Chapter 5

Protocols for Loosely-synchronous Stacks

5.1 Introduction

5.1.1 The Problem

The problem addressed in this chapter is how to extend existing protocol stacks to support real-time traffic, that is, synchronous or isochronous traffic. Traditional stacks are built to handle only asynchronous applications. New real-time applications, such as multi-media exchanges, demand synchronous or isochronous services from the protocol stack. Ideally, the new services must be provided as extensions of current protocols to maintain compatibility with existing systems.

5.1.2 Main Results

This chapter conveys two main contributions. Firstly, it describes a novel *Loosely-synchronous Transfer Mode (LTM)* that provides synchronization signals in order to support real-time services. Secondly, it explains how any protocol stack can be extended with synchronization signals from the LTM and become a *Synchronous Protocol Stack (SPS)*.

LTM. LTM is a novel transfer mode that informs periphery nodes about current network status, that is, destinations that are reachable and associated QoS. The information is supplied as *loosely-synchronous* signals. Upon reception of such signals, periphery nodes can best schedule their traffic streams according to current reachable destinations and necessary QoS.

The period between loosely-synchronous signals is a *band*. That is, each band represents reachable network destinations with associated QoS in terms of the end-to-end delay, jitter, and loss experienced in the communication.

LTM has many advantages when compared with other transmission modes:

- Sources and destinations are synchronized with the network through loosely-synchronous signals. That is, sources are synchronized with network bands (status) and not with frames or slots within frames as in Synchronous Transfer Mode (STM) [Halsall 92]. Bands last much longer than slots or frames. Consequently, the synchronization does not need to be as accurate as in STM but still can attain the same sort of synchronous service.
- QoS can be controlled and guaranteed because the loosely-synchronous signals coordinate how sources and network must interact. The signals convey information about current provided QoS that can be used by sources to best schedule their traffic needs. Guaranteed performance is essential for time-sensitive applications such as voice communications.
- Multiple protocol frames can be transferred without fragmentation and reassembly because there is no pre-defined frame structure in the communication. This simpli-

fies extensions of existing protocol stacks with the loosely-synchronous signals and interconnection of multiple stacks.

SPS. The SPS is any stack extended with the loosely synchronous signals provided by the LTM. An SPS forwards LTM synchronization signals through its layers up to the applications. In the SPS:

- Existing protocol stacks are supported unchanged. The new services can be implemented as orthogonal additions to the existing ones.
- Existing stacks are extended with novel source and destination synchronization. The synchronization can be used to implement novel real-time services at multiple stack layers, up to the applications.

5.1.3 Chapter Organization

The reminder of this chapter is organized as follows. Sections 5.2 and 5.3 introduce LTM and SPS respectively. Section 5.4 defines the LTM MAC services. Section 5.5 discusses how current transport protocols can be implemented and extended using LTM services. Section 0 shows an example of transport protocol services. Sections 5.7 and 5.8 discuss related and future work respectively. Finally, Section 5.9 summarizes.

5.2 Loosely-synchronous Transfer Mode (LTM)

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 re-

quire 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) [Halsall 92] and the Asynchronous Transfer Mode (ATM) [De Prycker 93, Le Boudec 92], 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. STM requires very tight synchronization that may be difficult to accomplish when transmissions occur at high speeds.

The primary function of a transfer mode is to move frames in a network. Toward this end, it needs to: (1) send and receive frames and (2) resolve contention for network resources such as intermediate links and buffers. In the context of HSNs, the transfer needs also to (3) provide QoS in the network.

In order to guarantee QoS, nodes must know how and when resources they use will be allocated to them. If this is not the case, the way in which the system resolves contention cannot be predicted and thus allocations cannot be guaranteed. For example, to deliver a frame every 33 ms in a video exchange, intermediate links in the path must be dedicated to transfer data frames from that connection every 33 ms.

