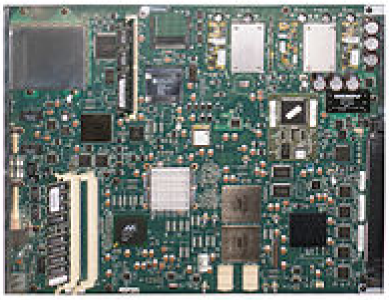
WIKIPEDIA

Forwarding plane

Cisco VIP 2-40, from an older generation of routers.



Performance Route Processor, from the high-end Cisco 12000 series.

In routing, the **forwarding plane**, sometimes called the **data plane** or **user plane**, defines the part of the router architecture that decides what to do with packets arriving on an inbound interface. Most commonly, it refers to a table in which the router looks up the destination address of the incoming packet and retrieves the information necessary to determine the path from the receiving element, through the internal **forwarding fabric** of the router, and to the proper outgoing interface(s). The IP Multimedia Subsystem architecture uses the term **transport plane** to describe a function roughly equivalent to the routing control plane.

In certain cases the table may specify that a packet is to be discarded. In such cases, the router may return an ICMP "destination unreachable" or other appropriate code. Some security policies, however, dictate that the router should drop the packet silently, in order that a potential attacker does not become aware that a target is being protected.

The incoming forwarding element will also decrement the time-to-live (TTL) field of the packet, and, if the new value is zero, discard the packet. While the Internet Protocol (IP) specification indicates that an Internet Control Message Protocol (ICMP) time exceeded message should be sent to the originator of the packet (i.e. the node indicated by the source address), the router may be configured to drop the packet silently (again according to security policies).

Depending on the specific router implementation, the table in which the destination address is looked up could be the routing table (also known as the routing information base, RIB), or a separate forwarding information base (FIB) that is populated (i.e., loaded) by the routing control plane, but used by the forwarding plane for look-ups at much higher speeds. Before or after examining the destination, other tables may be consulted to make decisions to drop the packet based on other characteristics, such as the source address, the IP protocol identifier field, or Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port number.

Forwarding plane functions run in the forwarding element.[1] High-performance routers often have multiple distributed forwarding elements, so that the router increases performance with parallel processing.

The outgoing interface will encapsulate the packet in the appropriate data link protocol. Depending on the router software and its configuration, functions, usually implemented at the outgoing interface, may set various packet fields, such as the DSCP field used by differentiated services.

In general, the passage from the input interface directly to an output interface, through the fabric with minimum modification at the output interface, is called the *fast path* of the router. If the packet needs significant processing, such as segmentation or encryption, it may go onto a slower path, which is sometimes called the *services plane* of the router. Service planes can make forwarding or processing decisions based on higher-layer information, such as a Web URL contained in the packet payload.

Contents

Issues in router forwarding performance

Vendors design router products for specific markets. Design of routers intended for home use, perhaps supporting several PCs and VoIP telephony, is driven by keeping the cost as low as possible. In such a router, there is no separate forwarding fabric, and there is only one active forwarding path: into the main processor and out of the main processor.

Routers for more demanding applications accept greater cost and complexity to get higher throughput in their forwarding planes.

Several design factors affect router forwarding performance:

- Data link layer processing and extracting the packet
- Decoding the packet header
- Looking up the destination address in the packet header
- Analyzing other fields in the packet
- Sending the packet through the "fabric" interconnecting the ingress and egress interfaces
- Processing and data link encapsulation at the egress interface

Routers may have one or more processors. In a uniprocessor design, these performance parameters are affected not just by the processor speed, but by competition for the processor. Higher-performance routers invariably have multiple processing elements, which may be general-purpose processor chips or specialized application-specific integrated circuits (ASIC).

Very high performance products have multiple processing elements on each interface card. In such designs, the main processor does not participate in forwarding, but only in control plane and management processing.

Benchmarking performance

In the Internet Engineering Task Force, two working groups in the Operations & Maintenance Area deal with aspects of performance. The Interprovider Performance Measurement (IPPM) group focuses, as its name would suggest, on operational measurement of services. Performance measurements on single routers, or narrowly defined systems of routers, are the province of the Benchmarking Working Group (BMWG).

RFC 2544 is the key BMWG document.[2] A classic RFC 2544 benchmark uses half the router's (i.e., the device under test (DUT)) ports for input of a defined load, and measures the time at which the outputs appear at the output ports.

Forwarding information base design

Originally, all destinations were looked up in the RIB. Perhaps the first step in speeding routers was to have a separate RIB and FIB in main memory, with the FIB, typically with fewer entries than the RIB, being organized for fast destination lookup. In contrast, the RIB was optimized for efficient updating by routing protocols.

Early uniprocessing routers usually organized the FIB as a hash table, while the RIB might be a linked list. Depending on the implementation, the FIB might have fewer entries than the RIB, or the same number.

When routers started to have separate forwarding processors, these processors usually had far less memory than the main processor, such that the forwarding processor could hold only the most frequently used routes. On the early Cisco AGS+ and 7000, for example, the forwarding processor cache could hold approximately 1000 route entries. In an enterprise, this would often work quite well, because there were fewer than 1000 server or other popular destination subnets. Such a cache, however, was far too small for general Internet routing. Different router designs behavedin different ways when a destination was not in the cache.

