Improving Quality of Service from TCP/IP Performance Degradation

Dr.R.Seshadri,

Prof.& Director of university computer center
S.V.University, Tirupati-517502,A.P.

Prof.N..Penchalaiah

Dept of CSE ASCET, Gudur-524101,A.P.

***Abstract*-TCP is currently the dominate congestion control protocol for the Internet. However, as the Internet evolves into a high-speed wired-cum-wireless hybrid network, performance degradation problems of TCP have appeared, such as underutilizing high-speed links, regarding wireless loss as congestion signal, and unfairness among flows with different RTTs. In order to improve the quality of service for such high-speed hybrid networks, we propose a router-assisted congestion control protocol called Quick Flow Control Protocol (QFCP). The convergence of many traditional services over IP-based infrastructures drastically increases the amount of IP data traffic to be delivered to user clients, thus raising questions about the management of quality of service in such networks. Quality of service will be of primary importance in order to ensure right operation, and to face the occurrence of congestion conditions, due to bandwidth demanding multimedia services. in this paper, shows that QFCP can significantly shorten flow completion time, fairly allocate bandwidth resource, and be robust to non-congestion related loss. Also we consider a possible scenarios in which multiple multimedia and control streams are conveyed over the same HAN, and study a possible solution for the implementation of an easily manageable QoS framework, that relies on a QoS router based on open source software.**

1. INTRODUCTION

TCP [1] has been the dominate congestion control protocol for the Internet since 1980's. However, it also demonstrates some performance degradation in nowadays high-speed wired-cum-wireless networks. First, TCP's Additive Increase Multiplicative Decrease (AIMD) algorithm is too conservative for high-speed or long-delay links. After experiencing a packet loss, TCP needs to take many Round-Trip Times (RTTs) to recover the high throughput as it only increases the sending window by one packet per RTT. Flows often need more time to finish than expected by the users and large capacity of the bandwidth is wasted. Second, TCP assumes any packet loss as congestion signal, but it cannot distinguish non-congestion-related loss (transmission bit error) from congestion-related loss(router buffer overflow) leading to underutilization of wireless link. Third, TCP cannot fairly allocate bandwidth resource among competing flows with different RTTs, or among uploading and downloading flows in IEEE 802.11 Wireless LAN. As more and more high-speed (e.g., optical fibers), long delay (e.g., satellite links, trans-ocean cables), and wireless (e.g., WLAN, CDMA) links will be employed in the Internet, this situation will continue and may even be worse. Users will complain for the poor quality of service though they have paid for the costly network equipments or services. If we want to improve the Quality of Service (QoS) for congestion control in high-speed wired-cum-wireless networks, we must design an advanced mechanism to utilize the network resource more efficiently and more fairly. There are many QoS criteria regarding to network communication such as delay jitter, drop rate, priority service and so on. Here we will focus on three aspects that the Internet users may be most interested in, i.e., flow completion time, fair bandwidth allocation, and robustness to wireless loss.

The definition of "Home Area Network" refers to a network contained within a user's home and today translates into an IPbased network that covers the whole house and conveys all kinds of users services. Modern houses are provided with digital control systems for functional services, like household appliances, lighting and surveillance. Moreover, digital entertainment contents are delivered in each single room, as, for example, digital radio and television services, movies, music and others. In addition, Internet services are becoming widespread, so broadband connectivity is now an ubiquitous requirement. This allows to convey many traditional services over IP-based networks, with favorable cost to benefit ratio [3], [4].

In this paper, we consider IP-based HANs in which entertainment and home services must coexist, in order to provide all users with unique access stations both for their entertainment, for the delivery of home data (such as video surveillance streams) and for the control of domestic facilities. We adopt two possible frameworks for this kind of networks: a decentralized framework with independent servers, and a centralized framework with a unique server platform. Both solutions are based on common hardware and open source software. Quality of service relies on an open source QoS router, able to differentiate all the services and to impose proper rules for their associated priority and bandwidth.

