Flow Aggregation Criteria in Networks with Rate-Guaranteeing Servers

Jinoo Joung*, Jongtae Song**, Soon Seok Lee** *Regular Members*

ABSTRACT

An effective method for calculating delay bounds of flows through flow aggregations and deaggregations is given. Based on this calculation, it is suggested a simple criteria for flow aggregation whether the aggregation will induce an increased delay bound. The criteria is evaluated in a few realistic scenarios.

Key Words : QoS, Delay bound, Flow aggregation, LR server, Aggregation Criteria

Ⅰ. Introduction

The problem of guaranteeing Quality of Service (QoS) within a packet switching network has been extensively studied and a myriad of solutions to this problem has been suggested. At the extreme corners of the complexity-performance plane, there exist two exemplary QoS management architectures: the Integrated Services (IntServ) and the Differentiated Services (DiffServ). IntServ performs ideally but not scalable at all. DiffServ is simple enough to be adopted in today's core networks, but without any guarantee on the performance. The flow aggregation technique has been suggested as an effective solution to compromise in between. This is to request and allocate resources as the IntServ would, but handling in data plane is based on flow aggregates so that the complexity problem is greatly mitigated. The impact of flow aggregation on the delay performance is, however, still an open problem. Researchers report that in some environments the aggregation of flows has no negative effect on the mean delay of the flows, and actually leads to a reduction of delay in the tail of the delay distribution for the flows[1]. There are researches that suggest the aggregation leads to a maximum delay bound reduction as well within an aggregation region[2]. The frequent flow aggregation and deaggregation, however, can yield a significant degradation on end-to-end delay performance. This fact becomes clear when we consider DiffServ networks. In a DiffServ network, flows are aggregated into a class at the output port of the every node then de-aggregated at the input port of the very next node. The consequence is that the delay bound explodes into infinity if the network utilization is above a threshold [3]. It is reasonable, at this point, to conclude that in some cases the aggregation worsens the delay and in the other cases it does not. In this research, it is investigated how we figure out whether the aggregation may take place without performance penalty.

Ⅱ. Delay bounds with generalized flow aggregation

QoS characteristics of the network with IntServ architecture have been well studied and understood by numerous researches in the past decade. Providing

※ This work was supported by the IT R&D program of MKE/IITA, Rep. of Korea [2006-S-064-01, Research on BcN Network Engineering Technologies].
* Jinoo Joung (phone: +82 2 2287 5452, email: jjoung@smu.ac.kr) is with the Department of Computer Science, Sangmyung University, Seoul, Rep. of Korea.
** Jongtae Song (email: jsong@etri.re.kr) and Soon Seok Lee (email: sslee@etri.re.kr) are with Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon, Rep. of Korea.
논문번호 : KICS2007-11-527, 접수일자 : 2007년 11월 28일, 최종논문접수일자 : 2008년 10월 21일

the allocated bandwidths, or service rates, or simply rates of an output link to multiple sharing flows plays a key role in this approach. A myriad of scheduling algorithms has been proposed. The Packetized Generalized Processor Sharing (PGPS) and Deficit Round Robin (DRR), and many other rate-providing servers are proved to be a Latency-Rate server [4], or simply LR server. All the work-conserving servers that guarantee rates can be modeled as LR servers. For a scheduling algorithm to belong to the LR server class, it is only required that the average rate of service offered by the scheduler to a busy session over every interval starting at time Θ from the beginning of the busy period, is at least equal to its reserved rate. The parameter Θ is called the latency of the scheduler.

The behavior of an LR server is determined by two parameters, the latency and the allocated rate. The latency of an LR server may be considered as the worst-case delay seen by the first packet of the busy period of a flow. It was shown that the maximum end-to-end delay experienced by a packet in a network of LR servers can be calculated from only the latencies of the individual servers on the path of the flow, and the traffic parameters of the flow that generated the packet. More specifically for a leaky-bucket constrained flow,

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \sum_{j=1}^{N} \Theta_i^{S_j}$$

where D_i is the delay of flow i within a network, σ_i and ρ_i are the well known leaky bucket parameters, the maximum burst size and the average rate, respectively, L_i is the maximum

packet length and $\Theta_i^{S_j}$ is the latency of flow i at the server S_j.

