Interactive Web Service via Satellite to the Home

Hyoung-Kee Choi, Georgia Institute of Technology
Osama Qadan, Astrolink International LLC
Dolors Sala and John O. Limb, Broadcom Corp.
Jeff Meyers, Echo Star

ABSTRACT

The satellite distribution of digital broadcast signals to the home has become widespread in the last seven years, delivering hundreds of television channels. More recently we have seen other services appear, such as Direct PC™ where data is broadcast. This service may be used to deliver information like stock quotes to a large number of users, or information such as Web pages may be requested by a single user. In the latter case, however, a return path is needed from the end user to the service provider in order to provide the specific information requested by the user. In services offered today this information is usually provided by a terrestrial return link, most often a telephone line. While a telephone line with a modem is adequate in most cases, it has the drawback of tying up an existing telephone line for a considerable amount of time for very little data transmission, or requiring an additional line. Furthermore, in many parts of the world a telephone line may not be readily available. In these cases a satellite return link may be a viable option using VSATs. In this article we describe an investigation and simulation of a return path VSAT link designed specifically for delivery of Web traffic requests to a base station and thence via a terrestrial link, using the Internet, to a Web server. The focus in this work is on the MAC protocol to support such a satellite channel and on modeling the whole service from traffic generation through transport, network, and MAC layers to the physical channel.

INTRODUCTION

Satellites have played a major role in global telecommunications for the past four decades. Their ability to broadcast over wide geographical regions has made satellites attractive for broadcast services such as digital video and data. In traditional satellite systems, earth stations act as peers on the two sides of the connection. Recent trends allow smaller terminals such as very small aperture terminals (VSATs) to communicate with other Earth stations around the globe. We wish to determine how well suited this channel is for providing interactive World Wide Web traffic.

There are many applications running over the Internet; for example, Hypertext Transfer Protocol (HTTP) [1], File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Network News Transfer Protocol (NNTP), and Post Office Protocol (POP). The amount of HTTP traffic has been increasing rapidly as people gain access to the Internet at more affordable rates. About 75 percent of traffic is associated with HTTP traffic in the Internet backbone [2]. Due to this high demand, researchers are studying Web traffic characteristics and seeking methods to improve the speed at which it may be delivered.

The significant characteristic of Web traffic is its asymmetrical nature. More traffic flows in the forward link (from Web servers to stations) than the return link (from stations to Web servers). Users usually make short requests to download large responses stored at servers. The asymmetrical nature of Web traffic suggests a good match to satellite systems since the VSAT return link capacity in a satellite channel would be much smaller than the forward link capacity.

There is a strong similarity between satellite channels and hybrid fiber coax (HFC) networks [3]. Similarities include the asymmetrical nature of the channel, the fact that there is no direct connection between stations (communication must pass through the hub/headend), and the round-trip delay between stations and the hub cannot be ignored. These similarities suggest that medium access control (MAC) protocols, originally devised for HFC networks, might easily be adapted to the satellite channel. In this article, an HFC MAC protocol called centralized priority reservation (CPR) [4] is adapted for use with a satellite channel. This protocol is of the class *reservation-ALOHA* [5], and is described. But there are significant differences between HFC networks and satellite channels.

0163-6804/01/$10.00 © 2001 IEEE

The most important is the round-trip time (RTT) between stations and hub/headend. It is approximately 0.5 ms for an HFC system and 500 ms for a satellite system.

The remote stations under consideration are Web clients using the TCP stack as their primary transport protocol. Remote stations send a request to the hub to gain access to the physical channel. The hub receives the request and schedules it according to the CPR protocol. The physical channel is divided into a number of forward and return links. Forward links are generally larger in capacity than return links, since they carry more data to the remote terminals than the return link carries to the hub. A number of 32 kb/s return links deliver requests from stations to the Internet, and a 45 Mb/s forward link delivers responses to stations in a time-multiplexed fashion.

We describe the basic MAC protocol. See also [6] for a review of MAC protocols applied to satellite communications. Descriptions of the Web traffic model and generator are then given. The simulation structure is presented, and the results obtained are presented and analyzed. A comparison between the existing protocol and CPR is made and the performance evaluated. We conclude with a discussion of the limitations and extensions of the protocol.

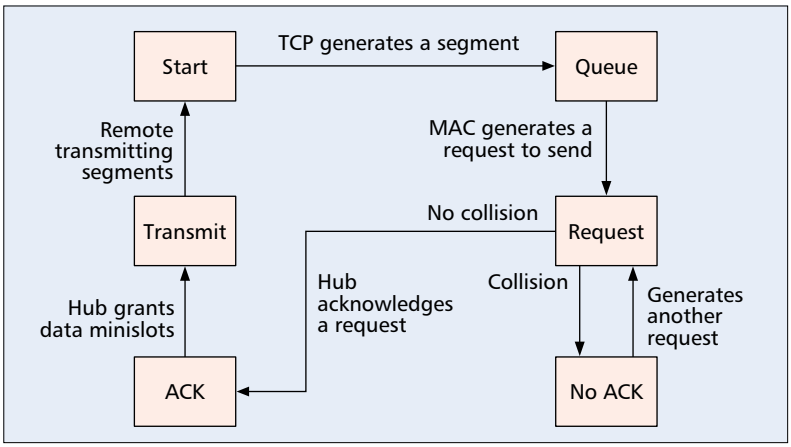

Figure 1. *A state diagram of the MAC protocol.*

Protocol Description

Slot Structure

A *minislot* is the basic unit of transmission used by the MAC layer. A sequence of minislots allocated for transmission of data is called a *data slot*. A data slot consists of a variable number of data minislots (DMSs) and a header minislot (HMS). A request for DMSs to the hub fits in exactly one minislot, called a contention minislot (CMS). The HMS and CMS are similar except that the CMS carries a request-size field. A data-type field or priority field could also be added to the CMS. The hub controller schedules the requested number of minislots to carry the variable-length MAC frame. The hub broadcasts to the station information about the return link minislot structure using a periodic bitmap. The broadcast structure includes the type of the minislot, the station that may use it, and other control information.

