Chapter 2

Related Work

2.1 Introduction

2.1.1 The Goal

This chapter has two goals. First, it identifies the main desirable features that HSN Switching Architectures (HSNSAs) should posses in order to efficiently support application requirements. Second, it uses the defined features to characterize and compare proposed HSNSAs.

An HSNSA must be designed to take maximum advantage of high-speed transmissions while maintaining backward compatibility with existing technology. To best profit from high-speed transmissions, an HSNSA should:

- Minimize processing and queueing latency in the end-to-end transport path;
- Support speed-scalability for link bandwidth ranging from hundreds of megabits to terabits per second;
- Support scalability for large size networks (ranging from ten to hundred million end nodes);
- Support efficient interfacing with end nodes.

Toward backward compatibility, an HSNSA should support:

- End-to-end transport of multiple protocol frames;
- Efficient internetworking with existing stacks and media.

Additionally, high-speed transmissions enable a whole new set of applications with novel requirements. For these, it is important to support:

- Tunable guaranteed QoS of end-to-end transport;
- Asynchronous, synchronous and isochronous traffic streams;
- Efficient bandwidth sharing and flexible adaptation to service demands.

These features will be used to evaluate architectures for HSNs. The following sections start by analyzing the more traditional architectures: Circuit Switching (CS) [Halsall 92, Tanenbaum 88] and Packet Switching (PS) [Halsall 92, Tanenbaum 88]. These analyses are used as cornerstones to study the following HSN extensions of the basic architectures: Burst Switching (BS) [Amstutz 83, Haselton 83], Asynchronous Transfer Mode (ATM) [De Prycker 93, Le Boudec 92], Metanet [Ofek and Yung 90, Ofek 94], High-ball [Mills et al. 90], Wavelength Division Multiplexing (WDM) [Acampora and Karol 89, Brackett 91, Dono et al. 90], and Linear Lightwave Networks (LLNs) [Stern 90]. It will be concluded that these architectures may not fully support some of the features outlined.

HSN architectures that are not switched in nature will not be considered in this study. These architectures are usually restricted to local or metropolitan size networks. Examples include FDDI and DQDB [Bertsekas and Gallager 92].

2.1.2 Chapter Organization

This chapter is organized as follows. Section 2.2 further discusses the features to be supported by an HSNSA in descending order of importance. Sections 2.3, 2.3, 2.5, 2.6,

2.7, 2.8, 2.9, and 2.10 analyze the proposed architectures for HSNs. Finally, Section 2.11 summarizes.

2.2 Qualitative Characterization of HSNSAs

This section defines qualitative measures that will be used in evaluating HSNSAs. The measures capture how well an ideal HSNSA will support diverse application needs at the user interface to the network. The measures are the following:

1. *Support efficient end-to-end transport of multiple protocol frames.* On the one hand, QoS can be severely degraded if internal HSNSA operations are anchored on a single protocol suite. In such scenario, the adaptation between external protocols and the internal one may severely delay and complicate end-to-end communications. On the other hand, multiple protocols already inhabit today's interconnected networks providing services that cannot be eliminated after transition to high-speed transmissions. The ideal HSNSA must efficiently support the transmission of multiple protocols without adaptation between them.
2. *Support tunable guaranteed QoS.* Application needs range from best-effort service to guaranteed QoS. For example, electronic mail delivery applications can use best effort services at low cost while an embedded multiprocessor system must have strict QoS guarantees at higher cost. An HSNSA must provide flexibility in resource allocation to enable fine tuning of QoS needs given budgetary constraints.
3. *Support asynchronous, synchronous and isochronous traffic streams.* Future integrated network services include an array of synchronization needs, that is, asynchronous (e.g., electronic mail delivery), synchronous (e.g., embedded parallel processors), and isochronous (e.g., voice communications) streams. The provision of integrated services to higher layers may present difficulties if the underlying HSNSA is synchronous or asynchronous in nature. For example, if the HSNSA is

asynchronous, synchronization at higher layers will incur errors bound only by the maximum end-to-end jitter. It may be unacceptable for some applications. On the other hand, if the HSNSA is synchronous, the provision of asynchronous services may be inefficient or eliminate resource multiplexing advantages. The ideal situation is thus to directly support all synchronization needs in the HSNSA.

