A Highball Architecture for the ACTS TDMA Network

Proposal to: Defense Advanced Research Projects Agency

From: University of Delaware, Electrical Engineering Department

In response to:

Broad Agency Announcement BAA 92-36
SCALABLE NETWORKING TECHNOLOGY
Commerce Business Daily, 6 August 1992

Principal Investigator: David L. Mills, Professor
Electrical Engineering Department, University of Delaware,
(302) 831-8247, mills@udel.edu

A. Proposal Abstract

This is a proposal for a joint project with the University of Delaware and COMSAT Laboratories to support the DARPA/NASA Advanced Communications Technology Satellite (ACTS) program and to test and evaluate the Highball high speed burst-switching technology as applied to the ACTS system. It involves the design and construction of three SONET Multiplexors to serve as interfaces between computers supporting the High Performance Parallel Interface (HPPI) and the Synchronous Optical Network (SONET) earth terminal used by ACTS.

The Highball reservation-TDMA technology developed by the University with DARPA support is proposed as the basis of the scheduling algorithm which controls access to the satellite beams and regulates the data flow. This algorithm also supports multicast delivery using single and multi-hop configurations, as well as reserved-bandwidth streams.

This project is designed to operate concurrently with other ACTS experiment programs, although the development, testing and evaluation of the technology do not depend on any other programs. The three SONET Multiplexors and their workstation controllers are planned to be located at existing supercomputer centers yet to be determined. The SONET Multiplexor uses the standard HPPI technology as the primary interconnect mechanism and can even function as a HPPI crossbar switch.

The proposed test and evaluation program is to be conducted remotely in much the same fashion as the DARTNET experiment program. In fact, the proposed program draws heavily on the DARTNET experience and utilizes much of the software and operating methodology developed for that system.

B. Summary of Innovative Claims

This proposal advocates the Highball burst-switching technology for use in the ACTS experiment program. The Highball technology is designed for high speed networks where very large data sets, such as high resolution imagery or digital maps, are to be transmitted in the shortest times feasible over long distances. It is claimed that a reservation-TDMA, burst-switched technology such as Highball is best suited to such traffic. Conventional packet-switching technologies such as ATM may not be optimal in such cases, since the ATM technology is designed for aggregated traffic scenarios where the data rate of a single subscriber is usually much less than the available channel capacity.

The Highball reservation-TDMA technology requires that a reservation message be sent to all nodes prior to the transmission of a burst or stream of repeating bursts. A distributed scheduling algorithm determines from all reservations received the detailed switching schedule for each node and burst. While this incurs nontrivial delays in the case of geosynchronous satellites or long terrestrial circuits, it does provide for a deterministic, very efficient, burst schedule free from contention. This design provides the maximum transfer rate between the nodes together with efficient use of the satellite resources.

For application of the Highball technology to the ACTS experiment program, the satellite system itself takes the place of the crossbar switch. The proposed switching mechanism takes the form of a set of HPPI and SONET interfaces, configurable multiplexors and FIFOs. The multiplexor configuration can be changed quickly and synchronously at each node. The switch schedule can be changed in response to burst and stream requests and provide multiple-destination and multiple-hop connectivity when justified by bandwidth and routing requirements.

In contrast to other more conventional designs, the Highball scheduling algorithm operates in a distributed, replicated fashion. The nodes maintain robustness by snooping each other's schedules in order to detect inconsistencies. This design also provides the infrastructure to implement globally synchronized delivery of data to multiple sites and to manage the resources of the system on a global basis, rather than individually at each node, as in conventional designs.

The proposed hardware design is simple, low-risk and relatively inexpensive. The proposed software design builds on two previous projects, the Highball development program at the University of Delaware and the various DARTNET experiment programs. The previously developed Highball scheduling and reservation algorithms are directly applicable to the proposed work. The use of DARTNET technology, including existing kernel and driver software, experiment utilities and synchronization programs, provides a low-risk and cost-effective development environment.

C. Statement of Work

This is a joint proposal of the University of Delaware as prime contractor and COMSAT Laboratories as subcontractor to the University. COMSAT Laboratories is to provide a prototype and three production units of special-purpose computer peripherals called SONET Multiplexors. COMSAT will provide all hardware, diagnostic programs and control programs as required to implement these units. The detailed description of this equipment is provided in the subcontract proposal appended to this document. COMSAT will provide transportation, installation and maintenance of this equipment throughout the 30-month performance period of the subcontract.

The University will provide three conventional workstations called Node Controllers for the SONET Multiplexors, as well as the software scheduling program used by them. The cost of these workstations, including HPPI interfaces, HDLC serial synchronous interfaces, transportation and installation, is included in the budget proposal. The University will provide all workstation software, including the device drivers, scheduling program and diagnostic and testing programs. It is anticipated that the vehicle upon which this software is built is the DARTNET kernel, plus the various drivers and utility programs developed for that effort.

An experiment is proposed involving three or more ACTS earth terminals and co-located Highball switching nodes. Each node is equipped with a SONET Multiplexor, dedicated Node Controller, and interfaces to HPPI, SONET, HDLC and Ethernet. The switching nodes operate with an aggregate throughput of OC-12, include only enough buffering as required by the HPPI protocol, since in the Highball model data are buffered at the source. The experiments are to involve the Node Controllers as traffic generators, as well as other available supercomputers and related equipment on site.

The experiment is designed to test and evaluate the utility of the Highball technology operating in the ACTS system. It will involve exploration of the performance envelope under various conditions of offered load and satellite configuration. At least some experiments will involve multicast modes and multiple-hop configurations. Every effort will be made to utilize scenarios involving real data traffic and existing supercomputer applications.

