The Acts Experiments program at the Center for Satellite and Hybrid Communication Networks

by D. Friedman, S. Gupta, C. Zhang, . A. Ephremides

CSHCN T.R. 97-9
(ISR T.R. 97-20)

THE ACTS EXPERIMENTS PROGRAM AT THE CENTER FOR SATELLITE AND HYBRID COMMUNICATION NETWORKS[1]

Daniel Friedman, Sonjai Gupta, Chuanguo Zhang and Anthony Ephremides

Center for Satellite and Hybrid Communication Networks
Institute for Systems Research
University of Maryland
College Park, MD 20742

Abstract

This paper describes experiments conducted over ACTS and the associated T1 VSAT terminal. The experiments were motivated by the commercial potential of low-cost receive-only satellite terminals that can operate in a hybrid network environment, and by the desire to demonstrate frame relay technology over satellite networks. The first experiment tested highly adaptive methods of satellite bandwidth allocation in an integrated voice-data service environment. The second involved comparison of FEC and ARQ methods of error control for satellite communication with emphasis on the advantage that a hybrid architecture provides, especially in the case of multicasts. Finally, the third experiment demonstrated hybrid access to databases through the use of Mosaic and compared the performance of internetworking protocols for interconnecting LANs via satellite. A custom unit termed Frame Relay Access Switch (FRACS) was developed by COMSAT Laboratories for these experiments; the preparation and conduct of these experiments involved a total of twenty people from the University of Maryland, the University of Colorado, and COMSAT Laboratories, from late 1992 through 1995.

1 Introduction

The Center for Satellite and Hybrid Communication Networks (CSHCN) at the University of Maryland was founded to explore and develop the commercial possibilities of satellite and combined satellite/terrestrial communication technologies. In many communication applications there is a need to transmit much information primarily in one direction between two points and much less information in the opposite direction. This is typically the case in file transfer and database services. While a two-way satellite channel may be used for such asymmetric applications, it is also possible

---

[1] This work was supported by the National Aeronautics and Space Administration under Grant No. 01526093.

to support them by means of a combination of a one-way satellite channel for the bulk information transfer and a parallel terrestrial channel for the low-bandwidth portion of the application traffic. In such a hybrid network—with parallel satellite and terrestrial channels—the satellite terminal need not have transmit capability, and thus it can be a much less expensive receive-only terminal. In addition to a cost savings, it is possible that improvement in network performance may also be achieved since the terrestrially-carried traffic does not suffer the high propagation delay incurred through satellite links.

These advantages of hybrid networks, coupled with the availability of low-cost, receive-only satellite terminals, suggest commercial potential for hybrid networks. To develop this potential, the CSHCN proposed in 1992 a series of experiments in hybrid interconnection of local area networks (LANs). The scope of the experiments was later expanded to include other facets of hybrid networking, and three experiments were ultimately devised. The first one examined dynamic allocation of satellite bandwidth in response to variations in the amount of traffic to be sent through the satellite. The second experiment investigated error-control schemes for use in both point-to-point and point-to-multipoint hybrid networks. Finally, the third experiment considered using a hybrid network architecture for remote multimedia database access and also compared the performance of some networking protocols in local area network interconnection.

Frame relay was used over the satellite link, using a prototype Frame Relay Access Switch (FRACS) developed for the CSHCN by COMSAT Laboratories. Although frame relay has been used over terrestrial networks, this was the first instance of its use over a satellite link. One of the experiments described below compared the performance of frame relay with that of the more traditional X.25 protocol.

All the experiments were based on a two-node satellite network configuration shown in Figure 1. One node was at the University of Maryland at College Park, and the other was at the University of Colorado at Boulder. The experiment equipment configuration at Maryland consisted of two Sun workstations connected through high-speed serial interfaces to the FRACS, and the FRACS was in turn connected to the ACTS T1 VSAT. The FRACS-to-VSAT connection consisted of a T1 connection for traffic, and an RS-232 connection for control messages (such as call setup and bandwidth allocation commands). The workstations ran CSHCN-developed software to implement the bandwidth allocation algorithm, the error-control schemes and the multimedia database server. SunLink Frame Relay software was used for the frame relay connections between the workstations and the FRACS.

A similar arrangement was used at the University of Colorado, except for an important difference. The Colorado T1 VSAT was not collocated with the FRACS but instead was on the premises of the National Telecommunications and Information Administration (NTIA), also in Boulder. An optical link was used for the T1 traffic connection between the University of Colorado and NTIA sites, and a modem link was used for the FRACS control messages.

The experiments were motivated by the availability and potential commercial uses of low-cost, receive-only satellite terminals, as mentioned earlier. However, the scope of the experiments was expanded to include many more facets of hybrid network operation. They were designed to support investigations into the network management challenges that will confront future commercial and military network service providers, as they try to meet market demands and mission requirements for multimedia network services.

2 Experiment 1: Dynamic Bandwidth Allocation

The purpose of the first experiment was to find effective methods to rapidly adapt the link bandwidth to fluctuating traffic levels, both for data traffic and for mixed-media traffic such as packetized voice and data. This adjustment of bandwidth was accomplished using a feature of the ACTS system, the ability to establish circuits of different bandwidths, in multiples of 64 kbit/s channels. The rationale behind the experiment is to request, and use, the minimum necessary amount of bandwidth that will permit achievement of satisfactory performance levels, and release it when not needed. Since the performance criteria are several and antagonistic to each other, there is a need for fine-tuned trade-offs amongst them.

The bandwidth allocation algorithms investigated for data traffic were a rate-based algorithm implemented in the FRACS and a threshold-based algorithm of our own design. For mixed-media (voice and data), an algorithm that generalizes the concept of thresholds and is based on so-called switch-functions was tested, both for a single station and in a setting of two stations competing for bandwidth. The FRACS was used to signal the ACTS system for requesting or releasing channels either according to its own algorithm, or upon instruction by our experiment software, for our algorithms.

The general experiment configuration is shown in Figure 2. The transmitter and the receiver may either be located at separate nodes, or they may be collocated at the same node, but on separate computers. It was important to run the transmitter and receiver software packages on separate workstations, in order to avoid any delays caused by the sharing of the computer processor. All the workstations involved in the experiment were isolated from the network to prevent unauthorized users from stealing processor time.

The software for the transmitter and the receiver was developed in-house. The transmitter software consisted of a source, a "packetizer" to append index numbers to the data packets, a "transmit-logger" to store the relevant statistics, a "transmit-buffer" for the data packets arriving from the source, a "threshold-monitor" to implement the appropriate bandwidth allocation algorithm, and a transmitter to send out the packets at the appropriate rate. The statistics stored by the "transmit-logger" included the times of packet arrivals into the "transmit-buffer", the times when bandwidth requests were placed, the times when bandwidth changes actually took place, etc.

A problem presented by the ACTS system for this experiment was the significant delay in allocating the requested bandwidth. The encountered delay for allocating bandwidth using the

ACTS system varied between 2 and 15 seconds and was not fixed, but changed as a function of the overall load on the ACTS network. To simulate a system in which the delay is either fixed and/or smaller, and hence to assess the effect of such delay on performance, a scheme was devised and tested for pre-allocating, through request from the ACTS system, all necessary satellite channels, but using only those required according to our dynamic bandwidth allocation algorithms. Under such a scenario, the amount of delay used was completely under our control.

