RC24828 (W0907-056) July 27, 2009
Electrical Engineering

IBM Research Report

VLAN-Based Routing Infrastructure for an All-Optical Circuit Switched LAN

Xiaolan J. Zhang*, Rohit Wagle, James Giles
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

*Currently at University of Illinois, Ubana-Champaign

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

VLAN-Based Routing Infrastructure for an All-Optical Circuit Switched LAN

Xiaolan J. Zhang*
Department of Electrical and Computer Engineering
University of Illinois, Urbana–Champaign
Email: xzhang29@crhc.illinois.edu

Rohit Wagle and James Giles
IBM T. J. Watson Research Center
Hawthorne, NY
Email: rwagle@us.ibm.com, gilesjam@us.ibm.com

***Abstract*—Exploring the use of all-optical MEM switches on Local Area Network (LAN) to increase bandwidth and reduce energy cost has received increasing attention. Compared to traditional electronic switches, an all-optical switch provides higher bandwidth, less energy cost and cheaper wiring. Once the optical port count per switch becomes a resource constraint, an all-optical switched core can change the network topology dynamically to redistribute bandwidth resources between computing hosts. This feature is particularly useful for stream processing systems whose communication patterns vary over time and where rebalancing of networking resource is needed periodically.**

The work we present in this paper is a practical solution for high level software systems to route through a reconfigurable optical MEM switched LAN. Our solution can be readily applied on commercial switches with standard Layer-2 protocols. Network reconfiguration time and round-trip delay are measured. Our implementation is validated with the IBM System S stream processing system.

I. Introduction

The need for higher bandwidth on LANs continues to increase to support high performance parallel computing applications. The advances of electronic switching technologies begin to support 10Gbps end-to-end bandwidth and beyond (e.g. Infiniband [1]) but at a steep increasing of upgrading and maintenance cost per unit bandwidth. Instead, using an all-optical switch, the all-optical part of the network can sustain the growth of bandwidth at the electronic edge without equipment upgrades at the core [2]. In addition to the benefit of increased capacity of network at a lower cost, an all-optical micro-electrical-mechanical (MEM) switch can be dynamically controlled by software to change the physical topology of the network, which provides another dimension of optimization beyond load balancing at computing hosts.

Although an optical MEM switch provides many benefits, to be practical, it must be integrated into existing network infrastructure. In particular, the MEM switch is connected to IBM BladeCenters' Ethernet switches to form an optical circuit switched (OCS) network, providing a high-bandwidth circuits between BladeCenters as shown in Figure 1. The network system supports high performance distributed stream processing systems [3], [4], such as IBM System S platform [5].

*The author was affiliated with IBM T. J. Watson Research Center when the paper was written.

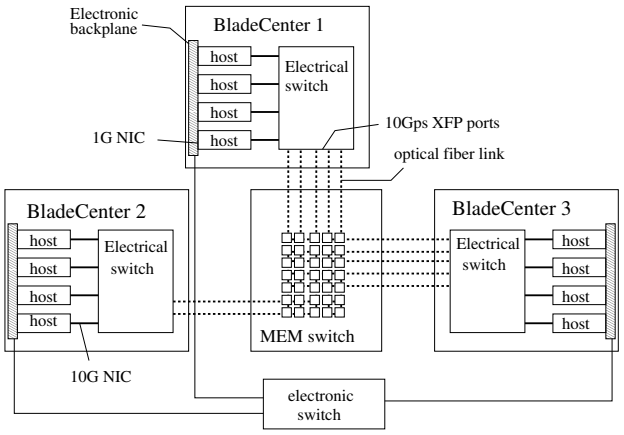

Fig. 1. Inter-cluster networking model for three BladeCenters and one MEM switch. More hosts and BladeCenters can join the optical circuit switched (OCS) network. More MEM switches can be added to expend the OCS network.

There are many routing issues to be considered for high-level software components to send data through an all-optical switched LANs. The first issue is network utilization. At network Layer-2, each Ethernet switch discovers the routes (via ARP and forwarding tables) between hosts by forwarding initial packets to all possible ports. If the network topology includes a loop, these packets are constantly circulated and cause serious congestion, also known as flooding or count-to-infinity [6]. Spanning tree (ST) protocol [7] is widely used in LANs to enforce a loop-free switch forwarding topology by disabling redundant links. However, the network bandwidth utilization is reduced significantly and ST suffers long convergence period [8]. Using multiple virtual area networks (VLAN) [9] and ST approaches to solve utilization problem on static topological LAN has been proposed by Viking system [10] and layered routing approach [8]. These works aim at improving the link utilization as well as traffic engineering at Layer-2. Others [11] have proposed new protocols to improve tree constructions. Many commercial switches have per-VLAN spanning tree (PVST) protocol readily implemented. And some of them have IEEE 802.1w rapid spanning tree (RSTP) and IEEE 802.1s multiple rapid spanning tree (MSTP) implemented as well. The goal of our work is to take advantage of these technologies and provide an integrated

scheme to achieve simple end-to-end routing through a general topology reconfigurable network.

