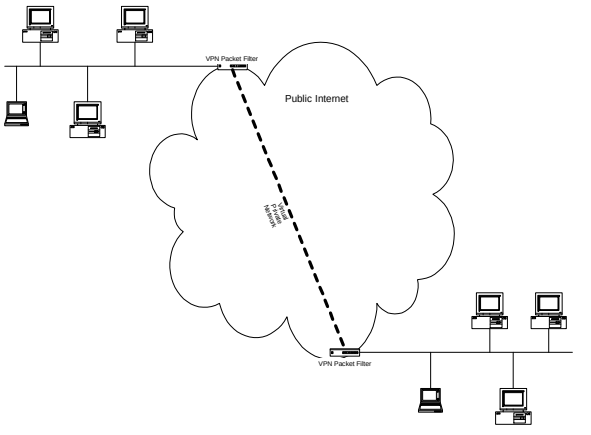
Packet Filtering – Using Network Search Engine Technology in VPN Applications

Packet Filtering

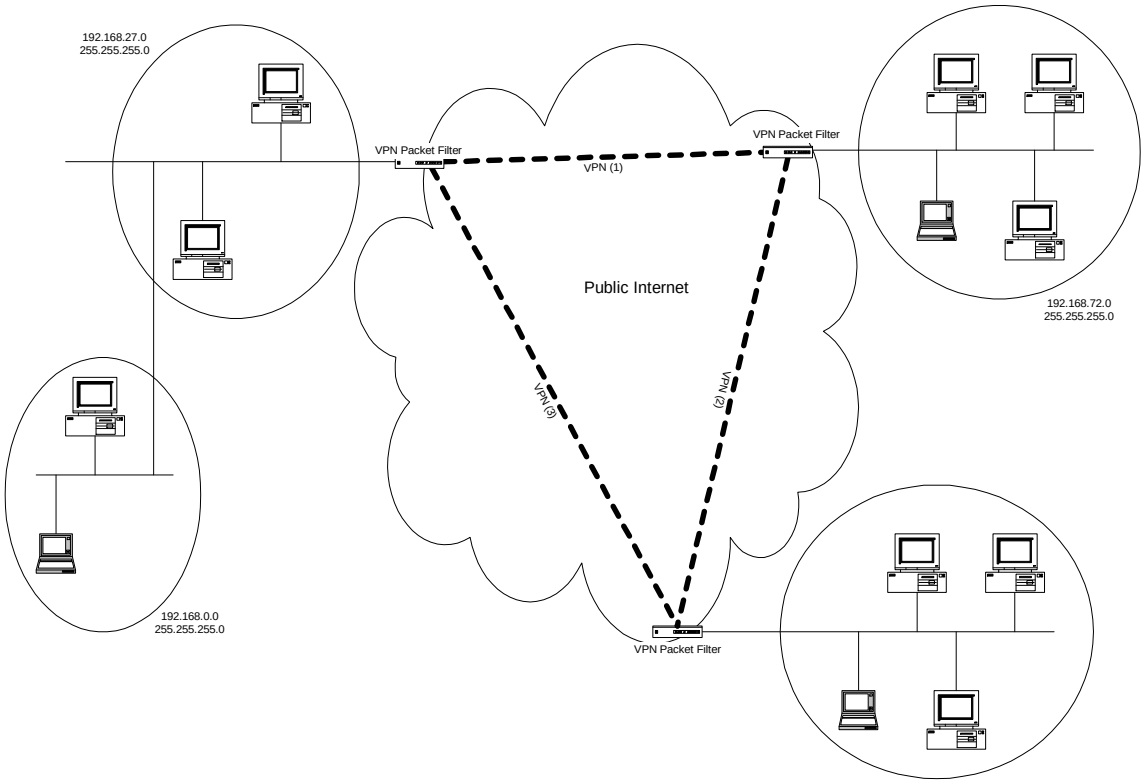
Virtual Private Networks use the Public Internet to connect together two (or more) private IP networks. This has to be achieved securely, without allowing other devices in the Public Internet access to the networks at either end of the VPN. The two technologies that are deployed to achieve this, are encryption and packet filtering. This paper looks at the use of Network Search Engine (NSE) technology to address the packet filtering process. Packet filtering is used to determine which source/destination addresses (or subnets) will be allowed to transmit packets into and out of the private networks. An example of such a VPN is shown in figure 1.

Figure 1



Many organisations will use a collection of VPNs to connect their different sites. Each of the LANs at these sites may contain a number of subnets, thus producing a complex set of filtering rules that need to be applied based on source port and the destination IP address for packets entering each VPN and the destination IP address for packets leaving the VPN. An example of a more complex VPN architecture is shown in figure 2.

Figure 2



The packet filtering process can be CPU intensive and produce a significant processing overhead, particularly in high data rate (Gigabit), multi-port network devices. Network Search Engine (NSE) technology can be deployed to implement a database co-processor that offloads the packet filtering process. This simplifies existing designs and allows network devices to support higher data rates and larger port counts than traditional software solutions.

