End to End Adaptation for the Web: Matching Content to Client Connections

Kristoffer Getchell, Martin Bateman, Colin Allison, and Alan Miller

School of Computer Science, The University of St Andrews, St Andrews, Fife, Scotland
{kg,mb,ca,alan}@dcs.st-andrews.ac.uk

Abstract. The size and heterogeneity of the Internet means that the bandwidth available for a particular download may range from many megabits per second to a few kilobits. Yet Web Servers today provide a one size fits all service and consequently the delay experienced by users accessing the same Web Page may range from a few milliseconds to minutes. This paper presents a framework for making Web Servers aware of the Quality of Service that is likely to be available for a user session, by utilizing measurements of past traffic conditions. The Web Server adapts the fidelity of content delivered to users in order to control the delay experienced and thereby optimize the browsing experience. Where high bandwidth connectivity and low congestion exist high fidelity content will be delivered, where the connectivity is low bandwidth or the path congested lower fidelity content will be served and delay controlled.

Keywords: Internet, monitoring, quality of service, adaptation, media, perception.

1 Introduction

Access bandwidths now range from a few kilobits per second through a mobile phone up to gigabits per second through Local Area Networks. Despite the bandwidth at the core of the Internet increasing, many bottlenecks continue to exist within the system; even users with high bandwidth connectivity may experience significant delays when engaging in network communication. At present the Internet largely relies upon the congestion control mechanisms embedded in TCP (Jacobson 1988) to prevent congestion collapse (Jain and Ramakrishnan 1988), which would render it unusable, yet applications are in a strong position to know how best to react to the onset of congestion (Floyd, Jacobson et al. 1997). If an application was made congestion aware it could directly control the amount of data sent. If congestion was high it could reduce the absolute number of bytes transmitted as well as the rate at which they were sent.

This paper addresses two questions: How is it possible for an application to become aware of network conditions and, given this awareness, how can a system be designed to allow application led adaptation to occur. A framework, which consists of three components, is proposed. A Network Monitor provides a Server with measurements of the Quality of Service (QoS) that the network is providing. A Network Aware Web Server handles the dynamic decisions required in order to

J. Filipe et al. (Eds.): ICETE 2005, CCIS 3, pp. 78–90, 2007.

determine the optimal version of a site to send to a connecting client and a Content Adaptation Tool allows content providers to generate, from a single high quality site, versions that are appropriate for different connection types.

From the user perspective a number of factors contribute to the perceived QoS offered, with both the speed of presentation and the quality of resources being important. A fast loading, simple Web Site is not necessarily going to be considered to be high quality by all users, and neither is a slow loading, highly detailed one (Bouch and Sasse 1999; Bouch, Kuchinsky et al. 2000; Bouch and Sasse 2000). From the content provider's perspective, a trade-off must be met whereby a balance between the speed at which resources can be provided to the target audience and the quality of the resources provided is achieved. Currently this trade-off is managed offline with content providers producing Web Sites which will be viewable within reasonable periods of time by the majority of Internet users. Those Internet users with connections slower than the speeds accounted for during the development of a site will be left with a poor browsing experience, whilst those with faster connections could have been supplied with pages of a higher fidelity. As the diversity of connection technologies continues to expand, this disparity is set to grow yet further.

By considering the QoS offered to the browsing client by the network, an adaptation framework would afford content providers the ability to focus on producing the highest quality resources, without having to consider the limitations that slow network links may cause. Lower fidelity resources can then be generated automatically using adaptation tools.

This paper continues with a discussion of related work in section 2 which leads to a discussion of the QoS issues of relevance to this work. Section 4 describes the framework used to allow the Network Aware Server to modify its behaviour as described in section 5. An evaluation of the framework thus far is provided in section 6, with section 7 concluding and drawing the paper to a close.

2 Related Work

Numerous studies have advocated the adaptation of content delivery in response to the load placed upon a Web Server (Barnett 2002; Pradhan and Claypool 2002). A similar approach to content adaptation as advocated in (Barnett 2002; Pradhan and Claypool 2002) is adopted by our work. However unlike (Barnett 2002; Pradhan and Claypool 2002) we address the case where the total load on the server is manageable but specific users are receiving a poor service because of network congestion or connectivity issues.

