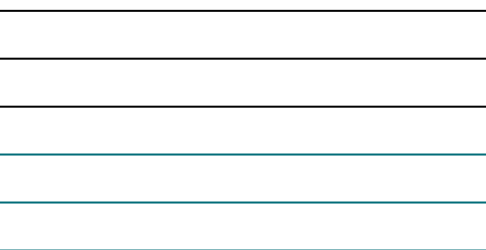




Network Content Processing

Addressing the growing processing gap in data networking equipment



Abstract

Seaway's Network Content Processor (NCP) is a high-performance processor optimized for layer 4 protocol termination and layer 5 to 7 content processing. It brings to market a new level of integration, incorporating TCP termination, stream management, and content processing features. Instead of examining data packet-by-packet, it converts packet-based traffic into contiguous streams of data and processes the resulting streams. The NCP enables system vendors to achieve unprecedented performance and ultimate flexibility while reducing cost and accelerating time-to-market.

Document SDN 00024
Version 1.0
March 2003

Seaway Networks Inc. assumes no responsibility whatsoever for the uses made of this material or for decisions based on its use and supplies this material "AS IS" and without any warranties, either expressed or implied, regarding the contents of this material, its completeness, accuracy, merchantability, non-infringement or fitness for any particular purpose. Seaway Networks Inc. may make improvements and/or changes in the products and/or options described in this document at any time and without notice.

Seaway Networks is a trademark of Seaway Networks Inc.
Streamwise is a trademark of Seaway Networks Inc.
Motorola is a registered trademark of Motorola, Inc.
All other product and brand names are the property of their respective owners.

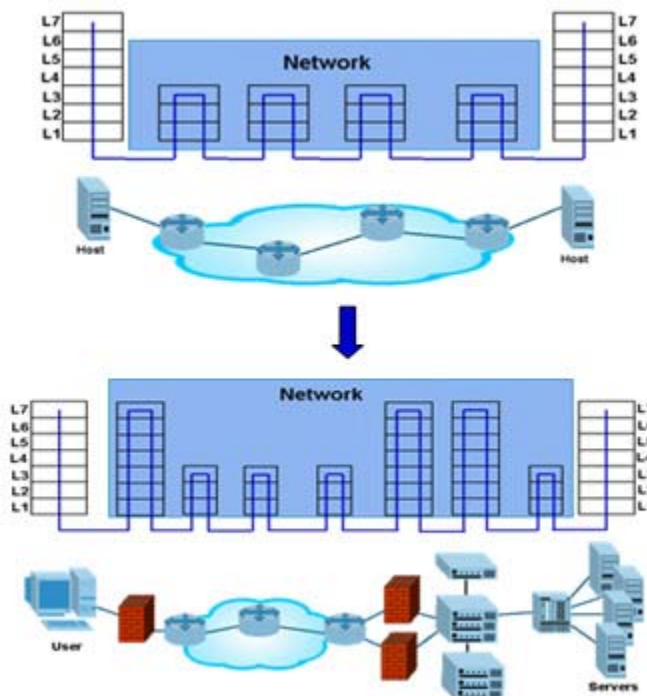
Copyright © 2003 Seaway Networks Inc. All rights reserved. Information subject to change without notice.

1. Layer 4-7 Landscape

Traditionally, network equipment has operated at layer 3 in the protocol stack and below. Today, the need for improved security, quality of service and traffic control is driving edge and data center equipment to operate at layers 4 and above. Unfortunately, the need for network-based layer 4-7 equipment has created a significant set of design challenges for networking equipment vendors.

- The equipment must be able to terminate layer 4 protocols such as TCP.
- The equipment must provide a high level of content processing to enable advanced layer 5-7 services. Content processing is complex, time-consuming and must often be tailored for a specific application.
- Due to the location in the network, this equipment must support a large number of simultaneous connections.
- The data center and network edge are points of traffic aggregation. Equipment in these locations must support multi-gigabit line rates to ensure network performance is not compromised.

