Optical Communication Networks

EE654

Lectuer - 7

Spring 2016

Photonic Packet Switching

The optical networks that we have studied so far provide circuit-switched services. These networks provide lightpaths, which can be established and taken down as needed. In these networks, the optical nodes do not switch signals on a packet-by-packet basis, but rather only switch at the time a circuit is established or taken down. Packet switching is done in the electronic domain by other equipment such as IP routers or Ethernet switches. These routers and switches make use of lightpaths provided by the optical layer to establish links between themselves as needed. In addition to switching packets, routers and switches make use of sophisticated software and hardware to perform the control functions needed in a packet-switched network.

In this chapter, we will see that all the building blocks needed for optical packet switching are in a fairly rudimentary state today and exist only in research laboratories. They are either difficult to realize, very bulky, or very expensive, even after a decade of research in this area. Moreover, it is likely that we will need electronics to perform the intelligent control functions for the foreseeable future. Optics can be used to switch the data through, but it does not yet have the computing capabilities to perform many of the control functions required, such as processing the packet header, determining the route for the packet, prioritizing packets based on class of service, maintaining topology information, and so on.

However, there are a few motivations for researching optical packet switching. One is that optical packet switches hold the potential for realizing higher capacities than electronic routers (although this potential is yet to be demonstrated!). For instance, the capacity of the best routers today is less than 100 Tb/s, with the highest-speed interfaces being at 40 Gb/s. In contrast, optical switches are, for the most part, bit rate independent, so they can be used to switch tens to hundreds of Tb/s of traffic.

Another motivation for studying optical packet switching is that it can improve the bandwidth utilization within the optical layer. The notion is that high-speed optical links between routers are still underutilized due to the bursty nature of traffic, and using an underlying optical packet layer instead of an optical circuit layer will help improve link utilizations. The question is whether having another high-speed packet-switched layer under an already existing packet-switched layer (say, IP) will provide sufficient improvement in statistical link utilization. The answer depends on the statistical properties of the traffic. The conventional wisdom is that because many lower-speed bursty traffic streams are multiplexed through many layers, the burstiness of the aggregate stream is lower than that of the individual streams.

In this case, having an optical packet layer under an electrical packet layer may not help much because the traffic entering the optical layer is already smoothed out. However, it has been shown recently that with some types of bursty traffic, notably the so-called self-similar traffic, the burstiness of a multiplexed stream is not less than that of its constituent individual streams [PF95, ENW96]. For such traffic, using an optical packet layer provides the potential to improve the link utilization.

Figure 12.1 shows a generic example of a store-and-forward packet-switched network. In this network, the nodes *A*–*F* are the switching/routing nodes; the end nodes 1–6 are the sources and sinks of packet data. We will assume that all packets are of fixed length. Packets sent by an end node will, in general, traverse multiple links and hence multiple routing nodes, before they reach their destination end node. For example, if node 1 has to send a packet to node 6, there are several possible routes that it can take, all consisting of multiple links and routing nodes. If the route chosen for this packet is 1–*A*–*B*–*D*–*F*–6, this packet traverses the links 1–*A*, *A*–*B*, *B*–*D*, *D*–*F*, and *F*–6. The routing nodes traversed are *A*, *B*, *D*, and *F*. Note that the route chosen may be specified by the packet itself, or the packet may simply specify only the destination node and leave the choice of route to the routing nodes in its path. In the remainder of the discussion, we will assume that the route is chosen by the routing nodes based on the packet destination that is carried in the packet header.

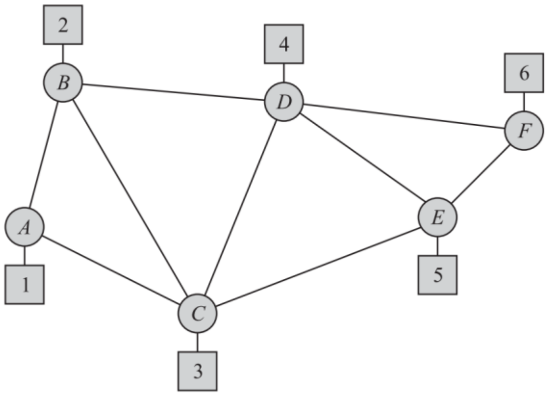

Figure 12.1 A generic store-and-forward network.

