On Congestion Control for Interactive Real-time Applications in Dynamic Heterogeneous 4G Networks

Filipe Abrantes, and Manuel Ricardo

***Abstract*— Real-time multimedia applications, such as VoIP, video conferencing or on-line gaming are key applications to the success of 4G. In today's Internet these applications are not subject to congestion control, therefore the growth of popularity of these applications may endanger the stability of the Internet. On the other hand, 4G networks will be much more dynamic due to mobility and multi-homing, and the network conditions may vary abruptly requiring a fast response from the control mechanism.**

In this article we compare the feedback-based to the reservation-based congestion control approach and focus on the first one, by evaluating some mechanisms with respect to *Media Friendliness*, *Scalability* and *Dynamic Behaviour*. We also present a set of requirements for the *ideal* congestion control mechanism of real-time flows in 4G networks.

***Index Terms*— Congestion control, 4G networks, Multimedia transport, Heterogeneous networks, QoS.**

I. Introduction

In today's Internet, real-time applications such as VoIP, videoconferencing and on-line gaming mostly use RTP over UDP or UDP alone to transport data. Because these protocols are unresponsive to congestion events, the growing popularity of applications that use them endangers the stability of the Internet [1]. So, to make it possible that real-time applications are widely adopted, common congestion control mechanisms suitable for real time multimedia are expected to be deployed.

Also, a variety of wireless and wired technologies have been developed in the past years. The vision for the next generation of mobile communications networks consists in having these technologies integrated and handovers between them occurring seamlessly. These handovers may cause that during a connection the bandwidth available varies in one or more orders of magnitude. More volatile scenarios, such as ad hoc or sensor networks, are also expected. Most probably, next generation terminals will be multi-homed and will act as mobile routers.

For these reasons, the control of real time flows in 4G networks is still an unsolved issue. New solutions are required so that the network stability is maintained even when conditions vary abruptly, and the quality perceived by interactive real-time applications is not degraded by the mechanisms controlling the flow.

The article addresses these issues. In Section II existing congestion control approaches are described and compared. In Section III the feedback-based congestion control mechanisms proposed in the past are further discussed. In Section IV, we define 3 parameters which are used to evaluate congestion control mechanisms. In Section V we describe the simulations carried out using selected congestion control algorithms and discuss the results obtained. Based on these results, we discuss in Section VI the future directions for researching this topic, and introduce a set of requirements that new algorithms must meet, in order to be usable in 4G networks. Finally, we conclude the article in Section VII.

II. Congestion Control Overview

Congestion control consists of two main procedures: the network conditions estimation procedure, and the source control procedure. We compare them for both the reservation-based and the feedback-based congestion control approaches.

A. Reservation-based approach

In reservation-based architectures, the estimation of network conditions is done at the beginning of a connection using admission control, resource reservation and, sometimes, a signaling protocol. If the resources requested by the application are available then the flow is admitted; otherwise it is rejected. After admittance, the source is controlled to guarantee that it does not exceed the utilization requested, usually through *token bucket filters* (TBF), which regulate its sending rate.

Besides congestion control, these procedures are also used to provide QoS to individual or aggregated flows. They introduce complexity in network elements, in terms of processing time and memory usage; not only do these nodes need to perform resource reservation and understand a common signaling protocol, but they also need to monitor incoming flows. For this reason, such architectures are hard to deploy in a larger scale and are specially hard to implement in dynamic wireless networks, since in handovers the admittance and reservation procedures need to be repeated or transferred to the new link. Examples of reservation-based architectures include the Intserv [2] and Diffserv [3] models, as well as the SWAN [4] architecture which is a mixture of the previous two specially designed for ad-hoc networks.

Manuscript received March 17, 2005. This work was funded by the Portuguese Science and Technology Foundation.

Filipe Abrantes (e-mail: fla@inescporto.pt) is with INESC Porto, Porto, Portugal; Manuel Ricardo (e-mail: mricardo@inescporto.pt) is with FEUP and INESC Porto, Porto, Portugal.