With the expansion of the e-commerce sector, the ways in which people interact with online vendors is of great importance to business sectors. (Bhatti, Bouch et al. 2000) discuss the issues surrounding user tolerance, with regards to the levels of service offered, reiterating the importance of user perceptions. Through experiment-tation the level of user satisfaction is tested, with the results showing that if a user is provided with some feedback relatively quickly, they are generally satisfied to wait for extended periods while the remainder of their request is fulfilled.

Taking the concept of adaptation to mobile devices, (Abdelzaber and Bhatti 1999; Shaha, Desai et al. 2001) discuss the issues surrounding the need to augment our traditional understanding of QoS in order to make the best use of this new wave of access device. Existing services designed for PC delivery are not well suited to PDAs. Owing to the poor quality of PDA delivered content it is hardly surprising that the mobile device users are often left frustrated and alienated. Media adaptation techniques, however, may help by allowing mobile users to more readily gain access to the QoS adapted versions of a particular resource.

3 Quality of Service Issues

In this section the characteristics of QoS as perceived by users and as observed on the network are discussed, along with the ways in which an application may be able to adapt media. The aim of this work is to provide mechanisms that can maximise the QoS perceived by users for a given set of network conditions. In order to do so it is necessary first to establish metrics for both the user perceived and network observed QoS.

3.1 User Perceived Quality of Service

There is no single mapping between network level metrics and the way in which a user perceives the QoS being offered. Here we identify three factors that contribute to whether a user perceives a Web Browsing session as satisfying or not.

1. **Fidelity:** The quality of data sent to clients is vitally important – if the technical quality is too low (i.e. videos are blurry, sound is skewed, images are fuzzy and too small etc) then the user will perceive poor resource quality. However if the technical quality is too high (i.e. videos and sounds are jumpy because there are delays in getting data to the client, images are too large to fit on screen etc) then the user will again perceive poor quality.
2. **Delay and Feedback:** When a user is interacting with a Web Site, they require feedback for their actions. If it takes 20 minutes to load each page on a site, then the length of time required in order to supply feedback to the user (i.e. the next page they asked for) would be considered excessive and so user-perceived quality will be deemed as low. However if the pages load too quickly, then it is possible that the user will not recognise the true speeds – delays of anything under 30ms are almost unnoticed by users and as such resources should not be wasted trying to reduce delays below 50ms or so (Abdelzaber and Bhatti 1999).
3. **Consistency:** If there is consistency of presentation within a session the user will perceive a higher QoS during their browsing session (Bouch and Sasse 2000).

By controlling the fidelity of data it is possible to control the delay that users experience. Thus for a known set of network conditions it is possible to trade off fidelity for delay with the aim of maximising user satisfaction. However this presupposes knowledge about the state of the network.

3.2 Network Quality of Service

QoS is usually quantified at the network level. At this level there are two challenges; how to discover the QoS that is being experienced by traffic and how to communicate this information in a timely and useful way to the application.

1. **Round Trip Time:** The length of time to send a packet to a remote host and back to originator. In general, the lower this value the better – links with a high RTT appear sluggish and unresponsive. However most users will not notice delays around the 10s or possibly 100s of ms range (Abdelzaber and Bhatti 1999).
2. **Jitter:** The variation in the RTT measurements recorded. Low jitter is better as it indicates a more stable and therefore more predictable link. Jitter is important for interactive, real-time or streaming applications. The amount of jitter defines the buffer size required for playback.
3. **Bandwidth:** The amount of data that is transferred per unit time. Together with RTT this impacts on the delay experienced by a client. To use an analogy with water distribution; RTT is the length of the pipe and bandwidth the width.
4. **Packet Loss:** The proportion of packets lost on a link in a given time period. Lower loss values are better and indicate a more efficient use of a link: Lost packets are wasteful as the network resource used to send the lost packet is wasted and further wastage is introduced as a duplicate packet has to be sent.

The approach adopted in this work is to alter the fidelity of media in response to feedback on the level of network congestion, thereby achieving the appropriate trade off between technical quality and delay. Consistency of presentation is achieved by setting the fidelity at the start of a user session. In the absence of sharp and prolonged changes in network QoS, a consistent fidelity will be presented throughout a session.