We consider the rate-guaranteeing servers with LR server model. The server has many queues. The queues are served with a rate-guaranteeing scheduler such as Weighted Fair Queueing (WFQ) or Deficit Round Robin (DRR). Among the queues, a queue is for a flow aggregates that is composed of a flow i and j. Let ρ_i and ρ_j be their data rates, respectively. We'll denote this flow aggregate with $i \oplus j$. Obviously the combined data rate of the flow aggregate $i \oplus j$ is $\rho_i + \rho_j$. The LR server in figure 1 serves the flow aggregate with this rate.

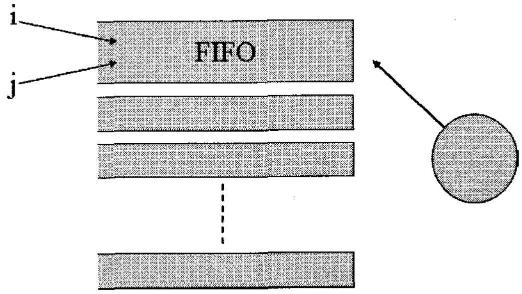


Fig. 1. An LR server with a flow aggregate

We assume there is a maximum length for packets, and denote with L. Under a condition that the packets within the flow aggregate is served in FIFO manner, we argue that the service given to an individual flow i is lower bounded as the following.

Lemma 1. Under a condition that flows i and j are leaky-bucket constrained, and i and j are aggregated into a FIFO queue, during a flow i busy period an LR server can provide service to flow i as the following:

$$W_i^S(T_0, t) \ge \max(0, (\rho_i + \rho_j)(t - T_0 - \frac{\sigma_j}{\rho_i + \rho_j} - \Theta_{i \oplus j}^S)),$$

where T_0 is the starting time of flow i busy period.

Proof. We omit the proof because of the space limitation. The proof takes a similar process with that of Lemma 5 of [5].

Without loss of generality, the flow j in the above lemma can be replaced with a set of flows, each is constrained with a leaky bucket. Let us denote the flow aggregate, including i, with I. Similarly denote with ρ_I and σ_I the leaky bucket parameters of I. The following theorem is a direct consequence from the definition of LR servers and lemma 1.

Theorem 1. Under conditions that all the flows in a flow aggregate are leaky bucket constrained in front of an LR server S, and the aggregated data rate is less than the link capacity, the LR server for the flow aggregate is still an LR server for individual flows with latency given as the following:

$$\Theta_i^S = \frac{\sigma_{I\ominus i}^S}{\rho_I} + \Theta_I^S ,$$

where I is the flow aggregate, to which i belongs in the server, and $\sigma_{I\ominus i}^S$ is $\sum_{k\in I, k\neq i} \sigma_k^S$.

The end-to-end delay of a network with LR servers can be obtained by the following inequality from [4].

$$D_i \le \frac{\sigma_i}{\rho_i} + \sum_{n=1}^{N} \Theta_i^{Sn},$$

where Sn is the *n*th server from the entrance of a network.

Now consider a series of N consecutive LR servers, S1 to SN, in a network. Obviously there can be more than N servers in the network. S1 and SN represent an entrance edge and an exit edge of the network, respectively. Consider flows i and j, which have the same path from S1 to SN. Should we aggregate the two flows? Let us assume that the LR servers in the network are all Packetized General Processor Sharing (PGPS) servers or equivalently WFQ servers, which show the best performance in terms of both delay and fairness among many LR servers. Let us denote by Θ_i^{Net} the network latency, or equivalently the sum of latencies in the network for flow i. The network latency in the case without aggregation is given as

$$\Theta_i^{\mathrm{Net\ without\ aggr}} = N\left(\frac{L_i}{\rho_i} + \frac{L_{\max}}{r}\right).$$

Now, for simplicity of calculation without losing the essential characteristics of the flow aggregation, we assume $L_i = L_j = L_{\max}$. The network latency in the case with aggregation of two flows i and j is given as

$$\Theta_i^{\mathrm{Net\ with\ aggr}} = \frac{\sigma_j}{\rho_i + \rho_j} + N\left(\frac{L}{\rho_i + \rho_j} + \frac{L}{r}\right).$$