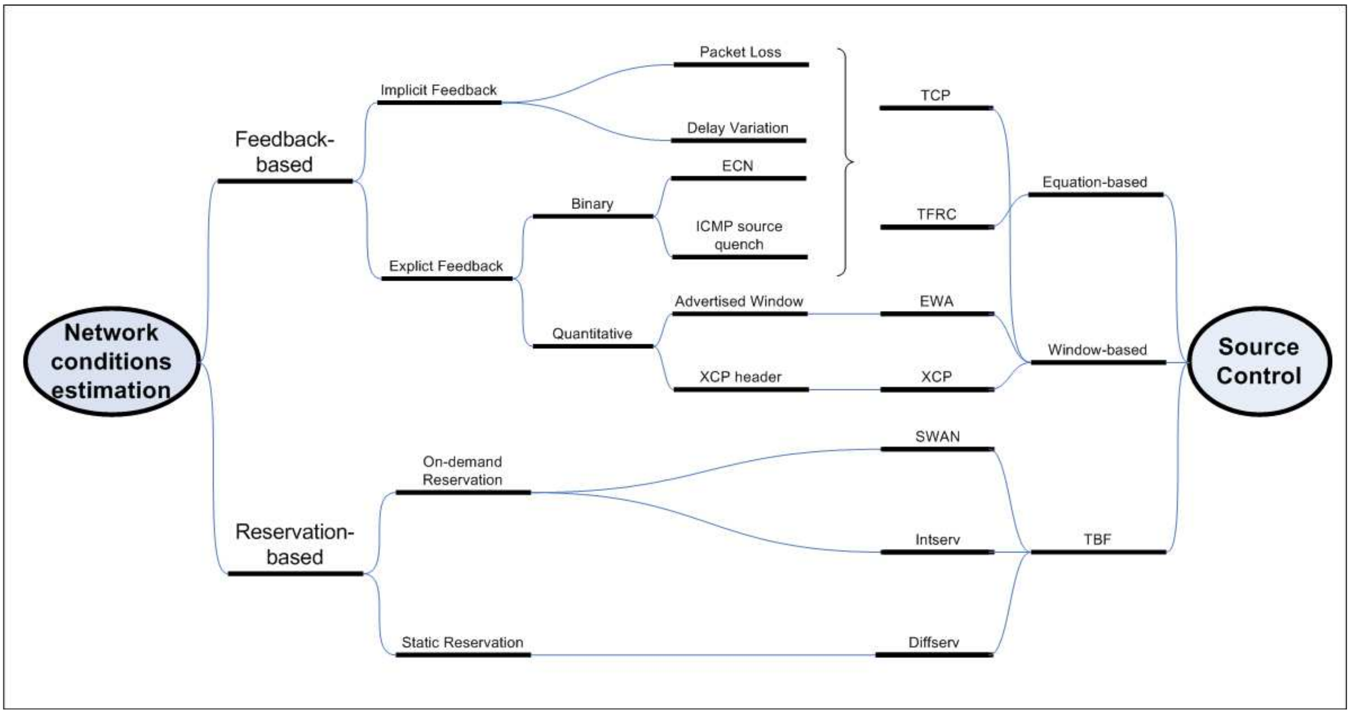


Fig. 1. Congestion Control overview.

B. Feedback-based approach

In feedback-based models, neither the nodes perform resource reservation nor do they know the path's characteristics before a connection is initiated. The feedback-based approach consists of a set of mechanisms usually applied in the transport protocol, which estimate network conditions based on feedback received from the network and control the rate of the source accordingly.

The rate of a source can be controlled using one of two methods: 1) by regulating the inter-packet sending interval (*equation-based*), or 2) by regulating the amount of outstanding packets that a flow may have in the network (*window-based*). Outstanding packets are those not yet acknowledged by the receiver; the number of outstanding packets is the *congestion window* (*cwnd*).

The feedback used to estimate network conditions may be either implicit or explicit. Implicit feedback uses information inferred from events, such as packet drop and end-to-end delay variation to identify congestion in the network; explicit feedback consists of information carried in protocol messages, and it can be either binary or quantitative in nature. Binary feedback is simple information only distinguishing congested from non-congested network state; this information can be carried in the IP header using the ECN bit which is set by intermediate congested routers, or using the ICMP source quench message. Quantitative feedback is more elaborated, and the intermediate nodes inform the sender about the bandwidth it may use. This information can be carried in the *Advertised Window* field of a TCP *Acknowledgment* message, or in the XCP header if XCP is used.