The size of the minislot is a design choice, and can be set independent of other parameters. The size of a minislot was chosen to be 20 bytes because the size of many IP packets (e.g., most FIN and ACK segments) sent by remote stations are a multiple of 20 bytes. SYN, FIN, and ACK segments constitute 86 percent of the total number of segments sent by remote stations [7].

Basic Protocol

The basic MAC protocol used in this work is similar to CPR in the HFC network [4]. The state diagram of the MAC protocol is shown in Fig. 1. When an IP packet is generated at the remote station, it is passed to the MAC layer. On receiving a packet the MAC divides it into data slots, which occupy multiple MAC-level minislots in the return link. Before transmitting data slots the remote station sends a request to the hub by using a CMS. The request is sent on a contention basis. The CMS is written in by any remote stations having a request to make at that time, and hence a collision may occur between remote stations trying to write in the same request minislot. Provided that no collision occurs on the request minislot, the hub acknowledges the request immediately. The hub then schedules a time for the remote to transmit and then sends a grant message to the remote station indicating the number of data slots the remote station has asked for. The hub is responsible for controlling access to the return link and broadcasting grant information to the remote stations as a bitmap. A simple scheme to implement the bitmap uses a single bit to distinguish a data minislot from a contention minislot. The grant is scheduled to arrive at the remote just prior to the data slots that the remote station will use. No collision will occur in a data slot since only the granted remote can write in a data slot. If a collision occurs when the request is being sent, the remote will not receive an acknowledgment from the hub after a round-trip delay and will then attempt to retransmit the request. A contention resolution algorithm (CRA) is used to resolve the collision [4].

The CRA defines the set of rules used to resolve the contention. We used the *p-persistent* algorithm in this study [5]. The p-persistent algorithm allows remote stations to transmit a request in a CMS with probability p. The performance of the algorithm is sensitive to the probability of retransmission, p. The throughput of the p-persistent algorithm is

$$\eta = N_a * p * (1 - p)^{(Na-1)}, \qquad (1)$$

where N_a is the number of stations contending in a particular CMS [4, 5]. The maximum throughput is achieved when $p = 1/N_a$, and asymptotically approaches $1/e = 36.7\%$ as the number of contending stations increases.

Since neither the hub nor the individual remotes are capable of determining the number of active stations (N_a) contending on the channel, an estimation algorithm is used. In this model we are using the *psuedo-Bayesian* algorithm proposed by Rivest [8].

The rule followed in allocating CMSs is rather simple: allocate all minislots that are not used

An accurate model of Web traffic is essential for estimating the load that may be carried on the channel. A complete model helps in studying interactions with the underlying MAC layer and determining how to get the best performance from the system.

Parameter	Description
Object size	The size of objects (main and in-line) stored on the remote server
Request size	The size of HTTP header sent when requesting URIs
Number of in-line objects	The number of embedded objects in a page in which requests are made
Viewing (off) time	User thinking time
Number of (non-)cached pages	The number of consecutive pages that are (not) locally cached in the browser
Parsing time	The time for a browser to parse the HTML code
Whole page delay	The time to download a page
In-line interarrival time	Interarrival time between starts of in-line objects

Table 1. *Descriptions of the modeled parameters.*

for data slots as CMSs [4]. This is a self-regulating rule; if the number of CMSs is low, few requests will arrive at the hub, and as a result more CMSs will be allocated. On the other hand, if the number of CMSs is high, the probability of successful requests increases, and, as a result, fewer CMSs will be allocated.

The Web Traffic Model

An accurate model of Web traffic is essential for estimating the load that may be carried on the channel. A complete model helps in studying interactions with the underlying MAC layer and determining how to get the best performance from the system.

To model Web traffic, traces were collected from 11 a.m. to 12 p.m. on Wednesday October 7, 1998, running on a Sun Ultrasparc2 (180 MHz) workstation. The traffic was measured on the backbone network of the Georgia Tech campus where a large cross-section of Web traffic can be observed. More than 1900 clients participated in Web browsing sessions producing about 24,000 Web pages. The packet capturing utility, *tcpdump*, was used to record packets flowing on the network [9].

We examined transactions associated with a single Web page retrieval and selected a set of parameters (Table 1) that can be used to define the Web traffic. To derive a statistical model for the Web traffic we extracted measurements of different parameters from the raw trace files using a practical extraction and report language (PERL) script. Once empirical distributions for individual parameters were obtained, we selected a best-fit distribution from a set of standard probability distributions. The parameters used to model the Web traffic and their expected distributions are listed in Table 2.

Based on the measurements, we generated synthetic traffic according to HTTP v. 1.0 with multiple connections [10, 11]. The traffic generator simulates an ON/OFF source where the ON state represents the activity of a page retrieval and the OFF state the silent period after the retrieval. The durations of ON and OFF states correspond to whole page delay and viewing time, respectively. Viewing time denotes any time the browser is inactive. Whole page delay is the time taken to fetch all objects in a Web page. Thus, it includes a number of round-trip times between client and server.