II. QFCP DESIGN

Quick Flow Control Protocol (QFCP) [2] is a router assisted congestion control protocol for high-speed wiredcum-wireless networks. There are some other router-assisted protocols such as Quick-Start [5], XCP [6], and RCP [7]. The common design incentive is that routers are the places where congestion happens and with explicit feedback from routers we should be able to utilize the network resource more efficiently. Unlike XCP, QFCP gives per-flow feedback on flow rate instead of per-packet feedback on window adjustment. There are three fields in the QFCP header of each packet: RTT, rate request, and rate-feedback. We use a similar framework of Quick-Start but we extend the rate-request-and-grant mechanism to the whole lifetime of a flow: (1) The sender sets the initial value of rate-request field in the header of each outgoing packet to be the desired sending rate of this sender;(2) When the packet reaches a router, the router compares the value in rate-request field with the router's own fair-share rate and puts the smaller one back into that field; (3) On receiving the packet, the receiver copies the rate-request field into the rate-feedback field of the corresponding ACK packet and sends it back to the sender; (4) When the sender receives the ACK packet, it reads the value in the rate-feedback field and adjusts its sending rate accordingly.

A router maintains a fair-share rate R for each output interface. This rate R is the maximum rate allowed for flows going through this interface during the current control interval T. T is set to be a moving average of RTTs of seen packets. At the beginning of every control interval the QFCP controller estimates the number of flows traversing this interface as:

$$N(t) = \frac{y(t)}{R(t-T)}, \qquad (1)$$

where y is the input traffic rate measured in the last interval T, and R(t-T) is the flow rate feedback given in the last control interval. Then the controller updates its fair-share rate R as:

$$R(t) = \frac{C - \beta \cdot \frac{q(t)}{T}}{N(t)}, \qquad (2)$$

where C is the capacity of the output link, q is the minimum queue length observed in the last control period T, and β is a constant of 0.5. When a packet arrives at a router, the controller compares the value in the rate-request field with its own fair-share rate R and copies the smaller value back into that field. This rate-request field will eventually be copied into the rate-feedback field of the corresponding ACK packet and sent back to the sender by the receiver. On receiving an ACK,the sender reads the feedback and adjusts its congestion window as:

cwnd = max(feedback . RTT ,MSS), (3)

where feedback is the routers' feedback on flow rate, RTT is the round-trip time measured by the sender, and MSS is the maximum segment size. Thus, flows can send data at the highest rate allowed by all routers along the path, while routers periodically update the fair-share rate based on flow number estimation. Due to the inability to set the exact capacity of a wireless link, we need to design an adaptive algorithm that can finds and sets this capacity parameter by itself. We observe that the output traffic rate can be used to estimate the link capacity for an active network interface and we add the following formula in QFCP for link bandwidth probing:

$$C = \begin{cases} output, & if \quad q \geq 1 \\ (1+\alpha)\cdot C, & else \end{cases}, \qquad (4)$$

where q is the minimum queue length in packets observed in the last control interval, output is the output traffic rate, and α is a constant of 0.1. The basic idea is as following:

1) If the minimum queue length q is greater than or equal to one packet, which means the output interface is busy and keeps sending data in the last control interval, then the output traffic rate can be a good estimation of the current link capacity.

2) If the minimum queue length is less than one packet, which means the output link is sometimes idle and underutilized during the last control interval, we can try to multiplicatively increase the link capacity estimation by a factor (1+α) and wait a control interval to see whether the queue is going to build up.

For a sender in lossy wireless environment, it had better differentiate two kinds of packet loss: for non-congestion related loss (bit error), it should maintain the current window size; and for congestion-related loss (buffer overflow), it should slow down to prevent congestion collapse. Unfortunately, currently router-assisted congestion control protocols cannot do such differentiation yet. For example, XCP simply inherits the standard TCP behavior when encountering packet loss [6]. That is, on receiving three duplicate ACKs, the congestion window cwnd is halved; and on retransmit timeout, cwnd is set to one. The assumption is that packet loss may reveal a congested non-XCP router in the path and transiting to standard TCP behavior is a conservative response. However, if we are sure that all routers along the path support router-assisted congestion control, such slowdown reaction should be unnecessary for packet loss caused by bit error. For TCP, the sender has to slow down on detecting packet loss because packet loss is the congestion signal for TCP. This is due to the design rationale of TCP congestion control: a TCP flow keeps increasing its sending rate and intentionally fills up the buffer of the bottleneck router to generate packets drops; through this approach TCP finds the available capacity of the path. But for router-assisted approach, since congestion information has already been wrapped in the special packet header and communicated to the sender, the sender should not insist treating packet loss as congestion signal now. Instead, it should use the information in the congestion header to adjust its congestion window. For example, in QFCP, if the loss is congestion-related, the rate feedback in subsequent ACK (or dup-ACK) will tell the sender to slow down; but if it is no congestion-related loss, the subsequent rate feedback will probably be similar to the current sending rate of this flow.