Feedback-based models seem to be easier to deploy at large scales, since the mechanisms it uses are distributed and they require less changes in the network elements than reservation-based architectures. Models based on explicit feedback however, require changes and additional processing on network elements which, in the case of quantitative explicit feedback, may be considerable.

III. Feedback-based congestion control mechanisms

The congestion control mechanisms described here are independent of the transport protocol service, e.g. reliable transmission and in-order delivery. For example, the *Datagram Congestion Control Protocol* (DCCP) which is intended for unreliable transmission may use both TCP-like or TFRC congestion control, or the *Stream Control Transmission Protocol* (SCTP) which enables multi-homing, uses roughly the same congestion control mechanism as TCP, therefore SCTP exhibits a behaviour similar to TCP.

1) TCP AIMD Congestion Control: TCP uses the *Additive Increase Multiplicative Decrease* (AIMD) mechanism to regulate the size of the *cwnd*. After probing the network for available resources, where *cwnd* growth is exponential (*slow-start*), the *cwnd* value is increased by one segment per RTT until a sign of congestion is detected; in this case the *cwnd* size is cut in half. This produces a rather unstable throughput which is unsuitable to transport interactive real-time flows.

2) TCP-Friendly Rate Control: TFRC [6] is an *equation-based* congestion control mechanism. It uses the TCP throughput equation [7] to define the sending rate:

$$X = \frac{s}{RTT\sqrt{2b\frac{p}{3}} + (T0(3\sqrt{3b\frac{p}{8}}p(1+32p^2)))} \quad (1)$$

The receiver periodically measures the loss probability *p*, estimates the *RTT* value, and reports them back to the sender;

s is the packet size, b is the number of packets acknowledged by each *Acknowledgement* (ACK) and *T0* is the TCP retransmission timeout . The p value is estimated by detecting lost or ECN marked packets using the sequence number or the ECN bit of the IP header of each packet. Based on the p and RTT values received, the sender calculates the average rate X of a TCP flow in equivalent conditions using Eq. 1, and adjusts the sending inter-packet interval to match this rate, which results in a stable throughput. This mechanism implies a slower response to network changes because the estimation and report of the parameters p and *RTT* is not instantaneous.

3) Explicit Window Adaptation: EWA [8] is a congestion control mechanism used to explicitly inform senders about the maximum bandwidth they may use. Routers calculate this value periodically based on the current free queue capacity, and on the utilization value α which varies with the state of the router's queue. If the mean queue length is above a certain threshold, α is multiplicatively decreased; otherwise, it is additively increased. The bandwidth value calculated is then placed in TCP's *Acknowledgement* packets in the *Advertised Window* field. This value is read by the TCP sender which uses it as the upper bound to its *cwnd* value. This mechanism requires that the reverse and the forward paths are the same.

4) eXplicit Control Protocol: XCP [9] performs congestion control by introducing a new header, between the network and transport layers, with three fields which are used by end-nodes to inform routers about the flow characteristics, and by routers to inform (feedback) end-nodes how to adjust their *cwnd* or sending rate. Routers don't need to keep per-flow information since the flow throughput and RTT variables are contained in the XCP header of each packet. The XCP router periodically calculates the spare capacity by performing the difference between the number of bytes that traversed the router in that interval and the actual outgoing capacity. A fraction of the spare capacity will then be distributed equally between outgoing flows if the spare capacity is positive; if it is negative all the flows receive negative feedback proportionally to their rate. The fraction value of the spare capacity to be distributed is chosen in order to make the system stable and this value is advised to be around 0.4 in the XCP specification. XCP decouples the efficiency control algorithm (MIMD), which has a fast response to enable high utilization, from the fairness control algorithm (AIMD), which is known to converge to fairness but has a slower response. XCP also maintains small-sized queues in the routers, which reduce end-to-end delay, and its operation requires that three sums and three multiplications are performed in the routers by incoming packet and that at least three multiplications and one sum are performed per outgoing packet. One drawback of XCP, is that an XCP router needs to know the capacity of its outgoing links, which in the case of shared mediums like 802.11 is hard to determine as it varies with load.