A typical Web page consists of a hypertext document. The hypertext document is in Hypertext Markup Language (HTML): coded text with links to other objects that make up the whole page. An object is an entity stored on a remote server as a file or generated by a server based on a request. There are two kinds of objects, a main

Parameters		Mean	Median	S.D.	Best-fit
Request size		360.4 bytes	344 bytes	106.52	Lognormal
Object size	Main	10709.8 bytes	6094 bytes	25032.1	Lognormal
	In-line	7757.74 bytes	1931 bytes	126168	Lognormal
Parsing time		0.132 s	0.056 s	0.187	Gamma
Number of in-line objects		5.55	2	11.35	Gamma
In-line interarrival time		0.86 s	0.17 s	2.15	Gamma
Whole page delay		11.35 s	3.67 s	23.85	Weibull
Viewing (OFF) time		39.45 s	11.71 s	92.57	Weibull
Number of pages	Noncached	12.58 pages	5 pages	21.623	Lognormal
	Cached	1.74 pages	1 pages	1.74	Geometric

Table 2. *Web traffic parameters and their statistics, including expected distributions.*

object and an in-line object. The file containing HTML code is a main object, and the objects linked from the hypertext document are in-line objects.

A request for a page results in the server transmitting the main object. Once the main object is received, there is a delay during which the main object is parsed and the server transmits the first in-line object, after which a request for the first in-line object is generated. The traffic of the second in-line object is then generated one in-line interarrival time after the start of the first. Subsequent in-line objects are generated in a similar fashion. After all objects are transmitted the model is silent for a period, simulating the time a user spends viewing the downloaded page. After the expiration of the viewing time, the model starts to generate a new page.

In TCP, segments are transmitted in a burst until the TCP window is closed. The interval between two successive bursts is called a *stall*. We use a TCP model in which the number of segments transmitted and the duration spent in a stall are chosen from an empirically derived distribution. A detailed discussion of the Web traffic model can be found elsewhere [7].

Traditional traffic models (Poisson process) cannot be applied to Web traffic because of the relatively large burstiness of the traffic. Web traffic is bursty on many timescales and exhibits a significant amount of self-similarity [12]. Hence, it is important that our traffic model also shows self-similarity. Self-similarity in our traffic model was explored and measured by a number of methods [10]. For example, the *Hurst* parameter was 0.81 for the trace and 0.74 for the traffic model. Thus, while not quite as great as the trace, the model exhibits significant self-similarity.

The trace was gathered from within a university where most of the users[1] are connected to Ethernet. The trace used in this study may not be typical of the satellite system simulation for two reasons: the uniqueness of students' interests and the small delay to access the Web students experience with Ethernet. Students more often visit technical Web sites, and the model parameters are affected by the design of the Web pages. Students experience less delay to access the Web because of the high-end computing environment of the university. In contrast, users experience relatively longer RTT in the satellite channel, and their behavior surfing on the Web might differ slightly from fast-access users. For example, users experiencing longer whole page delay tend to press the stop button before completing the downloading of a Web page. Although the simulation incorporated the longer RTT of the satellite channel, we were not able to reflect typical satellite users' behaviors in the simulation.

Our estimates of whole page delay for the satellite channel might contain some bias. However, the model used in this study is suitable for comparing the performance of the proposed protocol with other protocols and studying possible improvements to the protocol.

[1] *Georgia Tech allows students to connect through a dial-up connection. At any given time, the number of dial-up users is much smaller than that of Ethernet users.*

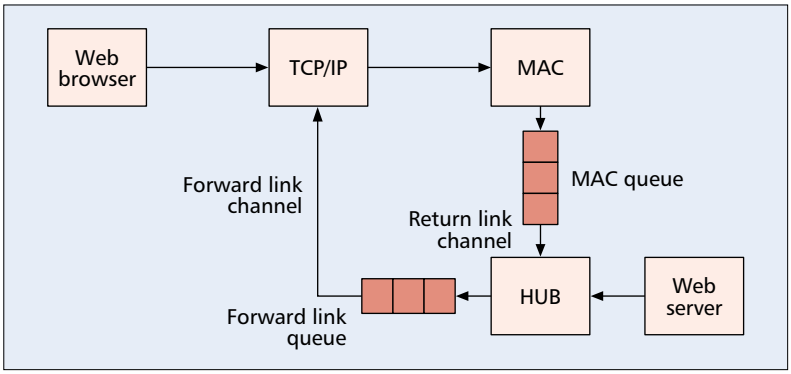

Figure 2. *Software implementation of the simulation.*

SIMULATION

SIMULATION CONSIDERATIONS

Some design decisions were made in the course of simulating the system. Those decisions were made after taking into consideration both practical implementation issues and performance issues.

The proposed MAC protocol and a Web traffic generator were simulated in software written in C. The length of the simulation was chosen to be 3750 s (750,000 minislots). This allows the system time to stabilize and the remote stations to transmit a number of Web pages. The results obtained from the simulation included whole page delay, channel utilization, and forward link bit rate.

RETURN LINK SIMULATION

The block diagram of the simulation implementation is shown in Fig. 2. The first module is the Web browser module. This module contains distributions to model the Web traffic (Fig. 2). The distributions determine how frequently IP packets are generated, the size of each generated segment, the number of connections in a page, and so on. The output of the module is then fed to the TCP/IP module that generates IP packets according to the TCP/IP protocols [13].

Packets generated by the TCP/IP module are then fed into the MAC module. Frames in this module are stored in a first-come-first-served (FCFS) queue until they leave. Frames travel over the return channel to the hub module, which acts as an Internet service provider (ISP) to the system, transmitting the IP packets to the Web server module. Upon receiving the IP packets, the server module responds according to the statistics generated by the model. When the server's response arrives at the hub module, the hub module transmits the frame to the remote stations via the forward link. The 45 Mb/s forward link was simulated as an FCFS queue. After receiving the frame from the hub module, the remote TCP/IP module goes to the next step in its TCP sequence.