It is anticipated the experiment will not require modification of the supercomputers or workstations serving as hosts. However, it may be necessary to adjust timeouts, as the reservation-transmission delays may be as much as two satellite roundtrips. In order to minimize the number of trips to transport the earth terminals from one location to another, this experiment is best conducted in conjunction with others involving existing supercomputer sites and in which the test time can be shared.

The work proposed involves a SONET interface between the SONET Multiplexor and ACTS terminal. The available SONET interface specifications do not include the details of how traffic is to be routed and multiplexed in the actual ACTS terminal. It is necessary that these details be provided in sufficient detail by month three of the performance period in order that timely development can continue.

D. Deliverables

The deliverables include the following:

1. Three Sun Microsystem IPC monochrome workstations, each with one HPPI and one HSI (HDLC) interfaces.
2. Cables and loopback connectors for all interfaces.
3. Workstation software, including the control program, scheduling program and synchronization programs to implement the Highball node.
4. A final report describing the design and operation of the system, results of experiments and lessons learned.

Deliverables required by the subcontract to COMSAT Laboratories and considered deliverables under this contract

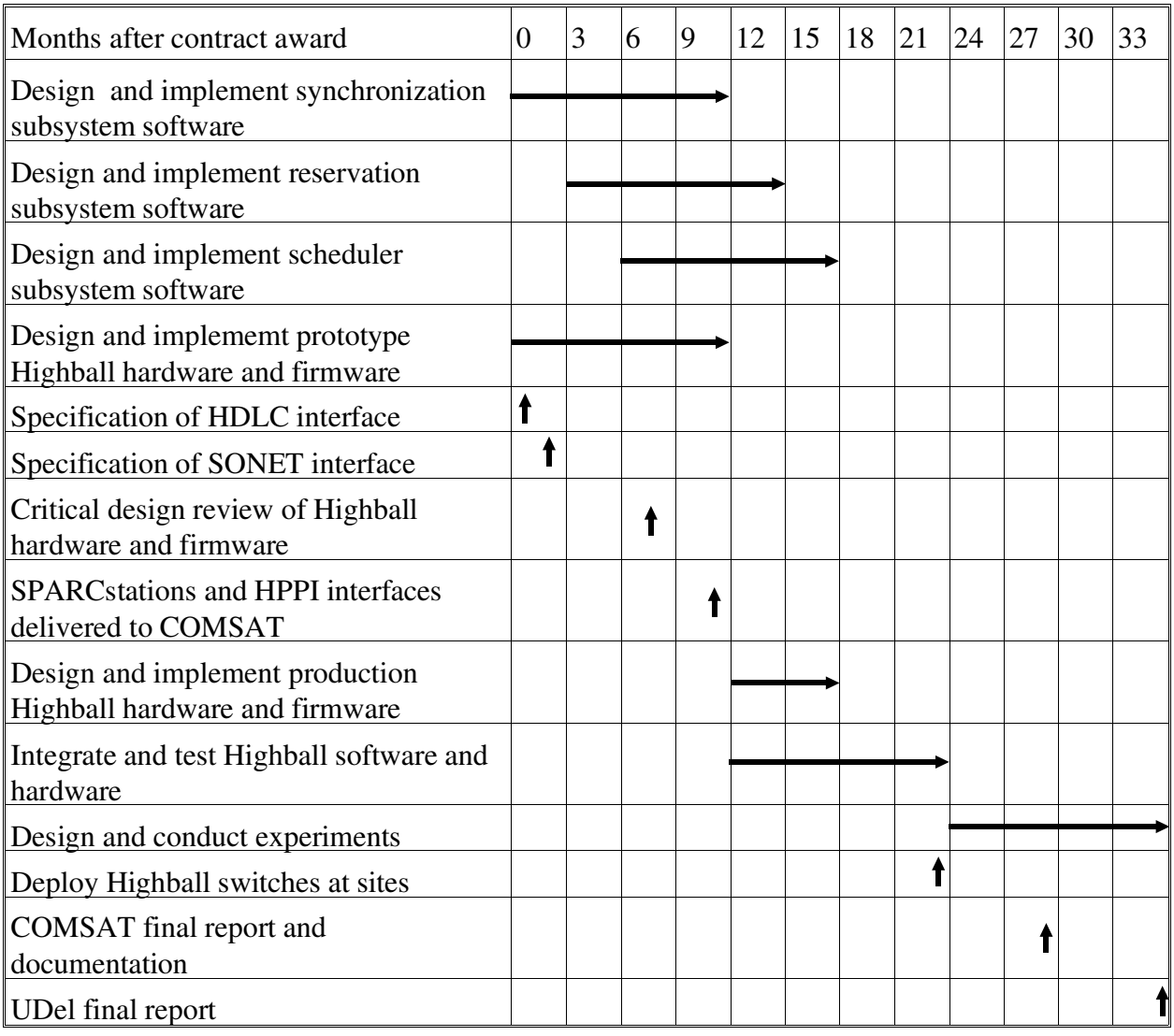
1. One prototype and three production SONET Multiplexors, each with one SONET interface, four HPPI interfaces and one HDLC interface.
2. Cables and loopback connectors for all interfaces.
3. One copy of a functional description of the equipment, including parts list, schematics and service information.
4. Monthly reports (1 copy).

E. Technology Transfer

There are three aspects to the technology-transfer plan of this proposal. First, the technology is described in the reports cited in the deliverables cited in section D. These reports form the institutional memory that survives the performance period of the contract. Second, the hardware and software components developed under this contract represent technology transfer in themselves, since they form models for production and from which further technology can be evolved. Third, realization of the goals of the proposed effort provides both a demonstration of the technology in actual use, as well as an experience base from which further applications can evolved.

