Freescale's Network Content Processing Technology

Freescale Semiconductor

Document Number: CNTNTPROCESSWP
Rev. 0
10/2005

OVERVIEW

High-speed content processing is becoming a requirement for a growing number of network elements. How well and how fast network content can be processed is largely a function of the processing devices powering the network element.

This white paper:

> Defines content processing

> Explains why it is important

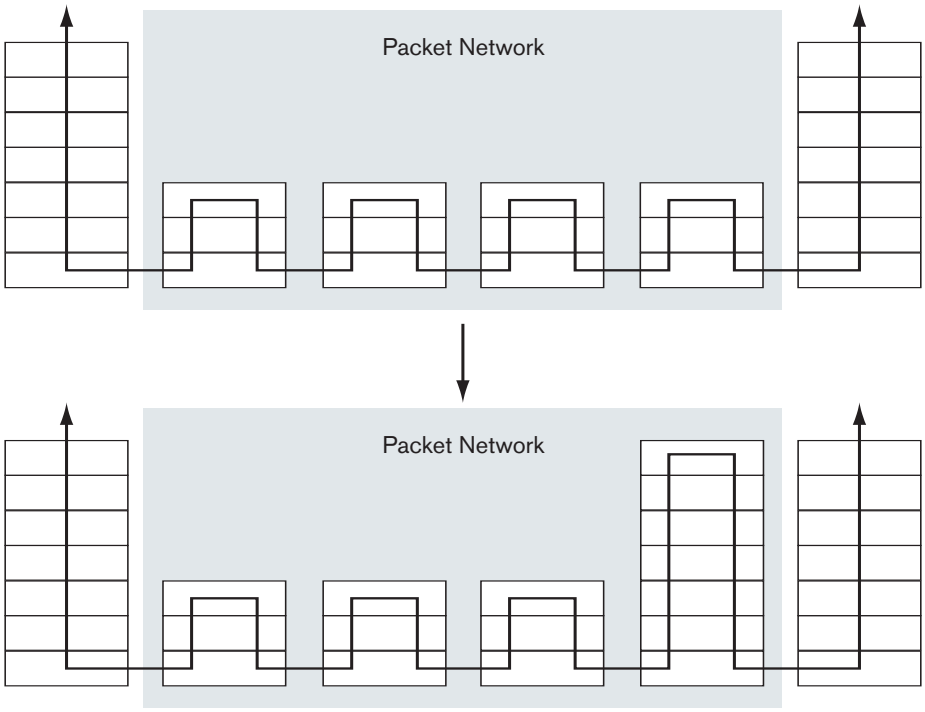
> Describes typical requirements and design challenges

This information is followed by brief descriptions of some of the network content processing Intellectual Property (IP) blocks that Freescale Semiconductor obtained in the August 2005 acquisition of Seaway Networks. These IP blocks enhance Freescale's ability to create advanced network content processing solutions and to strengthen its leadership position in communications processing and connectivity solutions for the networked communications and pervasive computing markets.

CONTENTS

1. What Is Network Content Processing? 3
2. Why Network Content Processing? 3
   - 2.1. Content-Based Network Security and Monitoring 3
   - 2.2. Content-Based Traffic Control 4
   - 2.3. IP Service Applications and Billing 4
   - 2.4. Network Elements with Content Processing 4
3. Content Processing Requirements
   - 3.1. Network Intrusion Detection/Prevention Processing 5
   - 3.2. Processing in Anti-Virus, Anti-Spam Filtering Systems 5
   - 3.3. Typical Functional Requirements 6
4. Design Challenge 6
5. Freescale's Network Content Processing Approach 7
   - 5.1. Traditional Processing Solutions 7
   - 5.2. Freescale Approach/Design Principles 7
   - 5.3. General-Purpose Processor Orchestrating Hardware Assists 8
   - 5.4. Distinctions Between Packet Processing and Content Processing Domains 8
   - 5.5. Dynamic Pipeline 8
6. Freescale Content Processing IP 9
   - 6.1. Packet Engine 9
   - 6.2. Content Engine 9
   - 6.3. Pattern Matching Engine 10
   - 6.4. Stream Switch 10
7. Conclusion 10

1. What Is Network Content Processing?



Traditionally, network elements operate at Layer 3 and below, as depicted in the classic OSI Model shown in the top section of the diagram above. The role of a network element is to forward packets based on a network layer address in the case of a router or a datalink layer address used in a LAN switch. Processing the content carried in the packets is strictly the domain of end systems, i.e., the PCs and servers attached to the network.