2.1 Data Traffic

In the data-traffic portion of the experiment, the data was generated according to a 3-state Markov-modulated Poisson process (MMPP) model [1]. The MMPP source was chosen because it is better-suited for modeling LAN traffic than a simple Poisson source since it has the ability to model the time-variation of the traffic arrival rate. An MMPP is a Poisson process whose instantaneous rate varies according to a m-state Markov chain. For this experiment, m was chosen to be three.

Two bandwidth control algorithms were investigated: a rate-based algorithm implemented by the FRACS and a threshold-based algorithm executed by our software. Both algorithms were evaluated on the basis of per-packet average end-to-end delay, number of packets lost due to buffer overflows, and the average number of satellite channels used, where a channel represented a 64 kbit/s circuit. Depending on the relative importance of these parameters a suitable cost function can be devised that also includes the cost of the bandwidth used.

2.1.1 Rate-Based Algorithm

The FRACS's rate-based algorithm determined the amount of bandwidth required by measuring the arrival rate of the incoming traffic. This traffic rate was measured over 100 ms intervals and was compared with the allocated rate for the outgoing traffic. Whenever the rate of the incoming traffic exceeded the allocated bandwidth, more bandwidth was requested (in 64 kbit/s chunks). Slightly below the allocated bandwidth was a "release" bandwidth. When the rate of the incoming traffic dropped below the "release" rate, bandwidth was released.

2.1.2 Threshold-Based Algorithm

We also proposed and tested a sophisticated threshold-based algorithm to compare with the FRACS's crude rate-based algorithm. The threshold idea was motivated by theoretical analysis of the slow server problem, where for exponential servers an optimal policy was found to be of a threshold type [2, 3, 4, 5]. In this algorithm, the amount of bandwidth required was determined by measuring the amount of data queued for transmission. Whenever the queue size exceeded a specified fraction of the maximum buffer size, an additional 64 kbit/s channel was requested from the ACTS system. Slightly below each such "request" threshold was a "release" threshold, and a channel was released whenever the queue size fell below the release threshold (Figure 3). Tests were done to find the optimal difference between pairs of request and release thresholds, and also between pairs of thresholds. The “hysteresis” between up-crossing and down-crossing thresholds was motivated by the need to smoothen the effect of short-term traffic fluctuations.

2.2 Mixed-Media Traffic

In the mixed-media portion of the experiment, the traffic consisted of voice and data packets. The data packets were generated using an MMPP source as in the data-traffic portion of the experiment. The voice calls arrived according to a Poisson model, and the call durations were exponentially distributed and independent.

The bandwidth allocation algorithm for mixed-media traffic used two-dimensional “threshold” structures, known as switch functions (Figure 4). The state of the system, i.e. the number of voice calls in progress and the number of data packets in the transmit-buffer, can be represented by a point in the abstract decision space of data buffer occupancy and amount of bandwidth used for voice calls (Figure 4). Thus, a state (i, j) indicates that there are i voice calls in progress and j packets of data traffic in the queue. The object is to decide what to do when a new voice-call request arrives. The options are to reject it, admit it, or admit it at a “compressed”, reduced-bandwidth (and reduced-quality) level. As the acceptance of a new voice call may result in reducing the available bandwidth for data-service excessively, a decision must also be make whether to request more bandwidth. The purpose of the algorithm is to minimize the cost function

$$E(D) + \alpha P_B$$

where $E(D)$ is the average delay for data packets, P_B is the voice-call blocking probability, and α is a coefficient that reflects the desired relative weighting of the two cost criteria. Under a number of conditions, the optimal call admission policies have been shown to take the form of a set of switching curves in this abstract space [6]. The switch functions consist of different threshold values in both dimensions, of i (the data queue size) and of j (the number of on-going calls). To the left and below a switch-curve, an arriving call is accepted, while to the right and above that curve that call is rejected. In addition, based on the same philosophy we introduce additional switch function curves that govern additional bandwidth request and release. A separate threshold on the voice-call blocking rate is also enforced at all times. If blocking a call leads to a higher than acceptable blocking rate, additional bandwidth is requested and the call is accepted no matter where the instantaneous state lied in the decision space.

This algorithm was evaluated on the basis of per-packet average end-to-end delay, number of packets lost due to buffer overflows, average number of satellite channels used, voice call blocking frequency, and the voice call quality (degree of compression). Depending on the relative importance of these parameters, a suitable cost function can be devised that must include the cost of the bandwidth used. For example, the user may wish to assign more weight to voice call blocking probability and to end-to-end packet delays than to voice quality and average number of satellite channels used, and so on.

In the experiment setting of two stations competing for satellite bandwidth, each of the Maryland and Colorado stations transmitted to each other. A central bandwidth control algorithm for the two stations was operated at the University of Maryland; bandwidth requests from the Colorado station were transported to Maryland through the Internet. While the algorithm was operated at Maryland, bandwidth was allocated on a first-come, first-served basis, and neither station had priority for being granted requested bandwidth.

2.3 Results

This experiment has not yet been completed. However, a few early results are available and summarized below.

Each run of the experiment involved sending approximately 4 million bits of data. This data was organized in 4000 packets of 126 bytes each. It was decided to send 4000 packets, in order to get reproducible results, as determined by earlier tests. Information about the send and receive times of the data packets was stored in the transmit and receive loggers, and the data packets were destroyed once they reached the receiver. For the case of pre-allocated channels and data-traffic only, the rate-based algorithm (FRACS's algorithm) had a lower average per packet end-to-end delay, but the threshold based algorithm used a lower number of channels. Further tests are being conducted to allow a fairer comparison of the two algorithms.

Some of the results obtained are shown in Table I. The buffer size used was 100 packets. An MMPP source was used. The four values shown indicate performance corresponding to different settings of the threshold values.

The rate-based algorithm was also run, both with and without memory. The results are shown in Table II. The three values shown indicate performance for the rate-based algorithm of the FRACS for three different internal configurations.

As can be seen from these numbers, the threshold-based algorithm suffers greater delay because it uses bandwidth more economically. The FRACS algorithm tends to request additional channels earlier than the threshold-based one and thus pays a higher penalty in bandwidth usage. This phenomenon is common in any service system. The relatively large difference in delay performance is due to the fact that the offered input load rate is close to the average channel usage for the parsimonious algorithm and, thus, tends to load the system to near-capacity levels. Additional tests in progress are expected to show a much smaller discrepancy in these values.

2.4 Implementation Considerations

In order to accurately measure the end-to-end delays for data packets, it was necessary for the clocks on the transmitting and receiving computers to be synchronized to within a few milliseconds. To achieve such synchronization, and to combat natural drifting of the clocks, XNTP (Network Time Protocol, Version 3) was used [7]. The Maryland computers were synchronized to stratum 2 time servers at the University of Maryland campus, while the Colorado computer was synchronized to a stratum 1 server in Boulder, Colorado. Synchronization of Maryland and Colorado computers to within a few milliseconds was thus achieved.

In order to do real-time channel allocation, requests were forwarded to the FRACS, which sent out the appropriate signals to the ACTS system. However, the combination of FRACS signaling and the Call Manager software of the T1 VSAT was unable to handle frequent requests. When the transmitter and the receiver stations were collocated at the same node, only one outstanding request was allowed at any given point in time. Thus, parts of the experiment had to be redesigned, and their testing is under way.

When using our bandwidth allocation algorithms, it was necessary to poll the FRACS to determine when bandwidth changes occurred, since the FRACS was not capable of reporting this information on its own. This polling was conducted by a portion of the transmitter software and thus it ran on the same workstation, competing with the data-transmission functions for processor time, and thus slowing the traffic generation and transmission processes. On the other hand, no such polling was needed while running the rate-based algorithm, since the FRACS itself ran this algorithm and kept a log of bandwidth changes. However, for the purposes of a fair comparison, FRACS polling was conducted in all configurations of this experiment.