The delay bound increment due to aggregation is given as

$$D^{\mathrm{aggr}} = \frac{\sigma_j - NL\rho_j/\rho_i}{\rho_i + \rho_j}.$$

Note that the delay bound can be actually reduced by aggregation. D^{aggr} is proportional to σ_j, and is inversely proportional to N, L, and ρ_j/ρ_i. If $D^{\mathrm{aggr}} > 0$ then $\frac{\sigma_j}{NL} > \frac{\rho_j}{\rho_i}$. We can substitute j in the above argument to $I\ominus i$, where I is the flow aggregate including i. In such a case σ_j becomes $\sigma_{I\ominus i} = \sum_{k\in I, k\neq i} \sigma_k$, and ρ_j becomes $\rho_{I\ominus i} = \sum_{k\in I, k\neq i} \rho_k$. If we want to make sure that any of the flows does not suffer from increased delay bound, then we should check the following criteria for flow aggregation.

Criteria 1. Aggregation of flows can be considered to induce no delay penalty if the following inequality holds:

$$\frac{\sigma_{I\ominus i}}{NL} < \frac{\rho_{I\ominus i}}{\rho_i},$$

for any i within I, where I is the intended flow aggregate.

Ⅲ. Numerical investigation

Let us assume a core network with a fixed number of hop count N = 10 and a fixed link capacity r = 149.760 Mbps. Further let in this network there are a bunch of VoIP calls that may be aggregated. If we adopt the TSPEC parameters defined for IEEE 802.11e wireless LAN, then we can use, for example G.711 Voice over IP, the mean data rate ρ = 83 Kbps, maximum burst size σ = 576 bytes, and maximum packet length L = 208 bytes [6]. The values of the parameters used in this scenario are listed in Table 1. The values for the other parameters have chosen accordingly.

With such homogeneous flows being aggregated,

Table 1. Parameter values used in the Wireless LAN VoIP scenario

Parameter	Value
Number of hop count (N)	10
Link capacity (r)	149.760 Mbps
mean data rate (ρ)	83 Kbps
Maximum burst size (σ)	576 bytes
Maximum packet length (L)	208 bytes

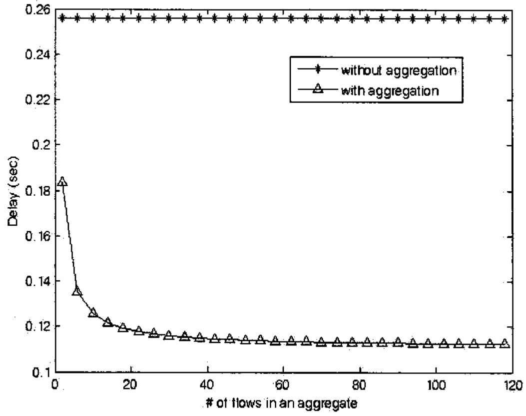


Fig. 2. The delay bounds without aggregation, with aggregation, and the delay bound increment due to aggregation in second: N=10, σ=576 bytes for all flows, ρ =83kbps for all flows, L=208 bytes, r= 149.760Mbps for all PGPS server S

from the criteria we can easily determine that any number of flows may be aggregated without delay penalty. In fact, with more flows being aggregated, the less delay bound we get as Figure 2 suggests.

An important fact in a core network, however, is that the maximum burst size of a flow at the entrance of the network is not what the end user has specified. It is a function of the latencies of servers in the path it has taken. The output traffic of flow i from server S conforms to the leaky bucket model with parameters $(\sigma_i + \rho_i \Theta_i^S,\ \rho_i)$ [4]. Therefore at the end of the series of N LR servers, the flow's maximum burst size becomes

$$\sigma_i^{\mathrm{out}} = \sigma_i^{\mathrm{in}} + \rho_i \sum_{n=1}^{N} \Theta_i^{S_n}.$$

For example, in the scenario we evaluated with parameters in table 1, the maximum burst size at the exit of the network (We will call this the burst-out. In contrast, the maximum burst size of a flow at the entrance of a network will be called burst-in.) is 2657 bytes. The effect of varying burst-in is listed in the Figure 3. In Figure 3 the number of flows in an aggregate is fixed at 20.

