JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks*

Ilia Baldine[1], Harry G. Perros[2], George N. Rouskas[2], and Dan Stevenson[1]

[1] MCNC ANR, Research Triangle Park, NC, USA
[2] NCSU Department of Computer Science, Raleigh NC, USA

Abstract. We present an architecture for a core dWDM network which utilizes the concept of Optical Burst Switching (OBS) coupled with a Just-In-Time (JIT) signaling scheme. It is a reservation based architecture whose distinguishing characteristics are its relative simplicity, its amenability to hardware implementation, support for quality of service and multicast natively. Another important feature is data transparency - the network infrastructure is independent of the format of the data being transmitted on individual wavelengths. In this article we present a brief overview of the architecture and outline the most salient features.

1 Introduction

The adoption of dWDM as the primary means for transporting data across large distances in the near future is a foregone conclusion, as no other technology can offer such vast bandwidth capacities. The current dominant technology for core networks are wavelength-routed networks with permanent or semi-permanent circuits set up between end points for data transfer. Many of the proposed architectures treat dWDM as a collection of circuits/channels with properties similar to electronic packet-switched circuits with customary buffering (potentially done in the optical domain) and other features of electronic packet-switching. In addition, transport protocols used today (e.g. TCP) developed for noisy low-bandwidth electronic links, are poorly suited for the high-bandwidth, extremely low bit-error rate optical links. The round trip times for signaling and the resulting high end-node buffer requirements are a poor match to the all-optical networks characterized by the high bandwidth-delay product. In order to address the processing and buffering bottlenecks, characteristic of the electronic packet-switching architectures, and, by extension, their dWDM derivatives, a wholly new architecture is required, which is capable of taking advantage of the unique properties of the optical medium, rather than trying to fit it into existing electronic switching frameworks. In addition, in dWDM networks data transparency (i.e. independence of the network infrastructure from the data format, modulation scheme etc., thus allowing transmission of analog as well as digital signals) becomes not only possible, but desirable.

In this paper we present an overview of an architecture for a core dWDM network. The type of architecture, described in this paper, is wavelength-routed, burst-switching, with the "just-in-time" referring to the particular approach to signaling, taken within this

* This research effort is being supported through a contract with ARDA (Advanced Research and Development Activity, http://www.ic-arda.org).

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1081–1086, 2002.

architecture. Signaling is done out of band, with signaling packets undergoing electro-optical conversion at every hop while data, on the other hand, travels transparently. For history of burst-switching the reader is referred to [6,9].

Just-In-Time (JIT) signaling approaches to burst-switching have been previously investigated in literature ([9,6]). The common thread in all these is the lack of the round-trip waiting time before the information is transmitted (the so-called TAG: tell-and-go scheme) when the cross-connects inside the optical switches are configured for the incoming burst as soon as the first signaling message announcing the burst is received. The variations on the signaling schemes mainly have to do with how soon before the burst arrival and how soon after its departure, the switching elements are made available to route other bursts. An example is the Just-Enough-Time (JET) scheme proposed in [7] which uses extra information to better predict the start and the end of the burst and thus use the switching elements needed to route the burst inside a switch for the shortest amount of time possible. Schemes also have been proposed for introducing QoS into the architecture ([8]). These schemes have been shown to reduce the blocking probability inside an OBS network with the disadvantage of requiring a progressively more complex schedulers ([5]).

In this short paper we present an overview and describe the salient features of the proposed Jumpstart architecture. For a more extensive treatment of the subject the reader is referred to [4,2].

1.1 Guiding Assumptions and Basic Architecture

The basic premise of this architecture is as follows - data, aggregated in bursts is transferred from one end point to the other by setting up the light path just ahead of the data arrival. This is achieved by sending a signaling message ahead of the data to set up the path. Upon the completion of data transfer the connection either times out or is torn down explicitly. Some of the basic architectural assumptions are summarized below:

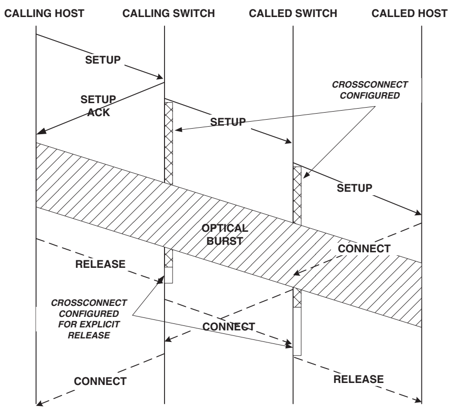

Fig. 1. Example of a burst

Out-of-band signaling - Signaling channel undergoes electro-optical conversions at each node to make signaling information available to intermediate switches.