3 Experiment 2: Error Control Schemes for Satellite and Hybrid Networks

As mentioned earlier, this program of experiments was motivated by the commercial potential of hybrid networks. Information transfered using a hybrid network must be protected against errors just as in a satellite network. However, the availability in a hybrid network of a terrestrial link with less propagation delay than the satellite channel presents additional problems as well as possibilities for error control, particularly in the case of automatic-repeat-request (ARQ) schemes. The second experiment explored such possibilities. Furthermore, a satellite is an excellent means for point-to-multipoint communication. Hence this experiment also investigated ARQ error control schemes for multicast communication in hybrid networks.

This experiment was additionally motivated by the possibility of improving throughput by sending ARQ acknowledgments terrestrially instead of by satellite, thus avoiding the satellite propagation delay for the acknowledgments. Throughput might be increased yet more by retransmitting packets (as may be necessary) terrestrially instead of by satellite. Calculations showed that judiciously using the hybrid configuration could indeed increase the throughput in some cases, sometimes significantly.

The experiment is depicted in Figure 5. Data was sent from one station to another via satellite. Before transmission, an error control protocol was applied by the transmitter to protect the data as it traveled through the satellite channel. Since the ACTS channel, when used with a T1 VSAT, typically exhibits a bit error rate of 10^{-7} or less, it was necessary to inject artificially-produced noise in order to study error control schemes. An independent, identically-distributed model of bit inversion was used for the artificial noise. After corruption by artificial noise, the receiver applied the error control protocol to the to correct any errors which may have developed in the data. The data was then stored for later comparison with the original data to determine a residual bit error rate.

Both forward error correction (FEC) and ARQ error control schemes were investigated. For all the error control methods tested, a continuous source of traffic was used. The FRACS was used as an interface between workstations and VSATs. The Internet and a 14.4 kbit/s telephone modem connection were each employed as the terrestrial link for hybrid operation. The parameters of interest were throughput, end-to-end delay, and the residual error rate of the data received. Not only were the results from FEC testing compared with each other, as were the point-to-point ARQ results, but the results of these two parts were compared as well.

For FEC testing, the data was sent via satellite from one workstation at Maryland to another beside it. For point-to-point ARQ testing, a terrestrial link was sometimes required, so using two adjacent workstations would yield an unrealistically small terrestrial link propagation delay. Hence the Colorado station was used for transmitting to the Maryland station during such testing. During point-to-two point ARQ testing, a workstation at Maryland transmitted to another beside it and to one in Colorado, with a software delay used for terrestrially-carried messages from the Maryland workstation.

To make meaningful statistical inferences from the data, about 100 error events (corrupted bits) were deemed minimally necessary. Hence at least 10 million bits would have to be sent through the channel when operating the noise effects at the lowest bit error rate (10^{-5}). A more meaningful approach, which was adopted, was to send at least 10 million *information* bits from one end of the system to the other. Hence, in all testing, at least 10 million information bits were transfered.

3.1 Forward Error Correction

The forward error correction (FEC) codes selected were the BCH (15, 7) and the Golay (23, 12) code since they are simple to implement and have rates of about one-half, as it had originally been hoped to compare the performance of these with the rate one-half code used by ACTS to combat rain fading. (It was later determined we could not test ACTS's coding since we cannot control the errors produced in the channel, nor even detect the occurrence of these errors, since the coding itself corrects them.) A third code, the BCH (15, 11) code, was later added to provide a broader set of FEC choices. For comparison purposes, plain uncoded text was sent as well. (The

Reed-Solomon (127, 123) and (127, 121) codes constructed over GF(2^7) had originally also been included to protect against expected burst errors, but were abandoned when the actual non-bursty error behavior of the ACTS channel was learned from another experimenter.)

The system used for FEC experimentation is diagrammed in Figure 6. As the encoder and decoder software developed for this work would not support real-time operation for the desired ACTS channel bit rate (128 kbit/s), encoding and decoding were conducted offline. The encoded data were sent from via satellite, corrupted with i.i.d. noise and stored. Later, the data were decoded and the resulting information was compared bit-by-bit to the original data.

3.2 Automatic-Repeat-Request

Both go-back-N and selective-repeat ARQ protocols were tested in satellite-only and hybrid configurations. The go-back-N and selective-repeat protocols tested followed the logic of the REJ protocol and SREJ[2] protocol with multi-selective reject option, respectively, specified in [8, 9]. The parameters of the protocols were modified for our satellite experimentation, and some parts not integral to error control, such as call setup and termination, were not included in the software implementation since they were accomplished by other means.

In the go-back-N scheme, the transmitter sent packets continuously to the receiver, which returned an acknowledgment for each valid packet received. If a packet required retransmission, the receiver discarded all subsequently-received packets until the required packet was received. In the selective-repeat scheme, the receiver could, to a limited extent, accept packets out of order, and so could specify to the transmitter a list of packets still required.

The system used for ARQ operation is shown in Figure 7. Here, a set of data packets were produced continuously and sent over the satellite. Each packet comprised 126 bytes of data, a 2-byte sequence number, and a 2-byte cyclic redundancy check (CRC-CCITT). Upon receipt, they were corrupted with artificial noise as described in the next paragraph. After checking for errors, the receiver would generate an acknowledgment according to the ARQ protocol and send this reply to the transmitter. Testing was conducted with the acknowledgments carried over ACTS or terrestrially via either Internet or via the public switched telephone network. For cases in which acknowledgments were carried terrestrially, retransmitted packets were sent either over satellite or over the same terrestrial link (in the opposite direction). Retransmitted packets were always carried via satellite if the acknowledgments were so carried.

The window size (N) was 66 for operation in a satellite network, and 63 for operation in a hybrid network. The window size was less for hybrid operation because of the lesser propagation delay through the terrestrial link than the satellite link and because the terrestrial link had less bandwidth than the satellite link. (For such purposes, the Internet was conservatively assumed to

---

[2] These terms (REJ and SREJ) are not pure abbreviations, but are standard nomenclature in [8, 9] for the go-back-N and selective-repeat ARQ protocols, respectively.

have the same bandwidth as available through modems connected through the telephone network.) The required ARQ timer periods were calculated to be 0.688 seconds for satellite operation and 0.657 seconds for hybrid operation but these values were increased slightly to allow for software processing time.

Because Transmission Control Protocol (TCP) employs an ARQ error control protocol of its own, which would interfere with our work, this protocol could not be used when employing Internet as the terrestrial link. Hence User Datagram Protocol (UDP) was the transport protocol employed.

For each combination of ARQ protocol and network configuration (satellite or hybrid, which terrestrial link, links for acknowledgments and retransmissions), four tests were conducted. In the first test, 500 packets were sent with no noise injected at either the transmitter or receiver. The second test was identical to the first except the 200th packet was intentionally discarded at the receiver. Comparing the results of these two tests yielded the time required for retransmitting a single packet, itself a measure of interest. In the third test, 9921 packets (corresponding to slightly more than 10 million information bits) were sent with no noise injected. In the fourth test, the same number of packets were sent and i.i.d. noise was injected at both the transmitter and the receiver.

The throughput, total transmission time, and residual error rate were the primary parameters of interest in ARQ experimentation. The number of packets sent, received, and received in error on each link were also recorded.

