Protocols for Loosely Synchronous Networks*

Danilo Florissi and Yechiam Yemini

Distributed Computing and Communications (DCC) Lab,
Computer Science Building, Columbia University, New York NY 10027, USA
Email: df@cs.columbia.edu, yy@cs.columbia.edu

Technical Report CUCS-017-94
(Paper published in the IFIP/IEEE Protocols for High-Speed Networks, 1994)

Abstract

This paper overviews a novel transfer mode for B-ISDN: Loosely-synchronous Transfer Mode (LTM). LTM operates by signaling periphery nodes when destinations become available. No frame structure is imposed by LTM, thus avoiding adaptation layers. Additionally, LTM can deliver a spectrum of guaranteed quality of services. New Synchronous Protocol Stacks (SPSs) build on LTM by synchronizing their activities to LTM signals. Such signals can be delivered directly to applications that may synchronize its operations to transmissions, thus minimizing buffering due to synchronization mismatches. SPSs can use current transport protocols unchanged and, potentially, enhance them with the real-time capabilities made possible through LTM.

Keyword Code: C.2.1; C.2.2; C.2.3
Keywords: Network Architecture and Design; Network Protocols; Network Operations

1. INTRODUCTION

Emerging Broadband Integrated Service Digital Networks (B-ISDNs) will have to integrate traffic requiring a broad range of guaranteed Quality of Services (QoS). The network transfer mode must be able to provide guarantees on delay, jitter, and loss to address the needs of data, voice, or video applications. Additionally, certain applications may require synchronization of remote activities and transfers. For example, synchronization is required among remote real-time computations or applications that use the network as a massively parallel computing resource. Current transfer technologies, the Synchronous Transfer Mode (STM) [4,9] and the Asynchronous Transfer Mode (ATM) [2,4], are limited in providing full coverage of these requirements. For example, ATM networks do not support guaranteed synchronization and offer a limited form of QoS guarantees.

* This research has been supported in part by ARPA contract F19628-93-C-0170, NSF contract NCR-91-06127, and Brazilian Research Council (CNPq) grant 204544/89.0.

This paper introduces a new transfer mode for B-ISDNs, Loosely-synchronous Transfer Mode (LTM). An LTM network enables transmissions by a source to a given destination during certain periodic time intervals or *bands*, much like STM networks. During bands, a source can transmit packets of arbitrary protocol structure and size (within set bounds). Unlike STM networks, the unit of transfer is not fixed, the size of bands is typically much larger than the transmission time of a unit of transfer and the periodicity of bands can be flexibly controlled. Once transmitted, an LTM packet can experience contention with other packets, as in ATM networks, as it moves towards the destination. Unlike ATM networks, the level of contention and with it the expected delay, jitter, and loss probability can be strictly controlled. Unlike ATM networks too, an LTM network does not require packets to be of a fixed size and structure, eliminating the need for adaptation layers at interfaces.

The main questions that LTM seeks to address in novel ways are: (1) how to synchronize source and destination with the network; (2) how to transfer multiprotocol frames without fragmentation and reassembly; (3) how to control and guarantee QoS; and (4) how to accomplish efficient bandwidth sharing.

The main goal of this paper is to describe the organization and functions of the interface stack of LTM networks. The primary purpose of this *synchronous stack* is to support isochronous application-application flow of packets. Two goals guide the design of this stack: (1) preserve as best as possible existing internet stacks; and (2) extend these stacks with a network-driven source and destination synchronization.

The first goal is accomplished by not creating a specialized packet structure for LTM networks (as in an ATM cell or in an STM frame). Instead, we treat the LTM as a media access layer and handle layers above through standard packet encapsulation techniques. It is important to note that the synchronous stack does not perturb packet structures or operations of current protocol entities. Indeed, an existing stack can be easily located above the LTM MAC through appropriate conversion of Service Access Points (SAPs). The synchronous stack extends the functionality of these existing stacks by providing an orthogonal service of synchronizing motions of packets through the stack and the network.

The second goal is accomplished by providing novel bottom-up synchronization signals from the network through the stack. Source and destination applications can synchronize their activities to the periodicity and size of network bands. For example, a video application can generate frames to synchronize with bands over which they are transmitted. Furthermore, through appropriate top-down signalling to the network, applications can exercise control over bands periodicity and size.

This paper is organized as follows. Section 2 presents an overview of Isochronets (a switching architecture that implements LTM) and LTM. Section 3 overviews SPS and how multiple traffic classes can be supported. Section 4 compares LTM with STM and ATM. Finally, Section 5 concludes.

2. ISOCHRONETS BACKGROUND

The goal of this section is to describe one particular existing LTM network: Isochronets [5,10]. It is important to emphasize that other LTM implementations exits, one example being the Highball network [7].

2.1. Architecture and Principles of Operations

Isochronets seek to provide flexible control of contention to accomplish desired QoS by routing network traffic along routing trees leading to respective destination nodes. Bandwidth is time-divided among, and synchronized along routing trees. The basic construct for bandwidth allocation is a time-band (*green-band*) assigned to a routing tree. Figure 1 depicts a network topology with the routing tree (marked with directed thick links) leading to the dark node. The graph on top plots traffic motion from source to destination through the gray nodes. The Location-axis shows the location of a given frame at the time marked in the Time-axis. During the green-band (shaded area in the graph), a frame transmitted by a source will propagate down the routing tree to the destination root (a typical frame motion is depicted using a line within the shaded area). If no other traffic contends for the tree, the frame will move uninterrupted, as depicted by the straight line.