4. *Minimize processing and queueing latency.* Network processing is expected to be a major bottleneck due to the imbalance between processing and transmission speeds and thus must be minimized. For example, at 2.4 Gb/s transmissions, an ATM cell has to be processed in 177 ns. Additionally, network queueing delays are unpredictable and must happen under controlled supervision, with the option of being completely eliminated for very demanding applications. An ideal HSNSA should minimize processing and control queueing to cope with high-speed transmissions.
5. *Support speed-scalability for link bandwidth ranging from hundreds of megabits to terabits per second.* HSNSA operations cannot rely on any assumptions about link speed. For example, when it is necessary to process frame headers, the available processing time per frame will decrease with link speed. As a consequence, the architecture will not adapt once faster links are installed. An HSNSA must avoid any operation at link speed pace.
6. *Support efficient internetworking with existing stacks and media.* Current protocol stacks will continue to exist in future networks because a large portion of legacy software uses them. Additionally, an HSNSA should not rely on any particular medium. For example, an HSNSA must be capable to operate using electronic or optical transmissions to maintain compatibility with existing transmission technologies on which present protocol stacks rely. An HSNSA must provide compatibility with existing stacks and media either by directly supporting them or by efficient interfaces.

7. *Support scalability for large size networks (ranging from ten to hundred million end nodes).* HSNs are expected to grow rapidly because they will bring integrated communication services all the way to the subscriber premises (which in many cases are homes). An ideal HSNSA must thus support large networks.
8. *Support efficient interfacing with end nodes.* The interface to the network can be a major bottleneck in end-to-end QoS performance. For example, if the interface needs to allocate resources prior to any transmissions, the latency may become unbearable and the complexity in designing the interface may render costs prohibitive to some markets. An HSNSA must enable the implementation of efficient and relatively simple interfaces.
9. *Support efficient bandwidth sharing and flexible adaptation to service demands.* Bandwidth sharing is the driving force to lower cost services. Nevertheless, QoS can be severely compromised if sharing is not strictly controlled and adapted to changes. Resources can be multiplexed to provide lower cost services when the network load is low. Additionally, an ideal HSNSA must be able to adapt the degree of multiplexing to protect QoS demanding applications from sudden load changes.

In the following sections, proposed HSNSA will be evaluated with respect to these measures. The number associated with each measure in this section will be used as a reference to it in the following sections.

2.3 Packet Switching

PS or store-and-forward networks operate by fragmenting information steams into variable size frames (or packets). Frames are equipped with special control headers that embody routing information such as destination addresses. PS is commonly used in data

networks, such as the Internet [Comer 91].

Switching in PS works as follows. Nodes in the path parse each incoming frame to extract the destination addresses. These addresses are then used to index a routing table that maps destination addresses into respective node output port identifiers. These parsing and mapping functions are performed by the processor at each node. It is in effect a shared resource among all frames crossing the node. The union of all the routing tables in network nodes form a set of routing trees, that is, a spanning tree that defines paths from all the sources in the network to a particular destination (the root of the tree) for each destination.

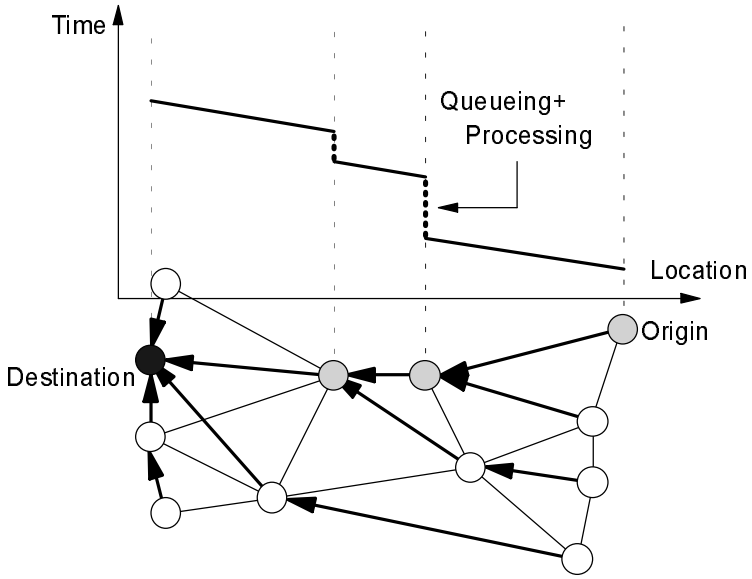

Figure 2.1: Store-and-forward routing on trees

Figure 2.1 highlights one such routing tree. The location and time diagram for a typical frame moving along the routing tree is depicted on top of it. Each point in the graph represents the location and time of the frame motion. Initially, the frame incurs some transmission and propagation delay while moving to the next node in the path to the destina-