Figure 12.1 is also the block diagram of a PPS network. The major difference is that the links run at very high speeds (hundreds of gigabits per second) and the signals are handled mostly optically within each routing node.

Figure 12.2 shows a block diagram depicting many of the functions of a routing node, or router. In general, there is one input from, and one output to, each other routing node and end node that this routing node is connected to by a link. For example, in Figure 12.1, routing node *A* has three inputs and outputs: from/to

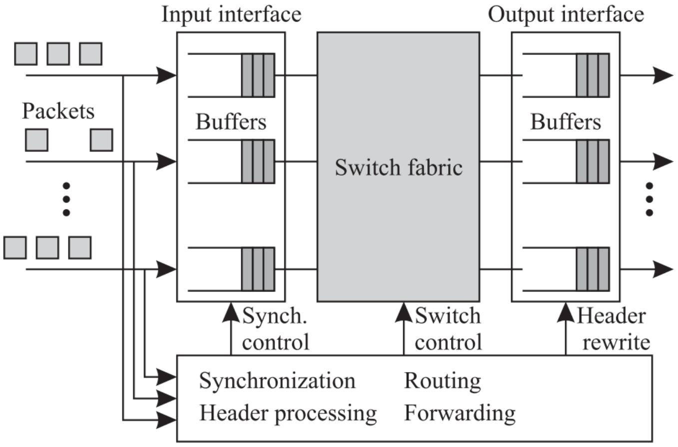

Figure 12.2 A routing node in the network of Figure 12.1.

routing node B, routing node C, and end node 1. Similarly, routing node C has five inputs and outputs. Routers perform the following functions (see Section 6.5 for a more detailed description of how these functions are performed by IP routers):

Routing. Routers maintain up-to-date information of the network topology. This information is maintained in the form of a routing table stored at each node.

Forwarding. For each incoming packet, a router processes the packet header and looks up its routing table to determine the output port for that packet. It may also make some changes to the header itself and reinsert the header at the output.

Switching. Switching is the actual process of switching the incoming packet to the appropriate output port determined by the forwarding process. The hardware that does the switching is often called the switch fabric, as shown in Figure 12.2.

In Chapter 3 we discussed the technologies that can be used to implement a switch fabric. An example of a switch fabric is shown in Figure 12.3. Its input ports are attached to a stage of tunable wavelength converters (TWCs), followed by an arrayed waveguide grating (AWG) and then another stage of wavelength converters (WC), which are followed by the output ports. To switch a packet from an input port to an output port, the input port's TWC has its outgoing wavelength tuned so that the packet will be routed through the AWG to the packet's output port. (See Figure 3.25 for a description of how signals are routed through the AWG depending on their wavelengths.) The WCs at the output ports have their outgoing wavelengths fixed. The switch fabric's switching speed is limited only by the switching speed of the TWCs. For optical packet transmissions in the 10s of Gb/s or higher, the switching speed should be in the nanosecond range or lower.

Synchronization. Synchronization can be broadly defined as the process of aligning two signal streams in time. In PPS networks, it refers either to the alignment of an incoming pulse stream and a locally available *clock* pulse stream or to the relative alignment of two incoming pulse streams. The first situation occurs during multiplexing and demultiplexing, and the second occurs at the inputs of the router where the different packet streams need to be aligned to obtain good switching performance.

PPS networks will have to perform all the functions described above. Some of these functions involve a fair amount of sophisticated logic and processing and are still best handled in the electrical domain. The routing and forwarding functions, in particular, fit into this category. To date, most PPS proposals assume that the packet header is transmitted separately from the data at a lower speed and processed electronically. We will, however, study some of the approaches to provide at least rudimentary header processing in the optical domain.

