Double Feedback Streaming Agent for Real-time Delivery of Media over 3G Wireless Networks

Gene Cheung, Wai-Tian Tan   Takeshi Yoshimura
Hewlett-Packard Laboratories   NTT DoCoMo, Inc.

***Abstract*— A network agent located at the junction of wired and wireless networks can provide additional feedback information to streaming servers to supplement feedback from clients. Specifically, it has been shown that feedbacks from the network agent have lower latency, and can be use in conjunction with client feedbacks to effect proper congestion control. In this work, we propose the double feedback streaming agent (DFSA) which further allows the detection of discrepancies in the transmission constraints of the wired and wireless networks. By working together with the streaming server and client, DFSA reduce overall packet losses by exploiting the excess capacity of the path with more capacity. We show how DFSA can be used to support three modes of operation tailored for different delay requirements of streaming applications. Simulation results show noticeable improvement of media quality using DFSA over existing streaming systems.**

I. Introduction

This paper considers media streaming in next generation 3G wireless networks [1] where streaming is from servers located in a wired network to mobile clients via last-hop wireless links. We consider an architecture with network agents at the intersection of wired network and wireless links. The use of such agents has been proposed earlier to support end-to-end wired network congestion control [2], application-level optimizations [3] [4] and to identify differences in available wired and wireless bandwidths [5]. In this paper, we provide a generalized agent architecture that subsumes the forementioned streaming agent work, and show how it can be used under different client delay requirements.

A. End-to-end Approach

Conventional practice for streaming media ignores the particularities of the last hop wireless link and employs endpoint media adaptations based solely on observable endpoint statistics. Since endpoint statistics are aggregated across all wired and wireless links, it is impossible to distinguish the respective conditions of the links.

This causes problems for the following reason. If losses are due to wired network congestion, the server should reduce its sending rate. If losses are due to wireless link failure, the server should increase the error resiliency of the stream but not reduce sending rate. By being unable to distinguish the type of loss, the proper action cannot be determined, resulting in decrease in performance.

B. RTP Monitoring Agent: Statistical Feedbacks

For proper congestion control, the use of *RTP monitoring agent* is proposed in [2]. It is a network agent placed at the intersection of wired network and wireless link that monitors existing streaming flows and periodically sends statistical feedbacks in the form of RTCP reports back to the senders of the flows. Let R_1^* be the permissible bandwidth of the wired network as determined by a standard wired network congestion control [6]. Let R_2^* be the maximum sending rate permissible for the wireless link, as determined by the base-station during wireless link resource allocation phase of the connection setup. The goal of the RTP monitoring agent is to provide feedback so that the source can determine min(R_1^*, R_2^*). This is achieved by placing a *shaping point* in front of the agent, that "adjusts the outgoing rate of all packet traffic to the rate of the radio link" [2]. As a result, when $R_1^* < R_2^*$, the shaping point does nothing, and the streaming server sends at rate R_1^* by virtue of wired congestion control. When $R_1^* > R_2^*$, the shaping point drops enough packets to trigger the server to transmit at rate R_2^*, as illustrated in Figure 1.

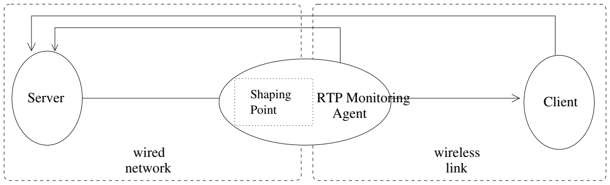

Fig. 1. RTP Monitoring Agent

C. Streaming Agent 1 (SA1): Timely Feedbacks

Because RTCP reports are sent in mid-term (on the order of seconds to minutes) and do not contain information unique to individual packets, it is argued in [3], [4] that they are neither timely nor specific enough for many application-level optimizations. One possible enhancement to RTP monitoring agent is to include *timely feedbacks* sent in short-term (within a second), with each feedback packet containing information unique to the most recent K packets in a stream. This enhanced agent is called streaming agent version 1 (SA1). Possible gains of using SA1 has been shown in [3], [4].