tion. When it reaches the next node, the frame contends for resources such as processor and outgoing link. It may thus incur some random motionless delay while waiting for these resources and while being processed. This period is depicted as a vertical line on top of the node. Eventually the frame moves to the next node in the path where similar disrupted motions take place.

It is important to notice that contention happens among all frames crossing the node (to different destinations) because they must share the same processor. Thus, the random queueing delay is dependent not only on contention due to the frames sent to one particular destination, but also on other frames crossing the node.

In PS:

1. Muti-protocol support is not efficient because it involves cost intensive operations to adapt between multiple external frame structures and the internal one. These operations usually incur fragmentation and reassembly of frames and require considerable processing. As the link speeds increase, the time to perform such operations becomes more and more limited.
2. QoS cannot be guaranteed or tuned. Upon contention for network resources such as processor or outgoing links, frames incur unpredictable queueing delays or eventual loss when queues overflow. As a consequence, end-to-end delays, jitters, or effective bandwidth use are difficult to predict. One partial solution to support QoS in PS is to suppress the very sources of traffic randomness via global admission control and policing mechanisms [Sidi et al. 89]. During the admission phase, resources are requested with associated QoS parameters. For example, bandwidth may be requested with some maximum allowed end-to-end delay and loss rate. These parameters are used by the underlying transport layer to allocate network resources (bandwidth, buffers, processing, etc.). During the policing phase,

sources are inspected to see if they are generating traffic that complies with the requested QoS parameters. For example, if the source requested a specific bandwidth, it cannot transmit over the agreed parameter. Admission delays and reduced network utilization are traded-off against reduced contention. Nevertheless, QoS can only be statistically guaranteed. The interruptions seen by a source depend on aggregated contention traffic of other random sources. For example, all sources contend for the processor and thus the queue at the processor contains an interleaved mixture of frames from all sources. Statistical QoS results in increased admission delays, reduced network utilization, and lower effective bandwidth seen by sources. If contention is to be eliminated with high probability, all resources need to be pre-allocated and consequently the very value of PS for on-demand resource allocation becomes questionable.

3. Asynchronous services can be provided directly, but limited support exists for synchronous or isochronous services. Synchronization can only be provided by upper protocol layers and is thus limited by the maximum end-to-end jitter in the network. Since in PS no such bound exists, synchronization is difficult to attain.
4. Processing takes place per frame. To switch, frame headers need to be parsed in order to extract destination addresses. Processors use these addresses to index routing tables and find the proper outgoing node port in the route to the destination. Additionally, contention for network resources such as processors and links is resolved by queueing frames. Such queueing delays are unbound and can contribute significantly for end-to-end delays.
5. Switching speed may represent a problem for speed scalability. When link speeds are increased, processing speeds must be increased accordingly. Nevertheless, processing technology may not be capable to cope with recent link speed. For example, at 2.4 Gb/s, a 53 byte packet has to be processed in 177 ns or about 17 instructions on a 100 MIPS machine.

6. Internetworking is complex due to the difficulty of supporting multiple protocols. Additionally, PS is difficult to adapt to non electronic media. Implementation of PS in all-optical networks would involve all-optical frame header processors. The technology to implement such processors is current beyond the state of the art.
7. Scalability in size is better supported than in other architectures. Frames allocate resources on-demand strictly to use them. That is, at any given time, the network links and processors are being used by the frames at the head of the queues. When the queue is empty, links and processors are ready for on-demand use. Since resources are never allocated to idle sources, this architecture is likely to support more nodes than others given a constant set of resources.
8. The interface is simple to implement. End nodes can directly relay frames at any time without resource pre-allocation. Additionally, since there is no need to pre-allocate resources, frame delay to access to resources is minimal when network load is low (because there is minimal queueing overhead due to contention in this case).
9. Bandwidth sharing is one of the most appealing features. Resources are shared and only allocated on-demand when necessary. Potentially, resource wastage does not exist in PS.

