Network Working Group
Internet-Draft
Expires: February 2, 2007

E. Lochin
National ICT Australia
L. Dairaine
ENSICA - LAAS/CNRS
G. Jourjon
National ICT Australia

August 2006

Guaranteed TCP Friendly Rate Control (gTFRC) for DiffServ/AF Network
draft-lochin-ietf-tsvwg-gtfrc-02

Status of this Memo

Copyright Notice

Abstract

This memo introduces gTFRC, a TCP-Friendly Rate Control providing throughput guarantee over the DiffServ/AF class. gTFRC is largely based on TFRC [2]. It provides a mean to take into account the quality of service negotiated with the network. As a result, the mechanism is able to reach a minimum throughput guarantee whatever

the flow's RTT and target rate.

Table of Contents

1. Introduction

This memo introduces gTFRC, a TCP-Friendly rate control providing throughput guarantee for unicast flows over the DiffServ/AF class. gTFRC is an adaptation of the TCP-friendly Rate Control (TFRC) [2]. This document only specifies the modification of TFRC and do not present the core TFRC mechanism that remains unchanged.

TFRC is a congestion control mechanism for unicast flows operating in a best-effort Internet environment [2]. Based on the TCP throughput equation, it is designed to be reasonably fair when competing for bandwidth with TCP flow. It generates a flow with a much lower variation of throughput over time than TCP. As a result, it is particularly suitable for multimedia application such as video streaming or telephony over Internet.

The DiffServ Assured Forwarding Class [1] has been designed to provide a guaranteed minimal throughput that many multimedia applications can take advantage of. The service offers is called Assured Service (AS) and built over the AF PHB. The minimum guaranteed throughput (also called target rate) is supposed to be known after a negotiation phase involving application level software. Adaptive application can make use of this guarantee, allowing to rely on a minimum rate when the network is congested, and possibly using higher rate otherwise. In this service class, a congestion control is required in such a way to discover the current available bandwidth and share it fairly with other competing flows. Nevertheless, due to the minimum bandwidth guarantee, the congestion control mechanism should never reduce the flow throughput at a value less than the negotiated guaranty.

When TFRC is used over a DiffServ/AF network, in spite of a good behavior in term of available bandwidth sharing, it not always reach the target rate. Even if the target rate is finally reached, a long time can happened (several tens of seconds) before the flow rate converges to this value. Then, depending on end-to-end delay and the loss probability of the various connections, the application does not obtained the requested target rate it should, even if the underlying network provides an adequate throughput guarantee.

This document suggests a simple approach to solve this problem. A minimal adaption of TFRC allows the application to quickly reach its target rate whatever the RTT value of the application's flow, while still sharing fairly the available bandwidth over the various TCP-friendly connections.

2. Guaranteed TCP Friendly Rate Control

In the context of the Additive Increase Multiplicative Decrease approaches like TCP, the only way to obtain a service differentiation with TCP protocol is to use DiffServ traffic conditioners. Indeed, the AIMD principle do not use the instantaneous TCP throughput as an input value for its congestion control and then can not make direct use of the target rate value. On the contrary, to compute the actual sending rate, TFRC uses the current rate in conjunction with the RTT and the loss event of flow. Nevertheless, the TCP equation that drives TFRC does not take into account the minimum guaranteed part of the network capacity.

gTFRC is made aware of the target rate value which is integrated into the transmit rate equation. Thanks to this knowledge, the application's flow is sent in conformance with the negotiated QoS while staying TCP-friendly in its out-profile part.

2.1. Transmit rate equation

The transmit rate is computed at sender side as the maximum between the TFRC rate estimation and the target rate. The throughput equation used in gTFRC is:

$$G = \max(g, X)$$

Where:

G is the transmit rate in bytes/second.

g is the target rate in bytes/second.

X is the transmit rate in bytes/second computed by the TCP throughput algorithm specified in RFC 3448 [2].

The rest of the gTFRC mechanism follows entirely the TFRC specification given in RFC 3448 [2].

