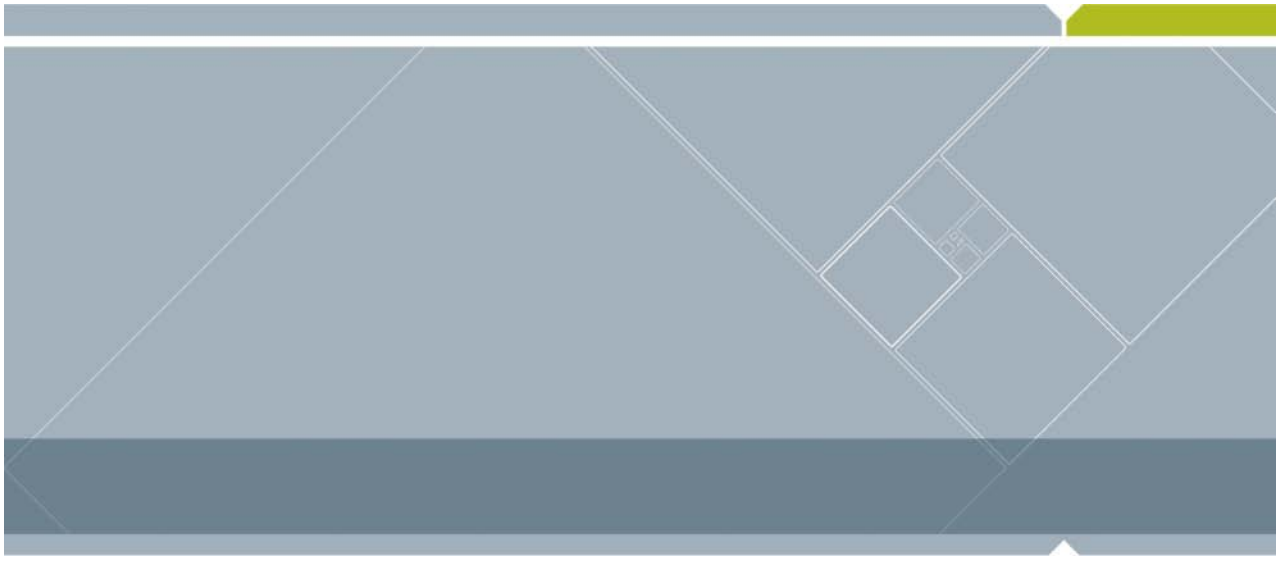


White Paper

Designing Application-Aware Networking Equipment with the PowerQUICC™ III MPC8572E

Document Number: MPC8572EWP
Rev 0
1/2007

freescale™ semiconductor

Overview

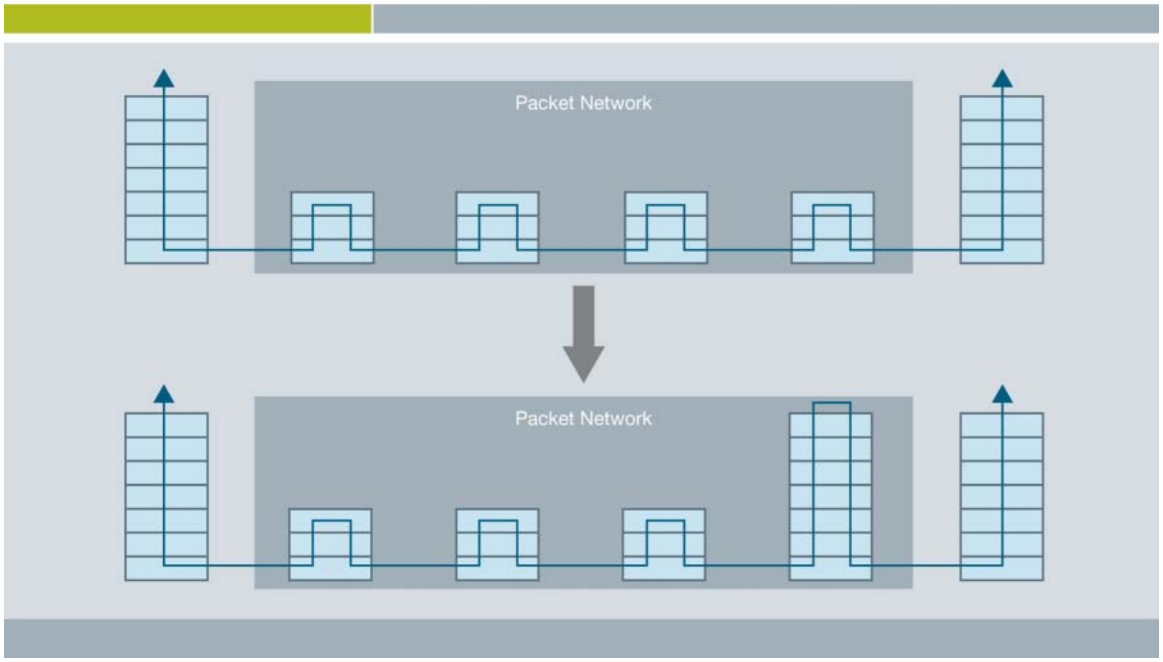
Freescale's PowerQUICC™ families of processors have long established themselves as the premier communications processors in the market, widely used in a variety of numerous networking devices including switches, routers and network security equipment. The MPC8572E, the first PowerQUICC III processor with an integrated pattern matcher, is specifically designed to satisfy additional application-aware requirements of high-performance networking devices. This white paper describes how high-performance, cost-effective application-aware networking equipment can be designed with the MPC8572E.

Contents

freescale™ semiconductor

1 Application-Aware Networking Overview

Trend Towards Application-Aware Networking



Traditionally, network nodes operate at Layer 3 and below, as depicted in the classic Open Systems Interconnection (OSI) Model shown in the top section of the above diagram. The key role of a network node is to forward packets based on the network layer address in the case of a router or the data link layer address in a LAN switch. The network node is not aware of the application content carried in the payload part of the packets—processing the application layer is strictly the role of end systems, i.e., the PCs and servers attached to the network.