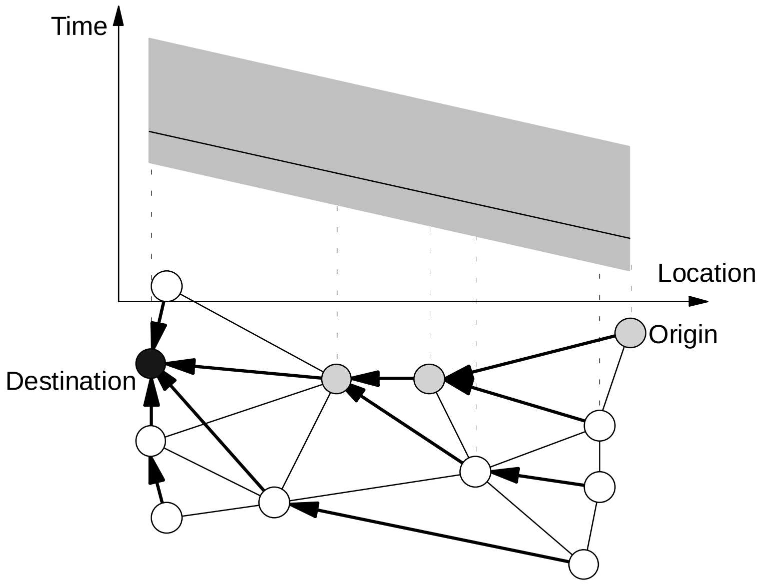

Figure 1. Green-band.

The green-band is maintained by switching nodes through timers synchronized to reflect latency along tree links. Synchronization is per band size, which is large compared to frame transmission time. It can thus be accomplished through relatively simple mechanisms. Routing along a green-band is accomplished by configuration of switch resources to schedule frames on incoming tree links to the respective outgoing tree link. A source sends frames by scheduling transmissions to the green bands of its destination.

Bands are allocated periodically as portions of a *cycle*. They need not occupy the same width throughout the network. Indeed, one can view a green band as a resource that is distributed by a node to its up-stream sons (as long as the bands allocated to sons are scheduled within the band of the parent). In particular, if the bands allocated to two sons do not overlap, their traffic does not contend. By controlling band overlaps, switches can fine-tune the level of contention and statistical QoS seen by traffic.

One may view these mechanisms to schedule traffic motions by way of band allocations as a media-access technique. The entire network is viewed as a routing medium consisting of routing trees. Bandwidth is time- and space-divided among these routes. Sources need access respective trees during their band times, seeing the network as a time-divided medium, much like Time Division Multiple Access (TDMA) [9]. This technique, accordingly, is called *Route Division Multiple Access (RDMA).*

A *contention band* is a band that may be shared by multiple sources simultaneously. Its name is derived from the fact that multiple sources may decide to use the band at the same time and thus *contend* for intermediate tree links. When contention occurs, the collision resolution mode used is designated in terms of the signs "–", "+", and "++". In *RDMA–*, only one of the colliding frames proceeds, while the others are discarded. In *RDMA+*, one colliding frame proceeds while the others are buffered, but only up to the band duration. *RDMA++* operates similarly to RDMA+, but also stores frames beyond band termination, rescheduling them during the next band.

Isochronets use *priority bands* and *multicast bands* in addition to contention bands. Priority bands are allocated to sources requiring absolute QoS guarantees, similar to a circuit service. Traffic from a priority-source is given the right of way, by switches on its path, during its priority band. Unlike circuit-switched networks, however, priority sources do not own their bands. Contention traffic may access a priority band and use it whenever the priority source does not. During a multicast band, the routing tree is reversed and the root can multicast to any subset of nodes.

Bands are thus shared resources that may be passed from intermediate nodes to subtrees. Nodes may decide to pass only portions of their bands to their sons. Also, the portions may be of different sizes. Thus, the final band allocation scheme may be designed taking advantage of the rich structure enabled by the band allocation possibilities.

A few observations regarding Isochronets are in order. Multiple simultaneous routing trees can schedule transmissions in parallel (have simultaneous green bands), depending on the network topology. Figure 2 shows two non-interfering routing trees.

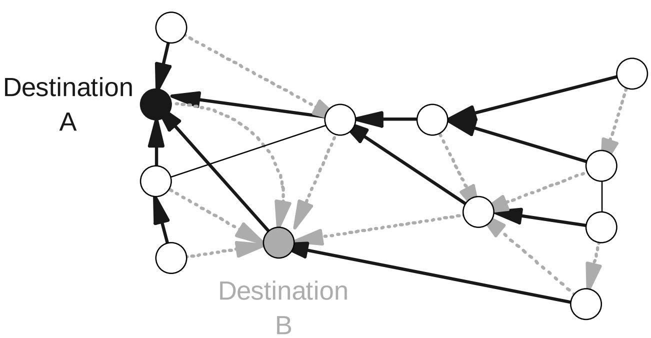

Figure 2. Multiple non-interfering trees.