However, in recent years, more and more network elements are inspecting the content of packets for security, traffic control and other reasons. The content may further be altered or dropped.

We use the term “network content processing” to describe the processing performed by network elements on the content portion (Layers 4 to 7) of network traffic.

2. Why Network Content Processing?

There are various reasons why a network element processes content. Key reasons include:

> Network security and monitoring

> Traffic control

> IP-based service applications, such as billing

2.1. CONTENT-BASED NETWORK SECURITY AND MONITORING

The traditional “stateful inspection” firewall is no longer adequate to ensure security. These firewalls typically use Access Control Lists to filter network traffic based on Layer 4 and 3 packet header fields. But these days, most successful attacks exploit application layer vulnerabilities. To stop these attacks, the basic “stateful inspection” firewall has to be enhanced to look into the application layer, i.e., into the content of network traffic.

IT managers need both network-based and desktop-based solutions to stop viruses, SPAM and other undesirable content from entering and leaving the enterprise. When a new virus strikes, a signature update must be applied. When network elements provide content processing capabilities, IT managers can quickly add the new signature to a smaller number of critical network elements to prevent the spread of the virus. This provides the time necessary to update all the servers, desktops and mobile computing devices, a process that can take days or weeks. It is more effective to contain the impact of a new attack using a network-based solution, and stopping undesirable elements at the door is vastly preferable to remediation afterward.

To filter viruses, SPAM and undesirable content, a network element has to look into the content of network traffic.

2.2. CONTENT-BASED TRAFFIC CONTROL

In most data centers, a server load balancer intercepts client requests and determines which server is best able to respond to the request. URLs and the corresponding IP address(es) are often not sufficient to determine which server is the best source for the requested content. Content processing allows a server load balancer to peek into the HTTP request message (the content) to determine which particular server in the server farm is to be used to serve the request.

New equipment classes, such as XML routers or switches and peer-to-peer (P2P) traffic managers, make forwarding decision based on application layer headers and payload as well. P2P traffic managers in particular are designed to help service providers contain costs by controlling bandwidth use. To do this, they look into content to identify, control and route P2P traffic.

2.3. IP SERVICE APPLICATIONS AND BILLING

Compared with billing for a connection-oriented network, billing for a connectionless IP network is more difficult. The number of service provider boundaries crossed far exceeds the connection-oriented provider model of the past. Providing services to users and other service providers is a growth area for service providers, and the requirement to bill based on both bandwidth consumed and applications used means that a network element has to look beyond IP addresses into the content of network traffic.

The enterprise environment is not alone in its vulnerability to viruses and other attacks. User and network equipment in IP-based telecom networks, such as VoIP and wireless networks, face similar problems. As our dependence on IP network elements grows, the requirements for content processing to maintain network security, reliability and regulatory compliance (lawful intercept) also grows.

2.4. NETWORK ELEMENTS WITH CONTENT PROCESSING

The following table shows more examples of network elements that process content.

