Application Note 006: Implementing Address Resolution Using NSE Technology in Multigigabit IP Network Interfaces

1 ADDRESS RESOLUTION PROTOCOL

The address resolution protocol (ARP) is the standard in Internet protocol (IP) networks for converting media access control (MAC) addresses into IP network addresses and back again as the IP packets are forwarded to the final destination. When a switching device receives a packet on an interface, the layer 2 switching algorithm extracts the IP address of the destination host from the packet header. It is then passed to the address resolution module (ARM), which looks up the IP address in the ARP table and, if the IP address is found, the ARM answers with the MAC address of the destination to which the packet should be forwarded on the next hop. If the address is not found, the packet is either discarded or forwarded to a default port. Subsequently, network interface assembles an ARP packet designed to request the required address and broadcasts it to all devices on the segment. When a reply is received, the information in the packet is used to update the ARP table.

All network interfaces are limited to the finite size of the ARP table, so a least-recently used (LRU) algorithm is used to discard or replace items that are not encountered frequently. A time-out is also implemented for entries that are not used for a certain period of time, in order to prevent incorrect information from persisting in the ARP table (e.g., if a host is moved or its IP address changes).

2 SOLUTIONS

When analyzing the process required to forward the packet, it is clear that the ARP table must be queried each time a packet is received by the network interface host. If the network interface is attached to a host system, the ARP table will have only a few records because each host normally only communicates with a small number of other hosts. If the network interface is a port on a switch or router within the Internet cloud, however, the ARP table might contain thousands of entries. As the number of hosts increases with the growth of the Internet, backbone devices must be able to support table sizes of up to one million entries to avoid packet loss and increased latency. A network device transporting 64-byte packets and using gigabit Ethernet interfaces to support IPv4 has 512 ns available for address resolution if the packets are forwarded at wire speed. Microprocessor-based solutions for ARP lookup are possible for network devices with a small number of ports. For the multiport devices required in the Internet backbone, however, alternative techniques are needed to maintain wire-speed forwarding.

2.1 Parallel Tables

Each port within the networking device acts independent of the others during the address resolution task. It maintains its own ARP table and processor so that resolution can be performed at a full data rate. The table held by each port is a copy of all the hosts known to the networking devices. Figure 1 shows the architecture of such a network device.

