Application Note 007: Using NSE Technology in Multigigabit Multiprotocol Label

1 MULTIPROTOCOL LABEL SWITCHING

MPLS[1] is an optimized switching technology for IP networks, which can be used with any underlying transport technology (e.g. ATM, Frame Relay, etc.). In MPLS, IP packets are prepended with a Layer 2 routing label as they enter an 'MPLS cloud', and then all forwarding within the cloud is performed based on that label. The label is then removed as the packet leaves the cloud and the packet is forwarded based on Layer 3 information such as the destination IP address. Within the 'MPLS cloud' packets are routed along label-switched paths (LSP) which are similar in concept to a PVC in ATM networks. An LSP is formed by the concatenation of a number of label-switched hops, allowing the packet to be forwarded from one Label Switching Router (LSR) (a router that supports MPLS) to another across and ISP's network. The path chosen by the LSR is not limited to that which would be used if OSPF were the route determination mechanism. Alternative paths may be used based on traffic or QoS parameters. There are three functions of an LSR that we must consider. At entry to the 'MPLS cloud' the label must be assigned to the packet. Inside the 'MPLS cloud' packets must be forwarded to the next hop defined for the label. At exit from the 'MPLS cloud', the label must be removed and the packet forwarded based on the destination IP address. An LSR at the edge of the cloud must be able to perform all of these functions. Internal LSRs may only need to perform MPLS forwarding, however in practice standard equipment is often used.

The principle objective of MPLS is to allow the Internet backbone to scale while providing low, predictable latency, high bandwidth and traffic engineering control. All three of the routing / switching processes required in an end to end MPLS network contribute to the latency component, thus the network quality will be improved by eliminating delays caused by any of these processes.

A packet received on the ingress router (entry to the MPLS cloud) must first be identified and labeled. The Label attached identifies the LSP that will be used to reach the destination (egress from the MPLS cloud). This label will be based on the values of the Source and Destination IP addresses (32 bits each), the Source and Destination Ports Ids (16 bits each) and the physical ingress port ID (16 bits) (this is used to identify the physical network that the packet originated from, thus allowing VPN and Virtual Router functionality). This information will be used to identify a record in the routing table. The routing table may also hold mask values for each entry, allowing a single routing entry to identify a whole sub-network, or packet classification. Once the routing entry has been identified, the label is attached to the packet, and the packet forwarded to the port identified by the routing entry.

A packet received that already has an MPLS label is switched to the next hop on the LSR specified by the label following the MPLS protocol. This requires a lookup of the LSR table to determine the outgoing port to be used. This is performed by a 32bit exact match lookup on the table.

If the label attached to an incoming packet is identified as an egress packet (leaving the MPLS cloud) then the non-MPLS egress port is identified and the MPLS label removed prior to forwarding the packet. This operation will require a traditional ARP resolution if the outbound port is Ethernet.

In order to implement full MPLS functionality in a router, three distinct lookup tables are required for the three possible processes that need to be performed. The LSR determination table is likely to be large and the searching algorithm is complex as each entry needs to have it's own mask to provide for route aggregation. The MPLS routing table will also have a large number of entries as

[1] RFC-2430,2547,2702,2917 www.rfc-editor.org

many LSRs may be defined between each point-to-point to implement QoS and CoS traffic engineering within the cloud.

2 SOLUTIONS

There are two alternative strategies that can be taken to provide the routing / switching processes required at the speeds required by Gigabit Ethernet and OC-48 or greater. High performance network processors can be deployed to process the complex search algorithms required, or 'Network Search Engine' technology can be used to accelerate the routing process.

2.1 Network Processor

The network processor is a specialized micro-processor which is specifically designed to meet the needs of the networking market. The architecture of the processor is customized to support the address sizes and common operations of networking equipment such as the lookup process required in routers that support MPLS. Typically network processor architectures allow multiple processors to be designed into the solution providing the required processing power to keep up with Gigabit data rates. A possible architecture using a network processor is shown in Figure 1.

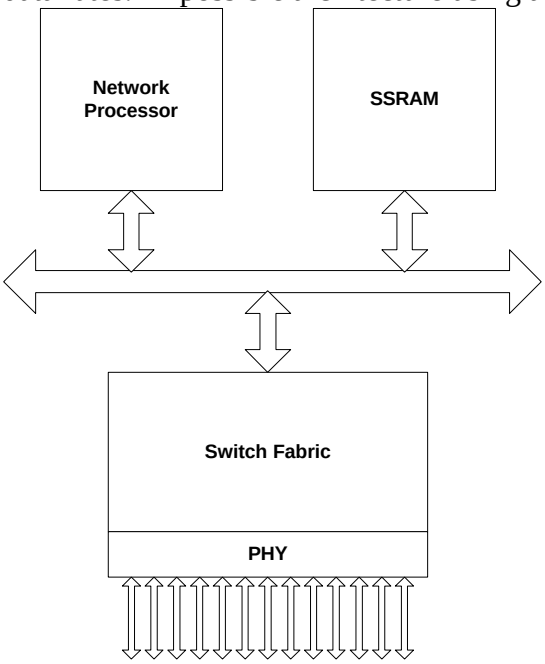

FIGURE 1

2.1.1. Advantages

- Complex routing algorithms can be supported.
- Algorithms are coded in software and can therefore be updated and improved.