The second problem is that the solution should work for dynamic reconfigurable connectivity with minimum interruption time so that workload and network traffic can be balanced over the time. Previous work mainly focus on static connected networks with occasional failures. Our focus is a more general topology reconfiguration problem. For streaming systems in particular, occasionally data rate reduction is undesired but still allowed during a short period of time. In a particular configuration (Figure 1), the BladeCenters are connected to two networks, one is a slower and fully-connected electronic switching network (1Gbps) and the other is the new faster optical network (10Gps). The electronic network is used to transmit control messages in the network and can also be used as a backup when the optical connections fail or are being reconfigured. Although the rate reduction penalty can be justified by long-term throughput gain of the system running on an optimized network, shorter affecting period still contributes to better overall performance. Typically, the mechanical switching time of a MEM optical switch takes about 10ms. However, the switch firmware adds another 50ms-100ms processing time to deliver the result to network management software. High level software may choose to query link status instead of waiting for response to reduce the switching time but it can take even longer to query link state for some edge switches. The reduction of optical switching time is beyond the scope of this paper. Moreover, reconfiguration of electronic edge switches (such as changing VLANs, trunking groups or ST groups) can take 10s of seconds depending on how much the switch firmware supports dynamic reconfiguration and how many changes are applied. An engineering decision needs to be made considering these effects and finds the best tradeoff for different performance requirements.

In this paper, we propose a multiple virtual local area network (VLAN) based solution to support Layer-2 automatic packet forwarding on topology reconfigurable LAN, without additional firmware support on switches or network cards. The solution provides for a full utilization of network resources with minimum reconfiguration overhead. We discuss the impacts and tradeoffs of implementation decisions, such as trunking and spanning tree protocol. We measure the topological reconfiguration time and round-trip-delay of our prototype system. We validate this technique with the IBM System S stream processing platform at the end.

This paper is organized as follows. Section II presents the networking model and the problems. Section III proposes our routing solution. Section IV addresses practical issues on implementation. Section V presents the performance study of network reconfiguration time and routing delays. Section VI introduces the integration of the networking system with System S. Section VII concludes the paper.

II. Network Model and Problem Description

Our networking hardware model is illustrated by Figure 1. This model can be expanded with more hosts, clusters and optical switches. Each cluster is a group of hosts (up to 14 blades in IBM BladeCenter) that share electronic bandwidth internally through high speed switching backplane. If a host intends to reach a remote host in another cluster, it must send packet through the intra-cluster network. An electronic edge switch with optical ports (we use Nortel switch[12]) is installed at each cluster. Each host in the cluster connects to the switch via a 10Gbps Network Interface Card (NIC). The electronic switch has a few optical ports (up to six 10Gps XFP transceivers in our installation) The protocol for optical communication is 10 Gigabit Attachment Unit Interface (XAUI) [13]. A Calient MEM switch connects directly to the XFPs of Nortel switches. Note that the fiber channels through the all-optical switching is transparent to electronic switches, it has no effect on the networking protocols at Layer-2 and beyond. Nortel switches only detect link on/off status as the fibers are connected/disconnected by MEM switch so we can reconfigure the electronic cluster network by switching the optical switch. A low speed backup electronic network still exists as an alternative, which is at most 1Gps host-to-host bandwidth in our system.

The number of optical ports (XFPs) at each electronic switch is limited. The topology is reconfigured once a while when traffic load between clusters changes. For example in Figure 2, cluster A has three XFPs. B and C have 2 XFPs. The optical network may be configured fully-connected with 10Gps links between each pair of clusters. Alternatively, A and B can get higher bandwidth by trunking two optical connections between A and B in Figure 3. In this case, B and C has no direct optical connection since B has no free XFP left. In general, the optical network is not fully connected due to the limits of optical ports at each cluster.

The utilization problem is illustrated as follows. On a given topology in which a loop exists, the switch forwarding table cannot be established unless spanning tree algorithm has been used to disable links for removing the cycle. For example, in Figure 2, the optical link B-C may be disabled by the ST algorithm. The traffic flow from B-C is actually carried by B-A-C path. Obviously, it is a very inefficient solution as 1/3 capacity in the network is wasted. Previous studies[10], [8] have proposed PVST solutions to use the disabled link. In this paper, we extend the idea for reconfigurable topologies and discuss several implementation issues.

III. VLAN-Based Routing Infrastructure

In this section, we introduce our VLAN-based solution to achieve efficient network reconfiguration and routing for high level streaming components to communicate.