The combination of fast line rates, a high number of connections and complex layer 4-7 processing has created a unique processing challenge for system designers. As a result, most layer 4-7 products are point-solutions which only provide an individual service. This piecemeal approach has created a heterogeneous environment composed of a high number of separate systems and a high number of potential points of failure. This reduces network reliability and increases total cost of ownership for the network operator. For example, according to a recent study by Gartner Inc., an integrated network security platform approach will increase network security and reduce the cost of ownership for perimeter security, while preserving best-of-breed options¹.



¹ [Network Security Platforms Will Transform Security Markets](#), Research Note, November 7 2002

2. Traditional Processing Solutions

System designers typically rely on general purpose processors, network processors or ASICs to provide processing functionality.

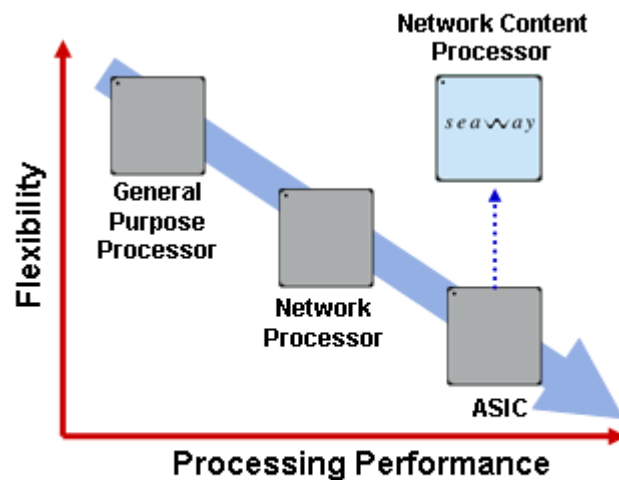
It is generally accepted that general purpose processors cannot scale to provide layer 4-7 processing at wire-rates. According to Linley Gwennap of the Linley Group, "A 400-MHz MIPS CPU would consume all of its cycles just trying to terminate [the TCP traffic of] a simple Fast Ethernet channel"².

At gigabit rates, there is no viable solution today. Waiting for Moore's Law to catch up and provide a sufficiently-powerful CPU is not the answer. General purpose processors expend a significant portion of their cycles implementing lower protocol layers. They are often stalled waiting for external data and are limited by I/O and memory bus bandwidths imposed by the systems. The conundrum is that a software solution on general-purpose processors is the most practical way by which layer 4-7 applications can be developed. Many functions require application-layer processing on the content of the packets, and the very nature of application-layer software development is characterized by relatively large code size with a strong need for software flexibility.

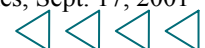
Traditional Network Processors are designed specifically for the layer 2-3 forwarding plane and offer acceleration for functions such as classification, lookups, checksums, queuing, and header manipulation. But, since NPUs are optimized for layer 3 and below, most do not have features for layer 4 protocol termination or layer 5-7 content processing. Architecturally, network processors typically use multiple processing engines to meet the processing requirements of the application. In such instances, complex software must be written using a particular instruction set and be designed to work with the processor's specific pipeline architecture. Their applicability is also handicapped by their small instruction code space, which is generally too limited to implement complex layer 7 protocols and applications. While it is sometimes possible to pipeline a number of processor cores together to overcome these limitations, finding the proper balance and work breakdown for each stage, is a painstaking, iterative endeavor.

ASIC's are the most specialized solution for layer 4-7 processing, but they are also the most inflexible. ASIC design times are long, expensive, and any updates or changes must be implemented through an expensive re-spin. Flexibility is a key requirement for a successful layer 4-7 processor. A layer 4 engine must be able to support emerging layer 4 protocols and support protocol updates. A layer 5-7 engine must be able to support the wide range of processing requirements demanded by content applications as well as support new features as new content applications are added.