D. Streaming Agent 2 (SA2): Rate-mismatch

In [5], a new agent proposal expands the capability of the streaming agent in two ways. First, it provides additional information to the streaming server prior to the start of the streaming session so that the maximum allowed transmission rate at both the wired and wireless parts of the network can be determined. Second, the agent actively acts as a

0-7803-7700-1/03/$17.00 (C) 2003 IEEE

relay between the streaming server and client to exploit any additional available bandwidth to reduce overall perceived distortion. Specifically, in the typical case when the available wired bandwidth, R_1^*, is higher than the available wireless bandwidth, R_2^*, the agent can coordinate with the streaming server to use the excess wired bandwidth of $R_1^* - R_2^*$ to reduce the effective packet loss rate of the wired network using application-level retransmission at the server. On the other hand, when $R_2^* > R_1^*$, the excess wireless bandwidth of $R_2^* - R_1^*$ is used to reduce the effective wireless loss rate by applying forward error correction (FEC) at SA2. Since SA2 exploits spare capacity to reduce the effective loss rate of either the wired network or wireless link, it reduces the overall experienced packet loss.

E. Summary and Limitation of Previous Approaches

To summarize the previous agent proposals, RTP monitoring agent sends statistical feedbacks to the server to perform proper wired network congestion control, SA1 sends timely feedbacks to the server to perform application-level optimizations such as format-adaptation or application-level retransmission, and SA2 exploits the rate-mismatch nature of the wired network / wireless link environment by using surplus bandwidth for retransmission in the wired network or FEC in the wireless link.

The latest agent proposal, SA2, combining the features of RTP monitoring agent and SA1, works well under a particular set of conditions: when R_2^* is known *a priori* at SA2 and unchanging over time, and when the client playback buffer is sufficiently large to tolerate wired network retransmissions but not large enough to tolerate end-to-end retransmissions through the wireless link. It is clear that SA2 is not optimal for all streaming applications, where the client playback buffer — and hence tolerance for end-to-end transmission delay jitter — can range from very small to very large. For example, very small client buffer may not tolerate even one application-level retransmission of packets. Motivated by the limitations of SA2, the proposal in this paper is a generalization of SA2 to handle a range of streaming applications for a set of more relaxed conditions.

The outline of the paper is as follows. We first position our research relative to other research efforts in wireless media streaming in section II. We then discuss the relevant characteristics of current 3G wireless networks in section III. We then give an overview of the design of the proposed agent, double feedback streaming agent (DFSA), in section IV. We then discuss the three modes of operations of DFSA in section V. Results and Conclusion are presented in section VI and VII.

II. Other Related Work

The most related literature to our agent-based approach to wireless streaming is [7], where a gateway situated at the wired / wireless boundary decodes and then encodes FEC so that packet-level FEC can be used for the wired network and byte-level FEC can be used for the wireless link. Our streaming scenario differs from [7] in that we assume the wireless link suffers from packet drops instead of bit errors.

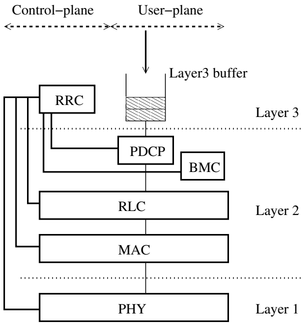

Fig. 2. IMT-2000 Protocol Stack

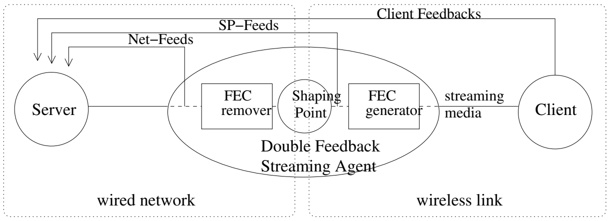

Fig. 3. Overview of DFSA

[8] proposes a gateway at the wired / wireless boundary that caches media packets and intelligently requests packet retransmission from the streaming server after performing application-level rate-distortion optimization. In contrast, our agent-based approach does not peek into the payload of the media stream and hence has much lower complexity overhead. We also avoid security issues since payload can be encrypted without affecting operation correctness.

III. Features of 3G Wireless Networks

In Today's 3G wireless network, typical one-way delay of radio links is quite large — on the order of 100ms — without link layer retransmission [9]. If link-retransmission is performed, as often required due to the unreliable nature of the wireless link, then the difference in delay between SA timely feedbacks and client timely feedbacks is even more significant.