A. VLAN Assignment

The initial VLAN assignment process is described in Algorithm 1. Basically, each unique set of clusters is assigned to a VLAN number. For example, in Figure 4, there are three clusters, A, B and C, each equipped with some packet switch interfaces to the OCS network. VLAN 1 is assigned to clusters $\{A, B\}$. VLAN 2 is assigned to clusters $\{A, C\}$. VLAN 3 is

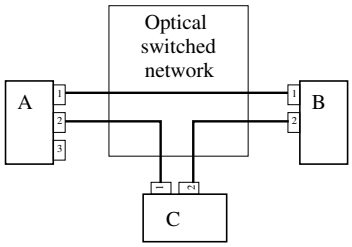

Fig. 2. Direct routing among three clusters. A has three XFPs. B and C have two XFPs each.

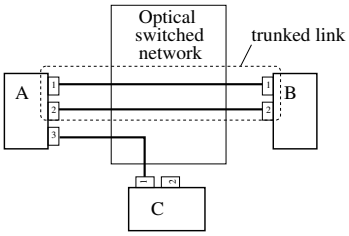

Fig. 3. Trunking of two links between A and B to provide 20Gps bandwidth. A has three XFPs. B and C have two XFPs each.

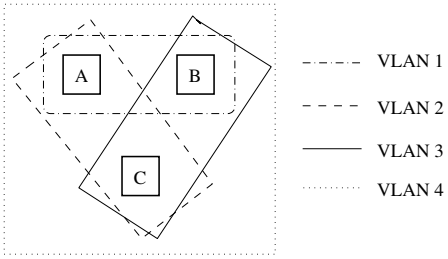

Fig. 4. Initial VLAN configuration.

assigned to clusters {B, C}. VLAN 4 is assigned to all the clusters {A, B, C}. The interfaces of A can receive packets from all of VLAN 1, 2, and 4. Those of B can received packets from VLAN 1, 3, 4 and C can receive from VLAN 2, 3, 4. If A wants to talk with B via the direct link, A can send packets through VLAN 1. Routing paths are effectively configured and separated by VLANs. The next section discusses routing in detail.

The paths provided by different VLANs must have one hop difference because there is at least one cluster difference in the cluster set of two different VLANs, using our VLAN assignment algorithm. This feature provides selection of different routes between the same source and destination. The ST issues are discussed in Section IV.

All clusters, including packet switches and host network cards, are preconfigured with their assigned VLAN group once. Each VLAN is assigned an unique IP address on the hosts. [1] An application running on the host can select paths to a destination by sending packets to a particular IP address and the destination receives packets by keep listening to all IP ports. Each cluster is assigned to more than one VLAN. Because all the switching interfaces join multiple VLANs, these VLANs should be globally assigned (packets keep their VLAN tag all the way through the switching network).

Let n be the number of clusters. h denotes the maximum number of hops supported. Equation 1 computes the total number of extra VLANs needed, which is the sum of cluster sets found at each stage. Equation 2 computes the total number of VLANs each cluster joins, since a cluster joins $\binom{n-1}{k-1}$ VLANs at each stage in the algorithm. The complexity of Algorithm 1 is exponential to n but in practice n is limited (see Figures 6(a) and 6(b)). The configuration is done in a few minutes for our system, which is small for an initial setup.

$$\sum_{k=2}^{\min(h+1,n)} \binom{n}{k} \quad (1) \qquad \sum_{k=1}^{\min(h,n-1)} \binom{n-1}{k} \quad (2)$$

B. Routing Schemes

1) Intra-Cluster Routing: IBM BladeCenter provides a high bandwidth backplane to connect computing hosts on the same cluster. Intra-cluster switching bandwidth can support up to

[1]The TCP/IP protocol stack on Linux platforms supports VLANs by creating virtual interfaces over physical interfaces. Each VLAN is assigned on a separate subnet and each virtual interface over all hosts for that VLAN is assigned a unique IP on that subnet.

1: VLAN $id \leftarrow 1$
2: cluster size $k \leftarrow 2$
3: **while** $k \leq h+1$ **do**
4: Find all unique $\binom{n}{k}$ cluster sets $\mathbb{S}_k$ of size k.
5: **for all** cluster set $S \in \mathbb{S}_k$ **do**
6: **for all** cluster $c \in S$ **do**
7: Assign VLAN id to all host and switch interfaces of c.
8: **end for**
9: $id \leftarrow id+1$
10: **if** cluster size $|S| > 2$ and the switch doesn't support packet flood suppression **then**
11: Enable ST on VLAN id.
12: **end if**
13: **end for**
14: $k \leftarrow k+1$
15: **end while**

Algorithm 1: VLAN assignment algorithm.

100Gbps switching capacity. But the current Ethernet NIC from each host to the backplane is still 1Gps. If the switching capability is not the limiting factor, the 10G NIC switch can also be used to route internal paths.