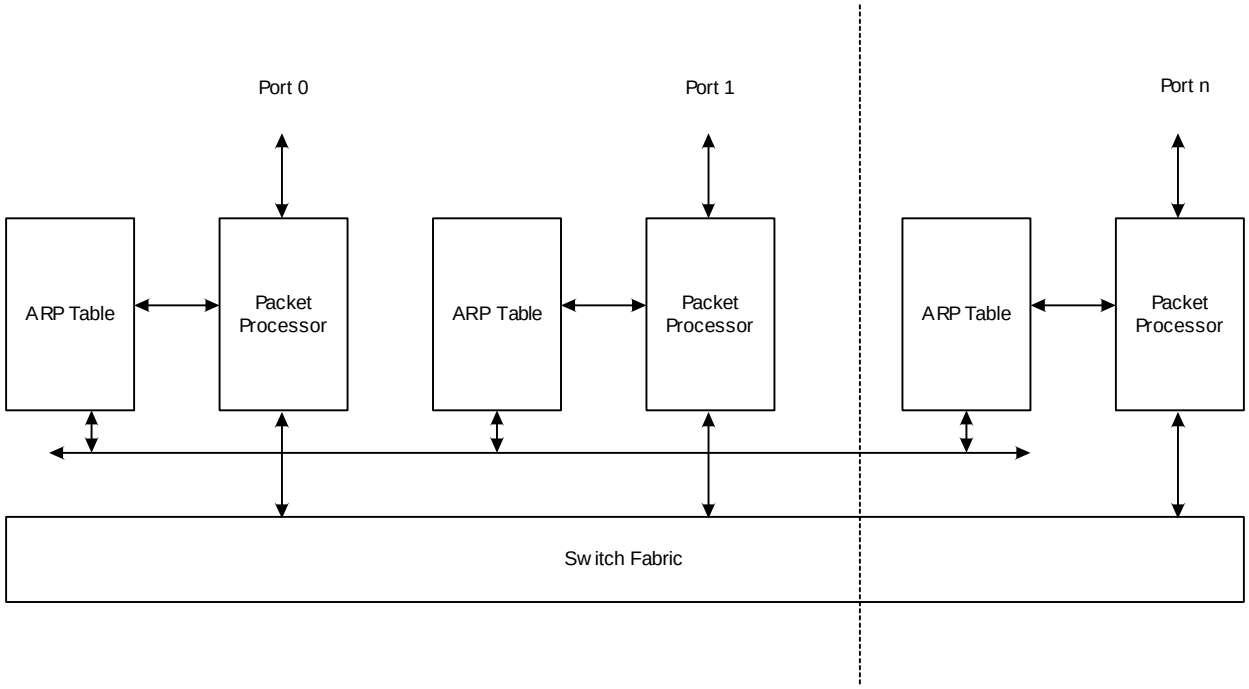

FIGURE 1. NETWORK DEVICE ARCHITECTURE

2.1.1. Advantages

Because each port is independent, high-volume traffic on one port does not affect any other port.

The number of ports implemented in the device is only limited by the capabilities of the switching fabric.

2.1.2. Disadvantages

There is a relatively high cost per port because in order to hold the table each port needs both processing and storage elements.

A mechanism must be implemented to propagate both changes and new table entries through the device.

As the table size grows, all ports are upgraded to support larger tables.

2.2 Centralized Address Resolution Module

The network device has a centralized ARM (CARM) that holds the ARP table for all ports in the device. The CARM is often implemented using a network processor optimized for such tasks. This improves the number and speed of the ports that can be supported. 2 shows the architecture of a network device with a CARM.

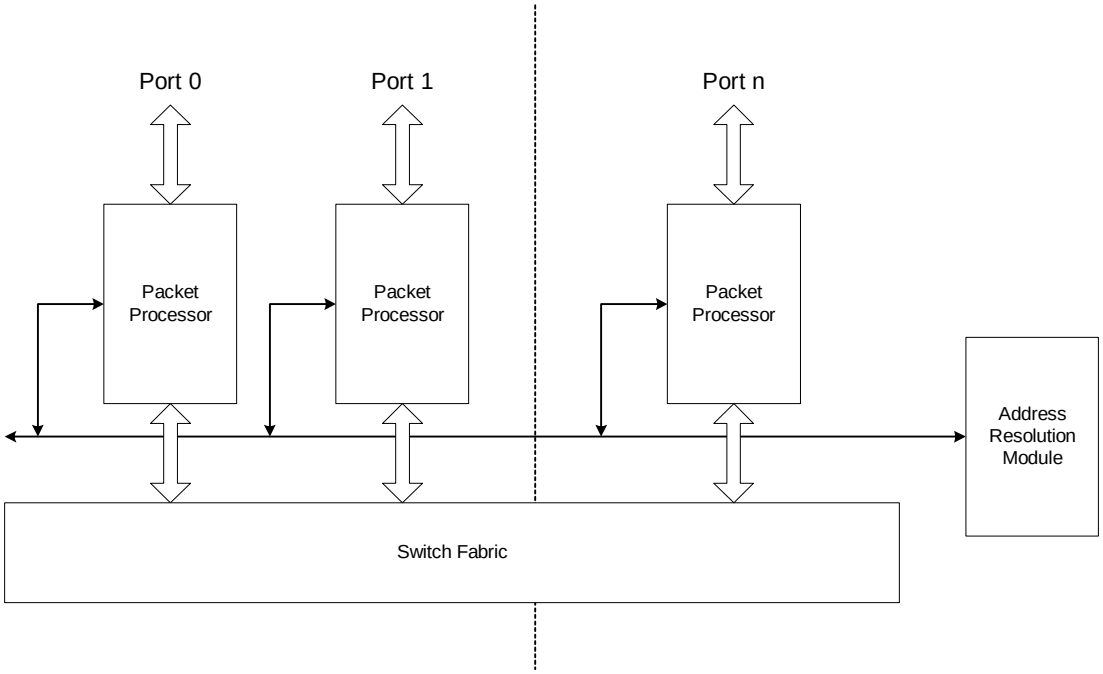

FIGURE 2. CARM NETWORK DEVICE

2.2.1. Advantages

There is simplified table management because only one table needs to be maintained.

