Report Number: WUCS-98-30

1998-01-01

Terabit Burst Switching Progress Report (6/98-9/98)

Jonathan S. Turner

This report summarizes progress on the Terabit Burst Switching Project at Washington University for the period from June 15, 1998 through September 15, 1998.

Terabit Burst Switching Progress Report (6/98–9/98)

Jonathan S. Turner
jst@cs.wustl.edu

WUCS-98-30

December 29, 1998

Department of Computer Science
Campus Box 1045
Washington University
One Brookings Drive
St. Louis, MO 63130-4899

Abstract

This report summarizes progress on Washington University's *Terabit Burst Switching Project*, supported by DARPA and Rome Air Force Laboratory. This project seeks to demonstrate the feasibility of *Burst Switching*, a new data communication service which can more effectively exploit the large bandwidths becoming available in WDM transmission systems, than conventional communication technologies like ATM and IP-based packet switching. Burst switching systems dynamically assign data bursts to channels in optical data links, using routing information carried in parallel control channels. The project will lead to the construction of a demonstration switch with throughput exceeding 200 Gb/s and scalable to over 10 Tb/s.

[0]This work is supported by the Advanced Research Projects Agency and Rome Laboratory (contract F30602-97-1-0273).

Terabit Burst Switching
Progress Report (6/98–9/98)

Jonathan S. Turner
jst@cs.wustl.edu

This report summarizes progress on the Terabit Burst Switching Project at Washington University for the period from June 15, 1998 through September 15, 1998.

1. Progress on Burst Storage Management

In our last progress report [6], we described a technique for managing storage within a Burst Switch Element of a multistage burst switching network. This technique used a search tree to represent the buffer occupancy curve of a Burst Storage Unit (BSU), but did not provide a complete solution to the problem of managing burst storage.

Since that time, we have refined the original search tree data structure to provide a complete solution. To decide if an arriving burst can be stored within a burst store, the *Burst Storage Manager* (BSM) must determine if there is enough storage space available to accommodate the burst throughout the entire time period that it will be in the BSU. The inputs to this decision are the time period during which the burst will occupy the queue and the amount of storage it requires during that period. (Note that the amount of required storage may be less than the burst size, since the burst may begin leaving the BSU before the end of the burst has arrived.) The BSM maintains information on the projected buffer occupancy, and uses this, together with the information on the arriving burst to decide if the arriving burst can be accepted. The left side of Figure 1 illustrates this. The plot of buffer occupancy vs. time provides the information that the BSM needs to make its decisions. Given such a curve, one can easily determine the impact on the buffer occupancy curve, by just adding the amount of required storage space to the portion of the curve corresponding to the time interval during which the new burst is to be stored. If this results in the curve exceeding the maximum storage capacity of the BSU, the arriving burst must be rejected. Otherwise, it can be accepted and stored.

The search tree on the right hand side of Figure 1 can be used to represent the buffer occupancy curve. In this search tree, the entries at the bottom represent the various segments of the buffer occupancy curve. For example, the entry containing the time value 10, and the buffer usage value 5, represents the segment starting at time 10, during which the value of the buffer occupancy curve is five. Since the bottom level entries are sorted according to their time values, by consulting the next entry to the right, we can see that the buffer occupancy remains at 5 until time 14. The