Another scenario we consider is the residential network environment where the maximum number of hops and the number of flows are confined and predictable. Moreover in such networks the demand for real-time service is strong, especially for video and

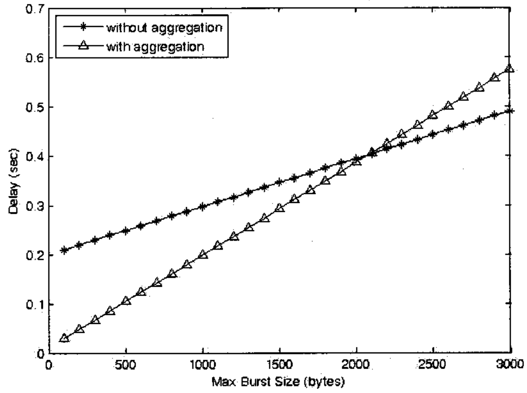


Fig. 3. The delay bounds without aggregation, with aggregation, and the delay bound increment due to aggregation in second: N=10, number of flows in the aggregate = 20, ρ=83kbps for all flows, L=208 bytes, r = 149.760Mbps for all PGPS servers

high quality audio applications. We assume the 100Mbps Fast Ethernet links are used across the network. If we are to transmit the MPEG-2 Transport Streams (TS) data whose lengths are fixed at 188 bytes with 12 bytes RTP fixed header, 4 bytes RTP video-specific header, 8 bytes UDP header, 20 bytes IP header and finally 26 bytes Ethernet header and trailer including preamble, then the maximum packet length in this case becomes 258 bytes. Considering the extended headers fields and Ethernet inter-frame gap, we set our maximum packet length at 300 bytes. Although the maximum burst size is not specified, the HD MPEG-2 video stream usually requests about 20Mbps bandwidth, while the SD stream requests about 8Mbps. The parameter values are summarized in Table 2.

Table 2. Parameter values used in the Residential Ethernet scenario

Parameter	Value
Number of hop count (N)	4
Link capacity (r)	100 Mbps
mean data rate (ρ)	20 Mbps
Number of flows in an aggregate	4
Maximum packet length (L)	300 bytes

In this environment, usually 3 or more channels from the set-top box are destined for a single DTV decoder. The delay bound increments due to aggregation with varying maximum burst size are calculated and summarized in Figure 4.

Again, the aggregation gives a negative impact on delay performance when the maximum burst size is large.

Let us finally consider a case when different types of flows are aggregated. Let us assume that in the core network of the first example the VoIP flows and HD MPEG-2 video streams are aggregated as follows. We first let 2~100 VoIP flows and 1~3 HD streams be aggregated with prescribed traffic parameters (N=10, σ_{VoIP}=576 bytes for all VoIP flows, σ_{HD}=1800 bytes for all HD streams, ρ_{VoIP}=83kbps for all VoIP flows, ρ_{HD}=20Mbps for all HD streams, L=300 bytes, r=149.760Mbps). Throughout every parameter set we tested, we obtained the negative delay increment value, which means the flow aggregation actually improved the delay performance. With only a slightly larger burst size for VoIP flows (σ_{VoIP}=1728 bytes), however, the criteria for HD streams begins to be not met, which means the VoIP flows get still better delays while the HD streams suffer from increased delay bounds due to the large burst size of VoIP flows.

Ⅳ. Conclusion

We have shown that the rate-guaranteeing servers with aggregated FIFO queues are still an LR servers for *individual* flows, and obtained the latency of such a server to individual flows. We then suggested a simple criteria for flow aggregation, by which it can be tested whether the aggregation will increase the end-to-end delay bound. We have investigated a few example scenarios with this criteria. In some of scenarios we looked into, the delay performance of flow aggregates are remarkably fine. We have shown that in some cases where maximum bust size is small and number of hops in aggregation region is large enough, the well known performance- complexity compromise is not applicable. In other words, reducing complexity by aggregating flows while achieving a better delay is indeed possible, only if the conditions on maximum burst size and number of hop count in the aggregation region are met. Furthermore, one can exactly estimate whether a network's parameters meet such a condition, by the criteria derived from this work. The maximum burst size especially at entrance of a core network, however, is not what is specified by the end user. It is rather a function of number of hops that the flow has previously passed through. In the future work, the combined effect of burst accumulation and flow aggregation should be taken into consideration.