We suggest that separate the data reliability control from congestion control when receiving duplicate ACKs. When the sender receives a duplicate ACK, it suggests that a data packet has successfully reached the receiver but its sequence number is greater than that expected by the receiver. Thus, for data reliability control, upon reception of 3 duplicate ACKs, the sender should retransmit the packet with the expected sequence number. While for congestion control, when a QFCP sender receives a duplicate ACK, it adjusts the congestion
window to:

$$cwnd = feedback .RTT + num\_dupACK \qquad (5)$$

where feedback is the rate feedback from routers, RTT is the sender's estimation of round-trip time, num_dupACK is the number of duplicate ACKs received. The inherent idea is that the sender temporarily keeps the successfully-transferred but not-in-order packets in buffer and opens the congestion window so that it can continue sending data at the router allowed rate. The counter num_dupACK is reset to zero when a new ACK packet arrives and cumulatively acknowledges all data packets sent before the detection of the loss. Note that we do not address complicate situations such as loss of the retransmission packet here and leave them for future study.XCP is a little different from QFCP. QFCP directly uses the fair-share flow rate as the feedback and this rate is not changed during the current control interval. The rate feedback information in any single ACK is sufficient for us to compute the target window size. But for XCP, we may not be able to compute the correct window size base on the feedback when encountering loss. Because in XCP, each ACK carries unique per-packet feedback information on window adjustment and the information carried on lost packets may not be negligible. Any packet loss will cause mismatching between the actual window size of the sender and the target window size expected by the routers. XCP-r [8] suggests computing the congestion window size at the receiver side and sending the value back to the sender through ACK packets. This modification on XCP only deals with ACK loss but packet loss on the forward path may still cause the window mismatching problem.

Another possible solution is to keep the window unchanged on non-congestion loss and halving the window on congestion loss. But firstly we need to distinguish the two kinds of loss in XCP. Intuitively we may say if the feedback is positive, it is non-congestion-related loss; and if the feedback is negative, it is congestion-related loss. However, the feedback is also used for fairness control. A negative feedback may possibly only want to change the flow's rate toward fair-share rate and may not necessarily suggest congestion. Halving the cwnd or change cwnd to 1 is too aggressive for this case. But if the loss is congestion related, window adjusting only based on feedback may be not enough since some feedback on window reduction may be lost. In sum, unlike QFCP, it is not so easy for XCP to differentiate the two kinds of packet loss based on feedback information.

While for packet loss event triggered by retransmit timeout, since no feedback information available at this instant and the loss may be caused by severe congestion, conservatively set congestion window to one should be better. And if this is not a congestion loss, any subsequent ACK will recover the congestion window to the proper size in QFCP.

In addition, if a router drops packets due to buffer overflow, it should also sum up the number of dropped packets and use the virtual queue length when running the control algorithm. That is, substitute q in the algorithm with:

$$virtual\_q = q + num\_drop . \qquad (6)$$

Thus, if packets are dropped by routers, the feedback computed using the virtual queue length can still precisely reflect the congestion condition.

III. HAN QOS FRAMEWORK

As an example, we have considered the case of a home network based on a home server platform, able to concentrate all the user's services in a unique hardware/software solution. For this purpose, we have adopted the LinuxMCE [9] client/server software platform, that allows the management of all home entertainment and automation services. LinuxMCE integrates a media and entertainment system for music, movies, and TV, a home automation system to control lighting and household appliances, a phone system with video conferencing, a security system with video surveillance and a home PC solution. A first difference with respect to the previous implementation is in the fact that LinuxMCE, that rely on Myth TV [10] for the management of digital television services, does not transmit directly the received DVB-T transport stream. On the contrary, digital television contents are saved in local buffers, thus allowing the implementation of a personal video recorder (PVR) service, and then retransmitted on uncast links. Moreover, no real time protocols are adopted for retransmission; instead, TCP connections are used for point-to-point streaming. Differently from RTP, TCP implements congestion control, so we want to verify the need and effectiveness of QoS management also in this different scenario. This implementation of the QoS framework is depicted in Fig. 1.