A Network Content Processor (NCP) is a new category of processor which takes a unique approach by combining the processing speed of ASIC technology with the flexibility of an industry-standard instruction set and development tools. This is achieved by embedding tedious and processing-intensive functions in specialized silicon while keeping protocol control and error handling in



Processing performance of an ASIC with the flexibility of a General Purpose Processor



firmware. This architecture offloads CPU-intensive work from the general purpose and application processors leaving them free to perform high-value tasks.

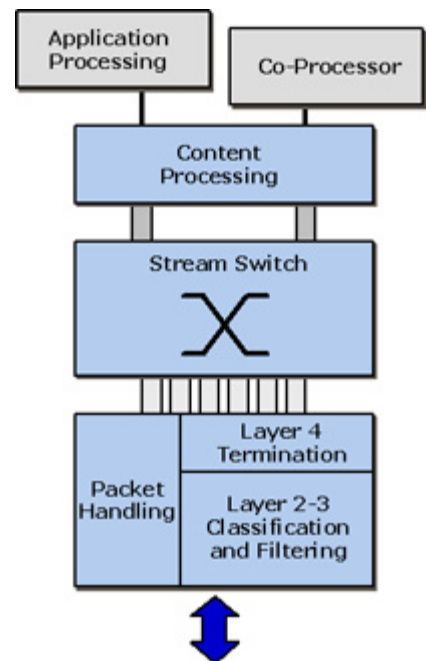
3. The Network Content Processor Concept

The Network Content Processor concept arises from the fact that layer 4-7 networking equipment requires a tightly coupled combination of application processing and networking stack processing. Any solution which addresses these requirements individually will have significant inefficiencies. In addition, because application data has no correlation to packet boundaries, the performance of a processing solution ultimately depends on the efficiency of the transition between the packet processing and application processing domains.

A Network Content Processor (NCP) should provide an effective and efficient approach for layer 4-7 processing. An NCP should offer system designers a blend of packet handling, layer 4 termination and content processing features. Instead of examining data packet-by-packet, an NCP should also convert packet-based traffic into contiguous data streams and process the resulting streams. This processing architecture will improve processing efficiency and maximize performance. Recognizing that functionality such as high throughput cryptography is sufficiently complex that it requires its own device, an NCP should support the interfaces needed to connect state-of-the-art third party co-processors.

A NCP should provide the following processing features:

- Layer 2-4 processing including packet filtering, classification and full layer 4 termination
- Layer 5-7 processing including hardware-based content examination, modification, and replication
- High speed interfaces to state-of-the-art co-processors



The Benefit of Stream Switching

Frequently, the bottleneck in traditional systems is caused by memory bus and I/O bus bandwidth constraints. In relatively simpler systems where a limited amount of processing is required on each packet, a single high-speed bus interconnecting processing elements and memory may be sufficient. As the processing and memory requirements increase, however, such as when multiple processing elements need to act on a single packet, a switched architecture reduces bus contention and improves efficiency.

The key for handling multiple functions is the ability to switch data amongst the different processing elements. Sometimes a packet requires IPSec processing, another packet may require TCP termination, yet others may require anti-virus scanning, and others still may require all these functions. Because each traffic flow may be subject to different policies, the behaviour of a packet as it enters the system cannot be hard-coded in the design of the appliance, nor can one function be optimized at the expense of others. It is important for the system architecture to have the flexibility to handle different situations, where a single packet may need to be processed once, or numerous times by different elements, as dictated by packet content and/or policy.

A stream switch provides point-to-point connections to each processing element. This allows each individual stream to be managed separately with data scheduling and presentation mechanisms to



improve processor utilization and minimize processor interruptions. The stream switch brings a layered approach to processing by enabling the data to be switched multiple times through multiple levels of processing. This enables the NCP to make use of cut-through techniques, which ensures that processors will only be in the data path when required.

A Network Content Processor should:

- Provide integrated stream switching
- Manage each stream individually

Application Environment

Since an NCP is expected to support complex layer 4-7 applications, its application programming environment should be simple to use and provide as much support for software portability as possible. These requirements suggest that an industry standard instruction set and standard development tools would be the best method to keep the application environment open and flexible.