However, in recent years more and more network nodes are inspecting and even altering the application layer protocol and content in order to carry out their networking functions.

In this white paper, we will use the term “application-aware networking” to refer to the function of network nodes processing application layer protocol and content, in addition to packet header, in order to perform the networking features they are designed to provide.

1.1 Application-Aware Networking Examples

Below are some networking functions that are now being performed in an application-aware manner in some network nodes:

- Forward data based on content
- Ensure data being forwarded is secure
- Traffic manage the data being forwarded
- Collect statistics on the data being forwarded

1.1.1 Application-Aware Data Forwarding

The most fundamental networking function is to simply forward packets to their respective destinations.

In server load balancing, the destination IP address is insufficient to determine the best server to provide the requested content. The server load balancer or content switch inspects the uniform resource identifier (URI) content and other application protocol fields in order to forward the packet to the most appropriate destination.

Taking one big step further, XML appliances validate, transform and route XML messages based on the application content.

1.1.2 Application-Aware Security

The requirement for network security equipment to be application-aware has dramatically increased as the amount and frequency of harmful traffic has exploded. In recent years, over 90 percent of network attacks have exploited application vulnerabilities.[1] In essence, the security equipment needs to look into the application content in order to determine whether a flow of packets is safe or not. This white paper, focuses more on the non-security aspect of application-aware networking. Readers interested in the security aspect are urged to read other white papers in this series.[2, 3, 4, 5]

1.1.3 Application-Based Traffic Management

In many organizations, there is not enough network bandwidth to provide good application performance to all the employees, partners and customers. The unsatisfactory application performance is exacerbated by two trends:

- Non-business applications such as peer-to-peer (P2P) for music and video sharing are cluttering corporate networks, taking bandwidth away from business critical applications
- Corporations are centralizing servers, forcing applications originally designed to work over LANs to now work across WAN facilities that have significantly lower bandwidth and higher latency than their LAN counterparts

Application-aware traffic management devices are designed to help remote applications run faster by managing traffic flows based on application.

Enterprise networks are not exclusively responsible for managing how limited network resources are used by application. ISPs also want to make sure that their uplinks to the core Internet are properly utilized by applications, or the performance experienced by all of their users will be negatively impacted.

The most useful function is to provide visibility of network traffic based on application. For example, instead of showing that the link to the Internet is 99 percent utilized, the device also shows the usage of the bandwidth by applications, for example:

- P2P: 75 percent
- Web: 17 percent
- e-Mail: 4 percent
- VoIP: 3 percent

The next step is to allocate bandwidth and provide priority on a per-application basis. In the example above, realizing how the limited bandwidth is used, the network operator has the capability to limit the bandwidth allocated to P2P and assign a high priority to VoIP traffic.

In addition to traffic management, some application acceleration devices may compress and decompress application content, adjust chatty protocols and cache content in order to achieve higher performance. To provide these functions, the network node needs to be application-aware.

1.1.4 Application-Based Statistics Collection

Application-based statistics collection provides visibility of network traffic based on application, as mentioned in section 1.1.3.

Some service providers want to bill based on application in addition to, or instead of, traffic volume and link speed.

[1] Symantec Internet Security Threat Report, Trends for July 05–December 05, Volume IX, March 2006
[2] Designing Firewall/VPN with the PowerQUICC III MPC8572E
[3] Designing IDS/IPS with the PowerQUICC III MPC8572E
[4] Freescale / Kaspersky Accelerated Anti-Virus (Accelerated AV) Solution Platform for OEM Vendors
[5] Designing UTM with the PowerQUICC III MPC8572E

2 Design Challenges

One of the first challenges facing the designer attempting to build a competitive piece of application-aware networking equipment is to select the right processor for the job.

The processor must be flexible enough to enable the rapid development of the increasing number of complex features. It must have the performance to drive line-rate throughput with low latency. To be competitive, it must enable a cost-effective system design with short development cycle. And, as an embedded processor, it must operate within a tight power budget.

The functions in a traditional networking device and an application-aware networking device are the same in terms of forwarding, security, traffic management and statistics collections. Hence, the processor for application-aware networking must first and foremost be a good processor for traditional networking. In addition, it needs to be able to perform the new "application-aware" operations at high speed.

Let us first take a look at some key features of a processor optimized for networking before we discuss the application-aware specifics:

- General-purpose CPU core with standard ISA to enable the rapid development of complex features
- One or more high-performance CPU cores to enable high performance while managing power dissipation
- Large cache to enable efficient utilization of CPU cycles while performing complex functions on a large number of flows by minimizing cache misses
- Large bus and memory bandwidth to avoid bottlenecking data movement in high-throughput networking applications
- Integrated memory controller to save system cost
- Integrated hardware to accelerate and offload CPU-intensive operations and to lower power dissipation. Examples include:
  - Lookup table for various forwarding operations
  - Checksum calculation for integrity assurance of transmit and receive frames
  - Cryptographic operations for security
  - Priority and other queuing mechanisms for Quality of Service (QoS)
- Integrated network interfaces to enable lower cost
- Other integrated standard interfaces for high-speed connectivity to other components in the system
- Good system design so that various functional blocks operate efficiently together in a pipeline to achieve high throughput and low latency

With respect to being application-aware, Section 4.2, will explain that the difference in the application-aware approach is that a packet flow has to be classified differently, based on the packet flow's application, before networking functions are performed on the flow.