3.3 Adaptability of Media

When considering the types of media available within Web Pages, they can broadly be categorised into one of several categories, some of which may be adjusted to account for the QoS that a connecting client is receiving, and some which may not. Table 1 provides an overview of the type of resources and the level of QoS adaptation that can be applied to them:

Table 1. QoS Adaptability of Media

Media Type	QoS Adjustable?
Plain text – HTML, TXT, PDF etc.	Limited
Graphics – GIF, JPG, TIFF etc.	Yes
Video – AVI, MPEG, DIVX etc.	Yes
Streaming Media	Yes
Active Content - Flash, Java etc.	Limited

Document: The document type limits the transformations that can be performed on a given document. Transformations can be made converting Word documents into PDF, which will reduce the file size but also reduce the usability of the document since they are no longer editable. For still lower quality the PDF can be turned into an HTML page. A smarter extraction mechanism can be envisioned which not only extracts the text but also the images. HTML down to plain text is a simple transformation but can result in a large loss of information, such as images, media files and the links relating to other documents. A better solution is to apply a filtering mechanism to the HTML mark-up removing items such as embedded scripting and cascading style sheet information. Additional mark-up within the HTML could also be used to provide QoS hints.

Graphics: Images, with the exception of photographs, could be described as SVG files and then rendered to a lower quality for inclusion in the final Web Sites. SVG is a vector based image format which, when rendered into the more traditional image formats (PNG, GIF etc), will give a better quality of image than if a GIF was reduced in quality. This is because GIF to GIF transcoding has to deal with encoding artefacts from the original GIF image whereas an SVG to GIF transformation does not.

Downloadable Media: Downloadable media, files that are downloaded wholly to the local machine before playback commences, can be adjusted in the temporal and spatial domains for lower file size. Other aspects of the media can be altered such as the encoding CODEC, perhaps choosing a lossy CODEC with a high compression rate over a CODEC with a lower rate of compression but less lossy output.

Streaming Media: Streaming media files that are played back whilst still downloading to the client machines have an extra complication as retransmission of lost packets is not possible due to timeliness constraints imposed by instantaneous playback. The choice of packet repair technique is therefore of vital importance as it can have a large influence on the experienced QoS.

Active Content: Active content proves difficult to adapt since there is a programmatic element to the resource. Embedded media within the active content may be transformed as in the case of graphical or downloadable objects. Other than this a static image of the first frame of the object could be generated but it is unlikely that this will be of more than marginal use.

4 System Framework

There are three main components to the framework: a Network Monitor, a Network Aware Server or Servers and Content Adaptation Tools. Figure 1 illustrates the Network Monitor positioned so that it is able to monitor all incoming and outgoing traffic, with the resulting QoS data being distributed into a Quality of Service Multicast Zone, where all interested servers can then receive it. The Network Monitor uses online passive monitoring techniques (Mogul 1990; Mogul 1992) to discover the QoS that flows are receiving. Passive monitoring is used in preference to active probes for two reasons. Firstly, no extra wide area traffic is generated which

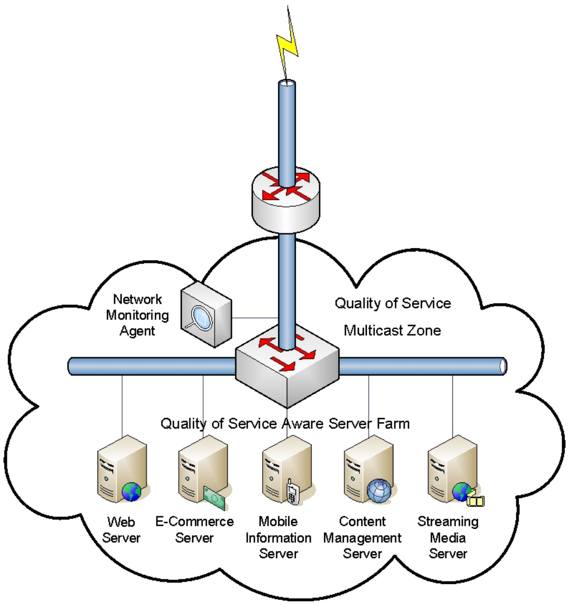

Fig. 1. Network Monitor Positioning

could adversely impact upon the client traffic. Secondly, by extracting measurements from observed traffic there will be the most data for the locations where there is the most traffic. Consequently, predictions of expected network conditions will be more accurate for those locations that use the service the most. Within this framework it is possible to have several different Network Aware Servers, providing a variety of services to connecting clients, each receiving their QoS information from a single Network Monitoring Agent: A large server farm is not required to have a dedicated Network Monitoring Agent for each server in operation.