A network content processor should provide:

- Programmable processors which enable the NCP to support new protocols and new services.
- An industry standard programming model and instruction
- An application development model that enables software portability

4. Market

The Network Content Processor was designed for data communication and telecommunication equipment manufacturers to accelerate and enhance the level of layer 4-7 functionality in their products. The Network Content Processor can fit into appliances, cards for slot-based chassis, server blades, and PCI cards. Seaway has identified Network Security, Web Infrastructure, and IP Edge Services as the principal application areas that can benefit from the technology.

Network Security Applications	Web Infrastructure Applications	IP Edge Service Applications
<ul style="list-style-type: none"> ▪ Security gateways ▪ Firewalls ▪ IPSEC VPN gateways ▪ Intrusion detection ▪ Virus filtering systems ▪ Content filtering systems ▪ SSL VPN gateways ▪ Lawful intercept 	<ul style="list-style-type: none"> ▪ Content switches ▪ Web caches ▪ Multi-media delivery platforms ▪ SSL accelerators ▪ XML Routers 	<ul style="list-style-type: none"> ▪ Wireless IP services switches/gateways ▪ Broadband IP services switches



5. Seaway's Network Content Processor

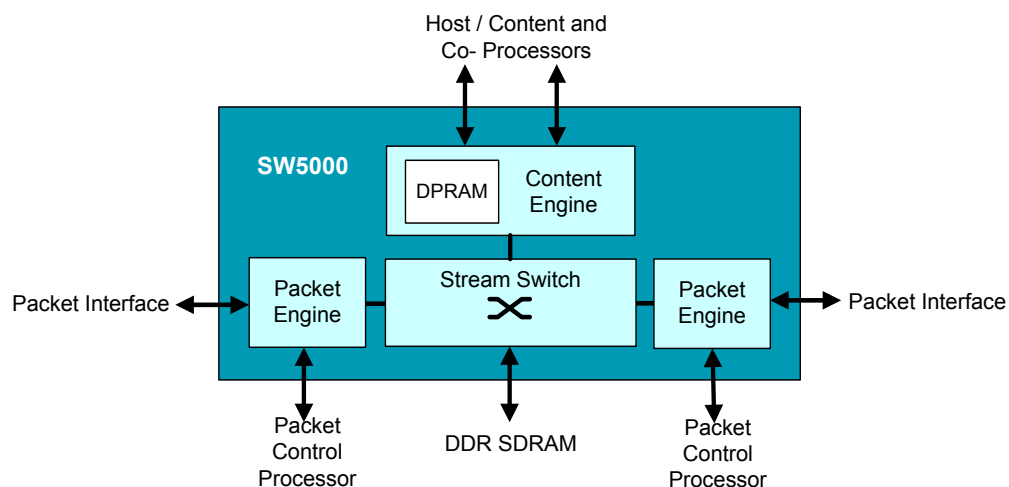
The Network Content Processor (NCP) from Seaway Networks is a high-performance processor optimized for layer 4 protocol handling and layer 5 to 7 content processing. Silicon-based features combined with a highly efficient architecture provide significant performance advantages over other solutions.

- 20 times improvement in system performance (over current NPU and ASIC solutions)
- 5 Gbps throughput (full duplex)
- 2 million simultaneous layer 4 sessions
- 300,000 TCP connection setups per second
- 64k virtual domains (e.g. IP address or security domains)
- 150,000 layer 7 switching decisions per second
- 66 million ACL comparisons per second
- 17 Gbps layer 5-7 pattern matching search

5.1 Major Functional Blocks

The three main functional blocks of the NCP are the Packet Engine, Stream Switch and Content Engine.