2.1.2. Disadvantages

- Relatively high cost as the processor is application specific.
- Significant SSRAM space is required to support routing tables to optimise lookup performance.

- High clock speed required in order to resolve routes in the required period.

2.2 Network Search Engine Technology

As a cost effective alternative 'Network Search Engine' technology (such as the LNI7020 product from Lara Networks, Inc.) may be deployed, either in conjunction with a Network Processor, or with a standard RISC microprocessor. A typical architecture is shown in Figure 2.

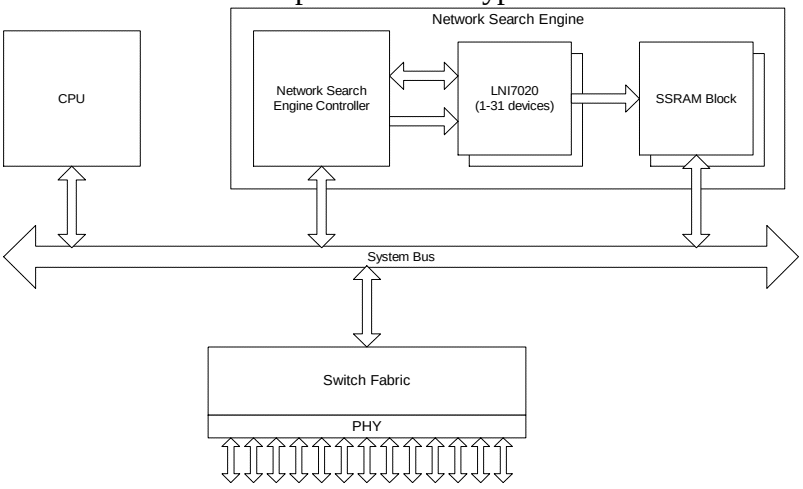

FIGURE 2

The 'Network Search Engine' is used to hold the routing tables required by the design. The LNI7020 'Network Search Engine' can be configured to hold the multiple tables that will be required by the design. The configuration may be modified dynamically and so the same design can be used for both an MPLS edge router and those within the core of the 'MPLS cloud'.

2.2.1. Advantages

- Network Search Engine performs route table lookup, so lower clock speeds can be used.
- Smaller SSRAM block required as route table is stored in the Network Search Engine with better bit efficiency than associated SSRAM techniques.
- Reduced time to market as complex software algorithms do not need to be developed.

2.2.2. Disadvantages

- Designers need to learn how to deploy this new technology.

3 IMPLEMENTING MPLS USING NETWORK SEARCH ENGINE TECHNOLOGY

The route tables would be implemented using a Network Search Engine such as the LNI7020[2] from Lara Networks, Inc. The LNI7020 part is a 16K x 136bit Search Engine and can be cascaded to up to 31 devices, supporting tables of up to 992K entries. The LNI7020 can be configured to support tables of 34, 68, 136 or 272 bits wide and can support multiple tables of differing sizes. In this application we will implement three tables for the three lookup functions that are required by our routing engine.

3.1 Architecture

A typical routing engine architecture is shown below:

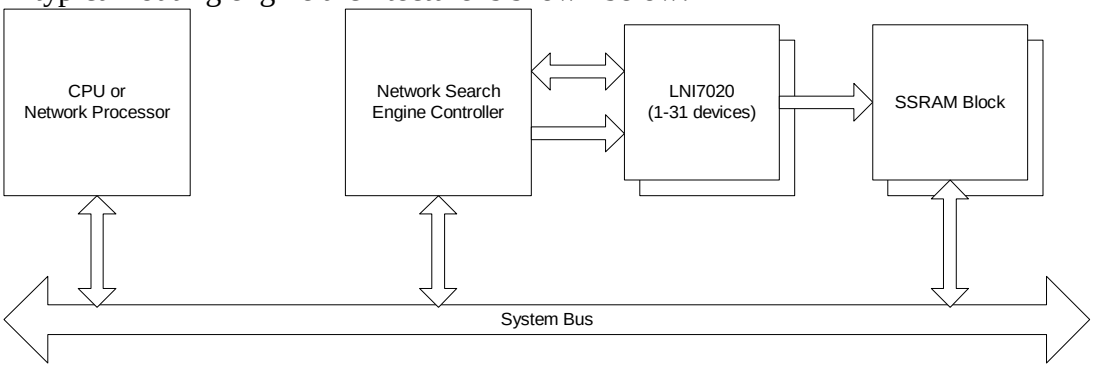

FIGURE 3

The Network Search Engine controller could be either a custom ASIC or it could be the Search Engine Controller LNI8010[3] from Lara Networks, Inc. A high performance RISC CPU or Network Processor would be used with a system bus clock of 200MHz or above, to achieve maximum performance. In this application a table size of 200,000 entries would be required. 12 LNI7020 devices will be required. As core networks grow, the number of routes and thus the table size may increase beyond this level. The Network Search Engine can then be expanded to support the larger tables without any degradation in performance.

The Network Search Engine will be configured to implement three tables. The size of the tables will be controlled by software so that the ratio may be determined based on the function of the router (MPLS edge or core) and may also be dynamic as requirements change. The simplest of the tables will be used for ARP in egress packets. For implementation details and considerations please refer to "Implementing ARP using Network Search Engine Technology"[4]. The other two tables would be implemented as below:

3.2 Core MPLS route table

The core routing table will consist of 32 bit label entries which will identify the next label switched hop for the incoming packet. The Network Search Engine table will be configured to be 68 bits wide. As all searches will be for exact match, the table entries will all be unique and can therefore be held in any order in the array. The resultant word in the SSRAM block will be 64 bits wide. This

[2] LNI7010/LNI7020 Network Search Engine Datasheet, Lara Networks, Inc.
[3] LNI8010 Network Co-processor Datasheet, Lara Networks, Inc.
[4] Implementing ARP using Network Search Engine Technology, Lara Networks, Inc.

will be used to identify the outbound port and any additional characteristics required for QoS purposes. The organization of the table will be as shown in figure 4.

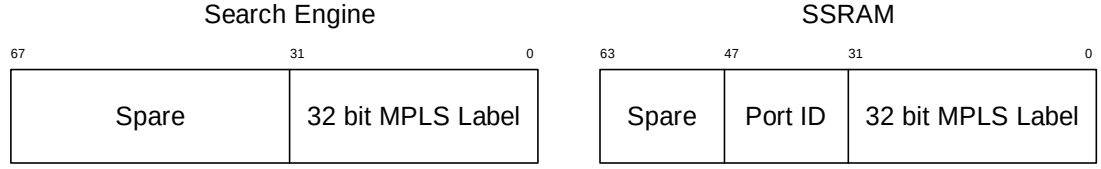

FIGURE 4

3.3 Ingress MPLS route table

The ingress routing table will be more complex. This table converts the incoming packet's Layer 3 routing information into the label required for MPLS routing. Entries in this table will be configured as 136 bits wide and will be organized as shown in figure 5. The resultant word will also be 64 bits wide and will contain the label and other parameters required for QoS and Traffic Engineering.

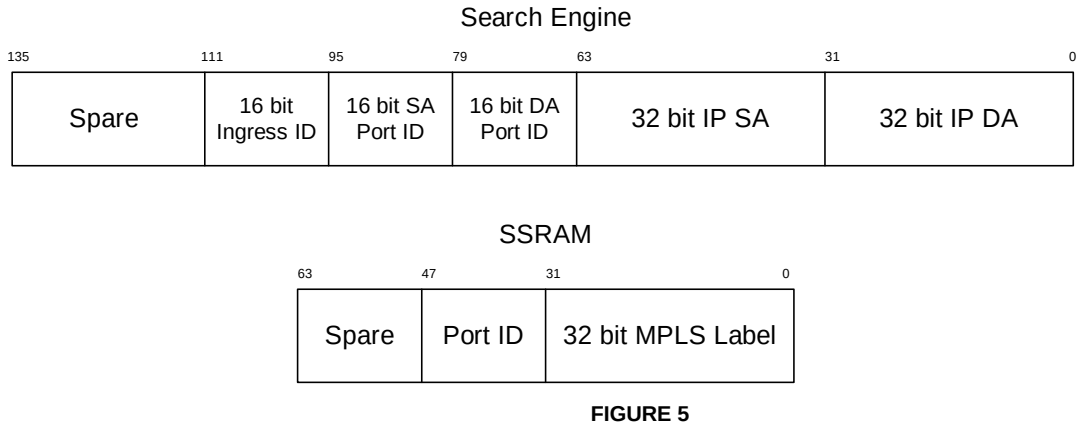

FIGURE 5

3.4 Configuration

Each of the devices used in the array may be configured in four quadrants. Each quadrant can be either 8K x 68 bits, 4K x 136 bits, or 2K x 272 bits. In this application, we will configure tables for both 136 bit and 68 bit entries, by setting the CFG bits in the Command Register for each of the devices. For full details see the LNI7020 Network Search Engine Datasheet.

