physical topology of underground mining RTE-WAN is constrained: wires have to run along the grids of tunnels; and wireless LANs have to cover every segment of the tunnels to enable mobile robots/vehicles.

3) The active mining area may change as the mine evolves and as the production demand shifts.

Therefore, scalability, configurability, and flexibility are important to underground mining RTE-WAN. As pointed out in Section I, these call for virtual topology, and the building tools for virtual topologies (virtual links) are GR-aggregates and GD-aggregates. In the following, we compare the performance of the two types of aggregates.

B. E2E Delay Guarantee Comparison

Without loss of generality, we look at virtual link AB in Fig. 8, which is along the routes connecting the above-ground remote control room with robots in the active mining area, and also along the routes connecting the above-ground remote control room with surveillance cameras near Panel 1. Without loss of generality, we assume virtual link AB consists of following GR/GD-aggregates:

1) a tele-robotic sensing/actuating aggregate F_1 containing N flows, each comes with a constant packet size of 400bit, and constrained by token bucket ($\sigma_{f1} = 400$bit, $\rho_{f1} = 4$Kbps);

2) a tele-robotic video aggregate F_2 that also consists of N flows, each with a packet size of 12Kbit, and constrained by token bucket ($\sigma_{f2} = 180$Kbit, $\rho_{f2} = 4.5$Mbps);

3) a soft real-time aggregate F_3 that occupies the rest of the bandwidth, and consists of surveillance camera video and FTP flows, each with a packet size of $0.1 \sim 12$Kbit, and constrained by token bucket ($\sigma_{f3} = 180$Kbit, $\rho_{f3} = 4.5$Mbps).

Note we assume each robot creates two flows: one for sensing/actuating and one for video. Therefore both F_1 and F_2 consist of N flows for N robots. Denote $R_1 \sim R_3$ as the capacity allocated to $F_1 \sim F_3$ respectively. Then $R_1 : R_2 \equiv 4Kbps : 4.5Mbps$.

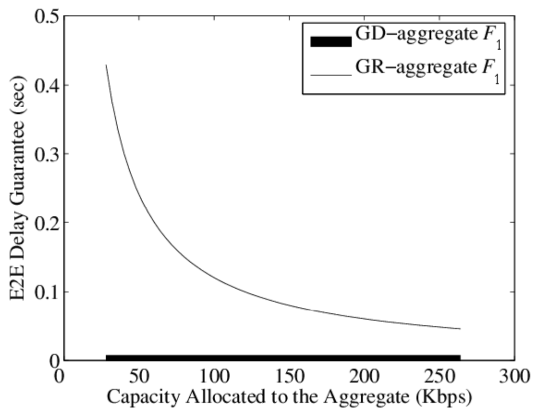
As for soft real-time aggregate F_3, it mainly consists of surveillance camera video and FTP flows. We assume the physical link capacity C is always 1Gbps, and assume the capacity demand of F_3 is at least 700Mbps, that is, $R_3 \geq 700$Mbps. This leaves the total capacity demand of tele-robotic traffics F_1 and F_2 to be no more than 300Mbps, that is $R_1 + R_2 \leq 300$Mbps. This assumption reflects the fact that tele-robotics only take place that a small portion of the mine area (the active mining area), while surveillance cameras have to cover the whole mine of tens of square kilometers.

According to [34][35], tele-robotic traffic F_1 and F_2 require E2E delay no more than 50msec, while soft real-time traffic F_3 allows E2E delay up to several seconds.

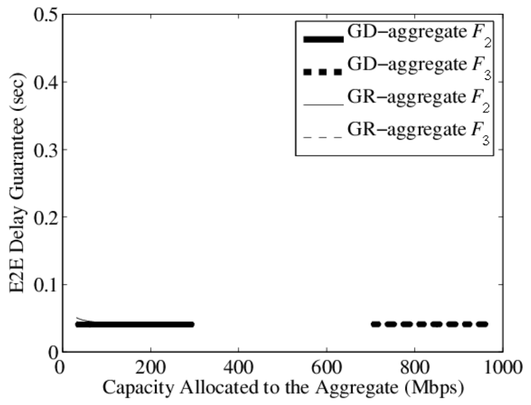
According to Fig. 8, we assume the number of underlying physical links (hops) K of virtual link AB is 31.