Cache miss issues

A **cache miss** condition might result in the packet being sent back to the main processor, to be looked up in a **slow path** that had access to the full routing table. Depending on the router design, a cache miss might cause an update to the fast hardware cache or the fast cache in main memory. In some designs, it was most efficient to invalidate the fast cache for a cache miss, send the packet that caused the cache miss through the main processor, and then repopulate the cache with a new table that included the destination that caused the miss. This approach is similar to an operating system with virtual memory, which keeps the most recently used information in physical memory

As memory costs went down and performance needs went up, FIBs emerged that had the same number of route entries as in the RIB, but arranged for fast lookup rather than fast update. Whenever a RIB entry changed, the router changed the corresponding FIB entry

FIB design alternatives

High-performance FIBs achieve their speed with implementation-specific combinations of specialized algorithms and hardware.

Software

Various search algorithms have been used for FIB lookup. While well-known general-purpose data structures were first used, such as hash tables, specialized algorithms, optimized for IP addresses, emerged. They include:

- Binary tree
- Radix tree
- Four-way trie
- Patricia tree[3]

A multicore CPU architecture is commonly used to implement high-performance networking systems. These platforms facilitate the use of a software architecture in which the high-performance packet processing is performed within a fast path environment on dedicated cores, in order to maximize system throughput. A run-to-completion model minimizes OS overhead and latency[4]

Hardware

Various forms of fast RAM and, eventually, basic content addressable memory (CAM) were used to speed lookup. CAM, while useful in layer 2 switches that needed to look up a relatively small number of fixed-length MAC addresses, had limited utility with IP addresses having variable-length routing prefixes (see Classless Inter-Domain Routing). Ternary CAM (CAM), while expensive, lends itself to variable-length prefix lookups.[5]

One of the challenges of forwarder lookup design is to minimize the amount of specialized memory needed, and, increasingly, to minimize the power consumed by memory.[6]

Distributed forwarding

A next step in speeding routers was to have a specialized forwarding processor separate from the main processor. There was still a single path, but forwarding no longer had to compete with control in a single processor. The fast routing processor typically had a small FIB, with hardware memory (e.g., static random access memory (SRAM)) faster and more expensive than the FIB in main memory. Main memory was generally dynamic random access memory (DRAM).

Early distributed forwarding

Next, routers began to have multiple forwarding elements, that communicated through a high-speed **shared bus**[7] or through a **shared memory**.[8] Cisco used shared busses until they saturated, while Juniper preferred shared memory.[9]

Each forwarding element had its own FIB. See, for example, the Versatile Interface Processor on the Cisco 7500[10]

Eventually, the shared resource became a bottleneck, with the limit of shared bus speed being roughly 2 million packets per second (Mpps). Crossbar fabrics broke through this bottleneck.

Shared paths become bottlenecks

As forwarding bandwidth increased, even with the elimination of cache miss overhead, the shared paths limited throughput. While a router might have 16 forwarding engines, if there was a single bus, only one packet transfer at a time was possible. There were some special cases where a forwarding engine might find that the output interface was one of the logical or physical interfaces present on the forwarder card, such that the packet flow was totally inside the forwarder. It was often easier, however, even in this special case, to send the packet out the bus and receive it from the bus.

While some designs experimented with multiple shared buses, the eventual approach was to adapt the crossbar switch model from telephone switches, in which every forwarding engine had a hardware path to every other forwarding engine. With a small number of forwarding engines, crossbar forwarding fabrics are practical and efficient for high-performance routing. There are multistage designs for crossbar systems, such as Clos networks.

See also

- Network processor
- Network Search Engine

References

1. Forwarding and Control Element Separation (ForCES) Framework (ftp://ftp.rfc-editor.org/in-notes/rfc3746.txt), RFC 3746, Network Working Group, April 2004
2. Methodology for Network Interconnect Devices (http://www.ietf.org/rfc/rfc2544.txtBenchmarking), RFC 2544, S. Bradner & J. McQuade, March 1999
3. Routing on Longest Matching Prefixes (http://portal.acm.org/citation.cfm?id=227248.227256), ID, W. Doeringer 'et al.', IEEE/ACM Transactions on Networking, February 1996
4. "6WINDGate Software Modules" (http://www.6wind.com/products-services/6windgate-software). *6WIND*. Retrieved 14 August 2015.
5. Efficient Mapping of Range Classifier into Ternary-CAM (http://www.hoti.org/archive/hoti10/program/Liu_tcam.pdf), IEEE Symposium on High-Speed Interconnects, H. Liu, August 2002
6. Reducing TCAM Power Consumption and Increasing Throughput (http://www.hoti.org/archive/hoti10/program/Sharma_power.pdf), IEEE Symposium on High-Speed Interconnects, R Panigrahy & S. Sharma, August 2002
7. High Performance IP Forwarding Using Host Interface Peering (http://www.isi.edu/touch/pubs/isi-lanman98abs.pdf), J. Touch *et al.*, Proc. 9th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), May 1998
8. Shared Memory Multiprocessor Architectures for Software IP Routers (http://www.cs.ucr.edu/~bhuyan/papers/tpds.ps), Y. Luo *et al.*, IEEE Transactions on Parallel and Distributed Systems, 2003

9. Juniper Networks Router Architecture (http://www.informit.com/articles/article.aspx?p=30631), *Juniper Networks Reference Guide: JUNOS Routing, Configuration, and Architecture*, T. Thomas, