View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by CiteSeerX

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 253495, 5 pages
doi:10.1155/2010/253495

Research Article

On the QoS of IPTV and Its Effects on Home Networks

Dongyu Qiu

Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada H3G 1M8

Correspondence should be addressed to Dongyu Qiu, dongyu@ece.concordia.ca

Received 2 October 2009; Accepted 20 May 2010

Academic Editor: Georgios Gardikis

Copyright © 2010 Dongyu Qiu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In Internet Protocol Television (IPTV) systems, Quality-of-Service (QoS) is a critical factor for user satisfaction. In this paper, we first propose a queueing model for IPTV systems and discuss how to ensure the QoS of IPTV. We then use the model to analyze the effects of IPTV traffic on other applications in home networks. We find that TCP congestion control may not work well under this circumstance and propose a new approach to improve the performance of TCP. We also verify our results through *ns*2 simulations.

1. Introduction

In recent years, we have seen a tendency of various services converging to the ubiquitous Internet-Protocol- (IP-) based networks. Besides traditional Internet applications such as web browsing, email, file transferring, and so forth, new applications have been developed to replace old communication networks. For example, Voice over IP (VoIP) can be used as an alternative to traditional telephone network. There are also efforts to provide digital television service over IP networks. IPTV is one of the solutions. In the future, we expect a single network, the IP network, to provide services that have been carried by different networks today.

Transmitting video over IP networks is not a new idea. People have been interested in it since the very early stage of the Internet. However, there are major technical difficulties that prevent transmitting video over IP with satisfactory quality. The IP network is designed to be a best-effort network, which means that it does not provide any QoS guaranty. QoS, on the other hand, is essential to the quality of video. To get good quality, the network has to satisfy certain requirements on bandwidth, end-to-end delay, and jitter, and so forth. Another important issue is scalability. Many systems perform well when the number of users is small. However, when there are thousands of users accessing the service at the same time, which is typical in a broadcasting TV system, it will stress the servers.

With the advances of video encoding and the quick development of communication networks, transmitting video with at least VHS-quality has become possible in recent years. Digital video encoded with MPEG-1 around 1.5 Mbps can provide VHS-quality and MPEG-2 can offer High-Definition TV-quality video around 20 Mbps or higher. The next generation codecs, such as H.264 and VC1, can offer DVD- and HD- quality stream under 10 Mbps. At the same time, Internet access technology has been improved significantly. For example, ADSL can provide upto 8 Mbps download rate and the newest VDSL has up-to 52 Mbps download rate. Cable Modem can provide similar or even higher download bandwidth. With the recent deployment of Fiber to the premises (FTTP), we can expect higher Internet access bandwidth to be provided to home users. Hence, transmitting VHS-quality or even HDTV- quality video over the Internet becomes technically feasible.

In the current Internet, there are basically two solutions to deliver digital television service. The first one is Peer-to-Peer (P2P) video streaming. In [1], a P2P media streaming system called DONet has been proposed. Basically, all P2P video streaming systems work similarly. In such a system, the video is divided into many pieces. When a user obtains a piece, it will serve other peers by uploading this piece. Hence, each user serves as a client and a server at the same time. P2P systems have been proved to have very good scalability [2]. Hence, there is no need for expensive servers and high upload bandwidth to provide P2P video streaming. Due to the low setup cost and excellent scalability, P2P video streaming applications, such as PPLive, PPStream, and sopcast and so forth, have become very popular recently. However, since

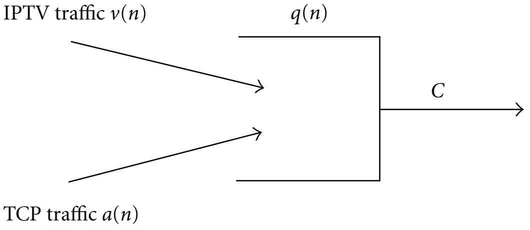


Figure 1: The queueing model.

P2P video streaming traffic may cross the whole public Internet, it is very hard to guarantee the QoS in such a system. There is normally little or no QoS management in P2P video streaming. IPTV, on the other hand, takes a different approach to solve the scalability and QoS issues. IPTV has been developed and deployed mainly by large telecommunication providers as a competitive replacement product for digital cable and satellite services. Hence, IPTV systems normally use a closed network infrastructure. In such a system, the IPTV service provider is also the Internet service provider. Since the video streaming traffic is transmitted in a closed network, the QoS management is much easier than that in a P2P system. IPTV uses the multicast technology to solve the scalability issue. When more than one user are watching the same TV channel, the service provider will multicast the video to the users. Hence, only one copy of the video is transmitted and the system has good scalability.