IV. Definitions

Three parameters seem to be relevant when comparing congestion control mechanisms in 4G networks:

- *Dynamic Behaviour* - Characterizes the ability of the mechanism to make the flow adapt to new bandwidths. A mechanism providing an ideal *Dynamic Behaviour* would make a flow use its share of the available bandwidth at all times.
- *Media Friendliness* - Specifies how the mechanism affects the interactivity of a media application specially with respect to jitter. The ideal mechanism would deliver the packets to the receiver with the same inter-packet interval as they were created by the source. End-to-end delay is implicitly associated to this parameter since jitter is caused by end-to-end delay variation.
- *Scalability* - It describes the cost of deploying the new solution at a large scale taking into account the legacy Internet.

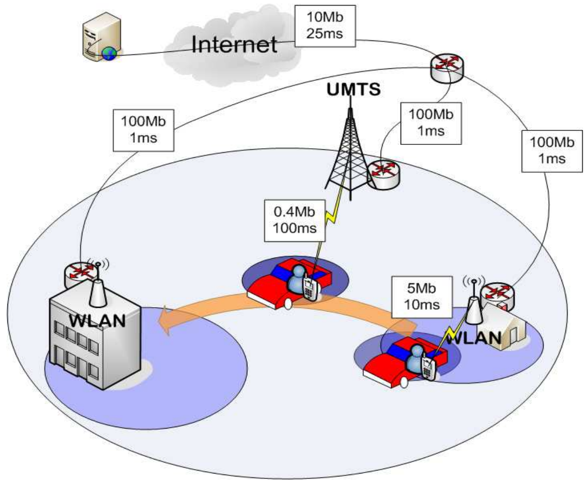


Fig. 2. Simulation Scenario.

V. Simulations and Results

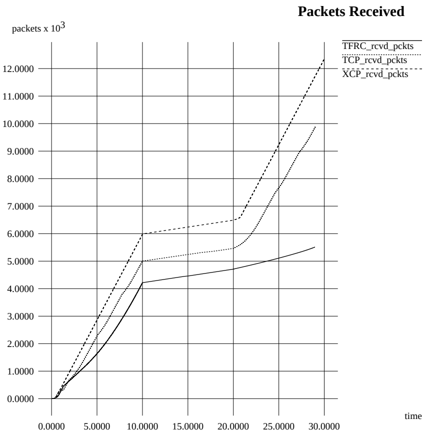
We evaluate through simulation the *Dynamic Behaviour* and *Media Friendliness* of the following congestion control mechanisms: TCP, TFRC and XCP. EWA was not simulated because there was no implementation available for ns-2 at the time.

The scenario adopted for the simulation is illustrated in Fig. 2. It represents an handover between WLAN and UMTS, which means doubling the end-to-end delay and cutting the rate to a tenth of the initial rate. The inverse handover is also simulated after the first handover. The WLAN-UMTS handover occurs at time 10 seconds, and the UMTS-WLAN handover occurs at time 20 seconds. We consider the handover to be ideal, that is, it is instantaneous and packets that were sent on the old link are duplicated in the new link.

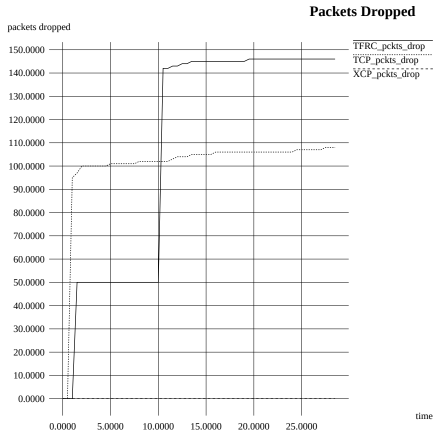
Sources are modeled as greedy although we are analyzing the problem of real-time flows, because this allows to have a better understanding of the mechanism response. Packet size is fixed at 1 kbytes, and routers have queues of 64 kbytes. All simulations were run on ns-2 and the TCP version used was Reno. New Reno was also simulated with no significant differences.

A. Dynamic Behaviour

In order to evaluate the *Dynamic Behaviour* of TCP, TFRC and XCP, we measure the number of packets dropped (Fig.