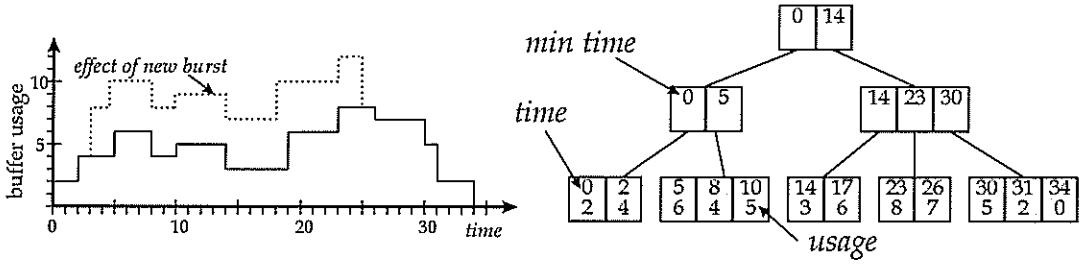

Figure 1: Using Search Tree to Manage Burst Storage

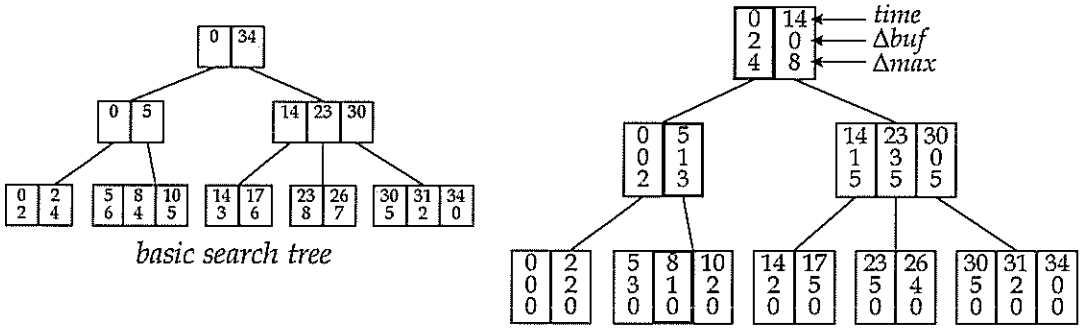

Figure 2: Differential Search Tree for Burst Storage

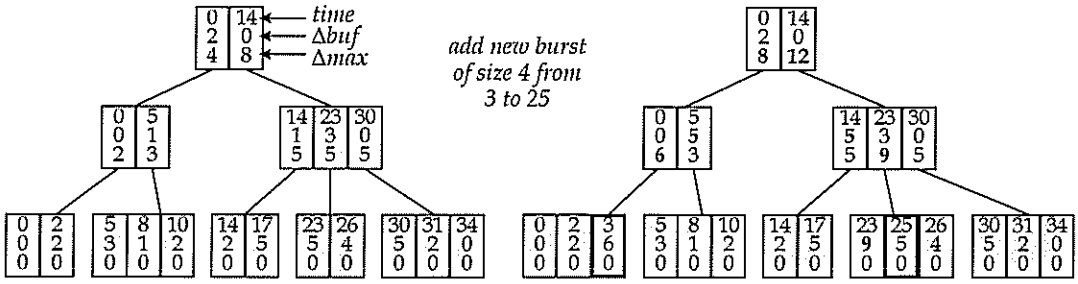

Figure 3: Updating Search Tree to Insert New Burst

entries in the internal nodes of the search tree give the minimum time values appearing in their respective subtrees. This allows us to quickly determine the value of the buffer occupancy curve at any particular time.

While the basic search tree of Figure 1 provides a useful starting point, it does not give us a complete solution. We need to extend the search tree to provide the capabilities required by the BSM. Figure 2 illustrates the extensions needed. In this tree, every entry has three fields, a *time* field, a field called Δbuf and a field called Δmax. The time fields are exactly as in the basic search tree. At the leaves, they define the starting points of segments of the buffer occupancy curve. At the internal nodes, they specify the smallest time values within the respective subtrees. The Δbuf values are used to determine the buffer usage at any point in time. In particular, for any leaf entry, the buffer usage starting at the time value specified by that entry, is the sum of the Δbuf values from the entry to the root of the tree. This is illustrated in the figure, where the Δbuf values in the highlighted entries sum to 4, which matches the value in the corresponding leaf entry of the basic search tree.

The Δmax values can be used, along with the Δbuf values, to determine the maximum buffer usage within any time interval. Consider any entry in an internal node of the search tree and let B be the maximum buffer usage within the subtree corresponding to that entry. If we add the Δbuf values from the root to the given entry together, and then add to this, the Δmax value for the entry, we will get B. As one example of this, consider the middle entry of the right hand node in the second level of the search tree. If we add the Δbuf values along the path from the root to this node, we get $0 + 3 = 3$. Adding the Δmax value (5) of the entry gives us a result of 8. This means that the maximum buffer usage within the subtree of this entry is 8. This can be confirmed by looking at the corresponding leaf node in the basic search tree on the left side of the figure.

This *differential search tree* allows a BSM to quickly determine if an incoming burst can be handled. It requires two probes of the search tree, both starting from the root and proceeding to the leaves. One search goes to the leaf corresponding to the start of the time interval during which the burst is to be stored. The other goes to the leaf corresponding to the end of the time interval. Using the Δbuf and Δmax fields, we can easily determine the maximum buffer occupancy during the required time period. If the sum of this value and the amount of storage needed by the arriving burst exceeds the buffer capacity, then the burst must be discarded.

If we decide to accept a burst, we must update the search tree to reflect the change in the