Our VLAN-based solution already implemented intra-cluster routing. Because the source and destination host is on the same cluster, they share the set of VLAN groups. Therefore, the packets can be sent via any VLAN port that the cluster is member of. In the same configuration of Figure 4, if the source and destination hosts are both on cluster A, they can send packets via VLAN 1 or 2 to establish an intra-cluster path.

2) Direct Routing: If an optical link is available between two clusters, direct routing is always available even if there is a loop in the network. The VLAN that includes only the source and destination clusters never contains a loop. A routing example is illustrated in Figure 2. Each cluster has at least two interfaces connecting to the OCS network. In the current configuration, there are three direct links connecting A–B, B–C and A–C. Let the VLAN assignment in the system be the same as in Figure 4. Hosts in A can send packets to hosts in B via VLAN 1. Packets on VLAN 1 only flow between cluster A and B via the direct link A.1–B.1 (see Figure 2). Packets on VLAN 2 are visible only to the hosts in cluster A and C. Packets for VLAN 3 are visible only to the hosts in cluster B and C. Therefore all three fibers are fully utilized.

Readers might notice that VLAN 4 is redundant for the fully connected network. VLAN 4 can be useful when network topology changes, for example, into Figure 3 where B and C

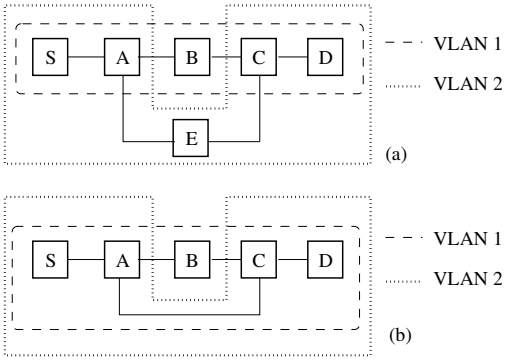

Fig. 5. Multi-hop and multi-path routing between S and D. (a) Two paths can be picked by choose different VLANs. (b) VLAN 2 may provide the same or different paths to VLAN 1 depending on ST construction.

does not have any direct optical connection. In the next section, we discuss the use of VLAN 4 if the application wants to use multi-hop routing between B and C.

3) Multi-Hop and Multi-Path Routing: In an optical switched network, it is desirable to have direct connections between clusters to avoid extra electronic processing at intermediate electronic cluster switches. However, the benefit gained from sharing high bandwidth via multi-hop routing could outweigh the drawbacks of optical to electrical conversions, if the optical transport provides significantly more bandwidth than the electronic backup network when a fully connected topology is impossible. Multi-hop routing is available for the clusters that are not directly connected. The simplest way to route over the shortest path between two clusters, which is guaranteed to be loop free (otherwise, a shorter path is able to find). For example, to route B and C in Figure 3, one can send packets on VLAN 4 which does not contain a loop topology. However in Figure 2, VLAN 4 should not be used unless per VLAN ST is enabled.

Multi-path routing is also available if multiple non-inclusive paths exist. Non-inclusive means that the set of clusters for one path should not be a subset/superset of clusters for another path, i.e. each path should at least have one cluster different from the other. For example, in Figure 5(a), clusters S and D contains the source and destination hosts respectively. Two routes are available from S to D. One is via VLAN 1 that includes clusters $\{$S, A, B, C, D$\}$. The other is via VLAN 2 that includes clusters $\{$S, A, E, C, D$\}$. Note that both VLANs are loop free and one VLAN is not the subset of the other. The optimizer can therefore choose either VLAN to send packets. However in the example shown in Figure 5(b), multiple choices may not be available. VLAN 2 is a safe choice here. If ST is enabled on VLAN 1, VLAN 1 can also be used but the path is decided by tree construction. Of course, one can setup root priorities to determinize the tree but the additional complexity added in configuration does not add much value.

#hop h	#cluster n
1	45
2	18
3	12
4	11
5-9	10

(a) ST disabled.

#hop h	#cluster n
1	45
2	9
3-6	7

(b) ST enabled.

Fig. 6. Maximum number of clusters supported given maximum available path hop count.

IV. Implementation issues

In this section, we analyze the impact of several implementation decisions to the performance of the system, in terms of network reconfiguration time and utilization.

A. Trunking

Trunking multiple parallel links can increase the bandwidth between a pair of clusters. Otherwise, the electronic switch considers these multiple links as a loop. If ST is enabled, only one link is active and the rest of the link bandwidth is wasted. Link trunking has to be performed on each packet switch accordingly to the topology. Figure 3 shows that link A.1–B.1 and A.2–B.2 are trunked on the both side of A and B. These two links appear logically as a one fat link to the IP layer. However, as we show later, trunking can significantly slow down the process of building forwarding tables after reconfiguration.