F. Schedule and Milestones

The proposed period of performance is three years commencing in January 1993 or as soon thereafter as programmatically possible and continuing for 36 months.



G. Proprietary Claims

None.

H. Technical Rationale

H.1. Introduction

This is a proposal for a joint project with the University of Delaware (the University) and COMSAT Laboratories (the Laboratories) to support the DARPA/NASA Advanced Communications Satellite (ACTS) experiment program and to test and evaluate the Highball high speed burst-switching technology as applied to the ACTS system. At the University the project will be led by Prof. David L. Mills, while at the Laboratories the project will be led by Dr. Anil Agarwal. This proposal covers the overall technical rationale and approach of the project, while a separate proposal from the Laboratories (included as an annex to this proposal) covers the equipment to be built in support of the project.

The DARPA/NASA ACTS experiment program is expected to provide connectivity between up to five sites using SONET as the interface between the space and earth segments. The experiment includes a satellite-switched TDMA mode with three TDMA-switched beams and supporting SONET streams at OC-12 (622 Mbps) and OC-3 (155 Mbps). The static TDMA connectivity matrix repeats at 1-ms intervals and is uploaded from the operating center.

The Highball network architecture developed at the University of Delaware is designed for high demand, bursty applications, including supercomputers, real-time distributed experiments, retrieval of digitized maps, and visualization. Crossbar switches controlled by dedicated processors connect any input to one or more outputs "just in time" as bursts arrive. Bursts are not buffered in the network itself, except at the originating node, and are launched only as permitted by a distributed scheduling algorithm. Because of the high speeds involved, burst times are small compared to network propagation delays, which means bursts will often be completely contained on the network links or, in the case of geosynchronous satellites, in space.

The distributed scheduling algorithm is driven by reservations broadcast to all nodes using a dedicated control channel, either in-band or out-band. The algorithm is replicated at each node and operates synchronously to produce the switch schedule needed by that node. In the current model the switching nodes do not need to encapsulate or decapsulate the data bursts, so the burst encoding is not constrained. The design assumes nodal time can be synchronized throughout the network to a high order using methods adapted from existing Internet technology. System synchronization is maintained using methods borrowed from the field of parallel, discrete-event, simulation technology.

For application to the ACTS experiment the crossbar switch is represented by a set of FIFOs and multiplexor arrays in the SONET Multiplexor. These are used to switch incoming HPPI bursts to a SONET or another HPPI interface as appropriate. In the case of SONET, the multiplexor can replicate a burst at several times during the SONET payload cycle in order to provide a multicast service. The scheduling algorithm can operate on on a priority basis with either a burst-reserved or stream-reserved mode and with either explicit or implicit reservation requests.

The proposed design provides a maximum throughput via ACTS and Highball nodes between two HPPI hosts in different areas at the maximum ACTS channel rate of 622 Mbps and between two HPPI hosts in the same area at the maximum HPPI rate of 800 Mbps. This is sufficient, for example, to support applications where a supercomputer in one area multicasts real-time imagery to a number of frame buffers located in other areas. Other applications involving shared use of the SONET payload are possible, as long as the total offered traffic does not exceed the SONET channel rate.

The proposed work includes a generous degree of experiment time to test and evaluate the technology. While definitive designs for specific experiments are not possible at this stage, it is expected that these will follow on the models used in the SATNET and WIDEBAND systems to establish the system throughput, stability and agility regimes, as well as the systemic loss and fault-recovery mechanisms. These experiments will be conducted by both the University and Laboratories using existing software, workstations and networks at no additional cost to the contract.

H.2. 2. Motivation and Rationale

This experiment is designed to evaluate reservation-TDMA methods and scheduling paradigms applied to high speed, multiple-access satellite networks carrying bursty traffic. The satellite-switched nature of ACTS lends itself to a wide variety of network configurations and experiment scenarios, including some that involve multiple hops and temporary storage in space. The Highball technology provides the capabilities to

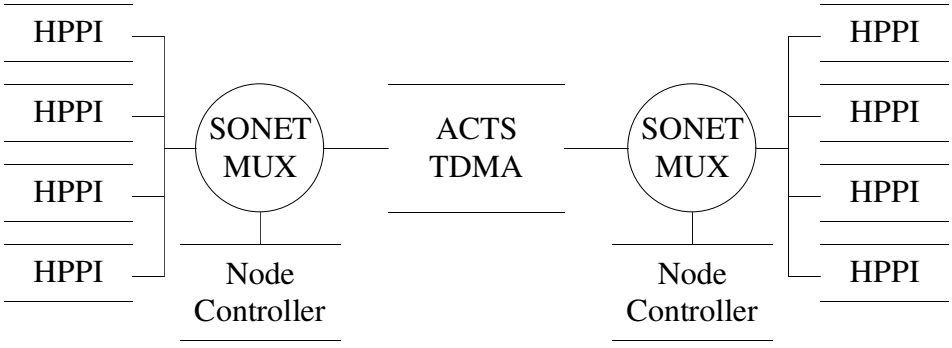
* Control access of multiple hosts to the various beams
* Regulate burst flow (using HPPI signalling) to maintain fairness
* Set up persistent end-end burst schedules for streams
* Set up multicast configurations and replicate bursts
* Cross-connect beams to provide additional bandwidth on demand

In the model assumed here, HPPI virtual circuits to be routed via the satellite are terminated at the Highball node, while HPPI data to be routed from one HPPI interface to another are directly switched and not buffered internally. Insofar as possible, it is the intent of the experiment that the switching node and ACTS space segment appear transparent to the hosts, except for the unavoidable address translations and delays. At the HPPI PH level the packet boundaries are preserved, while at the HPPI ULP level the packet-header formatting is preserved.