2.2. Target rate default value

The target rate g MUST have a default value of zero byte/second. In this case, the default behavior of gTFRC corresponds to TFRC.

2.3. Target rate setting

gTFRC requires the knowledge of the target rate the DiffServ/AF network service provides to the session. This knowledge MAY be achieved by the use of a new socket option.

3. Simulation of gTFRC

3.1. Model and hypothesis

gTFRC has been evaluated over a DiffServ network using ns-2.28 simulator. gTFRC has been implemented from the TFRC ns-2 code base. The Nortel DiffServ model [3] has been used as QoS testbed.

The network architecture is shown in the following figure.

```
----                                                        ----
|s1|--------                                      --------- |r1|
----  10 Mb \                                    /  10 Mb   ----
      5 ms   \                                  /    x ms
              \------       ------       ------/
               |edge|-------|core|-------|edge|
              /------       ------       ------\
             /        10 Mb          1 Mb       \
----        /          5 ms          10ms        \          ----
|s2|--------                                      --------- |r2|
----  10 Mb                                         10 Mb   ----
       5 ms                                          y ms
```

where x and y take different RTT values in function of the experiment.

Figure 1

In these experiments, the objective was to compare the performance of TFRC and gTFRC.

The simulation has been achieved with the two following scenarios:

1. the network is exactly-provisioned (it means there is no excess bandwidth for the out-profile traffic).

2. network is over-provisioned (when there is excess bandwidth).

A network is under-provisioned when the amount of in-profile traffic is higher than the resource allocated to the AF class. This case is considered as a bad network provision and then is excluded from the field of this study.

In the simulations:

o packet size is fixed to 1500 bytes;

- we use a two colors token bucket marker with a bucket size of 10^4 bytes defined in RFC 2697 [5];

- the queues size are 50 packets and RIO parameters are: (MIN_out,MAX_out,P_out, MIN_in,MAX_in,P_in) = (10,20,0.1,20,40,0.02);

- the bottleneck between the core and the egress router is 1000Kbits/s;

- measurements are carried during 100 seconds.

For each experiments, we evaluate the throughput at the server side and compute the instantaneous throughput and the average throughput for the experiment. We resport the instantaneous throughput values at 20s, 50s and 100s. Because some flow can cross one or several DiffServ domains and then, obtain a very large RTT difference, we compare flows with a high RTT difference (i.e., 600ms).

3.2. Results

The following table presents the comparative results between TFRC and gTFRC for an exactly provisioned network.

Protocol #flow	RTT (ms)	Target (Kb/s)	After 20s	After 50s	After 100s	Average
TFRC#1	640ms	800	376	584	784	571
TFRC#2	40ms	200	584	416	232	419
gTFRC#1	640ms	800	376	784	800	722
gTFRC#2	40ms	200	584	224	200	271

Figure 2

The following table presents the comparative results between TFRC and gTFRC for an over-provisioned network with either same or different RTT values for the competing flows.

Protocol	RTT (ms)	Target (Kb/s)	After 20s	After 50s	After 100s	Average
TFRC#1	250ms	700	296	744	744	654
TFRC#2	250ms	100	704	256	248	319
gTFRC#1	250ms	700	744	800	696	727
gTFRC#2	250ms	100	256	200	304	254
TFRC#1	640ms	600	376	520	608	504
TFRC#2	40ms	200	584	480	400	489
gTFRC#1	640ms	600	376	600	600	554
gTFRC#2	40ms	200	584	408	400	439

Figure 3

Extended results of this simulation campaign are available in [6]

3.3. Analysis

From these simulations, we see that gTFRC allows to reach a target rate more quickly than TFRC. This is true whatever the RTT or the target rate of the flow. The reason is obvious since at the first rate decrease evaluation of the algorithm, gTFRC returns a rate equal to the target rate. If the evaluated rate is higher than the target rate, the classical TFRC algorithm is applied. Concerning the DiffServ network behavior, the use of gTFRC raises the number of in-profile packets in the network and avoid the problem of the bandwidth sharing of the out-profile traffic. For information purpose, concerning the Figure 2, between TFRC and gTFRC, the number of in-profile traffic raises from 73.7% to 90.16%.