Network Search Engine (NSE) Technology

NSE technology uses silicon search engines to provide lookup functionality as a co-processor to an existing Microprocessor or Network Processor design. The silicon search engine implements a table (or number of tables) using Associative Memory, and therefore NSE co-processors are able to provide the results of a query to the host processor a few CPU cycles after the request has been made. The LNI7040/LNI8010 co-processor from Lara Networks, Inc. is capable of providing one result every three cycles for a 72-bit wide record, at a clock speed of up to 100MHz. The alternative to using an NSE co-processor is to implement the lookup tables required in software. NSE provides the following benefits over software-based solutions:

- Faster – less CPU cycles until a match is achieved
- Offloads processing – the host CPU is free to perform other tasks
- Deterministic – performance is constant – even as table sizes grow past 1M records!
- Simpler Design – no complex software algorithms to de-bug

Logically, an NSE co-processor can be considered a number of tables of records with the configured width. The LNI7040 NSE device from Lara Networks, Inc. may be configured to create tables that are 34, 68, 136, or 272 bits wide. Each of the tables can be considered to be independent and the format of the record within the table is determined by the controlling software. The record structure described in this paper is only an example for this application and should be modified to suit the precise requirements of the user.

Each record in the table also has an associated mask record, the same width as the record. This can be used to mark any bit of the record as ‘don’t care’ for the purposes of the match process. This allows the tables to be further divided into rules that are generic and those which are more specific.

In an associative memory search it is only possible for one entry in the table to be returned as the match. Due to the mask capabilities it is likely that multiple matches will occur when the search is carried out. This is handled by the built in priority encoder which ensures that the match with the lowest address is returned in the case of multiple matches. If the tables are organised such that the more specific rules occupy the lower addresses in the table, then this process ensures that specific rules override more generic entries. This feature is particularly important when devising IP based routing, filtering or queuing rules as rules may apply to network subnets and then be overridden by rules that apply to a specific host. Having one mask bit available per data bit in the table allows this NSE to support CIDR and other classless routing and filtering processes.

Using NSE in Packet Filtering

Table layout for Multi-layer Packet Filtering

The Public Internet is an IP based network which provides native carriage of IP packets on a best effort basis. When implementing a VPN, additional network protocols are implemented by encapsulating those packets within IP packets for routing through the Internet. Thus, when considering the Packet Filtering task at the entry and exit points of the VPN we only need to consider the IP protocol. Multi-layer filtering is implemented by building the rule table based on the structure shown in Figure 3.

Figure 3

NSE record*						**Related SSRAM record**		
Source IP	Source Port	Dest. IP	Dest. Port	Prot. Type	Spare	Allow flag	Egress Port	Spare
32 bits	16 bits	32 bits	16 bits	4 bits	28 bits	1 bit	16 bits	47 bits

*The NSE is configured to provide a 136bit wide search table.

Layer 3 filtering and routing rules would be implemented by masking the Layer 4 fields in the NSE record. Lara Network's NSE products allow the rules database to be split into segments as required. To provide priority of specific rules over generic rules, the table would be split into segments as follows:

NSE Addr.	Source IP*	Dest. IP*	Prot. Type	Source Port	Dest. Port	Comment	Size
Lowest	Masked	Masked	Masked	Masked	Masked	This is a single default entry (Allow flag set to N).	One entry
		Masked	Masked	Masked	Masked	Layer3 Source rules only. (This segment would also be split into groups of each mask length)[1]	From one to as many hosts as there are in the source subnet.
	Masked		Masked	Masked	Masked	Layer3 Destination rules only. (This segment would also be split into groups of each mask length)[1]	From one to as many hosts as there are in all destination subnets.
			Masked	Masked	Masked	Specific Layer 3 rules. (This segment would also be split into groups of each mask length)[1]	Potentially one entry per valid source / destination pair.
Highest						Specific rules based on all Layer 3 and Layer 4 information.	Many entries for specific filters.

*Source and Destination address columns may be reversed to provide priority to the Source address rules. This table shows priority given to Destination address rules in situations where both a source address only and destination address only rules provide a valid match.

[1] To provide CIDR-like rule priority (or additional CIDR routing functionality), the table will be split into segments which are masked to each of the 31 possible mask lengths wherever they are shown as not masked in the above table. For details on how to implement an NSE that supports CIDR rule functionality please consult our Application Notes.

Table Management

When lookup tables are initially configured, blocks of entries will be reserved for each of the segments in the above table, based on the estimated number of rules in each category. This will allow entries to be added to the table without re-ordering or moving of existing entries being required. One bit of the record would be used to indicate that an entry is 'empty' and will be ignored by lookup operations. If the reserved allocation is filled, then entries need to be moved up or down to provide more space in the required area. This can be achieved within the NSE devices by using the move entry command (supported by LNI8010 co-processor). This feature ensures that lookup table management does not produce a significant overhead.