Because of technological constraints, it is quite difficult to perform even the remaining functions of switching, buffering, multiplexing, and synchronization in the optical domain. This will become clearer as we explore the different techniques for performing these functions. Therefore, PPS networks are at this time still in research laboratories and have not yet entered the commercial marketplace. To simplify the implementation, especially the control functions, many PPS proposals also assume the use of *fixed-size* packets, and we will make the same assumption in this chapter. Of course, in reality we have to deal with varying packet sizes. If a fixed packet size is used inside the network, then the longer packets will have to be segmented at the network inputs and reassembled together at the end. Alternatively, we could design the PPS nodes to switch variable-sized packets, a more complex proposition.

7.1 Optical Time Division Multiplexing

At the inputs to the network, lower-speed data streams are multiplexed optically into a higher-speed stream, and at the outputs of the network, the lower-speed streams must be extracted from the higher-speed stream optically by means of a demultiplexing function. Functionally, optical TDM (OTDM) is identical to electronic TDM. The only difference is that the multiplexing and demultiplexing operations are performed entirely optically at high speeds. The typical aggregate rate in OTDM systems is on the order of 100 Gb/s, as we will see in Section 12.6.

OTDM is illustrated in Figure 12.4. Optical signals representing data streams from multiple sources are interleaved in time to produce a single data stream. The interleaving can be done on a bit-by-bit basis as shown in Figure 12.4(a). Assuming the data is sent in the form of packets, it can also be done on a packet-by-packet basis, as shown in Figure 12.4(b). If the packets are of fixed length, the recognition of packet boundaries is much simpler. In what follows, we will assume that fixed-length packets are used.

In both the bit-interleaved and the packet-interleaved case, *framing pulses* can be used. In the packet-interleaved case, framing pulses mark the boundary between packets. In the bit-interleaved case, if n input data streams are to be multiplexed, a framing pulse is used every n bits. As we will see later, these framing pulses will turn out to be very useful for demultiplexing individual packets from a multiplexed stream of packets.

Note from Figure 12.4 that very short pulses—much shorter than the bit interval of each multiplexed stream—must be used in OTDM systems. Given that we are interested in achieving overall bit rates of several tens to hundreds of gigabits per second, the desired pulse widths are on the order of a few picoseconds. A periodic train of such short pulses can be generated using a mode-locked laser, as described in Section 3.5.1, or by using a continuous-wave laser along with an external modulator, as described in Section 3.5.4. Since the pulses are very short, their frequency spectrum will be large. Therefore, unless some special care is taken, there will be significant pulse broadening due to the effects of chromatic dispersion. For this purpose, many OTDM experiments use suitably shaped return-to-zero (RZ) pulses, which we studied in Sections 2.6 and 4.1.

Assume that n data streams are to be multiplexed and the bit period of each of these streams is T. Also assume that framing pulses are used. Then the interpulse width is $\tau = T/(n+1)$ because $n+1$ pulses (including the framing pulse) must be transmitted in each bit period. Thus the temporal width τ_p of each pulse must satisfy $\tau_p \leq \tau$. Note that usually $\tau_p < \tau$, so that there is some guard time between successive pulses. One purpose of this guard time is to provide for some tolerance in the multiplexing and demultiplexing operations. Another reason is to prevent the undesirable interaction between adjacent pulses that we discussed earlier.