To appreciate the challenge associated with application-based classification, let us use an open-source example. L7-filter is an application layer packet classifier for Linux®. It has been reported that "when all 70 protocol filters[6] are enabled … the system throughput drops to less than 10 Mbps… Moreover, over 90 percent of the CPU time is spent in regular expression matching, leaving little time for other … functions."[7]

The key additional attribute of a processor optimized for application-aware networking is the ability to look into the payload of a packet flow to determine the nature of the application protocol and content at high speed. A processor with an integrated pattern matcher with Regex capability will likely be able to enable the identification of an application protocol accurately and quickly. Regex pattern matching hardware is also extremely useful in accelerating content security operations.[3, 4, 5]

[6] There were 96 supported protocols on July 18, 2006
[7] Fang Yu et al., Fast and Memory-Efficient Regular Expression Match for Deep Packet Inspection

3 MPC8572E PowerQUICC III Processor Overview

The MPC8572E is a new PowerQUICC III processor purposely built to meet the requirements of high-performance application-aware networking and content security. It is based on the highly successful PowerQUICC system-on-chip (SoC) platform, well-proven in traditional networking, and enhanced with further integration of new hardware, optimized to process application content at high speeds.

The MPC8572E consists of dual e500 cores built on Power Architecture™ technology, achieving clock speeds from 1.2 GHz to 1.5 GHz. The CPU cores, each with 32 KB I-Cache and 32 KB D-Cache, share 1024 KB of integrated L2 cache. For memory, the MPC8572E includes two integrated 64-bit DDR2/DDR3 SDRAM controllers.

To further speed up processing while keeping power dissipation down, the MPC8572E integrates powerful engines: a security engine that accelerates crypto operations in IPSec and SSL/TLS; a pattern-matching engine to handle regular expression matching; a deflate engine to manage file decompression; and two table lookup units (TLU) that manage complex table searches and header inspections.

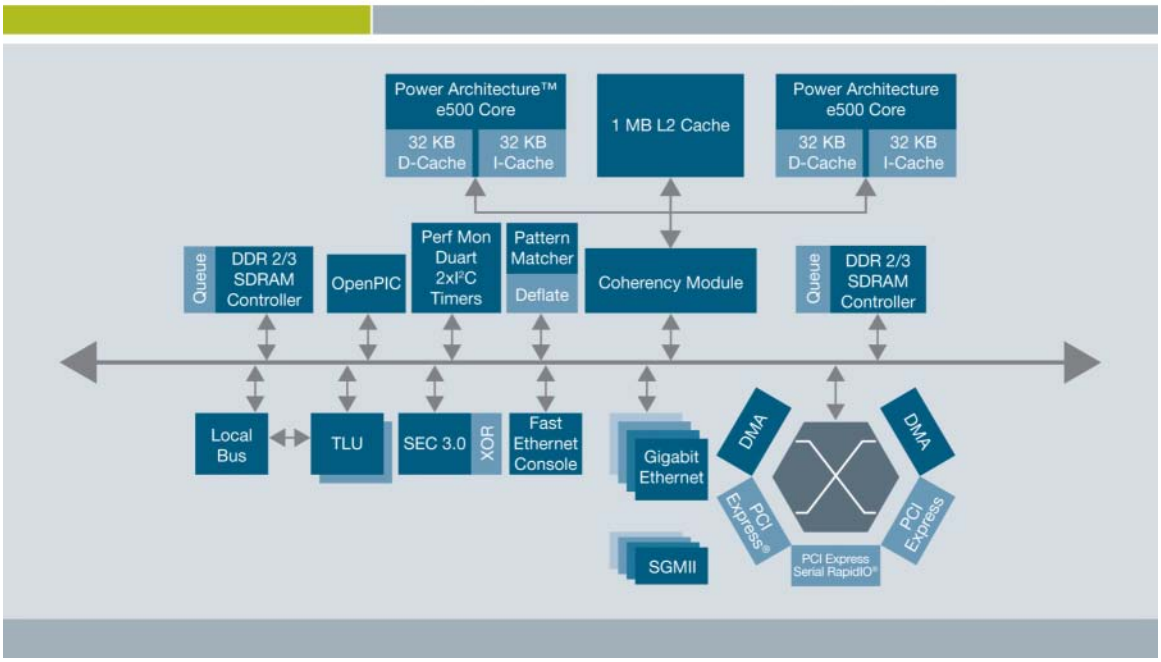
The MPC8572E offers a combination of network interfaces, including four integrated enhanced Triple-Speed Ethernet controllers (eTSEC). These controllers accelerate packet I/O by offloading checksum calculation. They also provide QoS support with eight Rx and eight Tx hardware queues to accelerate traffic management.

For high-speed connectivity to other devices, the MPC8572E supports PCI Express®, Serial RapidIO® and DMA interfaces.

All major processing and I/O elements are integrated into the MPC8572E with a highly optimized internal interconnect architecture to ensure high bandwidth, low latency and efficient pipeline operation, balancing processing performance with I/O system throughput.

Based on Freescale's 90 nm silicon-on-insulator (SOI) copper interconnect process technology, the MPC8572E is designed to deliver higher performance with lower power dissipation.

MPC8572E PowerQUICC™ III Block Diagram



The Pattern Matcher is the key contributor to the MPC8572E's ability to process packet content at high speed, for application-aware networking and content security applications.

The Pattern Matcher is an integrated hardware block inside the MPC8572E with the following capabilities:

- High-performance, feature-rich hardware pattern matching of compressed and uncompressed data
  - Patterns expressed in regex with significant capabilities beyond that provided by the regex language
  - Stateful Rule – correlates multiple pattern matches and maintains state between matches
- Improvements over other pattern matching technologies:
  - No pattern “explosion” to support “wildcarding” or case-insensitivity
  - Fast compilation of pattern database
  - Fast incremental additions to pattern database
  - Live pattern database update
  - Patterns stored in main DDR DRAM, not SRAM or FCRAM
- On-chip hash tables for low system memory utilization, removing need for costly low-latency memory technologies

Pattern matching across data “work units” (e.g. can match patterns split across TCP segments)

4 Designing Application-Aware Networking Equipment with the MPC8572E

In order to understand what an ideal application-aware networking platform looks like, let us first examine the key operations performed in a regular networking device. We then examine what else is required if the device is to become application aware.