Fig. 9 compares the E2E delay guarantees by GD and GR-aggregates. For GD-aggregates, we follow the PAWA scheme with $\Pi = 3$. F_1, F_2, and F_3 belong to priority 1, 2, and 3 respectively. The comparison assumes aggregates' allocated capacities are fully utilized.

[5]Such design also reflects our vision on future industrial fieldbus: for industrial networks, the last hop shall be wireless to achieve mobility, while the backbone shall be wireline, so as to guarantee reliability/robustness, support large data throughput, and effectively utilize legacy infrastructure.



(a) Note we do not plot the curve for $R_1 \geq 268$Kbps because: Each flow in tele-robotic sensing/actuating aggregate F_1 corresponds to a flow in tele-robotic video aggregate F_2. When $R_1 \geq 268$Kbps, R_2 exceeds 300Mbps. This leaves no enough capacity for soft real-time aggregate F_3, whose capacity $R_3 \geq 700$Mbps.



(b) Note because soft real-time aggregate F_3 is at least of 700Mbps, we only plot curves for $R_2 \leq 300$Mbps, and $R_3 \geq 700$Mbps. The GD-aggregate and GR-aggregate's curves nearly overlap for both F_2 and F_3.

Fig. 9. Comparison of GD/GR-aggregate E2E delay guarantee for typical traffics in underground mining RTE-WAN. Aggregate $F_1 \sim F_3$ are for tele-robotic sensing/actuating, tele-robotic video streams, and soft real-time traffics (for surveillance video and FTP) respectively, and are allocated with capacity $R_1 \sim R_3$ respectively. The total physical link capacity $C = 1$Gbps.

Fig. 9(a) shows that GD-aggregate guarantees short E2E delay for tele-robotic sensing/actuation aggregate F_1, though F_1's data throughput is small. In comparison, during most of the time, GR-aggregate cannot guarantee F_1 an E2E delay less than 50msec, the acceptable maximum E2E delay.

Fig. 9(b) shows that for aggregates with large data throughputs, such as F_2 and F_3, both GD and GR-aggregates provide satisfactory E2E delay guarantees (the GD/GR-aggregates' curves almost overlap).

References

[1] L. Sha, A. Agrawala (Eds), T. Abdelzaher, C. D. Gill, R. Rajkumar, and J. A. Stankovic (Authors), "Report of NSF workshop on distributed real-time and embedded systems research in the context of GENI," *GENI Design Document 06-32 (GDD-06-32)*, September 2006.

[2] "Working group summary: Critical physical infrastructure," *NSF Cyber-Physical Systems Workshop*, October 2006.

[3] R. Gupta and K. G. Shin, "Working group summary: Infrastructure and building blocks," *NSF Cyber-Physical Systems Workshop*, October 2006.

[4] "Working group summary: Scientific foundations and education," *NSF Cyber-Physical Systems Workshop*, October 2006.

[5] M. Spong and K. Nahrstedt, "Working group summary: Break-out session on tele-interaction," *NSF Cyber-Physical Systems Workshop*, October 2006.

[6] W. Sun and K. G. Shin, "End-to-end delay bounds for traffic aggregates under guaranteed-rate scheduling algorithms," *IEEE/ACM Trans. on Networking*, vol. 13, no. 5, pp. 1188–1201, October 2005.