Also unique to the 3G wireless network is the different available transmission modes at the link layer [1]. As an IP packet is passed down from layer 3 to layer 2, the radio link control protocol (RLC) provides segmentation and retransmission services. See Figure 2 for an illustration. Whether and how many retransmissions are done depends on the RLC modes. There are three modes: transparent, unacknowledged and acknowledged. For acknowledged mode, automatic repeat request (ARQ) is used for error control. The tradeoff between link quality and link delay can be done by adjusting the number of retransmissions. This parameter is set by Radio Resource Control (RRC) during configuration.

Symbol	Meaning
l_1	packet loss rate in the wired network
R_1^*	maximum permissible sending rate in wired network
R_1	initial transmission rate at server
R_1'	receiving rate at DFSA
$R_1^{(2)}$	rate inside DFSA after FEC removal
$R_1^{(3)}$	rate inside DFSA after Shaping Point
l_2	packet loss rate in the wireless link
R_2^*	maximum permissible sending rate in wireless link
R_2	initial transmission rate at DFSA
R_2'	receiving rate at client
r	streaming media coding rate

Fig. 4. Definitions for DFSA Media Streaming System

IV. Overview of Double Feedback Streaming Agent

Our proposed network agent is called double feedback streaming agent (DFSA). We assume R_2^*, though possibly time-varying, can be determined at DFSA using one of two ways depending on setup. First, it is determined by the wireless infrastructure during the wireless link resource allocation phase of the wireless session setup. In such case, R_2^* is not likely to vary unless the session has been cut off and needs to be re-established or the user moves to a different base station. Second, it is determined by simply observing the fullness of the outgoing network queue of IP packets at the base station (see Figure 2) waiting to be fragmented and transported in lower layers. R_2^* will likely be time-varying in this case.

We denote the initial transmission rate at the server in the wired network and at DFSA before the wireless link by R_1 and R_2, respectively. Rates R_1 and R_2 include all packet related to the media stream, such as retransmitted packets and FEC packets. Due to possible packet drops in the wired network and wireless link, the respective rate received at the end of the wired network and wireless link, R_1' and R_2', may be smaller: $R_1' \leq R_1$ and $R_2' \leq R_2$. Finally, we define the actual streaming coding rate to be r. See Figure 4 for a complete list of definitions.

DFSA essentially is divided into three parts: FEC decoder & encoder, shaping point and double feedback generator. We now discuss them in order.

A. FEC Decoder and Encoder

When packets first enter DFSA, a FEC decoder first channel decodes wired network FEC if present. This possibly lowers the receiving rate at DFSA R_1' to $R_1^{(2)}$. We assume FEC used is systematic — i.e. the generated FEC is the original data packets plus parity packets, and the parity packets are sent as a separate stream [10]. FEC decoder tries to reconstruct the missing original data packets using the parity packets.

Before packets leave DFSA, a FEC encoder checks $R_1^{(3)}$, the rate inside DFSA after Shaping Point (to be discussed), against the maximum rate R_2^*. If $R_1^{(3)} < R_2^*$, FEC encoder uses wireless link bandwidth surplus $R_2^* - R_1^{(3)}$ for FEC for protection on the wireless link. The end result is that the initial transmission rate in the wireless link R_2 is always approximately equal to R_2^*.

B. DFSA Shaping Point

Similar to [2], a shaping point is placed in the middle of DFSA to pro-actively drop packets should the rate inside DFSA after FEC removal, $R_1^{(2)}$, exceeds the wireless link bandwidth R_2^*. Should duplicate packets with identical RTP sequence numbers exist, they are detected and eliminated. The end result is that any loss between FEC encoder and the client is the result of wireless link failure only, not queue overflow.

C. Double Feedback Generator

Two feedback generators are located just before the FEC decoder and FEC encoder to provide feedbacks to the server. The pre-shaping point feedback generator sends *Net-Feeds* to the server, informing the server of the current wired network condition. Net-Feeds are statistical feedbacks containing summary of information collected over a window of packets, such as packet loss rates and mean and variance of round trip time (RTT) sent in the mid term (in the order of seconds). An example of such statistical feedbacks is RTCP reports [11]. The post-shaping point feedback mechanism sends *SP-Feeds* to the server, in the form of packet acknowledgment packets (ACKs), to the server so that it can determine what packets are dropped prior to wireless transmission. These are sent in the short term (within a second). See Figure 3 for an illustration.