4.1 Flow-Based Packet Processing

Typically, a flow-based packet processing approach is used in high-performance networking equipment. The operation can be separated into:

- Control path
- Data path

(Alternatively, the approach can be described as “first packet” and “subsequent packet” processing. For simplicity, we’ll loosely use the terms “control path” and “data path” in this paper, with the understanding that the “first packet” may or may not be carried in a separate “control” connection.)

The control path is performed at the beginning of a flow, i.e. when a new packet is received and there is no flow table entry matching the characteristics of the packet. The key steps are:

- Classify the flow itself and its child flows (as in FTP, SIP, etc.)
- Consult policy table to determine how the flows are to be processed, e.g., allowed, denied, “tunnel”, apply suitable QoS, keep statistics, etc.
- For each flow, add an entry to the flow table stating how subsequent packets in the flow are to be processed, including the packet at hand

Compared with data path, control path processing is more complex. The complexity mainly affects the latency of the first packet, unless control path processing is so time-consuming that it starves the data path of CPU cycles.

The data path is as follows:

- Receive packet
- Look up entry in a potentially very large flow table
- Process according to the “recipe” in entry, e.g. forward, apply QoS, collect statistics
- Transmit packet

The operation in the data path is less complex than that in the control path. The packet I/O, however, is typically quite high. The efficiency of the data path therefore largely determines the throughput performance of the traditional network node. Designers therefore put extra effort into optimizing the data path.

4.2 Application-Based Flow Classification

Classifying packet flows based on application is the prerequisite of application-based networking.

Traditionally by design, applications use a fixed port number. Hence, the application of a packet flow can be identified simply and quickly by looking up the protocol and port# values in the packet header in the table known port numbers like the following:

Protocol	Port#	Application
TCP	21	FTP
TCP	23	Telnet
TCP	25	SMTP
TCP	80	HTTP
UDP	53	DNS

But today, some applications, especially those used in P2P, deliberately disguise themselves by:

- Allowing users to change the default port#
- Use a random port#
- Use a port# that belongs to other applications, e.g., HTTP's port 80

For accurate classification based on application, the payload of a packet flow needs to be matched with a set of "application signatures." The table below shows some examples of such signatures from the L7-filter site.[8] L7-filter is an open source application layer packet classifier for Linux. It can classify packets as Kazaa, HTTP, Jabber, Citrix, Bittorrent, FTP, Gnucleus, eDonkey2000, etc., regardless of port#.

Unfortunately, as mentioned earlier in Section 2, it has been reported that "when all 70 protocol filters[9] are enabled … the system throughput drops to less than 10 Mbps… Moreover, over 90 percent of the CPU time is spent in regular expression matching, leaving little time for other … functions."[10]

Application-based flow classification is the prerequisite for application-aware networking equipment. The operations required are so processing-intensive that they can take away a significant portion of the available CPU cycles from the data path. As a result, the control path, in addition to the data path, needs to be optimized.

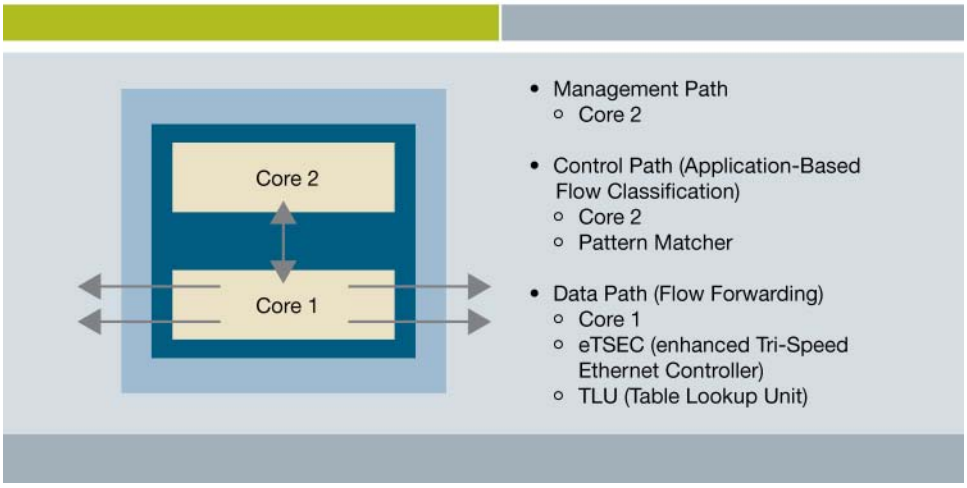
[8] http://l7-filter.sourceforge.net/
[9] There were 96 supported protocols on July 18, 2006
[10] Fang Yu et al., Fast and Memory-Efficient Regular Expression Match for Deep Packet Inspection