In this paper, we will focus on IPTV systems. The paper is organized as follows. In Section 2, we will propose a queueing model for IPTV systems and discuss how to ensure the QoS of IPTV in such systems. We will also discuss how this will affect other Internet applications in home networks. In Section 3, we will analyze the model and discuss how the system performance can be optimized. Simulation results using *ns*2 simulator will be shown in Section 4. Finally, we conclude this paper in Section 5.

2. QoS of IPTV

IPTV is a new technology to delivery digital television service over IP networks. In contrast to the popular P2P video streaming, IPTV systems are normally closed and proprietary. It has been deployed by major telecommunication providers to compete with traditional digital cable service. In an IPTV system, users subscribe to the IPTV service through their Internet service provider. The service provider sometimes also offers VoIP, service as an alternative to traditional telephone service. The combination of IPTV, VoIP, and Internet access is referred to as “Triple-Play” service. IPTV has some significant advantages over digital cable service. For example, it can make the TV viewing experience more interactive and personalized. IPTV also offers Video on Demand (VoD), in which a user can choose movies from a database and play it immediately. However, IPTV also has its own limitations. Since video quality is very sensitive to packet loss and delay, how to ensure the QoS in a best-effort network like an IP network is a challenge topic.

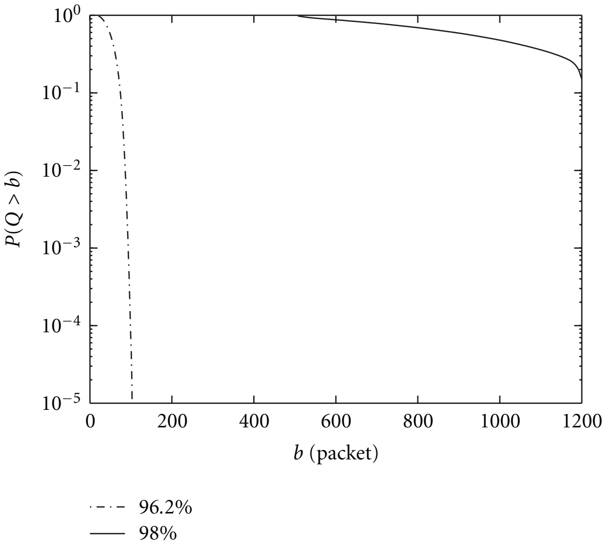


Figure 2: The effect of target utilization.

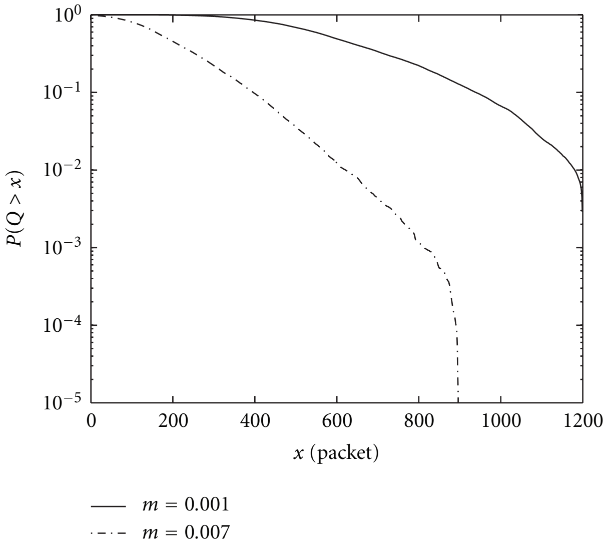


Figure 3: Queue distribution.

Internet is a distributed system and there is no centralized control of it. This makes the QoS management extremely hard. The video traffic may travel through some segments of the Internet, where the service provider may have no control at all. In such a scenario, it is very difficult to ensure QoS. Fortunately, in IPTV systems, QoS management is significantly simplified. IPTV is a closed network, in which the service provider not only controls the IPTV system, but also controls the Internet access of the users. In the current Internet, the bottleneck for most connections is normally at the user’s Internet access link, which is sometimes called the last mile. In the core of Internet, optical fibers have been deployed to provide tens of Gbps bandwidths and hence are unlikely to be the bottleneck. In an IPTV system, since the service provider also has control of the Internet access links, it can over provision the network to ensure the QoS of IPTV. For example, if the IPTV's bit rate is 3 Mbps, the service provider may over provision the capacity to ensure that the download bandwidth of the user is at least 5 Mbps, thus guaranteeing a minimum QoS level.