The Highball node can provide multiple-destination delivery, which may be very useful when distributing a high resolution image to a number of remote frame buffers. To do this, HPPI virtual circuits are set up between all participants and their local Highball nodes. The originating node calculates the spanning tree, sets up the interior nodes to replicate bursts accordingly, reserves the SONET bandwidth and flow-controls the source host as required. It is possible to control the delivery of data to the destination HPPI interfaces to a relatively high degree of precision, since the operation of the scheduling algorithm itself must be precisely timed.

The Highball node can function as a SONET cross-connect facility in which data received on one beam can be retransmitted to another beam. This provides an opportunity to capture unused capacity in order to temporarily increase the bandwidth available at a single site, for example. The bandwidth specification can be determined at the time the HPPI virtual circuit is set up and, in principle, changed dynamically during transmission by out-of-band messages received on the Highball Ethernet interface, for example.

The Highball node can provide address translations between the HPPI networks and admission-control functions for the space segment. The nodes communicate with each other and run the distributed scheduling algorithm using a dedicated, in-band “hello” burst transmitted by each node and containing housekeeping data and reservation requests. These bursts are transmitted in special



multiplexor slots embedded in the SONET payload. In addition, the hello bursts provide for the interchange of address-translation tables and as an order wire for system maintenance and low-rate data transmission.

There are three models for data transport that can be supported by a Highball node. The first two models require no intrusion in the end-system software, while the last requires some modification to provide a logical control path to the node. In the first model the arrival of a HPPI Connection Control Information (CCI) word at a node causes the HPPI protocol to be blocked and a reservation request to be sent to all other nodes via the control channel. At the scheduled time the HPPI protocol is unblocked and the burst is transmitted. The second model operates like the first, but the HPPI protocol is not blocked and the entire burst is buffered in the node. In the third model the host communicates directly with the node using an out-band Ethernet to assert burst-duration information, priority and destination address list. The proposed hardware design supports all of these models under software control.

In all three models transmission of subsequent bursts after the first one is under control of the SONET Multiplexor using the HPPI Ready signal. When all bursts have been transmitted the source terminates the connection and the distributed scheduling algorithm reallocates the channel bandwidth as required.

The Highball scheduling algorithm provides not only the microschedule that determines the individual burst transmission times and multiplexor configuration, but the macroschedule as well. The macroschedule amounts to a persistent (stream) reservation which is set up by explicit end-system requests and persists for a designated connection duration. It is set up at the time of initial connection by the Node Controller using information provided by the HPPI client in the CCI word or via an out-band Ethernet. The SONET Multiplexor implements the schedule by controlling the HPPI READY signals. The service specification for such streams can in principle include leaky-bucket and similar paradigms.

H.3. Technical Plan

The proposed ACTS Highball network consists of three switching nodes located on premises of existing supercomputer centers. Each node includes a SONET Multiplexor, described in the COMSAT proposal included as an annex to this proposal, together with a Sun Microsystems SPARCstation used as the Node Controller. Each SPARCstation is equipped with a CHI Systems HPPI interface and a Sun HSI synchronous serial interface. The HPPI interface is used primarily

for program development, baseline performance evaluation and for transmitting data bursts used for system monitoring, traffic generators and related functions.

The SONET Multiplexor is operated as an intelligent peripheral of the Node Controller. It includes one or more processor modules, each of which supports one SONET interface and four HPPI interfaces. The processor buffers data directly between the HPPI interfaces and SONET interface using 32-bit multiplexors and FIFO buffers, thus avoiding costly transfers to and from ordinary RAM or cache memory. Multiple modules can be incorporated in a single enclosure with common power supply and supporting circuitry to support multiple SONET interfaces or to expand the number of HPPI interfaces. It should be possible to provision a total of four SONET interfaces with a total of 16 HPPI interfaces without major modifications to the backplane or power supply.

The processor supports an aggregate of 622 Mbps in each direction between the processor and the ACTS SONET terminal. It will use available chipsets for SONET (e.g., AMCC S3005/S3006) and HPPI (e.g., AMCC S2020/S2021) which provide framing, control and clock recovery. The data can be routed between the SONET interface and any one or combination of four HPPI interfaces at the full burst rate (800 Mbps) of the HPPI specification. In addition, the processor can act as a HPPI crossconnect, where data received on one HPPI interface can be routed out another HPPI interface without degrading the throughput of other HPPI interfaces via each other or the SONET interface.

The above figure shows the system diagram for a network including two Highball nodes and teh ACTS satellite. The SONET Multiplexor and Node Controller communicate with each other at 1.544 Mbps using HDLC interfaces. This approach minimizes software and hardware development costs, since it allows the use of existing drivers and other software components developed in connection with DARTNET experiments and, indeed, provisions an interesting scenario where DARTNET and the ACTS Highball network can be interconnected.

The Node Controller software runs the suite of algorithms developed for the Highball project, including reservation, scheduling and synchronization functions. It also includes experiment-support functions such as the usual traceroute, tcpdump, ping and others found useful in DARTNET activities. As with other equipment involved in the ACTS experiments, the controller can be connected to an Ethernet which provides Internet connectivity for program development, testing and general system maintenance. If it becomes a burden on site-support staff to occasionally reboot the controller in case of a crash, the controller can be equipped with a telephone-activated reboot device similar to that used for the DARTNET routers located in telephone company PoPs.