No header processing is necessary in Isochronet nodes. Frames on incoming links can be mapped into the corresponding outgoing link based solely on the current routing tree structure which, in turn, may be derived from the current time. Thus switching can be accomplished without any processing that is dependent on frame contents. This means that Isochronets may operate at any link speed.

Since no frame processing is performed at intermediate nodes, all stack layers above the media-access layer are delegated to interfaces at the network periphery. That is, Isochronets may transport any frame structure without adaptation because frames do not need to be parsed to derive routing information. A typical stack organization for Isochronets is depicted in Figure 3. Interconnection of Isochronets can be accomplished by way of media-layer bridges using extensions of current well-understood technologies.

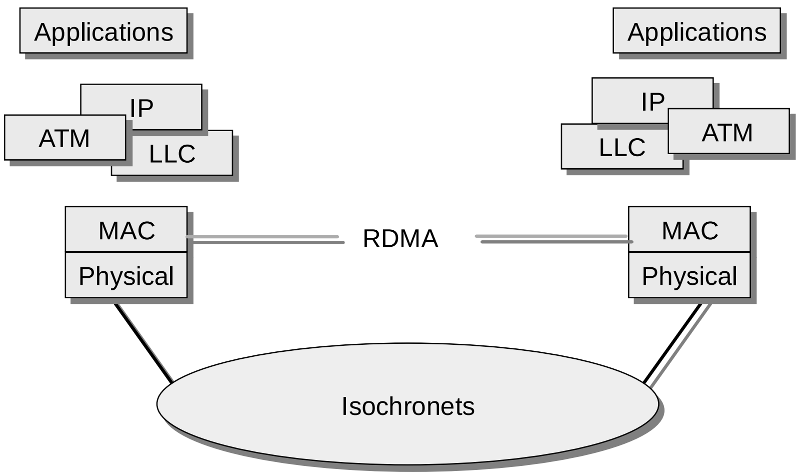

Figure 3. Multiple protocol stacks in Isochronets.

The following is a typical Isochronet operation scenario. A set of end nodes is connected to a backbone Isochronet network. The backbone periodically enables destinations in a cyclic manner until all destinations are covered. The end nodes interact with the Isochronet backbone switches by accessing the bands to deliver frames and by requesting services. Services are requests for band allocation with QoS demands in the form of band type, size, and periodicity.

An interesting question to be solved in RDMA is how to use contention or priority bands when they enable multiple backbone destinations. That is, the end node is attached to a RDMA backbone switch that signals contention or priority bands to multiple destinations. For example, a contention band may enable more than one non-interfering tree, as depicted in Figure 2. If the end node has only one link to the switch, its use must be multiplexed among all enabled destinations. One possibility is to partition the band at the periphery node among all destinations and signal each destination individually. This is equivalent to time-dividing the link between the periphery node and the attached switch among destinations. Another possibility is to use a local frame addressing scheme between the periphery node and the backbone. Each frame would contain the intended destination address so that the attached switch at the backbone can decide how to forward the frame. Notice that such addressing scheme is local between the periphery node and the attached switch, and is not used inside the RDMA backbone.

2.2. Isochronets Support LTM

This section defines the main properties of LTM networks. The nomenclature used follows the one in Isochronets.

LTM networks issue *synchronization signals* that can be used to schedule frame motion among source-destination applications. The period of time between signals is called a *band*. Bands embody two global network status: (1) a connection to destinations in the network, and (2) a certain QoS associated to the band. Bands are repeated periodically in a *cycle*.

Depending on the QoS offered, bands can be of three kinds: (1) contention, (2) priority, and (3) multicast. During a contention band, access is shared through some fair competition and the only guarantee provided is that the network will seek to optimize bandwidth use. During a priority band, the network will provide a circuit service to the destination. That is, frames from the sources will not be affected by any contending traffic. During a multicast band, the network provides contention free multicast to a group of destinations.

LTM can adapt to traffic characteristics and mimic advantageous characteristics of both STM and ATM. Similarly to STM, QoS can be guaranteed. That is, through the allocation of priority to destinations in the network, LTM can deliver the requested QoS. For example, end-end delay can be bound by the time waiting for the priority band (which in turn is bound by the cycle duration) plus the transmission and propagation delay in the network. Additionally, since sources get priority and not exclusive use of resources, bandwidth utilization is improved in LTM when compared to STM.

Furthermore, nodes at network periphery do not need to synchronize their clocks globally, as in STM. Necessary synchronization information is given by the network and nodes need to synchronize only locally with the network. Also, the necessary accuracy is much lower when compared with STM because nodes need to know only what is the current band (which usually lasts for a long time). For example, an 8 bit slot in a STM frame at 2.4Gb/s transmission rate lasts 3.3ns. Typical bands last between a few hundreds of nanoseconds up to a few scores microseconds.

Similarly to ATM, diverse traffic classes may be serviced by LTM. The transfer of a given traffic class is bound to periods in which the network is offering the most appropriate characteristics for the service requirements. For example, video traffic must be sent during periods when priority to the correct source/destination is enabled. Data traffic may be sent during any period in which the correct destination is enabled.