B. Spanning Tree Issues

Packets can circulate forever in a VLAN when there is a loop topology on network Layer-2, which is called flooding or count-to-infinity. The flooding effect in one VLAN can congest the entire switch and affect other loop-free VLANs. Enabling ST per VLAN is a solution but it has a few drawbacks.

I. ST takes long to converge after reconfiguration. Nortel switch suggests a 50-second convergence waiting time. Rapid spanning tree protocol (RSTP) may be enabled to expedite such process but Nortel switch only support RSTP for one VLAN. Cisco PVST+ does not have 10G compatible model for BladesCenter at the time. Nevertheless, temporary forwarding flooding may exist in a duration of tens of seconds [6] with RSTP protocols. Fast port forwarding should be enabled to reduce the waiting time.

II. The limitation of the number of STs support on each switch limits the scale of the optical network. Each Nortel switch can maximumly have 1023 user defined VLANs and 125 STs. Using Equation 1, Figure 6(a) shows the maximum cluster size given path hop limits if ST is disabled. As the number of hop limit increases, the maximum number of clusters supported decreases. Since the number of hops is primarily constrained by the number of clusters in relation of $h \leq n-1$, the maximum cluster size eventually converges at 10 for hop limit 5 to 9. Figure 6(b) shows the case that ST is enabled for these VLANs containing more than two clusters. The maximum cluster size stays the same for the one hop case since no ST needs to be enabled. Increasing the hop limit, the result converges to 7 clusters, for a maximum of 6

hops path. The IBM BladeCenter typically contains 14 blades (hosts) in one chassis (cluster). Therefore, a maximum of 98 hosts can be supported if ST is enabled and 140 hosts for disabled ST. However, we can overcome this limit by using hierarchical optical networks and electronic edge routers, or using switches with higher available number of VLANs and STs. But the study is beyond the scope of this paper.

If the switch firmware support flooding detection and suppression by disabling the VLAN that contains loops, ST is not needed. As we have seen from our VLAN setup, the shortest path (direct or multi-hop) is always loop-free so no ST is actually needed if the application only picks shortest path VLANs. Of course, as we have discussed in multi-path routing, ST may provide more choices of path (as VLAN 1 can be used in Figure 5(b)).

C. Initial VLAN Configuration on Hosts

The setup of VLAN interfaces on hosts is completely static and only done once at the beginning. No privileged access is needed when the software is running. Preferably, each host is preconfigured with all VLANs so that each host can always find the right virtual interface regardless of the VLAN memberships of their home cluster. Therefore, topology reconfiguration does not affect VLAN set up on hosts. Each VLAN is binded to a different IP subnet. The number of IP subnets required equals the number of VLANs computed by Equation 1. The 10Gbps NIC on each host is virtualized an IP interface per VLAN. The total number of IP addresses that are needed for b hosts is $b\sum_{k=1}^{\min(h+1,n)}\binom{n}{k}$. Although the number of IP addresses becomes nontrivial as n grows, many HPC clusters use internal networks with an abundant number of internal IP addresses. Another possible solution is to enable VLAN tag insertion at application level. It can be achieved by changing the TCP/IP stack or NIC TCP offload engine but the solution is limited in practice because it requires kernel patches and introduces security holes.

V. Performance Measurement

All experiments are done on the prototype system shown in Figure 1. The first experiment is to measure the total reconfiguration time for a new network topology. Total reconfiguration time is defined as the maximum waiting period before all VLAN routes are available (hosts on the same VLAN are reachable by pinging each other) after a topology reconfiguration command is issued. The software component is running on an Intel Xeon 2.66GHz CPU. Initially, PVSTs are assigned according to Figure 4. All Nortel XFPs are enabled with fast port forwarding, which forwards packets at best effort before STs are stabilized. The total configuration time for eight different changes of topology is tested. Test scenarios are shown in Figure 7. Each reconfiguration experiment is denoted as a labeled arrow in the chart. We show the maximum, average, minimum, and the percentage of 98% confidence interval of the reconfiguration time for each experiment, in a total of repeated 20 tests (experiment 5 has 100 runs). The result for the eight experiments are shown in Table I.

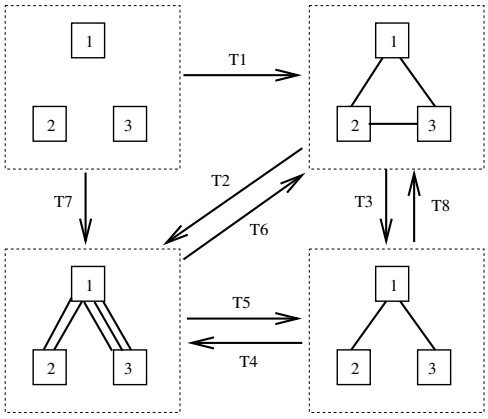

Fig. 7. Configuration time test scenarios.