There is efficient use of resources because there is no duplication.

The device has a lower cost per port.

Upgrading as the host numbers increase is practical because only the central module needs to be expanded.

2.2.2. Disadvantages

Performance is limited by the capabilities of the central module.

Only a limited number of ports can be supported by the central module.

2.3 Hybrid Approach

To reduce the cost per port of the parallel processing architecture but still support a larger number of ports than the centralized module architecture, some systems employ a hybrid in which groups of ports share a common ARP table. Such an architecture is shown in Figure 3.

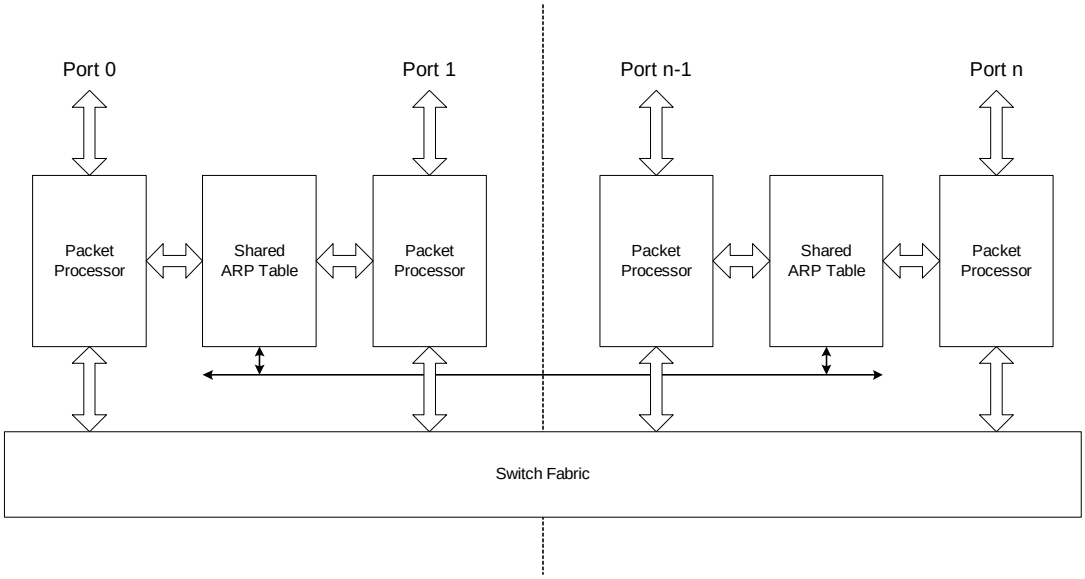

FIGURE 3. HYBRID NETWORK ARCHITECTURE

2.3.1. Advantages

There is a reduced cost per port compared with the parallel processing architecture.

Expansion capabilities are limited only by the switching fabric.

2.3.2. Disadvantages

The overall solution complexity increases as the complexity of the network architecture increases.

Inter-table synchronization and coherency is required.

An upgrade of the table size requires an upgrade of multiple modules.

2.4 Network Database Search Engine

The introduction of network database search engines (NDSEs) and coprocessors enhances the capabilities of the CARM architecture and eliminates the need to adopt a complex hybrid approach. Using an NDSE for address resolution raises the bar for network devices because performance is limited only by the capabilities of the switch fabric backbone. The architecture of such a solution is shown in Figure 4.