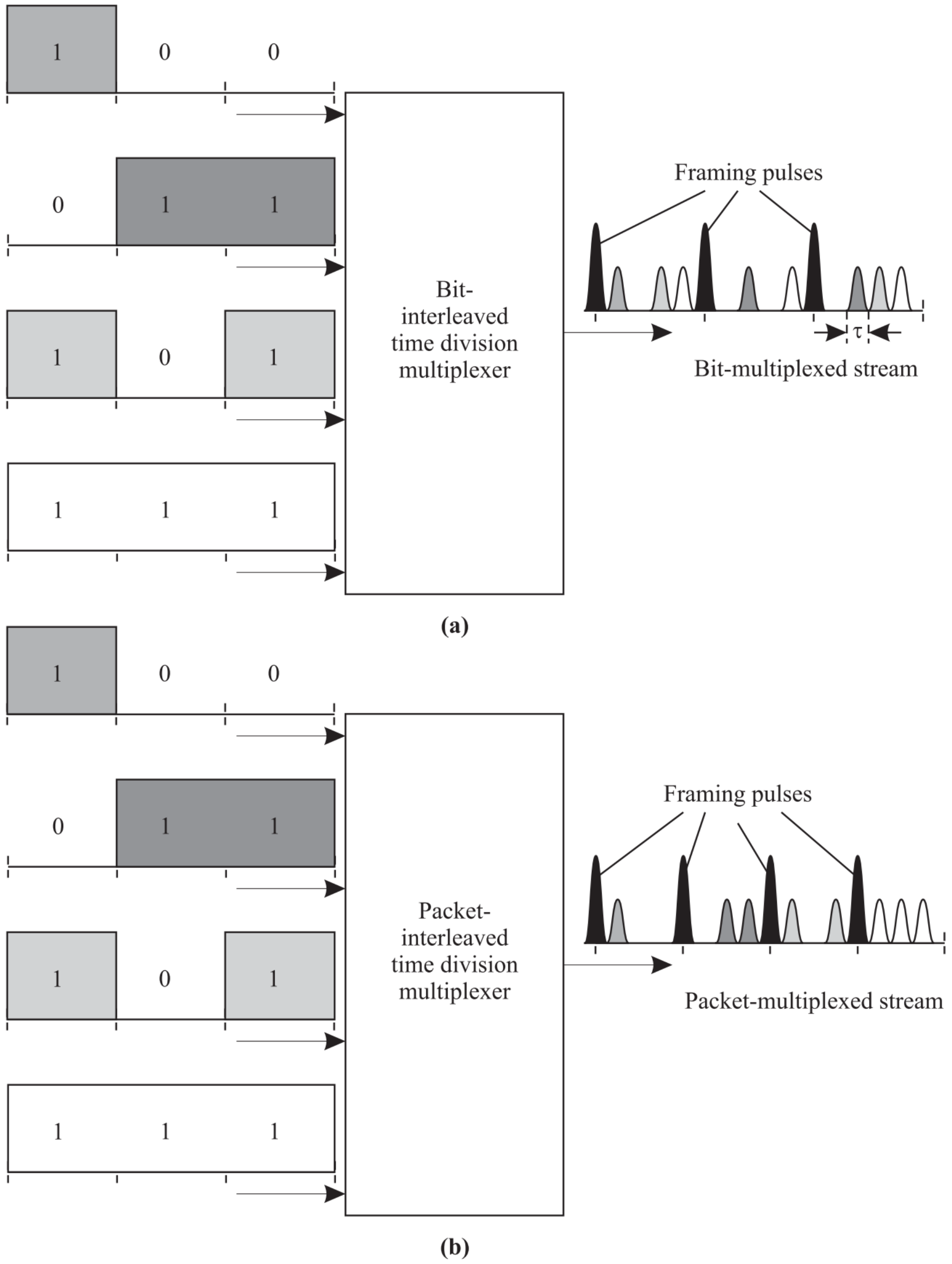

Figure 12.4 (a) Function of a bit-interleaved optical multiplexer. (b) Function of a packet-interleaved optical multiplexer. The same four data streams are multiplexed in both cases. In (b), the packet size is shown as 3 bits for illustration purposes only; in practice, packets are much larger and vary in size. Note that in both cases, the data must be compressed in time.

7.2 Synchronization

Synchronization is the process of aligning two pulse streams in time. In PPS networks, it can refer either to the alignment of an incoming pulse stream and a locally available *clock* pulse stream or to the relative alignment of two incoming pulse streams. Recall our assumption of fixed-size packets. Thus if framing pulses are used to mark the packet boundaries, the framing pulses must occur periodically.