Figure 1: Seaway's Network Content Processor Architecture



Packet Engine

Key Functions

- Packet filtering
- Fragment reassembly
- Flow identification
- Header and checksum validation
- Header parsing
- Header formatting/creation
- Segmentation/fragmentation
- Unacknowledged data management

Each packet interface on the NCP has its own Packet Engine. The Packet Engine performs layer 2 to 4 processing at wire speed. The Packet Engine autonomously processes incoming and outgoing packets as well as provides hardware-based assists that are used by the Packet Control Processors to accelerate layer 2 to 4 processing such as flow classification, flow table lookups, checksum calculation and other packet manipulation work. The Packet Engine is also the interface between the packet domain that exists outside the NCP and the stream switching domain that exists within the NCP. (A stream is a unidirectional flow of data identified by IP addresses and possibly layer 4 addressing information.)

For flexibility, programmable processors are used to orchestrate the hardware functions of the Packet Engine. Each Packet Engine has its own dedicated packet processor and dedicated packet interface. Each of the packet interfaces support an input/output throughput of 2.5 Gbps, resulting in a total system input/output of 5.0 Gbps full duplex.

Stream Switch

Key Functions

- Policy-based scheduling
- Queuing
- Switching

The patented Streamwise™ Stream Switch is the central component of the NCP. It controls memory access, manages streams, and provides interconnection among the major functional blocks and external interfaces. It can switch and manage up to 4 million data streams, enabling efficient pipeline operation between the various system components. This ability makes the NCP ideal for a multi-function system where data need to be moved across different processing elements for processing.

Content Engine

Key Functions

- Content Searching
- Stream Modification
- Stream Replication

The Content Engine provides hardware assists for high speed pattern matching, content modification, and content replication. These assists are used by an external Content Control Processor (CCP) to accelerate application processing. Through these assists, the CCP can dedicate its cycles to application processing, rather than the processing-intensive tasks of searching and moving data. The Content Control Processor can be any processor that supports a PBSRAM interface.

5.2 Content Processing Features

Searching through the content byte stream and moving data efficiently between and within the different processing elements can constitute a significant amount of processing in advanced systems. In most systems, data must be copied into application memory after the TCP byte stream has been recovered. Then, as the processor tediously searches through the content for specific byte patterns (which itself requires considerable movement of data between processor and local memory), new packets are constantly arriving and crossing that same processor-memory interface.

The NCP has been designed specifically to address the above common issues of content processing.

- Packet buffer memory is separate and distinct from the content processing memory. Packets are buffered in the local DDR SDRAM, while content processing operates on data within the internal dual-ported SRAM (DPRAM).
- The Content Control Processor operates on the data within the DPRAM directly, and does not need to copy the data to the processor's local memory. No processor copying is required from the time the Ethernet frame enters the system, through IP and TCP processing, through content processing including scanning the content, and finally out of the system as an Ethernet frame.
- The data within the DPRAM is stored in contiguous memory, making it easier for processing.
- The Content Control Processor can invoke a number of hardware assists to search, modify, or replicate the content within the DPRAM.

Unlike packet processing which is well understood and somewhat bounded, content processing can be quite open-ended. Unforeseen situations will occur where an increasing amount of processing is required at the content layer. An NCP-enabled system can scale in multiple ways:

- Since the NCP leverages external general-purpose processors, performance can be increased simply by replacing these processors with faster ones. An NCP system can effectively take advantage of the general-purpose processing performance curve.
- Performance can also be increased by attaching multiple content processors to the host/co-processor interface. The embedded switch core within the NCP can be used to pipeline content to and across these processors.
- In the event that the internal content searching hardware assist is insufficient, one or more pattern-matching co-processors can be attached across the same host/co-processor interface. Again, the embedded switch core within the NCP can be used to pipeline content to and across the attached devices.

5.3 Co-Processing Features

Utilizing a dedicated high speed interface and advanced stream scheduling features, the NCP-based solution provides an optimized co-processing environment that combines on-chip and off-chip processing. Devices that may make use of this interface include host processors, security processors and classification engines.

The NCP architecture provides an efficient separation of processing functions where initial processing and data formatting are performed on-chip and subsequent specialized processing is off-loaded to coprocessors. This model is flexible and expandable because the coprocessors can be upgraded as new best-in-class devices are released.