LTM may nevertheless achieve accurate traffic synchronization and potentially avoid buffering in the network. In ATM, such buffering is necessary to compensate synchronization mismatches due to network resource multiplexing. Since in LTM global network information is known, sources may tune traffic generation in order to minimize contention buffering in the network.

In the context of this work, LTM is to be used as the transfer technique in the backbone network. Signaling information is supplied by the attached switches to the periphery nodes that can use it to implement their protocol stacks.

3. THE SYNCHRONOUS PROTOCOL STACK

The SPS is the stack at peripheral nodes attached to a backbone network that uses LTM. In addition to the normal data flow between stack layers, SPS implements a bottom-up flow of synchronization signals from the underlying LTM network. These synchronization signals can be used at any layer, including the application, to implement synchronization functions. The general structure of SPS is depicted in Figure 4. This section briefly describes each SPS component.

The Physical Layer (PL) does not need to be bound to any special technology (i.e., electronic or photonic implementations may be employed) since LTM does not rely on any particular frame structure. The data link and network layers are collapsed into the LTM Media Access

(LTM-MAC) layer that uses LTM as the transfer mechanism. The Transport Layer (TL) is responsible for the allocation and control of network resources, such as bands with necessary QoS. The Application Layer (AL) interacts with the TL requesting necessary classes of services.

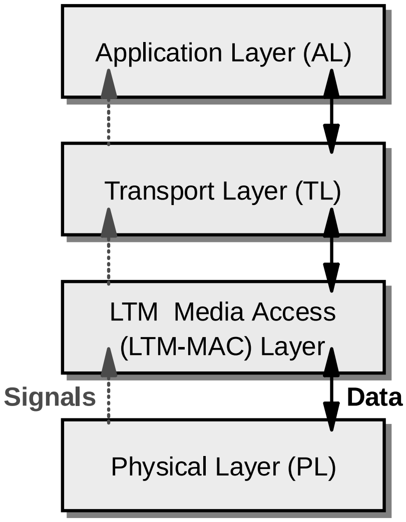

Figure 4. Synchronous Protocol Stack (SPS) structure.

Any traditional TL protocol currently used over STM or ATM may be used over the LTM-MAC directly with no changes other then adapting to the LTM-MAC SAPs. In addition, the signaling information from LTM-MAC can be used to enhance the functionality provided by any traditional TL for real-time service provision. These issues are detailed in Section 3.2.4.

A unique feature of SPS is that synchronization signals (and not only data) may be reflected all the way to the AL. The network may thus inform its current status directly to applications which may then schedule its operations to transmissions. For example, video traffic may be scheduled to generate frames when the proper band begins in each cycle.

In the general case, SPS may be a portion of the overall protocol stack, as depicted in Figure 5. Stack layers need to be able to operate in real-time to handle signaling from the MAC-LTM. The protocol stack is divided in two portions: a lower real-time protocol stack that can handle signals from LTM, and an upper non-real-time protocol stack that does not have real-time service provision. The *SPS boundary* is the interface between both stacks. The interface is responsible for buffering requests to overcome operational lack of synchronization between both stacks. Synchronization of operations with transmissions can be guaranteed only below the SPS boundary. The figure shows data being sent from protocol stack A to B, using the signaling from the LTM network. Notice that signaling is not passed above the SPS limit.

The extreme cases of this scenario are two. In the first, the SPS boundary is above the application layer, and thus LTM signals can be relayed up to applications. This can be the case when machine hardwares and operating systems provide real-time support, that is, when it is possible to predict upper bounds on execution times. In the second, the SPS boundary is at the interface with the LTM network and no signaling from the LTM network is forwarded to upper protocol layers. This could be the case, for example, when traditional protocols are to be implemented on conventional machines without real-time support.

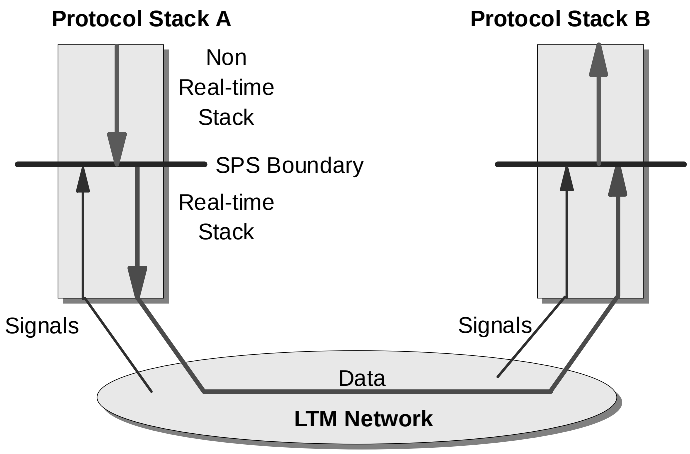

Figure 5. Handling signals in the protocol stack.

3.1. The LTM-MAC Service Access Points

This section summarizes the LTM-MAC SAPs. Notice that the LTM layer may be implemented using any technology as long as the LTM-MAC SAPs are kept unchanged and all protocol layers above the LTM-MAC can operate independently of the specific mechanisms used to implement the LTM.