Network Security and Monitoring	Traffic Control	IP Service Applications and Billing
Universal threat management system	Content switches	Wireless IP services switches/gateways
Firewalls	Web caches	Broadband IP services switches
IPsec VPN gateways	Multimedia delivery platforms	VoIP security
SSL VPN gateways	SSL accelerators	Billing
Intrusion detection/prevention	XML routers/switches	
Anti-virus, anti-spam, content filtering		
Lawful intercept		
Network forensics/surveillance		

3. Content Processing Requirements

The following two real-life examples illustrate typical processing in network elements requiring significant content processing:

> Network Intrusion Detection/Prevention System (NIDS/NIPS)

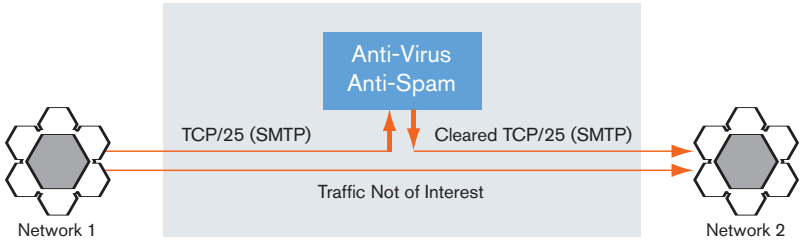
> Anti-Virus, Anti-Spam Filtering Systems

3.1. NETWORK INTRUSION DETECTION/PREVENTION PROCESSING

The following table shows the key processing performed in a NIDS/NIPS:

Detection: Packet Level	Receive packets (fragment) and classify packets into flows Reassemble IP fragments carefully to detect invasions and attacks Reassemble TCP segments carefully to detect invasions and attacks Detect packet-level protocol anomalies Match Layer 3 and 4 header portion of attack signature
Detection: Application Level	Detect application-level protocol and other anomalies Normalize and match relevant portion of the recovered stream against content signatures in a stateful or context-sensitive manner, within and across packet boundary
Response	Log suspicious packet flow/content Generate alerts, including alarm correlation and filtering Send alerts to management station (For inline operation only) Drop harmful content. Packetize and forward harmless content
System	Keep track of many data streams Copy data between input, processing and output components

3.2. PROCESSING IN ANTI-VIRUS, ANTI-SPAM FILTERING SYSTEMS



The objective of the filtering system shown in the above diagram is to transparently cleanse the traffic of virus and spam before delivery to the subscriber.

First, the system allows the specification of “interesting traffic” based on packet header fields, such as source address, destination address, protocol, source port and destination port. Non-interesting traffic, such as traffic belonging to a non-subscriber, is simply switched through the system as is. With interesting traffic, in this case e-mail traffic (TCP port25), the system behaves like a transparent TCP proxy.

Next, the system observes the SMTP protocol exchange in the recovered stream content of the TCP connection. e-Mails are captured on the fly. They are MIME-decoded, separated into component parts, e.g., attachments, and compared against thousands of virus and spam signatures.

If the e-mail contains undesirable content, it is modified or deleted. Safe e-mail is forwarded to its destination.

3.3. TYPICAL FUNCTIONAL REQUIREMENTS

The following functional requirements are based on the processing operations described above, as well as other network elements that process content:

> Packet Processing. Packet processing is a prerequisite to processing content. Unlike in the case of a host, the operation is inline, as in a proxy. In essence, enough packet-level processing is required to ensure the continued flow of traffic between the two end systems, despite the presence of an intervening network element. The content of the packets has to be recovered to enable processing; this may involve packet reassembly only or full TCP termination. Decryption may be required to convert to clear content. The exact nature of the processing depends on the application and even the stage of processing. Flexibility in packet processing is very important.

> Content Processing. Processing the recovered content often includes some form of L5 to L7 protocol processing or inspection. Decompression/compression, as well as other encoding/decoding, may need to be included. The clear content is then inspected, modified or deleted as required. Finding a field of interest may require an unanchored pattern search against thousands of patterns is required. Flexibility is even more important in content processing than in packet processing.