The high-speed host/coprocessor interface supports two modes of operation: direct memory access (DMA) mode and streaming mode. In DMA mode, the NCP supports HyperTransport or PCI-X access into the memory space of one or more connected devices. To facilitate interaction between the NCP and processing devices, the hardware-based assists can be invoked by the Host Content



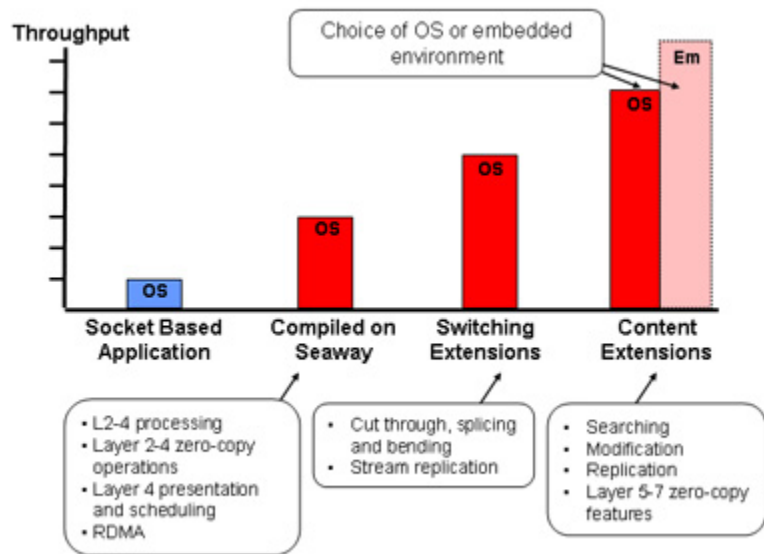
Processor or the Content Control Processor to move data from streams into memory (outbound DMA) or from memory onto streams (inbound DMA). In streaming mode, the coprocessor interface supports a variety of sub-modes, including POS-PHY Level 3, to connect to security coprocessors, including Corrent, Broadcom, Hifn, or any other POS-PHY Level 3-compliant device.

5.4 Application Development

When a new processor is incorporated into a design, application designers are often concerned that they will not be able to reuse existing code and will have to learn a new development environment and tool chain. Unlike network processors and other Application-Specific Standard Products (ASSPs), application development on a NCP-based system is accomplished through an industry-standard Instruction Set Architecture (ISA) and tool chain. Designers can program in C/C++ using standard tools and compilers in a familiar development environment. Seaway’s application framework provides a run-time environment and an advanced API library for the development of high performance layer 4 to 7 applications. The APIs can be used directly to develop embedded real-time applications or can be used to integrate the application into a conventional operating system.

Depending on the performance required, the application designer has a range of NCP integration options.

The simplest level of software integration is a recompile of existing socket-based applications, to make use of the NCP as a “fast TCP stack”. In this option, the NCP provides layer 2 to 4 processing including TCP offload. The application is further accelerated by layer 2-4 zero copy operations and layer 4 presentation and scheduling features.



The next level of integration occurs when the application designer chooses to use the NCP Switching Extension APIs. These features are very useful for content switching applications. These extensions call the NCP to perform hardware-based stream splicing, bending, replication, cut-through and Remote Direct Memory Access (RDMA).

For applications requiring the highest levels of content processing, the application designer can use the NCP’s Content Extension APIs. These features work in conjunction with the Content Control Processor (CCP) to perform hardware-based content searching, modification and replication. It also makes use of layer 5-7 zero-copy features to maximize application processing efficiency.