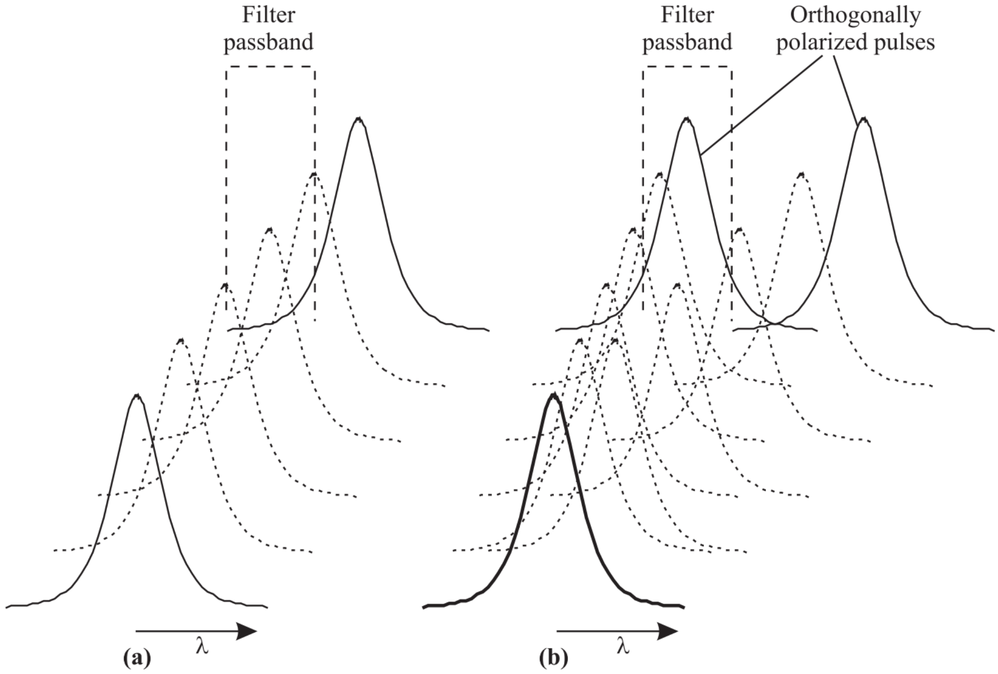

Figure 12.12 Illustration of the operation of a soliton-trapping logical AND gate. (a) Only one pulse is present, and very little energy passes through to the filter output. This state corresponds to a logical zero. (b) Both pulses are present, undergo wavelength shifts due to the soliton-trapping phenomenon, and most of the energy from one pulse passes through to the filter output. This state corresponds to a logical one.

The function of a synchronizer can be understood from Figure 12.13. The two periodic pulse streams, with period T, shown in Figure 12.13(a) are not synchronized because the top stream is ahead in time by ΔT. In Figure 12.13(b), the two pulse streams are synchronized. Thus, to achieve synchronization, the top stream must be delayed by ΔT with respect to the bottom stream. The delays we have hitherto considered, for example, while studying optical multiplexers and demultiplexers, have been *fixed* delays. A fixed delay can be achieved by using a fiber of the appropriate length. However, in the case of a synchronizer, and in some other applications in photonic packet-switching networks, a *tunable delay* element is required since the amount of delay that has to be introduced is not known a priori. Thus we will now study how tunable optical delays can be realized.