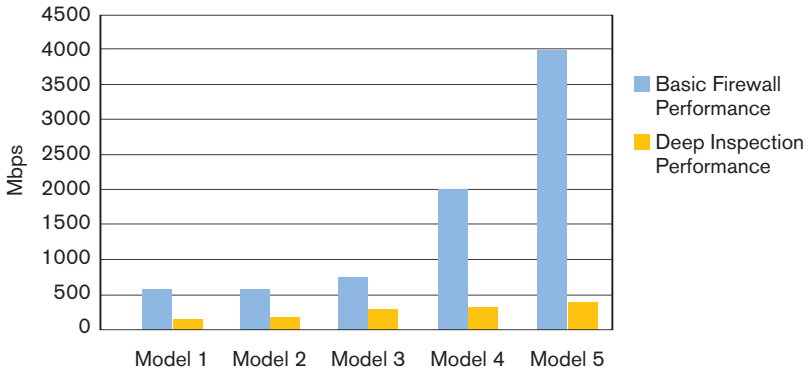
> System Operations. The system also needs to manage the many content streams associated with the connections being processed. Data may need to be copied between input, processing and output components.

4. Design Challenge

What is so special about network content processing? Host computers have been processing content since computer networks were invented. Indeed, all the processing tasks described in this white paper can be performed using software executing on a general-purpose processor.

The logical response to the rhetorical question above is performance. A PC processes content for the user of the PC. A server processes content for the clients being served. A network element processes content for many clients and servers, and the great potential of content processing in network elements can be lost if they become a bottleneck in the end-to-end traffic flow. The great design challenge of content processing is developing a solution offering flexible, line-rate content processing on high volume of traffic.

The challenge of line-rate content processing is illustrated in the graph (below) on firewall performance:



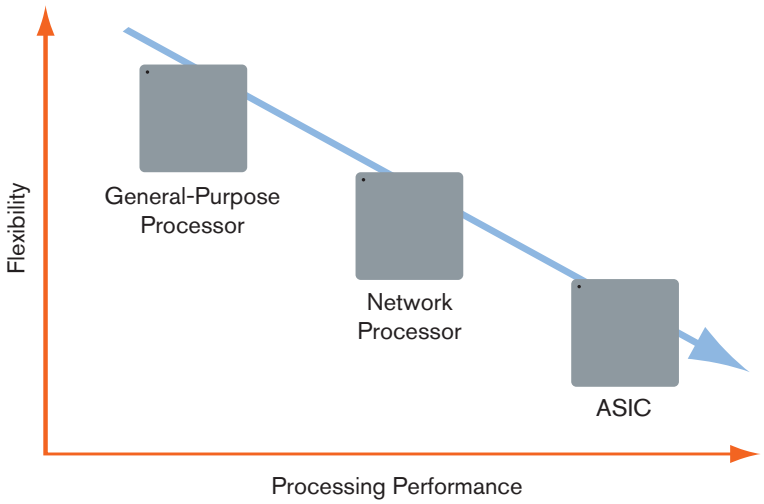
FIREWALL PERFORMANCE: BASIC VS. "DEEP INSPECTION"

Firewall performance is acceptably high when the firewall is operating in the "basic" mode, i.e., packet filtering mainly based on packet-level header information, but as mentioned earlier, basic firewalls are becoming irrelevant as security tools and must be enhanced to stop application layer attacks. Expanding a firewall's scope to include deep packet inspection and content processing leads to significant and unacceptable performance drops. An IT manager attempting to use a traditional firewall or router to stop application layer attacks by adding rules will come under enormous pressure to turn these security enhancements off and sacrifice security for performance.

5. Freescale's Network Content Processing Approach

5.1. TRADITIONAL PROCESSING SOLUTIONS

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



The general-purpose processor, with its rich instruction set and addressing space, is ideal for the functionalities of content processing. It has all the flexibility required. Unfortunately, it cannot scale to high line rates.

The other extreme is the ASIC, offering high performance but very limited flexibility. ASICs are good at providing specific, basic functions.

Network processors provide good performance with some flexibility. They are good at relatively simple LAN switching and IP forwarding functions—the applications they were designed for. Most network processors would be significantly challenged to perform Layer 4 TCP termination, and attempting to use them to process content is a waste of time.