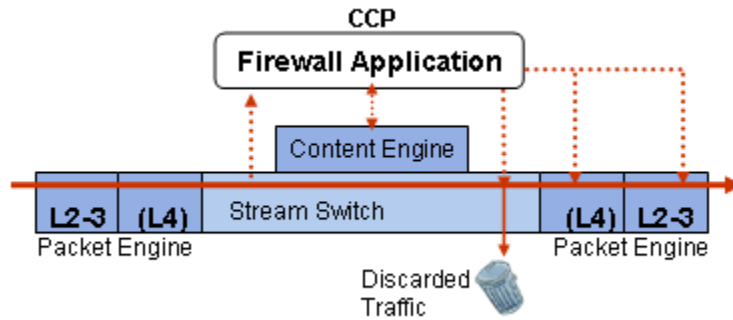
The NCP system provides deterministic TCP performance. This combined with NCP provided test cases and reference applications will simplify evaluation, reduce design time, and provide faster time to performance resulting in a 50% reduction in design effort. As a result, a vendor using the NCP can develop next-generation services and bring them to market faster and with lower development cost than with other processing technologies.

Application Examples

The following three examples demonstrate how the NCP is used to accelerate layer 4-7 applications. The diagrams provide a conceptual view of the data path and major functional blocks involved.

Firewall

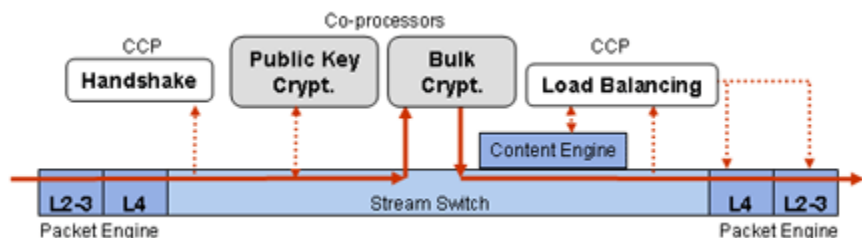
The firewall function can be divided into two phases: policy enforcement and flow forwarding. Policy enforcement requires the packet headers and content to be examined in sufficient detail in order to determine whether the flow is allowed or disallowed. This can be processing-intensive, and usually takes place in the first few packets of the flow. However, once it is determined that the flow is allowed, flow forwarding can take place for all subsequent packets of the same flow. Flow forwarding requires a simple packet-level lookup in the flow table to determine what action to take (accept, deny, NAT) and how to switch the packet. By decoupling the enforcement from the forwarding, only a small subset of the data would require intensive content-level classification while the rest can simply be switched.



As data enters the system, the packet engine performs ACL packet filtering, classifies the incoming flow and, if required, terminates the layer 4 protocol. Layer 4 termination is required if the firewall is providing full proxy capability. The stream switch presents a copy of the stream data to the CCP where it will determine how to respond to the new flow. The firewall application now has the option to search through the content using the content engine and, if required, modify and replicate the stream. If the data does not meet the requirements of the policy for the stream, the firewall application can instruct the Stream Switch to drop the data. If the stream is fine, the firewall application can instruct the Packet Engine to initiate an appropriate Layer 4 session (for full proxy) or send the data out with an appropriate layer 2-3 header (for NAT). All subsequent traffic for that stream can be processed in cut-through mode, without requiring CCP intervention.

SSL Accelerator

SSL acceleration requires processing at all layers of the stack, including TCP termination, SSL record accumulation, public key and bulk cryptography, and content switching or load balancing. Efficient processing requires handoff of data between layers, especially in systems where different processors are involved. For example, if the SSL record spans multiple TCP segments, all the segments must be collected before the record is passed on for SSL processing.



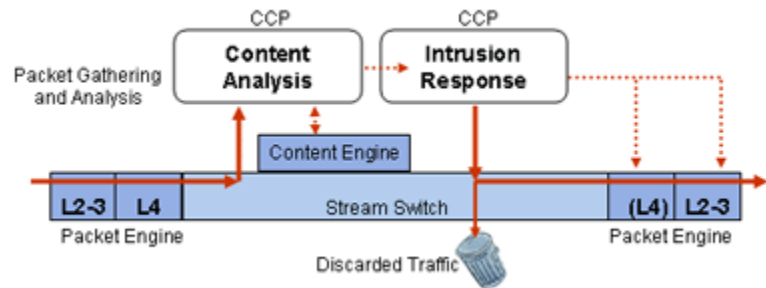
In the following example, the NCP is combined with security processors to perform public key and bulk cryptography. For clarity, two separate devices are shown but it could be done with a single dual function security co-processor. Data traffic enters the Packet engine and the TCP session is terminated and the TCP segments comprising the SSL record are accumulated. A copy of the SSL record is presented to the CCP which controls the SSL handshake and invokes the public key co-