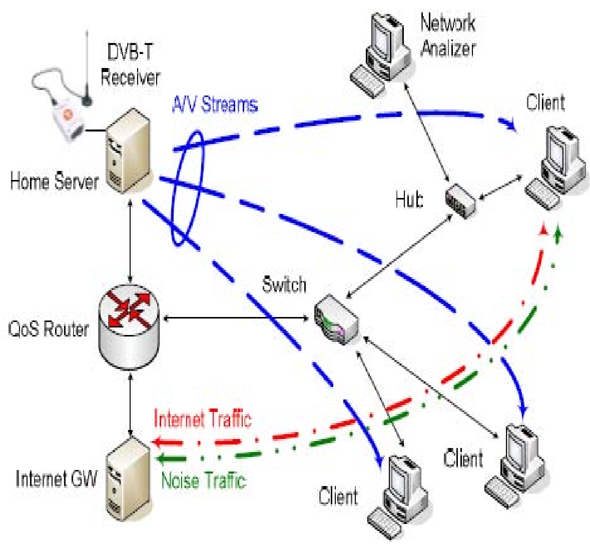

Fig. 1`: A QoS Framework for HANs.

As shown in the figure, we have considered the case of a unique home server, that exposes multiple services, an Internet gateway that ensures broadband data connectivity and multiple clients. We have focused on one of such clients, and analyzed its ingoing traffic through the network analysis software Wire shark [11]. Another relevant point is the uplink port of the layer 2 switch, that can be subject to congestion when the number of client increases. In this case, we have used the traffic analysis tools provided by the QoS router software in order to monitor such link.

A. Hardware and Software Infrastructure

This new setup adopts the following components:

- QoS router: Pentium III 800 MHz with 256 MB RAM and Zeroshell Linux operating system.
- Home server: Pentium IV 2.8 GHz with 512 RAM and Linux operating system with LinuxMCE.
- Internet gateway: Intel Mobile 1.2 GHz with 512 RAM and Windows XP operating system.
- Clients: Pentium processors with at least 512 MB RAM and suitable software clients.

In order to simulate a high number of clients, we have reduced the total bandwidth for the uplink port of the layer 2 switch to 10 Mbps. This way, our results also apply for the case of a more common 100 Mbps uplink bit rate, in association with up to 30 clients. Also in this case we have simulated different congestion conditions by means of an additional “noise” traffic, generated through Iperf .

B. Quality of Service

1) Measured Parameters

The new system model adopts TCP instead of RTP traffic for audio/video streaming, therefore we have adopted a different approach to measure the quality of service. Differently from the previous case, no real time protocols are implemented and, instead, TCP connections are exploited; therefore, QoS is mostly related to the allocated bandwidth. A first set of measures have been done on the ingoing traffic for the considered client. For the sake of brevity, we focus on the IP bandwidth for the audio/video stream. Other measures have been done on the layer 2 switch uplink port, through evaluating its total allocated bandwidth, and the bandwidth allocated to each service.

2) QoS Rules

Table reports the QoS traffic classes and rules for this second case. The new system has been scaled differently from the previous one to simulate an higher number of clients; therefore, the bandwidth values have been scaled accordingly.

QOS Rules For Implementation

Class	Priority	Maximum Bandwidth	Guaranteed Bandwidth
noise	high	variable	variable
home automation	high	500 kbps	500 kbps
Internet traffic	low	1 Mbps	56 kbps
DVB-T service	medium	7 Mbps	variable

Differently from the previous case, we have now varied the guaranteed bandwidth allocated to the DVB-T service, in order to verify whether, in the case of network congestion, such traffic can be effectively protected by means of the QoS router.

C. Test Results

We have verified that, in absence of additional traffic, the services we have simulated (that are 3 audio/video streams for the DVB-T service and a broadband Internet data stream) require 8.2 Mbps total bandwidth that is 82% of the available bandwidth. Then, we have added a “noise” traffic with variable bit rate (namely 800, 1200, 1800 and 2300 kbps) in order to simulate different congestion conditions (that can be due to other network services) and verify the effectiveness of the QoS rules set for the uplink port of the layer 2 switch. In the presence of all the considered streams, in association with the QoS rules reported in Table II, we have obtained results that confirm the effectiveness of the traffic shaping function of the QoS router. They are shown in Fig. 2.