The simulation proceeded in two phases. In the first phase, we wished to determine throughput of a single 32 kb/s return channel. For this

The return link simulation is used to produce the traffic trace on a single return channel. The traffic trace contains two kinds of information, the interarrival time of packets and the size of the packets.

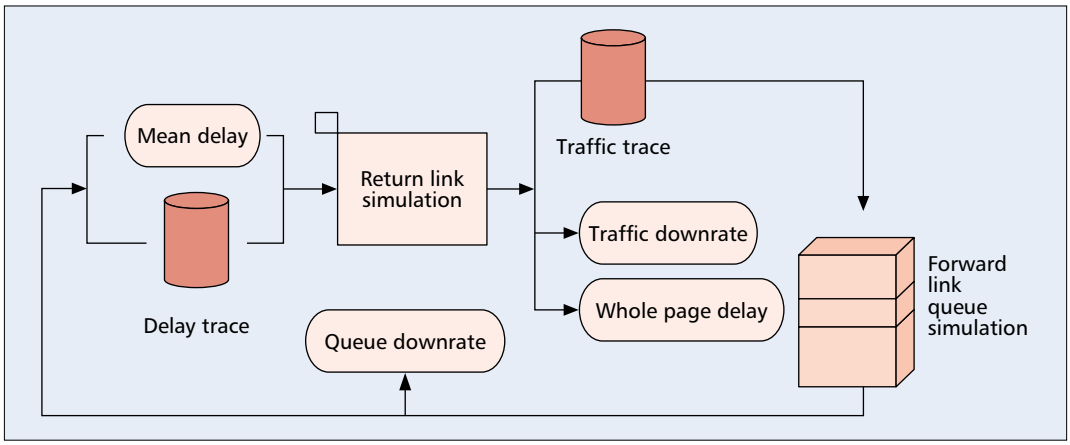

Figure 3. *Full simulation approximation.*

purpose, we assumed that the forward link added no queuing delay. This is tantamount to assuming that the forward link has infinite capacity. From this simulation, we decided that a good compromise between throughput and delay was achieved with a load of about 40 users per 32 kb/s return channel. We refer to this simulation as the return link simulation.

In the second phase of the simulation, we sought to answer the question of how many fully loaded return channels can be supported by a forward link capacity of 45 Mb/s. We refer to this simulation as the *full* simulation. A full simulation has to model the forward link in more detail. The full simulation assumes multiple return channels on the satellite link. A complete 45 Mb/s transponder channel is assumed for the forward link. The return link consists of multiple 32 kb/s channels, each channel serving 40 stations.

Full Simulation Approximation

To determine how many 32 kb/s return channels can be supported by a 45 Mb/s forward link, the 45 Mb/s forward link is loaded with as many return channels as are required to produce congestion in the forward link. However, to fill the 45 Mb/s forward link with traffic generated by a fully loaded 32 kb/s return channel requires at least 200 return channels or more (each with 40 stations). Simulating such a large number of return channels requires a huge amount of computational power and memory storage to obtain results in a reasonable time. The long runtime also makes it difficult to explore the parameter space as thoroughly as we would like. For this reason, we approximated the full simulation when the number of return channels we were simulating was large.

The approximation to the full simulation proceeds as shown in Fig. 3. Rather than simulating all return channels, we simulate just one. The effect of one channel on the others comes only from the delay experienced in the forward link when the Web pages from all requesting stations are multiplexed onto the forward link. We can approximate the forward-link delay from a forward-link queue simulation and introduce it into the return link simulation. This added delay will change the output of the return link simulation. The return link simulation and the forward link queue simulation are repeated, in turn, until the simulation converges.

The return link simulation is used to produce the traffic trace on a single return channel. The traffic trace contains two kinds of information, the interarrival time of packets and the size of the packets. In addition, whole page delay and the actual forward link rate (*traffic downrate*) are derived from the return link simulation. The traffic trace is then used as an input to the forward link queue simulation. The rate at which the queue is emptied is the same as the capacity of the forward link, 45Mb/s. The input traffic is generated by randomly choosing a starting point in the trace file. There is the same number of starting points in the trace file as the number of return channels in the full simulation. After the forward link queue simulation, we obtain the mean queuing delay, another trace and the intensity of the queue (*queue downrate*). The trace contains the queuing delay of every packet. The delay associated with the forward link

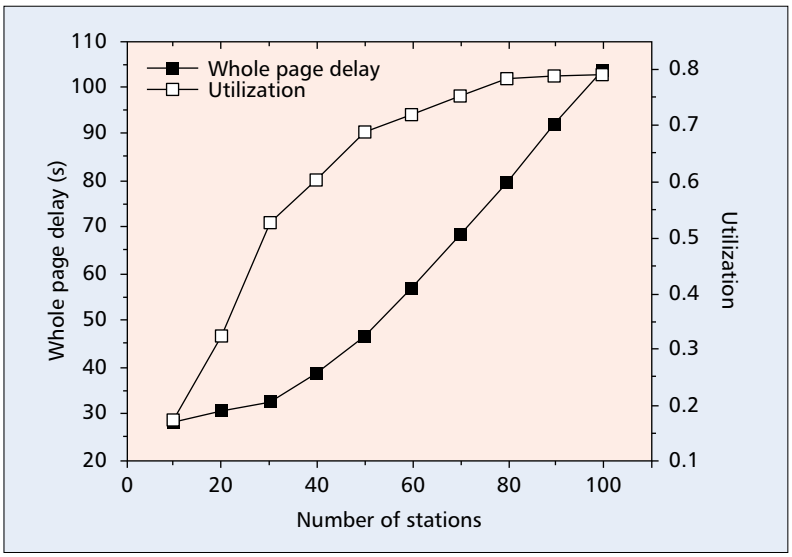

Figure 4. *Whole page delay vs. number of stations and channel utilization vs. number of stations (with piggybacking and grouping). After 50 stations the delay curve becomes linear.*

queue simulation is added into the RTT of the return link simulation to compensate for the effect of the forward link. Either the mean or distribution of queuing delay can be used in the return link simulation. The next iteration of the queue approximation produces another set of parameters. The iteration is continued until the difference between successive queuing delays is within a given range.