The Network Aware Web Server provides the same content that a "standard" Web Server would. It is an important constraint of this architecture that the Server be able to make QoS decisions automatically. The user is not being asked to continually give feedback on QoS but is able instead to concentrate on the content. A second important requirement is that client browsers should not need to be aware that they are interacting with a Network Aware Server. Consequently, the Server can and does work with the existing browser base. The Server supplies different versions of content depending on the QoS characteristics of a link. It does not do this by transforming the content on the fly as this would take up valuable processing power and introduce extra latency. Instead the Server chooses between different versions of a Web Site that are held on backing store.

Although it is possible for Web developers to supply different Web Sites for different connection qualities, if this was the only option, the extra work would probably be a disincentive to using the system. A tool suite has been developed which takes as input a single high quality Web Site and converts it into a number of versions appropriate for different link qualities. This submission process takes place only once and is automated. The tools also facilitate human intervention in order to allow the optimisation of an automatically generated site.

4.1 Network Monitor

The structure of the network monitor is shown in figure 2. The design, implementation, and evaluation of such a monitor is described in more detail in (Ruddle, Allison et al. 2000; Ruddle, Allison et al. 2002). TCPDump is used to capture packets. This gives flexibility in filtering traffic and allows trace files to be made for post-mortem analysis. The header fields from each packet are passed to the monitor. The monitor has a modular layered structure.

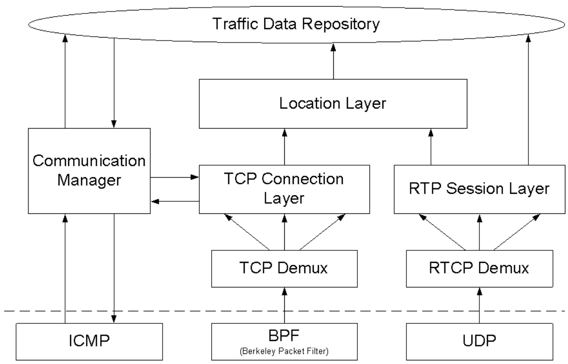

Fig. 2. Network Monitor Structure

The Location Information Server (LIS) extracts congestion information from Transport Control Protocol (TCP) data streams and feeds this information back to end points, The TCP-LIS is located on the WAN/LAN Interface, and uses passive monitoring of TCP traffic to obtain network level information such as congestion events and RTTs. Information extraction occurs within a connection layer where per connection state is maintained. Each reading is then passed to a location layer where statistics are maintained on a per location basis.

At the connection layer state is maintained for all open connections. As each packet arrives a hash table look up is done. This locates the state for the connection that the packet belongs to. Both TCP Macro state, which controls the set up and closure of connections, and Micro state, which controls the flow of data, are monitored. Packet loss is detected, by the observation of duplicate acknowledgements, or by the observation of a repeat packet. RTTs are measured by observing the delay between the receipt of an acknowledgement and the observation of the packet that generated the acknowledgement. Only one RTT measurement is attempted per window of data. When a connection finishes all state associated with that connection is reclaimed.

When events occur such as a packet being observed, a loss being discovered or the RTT being measured these are communicated to the Location Layer. The Location Layer maintains state for aggregates of IP addresses where a Location is defined as an aggregation of hosts to which traffic is likely to experience similar network conditions. State maintained for each location includes the proportion of packets lost, the RTT and the maximum bandwidth observed.

The observation of SYN packets allows new connection requests to be detected and predictions of likely network conditions for that connection to be communicated to the local host in a timely fashion. These predictions are contained within a Location Information Packet. Extensions to the LIS, which support the sharing of congestion information between TCP and Real Time Protocol (RTP) traffic, have been designed (Miller 2002).

4.2 Network Aware Web Server

A list of Quality of Service Aware media types was defined above. Here it is necessary to actually decide how to dynamically choose which quality version to send to a given client. In order to facilitate this decision mechanism, an integrated, modular framework into which Network Aware Servers can be plugged has been developed and is illustrated in Figure 3. A packaging system has been adopted which has seen the components in the systems separated into a number of distinct areas:

1. **Server Realm** – this package contains the server functionality and interfaces directly with the connecting clients, fulfilling their requests as appropriate.
2. **QoS Realm** – this package is responsible for gathering, and providing to the Server Realm, all available QoS information concerning currently active connections.
3. **DataStore Realm** – this is a utility package used by both the Server and QoS Realms and provides the functionality to quickly store and retrieve data on an as-required basis. Within this category there are several implementations of DataStores, both managed and unmanaged, that are customised to specific needs, whatever they may be.