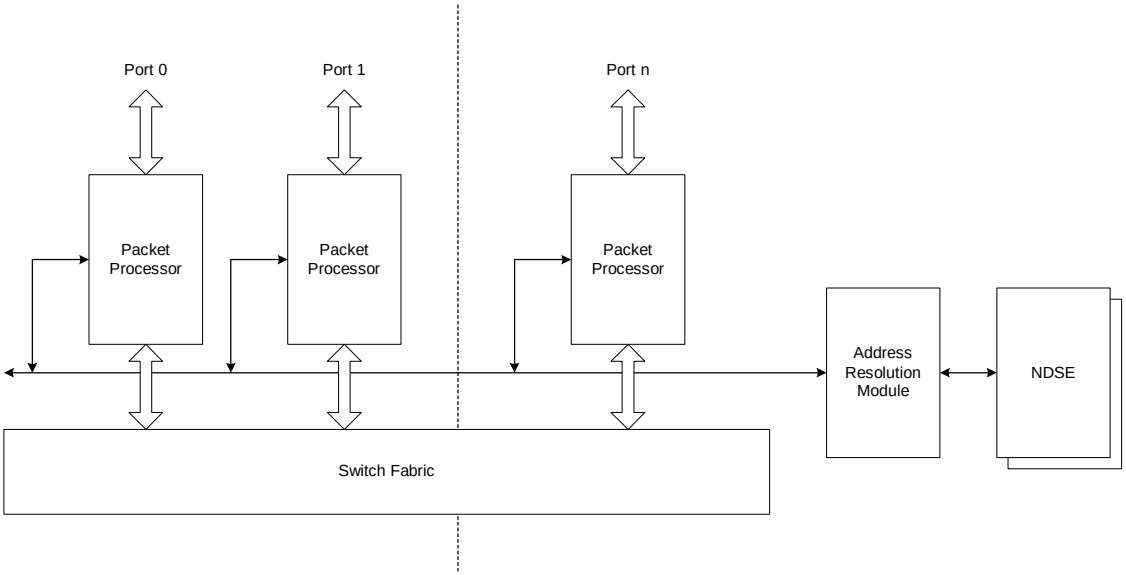

FIGURE 4. ARCHITECTURE INTEGRATING NDSE SOLUTION

2.4.1. Advantages

There is reduced cost per port.

Expansion capabilities are only limited by the switching fabric.

ARM performance reaches up to 100 million searches per second, supporting backbones of up to 20 Gbit Ethernet links for 64-byte IP packets.

Simplified table management because only one table needs maintenance.

Lack of duplication ensures a more efficient use of resources.

Upgrading as host numbers increase is practical because only the central module needs to be expanded. Additional NDSEs can be added to provide capacities of up to 1 million entries.

As table sizes exceed 32K records, adding NDSEs to the architecture provides a component cost advantage over similar SSRAM-based solutions.

2.4.2. Disadvantages

New techniques must be learned to implement NDSE-based devices.

3 IMPLEMENTING ARP USING NDSE TECHNOLOGY

The ARM can be implemented using an NDSE such as Lara Networks, Inc.'s (Lara's) LNI7040 device. The LNI7040 device is a 32K x 136-bit NDSE that can be cascaded to up to 31 devices, and that supports tables of up to 992K entries for this application. The LNI7040 device can also be configured to support tables that are 34, 68, 136, or 272 bits wide, and can support multiple tables of differing sizes. In this application, however, a single table is appropriate.

3.1 Architecture

A typical ARM architecture is shown in Figure 5.