2.4 Circuit Switching

In CS, a physical connection is established between the source and destination nodes. Each link in the network is multiplexed as follows. The physical layer generates a periodic frame (not to be confused with PS or higher layer data frames) that is divided into slots that are then assigned to connections. Slots are usually 8 bits long and the frame is repeated at 8 KHz, providing a throughput of 64 Kb/s per connection. CS is normally used

for voice communication. A connection is established by mapping idle incoming slots into idle outgoing slots in each node in the source to destination path. This switching techniques also called Time Division Multiplexing (TDM).

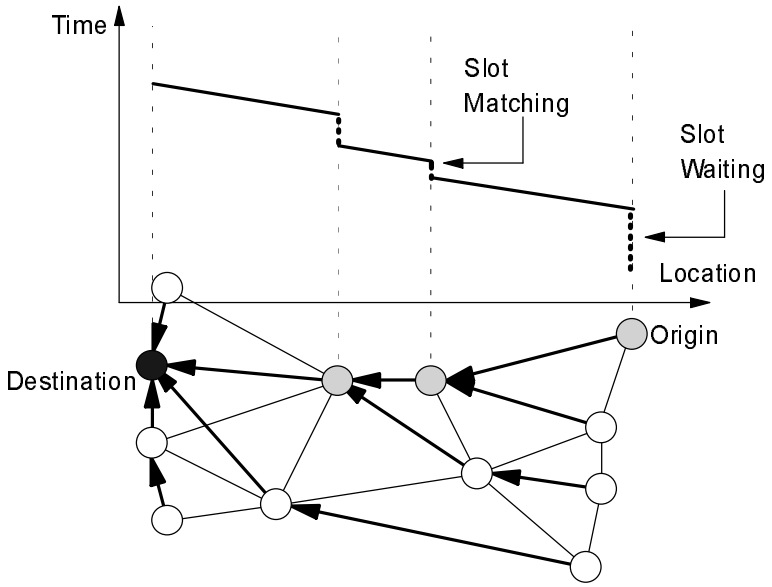

Figure 2.2: Circuit motion on trees

Switching at intermediate nodes is accomplished by mapping incoming ports and slots onto outgoing ports and slots, regardless of information content, as depicted in Figure 2.2. The delay at each node depicted in the figure is to match incoming and outgoing slots and is bound by the maximum frame size. Extensions of the CS concept [De Prycker 93] have been proposed to overcome some of the drawbacks to be described in this section, but they greatly increase the implementation complexity.

In CS:

1. Frames are transported transparently, that is, without header processing at internal switches. As a consequence, multiple protocol frames can be simultaneously switched without adaptation to a common internal frame structure.

2. QoS is strictly guaranteed because frame slots are dedicated to sources and destinations for the duration of the connection. Nevertheless, it cannot be finely tuned for applications that do not have strict QoS demands. Each communication uses dedicated resources and consequently has full QoS guarantees.
3. On the one hand, synchronous and isochronous communication streams are supported directly by the architecture. Signaling from frames and slots within frames can be used to synchronize sources and destinations. On the other hand, asynchronous communications have to be emulated within a synchronous infrastructure, with unnecessary added delays for media access and potential loss of resource multiplexing.
4. There is no frame header processing. Switching occurs solely by matching incoming and outgoing link frame slots. Additionally there is no queueing inside the network because slots are dedicated to the connection.
5. Link speed increase may generate problems in synchronization and switching speed. For example, if the slot size is 8 bits and transmissions happen at 1 Gb/s, each slot lasts 8 ns. In such scenario, peer nodes in a link must synchronize and switch slots within nanoseconds. A potential solution is to increase the slot size but this will incur bandwidth waste since many applications such as voice conversations will occupy only a few bits per frame. Additionally, the circuit reservation delays constitute a significant factor in the end-to-end communication efficiency when links are fast. For example, the protocol exchanges to allocate the circuits may incur end-to-end propagation delays in the order of hundreds of milliseconds when sources and destinations are far apart. Nevertheless, an exchange of 1 Kbyte of information at 1 Gb/s takes only 8 μs.
6. Internetworking with existing stacks is simple due to the support for multiple protocol transport. That is, multiple protocols can coexist on the same medium without adaptation. Additionally, TDM may function in multiple media, ranging from

all-optical to all-electronic networks.