Simulation Results

Return Link Simulation

Basic Result — Figure 4 shows the whole page delay for the return link simulation. As previously mentioned this simulation assumes no queuing delay in the forward link. As the number of users increases, the return link utilization slowly approaches saturation. Saturation starts when the number of users is approximately 40 to 50. Upon saturation, we would expect the delay to vary linearly with the number of remote stations. The feedback nature of the Web browsing application (HTTP) will limit the number of packets that can be outstanding so that at no point does the simulation collapse; the response time just becomes slower. In Fig. 4, we see that the linear portion of the delay curve begins when the utilization curve begins to level off.

Grouping and Piggybacking of Requests — The large delays experienced in the satellite environment were not contemplated in the design of CPR for the cable TV environment. Thus, a number of changes were made to the basic protocol to improve performance, including grouping and piggybacking. Earlier implementations allowed one request to be sent for each frame in the MAC queue. Hence, the first improvement made is to group all queued MAC requests into a single MAC request sent to the hub. Whenever a MAC request is made, the MAC layer checks its queue, and requests enough slots to accommodate all the data it has at that time.

The second improvement is the piggybacking of MAC requests on DMSs. When a station sends data in the DMSs, requests for the next data in the MAC queue, if any, are piggybacked at the end of the current DMS. Piggybacking reduces queuing delay by avoiding collisions of requests and adding next requests to the current DMS. Piggybacking can lead to unfairness if a single client monopolizes the system by piggybacking continuous requests. The unfairness may be taken care of at the hub where the scheduler can ensure that grants are equitably assigned to each client.

Grouping of requests reduces delay by approximately 6 s/Web page in the range of interest as shown in Fig. 5 (with piggybacking). The effect of grouping becomes more significant at high loads, where the MAC queues start to build. Under high load, piggybacking and grouping perform similar functions. Both techniques decrease the load on the reservation channel. Figure 5 compares the delay performance of these two techniques. It should be noted that both techniques start to improve performance only when the MAC queues increase in length beyond a service time of one RTT.

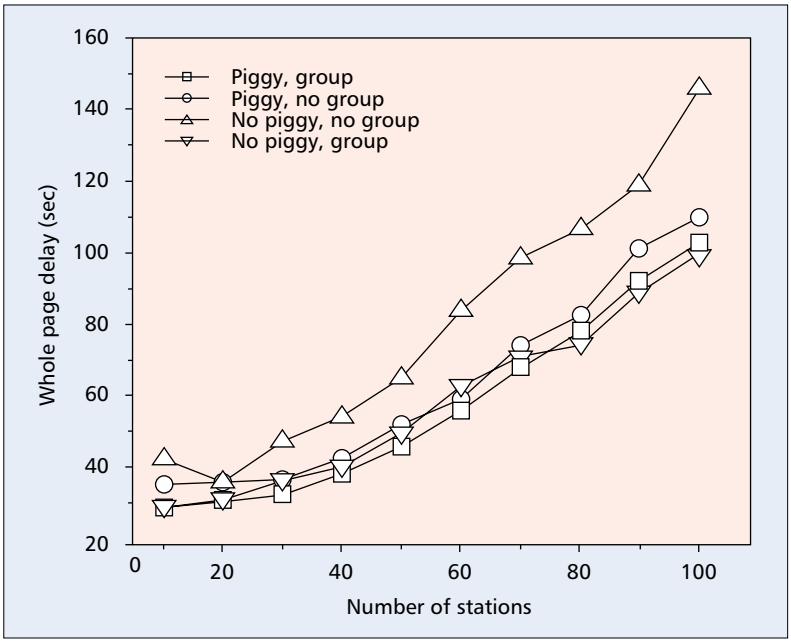

Figure 5. *Delay improvement by possible combination of grouping and piggybacking. The system with both grouping and piggybacking performs best.*

Bitmap Broadcast Frequency — Earlier implementations allowed bitmaps to be broadcast to stations at every minislot, thus increasing the computational load at remote stations [4]. We explored allowing the hub to calculate and broadcast the bitmap for a group of minislots. The hub accumulates successful requests and periodically schedules them. We have chosen two values of bitmap frequency to test the delay performance of the system: one minislot (0.005 s) and a maximum RTT (100 minislots, 0.5 s). The result is shown in Fig. 6. At low loads, the implementation introduces a delay equivalent to half the length of the time differ-