Test	Max.	Min.	Avg.	CI (98%)
T1	177	167	169.80	$\pm 0.50\%$
T2	6241	6222	6224.15	$\pm 0.02\%$
T3	307	167	216.75	$\pm 1.84\%$
T4	6176	6169	6171.30	$\pm 0.01\%$
T5	1207	174	395.87	$\pm 14.53\%$ (100 runs)
T6	7173	6227	6415.95	$\pm 2.25\%$
T7	6184	6173	6176.75	$\pm 0.02\%$
T8	1175	1169	1170.75	$\pm 0.09\%$

TABLE I
TOTAL TOPOLOGY RECONFIGURATION TIME FOR DIFFERENT RECONFIGURATION SCENARIOS IN MILLISECONDS.

Nortel switches and Calient switch are reconfigured in paralleled to reduce the switching time. Theoretically, the total configuration time consists of the maximum of Calient optical switching time and Nortel configuration time plus an extra waiting time for building forwarding tables. The Calient switch is able to create a bulk of connections in $110ms - 130ms$ and delete a bulk in $20ms - 30ms$[5]. The control network delay to their management modules is about $0.1ms$. Due to the difficulty of measuring the exact waiting time necessary for Nortel switches to enable the path[2], we wait $0s$, $1s$ or $6s$ to find the best effort waiting time after Calient is switched and Nortel has acknowledged commands. The variance of configuration time for different experiments is due to the variance of Nortel configuration response time and the extra waiting time.

In our system, a network change without trunking additional links, such as T1, T3, T8, can be done fairly fast. But routes in experiments that involve trunking, such as in experiments T2, T4, T6, and T7, are not ready on all VLANs in some of the 20 repeated runs until six-second delay. Experiment T8 needs an additional one second. We also find that the switch response time for deleting trunked links is not stable. The variance for T5 is very high even for an 100 runs. Changing a non-loop topology to a loop one, like T6 and T8, takes longer than the reverse direction. However, the worst waiting time is much

[2]We *ping* hosts to check if all routes are available. However, the switched forwarding table may get "stuck" if we send *ping* before the route is ready. We cannot delete unreachable host because they are connected once the network is ready. Therefore, the waiting time is the best effort upperbound.

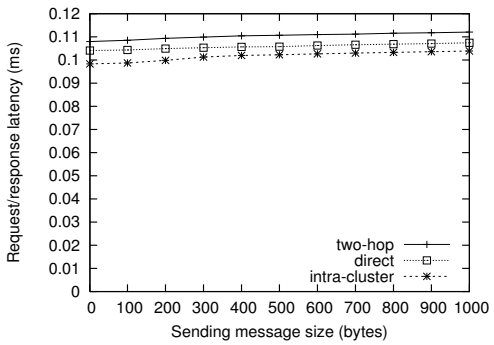

Fig. 8. Comparison of delays on intra-/inter- cluster routing using netperf.

less than the reported $50s$ for ST convergence. The minimum amount of interruption is less than $200ms$. Note that trunking is an integral part of network reconfiguration process itself regardless of any routing mechanism. The performance may vary depending on vendor specific switch implementations. Without the presence of trunking, network reconfiguration can be finished in less than 2 seconds. ST convergence of routes on multiple VLANs requires less extra time (about $1s$ in T8) time than changing trunking states (about $6s$ in T4). The results show that switching firmware should improve dynamic trunking to achieve better performance. The extra $1s$ overhead is mainly caused by the need to converge routes on VLAN 4 (the Nortel switch does not have flooding suppression enabled). However, the routes on VLAN 4 are actually never used. We can reduce that extra $1s$ if the switching hardware supports flood suppression.

The second experiment is to measure the request/response latency of three types of routes: intra-cluster, single hop and two hops. One round of request/response tests for a message size of k includes the entire process of a sender sending k bytes to the receiver, the receiver sending back 1 byte response after receiving the last byte, and the sender receiving the response. The total latency then consists of the sender's latency to send k bytes to the network, the receiver's processing delay, and one round trip time. Figure 8 presents the results of request/response latency between two blades, using network performance measurement tool *netperf*. Each data point is an average with a $\pm 5\%$ percentage interval of 95% confidence. The result shows that the delay of single hop routing is about $5\mu s$ more than inter-cluster routing, and two hops adds another $5\mu s$. Each hop adds a $2.5\mu s$ one way delay in the Nortel switch. No additional delay is introduced by routing on different VLANs. The data sending delay increases as message size increases.

VI. System S Integration

The solution described in this paper has been prototyped and used in a stream processing system designed by IBM (System S)[5]. The software architecture of System S is illustrated by Figure 9. The centralized part of System S includes SODA (Scheduling Optimizer for Distributed Applications)[14], dispatcher, streaming Job Management (JMN)[15], Resource Management (RMN), etc. The most relevant part is SODA that schedules PE-host deployment, network topology and routing path between PEs. The distributed part known as the Stream Processing Core (SPC)[16] manages the Processing Elements (PEs) and the stream connections between them. A PE is a fragment of a distributed stream processing application which can be deployed independently onto a host. It consumes and generates one or more streams which in turn may be consumed by other PEs.