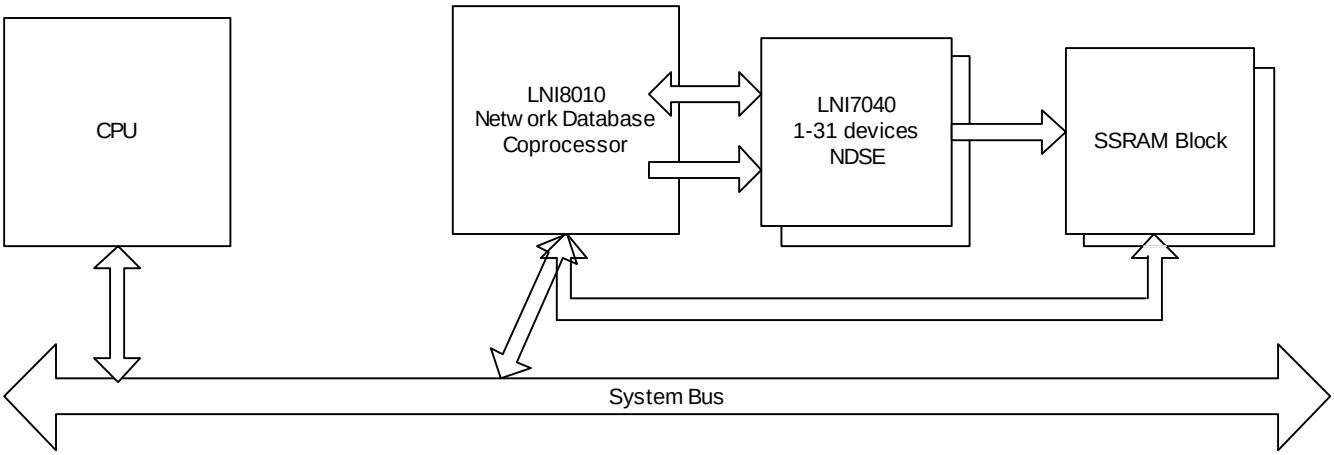

FIGURE 5. A TYPICAL ARM ARCHITECTURE

The NDSE controller can be either a custom ASIC designed for this application or Lara's network database coprocessor, the LNI8010 device. To achieve maximum performance, a high-performance RISC CPU or network processor is used with a system bus (SSRAM bus) clock of up to 100 MHz.

The network database coprocessor can be used to implement a single table containing entries for both 32-bit IP addresses and 48-bit MAC addresses. The corresponding IP or MAC address results are stored in the correct location of the SSRAM block. One bit of the record is used to indicate the address type. Records that have 49 or 33 bits of data therefore need to be implemented.

Our table is constructed of 68-bit-wide records in the NDSE, because this is the nearest configuration we have to the size we need. The unused bits may be utilized in a router or switch design to indicate the port on which the address is located. A GMR is loaded and the value 0 x 0 0001 FFFF FFFF FFFF is used to mask the unused bits during SEARCH operations. The IP address records have bits 32 through 47 written to zero so that the same mask register can be used for both SEARCH operations. The structure of the records is shown in Figure 6.

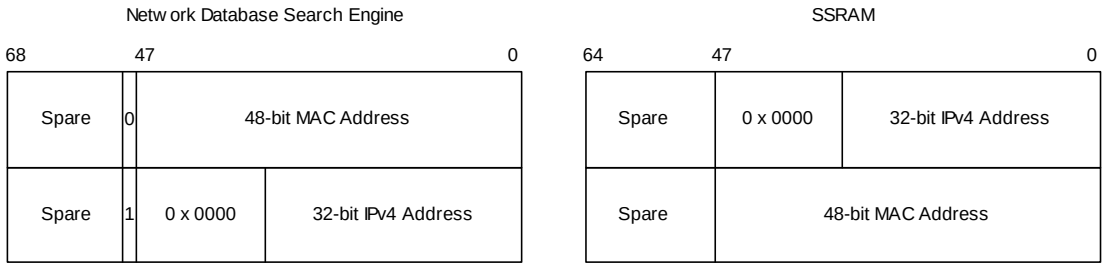

FIGURE 6. RECORDS STRUCTURE FOR TABLE WITH BOTH MAC AND IP ADDRESSES

To configure the NDSE into a single 68-bit-wide table, we must set the configuration (CFG) bits of the command register to 00000000. For full details, see Lara's LNI7040 datasheet.[1]

[1] http://www.laranetworks.com/LNI7040.pdf, *LNI7040 Network Database Search Engine*. Lara Networks, Inc. 2001.

As we are only looking for “exact match” results, the mask array is not used in this application. The data array and SSRAM are programmed with the known IP-MAC address pairs. Each address pair has two entries in our table so that we can perform SEARCH operations with either a known MAC address or a known IP address.

3.2 SEARCH Performance

When the NDSE operates with a full pipeline, the ARM achieves maximum performance. Given that the system’s SSRAM bus is 64 bits wide, three cycles must perform one address resolution: the first cycle must load the compare word into the coprocessor pipeline, the second cycle must start the SEARCH command, and the final cycle must read the results back from the coprocessor pipeline. With the system SSRAM interface running at 100MHz, the proposed subsystem can perform (peak) 33 million address resolutions per second.

The performance that may be achieved for a 64K record table (made up of two LNI7040 devices) is summarized in Table 1.

TABLE 1. ACHIEVABLE PERFORMANCE FOR 64K RECORD TABLE

OPERATION	PIPELINED MODE
WRITE SEARCH Word	10ns
WRITE Command	10ns
READ Result Word	10ns
Total response time	**30ns**

Table 2 details the number of ports and data rates that can be supported by this ARM in the centralized architecture.