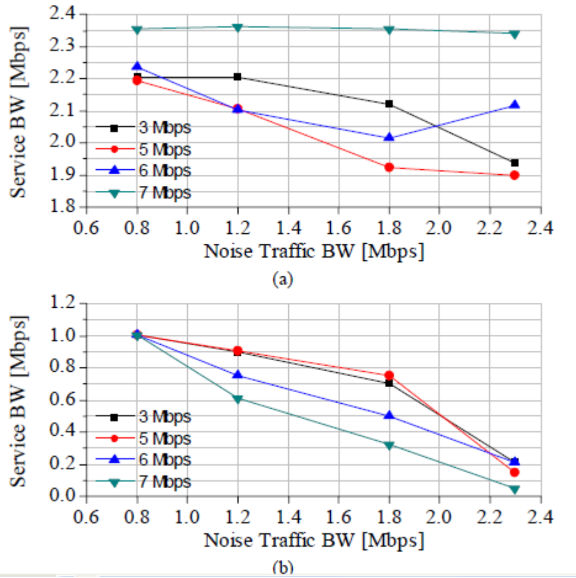

Fig. 2: Measured bandwidth for the DVB-T service (a) and for the Internet service (b), as a function of the noise traffic bandwidth, for several values of the maximum guaranteed bandwidth for the DVB-T service.

The priority assigned to the digital TV service, with respect to the Internet service, reflects on its bandwidth in different congestion conditions. The total guaranteed bandwidth is allocated to the different (3, in our case) audio/video streams, in order to preserve their quality. In the case of congestion, however, it may happen that a TCP connection is released, as occurs for the 6 Mbps guaranteed bandwidth curve in Fig. 2 (a), that has a slope change. In such case, the remaining connections can exploit additional bandwidth, thus improving their quality. The Internet service, on the contrary, has low priority, so its allocated bandwidth is immediately reduced when congestion increases, with a substantial quality loss.

IV. CONCLUSION

Quality of Service has been studied extensively for real time multimedia flows. However, it is a little strange that the service quality of other common TCP flows (HTTP, FTP,email transfer, etc.) has seldom been studied. Here point out that TCP suffers performance degradation in high-speed wired-cum-wireless networks. In order to improve the QoS we propose a router-assisted congestion control protocol named Quick Flow Control Protocol (QFCP) and also implement QoS framework for the HAN. This scenario of home area networks is being dominated by IP-based networks intended for home entertainment and automation. Quality of service plays a fundamental role in such scenario, where right operation must be ensured and congestion properly faced.Finally,the QoS improved by above specified TCP/IP based scenarios.

REFERENCES

[1] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC2581 1999.
[2] J. Pu and M. Hamdi, "New Flow Control Paradigm for Next Generation
[3] O. Martikainen, “Internet Revolution in Telecom,” in Proc. IEEE JVA '06, Sofia, Bulgaria, Oct. 2006, pp. 58–62.
[4] W. D. Jones, “A Bronx tale: broadband TV concern hooks up the 'hood,” IEEE Spectrum, vol. 42, Sep. 2005 pp.:10–12.
[5] A. Jain, S. Floyd, M. Allman, and P. Sarolahti, "Quick-Start for TCP and IP," in IETF Internet-draft, work in progress, 2005.
[6] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for HighBandwidth-Delay Product Networks," in Proceedings of Proceedings of ACM SIGCOMM '02, 2002, pp. 89-102.
[7] N. Dukkipati, M. Kobayashi, R. Zhang-Shen, and N. McKeown,"Processor Sharing Flows in the Internet," in Proceedings of International Workshop on Quality of Service, 2005.
[8] D. M. Lopez-Pacheco and C. Pham, "Robust transport protocol for dynamic high-speed networks: enhancing the XCP approach," in Proceedings of 13th IEEE International Conference on Networks, Kuala Lumpur, Malaysia, 2005, pp. 404-409.
[9] http://linuxmce.com/
[10] http://www.mythtv.org/
[11] http://www.wireshark.org/

First A. **Dr.R.SESHADRI** was born on 18-12-1959. He has completed B.Tech(ECE),Nagarjuna University, Guntur District , Andhra Pradesh,1981; M.E.(Control Systems Engg) , P.S.G.College of Technology, Coimbatore, Tamil Nadu,1984; Ph.D.(Simulation, Modelling & Compression of E.C.G. Data Signals. (Data compression Techniquies)ECE.), Sri Venkateswara University, Tirupathi, Andhra Pradesh,1998

He is working as Professor & Director, University Computer Centre, Sri Venkateswara University, Tirupati. He has richest of knowledge in Research field, he is guiding 10 PhD in Fulltime as well as Part time. He has vast experience in teaching of 26 years. He published 10 national and international conferences and 8 papers published different Journals.

Prof.N.Penchalaiah Research Scholar in SV University, Tirupati and Working as Professor in CSE Dept,ASCET,Gudur. He was completed his M.Tech in Sathyabama University in 2006. He has 10 years of teaching experience. He guided PG & UG Projects. He published 5 Inter National journals and 2 national Conferences.