3.3 Results

The FEC portion of the experiment has been nearly completed, while the ARQ portions are not yet complete. The residual bit error rates achieved by sending 10 million *encoded* bits through the satellite channel, including the noise effects generator, are given in Table III. These results will develop their value when compared with the results of the ARQ portion of the experiment, which is not yet complete.

One result already available from the ARQ portion of the experiment is that the 14.4 kbit/s modems introduce significant delay in transporting bits from one end of a telephone link to another. This delay, even after subtracting propagation delays, is about 140 ms in each direction. This delay is speculated to be due to compression/decompression of the data and the modems' trellis-coded modulation scheme, particularly the Viterbi decoding. The precise causes for this delay have not yet been determined. As the delay is a significant fraction of the time required to send a bit between two points via geostationary satellite, it is possible little throughput improvement over a satellite network might be achieved by using such a modem link as the terrestrial link in a hybrid network. This finding does not, however, diminish the significant cost savings achievable with the hybrid architecture instead of a pure-satellite architecture.

4 Experiment 3: Remote Database Access and Internetworking Protocol Comparison

In the third experiment, the hybrid network mechanism of data access was demonstrated and evaluated against the use of solely satellite connection or solely Internet connection. Two internetworking protocols were applied in the satellite link, namely CCITT's standard X.25 protocol and the frame relay protocol. The logic and the performance of these protocols were studied and compared. We focused on the data link layer functionalities since this is where the key differences of those two protocols reside. In addition, we studied the commonly used transport layer protocol, TCP, over the satellite link; in the process, a problem of using TCP was identified, and a solution was provided, which was to use the extended TCP (in a version that was developed by USC) [10]. Finally, and most importantly, two emulated LANs were interconnected by the satellite link, and a comparison of the performance of X.25 and frame relay used for their interconnection is being carried out as the culmination of the objectives of this experiment.

4.1 Remote Database Access via a Hybrid Network

A common communication need in many applications is to access a computer network with a personal computer for the purpose of transferring files and databases. While a telephone modem can be used to connect to the network, oftentimes the low bandwidth of such a connection is insufficient. A leased line of higher bandwidth (56 kbit/s, 1.544 Mbit/s T1, etc.) is an alternative, but an expensive one. A satellite link is also an option, but the required ground terminal can also be expensive. However, typical users of computer networks consume much more information from the network than they send to it. For such users with asymmetric bandwidth requirements the low-volume traffic to be sent to the network can be carried over a telephone modem connection instead of the satellite, while the bulk traffic from the network can be carried over a higher-bandwidth satellite channel. The transmit capability of the satellite terminal may so be eliminated, thus significantly reducing the cost of the terminal.

The first part of this experiment demonstrated this concept by remotely accessing a multimedia database using a hybrid network (Figure 8). In particular, all short interactive access requests were routed through Internet, and the bulk information from the database was routed through a T1 satellite link. ASCII text, audio and image files, ranging in size from 50 kilobytes to 6.4 megabytes, were transfered using the "point-and-click" Mosaic user interface. The performance obtained when using only Internet varied with the time of day, with throughputs as low as several kilobytes per second during peak network traffic hours up to 40 kbyte/s. Operation through the satellite exhibited much less variation, with throughput of about 39 kbyte/s, since the satellite link was a dedicated and directly-connected link. The throughput through satellite usually exceeded that through Internet, when Internet exhibited stable throughput. Table IV provides some of the specific measurement results.

The utilization of the T1 satellite link was also monitored during the satellite and hybrid tests. A utilization of 27% was achieved in the satellite test, but 41% utilization was achieved in the hybrid test. The reason for this is the acknowledgment delay of 290 ms through satellite was reduced to an average of 70 ms when an Internet terrestrial link was used, and this reduction significantly affected the window-based flow control applied by the the transmission layer of the applications.

In parallel with this work, and motivated by the same desire to support asymmetric communication with low-cost receive-only satellite terminals, the CSHCN helped Hughes Network Systems develop a recently-announced product named *DirecPC*. With this product, a user can send an information request to a computer network (in particular, Internet) using a modem and have the information returned to him via a satellite link. To accomplish this, it is necessary the computer accessed over the network return requested information to a satellite uplinking computer instead of to the computer connecting the user's dialup connection to the network. This required special software for encapsulating Internet TCP/IP frames within TCP/IP frames with suitably modified addresses so that the requested information would be routed properly. The CSHCN contribution to this product has been recognized by the University of Maryland Office of Technology Liaison as Outstanding Invention of 1994.

4.2 Protocol Comparisons

We first considered the relative merits and performance of the two basic protocols for internetworking, namely X.25 and frame relay. Following that we considered TCP and extended TCP over satellite and finally we engaged in the comparison of the performance of these protocols for interconnecting two LANs.

4.2.1 X.25 and Frame Relay

As we know, X.25 uses call-control packets for making and breaking virtual circuits that are carried on the same channel and virtual circuit as data packets. In effect, in-band signaling is used. With X.25, flow-control and error-control mechanisms are within, both, the network and the data link layers. Also, multiplexing of virtual circuits is conducted at the network layer. This approach results in considerable overhead. In contrast, frame relay is a protocol which seeks to eliminate as much of this overhead as possible, reduce network delay, increase throughput and provide more efficient bandwidth utilization. It is regarded as an improvement over the traditional packet switching technology of X.25. Frame relay uses call-control signaling which is carried on a logical connection separate from user data. Hence intermediate nodes in frame relay communication need not maintain state tables or progress messages relating to call-control on an individual per-connection basis. Also, frame relay multiplexes logical circuits at the data link layer instead of the network layer, eliminating an entire layer of processing. Frame relay has no node-by-node flow control nor error control; such functions are the responsibility of higher layers, if conducted at all. Thus frame relay offers a more streamlined approach to communication than X.25. Communication with frame relay requires less processing at each network node as well as at the user-to-network interface.

The protocol logic, operation and performance of these two protocols were studied and compared in this portion of the experiment. SunNet X.25 and SunLink Frame Relay software packages were used for this work. A T1 satellite link was used to interconnect Ethernet LANs at the University of Maryland and the University of Colorado. A workstation at each of the two sites, was configured as internetworking gateways for both X.25 and frame relay. As only two nodes were available for testing the protocols, and not a network of several, the differences in multiplexing and routing functions of the two protocols could not be tested. Hence only the differences in the data link layer could be studied by experimentation.

The FTP and TTCP were used to obtain throughput measurements. A throughput of up to 40 kbyte/s was obtained with frame relay. For X.25, the throughput was sensitive to the setting of the window size. The SunNet X.25 software allows a setting of 2 to 127, and calculation showed that using a window setting of 127 and a 1000 byte frame size should support non-stop transmission. However, the window size could not be set to higher than 18, likely due to a problem with the SunNet software. Hence only 9 kbyte/s of throughput was achieved. Assuming the window size could have been set to 127, the throughput of X.25 would be expected to still be less than that of frame relay, since X.25 requires four steps for transferring a packet: the link level sending and acknowledging, and the transmission level acknowledging sending and acknowledging. We observed this phenomenon by monitoring the link layer activity. From the foregoing discussion, frame relay can indeed be considered as an evolution of X.25, and can indeed support higher end-to-end throughput than X.25, especially on a high-quality links. Specifically for FTP traffic, Table V shows the relative performance of X.25 and frame relay in terms of delay, throughput and maximum link utilization.

4.2.2 Conventional and Extended TCP

It should be noticed from the previous section that when an FTP session was initiated over a frame relay link, even though frame relay doesn't have any link level flow control and error control which may restrict the end-to-end throughput, the actual throughput of the FTP session over a dedicated T1 link was only 39 kbyte/s, and the average link utilization was only about 24%. Responsible for this inefficiency, were the flow control and error control functions of the transport-layer TCP.