Besides Net-Feeds and SP-Feeds provided by DFSA, we assume the client also provide fine-grained timely feedbacks to the server.

D. Deriving Network Parameters

The various feedbacks provide the server with the following information. First, Net-Feeds to the server allow the server to determine the maximum permissible sending rate in the wired network, R_1^*. The wired network loss rate l_1 is explicit in Net-Feeds. Second, because the shaping point allows no more outgoing packets than R_2^*, the server can obtain R_2^* as the sum of SP-Feeds plus the parity part of the systematic code of the client feedback.

Third, SP-Feeds together with the client feedbacks provide enough information for the server to deduce the actual packet loss rate. By comparing SP-Feeds and the data packet part of the client feedbacks, the server can deduce the loss rate of the wireless link *after* FEC has been applied. Now looking at the parity packet part of the client feedbacks, the server can deduce the amount of FEC applied. Using these two pieces of information, the server can deduce the wireless channel loss rate l_2. With the deduced network parameters, the server can then use DFSA in one of three modes of operation to optimize streaming quality as detailed next.

V. DFSA Mode of Operations

Unlike SA2, DFSA is designed so that the network agent can transparently operate in one of three different modes of operation depending on application's delay requirement, which in turn depends on the client buffer size. We assume each application can determine *a priori* what delay requirement is suitable given the client's operational limitations. We now discuss the three modes of operation in turn.

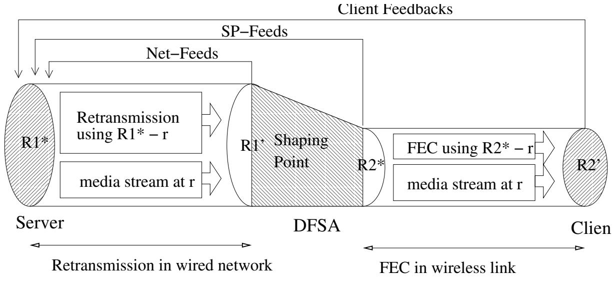

Fig. 5. Mode II: $R_1^* > R_2^*$

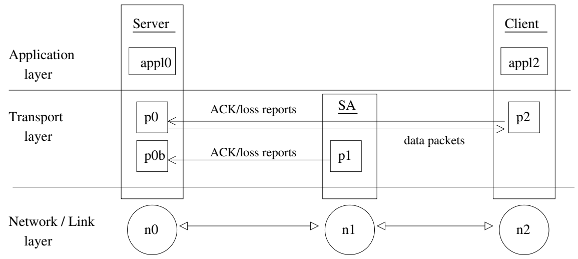

Fig. 6. *Simulation Setup.*

A. Mode I: Large Client Playback Buffer (ARQ/Ack)

When client playback buffer is large, we setup the DFSA system as follows. First, wireless link is configured with acknowledgment mode with a large maximum number of link-layer retransmissions. This results in a near lossless link with a large delay variation. Second, Net-Feeds from DFSA is used for wired network congestion control at the server. Third, SP-Feeds from DFSA is used for application-level retransmission [4]. Since the wireless link is almost lossless, the DFSA feedbacks mimics the client state nearly perfectly. Moreover, SP-Feeds would arrive at server much faster than client feedbacks. Since delay jitter is not a concern, we choose retransmissions over FEC in the wired network to achieve higher efficiency in bandwidth usage.

B. Mode II: Mid Client Playback Buffer (ARQ/FEC)

Mode II is the case where the buffer is sufficiently large to tolerate several retransmissions of packets in the wired network, but not retransmissions in the wireless link. In this scenario, the DFSA operates as follows.

Because the wireless link is typically much more rate-constrained that the wired network, we assume the case where $R_1^* > R_2^*$. As discussed earlier, the server can deduce R_2^*, R_1^* and l_2 using the various feedbacks. This allows a server to choose a media coding rate r so that FEC at channel coding rate $R_2^* - r$ is sufficient to combat the wireless loss rate l_2. The surplus bandwidth in the wired network of $R_1^* - r$ is exploited using application-level retransmissions. Since duplicate packets are dropped at the shaping point, possible duplicates resulted from unnecessary retransmissions will not overwhelm the wireless link. See Figure 5 for an illustration.