The LTM-MAC SAPs are summarized in Figure 6. The first service is *OUT_BAND.signal(band, size)*. It is a signal from LTM to mark the beginning of an outgoing band. The *band* parameter has the form *<band_id, type>* where *band_id* is an identifier for the band and *type* is one of the QoS associated with the band (contention, priority, or multicast). For example, *<5, c>* means that the band identifier is *5* and its type is contention. The *size* parameter is the length of the corresponding band in nanoseconds.

Similarly, *IN_BAND.signal(band, size, protocol)* signals the beginning of an incoming band. Additionally, since the MAC may multiplex multiple transport entities above it, the *protocol* parameter identifies the protocol that should service the band.

OUT_BAND.signal(band,size)	Signals the beginning of an outgoing band.
IN_BAND.signal(band,size,protocol)	Signals the beginning of an incoming band.
ESTABLISH_BAND.request(size, periodicity,type,destination,band_id)	Allocates a band of given size, periodicity, and type to a given destination.
ESTABLISH_BAND.response(reason)	Answers band allocation requests.
RELEASE_BAND.request(band_id)	Releases a band.
RELEASE_BAND.response()	Answers band release requests.
DATA.request(frame)	Sends a frame through the current band.
DATA.indication(frame)	Signals reception of a frame.

Figure 6. LTM-MAC SAPs.

The next services are used to establish a band. *ESTABLISH_BAND.request(size, periodicity, type, destination, band_id)* requests the establishment of a band of a given *size*, *periodicity*, and *type* to the respective destinations. The periodicity is the amount of time between occurrences of the same band. If it is 0, the band is allocated only once in a cycle. The band identifier for the connection is returned in the *band_id* field. The *destination* field denotes not only the destination machine address, but also the destination protocol used in the IN_BAND.signal SAP at the destination.

The manner in which the LTM-MAC is going to achieve band allocation is dependent on its internal operations. Contention bands are allocated as a portion of the LTM supplied contention band to the given destination. To allocate priority or multicast bands, the signaling of the backbone network must be used to negotiate the allocation and inform all nodes involved. Priority bands are allocated as a portion of the respective LTM supplied contention band. *ESTABLISH_BAND.response(reason)* indicates if the band was established or not (and in the latter the *reason* for failure). Band requests may fail because not enough resources are available to allocate the requested QoS.

SAP *RELEASE_BAND.request(band_id)* is used to release previously allocated bands. *RELEASE_BAND.response()* is used by LTM to signal when the request is finished.

Finally, the last services are used to send and receive frames through LTM. *DATA.request(frame)* sends the user supplied *frame* through the network. *DATA.indication(frame)* is used by LTM to signal the arrival of a *frame*.

3.2. Transport Protocols

This section discusses how current transport protocols (IP, ATM, etc.) can use LTM-MAC SAPs directly to implement their functionality. Additionally, the signalling features enabled by LTM are used to show how such protocols can be extended to implement asynchronous, synchronous, and isochronous services.

3.2.1. Mapping Destination Addresses onto LTM-MAC Bands

Transport protocol addresses need to be mapped into appropriate band identifiers at the network periphery to enable transmissions through the LTM-MAC. Translation tables are used to this effect. The fields in a translation table are destination address and band identifier. For each reachable destination address, the corresponding band identifier (that is, the identifier for the band that routes to that destination) is given.

The next question to be addressed is how such translation tables are set initially. That is, transport layer entities need to be able to find band identifiers connected to desired destination addresses. A variation of the ARP protocol of the Internet Protocol [3] stack is used to implement this function. The protocol works as follows. A special frame containing the address of the requesting TL entity is transmitted during a band to request the TL address of the associated destinations. The peer TL entities recognize the special frame and reply with their TL addresses, using the band to the requesting entity. If the band identifier of the requesting TL entity is not already known by a replying TL entity, its reply is sent during all bands in the cycle to cover all possible requesting entities. Notice that by sending such request frames during all bands in the cycle, all destinations are covered and the table in the requesting TL entity is accordingly initialized.

The address translation mechanism described may be implemented more efficiently (in terms of bandwidth use, that is, avoiding broadcasts of requests) by using a special name server accessible through a special band identifier. The server keeps the current mapping and answers requests for address resolution.

3.2.2. LTM Supports Asynchronous Traffic

Asynchronous traffic requires no time constraints or loss guarantees on frame delivery. Examples of applications that generate such traffic are electronic mail delivery, file transfers, etc. Since these applications do not have hard timing constraints, frame loss may be overcome by retransmission. Asynchronous traffic can be directly supported on top of contention bands.

For example, to implement a file transfer application, a contention band can be used to transfer each portion of the file. When errors occur, they are detected at the destination TL entity that requests retransmission using the reverse band to the source.

The Internet Protocol. The traditional Internet Protocol (IP) [3] is a an example of asynchronous communication protocol. To implement IP, a contention band is established to each destination through the ESTABLISH_BAND.request SAP. Packets received from upper layer entities are buffered in the IP layer according to their band identifiers (which is computed from frame destination addresses). When the beginning of an outgoing band is signalled to the IP layer, it forwards the respective buffered frames. Similarly, when the beginning of an incoming band to an IP entity is signalled, the entity receives the frames and forward them to upper layer entities.