TABLE 2. SUPPORTABLE CARM PORTS AND DATA RATES

AVERAGE PACKET SIZE	NUMBER OF 1-GBPS PORTS	NUMBER OF 2.4-GBPS (OC-48) PORTS	NUMBER OF 9.6-GBPS (OC-192) PORTS
64 bytes (i.e., VoIP)	16	8	1
128 bytes	8	4	2
512 bytes (i.e., WWW)	32	16	8
1K bytes (i.e., ftp)	64	32	16

This table demonstrates that by using NDSE technology it is possible to provide centralized address resolution for high-performance multiport network devices, even up to OC-192 data rates.

3.3 Table Management

Because the SEARCH function in this application is always for an exact match and there is no requirement to sort the table in any particular order, the table management process is simplified. The requirements comprise of only the implementation of the following processes.

3.3.1. Add Entry

This process is requested after an ARP packet has been received by an interface, indicating the identification of a new host on a segment. Two entries must be added to the data table for SEARCH by either IP or MAC address. Entries are added to the NDSE by executing a LEARN command. This command writes the data provided to the NDSE's data array to the next available free location, after which it asserts a WRITE cycle on the SSRAM that is used to write the SSRAM portion of the data record.

3.3.2. Delete Specific Entry

The NDSE maintains a data array of registers that hold the addresses of successful SEARCH operations, and bit0 of the data array is used to indicate whether the entry is used or free. This can be used to identify an entry to be deleted. To delete an entry, that entry must be identified, and then a 0 value is written to it in the data array to indicate that it is the record to be deleted.

The sequence of steps to delete an entry is as follows:

Perform a SEARCH command to find the required record.

Wait for the latency of the SEARCH operation.

Perform a WRITE command ("0") into bit0 using the Successful Search Register (SSR) returned by the SEARCH operation in order to provide the WRITE address.

In this application, the delete operation will have to be carried out twice: once for the MAC address, and once for the IP address.

3.3.3. Entry Aging

An ageing algorithm can be created from the spare bits in the data array in order to identify data-array entries that are not used for a period of time. The 16 spare bits can time-stamp the use of each record after each SEARCH with a counter value that is incremented by the host processor every second. Every second, the management process would then perform a SEARCH on these bits to find records that were last stamped 30 seconds ago (if, for example, the timeout is 30 seconds). These records are then deleted. This process would have to be repeated until no further successful matches are found.

As explained in this application note, it is very important to properly initialize all registers in the NDSE. Data and mask arrays can then be programmed with the appropriate data for SEARCH operations.

TABLE 3. TYPICAL BURST ADDRESS REGISTER INITIALIZATION FOR 16K X 136-BIT NDSE TO BURST WRITE

CYCLE	CMD[8:0]	CMDV	DQ[67:26]	DQ[25:21]	DQ[20:19]	DQ[18:6]	DQ[5:0]
1	xxxxxx001	1	Reserved	ID[4:0]	11	Reserved	111011
2	xxxxxxxxx	0	xx... .xx11	11111	11	xxxxx00000000	000000
3	xxxxxxxxx	0	xxxxxxxxxx	xxxxx	xx	xxxxxxxxxxxxxxx	xxxxxxx

4 CONCLUSION

Address resolution is a key issue in the implementation of network interface devices. As data rates increase, lookup performance in the ARP table becomes a critical issue, particularly with multiport devices such as switches and routers. A number of architectures have evolved to address this problem, the simplest being the

CARM architecture. To date, this architecture has been unable to keep up with increasing performance requirements, and so hybrid architectures have evolved. The introduction of Lara's network database coprocessor allows the implementation of multiport devices at data rates of up to OC-192 (and beyond) by using a simpler, centralized architecture. We can thereby avoid the complexities of distributed and hybrid architectures which results in reduced engineering effort and quicker time-to-market for those willing to deploy this technology.

As the number of hosts addressed by network devices in the Internet cloud increases, devices based on NDSE technology can expand to meet the demand with no adverse impact on performance. With current generation parts, table sizes of up to 992K entries can be created that will support up to 496K hosts.

CONTACT

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

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