H.4. Software Architecture

For the purposes of the proposed work, the software is that portion of the resident Unix kernel and application processes dedicated to the reservation, synchronization and scheduling functions required by the Highball technology. The underlying basis for these functions and the program-development cycle are based on the Unix system and, in particular the SunOS operating system and utilities. This is the basis for all ongoing work in the Highball project at the University and other projects at the Laboratories. Thus, program development and testing can be accomplished with no further expense to the contract.

The Highball Node Controller requires software drivers for the HPPI and HDLC interfaces. The CHI Systems HPPI interface includes software drivers with the product, while drivers for the Sun HSI interface, which supports the HDLC protocol, are already available from the DARTNET program. The kernel itself, with greatly improved timekeeping ability compared to the original issue,

is also available from the DARTNET program. This kernel also includes support for the ST-II and IP Multicast protocols, which may be useful both to support the control-message function, as well as the basis for some real-time data management.

The principal software components dedicated specifically to Highball operations include the reservation subsystem, scheduling subsystem and synchronization subsystems. The reservation subsystem captures CCI words as HPPI clients assert the CONNECT signal and determines the routing, dwell duration and dwell frequency of the switching schedule. These are then transmitted to all other nodes using the control channel in multicast mode.

The scheduling subsystem collects the reservations from all other nodes and runs the scheduling algorithm, several variants of which have been developed in the Highball project at the University. The algorithm produces a detailed time-evolved switch/multiplexor schedule that controls the actual multiplexor maps in the SONET Multiplexor. Each node strips out the particular pieces of the schedule that apply to its own traffic and downloads it via the HDLC interface along with computed dwell times to the SONET Multiplexor.

The synchronization subsystem disciplines the phase and frequency of the master timing generators used in all nodes in the network. The generators constitute an ensemble of disciplined clocks, where the discipline is provided by the algorithms of NTP. Each clock functions as an adaptive-parameter, type-II, phase-locked-loop (PLL), with measured time offsets between them exchanged periodically via the control channel. These offsets are processed by an ensemble-average algorithm to produce a master timescale more accurate than any of the clocks separately.

H.5. Timing and Multiplexor Subsystems

The Highball architecture requires that data be buffered at the source until enabled by the scheduling program. In order to use the network efficiently, it is necessary to synchronize all operations to within a relatively high degree of accuracy. Experiments with RISC workstations and conventional Unix kernels suggest the operating systems in the Node Controllers can be synchronized to within a few tens of microseconds without special-purpose timing interfaces. This provides sufficient accuracy for the global macroschedule (e.g., stream-setup and synchronization), resource management and general housekeeping. However, this is not good enough for burst scheduling in the SONET Multiplexors.

There are two ways in which the SONET Multiplexors can synchronized with accuracy sufficient for Highball microscheduling. One way is simply to tap the ACTS system clock in the form of the SONET clock, which is derived from the TDMA clock, which in turn is derived from a quality frequency source. This scheme would use the recovered SONET byte clock and frame pulse produced by the chipset to resolve the time to within the ambiguity of a SONET frame, then use the algorithms of the Network Time Protocol (NTP) to resolve the frame ambiguity. The second way is to provision each processor with a quality frequency source and to synchronize the epoch via the satellite network using protocols and algorithms adapted from NTP, which has been enhanced for just such a purpose. This approach would use the SONET pointer mechanism to establish the end-to-end epoch and discipline the quality source.

In order to support either type of synchronization, it is necessary to provide in the SONET Multiplexor a system of counters and latches which can be read by the Node Controller program. Using this system it should be possible to establish the epoch of the SONET pointer, to establish

the epoch of special control bursts used by the synchronization and scheduling protocols and to enable HPPI devices for transmission.

Since little data are buffered by the SONET Multiplexor, it is necessary to provide an enable function for the HPPI interfaces. In the HPPI specification the source drives the clock and data signals and the REQUEST signal, which is asserted by the source prior to transmitting a packet sequence, where each packet consists of a sequence of data bursts. The destination asserts the CONNECT signal when ready to begin and then a series of READY signals, one for each 256-word (1024 octet) burst which the destination is capable of buffering. The source maintains a counter which is incremented for every READY signal received and decremented for every burst transmitted. The destination sends a READY signal for each burst successfully delivered to its client. The source may transmit bursts as long as the counter is greater than zero.

The enable function required by the Highball scheduler can be implemented by controlling the CONNECT signal in response to the enable delay computed by the scheduling algorithm. Once the delay has expired, the CONNECT signal is asserted and a number of READY signals sent corresponding to the duration of the dwell computed by the scheduling algorithm. Assorted small speed mismatches are absorbed in the FIFOs. It is also possible to accomplish these functions by using only the READY signal, but this may result in a source timeout, since the enable delay can be quite long (up to a single satellite hop or 270 ms). In any case, the READY signal can be used to throttle the data flow in response to available buffers and SONET throughput.

There is some question whether the enable function can be provided as a programmed operation in the Node Controller or whether a programmable hardware timer is required in the SONET Multiplexor. If the scheduler can program the operation within an uncertainty of 20 us, as suggested by experience with RISC processors, all that is needed is a flipflop controlling the CONNECT signal. However, this translates to a jitter of over 12 kilobits at 622 Mbps, which may not be acceptable. A better solution may be the use of utility counters in the SONET Multiplexor which can be programmed by the Node Controller.