processor to perform the cryptographic function required in the handshake. Once the handshake is complete, all subsequent traffic from the session is passed through the bulk cryptographic processor. Before the decrypted session is sent to the server, the CCP can search the HTTP header using the content engine to perform cookie-based session persistence or some other intelligent load balancing algorithm. As the session leaves the NCP, an outgoing TCP session is initiated and data is formatted appropriately into packets.

Inline Intrusion Detection and Prevention System

Intrusion detection signatures consist of a packet header portion and a content portion. In order to match the content portion of the signature, the full layer 4 byte stream should first be recovered before any searches are performed. This ensures that signatures crossing packet boundaries can be detected normally, without requiring special treatment. Solutions that attempt to match signatures without first recovering the byte stream must treat cross-boundary signatures as special cases requiring additional software processing, often with unpredictable performance.



As data enters the system, the packet engine checks for packet-level intrusions or attacks and matches packet-level signatures. Alerts can be generated and attacks stopped at this point if configured for intrusion prevention. The recovered byte stream is passed to the Content Control Processor to be checked for application layer protocol anomalies. This check makes use of the hardware pattern-matching feature of the content engine. By operating on byte streams rather than packets, an attack signature that crosses packet boundaries can be detected just as easily as if the signature is contained within a single packet. If suspicious data is detected, the data in question is logged using the content replication feature in the content engine and an alert is generated to the network management station. If the received traffic does not contain any harmful content, the traffic is switched to the Packet Engine where data is formatted appropriately into packets and sent out of the system.

6. Conclusion

The NCP-based solution takes advantage of the flexibility and familiarity of general-purpose processors and combines it with the performance of specialized silicon. The NCP feature set and unique architecture provides a number of key differentiators.

- Multiple offload engines are packaged together. The NCP provides TCP offload, content searching, stream switching, as well as a number of assists at the packet layer and content layer.
- Multiple layers of concurrent processing. At any given time, the packet processor can be processing data and looking up tables, the content processor can be processing content and looking up tables, and packets can be arriving from the link and processed by the NCP and stored in packet buffers, all using different memory buses.
- Memory I/O is distributed. Packet buffer memory is separate and distinct from content processing memory. Packets are buffered in the local DDR SDRAM, while content processing operates on data within the NCP internal DPRAM. More specifically, packet buffers, packet and

flow classification tables, ACLs and other hardware tables, are stored in the DDR memory. Software packet processing tables and context are stored in the packet processor's SDRAM. Current content is stored in the DPRAM. Software content processing tables and context are stored in the content processor's SDRAM.

- No processor copying is required from the time the Ethernet frame enters the system, through IP and TCP processing, through content processing including scanning the content, and finally out of the system as an Ethernet frame. The packet processor operates on data directly from NCP registers while the content processor operates on data within the internal dual-ported RAM directly, with neither processor needing to move data across to its front-side bus.
- Software control permits addition of new layer 4 protocols and other updates without impacting hardware.
- The NCP solution can scale with the processor performance curve. Performance can be upgraded simply by upgrading the processors or co-processors.
- No learning of proprietary systems is required. Software development on the NCP uses an industry-standard ISA and tool chain.

7. Contact Seaway Networks

For more information on how a Network Content Processor can meet your system needs, please contact Seaway Networks at:

Seaway Networks Inc.
One Chrysalis Way
Ottawa, Ontario
Canada K2G 6P9

E-mail: info@seawaynetworks.com
Web: www.seawaynetworks.com
Phone: 613.723.9161
Fax: 613.723.8244