3.2.3. LTM Supports Isochronous Traffic

In *isochronous traffic,* frames must be played-back (that is, used) with minimal jitter between them (that is, frame access should happen at constant intervals) and some loss may be tolerated.

Such services can be implemented on SPS by making the TL compile requests into two parameters available through ESTABLISH_BAND.request: priority band size and periodicity (that is, how many cycles apart should the band be allocated). The mapping is performed according to the QoS requested, that is, depending on the requested jitter and bandwidth. The priority band periodicity is determined by the jitter requirements and buffering capacity at the destination. The priority band size is then computed from the requested bandwidth, link capacity, and granted periodicity.

For example in a video transmission, if the cycle period is 125μs, allocation can be implemented as follows. A priority band can be allocated for each frame every 264 cycles. Alternatively, a smaller priority band can be allocated for each frame with higher frequency, depending on the amount of buffers available at the destination. As long buffering space for 1 frame is available, the allocation can be done such that every 264 cycles 1 complete frame is delivered. A typical video transmission in this scenario is depicted in Figure 7. When an application receives a signal from the LTM-MAC, it is awaken and it transmits a video frame (or portion of a video frame). After that, the next video frame (or portion) is generated by the application which then goes to sleep waiting for the next signal. The signals thus pace the application to generate isochronous frames.

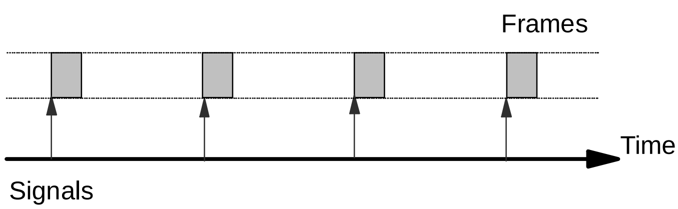

Figure 7. Isochronous transmissions.

Another possibility is to profit from the fact that some loss may be tolerated in this kind of communication. The TL may then allocate two kinds of contention bands: one for asynchronous traffic and another for isochronous traffic. The contention band for isochronous services can be used according to distributed protocols that allocate resources by maximizing multiplexing constrained to the tolerable loss allowed by applications, as it is done in the context of ATM networks [4]. That is, portions of the contention band for isochronous traffic are allocated not to guarantee lossless communication, but to deliver low probability of loss. In this manner, the portion of the band to be allocated is smaller than would be necessary for no loss. When sending frames, the TL always gives priority to isochronous traffic over asynchronous traffic.

After the band allocation phase is completed, the TL receives signaling information from LTM-MAC when corresponding bands begin. It then schedules signaling to applications when the corresponding priority bands are due. When signaled, applications may send data to TL which uses LTM-MAC to transmit them. Potentially, traffic generation may be scheduled to begin only when signaling is received from TL, thus minimizing buffering.

The ATM Adaptation Layer. The ATM Adaptation Layer (AAL) [2,4] protocols can be implemented using two alternative ESTABLISH_BAND.request options: priority or contention band. The ATM virtual path and virtual channel identifiers are translated into band identifiers. ATM services not requiring QoS are implemented on top of contention bands, in similarity to the IP protocol stack. QoS demanding services must be implemented on priority bands. The following overviews how each AAL protocols can be implemented.

The AAL 1 is intended to service constant bit rate applications such as uncompressed video transmissions. ALL 1 services can be directly implemented using a priority band with periodicity equal to the necessary sampling rate. If the sampling rate is too small and each sample contains more information than what can be allocated in one band, the sampling rate may be increased and the band size decreased. For example, to accommodate 100Mbits/s video transmissions, a band of size 3.3Mbits can be allocated every 33ms or a band of size 100Kbits can be allocated every 1ms.

The AAL 2 is intended for variable bit services. Such services can be accomplished in several ways, depending on the error rate to be allowed in the communication. One possibility is to allocate two contention bands, one for normal contention traffic, and another to service variable bit rate services, as explained previously. Another possibility is to guarantee error-free delivery by allocating a priority band.

The AAL 3/4 and 5 are intended for data communications sensitive to loss, but not to delay. This is the ideal application for a contention band, as explained for IP.

Notice that all frame structures of the various AAL protocols can be sent directly to LTM-MAC, without adaptation. This is because LTM does not rely on any particular frame structure to perform its operations.

3.2.4. LTM Supports Synchronous Traffic

In *synchronous traffic*, it is necessary to guarantee maximum end-end delay (that is, the delay to the destination may fluctuate, but must be bound by a pre-negotiated value) and error-free communication. This kind of traffic is supported by allocating a priority band.

For example, a virtual high-speed multiprocessor machine can be implemented using a set of machines interconnected by a network such as Isochronets. This application requires sporadic exchange of small amounts of data for interprocess communication. The transfer, nevertheless, needs to be reliable (error-free) and done in a timely fashion due to the high-speed of the processors. Priority bands can be pre-allocated for this sporadic communication in every cycle. The bandwidth size is computed from the maximum bandwidth required between processors.

Most observations from Section 3.2.3 in the context of scheduling isochronous traffic generation according to the synchronization signals from LTM are applicable for synchronous traffic as well.