One of several possibilities to implement the multiplexor system is a 32 x 32 x 1 programmable crosspoint switch (AMCC S2024 chip). This device, which includes on-chip address decoders and a buffered configuration register, can connect any of 32 outputs to any of 32 inputs, which provides an inherent multicast capability. In order to implement a rapid reconfiguration of the switch, as system of FIFOs and dwell counters is necessary. Such a system has already been implemented for the prototype Highball network at the University; its design may serve as the basis for the proposed system.

H.6. SONET Interface

The ACTS earth-station complement includes the antenna, modems and SONET terminal. The SONET terminal includes one interface operating at 622 Mbps (OC-12) plus four interfaces operating at 155 Mbps (OC-3). It can be configured to operate using a single OC-12 data stream or four OC-3 data streams. The SONET Multiplexor can be configured to match either of these configurations; however, in the interest of economy, the equipment to be built initially includes only one SONET interface. Expansion to four SONET interfaces is expected to require only the addition of interface modules in a preconfigured backplane.

It is not clear at this time what the exact structure of the SONET multiplexing regime will be, other that the basic unit of multiplexing will be the 8-bit byte, as per the SONET specification. The SONET

multiplexors can be programmed to assign bytes or bursts to beams as determined on either a raw-byte basis or embedded in the 125-us SONET payload using available line-layer or path-layer overhead-processing chipsets such as the PM5311 SLTX and PM5301 SSTX . If the decision is made to conform to these upper-layer overhead formats, the basic section-layer interface could be as follows:

The SONET specification calls for a basic rate of about 51 Mbps, called the OC-1 rate, consistent with conventional DS3 terminal equipment and digital cross-connect systems (DACS). The basic SONET frame consists of 810 bytes organized in 9 rows by 90 columns with a frame period of 125 us and transmitted by rows. Multiplexing is implemented by transmitting a byte from each source in turn in round-robin fashion. This results in a multiplexor word of three bytes for an OC-3 stream and 12 bytes for an OC-12 stream. The SONET Multiplexor interface can include a state machine to generate the various headers and pointers associated with each multiplexor byte-plane.

The above discussion assumes multiplexing each byte plane separately, which is certainly the most flexible. Alternatively, three OC-1 streams can be concatenated as one OC-3 stream but carrying the SONET headers in only one byte plane and ignoring the header data in the remaining planes. A similar strategy can be used in case of the OC-12 stream. However, this strategy is not likely to work in cases where a single SONET interface is to carry data for more than one network destination. Thus, it would seem prudent to include the capability for up to four independent state machines (one for each other ACTS destination) operating within some number of byte planes.

The multiplexing operations are programmable by the Node Controller. Specifically, it should be possible to assign each of four state machines to any combination of up to 12 multiplexor byte planes. Each state machine is equipped with a set of FIFOs and 12-stage serial/parallel registers. The first n of these stages would be assigned to n of the multiplexor planes and the header machine assigned arbitrarily to one of them.

In order to synchronize network operations, control messages are exchanged periodically between the Node Controllers. It is necessary to accurately encode the time of transmission relative to the system clock as each control message is sent and to accurately capture the system clock when a message is received. The size of these control messages is fixed in order to simplify the scheduling and should be in the order of 1K octets. A simple way to do this is a pair of 1024-octet FIFOs that can be read and written by the processor using programmed transfers (no DMA). The program constructs the message along with a departure timestamp calculated for some time in the future, then loads the transmit FIFO. A programmable counter in the SONET Multiplexor inserts a measured delay after which the data are transmitted via one or more of the multiplexor byte planes while other data are held. This of course requires a state machine to delimit the control burst in the same way as an ordinary HPPI data burst.

Upon reception, it is necessary to capture the value of the system clock when a control message arrives. This can be done conveniently by latching the counter when a control message arrives and inserting the value in the control-message FIFO. Since the control message includes the timestamp imprinted at transmission, the control program can calculate the clock offsets and transmission delays to a high degree of precision.

H.7. HPPI Interfaces

The HPPI interfaces are relatively simple devices. Each consists of a state machine, bus-interface latch and clock-recovery circuitry (included in the chipset), together with a system of FIFOs and

multiplexors. The HPPI architecture features a ruthlessly simple bridge implementation where the only significant functionality is to switch the interface and data leads and redrive the bus. For the sake of generality, it would be prudent to provide the bridge function (with no buffering) between HPPI ports on the same interface module. The program simply connects the interface leads on the incoming HPPI port directly to the leads of the outgoing HPPI port and lets nature take its course. The program does have to monitor the REQUEST and CONNECT signals in order to terminate the operation and regain control of the signal leads.

The program does have to know when a client asserts the REQUEST signal, since it has to read the CCI word placed on the bus by the source at that time. The timing requirements are relatively modest, so this can be serviced by a simple program interrupt. In order to handle the bridging function, program interrupts should be produced for any transition of either the REQUEST or CONNECT signals.

If the HPPI transaction involves the satellite link, the program asserts the CONNECT signal, which can be generated by a programmed latch. At this time the program will already have set up the timing schedule, FIFOs and SONET Multiplexor/state machine assignments. Following these steps the data flow from FIFOs through the multiplexors is managed automatically by the state machines and chipset.

H.8. Buffering Considerations

It is not clear what the best buffering strategy is at this early stage; however, it is clear that the data paths cannot traverse a conventional path from the interface device via processor memory, caches and so forth. A set of FIFOs and programmable multiplexors are included on each interface module. This allows each module to operate at the full data rate of the SONET interface; however, data requiring cross-connect from one module to another must take a tortuously slow path via other SONET Multiplexor paths, possible via memory in its embedded controller, unless some sort of intermodule high-speed bus is used. One somewhat ingenious solution to this problem may be to use the HPPI interfaces themselves as the crossconnect. The outgoing leads of one interface can be wired to the incoming leads of another and the program can switch the leads accordingly. In this way a crossconnect can be implemented with essentially no loss in performance.