However, the simple over provision itself is not enough for IPTV QoS. Besides the IPTV application, there may be many other Internet applications running in the home network. For example, people may watch TV and browse the web at the same time. The traffic from other applications then will compete with the IPTV for the access bandwidth. If the total download rate exceeds the download bandwidth, packets will be dropped and hence it will degrade the video quality of IPTV. The solution to this problem is to give IPTV packets priority over other packets. This mechanism can be implemented in the network layer (e.g., Diffserv [3]) or in the MAC layer (e.g., IEEE802.1p). In either cases, when packets are competing for the output link, packets from IPTV flows will be processed with higher priority. Hence, although there may be other Internet applications, their traffic will not affect the QoS of IPTV. Combined with an appropriate over provision scheme, this mechanism can be used to ensure the QoS of IPTV and it is widely used in current IPTV systems.

Since IPTV traffic is given high priority, other Internet applications have to compete for the residual bandwidth and their performance may be affected. Different with IPTV, where UDP is used to carry the traffic, most other Internet applications such as HTTP, FTP, and SSH, and so forth, use TCP as the transport layer protocol. TCP itself has a built-in congestion control mechanism [4], which means that when TCP detects that the network is congested it will decrease its data rate. This kind of traffic is called elastic traffic and it can adapt to the available bandwidth in the network. Since many advanced video codecs produce variable-bit-rate (VBR) outputs, the available bandwidth for TCP will be time varying. Studying the impact of IPTV traffic on the performance of TCP is an interesting topic. Next, we use a simple queueing model to study it.

As we discussed before, the bottleneck normally happens at the Internet access link. Here, we use a queue to model this link that serves both IPTV and TCP traffics (Figure 1). In this model, C is the link capacity. For example, if the user's download bandwidth is 8 Mbps, then $C = 8$ Mbps. There are both IPTV traffic and TCP traffic on this link. Without loss of generality, we use a discrete time model. $v(n)$ is the data rate of IPTV traffic at time n and $a(n)$ is the data rate of TCP traffic. $q(n)$ is the queue length at time n. In the IPTV system, to ensure the QoS of IPTV, the data rate of IPTV traffic should not exceed the link capacity and hence $v(n) \le C$. The link capacity available for TCP traffic is then $C - v(n)$, which is normally time varying as we discussed before. The queue length can be expressed as

$$q(n) = \left[q(n-1) + a(n) + v(n) - C\right]^{+}, \qquad (1)$$

where $[x]^{+} = x$ if $x \ge 0$ and $[x]^{+} = 0$ if $x < 0$.

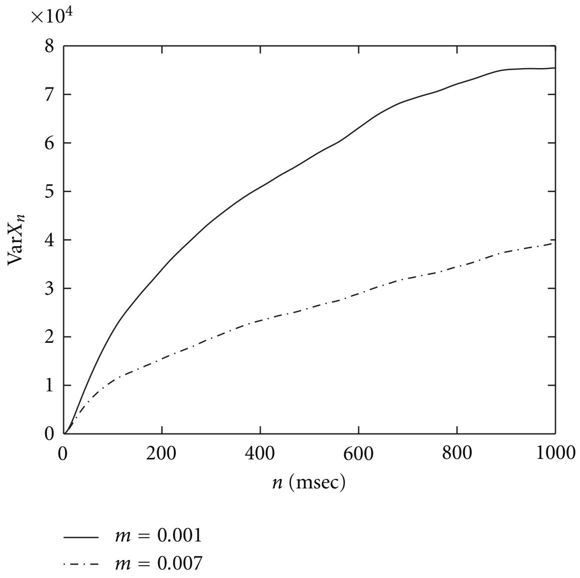


Figure 4: $\mathrm{Var}X_n$.

Note that $a(n)$ is the data rate of TCP traffic and hence it can be controlled by TCP. Ideally, we would like to have $a(n) = C - v(n)$ for all time n. If so, the link capacity is fully utilized and the queue length is always 0. However, this is impossible to achieve in real networks due to network delays, estimation errors, and so forth. Next, we discuss how to control $a(n)$ based on information about $v(n)$.