Data transparency - *Data is transparent to the intermediate network entities,* i.e. no electro-optical conversion is done in the intermediate nodes and no assumptions are made about the data rate or signal modulation methods.

Network intelligence at the edge - Most “intelligent“ services are supported only by edge switches. Core switches are kept simple.

Signaling protocol implemented in hardware - So as not to create a processing bottleneck for high-bandwidth bursty sources, the signaling protocol must be implemented in hardware

No global time synchronization - In keeping with the “keeping it simple“ principle, we do not assume time synchronization between nodes.

A basic switch architecture presumes having a number of input and output ports, each carrying multiple wavelengths (envisioned to be in 100’s to 1000’s). At least one separate wavelength on each port is dedicated to carrying the signaling traffic. Any wavelength on an incoming port can be switched to either the same wavelength on any outgoing port (no wavelength conversion) or any wavelength on any outgoing port (partial or total wavelength conversion). The switching can be done by using MEMS micro-mirror arrays or some other suitable technology. Switching time is presumed to be in the μs range, with anticipation that it could be reduced further as the technology develops. Additionally, each switch is equipped with a scheduler which keeps track of wavelength switching configurations and configures the cross-connects on time to allow the data to pass through.

Data Transparency. We have briefly alluded to data transparency as being a desirable property of a core network of the future. Indeed, the ability to transmit optical digital signals of different formats and modulations, as well as analog signals simplifies many problems commonly associated with adaptation layers. In a burst-switched network, which essentially acts as a broker of time on a particular wavelength with high temporal resolution, this feature becomes relatively easy to implement, considering that signaling is done out of band on a separate channel. This is why JumpStart architecture makes no assumptions about the types of traffic it carries and instead schedules time periods on wavelengths within the network. The particular format that an end node uses to transmit its data to the destination is of no consequence to the network itself.

Processing Delay Prediction. Unlike data, signaling messages propagate through the network and accrue a processing delay inside each intermediate switch. For a SETUP message, which announces the arrival of a new burst to intermediate switches, this means that it has to be sent far enough in advance before the burst, so that the burst does not catch up with it before the destination is reached. Knowing this delay apriori, at the ingress switch, and communicating it to the source node (via SETUP_ACK) is part of the network function. This delay can be deduced from the destination address in the SETUP message, and further refined by the ingress switch over time, as CONNECT messages are sent back from the destination, indicating the actual processing delay incurred while the corresponding SETUP message traveled to the destination.

Quality of Service. When one talks about Quality of Service (QoS) in the context of contemporary packet-switching networking technologies and protocols (DiffServ, IntServ), the criteria for evaluating the QoS of a given connection involve network bandwidth and buffer management inside the routers and end-nodes. In the context of an all-optical transparent network such as proposed here most of these issues become irrelevant: data is transparent to the network and no buffering is done inside the network switches. The network acts merely as a time-broker on individual links. As a result when we discuss QoS in JumpStart, it is separated into several areas:

– QoS requirements defined for the specific adaptation layer used
– Optical QoS parameters, on which specific adaptation layer requirements may be mapped to
– Connection prioritization - allows us to preempt less important connections in favor of more important ones. It is a stand-alone property, which enables the network to deal with preemption of existing connections in a predictable manner.

Optical QoS parameters allow the network to route a connection along the best suited route depending on the type of signal the connection carries. Examples of optical QoS parameters are: bit-error rate, dynamic range, signal-to-noise ratio, optical channel spacing.

Multicast Support. Support for multicast connections is essential for future networks, however support for them within an architecture such as JumpStart may not be trivial. The optical signal must be split at certain points along the path according to the multicast routing tree in order for the network to remain all-optical, i.e. avoid electro-optical conversions. Such splitting presents a number of issues for the implementation, namely:

– A switch must be equipped with an optical splitting mechanism (splitting signaling messages does not present a problem, since they undergo electro-optical conversions in each switch).
– A number of such splits that can be done on a single connection is in general bounded by the optical power budget [3].

The result is that each connection may have a limited fan-out (contrary to present day electronic networks, where such issues are not considered).

Given these restrictions, for our network architecture we presume that the switches capable of splitting the optical signal are not common in the network, and, in fact, are sparsely dispersed throughout the network. Each end-node gets assigned one such switch as its multicast server switch through either administrative mechanism or a separate signaling mechanism. These special switches also take care of setting up routing trees for multicast connections, so in addition to special hardware they need to allow to split the optical signal, they also have special firmware to allow them to manage and route multicast connections. Thus all signaling messages from the source node that pertain to its multicast connections get routed by the network to its assigned multicast switch.