Synchronous IP. An important feature in SPS is that the signals that are input from the LTM-MAC can be used to extend existing TL protocols towards providing synchronous services. For example, the IP suite can be extended with new SAPs to the application layer to support synchronous transport. Such SAPs would be implemented using priority bands at the LTM layer.

3.2.5. Compiling Higher Level QoS Parameters

Higher level QoS parameters such as end-end delay, loss, and jitter need to be compiled into the elements made available by the LTM-MAC layer, that is, type of band, band size, and band periodicity. Such compilation is performed by TL protocols, depending on the high-level QoS parameters they offer. In this section presents an example of how the translations can be performed.

In the example Protocol 1, two parameters (delay and bandwidth) are used by the application layer to request transport layer services: maximum end-end delay and bandwidth needed (Step 1). The variable P in Step 2 is used to always allocate only a portion of the LTM supplied band, to avoid compromising the whole band with one request, if possible. The allocation begins backwards in Steps 3 and 4 searching for idle portions in the cycles from the deadline (T+D) up to the current time (T). In Step 5, the allocation is tested. If it was successful, that is, the first allocated priority band begins at least at time T+O (where O is the overhead necessary before the first frame can be sent) the application is informed about the allocation. If not, new allocations are tried with a new vale for P. If, after all values for P have been tried, no feasible allocation exists, the failure is communicated to the application.

1. Let the requested delay be D, and the requested bandwidth be B.
2. Assign 50% to P.
3. Mark location T+D (where T is the current time) in the time line.
4. Search each cycle backwards beginning from T+D and fill at most P percent of the idle portion of the band assigned for the source and destination pair. The search is performed by requesting the LTM-MAC to allocate a portion of the requested size of the band to the destination. The search begins with the cycle in which the band ends before and closest to T+D. It ends with the one in which the band begins after and closest to T.
5. Let E be the instant in the time line when the first found portion of the band begins. If E ≥ T+O (where O is the overhead until the first transmission can happen), the allocation is feasible. Stop and inform the application. If E < T+O, the allocation is not feasible go to Step 6.
6. If P is less than 100%, add 10% to P and go to step 3. If P is 100%, stop. The requested service cannot be delivered. Stop and inform the application.

Protocol 1. Example end-end delay and bandwidth QoS compilation.

A few observations are important in the example described. Firstly, optimizations can be performed, but were not adopted for simplicity. For example, O could be estimated to avoid the situation in which the allocation succeeds, but the feasibility test fails. Secondly, this is only one possibility for mapping end-end delay and bandwidth requests into priority bands. Each transport layer protocol may have its one translation algorithm, most suitable for the services it intends to provide.

Notice that the implementation of requests for QoS in terms of jitter and bandwidth can be accomplished using Protocol 1 by substituting the required maximum jitter for the maximum delay. Also, care must be taken to request a periodic allocation, instead of a single allocation (where the period is input as the delay in the protocol).

4. RELATED WORK

This section compares LTM with STM and ATM as solutions for B-ISBN.

Plain STM generates a periodic fixed-size frame. The frame is divided in fixed-sized slots (usually of size 1 byte or a multiple of 1 byte) that can be used by sources to transmit information. Once allocated, bandwidth is guaranteed for the connection, thus delivering good QoS in terms of guaranteed end-end delay, and no jitter or loss. It is necessary, nevertheless, to keep a virtual global clock in order to synchronize all nodes in the network to the global frame and slots within the frame.

The main problems of adopting STM for B-ISDN is the lack of flexibility in the slot size and in supporting on-demand service allocation. Applications such as voice communication require small slots (usually 8 bits per frame), while video communication would best profit from large slots. If the slot size is defined too big, network bandwidth may be wasted while if it is too small, it may be difficult to allocate broadband services. STM lacks provision for asynchronous traffic as well (e.g., on-demand packet switching). When such traffic needs to access the network, slots must be allocated in the whole path from source to destination with unacceptable end-end delays.

Flexibility in bandwidth allocation is the main force pushing ATM as a solution for B-ISDN. In ATM, information is partitioned into fixed-size cells that are sent asynchronously to the destination. Destinations are recognized by using identifiers in the cells (as opposed to being identified by the location in a frame as is the case in STM) and, as a consequence, no global clock synchronization is required. Nonetheless, virtual connection (channel or path) establishment is necessary to allocate identifiers. Bandwidth can be flexibly allocated based on source demands by scattering incoming traffic into cells.

The main problems of ATM are limited support for asynchronous or synchronous communication and the trade-off between guaranteed QoS and efficient network utilization. The main drawback of asynchronous communications over ATM is that they need to be preceded by the virtual connection establishment phase, which involves end-end delays. The connection establishment phase in many applications may take longer than the transfer phase, which makes asynchronous communications inefficient both in end-end delays and in resource utilization. Some work [4,6] has been done to overcome this problem by allocating permanent virtual channels for the purpose of sending asynchronous traffic, but these solutions may require complex management of identifiers for all source/destination possibilities and may poorly use network resources.