There are two basic manners in which sources may be informed about resource allocation in the network. The first is the periphery-centered information. A periphery node requests resources and remembers how the allocation is structured. For example, in ATM resources are allocated beforehand. Based on the requests, the system decides how resource allocation is mediated among competing sources. Periphery nodes can derive how

their resources will be allocated based on their requests and on partial information about the resource allocation policy. If a node requests 1 Mb/s end-to-end connection, all it needs to do is remember to send information at this rate. Periphery-centered information works best when the network guarantees the allocations. Otherwise, requests do not match allocations and thus the information at the nodes is an assumption as opposed to a precise schedule on how resources are allocated.

The second method is network-centered information. Periphery nodes may still request resources, but the network notifies them when they become available, that is, nodes acquire status information from the network. This is the method used in the LTM. The network status embody information on the destinations that are reachable and on the associated QoS. It is periodically distributed to the nodes when the network it changes. Furthermore, sometimes resources do not need to be requested. For example, if only a route without QoS is necessary, sources can wait until the network status indicates that the route is available.

The network conveys its status through loosely-synchronous signals. The name reflects the fact that the signals are not necessarily synchronized with clocks, but with network status. Nevertheless, they still provide means to synchronize traffic motions. For example, a source may synchronize its generation of QoS-demanding traffic to a particular destination to the corresponding network signal.

Loosely-synchronous signals are issued by internal network switches and then forwarded to upper layers of the protocol stacks. Switches decide to issue these signals based

on network status changes. At periphery nodes, signals are captured by the network interface and forwarded through the periphery protocol stack, up to the application layer.

The period of time between loosely-synchronous signals is 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.

RDMA in Isochronets is an example of an LTM. The loosely-synchronous signals indicate the beginning of bands. The priority, contention, and multicast bands in Isochronets can exist in any LTM and this will be the assumption for the reminder of this chapter.

In reality, STM may be seen as an extreme example of LTM. STM is network-centered because it signals peripheries when frames and slots within frames begin. Nevertheless, STM does not exploit the full potential in LTM because it supports only dedicated resource allocation.

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-to-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. 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 2.4 Gb/s transmission rate lasts 3.3 ns. Typical bands last between a few hundreds of nanoseconds up to a few scores of microseconds because the network status (that is, routes and associated QoS provided by the network) is designed to change at this rate.

Similarly to ATM, diverse traffic classes may be serviced by LTM. Transfers of frames from a given class occur during 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 the network provides priority in the correct source and destination connection. Data traffic may be sent during any period in which the network enables the correct destination.

LTM may 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 network switches to the periphery nodes. The latter can use the signals to implement their protocol stacks.

5.3 The Synchronous Protocol Stack

Traditional protocol stacks were designed to support asynchronous (or non real-time) services. Two main events may trigger execution in traditional stacks: message arrival

from upper or lower layers and time-out signals. Time-out signals are usually generated by local clocks that may be out of synchronization with other clocks in the network while message arrivals are asynchronous in nature. It can thus be safely concluded that only asynchronous events fuel protocol motions in traditional stacks.

Synchronous and isochronous applications are difficult to build on traditional stacks. These applications have strict timing requirements that cannot be captured by asynchronous events. For example, a frame must be delivered every 33.3 ms in a video exchange. Such application requires dedicated network and stack resource for end-to-end transport without delays or jitter every 33 ms.

The challenge is to devise elementary functionality that could be added to a protocol stack to support real-time services. One possibility is to propagate loosely-synchronous signals through the stack. Such signals embody status information that can be used to synchronize transmissions at each layer in the stack. Such enhanced stack is an SPS.

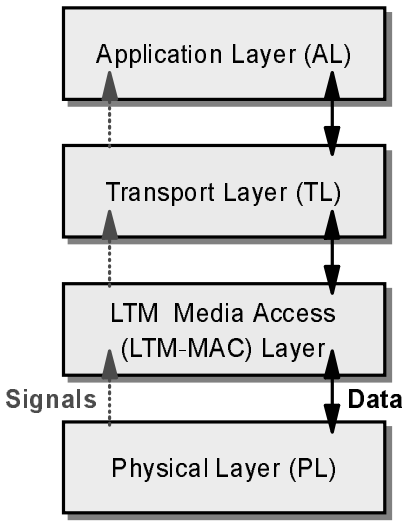

Figure 5.1. Synchronous Protocol Stack (SPS) structure

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 signals can be used at any layer, including the application, to implement synchronization functions. The general structure of an SPS is depicted in Figure 5.1.