Protocol	Protocol Type	Application Signature
100bao	P2P	^\x01\x01\x05\x0a
aim	Chat	^(*[\x01\x02].*\x03\x0bl*\x01.?.?.?.?\x01)lflaponltoc_signon.*0x
aimwebcontent	Chat	user-agent:aim/
applejuice	P2P	^ajprot\x0d\x0a
ares	P2P	^\x03[]Z].?.?\x05$
battlefield1942	Game	^\x01\x11\x10\l\xf8\x02\x10\x40\x06
battlefield2	Game	^(\x11\x20\x01\xa0\x98\x11l\xfe\xfd.?.?.?.?.?.?(\x14\x01\x06l\xff\xff\xff))l[]\x01].?battlefield2
bgp	Networking	^\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff..?\x01[\x03\x04]
biff	Mail	^[a-z][a-z0-9]+@[1-9][0-9]+$
bittorrent	P2P	\x13bittorrent protocolld1:ad2:id20:l\x08'7P\)[RP]l^azver\x01$l^get /scrape?info_hash=
ciscovpn		^\x01\xf4\x01\xf4
citrix		\x32\x26\x85\x92\x58
counterstrike-source	Game	^\xff\xff\xff\xff.*cstrikeCounter-Strike
cvs		^BEGIN (AUTHIVERIFICATIONIGSSAPI) REQUEST\x0a
dayofdefeat-source	Game	^\xff\xff\xff\xff.*dodDay of Defeat
dhcp	Networking	^[\x01\x02][\x01-]\x06.*c\x82sc
directconnect	P2P	^(\$mynick l\$lock l\$key)
dns	Networking	^.?.?.?.?[\x01\x02].?.?.?.?.?.?[\x01-?][a-z0-9][\x01-?a-z]*[\x02-\x06][a-z][a-z][fglmoprstuvz]?[aeop]?(um)?[\x01-\x10\x1c][\x01\x03\x04\xFF]
doom3	Game	^\xff\xffchallenge
edonkey	P2P	^[\xc5\xd4\xe3-\xe5].?.?.?.?([\x01\x02\x05\x14\x15\x16\x18\x19\x1a\x1b\x1c\x20\x21\x32\x33\x34\x35\x36\x38\x40\x41\x42\x43\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58[\x60\x81\x82\x90\x91\x93\x96\x97\x98\x99\x9a\x9b\x9c\x9e\xa0\xa1\xa2\xa3\xa4]l\x59.....?[~]l\x96....$)
fasttrack	P2P	^get (/.download/[~]*l/.supernode[~]l/.status[~]l/.network [~]*l/.filesl/.hash=[0-9a-f]*/[~]*) http/1.1luser-agent: kazaalx-kazaa (-usernamel-networkl-ipl-supernodeipl-xferidl-xferuidltag)l^give [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]?[0-9]?[0-9]?

4.3 Application-Aware Networking Operations on the MPC8572E

The dual-core MPC8572E can be used either in the Symmetric Multi-Processing (SMP) or the Asymmetric Multi-Processing (AMP) mode. This white paper illustrates how application-aware networking functions can be implemented in the AMP mode.

Dual-Core Usage Model for Application-Aware Networking Operations



In essence,

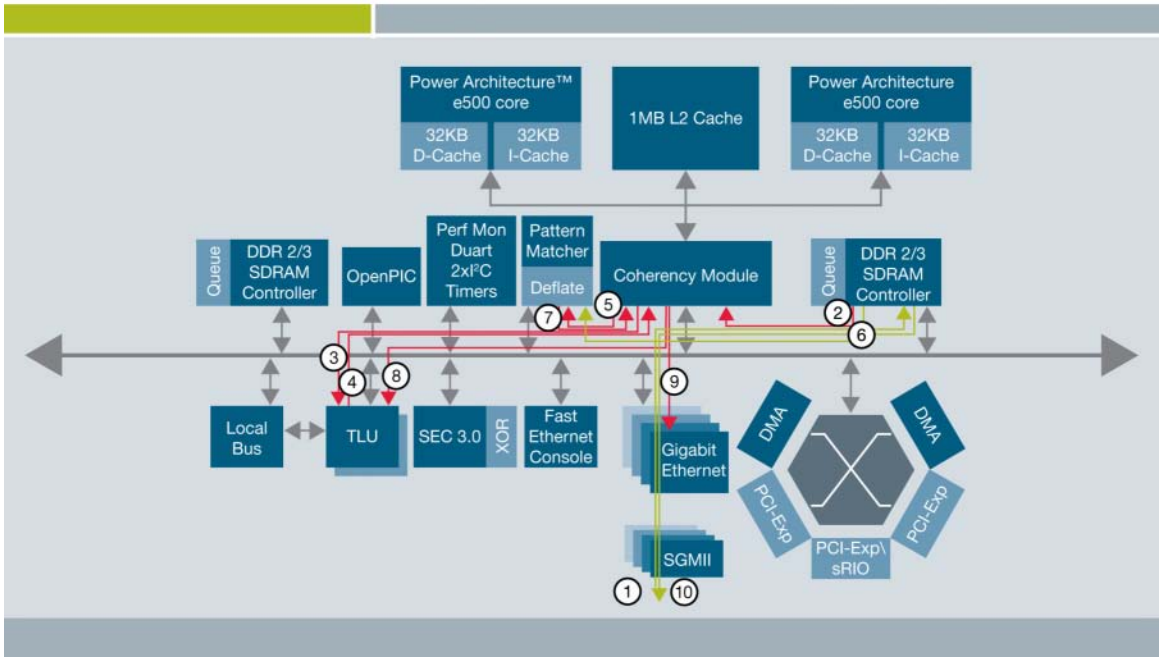
- Core2, working in conjunction with the Pattern Matcher, is used for the CPU-intensive matching of pre-classified network data with application signatures. Core2 is also used for the management tasks.
- Core1, working in conjunction with the TLUs and eTSECs, is used for the data path, such as packet I/O, forwarding, controlling QoS and updating statistics.