C. Mode III: Small Client Playback Buffer (FEC/FEC)

Here, we consider the case when the client buffer is very small so that it does not tolerate any retransmission in any part of the network. In such case, we first use FEC in the wired network that is tailored for the loss characteristics of the wired network. This FEC layer is removed at DFSA by the FEC remover. Then a different type of FEC is added by FEC generator that is tailored for the wireless link. We term this FEC conversion *FEC Transcoding*.

Since FEC is often used with some amount of interleaving, there is an inherent delay associated with this mode. For applications in which even such delays are inappropriate, DFSA may not be useful. In such scenario, additional techniques that improve source characteristics such as error resilient source coding can be used. The incorporation of such techniques is beyond the scope of this paper.

VI. Results

A. Simulation Setup

We performed simulations using Network Simulator 2 [12]. The setup is shown in Figure 6. It has a transport layer duplex connection (`p0`-`p2`) from the sender node `n0` to the client node `n2`, and a simplex connection (`p1`-`p0b`) from the SA node `n1` to the sender node `n0`.

In our simulation, the links `n0`-`n1` and `n1`-`n2` have constant delay and uniform loss rates. An instance of the application, `app0`, sits at sender node and sends packets to the client using the first connection. Each packet has a sequence number in the packet header. There is a filter at the link from `n1` to `n2` that sniffs out each packet targeted to the client and forwards a copy to `p1`, who sends ACKs (SP-Feeds) back to the sender using the second connection. Net-Feeds are not simulated in the experiments; we assume the streaming server can easily and properly receive Net-Feeds and correctly deduce R_1^* and l_1, which are non-time-varying in the experiments.

For real video data, we use H.263 video codec to encode the first 50 frames of the `carphone` sequence into a video stream, encoded at QCIF size, 230kps, 20frames/s and at I-frame frequency of 25 frames. The resulting average PSNR for the error-free compressed stream is 37.01dB. During the experiment, when the receiver is unable to decode a certain frame i, the most recently correctly decoded frame j is used for display for frame i, and we calculate the PSNR using original frame i and encoded frame j. If no such frame j is available, then PSNR is 0.

B. Mode I: Large Client Playback Buffer

For Mode I, the comparison is among three schemes. Scheme A is a streaming system that employs DFSA. Scheme B is a streaming system that relies on timely client feedbacks only. Scheme C is a streaming system that uses no timely client feedbacks. While all can use Acknowledgment mode to perform link-layer retransmission to reduce the wireless link packet loss rate l_2 to a negligibly small value, DFSA's Net-feeds maintain a relatively small wired network RTT instead of client feedbacks' end-to-end RTT. As a result, the deduced maximum sending rate for wired network R_1^* will be higher

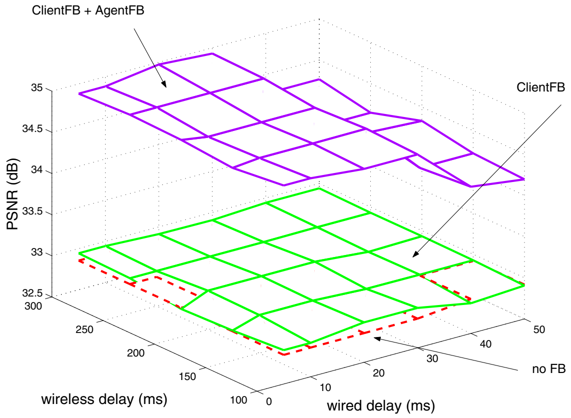

Fig. 7. *Comparison of various feedback schemes for Mode I*

wired loss	wireless loss	w/o DFSA	w/ DFSA
0.03	0.03	34.04	34.60
0.05	0.03	32.90	33.38
0.07	0.05	31.58	32.96
0.03	0.05	30.00	31.77
0.05	0.07	30.00	31.72
0.07	0.07	26.77	31.27

Fig. 8. Results for $R_1 > R_2$ Case

using conventional congestion control algorithms [6] where R_1^* is inversely proportional to RTT. Moreover, SP-feeds will also arrive at the server much faster than client feedbacks, allowing the server to react more quickly by retransmitting lost packets.