The flow control function uses a sliding window protocol similar to that of high-level data link control (HDLC) protocol, with maximum supported window size of 64 kilobytes. This poses a problem for using TCP with a satellite channel, or any other link with a high bandwidth-delay product: TCP will allow at most 64 kilobytes of unacknowledged data, while there can can easily be more than 64 kilobytes of data in the link at any instant. (In particular, 111 kilobytes of data can simultaneously be in a T1 link with a 580 ms round-trip delay). This leads to a "stop-and-wait" syndrome, in which TCP sends a chunk of data and then waits for an acknowledgment before continuing. Hence TCP prevents the link from being filled to capacity, and the throughput is reduced correspondingly. Additional concerns with using TCP over a satellite link include possibilities of sequence number wrap-around and earlier termination of the connection. These problems are addressed in an extended version of TCP, which was tested and compared with conventional TCP.

For the reasons mentioned above, a single session of conventional TCP was found to yield only 6.9 kbyte/s throughput over the satellite with a link utilization of 4%. Trying to increase the link utilization by initiating multiple TCP sessions yielded, as a best result, an average throughput of 5 kbyte/s and a link utilization of 92%, with 32 simultaneous sessions. A single session of extended TCP was found to give higher throughput and link utilization for all settings of the window size, with the best results obtained with an 80 kilobyte window size (94 kbyte/s throughput and 72% link utilization). Clearly, extended TCP performs much better than TCP over a satellite link, giving throughput more than 13 times greater than with conventional TCP. Table VI shows the quantitative differences that were measured.

4.2.3 LAN Interconnection

The ultimate objective of this experiment is to emulate two LANs interconnected over the satellite link, on which X.25 and frame relay protocols are applied as the internetworking protocol. The performance of particular applications in this interconnected system, over the two different protocols (either X.25 or frame relay), are to be compared. Specifically, the average packet delay of each individual application is the performance criterion.

The emulation of the LANs was composed of two components; the first was emulation of the background traffic of a LAN, and the second was the particular application that we are interested in. The background LAN traffic was emulated by using the so-called Self-Similar model [11, 12], that is based on recent Bellcore studies. It was found to be a more realistic traffic model for LAN traffic. The particular application messages were merged with this emulated LAN traffic and fed into the network gateway, through which they were sent over the satellite link using as internetworking protocol, either X.25 or frame relay (Figure 9).

As an extension, a two hop LAN interconnection was tested. Here, on the receiving end (now it is actually the intermediate node), another Self-Similar traffic source was built to represent the LAN traffic, and a portion of this traffic, considered as the inter-LAN traffic which is intended for the third node (ultimate destination), was merged with the incoming traffic from the first gateway. All these traffic streams were sent back through the satellite link to the first gateway, which virtually represented the third gateway. After the particular application reached the final destination, the end-to-end performance of it was logged. In this two-hop transmission over the satellite link, either the X.25 or frame relay was used as the link layer protocol.

This portion of the experiment has not yet been completed.

5 Conclusion

As of this writing, it is clear that the experiments have not yet been completed, largely due to uncontrollable factors. The ACTS VSAT suffered several problems, including incorrect installation, cable breakage, two complete failures of its transmit power amplifier and partial failures of its control computer. The weather in Colorado prevented experimentation several times by disrupting the optical link. Conducting experiments with equipment 1500 miles away has presented its own challenges, including operating computers through the unpredictable Internet. Despite these difficulties, work continues in earnest. We expect to follow up and present the complete and detailed results of these experiments at several upcoming forums in the near future.

6 Acknowledgments

This work would not have been possible without the kind and helpful support, including the loan of the T1 VSAT, of ACTS program personnel at the NASA Lewis Research Center. Further, the authors were only part of a larger team which designed, implemented and conducted the experiments. This team included: Dr. Apostolos Traganitis of the University of Crete, Greece; Timothy Kirkwood of the CSHCN; Yannis Konstantopoulos of MCI; Dr. Anil Agarwal, Moorthy Hariharan, and Anita Fitzwater of COMSAT Laboratories; Dr. Stanley Bush, Fan Huang, Gil Weiss, Neill Cameron, Wenjin Yang, and David Bailey of the University of Colorado at Boulder; Marjorie Weibul of NTIA; and Ruplu Bhattacharya, Samardh Kumar, and Anastassios Michail of the University of Maryland.

References

[1] W. Fisher and K. Meier-Hellstern, "The Markov-modulated Poisson process (MMPP) cookbook," *Performance Evaluation*, vol. 18, September 1993.

[2] W. Lin and P. R. Kumar, "Optimal control of a queueing system with two heterogeneous servers," *IEEE Trans. Automatic Conrol*, vol. 29, pp. 696–703, Aug. 1984.

[3] J. Walrand, "A note on 'Optimal control of a queueing system with two heterogeneous servers'," *Systems and Control Letters*, vol. 4, pp. 131–134, May 1984.

[4] M. Rubinovitch, "The slow server problem: A queue with stalling," *J. Appl. Prob.*, vol. 22, pp. 879–892, 1985.

[5] I. Viniotis, "Optimal control of a multiserver queueing system with heterogeneous servers," Master's thesis, University of Maryland, College Park, 1985.

[6] I. Viniotis and A. Ephremides, "On the optimal dynamic switching of voice and data in communication networks," in *Proc. Computer Networking Symposium*, pp. 8–16, April 1988.

[7] D. L. Mills, *Network Time Protocol (Version 3) Specification, Implementation and Analysis (RFC 1305).* University of Delaware, March 1992.

[8] International Standards Organization, *International Standard ISO 4335, "Information processing Systems – Data communication – High-level data link control procedures – Consolidation of elements of procedures".*

[9] International Standards Organization, *Proposed Draft International Standard ISO 7776/DAM 2, "Information processing systems – Data communication – High-level data link control procedures – Description of the X.25 LAPB-compatible DTE data link procedures – DAM 2: Multi-selective Reject Option".*

[10] R. B. V. Jacobson and D. Borman, *TCP Extensions for High Performance (RFC 1323)*, May 1992.

[11] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," in *Proc. ACM Sigcomm '93*, pp. 183–193, 1993.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, 1994.

Table I: Threshold based algorithm results.

Delay (seconds)	Average Channel Usage	Offered Load Rate (kbit/s)
0.837	1.82	58
0.883	1.81	62
0.866	1.78	76
1.878	1.19	70

Table II: Rate based algorithm results.

Delay (seconds)	Average Channel Usage	Offered Load Rate (kbit/s)
0.346	1.98	57
0.386	2.34	82
0.414	1.98	82

Table III: Residual bit error rates.

	Code			
Noise Effects Bit Error Rate	Plain Text	BCH (15, 11)	BCH (15, 7)	Golay (23, 12)
3.16×10^{-3}	2.048×10^{-3}	1.019×10^{-4}	2.786×10^{-6}	1.112×10^{-5}
10^{-3}	8.111×10^{-4}	9.545×10^{-6}	2.143×10^{-7}	1.342×10^{-6}
3.16×10^{-4}	2.957×10^{-4}	2.727×10^{-7}	0	0
10^{-4}	9.260×10^{-5}	0	0	0
3.16×10^{-5}	3.250×10^{-5}	0	0	0
10^{-5}	1.010×10^{-5}	0	0	0
0	0	0	0	0

Table IV: Access results.