4.3.1 Control Path

The control path is shown in detail in the diagram below:

1. Ethernet controller puts received packet into appropriate queue in memory and informs Core1
2. Core1 extracts 5-tuple key from the packet header
3. Core1 writes key to TLU to lookup flow table
4. Core1 reads back lookup (negative) result and informs Core2
5. Core2 instructs Pattern Matcher to scan content of packet against application protocol signatures
6. Pattern Matcher reads data from memory and scans for patterns
7. PM informs Core2 of scan result
8. Core2 adds flow table entry in TLU and informs Core1
9. Core1 process packet as per recipe and instructs Ethernet port to transmit
10. Data retrieved from memory and transmitted, apply QoS as required

Application-Aware Networking Control Path on MPC8572E

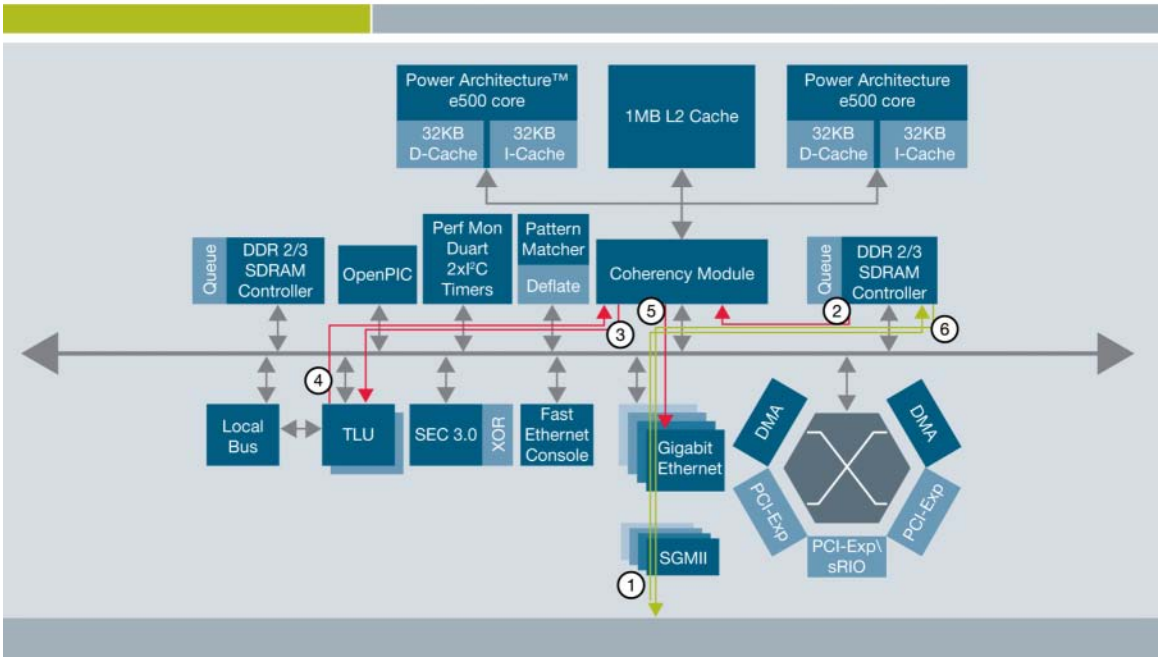


4.3.2 Data Path

In more details, the data path is shown in the accompanying diagram:

1. Ethernet controller puts received packet into appropriate queue in memory and interrupts Core1
2. Core1 extracts 5-tuple key from the packet header
3. Core1 writes key to TLU to lookup flow table
4. Core1 reads back (+ve) results of the lookup and retrieves additional flow entry data from memory if required, and processes packet as per “recipe” in flow table
5. Core1 instructs appropriate Ethernet controller to transmit packet
6. Ethernet controller transmits packet, applying QoS as required

Application-Aware Networking Data Path on MPC8572E



4.4 Performance Advantages of the MPC8572E in the Control Path

A powerful e500 core, working in conjunction with the hardware Pattern Matcher, is used in the control path that matches unclassified packet payload against application signatures.

This highly CPU-intensive operation is off-loaded and accelerated by the hardware Pattern Matcher.

The interaction between the e500 core and the Pattern Matcher is very efficient via descriptors in L2 cache.

As a result, the “first packet delay” is significantly reduced. In addition, there are more CPU cycles left for other operations, resulting in higher performance in general.

4.5 Accuracy Advantages of the MPC8572E in Control Path

The built-in Pattern Matcher that offloads and accelerates matching of a packet payload against application signatures has a number of features and operational characteristics that are conducive to accuracy:

- Regex
- Stateful rule
- Matching across packet boundaries
- Performance minimally dependent on the number of signatures

4.5.1 Regex

The Regex Compiler associated with the MPC8572E's Pattern Matcher supports a major subset of the Practical Extraction and Report Language (PERL) regular expression syntax as well as capabilities beyond that provided by PERL. The Pattern Matcher can match thousands of Regexes in parallel at multi-Gbps speed.

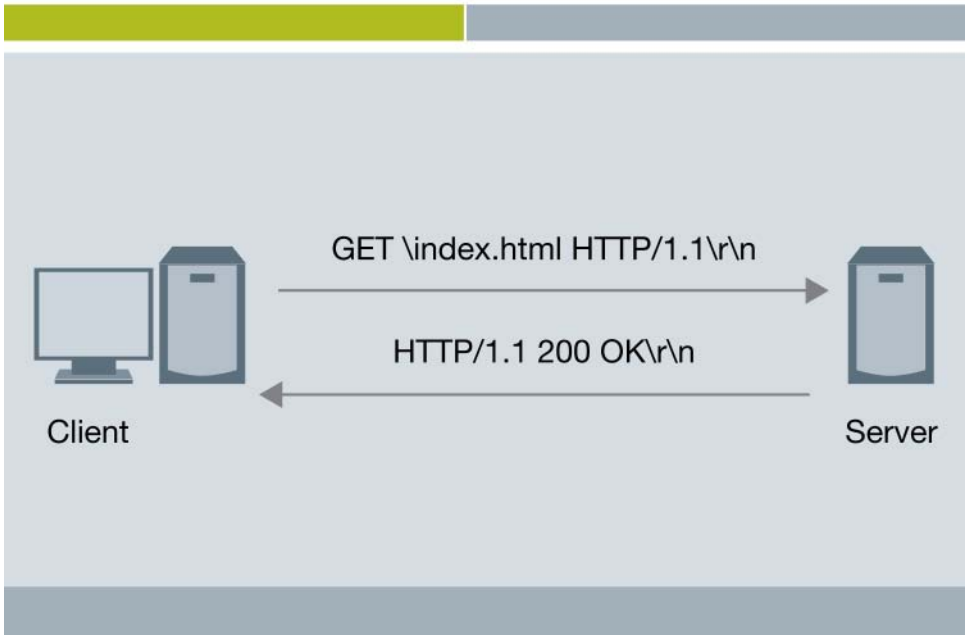
This means that the signature designer has a very powerful tool at his or her disposal to design sophisticated signatures to achieve high accuracy without worrying about performance.