3.5 Search Performance (Core Routing)

The maximum performance that will be achieved will be when the Network Search Engine is operated with a full pipeline. Given a system bus that is 64 bits wide, one cycle is required to load the compare word into the search engine. After the search latency has passed, the match address will be presented on the SSRAM address bus interface and after the access delay of the SSRAM device, the result word will be read from the SSRAM data bus. If our host CPU operates with a system bus of 200MHz, then this can be interleaved into the search words that are presented to the Search Engine.

The performance that can be achieved for a 192K record table made up from 12 of 83MHz LNI7020 devices is summarized in Table 1.

TABLE 1

Operation	Non-pipelined	Pipelined
Write Search Word	12ns	12ns
Latency	72ns	n/a
Read Result Word	5ns	5ns
Total response time	**89ns**	**17ns**

The CPU will also need to receive and interpret the address received from the packet processors which will add to the response time. While the traffic rate is low (i.e. below 5M packets / sec) then most operations will be resolved using the search engine non-pipelined. The performance enhancement of the pipeline comes into action when it is most important at high traffic densities. Table 2 details the number of ports and the data rates that may be supported for core routing functions.

TABLE 2

Average Packet size	# 1Gbps ports	# 2.4 Gbps (OC-48) ports	# 9.6 Gbps (OC-192) ports
64 bytes (i.e. VoIP)	40	16	4
128 bytes	80	32	8
512 bytes (i.e. www)	400	128	32
1K bytes (i.e. FTP)	800	256	64

The results in this table demonstrate that using Network Search Engine technology can be used to provide MPLS core routing, even up to OC-192 data rates.

3.6 Search Performance (Edge Routing)

The maximum performance that will be achieved will be when the Network Search Engine is operated with a full pipeline. Given a system bus that is 64 bits wide, two cycles are required to load the compare word into the search engine. After the search latency has passed, the match address will be presented on the SSRAM address bus interface and after the access delay of the SSRAM device, the result word will be read from the SSRAM data bus. If our host CPU operates with a system bus of 200MHz, then this can be interleaved into the search words that are presented to the Search Engine.

The performance that can be achieved for a 192K record table made up from 12 of 83MHz LNI7020 devices is summarized in Table 3.

TABLE 3

Operation	Non-pipelined	Pipelined
Write Search Word	24ns	24ns

Latency	72ns	n/a
Read Result Word	5ns	5ns
Total response time	**101ns**	**29ns**

The CPU will also need to receive and interpret the address received from the packet processors which will add to the response time. While the traffic rate is low (i.e. below 5M packets / sec) then most operations will be resolved using the search engine non-pipelined. The performance enhancement of the pipeline comes into action when it is most important at high traffic densities. Table 4 details the number of ports and the data rates that may be supported for core routing functions.

TABLE 4

Average Packet size	# 1Gbps ports	# 2.4 Gbps (OC-48) ports	# 9.6 Gbps (OC-192) ports
64 bytes (i.e. VoIP)	20	8	2
128 bytes	40	16	4
512 bytes (i.e. www)	200	64	16
1K bytes (i.e. FTP)	400	128	32

The results in this table demonstrate that using Network Search Engine technology can be used to provide MPLS edge routing, even up to OC-192 data rates.

4 CONCLUSION

The key benefit of MPLS to Network Service Providers, is the ability to apply advanced traffic engineering across the network which is not possible with traditional CIDR routing. MPLS networks will provide the backbone to the new Internet as data rates increase to OC-192 and beyond. Implementing routers that support MPLS is a complex challenge for systems designers as routing table sizes grow, and the granularity of the service metrics that are used reduces. Network Search Engine Technology simplifies the task for system designers by providing router table implementations that deliver reliable, deterministic response times and can be expanded to support greater capacity without degrading performance. The LNI7000 family is particularly suited to this task as they are able to support multiple tables in the array. This allows combination routers that perform edge and core routing functions to be combined into a single physical design. The ternary nature of the LNI7000 family is key to implementing route aggregation within MPLS, thus reducing the size of routing tables required. The discussion of MPLS in this paper has been centered on IP networking, however, the same principles can be applied to the transportation of any other network protocol with adjustments to the addressing capacity. As there are adequate spare bits within the LNI7000 family, multiple protocols could be easily supported by the architecture described in this paper.

CONTACT

Lara Networks, Inc.
110 Nortech Parkway
San Jose, CA 95134
www.laranetworks.com

Tel: 1-408-942-2000
Fax: 1-408-942-2099