[7] ——, “End-to-end delay bounds for traffic aggregates under guaranteed-rate scheduling algorithms,” *EECS Dept., Univ. of Michigan, Ann Arbor, Tech. Rep.*, no. CSE-RT-484-03, 2003.
[8] Q. Wang *et al.*, “Gd-aggregate: A WAN virtual topology building tool for hard real-time and embedded applications (appendices),” *[Online] available at https://agora.cs.uiuc.edu/display/realTimeSystems/Recent+Publications*.
[9] L. L. Peterson *et al.*, *Computer Networks: A Systems Approach (2nd Ed.).* Morgan Kaufmann, 2000.
[10] P. Goyal *et al.*, “Determining end-to-end delay bounds in heterogeneous networks,” *Multimedia Systems*, no. 5, pp. 157–163, 1997.
[11] A. K. Parekh, “A generalized processor-sharing approach to flow control in integrated servcies networks,” *PhD Thesis, EECS Dept., MIT*, 1992.
[12] J. C. R. Bennett *et al.*, “WF^{2}Q: Worst-case fair weighted fair queueing,” *Proc. of INFOCOM’96*, pp. 120–128, 1996.
[13] A. Parekh *et al.*, “A generalized processor sharing approach to flow control in integrated services networks: The single-node case,” *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.
[14] J. W. Liu, *Real-Time Systems*. Prentice-Hall, Inc., 2000.
[15] J. Sun, *Fixed Priority Scheduling of End-to-End Periodic Tasks.* Ph.D. Thesis, CS Dept., UIUC, 1997.
[16] Y. Chu *et al.*, “A case for end system multicast,” *Proc. of ACM SIGMETRICS*, 2000.
[17] *An Architecture for Differentiated Services*. RFC 2475, 1998.
[18] S. Wang, D. Xuan, R. Bettati, and W. Zhao, “Providing absolute differentiated services for real-tiem applications in static-priority scheduling networks,” *IEEE/ACM Trans. on Networking*, vol. 12, no. 2, pp. 326–339, April 2004.
[19] Z. Deng and J. W.-S. Liu, “Scheduling real-time applications in an open environment,” *Proc. of IEEE RTSS’97*, 1997.
[20] T. Kuo and C. Li, “A fixed-priority-driven open environment for real-time applications,” *Proc. of IEEE RTSS’99*, 1999.
[21] I. Shin and I. Lee, “Periodic resource model for compositional real-time guarantees,” *Proc. of IEEE RTSS’03*, 2003.
[22] G. Lipari and E. Bini, “Resource partitioning among real-time applications,” *Proc. of ECRTS*, 2003.
[23] L. Almeida, “Response time analysis and server design for hierarchical scheduling,” *Proc. of IEEE RTSS’03 Work-in-Progress*, 2003.
[24] R. Davis and A. Burns, “Hierarchical fixed priority preemptive scheduling,” *Proc. of IEEE RTSS’05*, 2005.
[25] R. Davis *et al.*, “Resource sharing in hierarchical fixed priority pre-emptive systems,” *Proc. of IEEE RTSS’06*, 2006.
[26] L. Georgiadis *et al.*, “Efficient network qos provisioning based on per node traffic shaping,” *IEEE/ACM Trans. on Networking*, vol. 4, no. 4, August 1996.
[27] P. Goyal *et al.*, “Generalized guaranteed rate scheduling algorithms: A framework,” *IEEE/ACM Trans. on Networking*, vol. 5, no. 4, pp. 561–571, August 1997.
[28] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, “Voice over sensor networks,” *Proc. of IEEE RTSS’06*, pp. 291–300, December 2006.
[29] *Costs and Benefits of CSIRO Robotic Mining R & D*. Australia: Commonwealth Scientific and Industrial Research Organization (CSIRO), October 2006.
[30] “Finsch leads the way in automated mining,” *Mining Review Africa*, no. 6, pp. 32–36, 2005.
[31] H. L. Hartman *et al.*, *Introductory Mining Engineering (2nd Ed.).* Wiley, August 2002.
[32] Q. Wang *et al.*, “Building robust wireless lan for industrial control with dsss-cdma cellphone network paradigm,” *IEEE RTSS’05*, pp. 3–14, December 2005.
[33] ——, “Building robust wireless lan for industrial control with the dsss-cdma cell phone network paradigm,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 706–719, June 2007.
[34] B. Fisher *et al.*, “Seeing, hearing, and touching: Putting it all together,” *SIGGRAPH’04 Course*, 2004.
[35] M. Glencross *et al.*, “Exploiting perception in high-fidelity virtual environments,” *SIGGRAPH’06 Course*, 2006.
[36] S. Park *et al.*, “Evaluation of token bucket parameters for vbr mpeg video transmission over the internet,” *IEICE Trans. on Communications*, vol. E85-B, no. 1, January 2002.