Application Note 003: Longest Prefix Match Using the LNI7010 Network Search Engine

1 INTRODUCTION

This document describes the method by which to perform a longest prefix match using the CYNSE70032/LNI7010 Network Search Engine (NSE). We will use IPv4 address lookups as an example by which to illustrate this application. The examples described herein deal with IPv4 addresses only. A similar approach can be used for other applications that use the longest prefix match concept.

2 IP ADDRESS FORMAT

IPv4 addresses are 32-bit binary numbers commonly represented in dotted decimal notation. A three-digit decimal number between 0 and 255 represents each group of eight bits. The three-digit groups are separated by a dot, e.g., xxx.xxx.xxx.xxx.

3 IP PREFIXES AND MASKS

Routers interpret the IPv4 address as being composed of a network address and a host address within the network. The IP prefix refers to the network address portion of the IP address. For example, a machine may have an IP address of 192.168.104.100.

To understand IP prefixes, it is more effective to use the binary representation of each address, which in the case of the address in the previous paragraph would be 11000000.10101000.01101000.01100100. Let us say that the first 18 bits represent the network address (i.e., the network address = 11000000.10101000.01). The remaining 14 bits (101000.01100100) represent the host address within the network. A router would use a prefix mask consisting of 18 “1”s followed 14 “0”s (11111111.11111111.11000000.00000000) to indicate that the network address (or prefix) is the first 18 bits of the IP address. The length of the prefix mask is not fixed, but can vary from network subnet to network subnet.

Routers and IP Lookups. One of the functions of a router is to maintain a routing table so that when an IP address is presented as a lookup key, it retrieves associated data that would include, among other things, the route port. Routing-table entries contain 32-bit IP addresses, but network-address portions can be of varying lengths, as represented by the leading ones in the prefix mask.

4 THE LONGEST PREFIX MATCH CONCEPT

Applications that require the longest prefix match do not consider all 32 IP address bits for the lookup, but only the prefix as indicated by the prefix mask. The addresses and prefixes can be stored in the LNI7010 NDSE as pairs in the data and mask arrays.

A “1” in the prefix mask specifies the bit position where a match is enabled for a lookup operation. It is possible that when the prefix mask is applied, more than one entry in the table matches a given address. This can be seen in the following set of four entries (and their respective masks) with addresses A, B, C, D (see Table 1).

TABLE 1. SAMPLE ENTRY TABLE

11000000.10101000.01101000.01100100	IP address A
11111111.11111111.11111111.00000000	Prefix mask for A
11000000.10101000.01101001.01100100	IP address B
11111111.11111111.11111110.00000000	Prefix mask for B
11000000.10101000.01101010.01100100	IP address C
11111111.11111111.11111100.00000000	Prefix mask for C
11000000.10101000.01101000.11100100	IP address D
11111111.11111111.11111111.10000000	Prefix mask for D
11000000.10101000.01101000.01111111	IP address being looked up L

When IP address L is presented as a lookup key, entries A, B, and C will all match L as a result of the address mask. Entry D would not match because of the mismatch in bit 7. The concept of “longest prefix match” specifies that the lookup operation should resolve multiple matches by reporting only the matched entry with the longest prefix mask. According to the example, then, a longest prefix match would report a match for entry A.

The LNI7010 NDSE is ideally suited for implementing longest prefix match operations. The data array is used to store the entry value. The mask array is used to store the prefix mask. The prefix masking operation is built in and occurs automatically in the LNI7010. The table management software sorts entries in order of decreasing prefix length so that the entry with the longest prefix has the lowest address. The lowest address location has the highest priority, and therefore in the above example IP address A has a lower address and a higher priority when compared to IP addresses B, C, and D. The LNI7010 is designed so that when multiple matches occur it outputs the index from the lowest address location. This way, the longest prefix match is easily implemented.

The LNI7010 is both flexible and fast. It will work with any length of subnet mask up to the word width, and is also capable of generating a longest prefix match result at the maximum data rate.