7. The increase in the number of nodes is a potential problem. The more end-nodes exist in the network, the richer the network connectivity and bandwidth must be to support end-to-end dedicated channels.
8. The interface is relatively complex to implement because it needs to allocate and maintain circuits prior to communication. It also must be fast to synchronize communications to frame slots at high speeds.
9. Bandwidth sharing or adaptation does not exist. Slots are dedicated to end-to-end connections and may not be used when the corresponding source is idle.

2.5 Fast Packet Switching and Asynchronous Transfer Mode

Many research efforts have been undertaken in recent years to develop Fast Packet Switching (FPS) architectures [Ahmadi and Denzel 89, Chao 91, Eng et al. 89, Giacopelli 91, Huang and Knauer 84, Lee 90, Murakami 91, Oie et al. 91, Pattavina 90, Tobagi 90, Tobagi et al. 91, Turner 86, Turner 88, Venkatesan 92, Widjaja and Leon-Garcia 92, Yeh et al. 87, Yum and Leung 92] of which Asynchronous Transfer Mode (ATM) is a representative example. FPS uses a combination of PS with virtual connections to transfer information. The idea is to attach stochastic QoS guarantees to connections while switching according to PS. Connections are monitored and controlled to deliver the negotiated QoS. ATM operations will be described as a representative architecture of FPS.

ATM networks transmit 53 byte fixed size cells structured as 5 byte header and 48 byte payload. ATM switches cells using Virtual Paths (VPs) and Virtual Circuits (VCs). A VP is a channel that may contain one or more VCs. Each ATM cell contains two identifiers: a VP Identifier (VPI) and a VC Identifier (VCI). Within a VP, switches use

only VPIs in each cell to take switching decisions. When a switch connects different VPs, both VPIs and VCIs are used. Prior to transmission, a VC must be established.

ATM can be regarded as a combination of PS with added CS characteristics. Switching per se operates like in PS. Nevertheless, a connection must be established prior to transmission like in CS and cells in the same connection follow the same path to the destination. ATM inherits many of its operational characteristics from the combination of PS and CS:

1. Multiple protocol transport may be inefficient and difficult to implement. It involves fragmentation of frames into cells and their transmission over end-to-end VCs when the latter is available for the requested end-to-end connection. If VCs are not available, frames may need to cross multiple VCs with corresponding fragmentation and reassembly at each router in the path further compromising efficiency.
2. ATM provides only a coarse level of QoS guarantees as far as cell structure goes: cells are either high or low priority. Guarantees can be provided only in a probabilistic sense because the underlying technology is PS with characteristic random behavior due to cell queueing and loss. For bursty traffic scenarios guarantees require worst-case input admission control to avoid internal cell queueing or loss. Conservative admission control mechanisms may lead to significant waste of bandwidth.
3. Asynchronous services can be provided directly, but limited support can be given for synchronous or isochronous services due to the intrinsic end-to-end jitter in ATM operations.
4. In similarity to PS, processing takes place per cell and queueing is used to overcome contention.
5. In similarity to PS, speed scalability support may generate problems in switching

speed. To attenuate this problem, many FPS switch implementations have been proposed (as the ones referenced in the first paragraph of this section). Many times they use multiple specialized processors or special interconnection fabrics. Nevertheless, they significantly add to the implementation complexity and cost and may still be difficult to scale due to the inherent processing per cell.

6. In similarity to PS, interconnection is complex.
7. Scalability with number of nodes is somewhere between CS and PS. On one hand, if QoS must be guaranteed for all connections, independent VCs must be allocated with no resource sharing. On the other hand, if no guarantee is necessary, VCs may share resources and thus maximize multiplexing as it is the case in PS. Scalability is dictated by the number of VCIs that can be allocated and the QoS that must be provided to end nodes.
8. In similarity to CS, transfers need to be preceded by a VC establishment. Transmissions can then proceed using the VCI. Nonetheless, transmissions need to cope with the agreed QoS parameters. A policing mechanism monitors and controls violations. The implementation of such policing mechanisms may further complicate the interface.
9. Bandwidth sharing happens between QoS demanding and non-QoS demanding traffic by using the priority mechanism in cells. It is thus limited to a two-level mechanism.