The necessary QoS parameters for synchronous or isochronous communications are negotiated during connection establishment. Nevertheless, if all network resources are to be allocated to guarantee QoS, ATM will poorly use network resources, similarly to what happens in STM. For example, video coding algorithms usually generate variable bit rate outputs. To guarantee no loss during a video section, ATM would need to allocate resources for peak bit rate. But, the peak to mean bit rate ratio is usually high, which means that network utilization may become poor. Due to this problem, resource allocation in ATM networks is usually performed based on a lower QoS than the one requested. The idea is that multiplexing several connections and granting lower QoS to each may deliver high QoS for the multiplexed ensemble while accomplishing high network resource utilization. Unfortunately, it is not clear what is the actual QoS delivered to a particular connection under this regimen. Usually such QoS can only be characterized using a probability distribution, with a low (but existent) chance of severe service degradation for some connections.

LTM merges the flexibility in bandwidth allocation of ATM with the support for guaranteed QoS communications found in STM. As opposed to ATM or STM, where network resources have to adapt to traffic characteristics, in LTM traffic can adapt to network operations. That is, the network is in charge of informing sources about its current status so that sources may adapt its traffic generation accordingly. The unit of transfer in LTM is not pre-set to a fixed structure or size. QoS (delay, jitter, or loss) may be guaranteed or offer a controlled form of probabilistic guarantees. In addition, LTM offers synchronization signals that can be used by sources to schedule operations, thus minimizing buffering in the network.

By giving updated status information, LTM enables scheduling of traffic generation and thus may potentially minimize network buffering due to synchronization errors. Synchronous services may be achieved by local control exchange between the network and periphery nodes, without incurring end-end delays as is necessary in traditional protocol stacks.

5. CONCLUSIONS

This paper introduced a novel transfer mode: Loosely-synchronous Transfer Mode (LTM). LTM operates by signaling periphery nodes when destinations become available and encompasses advantages of both Synchronous Transfer Mode (STM) and Asynchronous Transfer Mode (ATM). Similarly to STM, synchronous communications with guaranteed QoS can be supported directly on LTM. Bandwidth allocation flexibility, one great advantage of ATM, can be found in LTM as well. Nevertheless, many of the problems introduced by STM and ATM are overcome by LTM: (1) no frame structure is necessary for communication; (2) traffic adapts to network status (instead of the other way around); (3) buffering in the network may be significantly lowered by correlating traffic generation to network status; (4) strict QoS is supported directly; and (5) synchronization signals are provided to the protocol stacks at periphery nodes.

Isochronets are candidate hardware infrastructures to implement LTM. Isochronets divide network bandwidth among routing trees and allocate periodic time intervals (bands) during which the trees are enabled. Routing is achieved by sending frames during bands in which the target destination is the root of an enabled tree. No frame-dependent processing is necessary to route frames in Isochronets, thus making their operations independent of any specific protocol stack and their implementation independent of any specific technology (such as electronic or optical).

The Synchronous Protocol Stack (SPS) is a novel protocol stack that uses LTM as its Media Access (MAC) mechanism. Because synchronization signals flow in SPS from LTM upwards to the application, SPS may incorporate protocols to support asynchronous, synchronous, and isochronous communications. Traditional transport layer protocols may be directly implemented in SPS. Additionally, such protocols can be extended to offer real-time and multicast services.

REFERENCES

1. Balraj, T. S., and Yemini, Y., "PROMPT—a destination oriented protocol for high-speed networks", in *Protocols for High-Speed Networks, II*, ed. M. J. Johnson, North Holland, 1990.
2. Boudec, J. Y. L., "Asynchronous Transfer Mode: a tutorial", *Computer Networks and ISDN Systems*, vol. 24, no. 4, May 1992.
3. Comer, D. E., *Internetworking with TCP/IP*, Volume I, Second Edition, Prentice Hall, 1991.
4. De Prycker, M., *Asynchronous Transfer Mode: solution for Broadband ISDN*, Second Edition, Ellis Horwood, 1993.
5. Florissi, D., "Isochronets: a high-speed network switching architecture (thesis proposal)", Technical Report CUCS-020-93, Computer Science Department, Columbia University, 1993.
6. Gerla, M., Tai, T.-Y., and Gallassi, G., "LAN/MAN interconnection to ATM: a simulation study", in *Proceedings of INFOCOM*, IEEE, 1992.
7. Mills, D.L., Boncelet, C.G., Elias, J.G., Schragger, P.A., and Jackson, A.W., "Highball: a high speed, reserved access, wide area network," Tech. Rep. 90-9-1, Electronic Engineering Dept., University of Delaware, 1990.
8. O'Malley, S. W., and Peterson, L. L., "A highly layered architecture for high-speed networks", in *Protocols for High-Speed Networks, II*, ed. M. J. Johnson, North Holland, 1990.
9. Tanenbaum, A. S., *Computer Networks*, Second Edition, Prentice Hall, 1988.
10. Yemini, Y. and Florissi, D., "Isochronets: a high-speed network switching architecture", in *Proceedings of INFOCOM*, IEEE, San Francisco, CA, USA, April 1993.
11. Zimmer, W., "FINE: a high-speed transport protocol family and its advanced service interface", in *Protocols for High-Speed Networks, III*, eds. B. Pehrson, P. Gunningberg, and S. Pink, North Holland, 1992.