4.5.2 Stateful Rule

The Pattern Matcher's Stateful Rule capability can be used to track application protocols and create the context for stateful pattern matching to achieve high accuracy.

The following diagram shows a typical HTTP request-response exchange.

Typical HTTP Request-Response Exchange



As a first step, the application signature designer can create a very accurate signature using Regex based on the request or the response. To take a step further, the designer can build an even more accurate signature based on the protocol request-response exchange itself as well as the regexes representative of the request and the response. A simplified example is illustrated below:

1. Define regex signatures of HTTP request and response

```
http_request /^(get|post)\s.*?http/1\.\d$/i
http_response /^http/1\.\d\s200\sOK$/i
```

2. Define Stateful Rule matching the protocol exchange

```
STATEFUL_RULE: HTTP_Recognizer
     RESET_STATE:
             EVENT "http_request"
                     next_state AWAIT_response
     STATE AWAIT_response:
             EVENT "http_response"
                     # report HTTP traffic observed
                     report {0x00000001}
                     next_state RESET_STATE
```

4.5.3 Matching Across Packet Boundaries

Application messages do not necessarily follow packet boundaries. For accurate application-based classification, matching across packet boundaries is required. This capability is supported in the MPC8572E's Pattern Matcher.

4.5.4 Performance Minimally Dependent on the Number of Signatures

A classifier with low pattern matching performance, such as one implemented with software, puts the IT manager in an awkward position to choose between accuracy or speed:

- Configuring relatively few signatures to achieve higher performance at the risk of reducing application classification accuracy, or
- Configuring the complete set of signatures to detect all the known application protocols but suffer from lower performance as a result

To solve this problem, the throughput performance of MPC8572E's integrated Pattern Matcher is minimally affected by the number of patterns configured.

4.6 Performance Advantages of the MPC8572E in Data Path

4.6.1 Packet I/O

The application-aware networking device in question is, as the name says clearly, a networking device. As such, it should be able to receive and transmit packets at a high rate as a pre-requisite. The overhead in servicing a high rate of transmit and receive interrupts is high and can significantly slow down the performance of the device. The integrated Ethernet controller is able to coalesce interrupts, thereby reducing the interrupt servicing overhead and improve performance.

In a networking device where the I/O rate is high, memory access speed in addition to the availability of CPU cycles can have a significant impact on system performance. The integrated Ethernet controller on the MPC8572E stashes received packet headers in L2 cache while writing to memory. As a result, the e500 CPU core accesses data with reduced latency. In fact, the transmit and receive buffer descriptors can be locked in the L2 cache for fast access by the Ethernet controller and the e500 core.

4.6.2 Packet Processing

A powerful e500 CPU core of up to 1.5 GHz is dedicated to provide the CPU cycles (and flexibility) required for packet-layer processing.

Furthermore, the following operations are off-loaded from the CPU core:

- IP and TCP checksum calculations to the Ethernet controller
- Flow table lookup to the TLU

Checksum calculations are required for every packet received. Off-loading this calculation results in less software execution and higher performance.

As described earlier in Section 4.1, searching for an existing entry in a potentially very large flow table is performed every time a packet is received. This operation can be off-loaded to the built-in TLU on the MPC8572E.

4.6.3 Traffic Management

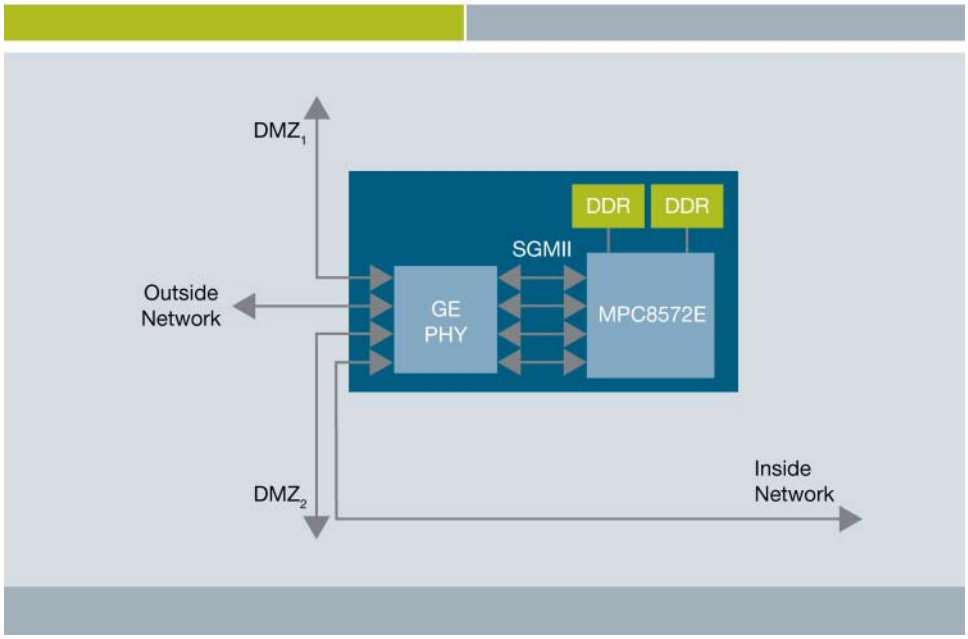
Some Traffic Management functions can be accelerated and off-loaded to the integrated Ethernet controllers. The eTSECs have built-in QoS support for eight Rx and eight Tx hardware queues. The transmit scheduling can be set to strict priority or modified weighted round robin.