The Physical Layer (PL) does not need to be bound to any special technology (i.e., electronic or photonic implementations may be employed) because LTM does not rely on any particular frame structure nor processes frame headers. The data link and network layers are collapsed into the LTM MAC (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.

Any traditional TL protocol currently used over STM or ATM may be used over the LTM-MAC directly with no changes other than adapting to the LTM-MAC SAPs (as explained later in Section 5.5.3). In addition, the signaling information from LTM-MAC can be used to enhance the functionality provided to provide real-time services (as explained later in Sections 5.5.4 and 5.5.5).

A feature unique to 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 that may then schedule its transmissions. For example, video traffic may be scheduled to generate frames when the proper band begins in each cycle.

In a more generic setting, SPS may be a portion of the overall protocol stack, as depicted in Figure 5.2. 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 portions. Synchronization of operations with transmissions can be guaranteed only below the SPS boundary. The figure illustrates a typical data path from protocol stack A to B, using the signaling from the LTM network. Notice that signaling is not passed above the SPS limit.

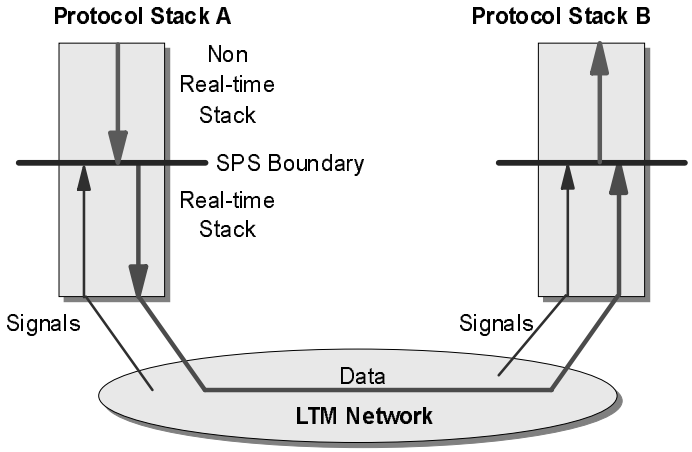

Figure 5.2. Handling signals in the protocol stack

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 machines and operating systems provide real-time support, that is, when it is possible to predict upper bounds on execution times of operating system calls. 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.

5.4 The LTM-MAC Service Access Points

This section summarizes the LTM-MAC SAPs. The LTM layer may be implemented using any technology as long as the LTM-MAC SAPs are kept unchanged. 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 5.3. 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. Other possible types are priority (p) and multicast (m). The *size* parameter is the length of the corresponding band in nanoseconds.

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 5.3. LTM-MAC SAPs

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.

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 destination or destinations (in the case of multicast bands). 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.

Notice that the naming structure for sources and destinations adopted so far is appropriate for the LTM-MAC layer because it deals with traffic that is already multiplexed. Upper layers of the protocol stacks may need do adopt new levels of naming to demultiplex the traffic streams. One example is to use ports at the transport level. The upper layer names are multiplexed (or de-multiplexed) to (or from) the LTM-MAC according to their protocol frame structures.

The manner in which the LTM-MAC is going to achieve band allocation is dependent on its internal operations. For example, contention bands can be allocated as a portion of the LTM supplied contention band to the given destination. To allocate priority or multicast bands, the signaling of the underneath transfer mode must be used to negotiate the allocation and inform all nodes involved. Priority bands can be 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 case 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*.

5.5 Transport Protocols

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

5.5.1 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.

One interesting issue is how to set such translation tables initially. That is, TL entities need to be able to find band identifiers connected to desired destination addresses. A variation of the ARP protocol [Comer 91] 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.

5.5.2 Mapping QoS onto LTM-MAC Bands

Transport protocol QoS also needs to be mapped into appropriate LTM-MAC band identifiers. Translation tables can be used to this effect as well. The fields in a translation table are band identifier and the associated QoS delivered by the band. This information can be acquired by broadcasting special request frames during all bands in a cycle, as explained in Section 5.5.1.

5.5.3 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 re-transmission. 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 re-transmission using the reverse band to the source.