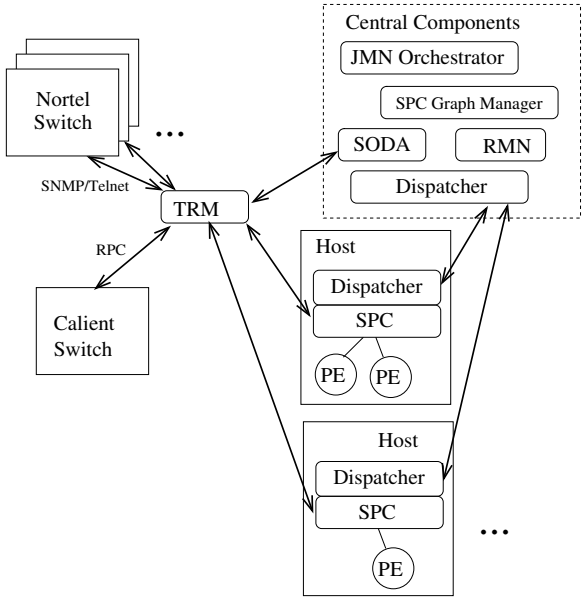

Fig. 9. System S with TRM integration to use the OCS network.

A new software component, called Transport Manager (TRM), was built to coordinate network reconfiguration and PE communication through the optical network. TRM is responsible for receiving reconfiguration decisions from SODA, controlling Calient MEM switch and Nortel switches to reconfigure the optical network, and notify SPCs to change corresponding transports to use for each pair of PEs when network reconfiguration is ready. TRM also collects network state information, such as the number of XFPs installed on each Nortel switch. It monitors each fiber connection, implementing fault tolerance procedures and notifying the rest of the system, should a failure occur.

The network reconfiguration procedure is the following. All VLANs on Nortel Switches (using Algorithm 1 and VLAN IP addresses on each host (using the method described in Section IV-C) are statically configured before System S starts up. At run-time, each PE listens on a dynamic chosen port (specific for each PE-PE connections) of all VLAN IP addresses (including the 1Gps backup Ethernet IP) of the host and receives PE connections from one or more of them. The sender PE is notified with the VLAN IP address to use so the destination PE can receive the data from the chosen VLAN path. Since the optical topology may not be fully connected due to the limited number of XFPs, the 1Gbps shared Ethernet

connection is always available and is the default transport for all new PE connections. When SODA decides a new optical topology and PE routing scheme, TRM reconfigures the optical network and notifies SPCs to switch all the appropriate sending PEs to use the new paths.

Here we have some further discussions about our experience with the all-optical switch for Stream S using the routing infrastructure. The static VLAN and IP assignment approach integrates well with the current System S implementation. In our system, PE communicates with other PEs via PE ports that are binded to particular IP addresses. Notifying changing of transport via IP addresses is the most efficient way for SPCs. Also, the available routing paths are automatically updated at Layer 2 once the optical network is reconfigured. We use trunking and enabled spanning tree to maximize reconfigurability and reliability of the optical transport. According to the analysis in Section IV-B, we can support up to 98 hosts (hundreds or thousands of PEs as more than one PEs can run on the same host).

All affected PE connections are temporarily switched onto the electronic network during optical network reconfiguration so data rate drop can occur during the period. However, compared to other load balancing solutions, like moving PEs to adjacent hosts or duplicating PEs, our methods show many advantages. First, PEs are still connected during the reconfiguration period at a reduced rate. It is a clear advantage for non-duplicable PEs that have to be shut down before moving. Second, reconfiguration time can achieve $200ms$ that is much faster than moving PEs (taking seconds) or duplicating PE paths (takes seconds to minutes) [3]. From the point of view of cluster maintenance, the all-optical optical switch is not bounded by future bandwidth increase and uses much less energy than electronic ones.

To minimize the impact of temporary data rate loss, we can further improve our system by reducing the reconfiguration time. From the result shown in Section V, the most switching overhead is due to trunking operations. If the Ethernet switch can be improved to provide faster trunking and spanning tree convergence, we can expect more than 80% reduced reconfiguration time. In addition, the switch operating system should improve its support dynamic reconfiguration and notification so the signaling can be done in tens of milliseconds that is close the physical MEM switching limit.