2.6 Burst Switching

Burst Switching (BS) is an extension of the PS concept to switch bursts of information, usually integrating digitized voice and data. Bursts are generated from speech spurts or data messages and are embedded into a frame (or burst) with a header that identifies its destination address and a trailer (because bursts have variable size). The novel aspect of

BS networks is that they disperse switching decisions into hundreds and thousands of processors connected through shorter link lengths (thus permitting higher bandwidth links). The links use TDM. When a burst is sent, one of the TDM slots is allocated to the burst.

BS is thus very similar to PS with very fast switches and TDM links. The following highlights only the points where BS and PS differ.

2. BS can guarantee QoS within bursts, but not between bursts. Additionally, to support voice communications, the switches are made very fast by distributing processing and thus increasing processing throughput to a level that minimizes contention.
3. BS supports synchronization within bursts by using TDM signals.
5. Even though fundamentally not very scaleable as it is the case with PS, link speed increase can be better accommodated since header processing happens per burst (which is larger than a frame).

2.7 MetaNet

The MetaNet network operates by embedding ring topologies within arbitrary networks. MetaNet uses a type of deflective routing algorithm through the rings by which packets are forwarded following the ring structures. The same algorithms for media access used for ring topologies are employed in the MetaNet to accomplish fairness, deadlockless, and dynamic self-routing.

The MetaNet operates essentially as a ring network and thus incorporate many of the characteristics of PS networks. Here are a few more specific observations:

2. QoS can be difficult to guarantee. Statistical QoS can be accomplished only through the addition of special features to provide virtual circuit allocation in the network. These mechanisms are orthogonal to MetaNet operations. Tuning of QoS seems not to be possible.
3. Asynchronous services can be provided directly, but synchronous or isochronous services can only be provided in limited fashion as in ATM networks. The underline transport does not contain synchronization signals.
4. Processing takes place per packet at each node to check if the packet is destined to the current node. Queueing is needed to solve contention for network resources such as processors and links.
5. Link speed scalability support may generate problems in switching speed. This may be attenuated by the fact that the kind of processing in a ring topology can be completely realized in hardware.
7. Scalability with number of nodes is supported by the introduction of clusters of rings to improve overall available bandwidth.
8. End nodes have to perform ring media access functions that can be quiet elaborate.
9. Bandwidth sharing only happens for asynchronous traffic, since the synchronous traffic must use separated virtual circuits.

2.8 Highball

In the Highball network proposal, nodes schedule traffic bursts by configuring switches to support uninterrupted motion similar to train motions through intersections. Nodes broadcast requests to all other nodes, specifying their data transmission needs to all

possible destinations. This information is then used to compute a train schedule at each node and establish time intervals during which output links are dedicated to specific input links. The scheduling problems are NP-complete and are thus approximated through heuristic solutions. Highball networks are geared to serve traffic that can tolerate the latency delays between requests to transmit and their granting.

Highball networks are similar to CS in that no processing of frames occurs in the network. Nonetheless, they are more flexible because bandwidth allocation is done according to traffic needs and is not bound to a fixed slot. The following highlights the main points where the Highball architecture and CS differ:

2. QoS is always guaranteed, but there is some flexibility in that bandwidth can be allocated according to traffic demands.
5. Speed scalability has less impact than in CS because slots assigned to particular end-to-end connections in Highball have variable size and tend to be much bigger than the CS fixed size slot.
7. Network growth can increase resource allocation complexity. Since the scheduling problems are NP-complete, network size may slow the resource allocation procedures.
9. Bandwidth allocation in Highball is more flexible and can very closely mimic usage patterns, thus decreasing the probability of wasting bandwidth assigned to idle sources.

2.9 Wavelength Division Multiplexing

Wavelength Division Multiplexing (WDM) networks provide dedicated access to des-

tinations via appropriate allocation of wavelengths. Routing is accomplished by configuring nodes to switch wavelengths in order to provide source-destination connectivity. Contention among simultaneous transmissions to the same destination must be solved at switches. Unfortunately, optical tuning of switches at incoming traffic rates is beyond the current state of the art. To cope with this limitation, current implementations of switched WDM use dedicated wavelengths between node pairs. Packets may only be sent to a node's peer. At the peer, packets need to be processed (and thus be converted to the electronic domain) in order to determine the destination route. WDM may operate either in CS or PS modes. Nevertheless, multiple hop (switched) versions usually employ PS. This will be the assumption in this section.