The Internet Protocol. The Internet [Comer 91] transport protocols, User Datagram Protocol (UDP) and Transmission Control Protocol (TCP), are examples of asynchronous communication protocols. One way to support them on LTM, is to first implement the Internet Protocol (IP) layer on LTM. UDP and TCP both use the IP layer to implement their services.

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 are computed from frame destination addresses). When the beginning of an outgoing band is signaled to the IP layer, it forwards the respective buffered frames. Similarly, when the beginning of an incoming band to an IP entity is signaled, the entity receives the frames and forwards them to upper layer entities.

5.5.4 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 computed periodicity.

For example in a video transmission, if the cycle period is 125 μs, allocation can be implemented as follows. A priority band sized to fit one video frame can be allocated every 264 cycles (or 33 ms apart which is the sampling rate for video). Alternatively, a smaller priority band can be allocated for portions of the video 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 5.4. 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 that then goes to sleep waiting for the next signal. The signals from the network pace the application to generate isochronous frames.

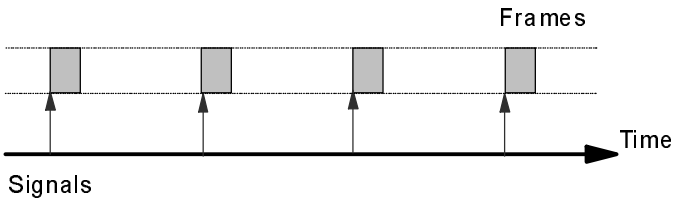

Figure 5.4. Isochronous transmissions

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 by the tolerable loss allowed by applications, as it is done in the context of ATM networks [De Prycker 93, Le Boudec 92]. That is, portions of the contention band for isochronous traffic are allocated not to guarantee loss-less communication, but to deliver low probability of loss. In this manner, the portion of the band to be allocated is smaller than what would be necessary to guarantee 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 that 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) [De Prycker 93, Le Boudec 92] protocols can be implemented using two alternative *ESTAB-*

LISH_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 protocol can be implemented.

The AAL 1 is intended to service constant bit rate applications such as uncompressed video transmissions. AAL 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 100 Mbits/s video transmissions, a band of size 3.3 Mbits can be allocated every 33 ms or a band of size 100 Kbits can be allocated every 1 ms.

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.

5.5.5 LTM Supports Synchronous Traffic

In *synchronous traffic*, it is necessary to guarantee maximum end-to-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 inter-process 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 5.5.4 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 toward providing synchronous services. For example, the IP suite can be extended with new SAPs to the application layer to support synchronous transport. Such SAPs can be implemented using priority bands at the LTM layer.

5.5.6 Compiling Higher Level QoS Parameters

Higher level QoS parameters such as end-to-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. This section presents an example of how the translations can be performed.

In the example Protocol 5.1, two parameters are used by the application layer to request transport layer services: maximum end-to-end delay and bandwidth needed (Step 1). The goal is to accommodate the request into a priority band. 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 value for P. If, after all values for P have been tried, no feasible allocation exists, the failure is communicated to the application.

Two observations are important in the example described. First, 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. Second, Protocol 5.1 is only one possibility for mapping end-to-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.

Protocol 5.1: *Translates higher level QoS parameters (end-to-end delay and bandwidth) into an LTM-MAC priority band.*

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 priority band assigned for the source and destination pair. The search is performed by requesting the LTM-MAC to allocate a portion (of the given 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 cycle 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 $E \geq T+O$ (where O is the overhead until the first transmission can happen), the allocation is feasible. Stop and inform the application. If $E < T+O$ or if E could not be found, 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.

Notice that the implementation of requests for QoS in terms of jitter and bandwidth can be accomplished using Protocol 5.1 by substituting the required maximum jitter for the maximum delay. For the jitter case, care must be taken to request a periodic allocation,

instead of a single allocation (where the period is computed according to maximum jitter demands).

5.6 Examples of Transport Layer SAP

This section illustrates how SPS concepts can extend traditional TL SAPs to implement new real-time services.