Internet:

Database Size (bytes)	Delay (seconds)	Throughput (kbyte/s)
5,299,770	170	30
5,299,770	132	40
3,311,616	240	14
103,488	9.5	11

Satellite:

Database Size (bytes)	Delay (seconds)	Throughput (kbyte/s)	Average Link Utilization (% of T1)
103,488	6.5	16	13.75
206,976	9.1	22	16
413,952	14	29	20.72
827,904	24	34	19.91
1,655,808	44	37	(to be calculated)
3,311,616	84	38	
6,623,232	170	39	

Hybrid:

Database Size (bytes)	Delay (seconds)	Throughput (kbyte/s)
103,488	4.2	24
103,488	4.1	24
206,976	5.9	34
206,976	6.3	32
413952	9.2	44
413,952	8.8	46
827,904	15	53
827,904	16	50
1,655,808	90	18
1,655,808	33	49
1,655,808	29	56
1,655,808	28	57

Table V: X.25 vs. Frame Relay

Performance of FTP over Frame Relay

Size (bytes)	Delay (seconds)	Throughput (kbyte/s)	Maximum Link Utilization
50K	5.2	9.7	15%
100K	6.5	16	23%
200K	9.1	22	27%
400K	14	29	27%
800K	24	34	27%
1.6M	45	36	28%
3.2M	84	38	28%
6.4M	160	39	28%

Performance of FTP over X.25

Size (bytes)	Delay (seconds)	Throughput (kbyte/s)	Maximum Link Utilization
50K	6.9	7.3	8%
100K	24	4.3	8%
200K	35	5.8	8%
400K	49	8.3	8%
800K	96	8.4	8%
1.6M	180	8.9	8%

Table VI: Conventional TCP vs. Extended TCP

Conventional TCP

# of Sessions	Average throughput (kbyte/s)	Average Link Utilization (% of T1)
1	6.65	4
2	6.65	8
4	6.65	16
8	6.60	30
16	6.50	62
32	5.0	92

Extended TCP : Single TCP session

Window Size (kilobytes)	Throughput (kbyte/s)	Link Utilization (% of T1)
10	14	10
20	29	22
30	43	30
40	57	46
50	70	56
60	84	68
70	92	72
80	94	72

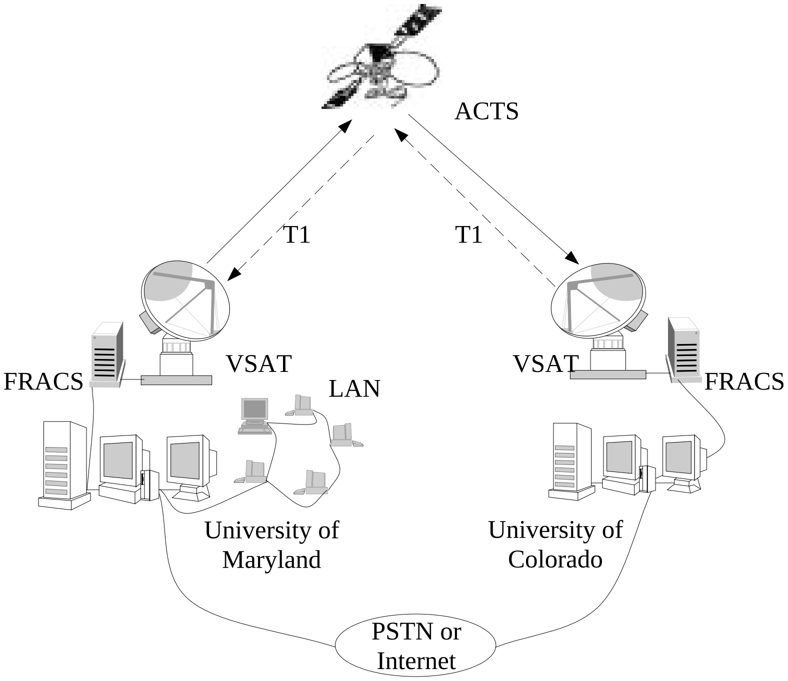

Figure 1: Configuration for ACTS experiments.

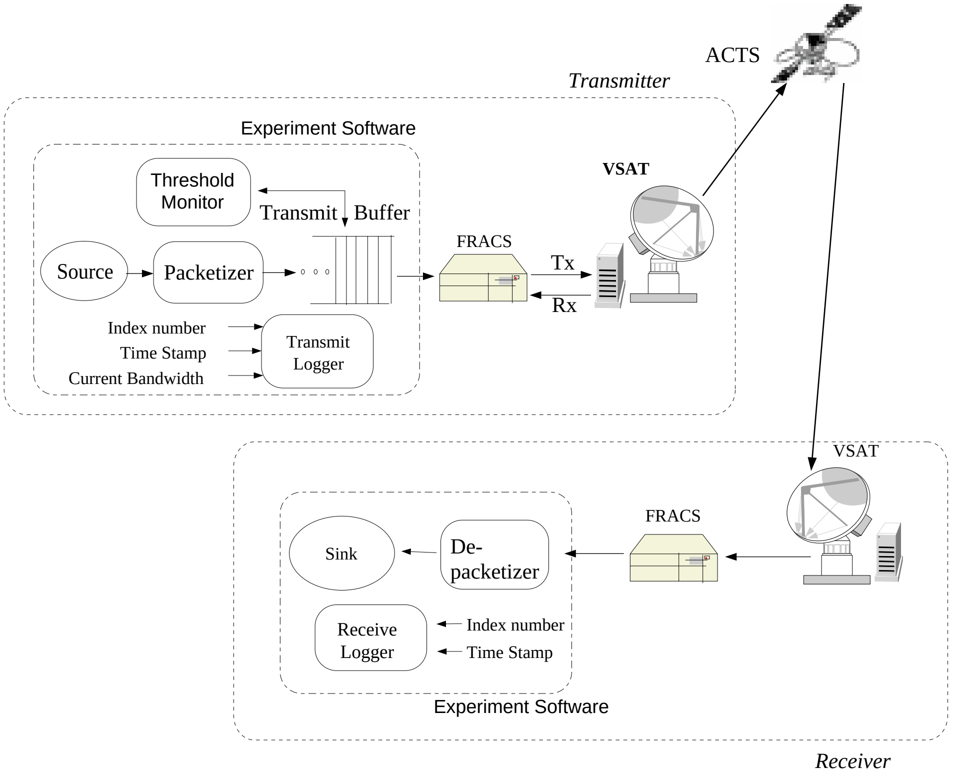

Figure 2: General Experiment 1 configuration.

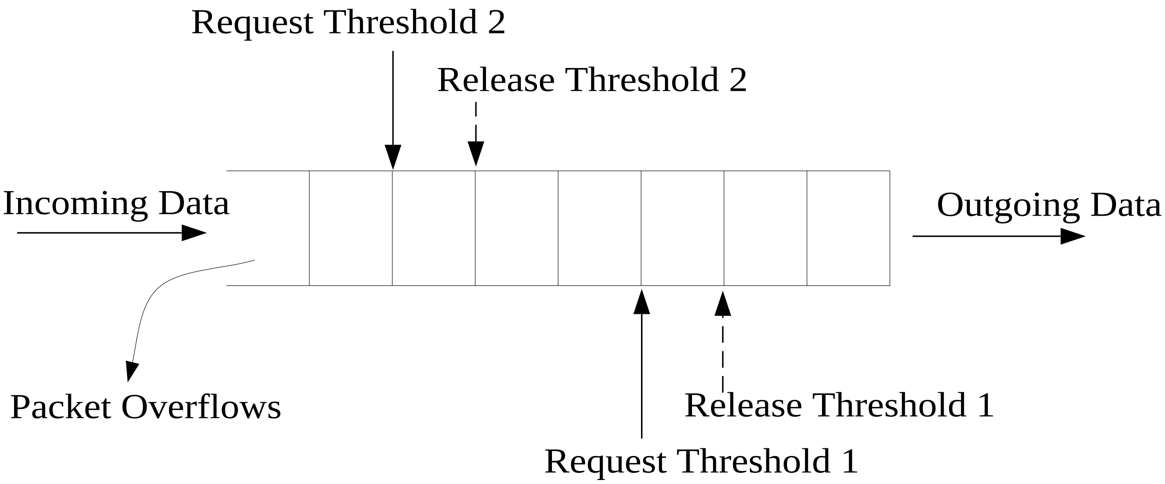

Figure 3: Thresholds on the data queue.