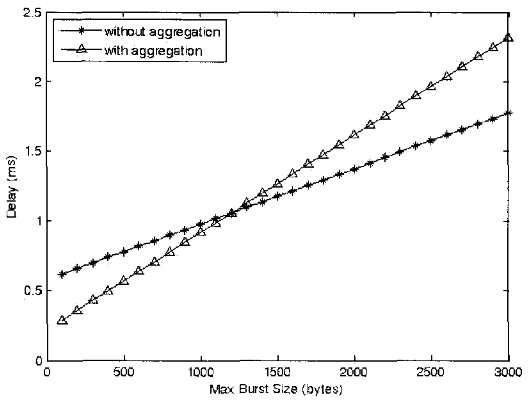


Fig. 4. The delay bounds without aggregation, with aggregation, and the delay bound increment due to aggregation in millisecond: N = 4, 4 flows are aggregated, ρ=20Mbps for all 4 flows, L=300 bytes, r = 100Mbps for all PGPS servers

References

[1] D. Clark, S. Shenker, and L. Zhang, Supporting Real Time Applications in an Integrated Services Packet Network: Architecture and Mechanism, in Proc. SIGCOMM '92, Sep. 1992.

[2] W. Sun and K. G. Shin, End-to-End Delay Bounds for Traffic Aggregates Under Guaranteed-Rate Scheduling Algorithms, IEEE/ACM Transactions on Networking, Vol.13, No.5, Oct. 2005.

[3] A. Charny and J.-Y. Le Boudec, Delay bounds in a network with aggregate scheduling, In Proc. First International Workshop of Quality of Future Internet Services (QOFIS), 2000.

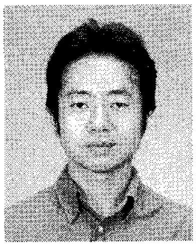
[4] D. Stiliadis, Traffic Scheduling in Packet-Switched Networks: Analysis, Design and Implementation, Ph.D. Dissertation, U.C. Santa Cruz, June 1996.

[5] J. Joung, B. Choe, H. Jeong, and H. Ryu, Effect

of Flow Aggregation on the Maximum End-to-End Delay, LNCS Vol.4208, 2006.

[6] Dimitris Skyrianoglou, Nikos Passas, and Apostolis K. Salkintzis, ARROW: An Efficient Traffic Scheduling Algorithm for IEEE 802.11e HCCA, IEEE Transactions on Wireless Communications, Vol.5, No.12, Dec. 2006.

정 진 우 (Jinoo Joung) 정회원



1992년 KAIST 전기전자공학과 공학사

1997년 Polytechnic Univ., NY, USA, 공학박사 (Ph.D.)

1997년~2001년 삼성전자 중앙연구소

2001년~2005년 삼성종합기술원

2005년~현재 상명대학교 컴퓨터과학부

<관심분야> 유무선 네트워크 QoS 및 Embedded Network Systems design

송 종 태 (Jongtae Song) 정회원

1990년 KAIST 전기전자공학과 공학사

1994년 Univ. of Southern California, USA, 공학석사

1998년 Polytechnic Univ., NY, USA, 공학박사 (Ph.D.)

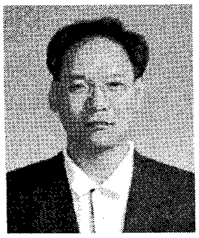
1998년~2001년 Bell Labs Lucent Technologies

2001년~2004년 Coree Networks, Parama Networks

2004년~현재 ETRI 융합네트워크연구팀

관심분야: BcN network architecture, QoS control, Flow-based network control architecture.

이 순 석 (Soon Seok Lee) 정회원



1988년 성균관대학교 산업공학과 (공학사)

1990년 성균관대학교 산업공학과 (공학석사)

1993년 8월 성균관대학교 산업공학과 (공학박사)

1993년 7월~현재 한국전자통신연구원 융합네트워크연구팀장

<관심분야> Network Architecture 설계, 네트워크 진화 전략, All-IP 융합 네트워크, 차세대 광인터넷, 개방형 통신기술, 네트워크 최적설계, 트래픽 엔지니어링, 네트워크 및 통신시스템 성능평가