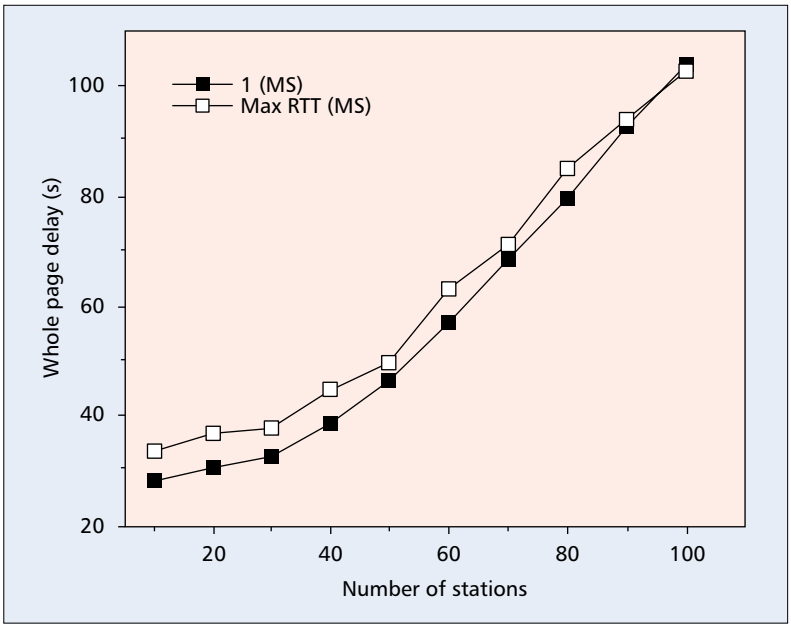

Figure 6. *Delay improvement by varying the frequency of bitmapping (with piggybacking and grouping). As the number of stations increases, the delay improvement becomes less significant.*

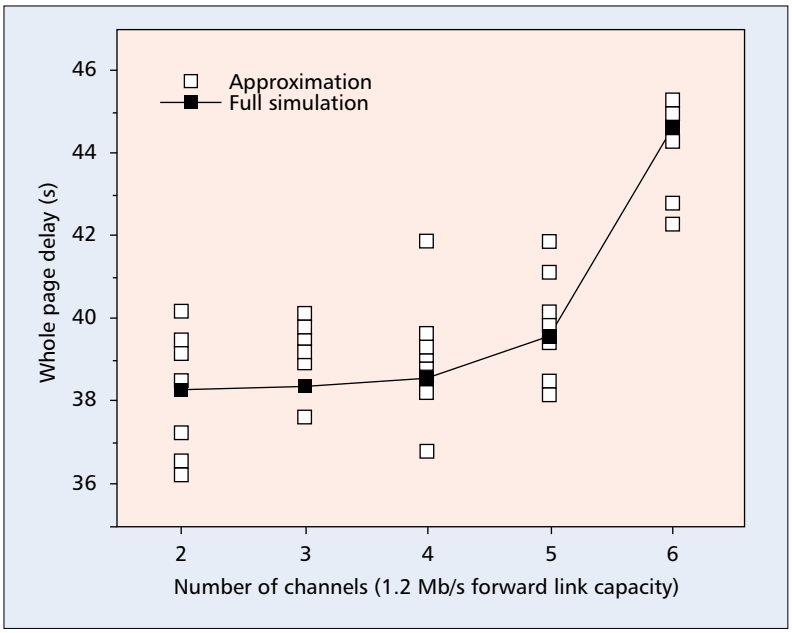

Figure 7. *A comparison between full simulation and approximation; 1.2 Mb/s forward link capacity and 40 stations/channel.*

ence between consecutive bitmap transmissions. The bitmap frequency of one minislot outperforms that of the maximum RTT by 6 s for 40 stations.

A simple calculation supports this result. A request for a grant arriving at the hub experiences an average delay of half the maximum RTT (0.25 s) in one case and a half minislot (0.0025 s) in the other. According to our traffic model there are, on average, 24 requests for a grant per page with grouping and piggybacking. Hence, we have about 6 s difference (24 requests * 0.25 s = 6) between two different bitmap frequencies. At high loads, there are more requests arriving at the hub, and beyond a certain limit requests for grants start to backlog because of the saturation of the link. For a bitmap frequency of one minislot, the backlogged delay waiting for grants in the hub is comparable to the delay of half the bitmap transmissions and the difference becomes less significant. The results of Fig. 6 are consistent with these expectations.

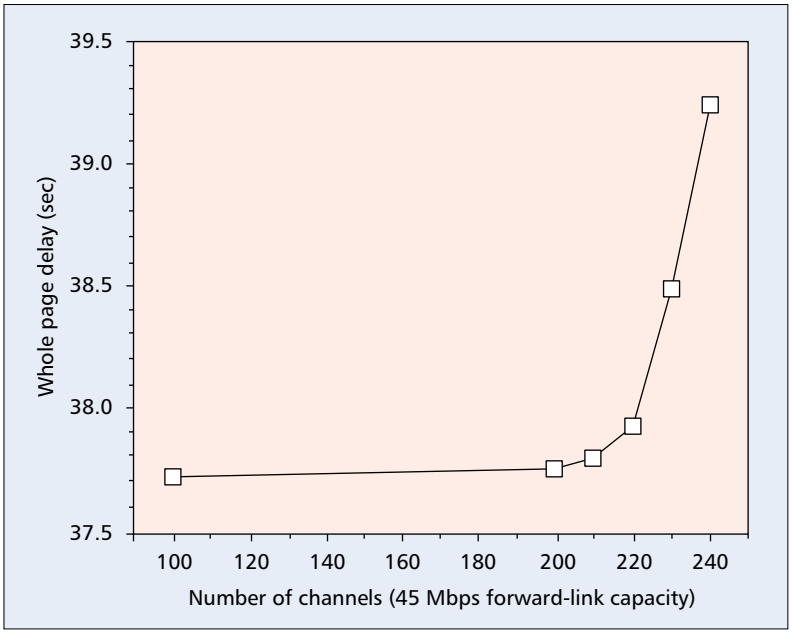

Figure 8. *Full simulation with 45 Mb/s forward link capacity and 40 stations/channel. Whole page delay starts to increase sharply after 220 channels.*

FULL SIMULATION

We first tested the accuracy of the full simulation approximation described. The capacity of the forward link was set to 1.2 Mb/s, where it was still feasible to run the full simulation without an approximation. The full simulation approximation and the full simulation were repeated a number of times; results in terms of whole page delay are shown in Fig. 7. This result confirms the accuracy of the full simulation approximation.