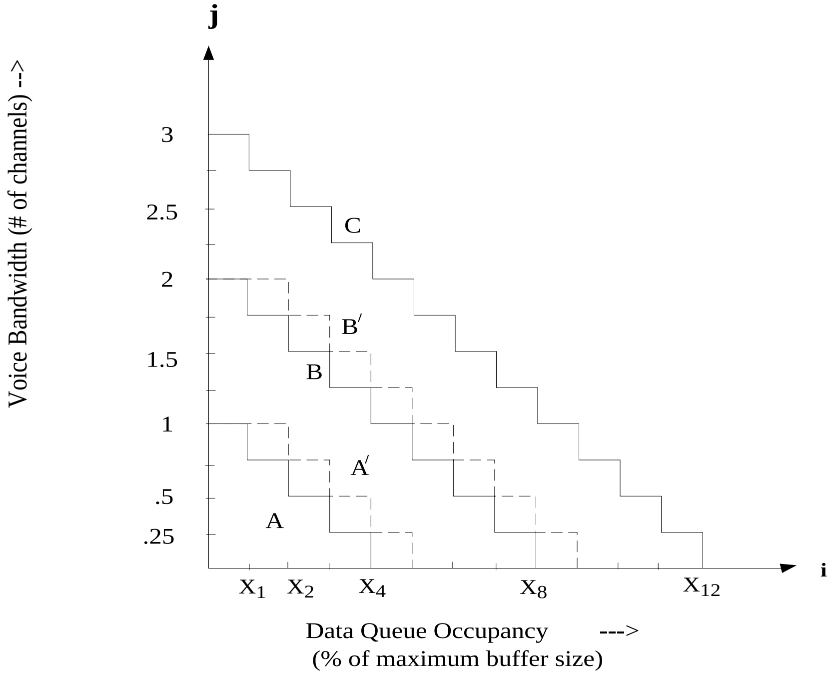

Figure 4: Switch functions for mixed-media bandwidth allocation.

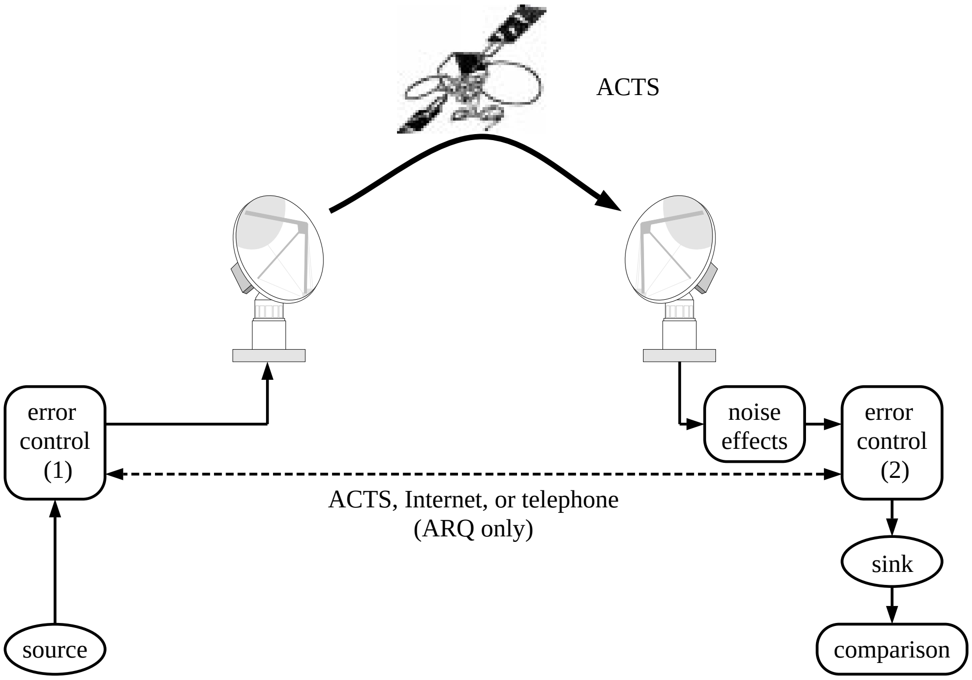

Figure 5: General Experiment 2 configuration.

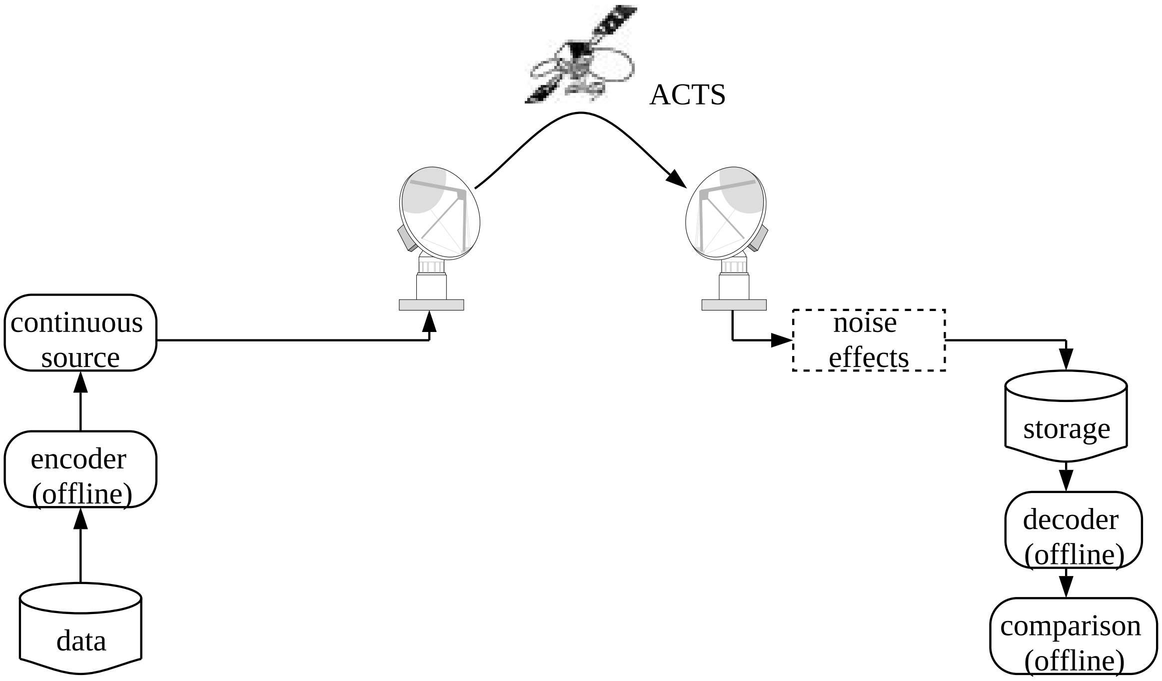

Figure 6: FEC Operation.