3.4. Discussion

3.4.1. Security concern

As we give the possibility to the application to instantiate through a setsockopt() function the target rate negotiated between the QoS network and the application, we can imagine that a misbehaving person could abuse of this functionality by giving an higher value to the guarantee g. In this case, the misbehaving person sends an UDP-like

traffic and increases its out-profile traffic. In the context of a DiffServ/AF class, the edge router will still mark in-profile the packets in respect with the negotiated profile and out-profile the excess part. As as result, in case of network congestion, the dropping precedence set in the core router will drop this excess traffic. The misbehaving person will not take advantage of the situation as the number of losses of its own flow increases as well. Then, the in-profile traffic remains protected in the network and the out-profile traffic perceives a kind of flooding attack. As the out-profile traffic is a best-effort traffic, the use of gTFRC does not disturb the DiffServ network.

3.4.2. Case of under-provisioned network

In the context of a DiffServ/AF class, a network is under-provisioned when the amount of in-profile traffic is higher than the resource allocated to the AF class. This case could occur if the Bandwidth Broker [4] of a DiffServ network sends or receives false information. In a DiffServ context, if the gTFRC source emits below its target rate and if the gTFRC flow gets losses, it means that the in-profile traffic is no guaranteed anymore in the network. In order to tackle this problem, two approaches are possible. The first one is to pursue to emit at the guarantee g. This behaviour is legitimate since the service provider must provide to the client the service that it paid for. The second one is to react to this congestion. This can be done by adding a second threshold (y) to gTFRC . This threshold can be applied as following: if the emission rate X returned by the receiver is y times below the target rate g, the sender MUST emit to X. In the case where X < g/y, it means that a bunch of losses has occurred in the in-profile part and that the congestion could be due to a wrong setting. We believe that this problem should not be solved inside gTFRC itself and should remain under the responsibility of the service provider.

4. Acknowledgements

This research work has been conducted in the framework of the EuQoS European project (http://www.euqos.org). The authors has been supported by funding from National ICT Australia (NICTA). The authors are also grateful Sally Floyd and Mark Allman and James M. Polk for useful discussions.

5. References

[1] Heinanen, J., Weiss, W., Wroclawski, J., and J. Heinanen, "Assured Forwarding PHB Group", RFC 2597, STD 1, June 1999.

[2] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, STD 1, January 2003.

[3] Pieda, P., Ethridge, J., Baines, M., and F. Shallwani, "A Network Simulator Differentiated Services Implementation", Open IP , Nortel Networks, available at http://www.isi.edu/nsman/ns, July 2000.

[4] Nichols, K., Jacobson, V., and L. Zhang, "A two-bit differentiated services architecture for the internet", RFC 2638, STD 1, July 1999.

[5] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, STD 1, September 1999.

[6] Lochin, E., Dairaine, L., and G. Jourjon, "gTFRC: a QoS-aware congestion Control Algorithm", 5th International Conference on Networking (ICN'2006) , October 2005.

Authors' Addresses

Emmanuel Lochin
National ICT Australia
Australia Technology Park
Eveleigh, NSW 1430
Australia

Phone: +61 8374 5541
Email: Emmanuel.Lochin@nicta.com.au
URI: http://www.nicta.com.au/

Laurent Dairaine
ENSICA - LAAS/CNRS
1, place Emile Blouin
Toulouse, Cedex 5 31056
France

Phone: +33 5 61 61 85 00
Email: Laurent.Dairaine@ensica.fr
URI: http://www.ensica.fr/

Guillaume Jourjon
National ICT Australia
Australia Technology Park
Eveleigh, NSW 1430
Australia

Phone: +61 8374 5206
Email: Guillaume.Jourjon@nicta.com.au
URI: http://www.nicta.com.au/

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.