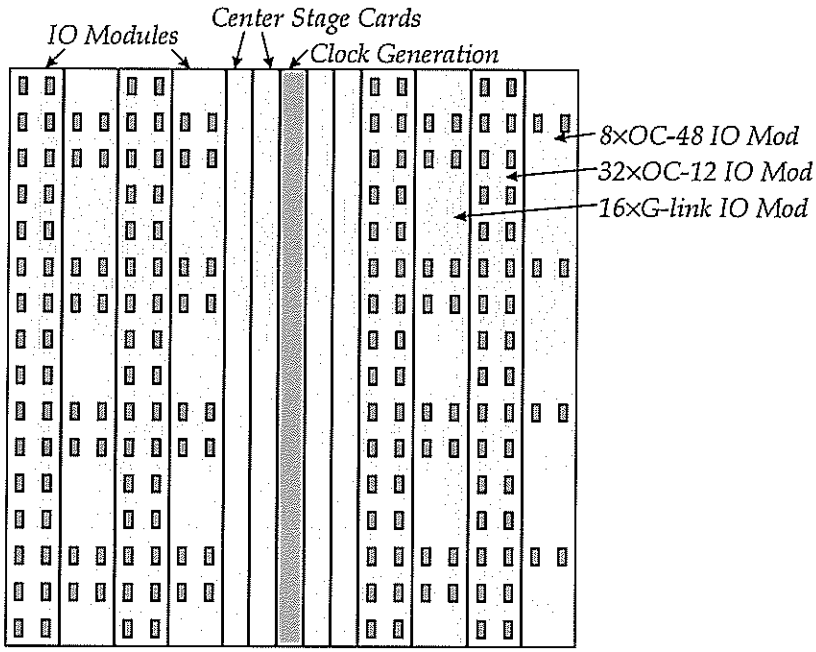

Figure 4: Physical Packaging Arrangement for ATM Switch

underlying buffer occupancy curve. This requires inserting new entries into the leaves, at the times which define the start and end of the interval during which the burst is to be stored. The Δbuf and Δmax values along the paths from these new entries to the root of the tree must also be updated. This is illustrated in Figure 3 which shows how the tree is updated to reflect the addition of a new burst.

If new entries cannot be inserted into existing nodes, new nodes must be created, requiring new entries in nodes at the next level up. This can ripple all the way up the tree, but even with this restructuring, the time required to check that a burst will fit and update the tree appropriately is $O(\log$ (# of stored bursts)).

2. Prototype ATM Switch

This project also includes construction of a 160 Gb/s ATM switch which will be connected to the burst switch to demonstrate end-to-end communication. In the last quarter we have completed the design and fabrication of a dual G-link line card supporting two interfaces operating at 1 Gb/s each. This required the development of a MUX/DEMUX component to allow the two external interfaces to share a single 2.4 Gb/s internal interface. We have also designed a quad OC-12 interface, providing four OC-12 interfaces on a single line card; this required development of another MUX/DEMUX component. Further verification work is required before the OC-12 card can be fabricated. Recently,

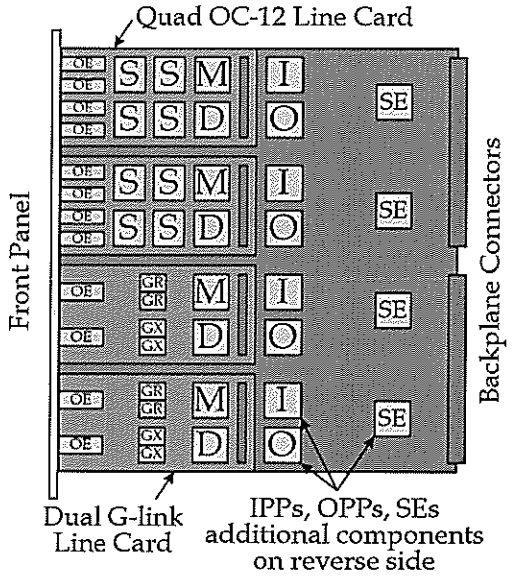

Figure 5: ATM Switch IO Module

vendors have begun producing integrated components for OC-48 SONET. We are evaluating these components for possible use in both the ATM and burst switches.

We have made some changes to our plans for the physical packaging of the ATM switch. Previously, we had planned to mount circuit boards in an orthogonal arrangement, on either side of a passive mid-plane. We have since determined that a more conventional packaging approach, with all boards mounted on one side of a passive backplane is possible. Figure 4 illustrates how the front panel will look. There will be eight IO modules, each containing eight line cards of various types. There will be four cards carrying the Switch Elements forming the center stage of the three stage interconnection network, plus one card generating clock and timing signals for the entire system.

We have also refined our plans for the IO modules. The new IO module design is shown in Figure 5. Each IO module consists of a main circuit board and eight line cards, four on each side. The optoelectronic components on the line cards align with openings on the module's front panel, eliminating the need for interconnecting cables. The main circuit board contains eight *Input Port Processor* and eight *Output Port Processor* chips, four on each side. These interface directly to the line cards through parallel board-to-board connectors. The main board also contains eight *Switch Element* chips. Each chip implements one-fourth of an eight port switch element with 32 bit data paths. One of the two resulting switch elements belongs to the first stage of the three stage interconnection network. The other belongs to the third stage.

References

[1] Chaney, Tom, J. Andrew Fingerhut, Margaret Flucke and Jonathan Turner. "Design of a

Gigabit ATM Switch," *Proceedings of Infocom*, April 1997.

[2] Cormen, Thomas, Charles Leiserson, Ron Rivest. *Introduction to Algorithms*, MIT Press, 1990.

[3] Siemens Semiconductor Group. "Parallel Optical Link (PAROLI) Family," http://w2.siemens.de /semiconductor/products/37/3767.htm, 1998.

[4] Turner, Jonathan S. "Terabit Burst Switching," Washington University Technical Report, WUCS-98-17, 1998.

[5] Turner, Jonathan S. "Terabit Burst Switching Progress Report (12/97-3/98)," Washington University Technical Report, WUCS-98-16, 1998.

[6] Turner, Jonathan S. "Terabit Burst Switching Progress Report (3/98-6/98)," Washington University Technical Report, WUCS-98-22, 1998.