This packaging of components simplifies code management and helps to make clear the lines of responsibility within the system. For example any server related operations are managed within the Server Realm whereas any QoS or Data Storage operations are managed within the QoS and DataStore Realms respectively.

Once the QoS information has been received by an interested Network Aware Server, it is then equipped to deal with a connecting client in the most suitable way. The flow of information through the proof-of-concept Network Aware Web Server is summarised in Figure 3 above. Owing to the unreliable transmission of QoS data into the framework, safeguards have been implemented to ensure the correct operation of the system in the absence of required QoS information. As Figure 4 shows, if QoS information is lacking, the framework reverts to “standard” operation, serving the non-QoS adapted version of the requested resource.

Assuming access to the required QoS information is possible, the framework implements only marginal changes to the model implemented in “standard” Web Servers, with the Connection Handler speaking to the Link Evaluation Mechanism and then the Content Provisioning Mechanism before returning the most appropriate resource to the requesting client. It is at this stage that indirection is used to send to a client the most suitable resource (based on the quality of the client link). At present

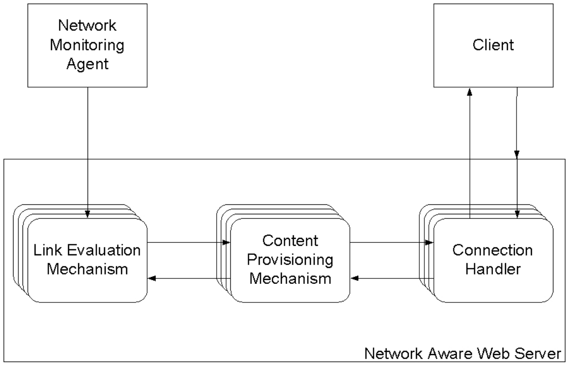

Fig. 3. Network Aware Web Server Architecture

the framework has been developed to support five distinct quality levels – this is done in order to provide distinct sets into which various links can be graded and thus reduce the possible problems caused by a slightly erroneous quality classification – there is less likelihood that small errors will cause a change in grading and so we protect the perceived user quality by reducing unnecessary quality reclassifications, therefore ensuring a more consistent session presentation.

The steady state behaviour of TCP provides a benchmark against which the behaviour of other congestion control schemes have been judged (Mahdavi and Floyd 1997; Floyd 2001). The steady state behaviour of TCP is given by the following equation (Mathis, Semke et al. 1997) where MSS is the maximum segment size on a link, C is a known constant, RTT is the round trip time and P is the probability of loss. Estimates of RTT and P are provided by the LIS, MSS for a connection is known and C is a known constant.

$$T = \frac{MSS * C}{RTT * \sqrt{P}} \tag{1}$$

If the data flow is application limited low levels of loss may suggest an available bandwidth in excess of the actual physical bandwidth. This can be accounted for by measuring the actual bandwidth utilisation and setting the estimated available bandwidth to the minimum of that estimated from congestion feedback and actual utilisation.

The bandwidth category that a connection request falls into (Q) is then given by:

$$Q = \min\left(\left[T = \frac{MSS \times C}{RTT \times \sqrt{P}}\right], \max(B)\right) \tag{2}$$

In this way the expected available bandwidth for a destination can be calculated based on past measurements. The result may not correspond to the actual access bandwidth, but may rather correspond to the share of bandwidth available on the bottleneck link. Having established the bandwidth available to a destination the server can then choose the appropriate version of the site to transmit.