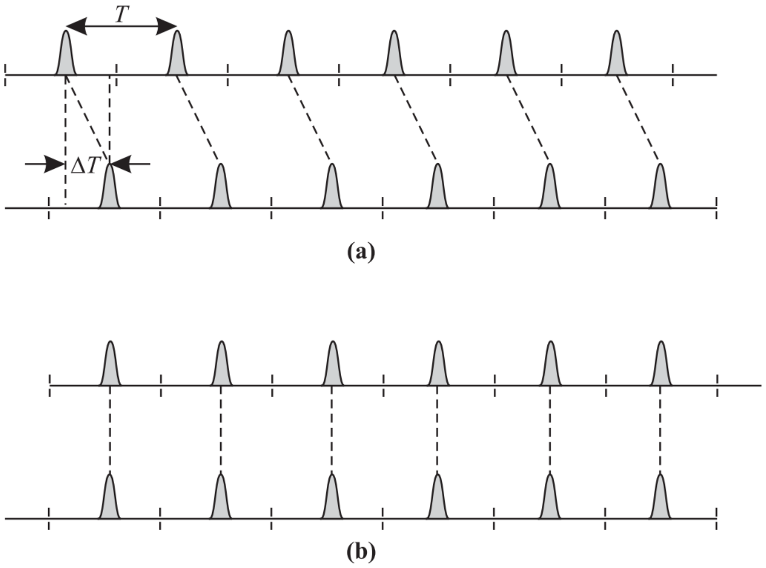

Figure 12.13 The function of a synchronizer. (a) The two periodic pulse streams with period T are out of synchronization; the top stream is ahead by ΔT. (b) The two periodic streams have been synchronized by introducing a delay ΔT in the top stream relative to the bottom stream.

7.3 Header Processing

For a header of fixed size, the time taken for demultiplexing and processing the header is fixed, and the remainder of the packet is buffered optically using a delay line of appropriate length. The processing of the header bits may be done electronically or optically, depending on the kind of control input required by the switch. Electrically controlled switches employing the electro-optic effect and fabricated in lithium niobate (see Section 3.7) are most commonly used in switch-based network experiments today. In this case, the header processing can be carried out electronically (after the header bits have been demultiplexed into a parallel stream). The packet destination information from the header is used to determine the outgoing link from the switch for this packet, using a look-up table. For each input packet, the look-up table determines the correct switch setting, so that the packet is routed to the correct output port. Of course, this leads to a conflict if multiple inputs have a packet destined for the same output at the same time. This is one reason for having buffers in the routing node, as explained next.

If the destination address is carried in the packet header, it can be read by demultiplexing the header bits using a bank of AND gates, for example, TOADs, as shown in Figure 12.8. However, this is a relatively expensive way of reading the header, which is a task that is easier done with electronics than with optics. Another reason for using electronics to perform this function is that the routing and forwarding functions required can be fairly complex, involving sophisticated control algorithms and look-up tables.

7.4 Buffering

In general, a routing node contains buffers to store the packets from the incoming links before they can be transmitted or forwarded on the outgoing links—hence the name *store and forward* for these networks. In a general store-and-forward network, electronic or optical, the buffers may be present at the inputs only, at the outputs only, or at both the inputs and the outputs, as shown in Figure 12.2. The buffers may also be integrated within the switch itself in the form of random access memory and shared among all the ports. This option is used quite often in the case of electronic networks where both the memory and switch fabric are fabricated on the same substrate, say, a silicon-integrated circuit, but we will see that it is not an option for optical packet switches. We will also see that most optical switch proposals do not use input buffering for performance-related reasons.

There are at least three reasons a packet has to be stored or buffered before it is forwarded on its outgoing link. First, the incoming packet must be buffered while the packet header is processed to determine how the packet must be routed. This is usually a fixed delay that can be implemented in a simple fashion. Second, the required switch input and/or output port may not be free, causing the packet to be queued at its input buffer. The switch input may not be free because other packets that arrived on the same link have to be served earlier. The switch output port may not be free because packets from other input ports are being switched to it. Third, after the packet has been switched to the required output port, the outgoing link from this port may be busy transmitting other packets, thus making this packet wait for its turn. The latter delays are variable and are implemented differently from the fixed delay required for header processing.

The lack of good buffering methods in the optical domain is a major impediment. Unlike the electronic domain, we do not have random access memory in the optical domain. Instead, the only way of realizing optical buffers is to use fiber delay lines, which consist of relatively long lengths of fiber. For example, about 200 m of fiber is required for 1 μs of delay, which would be sufficient to store 10 packets, each with 1000 bits at 10 Gb/s. Thus usually very small buffers are used in photonic packet-switching networks. Note that unlike an electronic buffer, a packet cannot be accessed at an arbitrary point of time; it can exit the buffer only after a fixed time interval after entering it. This is the time taken for the packet to traverse the fiber length. This constraint must be incorporated into the design of PPS networks.