Example 1 illustrates how the socket [Stevens 90] SAP can be extended to provide novel real-time services. The piece of code on top uses the extended socket SAP to forward frames in the network. While the condition is not true, it sleeps on the socket waiting for a loosely-synchronous signal. The *socket_sleep(s_skt, wake_up_time)* operation makes the process sleep on socket *s_skt* until *wake_up_time* nanoseconds before the incoming band signal. When the process is awaked, it proceeds to forward the frame through the socket and goes back to generate the next frame. Then, it sleeps waiting for the next signal.

The *socket_sleep* operation is implemented using the LTM-MAC SAPs. It loops forever to receive incoming band signals. When it receives one, it checks to see if it is the correct band. If it is, it waits the period to the next signal (the band size) minus the requested wake-up time.

Notice that the *wait_out_band()* operation can be implemented in terms of the *OUT_BAND.signal(band, size)* LTM-MAC SAP.

There could be other implementations of the same operations.

Example 4.1: *Shows how the socket SAP can be extended to provide real-time services using loosely-synchronous signals.*

```
...
While (!condition) {
    ...   /* generate next frame */
    socket_sleep (s_skt, wake_up_time);
    send (s_skt, frame); }
...
socket_sleep (synch_socket s_skt, time wake_up) {
    while (true) {
        response = wait_out_band ();
        if (check (response, s_skt.name)) {
            sleep (s_skt.next - wake_up);
            return ; }}}
```

5.7 Related Work

5.7.1 LTM between ATM and STM

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) that can be used by sources to transmit information. Once allocated, bandwidth is guaranteed for the connection, thus delivering strict QoS in terms of

guaranteed end-to-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 in adopting STM for B-ISDN are the lack of flexibility in the slot size and in supporting on-demand service allocation. Applications such as voice communications 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. Additionally, STM lacks provision for asynchronous traffic (i.e., on-demand PS). When such traffic needs to access the network, slots must be allocated in the whole path from source to destination with unacceptable end-to-end allocation 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.

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-to-end de-

lays. The connection establishment phase in many applications may take longer than the transfer phase, which makes asynchronous communications inefficient both in end-to-end delays and in resource utilization. Some work [Gerla et al. 92] 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 virtual connection identifiers for all source/destination possibilities and may poorly use network resources. Synchronous communications are difficult due to the lack of synchronization at the ATM layer. Synchronization is relayed to upper layer protocols. But, the accuracy of such protocols is bound by the maximum end-to-end jitter, which may be difficult to predict exactly in ATM.

The QoS parameters 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 in terms of maximum delay, jitter, or loss may be guaranteed.

LTM enables scheduling of traffic generation to its signals and thus can 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-to-end delays as it is necessary in traditional protocol stacks.

5.7.2 The SPS and Real-time Protocols

This section positions SPS with respect to proposed protocols that support real-time communications such as the Real Time Protocol (RTP) [Schulzrinne et al. 94], the Internet Stream Protocol version II (ST-II) [Topolic 90], and the XTPX Transport Protocol [Metzler et al. 92].

The real-time portions of these protocols try to accomplish two main functions:

- Request QoS to the underlying transport protocols; and
- Monitor and control the QoS delivered.

The first function is responsible to map application demands into concrete QoS parameters such as bandwidth necessary and supported loss rate in the communication. The second manages QoS to detect periods when the transport fails to deliver the requested QoS. Usually, frame headers include time stamps and sequence numbers to detect QoS violations.

Any real-time protocol can be placed in the SPS with the added advantage that its functionality can be significantly reduced. That is, the management portion can be reduced significantly because LTM effectively guarantees the QoS in the communication.

5.8 Future Work

The next challenge is to design and incorporate the extensions suggested in this chapter into an existing protocol stack to build an SPS and then evaluate its performance and functionality. One possibility is to extend the IP stack on RDMA using the implemented Isochronet switch (see Chapter 6).

5.9 Summary

This chapter introduced a novel transfer mode: LTM. LTM operates by signaling periphery nodes when destinations become available with specific QoS guarantees. It encompasses advantages of both STM and 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.

The SPS is a novel protocol stack that uses LTM as its 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 supported unchanged in SPS. Additionally, such protocols can be extended to offer real-time and multicast services.