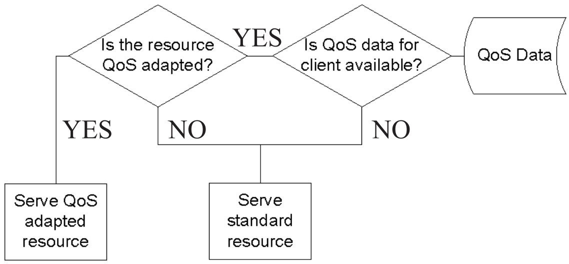

Fig. 4. QoS Decision Mechanism

The system makes use of the concept of a human session in meeting the consistency requirement. The aim is to provide a consistent fidelity through the lifetime of a session. Here we define a human session as an amount of time spanning a series of requests during which a user might alternate between assimilating the information within a page and downloading new pages. For example a user receives Page 1 from a site and spends 10 minutes reading it, he then proceeds to request Page 2; ideally we want to ensure that the quality of Page 2 is similar to the quality of the Page 1 that has already been received – this will ensure continuity throughout the browsing session. In this situation the Human Session timeout should not be set to anything below 10 minutes. During a session all pages should aim to be served from the same QoS adapted version of the site.

Regardless of any Human Session periods that attempt to give a unified feel to the browsing experience, changes in the QoS readings are tracked during a session. If an initial link quality evaluation is found to be consistently lacking then it may be necessary that remedial action is undertaken, resulting in the link quality being re-graded, in order to provide the most suitable browsing experience to the connecting client. However such a decision should not be taken lightly and cannot be based solely on one, or possibly a handful of bad readings, instead historical data should be considered, with the weight attached to historical data diminishing over time as the data becomes more and more out of date. In order to achieve this data smoothing an Exponentially Weighted Moving Average (EWMA) is used. It also deals with the ageing of historical data, something that the arithmetic mean cannot do.

5 Resource Mapping

There are perceptual limitations to the quality of media that is worth transmitting. For example there is little point in transmitting a picture with a resolution greater than that of the display. There is also little point in encoding a video file at more than 25 frames per second, with around 12 to 15 fps being the lowest point where motion is still perceived without jerkiness. When considering the delay associated with the transfer of resources, we must take care to ensure that the user is provided with feedback as quickly as possible; at tens of milliseconds delays become perceivable. Whilst at seconds delays impact negatively upon user perceived quality. Pages downloading in times exceeding tens of seconds have a serious impact on user perception, with people believing that the site is of low quality and thus highly likely to abandon their browsing sessions (Bouch, Kuchinsky et al. 2000). For each page there should be a maximum delay of ten seconds between the user request being made and the page being fully displayed. Unfortunately differing processing capabilities on client machines mean that this is an almost impossible target to achieve since we have no indication of how long a given page will take to render on each client device.

The approach adopted in this paper is to provide a series of tools which, when given a single high quality Web Site and a meta-level description, are able to transform the single site into a number of Web Sites which are suitable for a range of bandwidths. The transformation happens once offline and creates a series of Web Sites that are aligned for the quality levels chosen. With this approach we avoid the situation of having to re-encode media files on the fly which could add an unacceptable amount of delay to loading the Web Page. There are two issues that need to be addressed. Firstly, extra storage space is required for multiple versions of the Web Site. As storage is cheap and given that the amount used does not negatively impact on the user perceived QoS this is a reasonable cost to pay. Secondly, a method for the handling of dynamic content is needed. The use of a Meta Interface Model (MIM) would allow the customisation of the dynamic content along similar lines to static content. Little extra delay would be introduced since all the images and media would have already been encoded and it would only be customisation of the textual components (HTML, scripting, CSS etc) which would need to be done dynamically, templates could be generated in advance along with the static content.

Aligning the Web Sites with the bandwidths for common access technologies provides the choice of target bandwidths. We choose the following bands; mobile phones (at 0-10 Kbit/sec), modems (10 – 56 Kbit/sec), ISDN (56 – 150 Kbit), broadband 150 Kbit/sec – 1.5 Mbit/sec) and finally local area network (1.5 Mbit/sec and above).

6 Evaluation

The predictive powers of the network monitoring agent has been evaluated using live Internet experiments, between hosts in the US, the UK, Spain, Germany and Eastern Europe. Figure 5 shows the predicted verses the actual levels of congestion

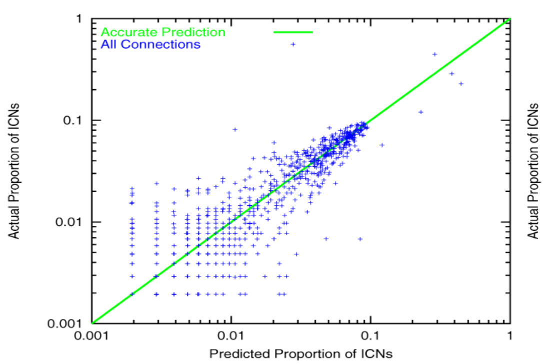

Fig. 5. Predicted vs. Actual Congestion

experienced. The metric used is the proportion of Implicit Congestion Notifications (ICNs). Packet loss is taken as an implicit indication of congestion. Yet in a single congestion event several packets may be lost, consequently in these experiments only the first packet loss in a window of packets is taken as an ICN.