3. Effects of IPTV Traffic on TCP Performance

The objectives of TCP congestion control are high link utilization and small packet loss probability. However, these two objectives are generally conflicting. TCP congestion control always tries to fully utilize the link capacity. In Section 4, we will see that this may cause unnecessary workload to the queue and hence does not work well in practice. Motivated by this, we define $\rho < 1$ to be the target link utilization. Then $u(n) = \rho C - v(n)$ is the residual link capacity that we want TCP traffic to utilize. For the simplicity of analysis, we will first use an explicit-rate control model, in which we assume that we can explicitly set the TCP data rate $a(n)$. Later, we will discuss how the result can be applied to real TCP networks. We also assume that the $a(n)$ is controlled by a linear system, in which

$$\mathrm{a}(n) = L[u(n)]. \qquad (2)$$

A linear feedback control has been found to give good results when we have video traffic as uncontrollable traffic [5], which is exactly what we have in an IPTV system. Let $a(z)$ and $u(z)$ be the Z-transforms of $a(n)$ and $u(n)$, respectively. We have

$$a(z) = H(z)u(z), \qquad (3)$$

where $H(z)$ represents a linear time-invariant system. For example, if $a(n) = u(n-5)$, where 5 is the round trip delay of the TCP connection, then $H(z) = z^{-5}$. This system has been studied in [6] and we list the main results here.

In [6], it has been found that when $H(1) = 1$ the system has some desirable properties. First, the actual utilization of the system will be equal to the target utilization ρ. Secondly, the queue length distribution decays very fast. The tail probability of the queue length can be approximated by

$$\mathbb{P}\{Q > x\} \approx e^{-x^2/2D}, \tag{4}$$

where $D = \sup_n \operatorname{Var} X_n$ is a constant and

$$X_n = \sum_{j=-n+1}^{0} (a(j) - u(j) - (1-\rho)C) \tag{5}$$

is the net input to the queue over a given time period with length n. So, under a given target link utilization, the system performance can be optimized if we can minimize D. Next, we discuss how this can be done in a TCP network.

TCP has been widely used in the current Internet and its congestion control scheme has been shown to work well in many cases. However, it also has some drawbacks. For example, it has no provision for the detection of incipient congestion. When a queue overflows, packets are simply dropped. In past years, many Active Queue Management (AQM) schemes have been proposed [7–9] to improve the performance of TCP. In these schemes, the routers try to detect incipient congestion and mark packets accordingly to control the data rate of TCP flows. In this paper, we choose Random Exponential Marking (REM) [8] as an example to show how our result can be applied to real TCP networks.

In REM, the router maintains a price information, which is calculated based on the TCP data rate and the available bandwidth. The price at time n is given by

$$p(n) = p(n-1) + m(a(n) - u(n)), \tag{6}$$

where $m > 0$ is the step size and is the parameter that we need to choose. Note that here we slightly modified the REM algorithm by replacing the link capacity with the time-varying residual link capacity $u(n)$ to reflect the presence of IPTV traffic. This price information is then sent back to the TCP source through packet marking and TCP will adjust its data rate accordingly. Basically, when the price increases, the data rate will be decreased, and vice verse. This control is normally not linear. However, as an approximation, we can linearize the data rate control around its average value (see [10] for details on how the linearization can be done). Let τ be the round trip delay of the TCP flows. For the simplicity of analysis, we assume that all TCP flows have the same delay. Then the TCP data rate at time n can be expressed as

$$a(n) = -kp(n-\tau), \tag{7}$$

where $k > 0$ is a constant related to the utility function used by TCP. Note that the only approximation we made here is the linearization. The TCP data rate is not reacting instantaneously to the price due to the delay τ. In Z-domain, we then have

$$\begin{aligned} p(z) &= \frac{m}{1-z^{-1}}(a(z) - u(z)), \\ a(z) &= -kz^{-\tau} p(z). \end{aligned} \tag{8}$$

Solving these equations, we have

$$a(z) = \frac{kmz^{-\tau}}{1 - z^{-1} + kmz^{-\tau}} u(z), \tag{9}$$

and hence

$$H(z) = \frac{kmz^{-\tau}}{1 - z^{-1} + kmz^{-\tau}}. \tag{10}$$

We can easily verify that the system satisfies $H(1) = 1$ and hence has the desirable properties we discussed before. Note that in this system, since k is a constant related to TCP and cannot be changed, we need to choose an appropriate m to optimize the performance. Recall that the system performance is optimized when $D = \sup_n \operatorname{Var} X_n$ is minimized. Hence, the problem becomes one of choosing the optimal m such that D is minimized. It is very hard to find a closed-form relation between D and m. However, in a real network, once we obtain the stochastic property of $u(n)$ (which is equivalent to that of $v(n)$, the data rate of IPTV traffic), it is relatively easy to find the optimal m numerically. Next, we show how this can be done through *ns*2 simulations.