It is necessary to provide FIFOs at the HPPI input and output interfaces only. In particular, since the HPPI data paths are already set up when transmission is enabled, the data can be stuffed on the multiplexor byte planes as fast as it arrives and FIFOs used only as elastic stores. As a HPPI burst is limited to 256 32-bit words (1024 octets) the FIFO needs to be at least that large. The worst jitter on the SONET path is caused by header columns, but that is only 3 octets at 51 Mbps or 47 octets at 800 Mbps.

It does not appear necessary to decode the header burst included by the upper level framing protocol (HPPI-FP) preceding the data bursts of each packet. The FP header contains information about the header and data extent, error handling and forwarding at the destination (ULP Identifier). However, the actual physical-layer addressing function is included in the CCI provided by the source when the REQUEST signal is asserted.

In order to assure transparency for HPPI clients at different locations, it is necessary that the packet structure be preserved across the SONET and satellite link, which is transparent to all data except the OC-1 framing structure. This will require a state machine to encode the PACKET signals along with the data itself, as well as addressing and other information necessary to route the data at the

receiving Highball processor. This can be done by the HPPI interface as part of the encoded data stream.

H.9. Testing and Evaluation

It is not clear whether installation and access of the processors at selected test sites can be accomplished without formal cooperative agreements or on an informal basis. Included in the hardware procurement for the project are three conventional workstations with HPPI interfaces for installation at the test sites along with the SONET Multiplexors. These can be used as test hosts without disturbing the resident service population at that site. At sites equipped with HPPI crossbar switches, the test hosts and Highball processor can be simply plugged into available switch ports.

Testing and evaluation can be accomplished via the test hosts using existing Internet paths and satellite paths in much the same way that DARTNET experiments are conducted now. This may require a dedicated telephone line and gadget box in order to remotely reboot the test host as done now for those DARTNET routers located on telephone company premises.

There are a number of specific experiments that can be conducted using the Highball nodes and ACTS system. For initial proof-of-concept demonstration, traffic generators internal to the Node Controllers will be used. Early experiments will operate with static schedules and focus on the throughput, delay and fairness regimes in order to verify the system works as expected under normal and abnormal conditions, such as a rain fade. Later experiments will operate with dynamic schedules and focus on the ability of the scheduling algorithm to find timely and efficient schedules for bursty traffic. Later experiments will focus on the ability of the various subsystems to recover from fault conditions where one or more nodes have lost synchronization or are experiencing control-channel errors.

Once proof-of-concept has been established, it would be highly desirable and in fact expected that, should the Highball technology perform as expected, provisions be made to utilize the system from machines other than the test hosts, including existing supercomputers and frame buffers. This may require minor modifications in HPPI drivers in order to support the extended addressing and multicast modes which the satellite system provides. In addition, some adjustment in timeouts may be necessary to cope with the increased delay for the CONNECT signal to propagate via the satellite. However, many of these modifications would be necessary anyway for those HPPI configurations likely to be used in other media, such as long distance lightwave.

Experiments that can be conducted with real user data and protocols include the usual suite of IP/TCP protocols and application programs such as FTP and NFS. A particularly interesting application is NFS, since it can generate large volumes of data on short notice. Other protocols designed for high speed networks, such as NETBLT, may also be suitable for demonstration. Another interesting area for exploration may be ST-II for use in the frame-buffer multicast experiment described previously.

During the experiment phase of the project the algorithms and protocols of the Highball software and firmware implementation will be subject to continuous review and improvement as experience accumulates. Improvements include refinements to the scheduling, synchronization and reservation algorithms, as well as the provision of new features found useful as the result of experiments. While it is not the intent of the proposed work to concentrate on issues in application-software or protocol design, it is highly likely that suggestions for specific improvements will come from the work.

It is likely that other experiments may be going on during the period the Highball evaluation is under way. It is the intent of this project to require as little dedicated network time as possible. There may be opportunistic times where the SONET configuration can be adjusted to give the Highball node a dedicated SONET connection, while allowing other SONET connections to operate in a mutually nonintrusive manner. There may be other times when the Highball connection will require physically changing cable connections. The manner in which this can be resolved without extensive onsite assistance is for study.

H.10.Management Plan

The proposed project will involve the principal investigator, Prof. David L. Mills plus two graduate students for the duration of the experiment or three years, whichever happens first. This is most conveniently handled as a follow-on to the current DARPA contract. Computing and networking resources are available from existing DARPA projects. Additional resources needed include three Sun Microsystem SPARCstation workstations to be located on premises at existing supercomputer centers participating in joint projects. These workstations are to be equipped with HPPI and HDLC interfaces and software drivers. Additional funds are needed for travel, transportation, publication and equipment maintenance. Resources needed by the Laboratories will be detailed in their budget proposal and included in the cost of this proposal.

I. Key Personnel

Biographical Information for David L. Mills

David L. Mills is Professor of Electrical Engineering at the University of Delaware and presently leads projects in high-speed networks and internetworking research sponsored by the Defense Research Projects Agency, National Science Foundation and Department of the Navy. He is a Principal Investigator for the Highball Project, which is studying high speed routing, switching and synchronization technology. His research activities have been concentrated in the areas of network architecture, protocol engineering and experimental studies using the DARPA/NSF Internet system.