It is of interest that on links experiencing high levels of congestion the accuracy of prediction increases. Furthermore it was found that the correlation between prediction and result remained high for periods of time in excess of 20 minutes. These results suggest that it is possible to use the measurement of past traffic to predict the network conditions that are likely to be experienced by future traffic. Consequently the approach adopted here is valid.

7 Conclusion

We have presented the design, implementation and evaluation of a framework for making the Web Quality of Service Aware. The approach adopted is to use passive monitoring of transport level headers to make predictions about the QoS that is expected to be experienced by a particular location. A mechanism for translating a single high fidelity Web Site into a set of Web Sites that are appropriate for different bandwidths has been outlined. When a user session starts the Web Server uses QoS information to determine which bandwidth is appropriate. It then serves up Web Pages from the appropriate Web Site. This approach allows Web designers to leverage the increasing bandwidth many clients have available without producing Web Pages that are inaccessible to others.

References

Abdelzaber, T., Bhatti, N.: Web Server QoS Management by Adaptive Content Delivery. In: International Workshop on Quality of Service, London (1999)

Barnett, C.M.: Adaptive Content Web Serving (2002) Retrieved June 20, 2005, from http://zoo.cs.yale.edu/classes/cs490/01- 02b/barnett.casey.cmb68/paper.htm

Bhatti, N., Bouch, A., et al.: Integrating User- Perceived Quality into Web Server Design. In: 9th International World Wide Web Conference, Amsterdam (2000)

Bouch, A., Kuchinsky, A., et al.: Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service. In: CHI 2000 (2000)

Bouch, A., Sasse, M.A.: Network Quality of Service: What do users need? In: 4th International Distributed Conference, Madrid (1999)

Bouch, A., Sasse, M.A.: The Case for Predictable Media Quality in Networked Multimedia Applications. In: MMCN 2000, San Jose, CA (2000)

Floyd, S.: Less Conservative Congestion Control (2001)

Floyd, S., Jacobson, V., et al.: A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Transactions on Networking 5(6), 784–803 (1997)

Jacobson, V.: Congestion Control and Avoidance. ACM Computer Communications Review 4(18), 314–329 (1988)

Jain, R., Ramakrishnan, K.K.: Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals and Methodology. In: IEEE Computer Networking Symposium, IEEE, Washington D.C (1988)

Mahdavi, J., Floyd, S.: TCP-Friendly Unicast Rate-Based Flow Control. Technical note sent to the end2end-interest mailing list (1997) http://www.psc.edu/networking/papers/tcp_friendly.html

Mathis, M., Semke, J., et al.: The Macroscopic Behavior of the Congestion Avoidance Algorithm. Computer Communications Review 27(3) (1997)

Miller, A.: Best Effort Measurement Based Congestion Control. Department of Computer Science. University of Glasgow, Glasgow (2002)

Mogul, J.C.: Efficient use of Workstations for Passive Monitoring of Local Area Networks. ACM SIGCOMM 20(4), 253–263 (1990)

Mogul, J.C.: Observing TCP Dynamics in Real Networks. ACM SIGCOMM 22(4), 305–317 (1992)

Pradhan, R., Claypool, M.: Adaptive Multimedia Content Delivery for Scalable Web Servers. International Network Conference, Plymouth (2002)

Ruddle, A., Allison, C., et al.: A Location Information Server for the Internet. In: IEEE International Conference on Computer Communications and Networks, IEEE, Las Vegas, NV (2000)

Ruddle, A., Allison, C., et al.: A Measurement Based Approach to TCP Congestion Control. European Transactions on Telecommunications 13(1), 53–64 (2002)

Shaha, N., Desai, A., et al.: Multimedia Content Adaptation for QoS Management over Heterogeneous Networks. In: International Conference on Internet Computing, Las Vegas, NV (2001)