5.2. FREESCALE APPROACH/DESIGN PRINCIPLES

The content processing solution originally created by Seaway Networks (now a part of Freescale's Digital Systems Divison) is based on a number of design principles:

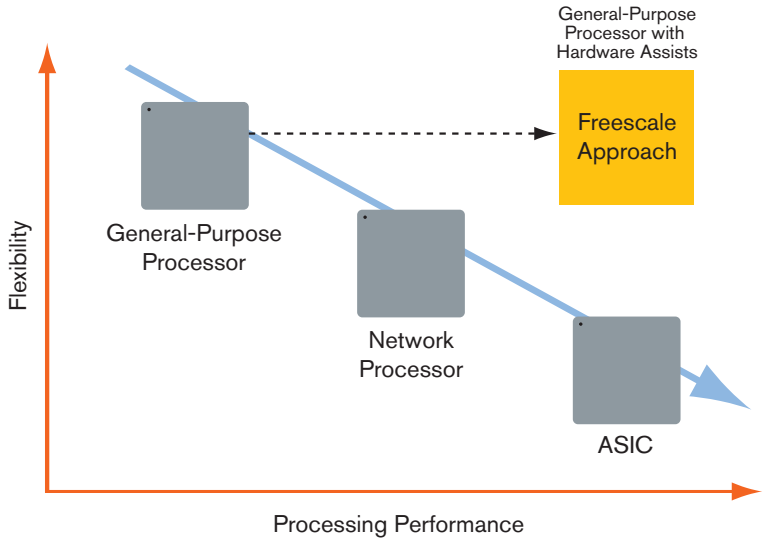
> General-purpose processor orchestrating hardware assists

> Strict distinction between packet processing and content processing domain

> Dynamic pipeline

5.3. GENERAL-PURPOSE PROCESSOR ORCHESTRATING HARDWARE ASSISTS

The following diagram illustrates the key idea behind the use of a general-purpose processor to orchestrate hardware assists, providing performance with flexibility. Hardware assists, provided by Freescale IP blocks for example, are used to do the “heavy lifting” in packet and content processing while the general-purpose processor provides the intelligence to make use of the hardware assists.



In addition to flexibility, general-purpose processors use a development environment software designers are familiar with. This is an important consideration for system designers as they select a platform.

5.4. DISTINCTIONS BETWEEN PACKET PROCESSING AND CONTENT PROCESSING DOMAINS

To properly process content, the layered principle should be adhered to, particularly for the line between layers 4 and 5. Early attempts to process content using “deep packet inspection” were flawed, or at least inadequate.

Efficient transmission is the main purpose of a packet switching network. Application messages (content) are packetized before being transmitted across the network. Application messages do not respect packet boundaries. They are not necessarily aligned with packet boundaries. They may span a number of packets.

To perform content processing properly, the content first should be recovered from the received packets, instead of peeking at a number of bytes beyond the packet header on a packet-by-packet basis. Content recovery is the key function provided in the packet processing domain, which then processes the content.

5.5. DYNAMIC PIPELINE

Pipelining is an established method of high-performance processing. In content processing, different functions are provided depending on the application and the different stages of the operation. Hence, the ability to form and change a dynamic pipeline between I/O and processing elements is very useful for flexible, high-performance processing.

6. Freescale Content Processing IP

The following table introduces Freescale's content processing IP blocks (in bold) in the context of content processing requirements.

Requirement	Solution
High-performance, flexible packet (L1–L4) processing	General-Purpose Packet Control Processor orchestrating packet processing hardware assists of **Packet Engine**
High-performance, flexible content (L5–L7) processing	General-Purpose Content Control Processor orchestrating content processing hardware assists of **Content Engine**
High-performance unanchored pattern search	**Pattern Matching Engine**
Offload processors from managing millions of flows	**Stream Switch**
Dynamic pipelining between I/O and processing blocks	**Stream Switch**

6.1. PACKET ENGINE

The packet engine provides hardware assists for packet processing functions. Working in conjunction with a general-purpose processor, it scales to multi-Gbps TCP or other L2–L4 processing. Key functions include:

> Automatic header parsing

> Automatic packet filtering

> Automatic fragment reassembly

> Automatic header and checksum validation

> Automatic flow identification

> Content recovery

> Header formatting/creation

> Segmentation/fragmentation

> Unacknowledged data management

> Header/trailer removal

6.2. CONTENT ENGINE

The content engine provides hardware assists to enable faster content processing by a general-purpose processor. Key functions include:

> Presentation

> Editing

> Replication

The content engine also contains multiple DMA engines.