4. Simulation Results

We use the *ns*2 simulator to simulate the Internet access link of a home network. The link capacity (or the download bandwidth) of the link is $C = 200$ Mbps. Note that the link capacity we use here is higher than that of most home networks due to two reasons. First, different link capacities should give the similar results if we scale all system parameters accordingly [11]. Hence, the absolute value of C is not essential here. Secondly, when C is larger, we can see the performance difference more clearly. The round trip delays of all TCP flows are $\tau = 10$ msec. We use TCP-Reno and the modified REM (see Section 3) as the AQM scheme. All TCP packets have the length of 1000 bytes.

In the first simulation, the link serves 100 TCP flows and there is no IPTV traffic. The REM parameter $m = 0.007$. The target link utilization is set to be 98% and 96.2% respectively. We measure the actual link utilization and find that it is 96.2%, in both cases. Under the same actual link utilization, we compare their queue distributions in Figure 2. We can see that, when the target link utilization is set to 98%, the tail probability is much higher than the other one and hence will have much higher packet loss rate. This tells us that setting a very high target link utilization does not guarantee that the actual link utilization is high and may only cause unnecessary workload.

In the next simulation, we add IPTV traffic into the network. The mean rate of the IPTV traffic is 100 Mbps. The IPTV traffic is generated by using a Gaussian process and is carried by UDP packets. The target link utilization is set to be 96%. The REM parameter m is set to be 0.001 and 0.007, respectively, where $m = 0.001$ is chosen according to the guidelines in [10] and $m = 0.007$ is chosen by our algorithm to minimize D. Our simulation results are shown in Figures 3 and 4. From Figure 3, we can see that, with different REM parameters, the queue distribution can be quite different. In Figure 4, we show the corresponding $\mathrm{Var}\, X_n$. We can see that the smaller the $\mathrm{Var}\, X_n$ (in the case $m = 0.007$), the smaller the queue length. Hence, in a TCP network, minimizing $\mathrm{Var}\, X_n$ will be an effective way to control the loss rate.

5. Conclusion

In this paper, we first discuss how over provisioning and differentiated services can be used to ensure QoS of IPTV. We then use a queueing model to study the effects of IPTV traffic on home networks. Using this model, we analyze the performance of TCP when there is IPTV traffic in the network. Based on the analysis, we give some guidelines on how the system performance can be optimized. We then use REM as an example to show how the results can be applied to real TCP networks. We also verify our results by *ns*2 simulations.

References

[1] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, Miami, Fla, USA, March 2005.

[2] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '04)*, pp. 367–377, August 2004.

[3] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '99)*, pp. 81–94, 1999, http://citeseer.ist.psu.edu/article/stoica99providing.html.

[4] V. Jacobson, "Congestion avoidance and control," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '88)*, pp. 314–329, 1988.

[5] Y. Zhao, S. Q. Li, and S. Sigarto, "A linear dynamic model for design of stable explicit-rate ABR control schemes," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97)*, pp. 283–292, April 1997.

[6] D. Qiu and N. B. Shroff, "Queueing properties of feedback flow control systems," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 57–68, 2005.

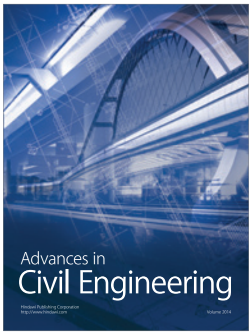
[7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

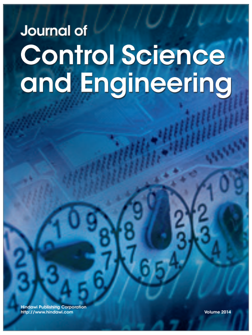
[8] S. Athuraliya and S. H. Low, "Optimization Flow Control II: Implementation," Internal report, Melbourne University, 2000.

[9] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '01)*, San Diego, Calif, USA, August 2001.

[10] F. Paganini, J. Doyle, and S. Low, "Scalable laws for stable network congestion control," in *Proceedings of the 40th IEEE Conference on Decision and Control (CDC '01)*, vol. 1, pp. 185–190, December 2001.

[11] P. Tinnakornsrisuphap and A. M. Makowski, "Limit behavior of ECN/RED gateways under a large number of TCP flows," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 873–883, April 2003.





Hindawi

Submit your manuscripts at
http://www.hindawi.com