He has for many years been an active contributor to the field of computer network time synchronization. Protocols he developed, prototyped and deployed evolved to the Network Time Protocol, which is now in widespread use in the Internet from the Pacific Rim to Eastern Europe to South Africa. One of his current interests is in extending the accuracy, stability and robustness of this technology to the submillisecond regime on desktop workstations.

He is a member of the Internet Research Steering Group, End-to-End Research Group and AT&T Network Architecture Advisory Panel, and formerly chaired the DARPA Internet Architecture Task Force and Gateway Algorithms and Data Structures Task Force. He served on the National Academy of Sciences Committee on Survivable Telecommunications Networks, DARPA Internet Activities Board and NSF Network Technical Advisory Group, as well as several groups involved with internetworking research coordination and the establishment of the National Research and Education Network and the High Performance Computer and Communications Initiative.

Before joining the faculty in 1986, Dr. Mills was a Director (Networks) at M/A-COM Government Systems Division (Linkabit) and led DARPA-sponsored R&D projects in packet-switching network architectures and application protocols. Before that he was a Senior Research Scientist at COMSAT Laboratories, where he worked in the SATNET program in the areas of packet-switching satellite and internetworking technologies, and Assistant Professor of Computer Science at the University of Maryland, where he worked on several research projects in distributed computer networks and operating systems.

Dr. Mills earned a Doctorate in Computer and Communication Sciences at the University of Michigan in 1971 and has held postdoctoral positions at the University of Edinburgh (Scotland) and U.S. Defense Communications Agency. He has published and lectured extensively on data communications, computer networks and operating systems and has been a consultant to the United Nations, as well as a number of corporations and government agencies. He is a member of Sigma Xi, Eta Kappa Nu, Tau Beta Pi, Association for Computing Machinery, IEEE Computer and Communications Societies and Internet Society.

J. Budget

Prepared by Contracts and Grants Office

Includes:

1. Summer salary for D. Mills for three Summers
2. Two graduate students for three years
3. Three SPARCstations with HPPI and HDLC interfaces
4. Travel, equipment maintenace, communications, bits and pieces

K. Recent Publications Related to Proposed Work

Palmer, L., and D.L. Mills. COMSAT participation in ARPA packet satellite program - final report. COMSAT Laboratories, October 1979.

Kirstein, P., D.L. Mills, et al. SATNET application activities. *Proc. National Telecommunications Conference* (Washington, D.C., November 1979), 45.5.1-45.5.7.

Chu, W.W., D.L. Mills, et al. Experimental results on the packet satellite network. *Proc. National Telecommunications Conference* (Washington, D.C., November 1979), 45.4.1-45.4.12.

Mills, D.L. Internetworking and the Atlantic SATNET. *Proc. National Electronics Conference* (Chicago, Illinois, October 1981), 378-383.

Cudhea, P.W., D.A. McNeill and D.L. Mills. SATNET operations. *Proc. AIAA Ninth Communications Satellite Systems Conference* (March 1982).

Palmer, L., J. Kaiser, S. Rothschild and D. Mills. SATNET packet data transmission. *COMSAT Technical Review 12, 1* (Spring 1982), 181-212.

Leiner, B., J. Postel, R. Cole and D. Mills. The DARPA Internet protocol suite. In: P.E. Green (Ed.), *Network Interconnection and Protocol Conversion*, IEEE Press, New York, NY, 1988. Also in: *IEEE Communications 23, 3* (March 1985), 29-34.

Mills, D.L. The Fuzzball. Proc. ACM SIGCOMM 88 Symposium (Palo Alto, California, August 1988), 115-122.

Mills, D.L. Measured performance of the Network Time Protocol in the Internet system. DARPA Network Working Group Report RFC-1128, University of Delaware, October 1989. Also published as: Electrical Engineering Department Report 89-9-3, University of Delaware, September 1989.

Mills, D.L. Final report on DARPA/IPTO Contract N00140-87-C-8901: Research on Internet architectures and protocols. Electrical Engineering Department Report 90-7-2, University of Delaware, July 1990.

Mills, D.L. On the accuracy and stability of clocks synchronized by the Network Time Protocol in the Internet system. ACM Computer Communication Review 20, 1 (January 1990), 65-75.

Mills, D.L., C.G. Boncelet, J.G. Elias, P.A. Schragger and A.W. Jackson. Highball: a high speed, reserved-access, wide-area network. Electrical Engineering Department Report 90-9-3, University of Delaware, September 1990.

Mills, D.L. Internet time synchronization: the Network Time Protocol. IEEE Trans. Communications 39, 10 (October 1991), 1482-1493.

Mills, D.L. On the chronology and metrology of computer network timescales and their application to the Network Time Protocol. ACM Computer Communications Review 21, 5 (October 1991), 8-17.

Mills, D.L. Network Time Protocol (Version 3) specification, implementation and analysis. DARPA Network Working Group Report RFC-1305, University of Delaware, March 1992, 113 pp.

Mills, D.L. A computer-controlled LORAN-C receiver for precision timekeeping. Electrical Engineering Department Report 92-3-1, University of Delaware, March 1992, 63 pp.

Mills, D.L. Modelling and analysis of computer network clocks. Electrical Engineering Department Report 92-5-2, University of Delaware, May 1992, 29 pp.

Mills, D.L. Simple Network Time Protocol (SNTP). DARPA Network Working Group Report RFC-1361, University of Delaware, August 1992, 10 pp.

Boncelet, C.G., and D.L. Mills. A labeling algorithm for just-in-time scheduling in TDMA networks. Proc. ACM SIGCOMM 92 Symposium (Baltimore Maryland, September 1992).