6.3. PATTERN MATCHING ENGINE

Efficient pattern matching is a prerequisite to high-performance content processing. Fields in application protocol data units, unlike their counterparts in packet layers, do not have fixed offsets and widths. Hence, to process content, one has to search for patterns that correspond to the fields of interest first. This is more processing intensive than finding a packet header field. Even more processing intensive is to look for virus or spam signatures in content using unanchored searches through the content and matching against thousands or perhaps tens of thousands of patterns.

The Pattern Matching Engine is a second-generation hardware pattern matcher. It solves some problems in first-generation pattern matchers available today.

The following table shows the key features and benefits of the Pattern Matching Engine.

Feature	Benefit
H/W Pattern Matcher, multi-Gbps, thousands of patterns	Removes performance bottlenecks in network security devices requiring pattern matching
Regular Expression	Enables the use of Regex to create sophisticated and accurate signatures without performance concerns
Stateful Rule	Enables higher system performance (and accuracy) by minimizing CPU involvement in determining stateful relationships between patterns, such as in tracking application layer protocol states
High performance with a large number of patterns	No need to sacrifice performance for accuracy; performance remains high and predictable even when large number of patterns are configured
Case insensitivity	Minimizes pattern explosions for high performance and low cost
Fast pattern compilation and updates	Enables quick responses to hackers (and competitors) and new attacks
DRAM instead of SRAM or FCRAM	Lower system cost

6.4. STREAM SWITCH

The Stream Switch enables dynamic pipelining and offloads the processors from managing large numbers of content streams. Key functionality includes:

> Interconnection among I/O and processing elements

> Controls memory access

> Switching and stream management

> Policy-based scheduling

> Prioritization

> Queuing

> Copying

7. Conclusion

Driven by the need for greater network security, more efficient traffic control and more profitable IP services, network content processing is becoming an indispensable requirement for more and more network elements.

Freescale's high-performance content processing solutions are based on the combination of general-purpose processors (such as Freescale's PowerQUICC™ communications processors) working in concert with acceleration blocks such as those developed by Seaway Networks.

Freescale's acquisition of Seaway Networks in August 2005 has enabled the company to expand its portfolio in network content processing IP blocks, enhance its ability to create compelling new devices and add to its leadership in communications processing and connectivity solutions for the networked communications and pervasive computing markets.

How to Reach Us:

Home Page:
www.freescale.com

e-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor

Technical Information Center, CH370

1300 N. Alma School Road

Chandler, Arizona 85224

1-800-521-6274

480-768-2130

support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH

Technical Information Center

Schatzbogen 7

81829 Muenchen, Germany

+44 1296 380 456 (English)

+46 8 52200080 (English)

+49 89 92103 559 (German)

+33 1 69 35 48 48 (French)

support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.

Headquarters

ARCO Tower 15F

1-8-1, Shimo-Meguro, Meguro-ku,

Tokyo 153-0064, Japan

0120 191014

+81 3 5437 9125

support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.

Technical Information Center

2 Dai King Street

Tai Po Industrial Estate,

Tai Po, N.T., Hong Kong

+800 2666 8080

support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor

Literature Distribution Center

P.O. Box 5405

Denver, Colorado 80217

1-800-441-2447

303-675-2140

Fax: 303-675-2150

LDCForFreescaleSemiconductor

@hibbertgroup.com

Document Number: CNTNTPROCESSWP
Rev. 0
10/2005

freescale semiconductor