The big advantage in WDM is that the optical bandwidth available is much larger than the transmission speeds achievable by end nodes. Multiple channels may be created using different colors. In fact, the network topology may be completely changed by simply tuning optical transmitters and receivers to build new links using colors. Each color corresponds to a different link. The following highlights the main differences between WDM and PS:

4. In addition to normal frame header processing, WDM network need to perform optical to electronic frame conversion which may significantly delay communications in the current state of the art.
6. WDM relies on optical technology and thus internetworking must incur conversions between optics and electronics.
7. The optical bandwidth is much larger than the traffic electronic end nodes may

generate and thus can be divided in many wavelengths that can be used as multiple independent virtual networks. As a consequence it can support a larger number of end nodes.

8. The interface must perform media conversion between optics and electronics which may involve complex circuitry.

2.10 Linear Lightwave Networks

Linear Lightwave Networks (LLNs) provide all-optical communication using wavebands and wavelengths within wavebands. Wavebands are used to create various channels in each link while wavelengths are used to provide end-to-end connectivity. Optical switches can switch only wavebands (wavelengths are indistinguishable for them), while end nodes can separate wavelengths within wavebands. All switching operations are optical. They are realized by generating power at an output ports from a linear combination of the powers at the input ports. Wavebands are broadcast through all possible paths that carried wavelengths need to traverse. When accepting a new call, a wavelength and a path to the destination must be assigned it. The procedure toward this goal is to examine each link and verify that the candidate wavelength is not being broadcast through it.

It is important to notice that there are situations in which it may not be possible to assign the same wavelength to two distinct pairs of nodes even if they have disjoint paths. The reason is that LLNs have to broadcast wavebands (and consequently wavelengths within them) through all paths the wavebands connect. As a consequence, the assignment of different wavelengths for connections may become intricate. It is necessary to make sure that all combined wavelengths in a given link do not interfere with the new wave-

length that is being assigned. The assignment of wavelengths to incoming calls is NP-complete and needs to be approximated using heuristics.

LLNs also divides bandwidth in the waveband domain, in similarity to WDM. Nevertheless, no processing or buffering is necessary at intermediate switches. It is designed to serve applications that can tolerate the potentially long call set up delay to find a proper wavelength for the call.

LLNs operate in a way very similar to CS. The main departure from CS occurs due to the large bandwidth available in the optical domain, due to the bandwidth division among wavelengths (and not slots), and due the fact that all internal switching is optical. Here are some more specific differences:

3. Synchronous and isochronous communication streams can be supported because the network guarantees the communication QoS. Nevertheless, there is no signaling provided by the network.
5. Link speed scalability is not a problem in LLNs because bandwidth is divided in wavelengths as opposed to CS time slots.
6. LLNs must operate in the optical domain and thus electronic conversion is necessary for interconnection.
7. The network can grow due to the enormous bandwidth available in the optical domain.
8. The interface must perform media conversion between optics and electronics that may involve complex circuitry. Additionally, the allocation of wavelengths may complicate the interface design.

2.11 Summary

Proposed HSNSAs have been evaluated using a series of important qualitative measures. Figure 2.3 summarizes the main conclusions. Each column represents an HSNSA and each row represents a measure. A gray mark is placed only at the intersection of HSNSAs that fully support the corresponding measure.

Many measures cannot easily be satisfied by current architectures. The most interesting findings are that:

- No architecture fully support both guaranteed and tunable QoS;
- No architecture has explicit support for the three types of synchronization needs;
- No architecture supports the whole set of measures defined.

The next chapter develops a new architecture for HSNs that incorporates each of these features.

	Circuit Switching	Packet Switching	Fast Packet Switching	Burst Switching	MetaNet	Highball	Wavelength Division Multiplexing	Linear Lightwave Networks
Efficient multiple protocol frames	■					■		■
Tunable guaranteed QoS								
Asynchronous, synchronous, and isochronous streams								
Minimal processing/queueing	■					■		■
Speed scalability						■	■	■
Internetworking with existing stacks/media	■					■		
Size scalability		■	■	■	■		■	■
Efficient interfacing		■		■				
Bandwidth sharing		■	■	■	■		■	

Figure 2.3: Characteristics of proposed high-speed architectures