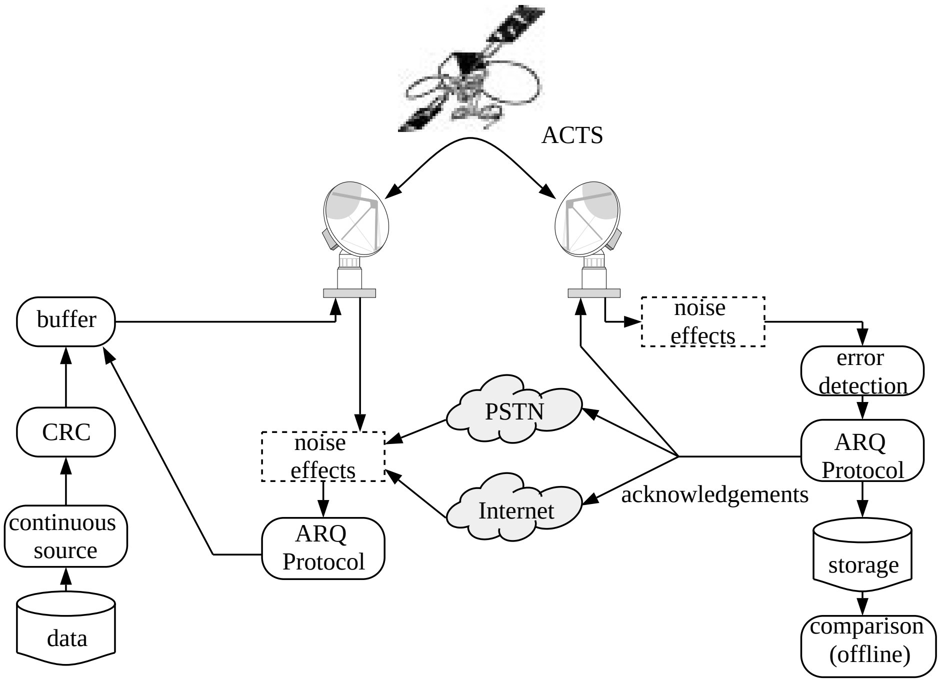

Figure 7: ARQ operation.

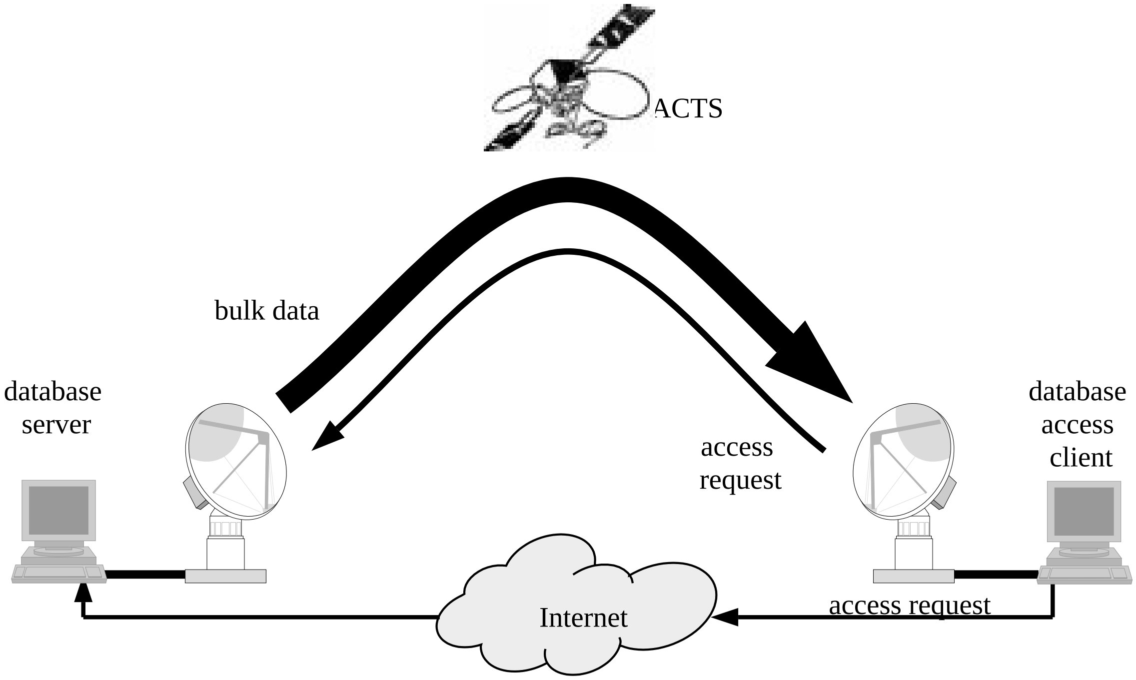

Figure 8: Remote multimedia database access with a hybrid network.

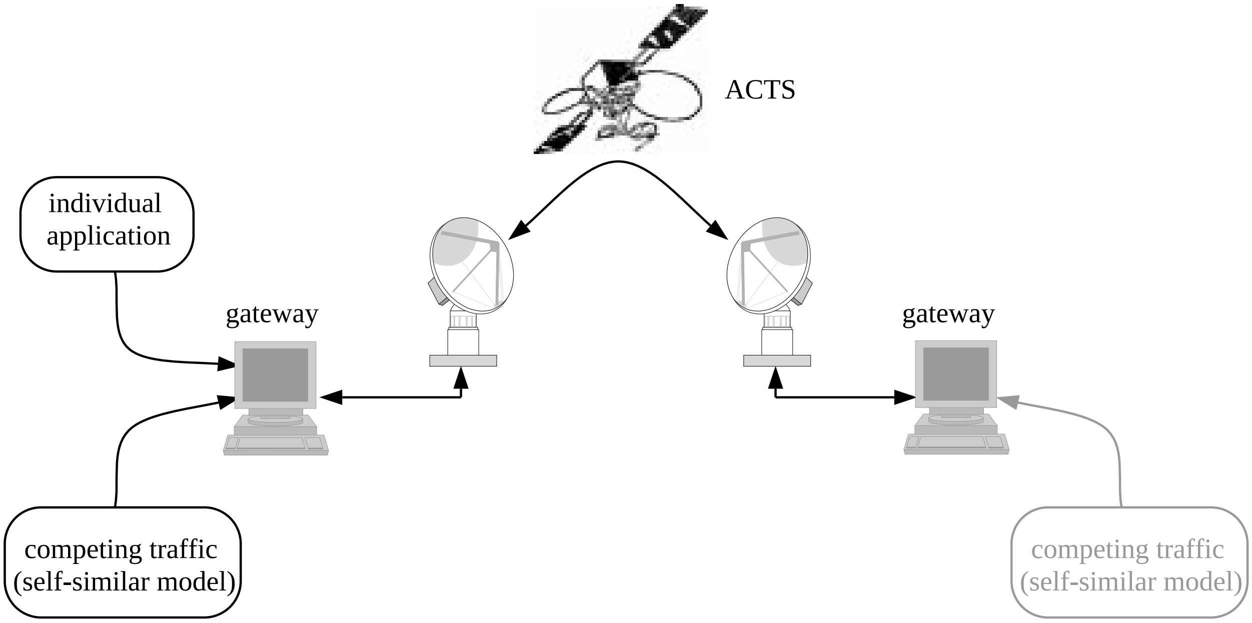

Figure 9: LAN interconnection.