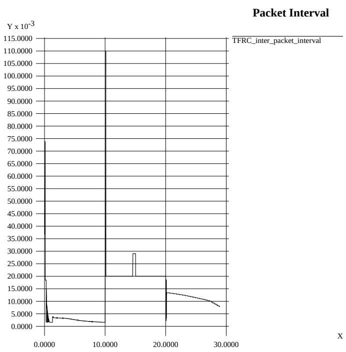


(a) Number of packets received by the receiver.

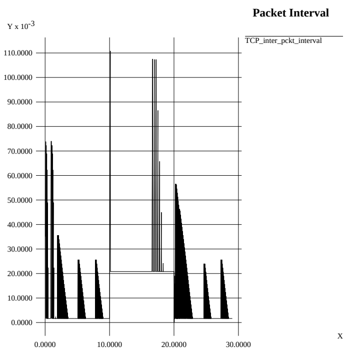


(b) Number of packets dropped at the bottleneck queue.

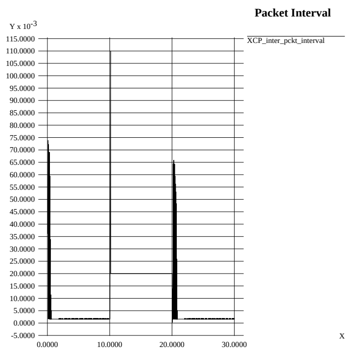
Fig. 3. *Dynamic Behaviour*



(a) Inter-packet interval at the receiver node using TFRC.



(b) Inter-packet interval at the receiver node using TCP



(c) Inter-packet interval at the receiver node using XCP

Fig. 4. *Media Friendliness*

3(b)) in the bottleneck queue, which is the queue associated with the UMTS link, and the number of packets received by the Mobile Terminal (Fig. 3(a)). The number of dropped packets gives information about the time the source took to adapt its rate, and the evolution of the sum of received packets helps to follow the sending rate dynamics.

We can observe in Fig. 3(b) that the TFRC source fails to quickly adapt its sending rate to the lower available bandwidth (t=10s) in the slower link (UMTS), which causes that approximately 90 packets are discarded - meaning 720 kilobits, almost 2 seconds of the UMTS channel transmission time . Both TCP and XCP succeed in quickly adapting their rate. Most TCP losses occur in the *slow start* phase, when TCP is probing the network for available capacity, while XCP which uses explicit quantitative feedback does not suffer any packet loss.

Fig. 3(a) shows the sum of received packets at the receiver node; again TFRC is slow adapting to new network conditions, now in the case of an handover from a slower to a faster link (t=20s). XCP and TCP take only a few RTT's to adapt. It is also noticeable that XCP's resultant rate is more accurate at using the path's full capacity.

B. *Media Friendliness*

In order to evaluate the *Media Friendliness* characteristic of TCP, TFRC and XCP, we measure the inter-packet interval at the receiver, which is also the inverse of the instantaneous rate. User-perceived quality of an interactive application suffers with the increase of end-to-end delay and also with delay variation (jitter) and it is important that these parameters are as low as possible. End-to-end delay depends on the load of the network and on the queue management policy more than on the congestion control mechanism, therefore it is not studied here. Jitter, however, depends not only on network stability, but also on the congestion control mechanism used to control the source. In Figures 4(a), 4(b) and 4(c), the inter-packet interval perceived by the (Mobile Terminal) is shown, using TFRC, TCP and XCP respectively. Jitter is the difference between consecutive inter-packet delays.

TABLE I

COMPARATIVE TABLE OF THE DIFFERENT MECHANISMS PRESENTED IN TERMS OF *Media Friendliness*, *Scalability* AND *Dynamic Behaviour*.

	Media Friendliness	*Scalability*	*Dynamic Behaviour*
Intserv	very high	very low	low
Diffserv	very high	low	low
SWAN	very high	low	medium
XCP	very high	low	very high
EWA	medium	medium	high
TFRC	high	medium	low
TCP	low	high	high

TCP inter-packet delay 4(b) is unsuitable for real-time interactive media, since consecutive interval values may vary in a range of 5 which results in a high jitter value, and this occurs in the optimal situation of a channel being used by a single flow. On the other hand, both XCP and TFRC produce constant or slowly varying inter-packet intervals which results in a jitter near 0, except for handovers (t=10s, and t=20s), as shown.