VII. Conclusion

The main challenge of integrating all-optical MEM switched LAN is to design a routing infrastructure that is self-adaptive to network topology changes and provides high utilization. As a solution, we proposed a multiple VLAN-based routing infrastructure that can be readily implemented on commercial networks without additional firmware support. The initial configuration of per-VLAN-STs is completely static. Up to 98 hosts can be supported in our prototype network with an opportunity of further scaling up. Less than $200ms$ optical network reconfiguration time is achieved for the best scenarios. We learn that 84% of the overhead for pro-longed reconfiguration time is spent on handling link trunking. Improvement on electronic edge switch firmware can potentially speed up the configuration process. The delays of intra-cluster, direct and multi-hop routing are measured. Finally, We show the integration of the routing infrastructure into IBM stream processing system (System S). Our routing solution on a reconfigurable high-bandwidth optical network shows an advantage over existing load balancing mechanisms used by System S.

[3]The PE relocating time depends on the type of PEs. It is more complex and less desirable to move state PEs then stateless PEs.

Acknowledgment

We wish to thank the extended team at the IBM T. J. Watson Research Center, especially Philippe L Selo for the support on System S transport and Laurent Schares for the support on optical systems.

References

[1] S. Sumimoto, A. Naruse, K. Kumon, K. Hosoe, and T. Shimizu, "PM/InfiniBand-FJ: a high performance communication facility using infiniband for large scale pc clusters," *High Performance Computing and Grid in Asia Pacific Region, 2004. Proceedings. Seventh International Conference on*, pp. 104–113, July 2004.

[2] T. E. Stern and K. Bala, *Multiwavelength Optial Networks: A Layered Approach*. Prentice Hall, 2002.

[3] K. J. Barker, A. Benner, R. Hoare, A. Hoisie, A. K. Jones, D. K. Kerbyson, D. Li, R. Melhem, R. Rajamony, E. Schenfeld, S. Shao, C. Stunkel, and P. Walker, "On the feasibility of optical circuit switching for high performance computing systems," in *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. Washington, DC, USA: IEEE Computer Society, 2005, p. 16.

[4] E. Redmond, Ian; Schenfeld, "Optical interconnection network," *U.S. Patent*, no. 5414819, May 1995. [Online]. Available: http://www.freepatentsonline.com/5414819.html

[5] L. Schares, X. J. Zhang, R. Wagle, P. S. Deepak Rajan, S.-P. Chang, J. Giles, K. Hildrum, D. Kuchta, J. Wolf, and E. Schenfeld, "A reconfigurable interconnect fabric with optical circuit switch and software optimizer for stream computing system," in *Optical Fiber Communication Conference and Exposition (OFC'09)*, San Diego, California, USA, Mar. 2009.

[6] K. Elmeleegy, A. L. Cox, and T. S. E. Ng, "On count-to-infinity induced forwarding loops ethernet networks," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.

[7] "IEEE standard for local and metropolitan area networks media access control (mac) bridges," *IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998)*, 2004.

[8] S.-A. Reinemo and T. Skeie, "Effective shortest path routing for gigabit ethernet," *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 6419–6424, June 2007.

[9] "IEEE standards for local and metropolitan area networks. virtual bridged local area networks," *IEEE Std 802.1Q, 2003 Edition (Incorporates IEEE Std 802.1Q-1998, IEEE Std 802.1u-2001, IEEE Std 802.1v-2001, and IEEE Std 802.1s-2002)*, 2003.

[10] S. Sharma, K. Gopalan, S. Nanda, and T. Chiueh, "Viking: a multi-spanning-tree ethernet architecture for metropolitan area and cluster networks," *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2283–2294 vol.4, March 2004.

[11] K. Lui, W. Lee, and K. Nahrstedt, "STAR: a transparent spanning tree bridge protocol with alternate routing," 2002. [Online]. Available: citeseer.ist.psu.edu/lui02star.html

[12] *Application Guide: Nortel 10Gb Ethernet Switch Module for IBM BladeCenter*, Blade Network Technologies, Santa Clara, CA, January 2007.

[13] J. Q. J. D'Ambrosia; S. rogers, *XAUI: an Overview. Version 1.0*, 10 Gigabit Ethernet Allianc, March 2002.

[14] J. Wolf, N. Bansal, K. Hildrum, S. Parekh, D. Rajan, R. Wagle, and K.-L. Wu, "SODA: An optimizing scheduler for large-scale stream-based distributed computer systems," in *Middleware '08: Proceedings of the 9th international Middleware Conference*, Dec. 2008.

[15] G. Jacques-Silva, J. Challenger, L. Degenaro, J. Giles, and R. Wagle, "Towards autonomic fault recovery in system-s," in *ICAC '07: Proceedings of the Fourth International Conference on Autonomic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, p. 31.

[16] L. Amini, H. Andrade, R. Bhagwan, F. Eskesen, R. King, P. Selo, Y. Park, and C. Venkatramani, "SPC: a distributed, scalable platform for data mining," in *DMSSP '06: Proceedings of the 4th international workshop on Data mining standards, services and platforms*. New York, NY, USA: ACM, 2006, pp. 27–37.