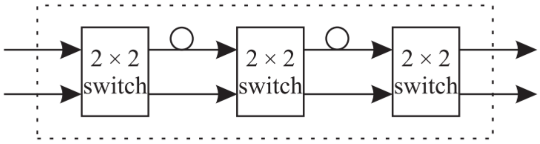

Figure 12.15 Example of a 2 × 2 routing node using a feed-forward delay line architecture.

Of course, by repeated traversals of the same piece of fiber, packet delays that are multiples of this basic delay can be obtained.

PPS networks typically make use of delay lines in one of two types of configurations. Figure 12.15 shows one example of a *feed-forward* architecture. In this configuration, a two-input, two-output routing node is constructed using three 2 × 2 switches interconnected by two delay lines. If each delay line can store one packet—that is, the propagation time through the delay line is equal to one slot—the routing node has a buffering capacity of two packets. If packets destined for the same output arrive simultaneously at both inputs, one packet will be routed to its correct output, and the other packet will be stored in delay line 1. This can be accomplished by setting switch 1 in the appropriate state. This packet then has the opportunity to be routed to its desired output in a subsequent slot. For example, if no packets arrive in the next slot, this stored packet can be routed to its desired output in the next slot by setting switches 2 and 3 appropriately.

7.5 Burst Switching

Burst switching is a variant of PPS. In burst switching, a source node transmits a header followed by a packet burst. Typically, the header is transmitted at a lower speed on an out-of-band control channel, although most proposals assume an out-of-band control channel. An intermediate node reads the packet header and activates its switch to connect the following burst stream to the appropriate output port if a suitable output port is available. If the output port is not available, the burst is either buffered or dropped. The main difference between burst switching and conventional photonic packet switching has to do with the fact that bursts can be fairly long compared to the packet duration in packet switching.

In burst switching, if the bursts are sufficiently long, it is possible to ask for or reserve bandwidth in the network ahead of time before sending the burst. Various protocols have been proposed for this purpose. For example, one such protocol, called Just-Enough-Time (JET), works as follows. A source node wanting to send a burst first sends out a header on the control channel, alerting the nodes along the path that a burst will follow. It follows the header by transmitting the burst after a certain time period. The period is large enough to provide the nodes sufficient time to process the header and set the switches to switch the burst through when it arrives, so that additional buffering is not needed for this purpose at the nodes.

7.6 Testbeds

Several PPS testbeds have been built over the years. The main focus of most of these testbeds is the demonstration of certain key PPS functions such as multiplexing and demultiplexing, routing/switching, header recognition, optical clock recovery (synchronization or bit-phase alignment), pulse generation, pulse compression, and pulse storage. We will discuss some of these testbeds in the remainder of this section. The key features of these testbeds are summarized in Table 12.2.

Table 12.2 Key features of photonic packet-switching testbeds described in Section 12.6.

Testbed	Topology	Bit Rate	Functions Demonstrated
KEOPS	Switch	2.5 Gb/s (per port)	4 × 4 switch, subnanosecond switching, all-optical wavelength conversion tunable lasers, packet synchronizer
KEOPS	Switch	10 Gb/s (per port)	16 × 16 broadcast/select, subnanosecond switching
FRONTIERNET	Switch	2.5 Gb/s (per port)	16 × 16, tunable laser
NTT	Switch	10 Gb/s (per port)	4 × 4 broadcast/select
Synchrolan (BT Labs)	Bus	40 Gb/s (aggregate)	Bit-interleaved data transmission and reception
BT Labs	Switch	100 Gb/s (per port)	Routing in a 1 × 2 switch based on optical header recognition
Princeton	Switch	100 Gb/s (per port)	Packet compression, TOAD-based demultiplexing
AON	Helix (bus)	100 Gb/s (aggregate)	Optical phase lock loop, pulse generation, compression, storage
CORD	Star	2.5 Gb/s (per port)	Contention resolution