NSE lookup tables are always ordered such that the implicit priority mechanism of the associative memory returns the most specific entry that matches the search criteria. Hence, there is no need to deal with multiple matches as the result returned by the NSE will be the most appropriate. Generic rules should occupy the highest addresses, growing from the bottom of the table and specific rules should occupy the lowest addresses and grow from the top of the table.

Duplicate (identical) rules are not allowed, so there is no need for entries within each category to be ordered in any particular way as there will never be multiple hits from within a set of records with the same mask.

The global NSE table can be subdivided into smaller units by using the spare bits to identify the sub tables. Using this flexible feature of the NSE, tables can be built to support a number of different applications within the same device. In the case of a router that is performing VPN processing as well as traditional CIDR, the tables required for the VPN packet filtering could co-exist with the routing entries for the network, without any degradation in performance for either process.

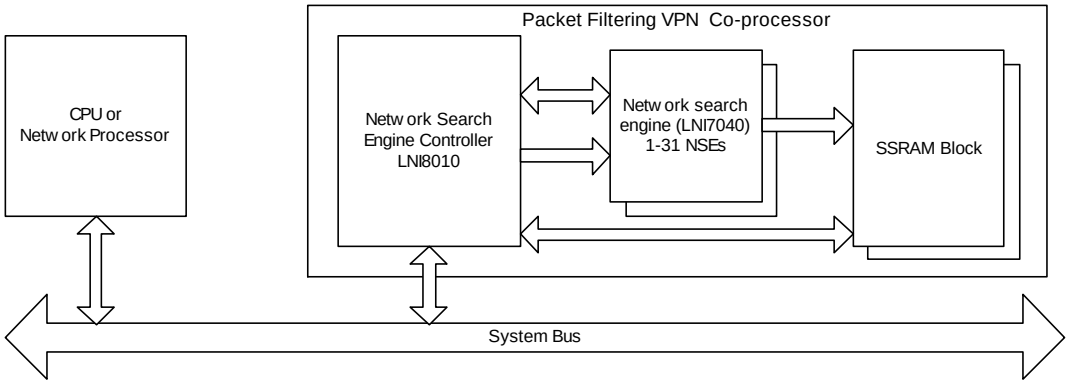
The LNI7040 device from Lara Networks, Inc. may be configured in blocks to create multiple tables of differing record width. Please consult the LNI7040 data sheets for configuration details.

Implementation

Architecture

To implement a 'Packet Filtering VPN Co-Processor', 'Network Search Engine' technology (such as the LNI7040 and LNI8010 products from Lara Networks, Inc.) may be deployed, either in conjunction with a Network Processor, or with a standard RISC microprocessor. A typical system architecture is shown in Figure 4.

Figure 4



Performance

In order to determine the number of concurrent ports that the "Packet Filtering VPN Co-Processor" will support, it is necessary to look at the query throughput that can be achieved by the design. The LNI8010 Network Co-Processor may be connected to a 64 bit, 100MHz system bus. Three bus cycles are required to perform a 68 bit search, and Four cycles are required to perform the 136 bit search.

Figure 5

Operation	68 bit search	136 bit search
Write Search Word	10ns	20ns
Write Command	10ns	10ns
Read Result Word	10ns	10ns
Total response time	**30ns**	**40ns**

The filter needs to be applied for each packet that is transmitted through the device and so the time available to evaluate the parameters is dependant on the size of the packet. Average packet sizes are determined by the mix of applications that are supported by the network. The table shown in Figure 6 details packet sizes for a number of applications and therefore the number of ports that the "Packet Filtering VPN Co-Processor" could support at each data rate.

Figure 6

Average Packet size	Layer 3 only Filtering			Multi-Layer Filtering (3&4)		
	# 1Gbps ports	# 2.4 Gbps (OC-48) ports	# 9.6 Gbps (OC-192) ports	# 1Gbps ports	# 2.4 Gbps (OC-48) ports	# 9.6 Gbps (OC-192) ports
64 bytes (i.e. VoIP)	16	8	1	12	6	1
128 bytes	32	16	3	24	12	2
512 bytes (i.e. www)	128	64	13	96	48	10
1K bytes (i.e. FTP)	256	128	27	192	96	21

Conclusion

Packet Filtering is a key component in the implementation of VPN services which utilise the Public Internet. As data rates and access reliability of the Public Internet improve, many more corporations will realise the cost benefits of using secure VPNs to connect their regional and international offices in preference to more expensive dedicated networks. As port data rates increase to Gigabit Ethernet and up to OC-192, conventional software techniques will not be able to sustain wire speed transport, particular for small packet technologies such as VoIP. The addition of a "Packet Filtering Co-Processor" can provide a mechanism to offload the required processing, thus making the central CPU or Network Processor available to control the overall process, and perform other processing such as the encryption/decryption of data. The "Packet Filtering Co-Processor" also provides significant performance enhancement thus allowing devices to support either higher data rates up to OC-768, or a greater number of ports at lower data rates. The flexible architecture provided by "Network Search Engine" Technology allows a common design to be deployed to implement CIDR, Packet Filtering, QoS, and MPLS functionality within a networking device.