The full simulation approximation produces the mean packet delay and the delay distribution of packets. Either output can be used as the input to the return link simulation. It is faster and simpler to approximate the forward link delay with the mean delay. The results for both cases are very similar. Consequently, all subsequent results are obtained using the mean packet delay.

Figure 8 shows the result of the full simulation approximation for the forward link of 45 Mb/s. The result shows that up to 220 return channels can safely share the forward link capacity, keeping the whole page delay at less than 38 s.

COMPARISON

Comparing the performance of the protocol described here with other published results is difficult for a number of reasons: there are very few actual results reported; the parameters of the simulations differ; different CRAs are used. Furthermore, most of the previous work has been simulated with Poisson-type traffic, while ours was simulated with Web traffic.

Slotted Aloha is an attractive protocol because of its simplicity and because it acts as a benchmark for other protocols. We simulated slotted Aloha with the p-persistent algorithm. The value of p was made adaptive to the number of users waiting to transmit. Slotted Aloha, by definition, has a fixed-length slot. We needed to choose a best size for the slot. Figure 9 compares the performance of slotted Aloha, with different slot sizes, to the CPR in terms of utilization vs. the number of remote stations. Slotted Aloha achieves its best utilization at a lower number of users than the CPR, and the maximum utilization is much less. Whole page delay of the CPR in the range of interest is approximately half what it is for slotted Aloha (Fig. 9). At low loads, a 40-byte TCP/IP packet takes one RTT in slotted Aloha to get it to the hub given that the payload size is two minislots. In the CPR, it takes one RTT to send the request successfully, and a half RTT to send the TCP/IP packet to the hub. With larger TCP segments, our protocol still takes one and a half RTTs to get the segment to the hub, while it takes several more RTTs with slotted Aloha.

DISCUSSION

The network explored in this article uses the satellite link as the physical medium to transport forward and return link traffic. Existing commercial systems, however, utilize the satellite link to carry forward link traffic only, while they use the public switched telephone network (PSTN) to carry return link traffic. Current commercial systems, implemented by Hughes Network Systems (HNS) Inc., employ a forward link capacity of 400 kb/s, while using the PSTN as the return link to the Internet service provider. Telephone return is attractive because many countries in which these systems are currently implemented prohibit individual, unmonitored broadcasting of data on their airwaves. As a result, private VSATs cannot transmit to satellites directly.

The major disadvantage of such systems, however, is that they depend completely on the PSTN to transport their traffic. This means that the system is dependent on the quality and availability of the phone network. The phone line may not be accessible to voice while being used for data.

CONCLUSION

In this article we have shown that existing MAC protocols originally designed for HFC systems can be used successfully to support Web traffic over a two-way satellite channel using a VSAT terminal. The asymmetrical nature of Web traffic can be matched to the channel by designing for an asymmetrical satellite configuration.

The large delay of the satellite link limits the overall throughput of the system, and decreases the utilization of the physical channel. To partially mitigate this problem, some changes were made to the basic protocol. Grouping and piggybacking of requests were the most important changes made. By employing such changes, we gain a delay improvement of 6 s by grouping and of 13 s by piggybacking for 40 stations. When combined, we gain a delay improvement as high as 19 s. In addition, this article has investigated the changes in performance as the frequency of the bitmap transmission was varied. For practical reasons this frequency should be low.

Our simulation results show that for a satellite channel the whole page delay is about 30 s with 20 stations. To put this figure in perspective, clients surfing the Web over a fast communication channel experience about a 12 s whole page delay. For a user on a dial-up modem the delay is significantly greater. When downloading larger objects the whole page delay for the satellite link may be shorter than for the dial-up link because of the higher forward link capacity.

Finally, the article has compared the performance of the CPR, after adding the previous improvements, to slotted Aloha in terms of utilization and whole page delay. Simulation results have shown that our protocol outperforms slotted Aloha in both measures. For 40 stations our protocol reduces whole page delay from 140 s to 40 s over the best-performing slotted Aloha.

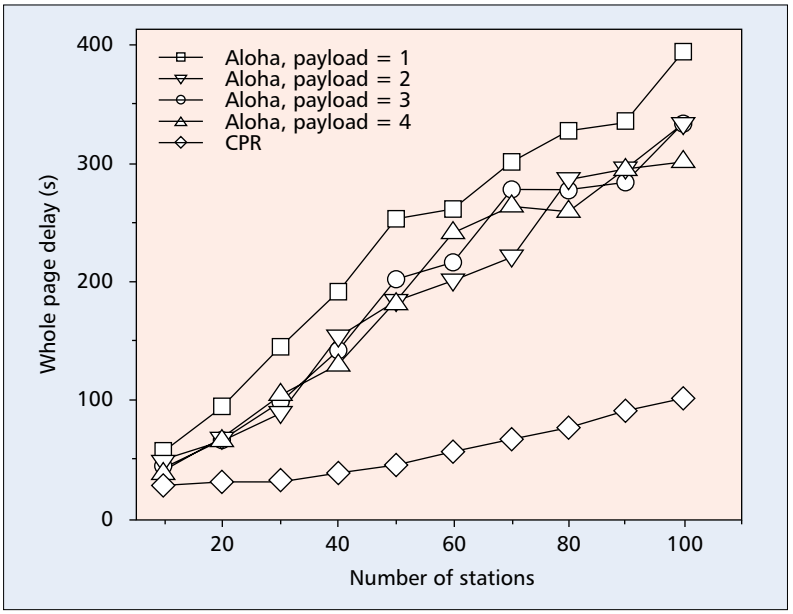

Figure 9. *Delay comparison between slotted ALOHAs and CPR (with grouping and piggybacking).*