4.7 Hardware Platform Design with the MPC8572E

In essence, an application-aware networking device essentially receives packets, processes the packet's header and the application content in the packet payload, and transmits the packet.

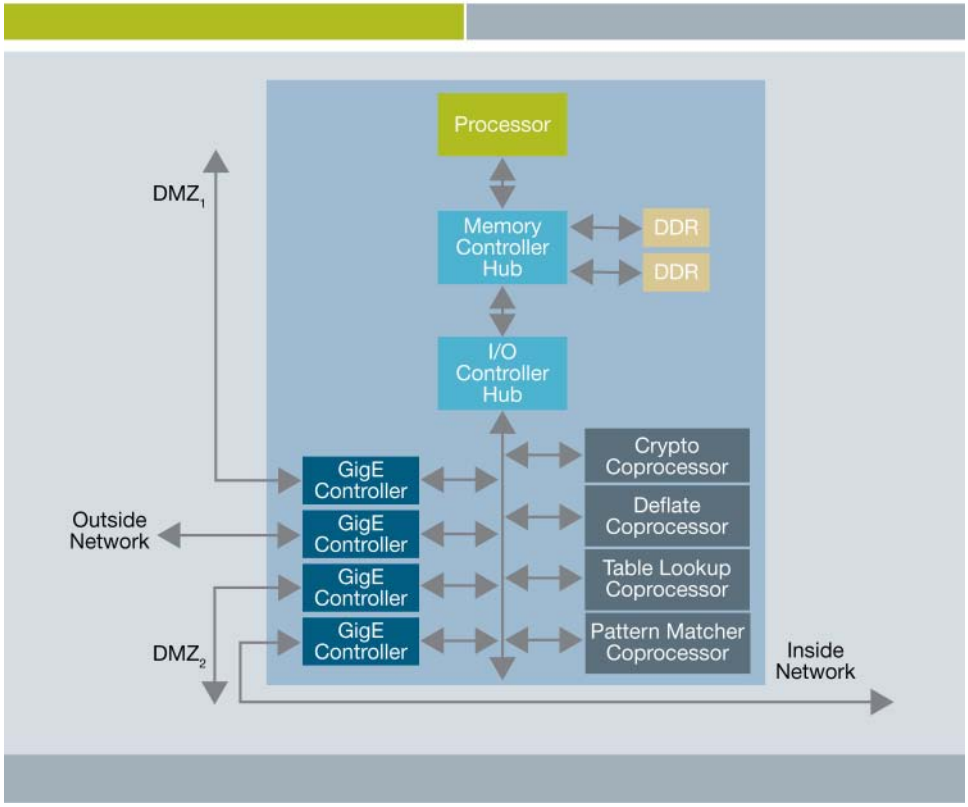
The simplified block diagram below shows the essence of a 4-port network appliance, illustrating how easy a system design using the MPC8572E can be.

4-Port Networking Appliance with the MPC8572E



In contrast, an equally simplified block diagram of a different 4-port network appliance of similar capabilities designed with a typical processor is shown below:

4-Port Networking Appliance with Less Integrated Processor



4.7.1 Cost Advantages

The very simple system design—a direct result of the exceptional integration in the MPC8572E—enables significantly lower system cost and shorter time to market.

There is no separate memory controller hub, I/O controller hub, Gigabit Ethernet controllers, table lookup coprocessor and pattern matcher coprocessor to complicate the design and add to the cost.

Specific to the Pattern Matcher, there is also no separate expensive low latency memory—the MPC8572E's built-in Pattern Matcher does not need it for high performance, unlike other pattern matching engines on the market.

5 Summary

Freescale's PowerQUICC family of processors has a long legacy and an established reputation as the premier family of communications processors in the market, widely used in a variety of networking devices including switches, routers and network security devices. The MPC8572E, the first PowerQUICC III processor with an integrated Pattern Matcher, is specially designed to satisfy the additional requirement of high-performance networking devices to be application aware.

The MPC8572E's dual e500 cores provide CPU cycles and flexibility to execute the software for control and data plane operations. The performance of the MPC8572E is further enhanced by the integrated hardware blocks that off-load and accelerate CPU-intensive operations with low power dissipation:

- TLU: manages various table lookup operations widely used in packet forwarding and security
- eTSEC: allows traffic management and checksum calculation of sent and received packets
- SEC: enables crypto operations
- Deflate: provides decompression
- Pattern Matcher: matches packet payload against signatures of application protocols and undesirable content

The built-in Pattern Matcher, with its features and operational characteristics, is particularly conducive to providing high performance and accuracy simultaneously in application-based flow classification, the pre-requisite of application-aware network equipment:

- Regex allows the creation of sophisticated application signatures that are fingerprints of messages of specific applications
- Stateful rule enables even more accurate signatures by tracking application protocol exchange in addition to application messages
- Matching patterns across packet boundaries to increase accuracy by matching application messages that span packet boundaries
- Performance minimally dependent on number of signatures enables the IT manager to configure the complete set of signatures to provide fine-grain granularity and accuracy without worrying about degradation of performance

The MPC8572E processor, with all major processing and I/O elements included, enables very simple, elegant system design, with low system cost and a short design cycle. Contributing further to cost-effectiveness is the Pattern Matcher's use of DRAM instead of expensive low latency SRAM or FCRAM.

As a result, OEMs can count on using the MPC8572E to deliver highly competitive application-aware networking products to the market.


freescale™
semiconductor

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: MPC8572EWP
REV 0

freescale™ semiconductor