Within the multicast variety of connections we can identify two ways to setup a multicast session:

- Source-managed multicast: the source of the multicast knows the addresses of all of the members of the multicast group and that number is relatively small. In this case, the addresses of the members are directly included into the appropriate signaling messages by the source.
- Leaf-initiated join: in this scenario, a source may announce the existence of a multicast session, with a session id that is unique inside the network. Multicast servers in the network will learn of this new session through means that are outside of the scope of this discussion, and the end nodes will be able to join existing multicast sessions by communicating with their domain multicast servers.

In practice we would like to allow for a combination of both. A source may begin by specifying a few end nodes and allow the rest to use the leaf-initiated join capability. In the extreme case the source node simply announces an existence of a session and lets nodes in the network join as they wish. One additional option of multicast sessions is the session *scope*. The scope limits the availability of the session to nodes belonging only to specific domain(s). Additionally a source node may specify as part of the connection options that only it has the authority to add new leaves to the tree, in which case no leaf-initiated join connections will be allowed.

Label Switching. Label switching concept will be utilized by the signaling channel in order to achieve several goals:

- Speed up in accessing call state in the switch (based on label, not call reference number, which is not unique within the network).
- Guarantee that forward and backward routes coincide (while connections in a JumpStart OBS network are unidirectional, signaling paths are not, and it is necessary that signaling messages travel the on the same path both in the forward and in the backward direction).
- Speed up routing (once a connection path has been established, it is desirable that further signaling messages do not consult the routing table but use a pre-established path).

For this purpose, labels similar to MPLS will be used as part of the signaling message format. These labels will have link-local significance (unique on one link, but not within a switch or the network). Similar to ATM and MPLS, these labels will be rewritten as the signaling message traverses the network. Special tables within the switch will be needed to maintain the forward and the backward label mapping. No label stacking will be allowed.

Label distribution will be done on-the-fly, as part of signaling, while the connection is being setup, instead of utilizing a label distribution protocol like LDP or modified RSVP. Additional multicast support in a labeling mechanism will be necessary in those nodes that support multicast routing (multicast nodes). Unlike unicast-only nodes, which only need to maintain one-to-one label mappings for each connection, multicast nodes will require one-to-many and many-to-one mappings for label mechanism.

Persistent Connections. For some applications there will be a need for all bursts to travel the same route through the network, especially to those applications that are particularly sensitive to jitter or sequential arrival of information. Defining a persistent route service that precedes a series of bursts can allow the network to "nail down" a route for all subsequent bursts to follow. There are some network traffic engineering implications of persistent routes. If a significant portion of the network connections are established with fixed routes, then dynamic load balancing through routing changes in the network will become inefficient and perhaps fail. To minimize this potential problem, network service providers may choose to treat persistent route connections as a premium service offering. This service would be more expensive for service providers to support.

Multicast service as we have defined it also requires persistence. The first phase of establishing multicast service is to declare a session and build a routing tree. This is followed by one or several data transmission phase. Maintaining a persistent session is necessary so that the network can maintain state for multicast session routing as leafs are added and dropped through its lifetime.

1.2 Conclusions

In this paper we presented a short description of Jumpstart - a new proposed architecture for all-optical WDM burst-switched networks. We described and justified the need for the most important features in the network. Fore more information about the project we suggest [1].

References

1. Jumpstart project. In *http://jumpstart.anr.mcnc.org*.
2. Ilia Baldine, George Rouskas, Harry Perros, and Daniel Stevenson. Signaling Support for Multicasting and QoS within the Jumpstart WDM Burst Switching Architecture. *Optical Networks Magazine*, 2002. submitted for publication.
3. Karthik Chandrasekar, Dan Stevenson, and Paul Franzon. Optical Hardware Tradeoffs for All-Optical Multicast. In *Submitted to OFC*, 2002.
4. I.Baldine, G.N.Rouskas, H.Perros, and D.Stevenson. Jumpstart - a Just-In-Time Signaling Aarchitecture for WDM Burst-Switched Networks. *IEEE Communications*, page p.82, Feb. 2002.
5. Pronita Mehrotra, Ilia Baldine, Dan Stevenson, and Paul Franzon. Network Processor Design for use in Optical Burst Switched Networks. In *International ASIC/SOC Conference*, September 2001.
6. Chunming Qiao and Myungsik Yoo. Optical Burst Switching (OBS). *Journal of High Speed Networks*, 8, 1999.
7. Myungsik Yoo, Myongki Jeong, and Chunming Qiao. A High-Speed Protocol for Bursty Traffic in Optical Networks. In *SPIE*, volume 3230, pages pp.79–80.
8. Myungsik Yoo and Chunming Qiao. A New Optical Burst Switching Protocol for Supporting Quality of Service. In *SPIE*, volume 3531, pages pp.396–405.
9. John Y.Wei and Ray McFarland. Just-In-Time Signaling for WDM Optical Burst Switching Networks. *Journal of Lightwave Technology*, 18(12):pp.2019–2037, December 2000.