REFERENCES

[1] R. Fielding *et al.*, "Hypertext Transfer Protocol – HTTP/1.1," Internet RFC 2616, June 1999.
[2] K. Thompson, G. J. Miller, and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," *IEEE Network*, Nov 1997.
[3] N Golmie *et al.*, "A MAC Protocol for HFC Networks; Design Issues and Performance Evaluations," *Comp. Commun.*, vol. 20, 1997, pp. 1042–50.
[4] J. O. Limb and D. Sala, "A Protocol for Efficient Transfer of Data over Hybrid Fiber/coax Systems," *IEEE/ACM Trans. Net.*, Dec. 1997, pp. 872–81.
[5] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.
[6] H. Peyravi, "Medium Access Control Protocol Performance in Satellite Communications," *IEEE Commun. Mag.*, vol. 37, no. 3, Mar. 1999, pp. 62–71.
[7] H.-K. Choi and John O. Limb, "A Transient-State Model of TCP," in preparation.
[8] R. L. Rivest, "Network Control by Bayesian Broadcast," *IEEE Trans. Info. Theory*, vol. IT-33, no. 3, May. 1987, pp. 323–28.
[9] V. Jacobson, C. Leres, and S. McCanne, Tcpdump Manual and Software.
[10] H.-K. Choi and John O. Limb, "A Behavioral Model of Web Traffic," *Proc. IEEE ICNP '99*, Toronto, Canada, Oct. 1999.
[11] H. F. Neilson *et al.*, "Network Performance Effects of HTTP/1.1, CSS1, and PNG," *Proc. SIGCOMM '97*, Cannes, France, Sept. 1997.
[12] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic, Evidence and Possible Cause," *Proc. ACM SIGMETRICS '96*, Philadelphia, PA, Apr, 1996.
[13] J. Postel, "Transmission Control Protocol," RFC 793, 1981.

BIOGRAPHIES

HYOUNG-KEE CHOI (hkchoi@ee.gatech.edu) received his B.A. degree in electronic engineering from Sung Kyun Kwan University, Seoul, Korea, in 1992 and his M.S. degree in electrical engineering from Polytechnic University, New York, in 1996. He is currently working toward a Ph.D. degree at Georgia Institute of Technology, Atlanta. His research interests include Web traffic characterization and modeling, Internet access technologies, and server load-balancing technologies.

OSAMA QADAN (osama.qadan@astrolink.com) received his B.A. degree in electrical engineering from Jordan University of Science and Technology in 1996, and his Master's degree in electrical engineering from Georgia Institute of

Existing MAC protocols originally designed for HFC systems can be used successfully to support Web traffic over a two-way satellite channel using a VSAT terminal.

Technology in 1997. Since that time, he has held several positions in a number of satellite companies including Intelsat and Astrolink International LLC., where he worked on several topics related to network systems development and network management. While working at Astrolink, he is also pursuing his Master's in business administration in John Hopkins University.

JOHN O. LIMB (limb@broadcom.com) received his Ph.D. in electrical engineering from the University of Western Australia in 1967. He joined Bell Laboratories, Holmdel, New Jersey, in 1967 and became department manager, Visual Communications Research Department in 1971. He worked for a number of years on the coding of color and monochrome picture signals to reduce channel capacity requirements and has published widely in this area. He also worked and published in the areas of visual perception, office information systems, and local/metropolitan are networks. In 1984 he joined Bell Communication Research and in 1986 he was appointed director of the Networks and Communications Laboratory at Hewlett-Packard Laboratories, Bristol, England. In June 1989 he returned to the United States with Hewlett-Packard as lab manager, technology analysis, Cupertino, California. In 1992 he returned to HP Labs as lab director, Media Technology Lab. In July 1994 he joined Georgia Institute of Technology to accept the chair in Advanced Telecommunications in the College of Computing. Together with Daniel Howard he formed the Broadband Telecommunications Center at Georgia Tech in December 1995. In 1998 he co-founded the company Digital Furnace Inc. which was acquired by Broadcom in March 2000. He now works for Broadcom. He is a past Editor-in-Chief of *IEEE Transactions on Communications* and *IEEE Journal on Selected Areas in Communications*.

DOLORS SALA (dolors@broadcom.com) received a B.S. degree in computer science from the Universitat Autonoma de Barcelona (UAB), Spain, in 1990, and M.S. and Ph.D. degrees from the Electrical and Computer Engineering School at Georgia Institute of Technology, Atlanta, in 1995 and 1998, respectively. Her research focused on MAC protocols for hybrid fiber coaxial systems. She taught undergraduate courses at UAB from 1990 to 1993. She worked on several industry projects in the area of protocols for cable systems as a member of the Broadband Telecommunications Center at Georgia Tech. After her graduate studies, she joined Motorola in Massachusetts where she worked for more than a year on supporting quality service in ATM networks. She joined Digital Furnace Corporation in June 1999 to lead the design of a high-performance MAC protocol for cable systems. She is currently at Broadcom Corporation since Broadcom acquired Digital Furnace in March 2000.

JEFF MEYERS (jeff.meyers@echo.star) received a degree in electrical engineering from Louisiana State University in 1977. He has spent his career developing modem products and systems in the digital communication area. These include terrestrial, microwave radio, and satellite modems. He developed a 16 kb/S 64-QAM secure voice terminal at Harris for the Air Force, and a digital audio distribution satellite system for the major networks at Scientific Atlanta. He also cofounded Digital Transmission Systems where he developed a 256-QAM microwave radio modem for MCI and a DS3/1 cross-connect switch. He then developed the variable-rate modems required for the Scientific Atlanta DAMA voice system. Later he cofounder Media 4 where he developed a two-way Internet-over-satellite terminal. Now at Echostar, his interests are in the area of one-way package delivery.