In summary, the steps to building table for longest prefix match are as follows:

1. Write the entry values so that they are sorted in order of decreasing prefix length.
2. Write the corresponding mask values.
3. The Global Mask Register (GMR), which has been written with all “1”s for WRITEs and SEARCHes so that the global mask does not modify any data bits on a WRITE or enables a SEARCH for all 32-bit positions.

5 TABLE MAINTENANCE FOR LONGEST PREFIX MATCH

Assume that there are 31 possible prefixes, as shown in Figure 1. The table is then divided into 31 regions that are arranged in order of decreasing prefix length. The size of the regions need not be equal. Depending on knowledge of the application space, a user may expect certain prefix lengths to occur more frequently than others. The user can then allocate a bigger region size for the more frequently occurring prefixes. The system

maintaining the table needs to keep track of 31 pointers, each marking the start of a region. In addition, each region also needs a pointer to the next free location in that region. To add a new entry with a certain prefix length, a user can write to the next free location in the corresponding region.

As entries get added and deleted over time, regions can become fragmented and will have to be defragmented. For example, a “hole” can be created between prefixes based on the frequency of certain prefixes, thereby allowing a prefix to expand. When the user reaches the boundary of the next prefix, entries from the next prefix set can be relocated to the bottom of its own space, thereby creating room for the prefix above to push downwards.

In a dynamic environment it is possible for regions to get filled up. Region sizes can be adjusted by relocating entries and changing the region pointer to reflect the new sizes. By utilizing burst READ and burst WRITE support, a user can achieve very high throughput in moving blocks of data. **Note**. These pointers are all maintained by table management software; not by the LNI7010 device. The on-chip burst counters for READ and WRITE, however, help in data movement.

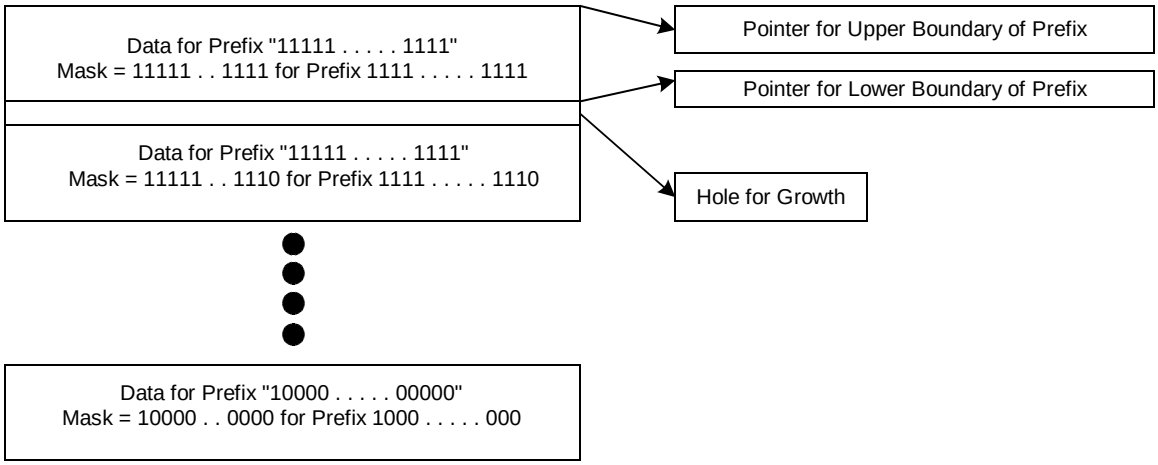

FIGURE 1. ARRANGING THE DATA ARRAY IN A SORTED ORDER OF PREFIXES

6 CONCLUSION

This application note describes the use of the LNI7010 NDSE for longest prefix match applications. Although this document uses a simple IPv4 example to convey the concept, the LNI7010 device can be used on other (similar) applications requiring a form of longest prefix match. The subject of table management was also discussed herein. The LNI7010 is a configurable NDSE. Although it was optimized for 68-bit, 136-bit , and 272-bit operations, it can efficiently handle 32-bit look-ups as well.

CONTACT

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

Tel: 408 942 2000
Fax: 408 942 2099