Fig. 7 shows the PSNR achieved by the three schemes at different wired and wireless delay when the loss rates at the wired and wireless networks are 5 and 1%, respectively. Again, we see that client feedback alone is effective over the wide range of wired delay between 0 to 50ms, and wireless delay between 100 to 300ms. In contrast, the additional use of DFSA feedback sustained a PSNR improvement of 1 to 2dB over the same range. As expected, the PSNR improvement decreases as the wired delay increases. Nevertheless, at a relatively high wired delay of 50 ms and low wireless wireless delay of 100 ms, a PSNR difference of 1.28dB is still maintained.

C. Mode II: Mid Client Playback Buffer

For Mode II of the experiment, we assume an 1s client buffer size. The delays on the wired network and the wireless link are 50ms and 100ms respectively.

For the experiment, we assume a 20% bandwidth excess $R_1^* - R_2^*$ is available for retransmission in the wired network. The retransmission scheme is a rate-distortion optimized data unit selection algorithm discussed in [4]. Essentially, the most beneficial frames in the rate-distortion sense are chosen for retransmission given received feedbacks from DFSA and client at optimization instance. See [4] for details.

The results between the retransmission scheme not using DFSA and using DFSA are shown in Figure 8. The PSNR improvements range from $0.48dB$ to $4.50dB$, with the large improvement taking place when the wired network is most poor and the wireless link is most clean.

D. Mode III: Small Client Playback Buffer

For Mode III, we compare two schemes: scheme A uses DFSA to perform FEC transcoding, scheme B uses FEC for the end-to-end media transmission. For the first case, we assume a (6,5) Reed-Solomon code is used for the wired network as well as the wireless link. The packet loss rate for both parts are 5%. Under these conditions, scheme A obtains an end-to-end average PSNR of $35.46dB$ compare to scheme B's $33.63dB$ — a 1.83dB improvement.

For the second case, we assume $R_1^* > R_2^*$, and hence (7,5) code is used by scheme A for the wired part while a (6,5) code is still used for the wireless link. Note that scheme B cannot take advantage of this channel mismatch without using DFSA. In this case, the performance is improved to $36.28dB$ — a $2.75dB$ improvement over scheme B.

VII. Conclusions

Previous proposals for network agents to assist streaming have neglected the potential discrepancy between the capacity of the wired and wireless parts of the delivery path. In this paper, we propose to expand the capability of previous network agents in two ways. First, DFSA provides additional feedbacks so that the server can determine and, together with DFSA, exploit the maximum allowed transmission rate at both the wired and wireless parts of the network. Second, DFSA switches flexibly among three modes of operation depending on the end-to-end delay requirement of the application, as dictated by the client buffer size. Simulation results have shown significant improvements in terms of PSNR compared to schemes that do not incorporate the second feedback.

References

[1] H. Holma and A. Toskala, Eds., *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, Wiley, 2001.

[2] T. Yoshimura et. al., "Rate and robustness control with rtp monitoring agent for mobile multimedia streaming," in *IEEE ICC 2002*, April 2002.

[3] G. Cheung and T. Yoshimura, "Streaming agent: A network proxy for media streaming in 3g wireless networks," in *Packet Video Workshop*, 2002.

[4] G. Cheung, W. t. Tan, and T. Yoshimura, "Rate-distortion optimized application-level retransmission using streaming agent for video streaming over 3g wireless network," in *ICIP*, 2002.

[5] G. Cheung, W. t. Tan, and T. Yoshimura, "Streaming agent for wired network / wireless link rate-mismatch environment," in *MMSP*, 2002.

[6] S. Floyd et. al., "Equation-based congestion control for unicast applications," in *SIGCOMM*, 2000.

[7] A. Lee et. al., "Optimal allocation of packet-level and byte-level fec in video multicasting over wired and wireless networks," in *GLOBECOM*, 2001.

[8] J. Chakareski, P. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," in *MMSP*, 2002.

[9] G. Montenegro et. al., "Long thin networks," January 2000, IETF RFC 2757.

[10] J. Rosenberg and H. Schulzrinne, "An rtp payload format for generic forward error correction," December 1999, IETF RFC 2733.

[11] H. Schulzrine et. al., "Rtp: A transport protocol for real-time application," January 1996, IETF RFC 1889.

[12] "The network simulator ns-2," June 2001, release 2.1b8a, http://www.isi.edu/nsnam/ns/.