VI. FUTURE DIRECTIONS

None of the approaches studied solves efficiently the problem of supporting interactive real-time applications in dynamic heterogeneous networks. Reservation-based models seem to be costly to deploy as they have low *Scalability*. Over-provisioning is, from this point of view, much easier to achieve. On the other hand, there is no single feedback-based approach which requires changes only in end-nodes, and is capable of support interactive real-time flows efficiently, that is without degrading the user-perceived quality of the communication.

XCP is highly *Media Friendly* and also has a good *Dynamic Behaviour*; however the changes and capabilities it requires in all network elements make it have low *Scalability*. Also, XCP needs further research and testing in order to be ready for shared access mediums where the available capacity is not deterministic.

TCP is the best solution, from the *Scalability* point of view, not only because it is a simple mechanism, which only runs on end-nodes, but mainly because it is already widely deployed and has proved in the past to maintain the stability of the Internet. TCP also has a good *Dynamic Behaviour* succeeding in quickly adapting to abrupt changes in the network; however TCP is not *Media Friendly* because its congestion control mechanisms produce an unstable rate with high jitter.

TFRC is exactly the opposite of TCP; it is *Media Friendly* because it produces a stable rate but it has a poor *Dynamic Behaviour* failing to adapt rapidly to abrupt changes in the network. In terms of *Scalability* TFRC is a mechanism that only runs on end-nodes, however it is not as *scalable* as TCP because TCP is already widely deployed nowadays. It is clear that TFRC could benefit from cross-layer optimizations like the notice of link changes, which would reset the congestion state of the TFRC algorithm, resulting in a better *Dynamic Behaviour*.

EWA, we say, inherits the traditional behaviour of TCP. However, since routers explicitly inform senders about the maximum *cwnd* they may use, less losses should occur, resulting in a more stable rate of EWA flows and in a more *Media Friendly* mechanism.

Future research in this topic should focus on enhancing and further testing of the XCP protocol and other explicit-feedback models like EWA, and on optimizations to the TFRC mechanism in order to enhance its *Dynamic Behaviour*.

VII. CONCLUSION

In this article we described the problems of supporting interactive real-time traffic in dynamic heterogeneous networks such as 4G networks. We compared the reservation-based to the feedback-based approach, and explained why interest in feedback-based approaches has been renewed. The most important feedback-based solutions were identified and we concluded that the ideal congestion control mechanism would have the *Dynamic Behaviour* of XCP or TCP, the *Media Friendliness* of XCP or TFRC and the *Scalability* of TCP or TFRC.

REFERENCES

[1] S. Floyd and K. Fall, Promoting the Use of End-to-End Congestion Control in the Internet, *Networking, IEEE/ACM Transactions on*, August 1999.

[2] R. Braden, D. Clark and S. Shenker, Integrated Services in the Internet Architecture: an overview, RFC 1633, June 1994.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475, December 1998.

[4] G. Ahn, A. Campbell, A. Veras, and L. Sun, Supporting Service Differentiation for Real-time and Best-effort Traffi c in Stateless Wireless Ad Hoc Networks, *Mobile Computing, IEEE Transactions on*, 1(3), July - Sept. 2002.

[5] M. Allman, V. Paxson and W. Stevens, TCP Congestion Control, RFC 2581, April 1999.

[6] M. Handley, S. Floyd, J. Padhye and J. Widmer, TCP Friendly Rate Control (TFRC): Protocol Specifi cation, RFC 3448, January 2003.

[7] J. Padhye, V. Firoiu, D.F. Towsley, and J.F. Kurose, Modeling TCP Reno Performance: a Simple Model and its Empirical Validation, *Networking, IEEE/ACM Transactions on*, 8(2):133-145, April 2000.

[8] L. Kalampoukas, A. Varma, and K.K. Ramakrishnan. Explicit Window Adaptation: a Method to Enhance TCP Performance, *Networking, IEEE/ACM Transactions on*, 10(3):338-350, June 2002.

[9] Aaron Falk, Dina Katabi, Specifi cation for the Explicit Control Protocol (XCP), draft-falk-xcp-00.txt (work in progress), October 2004.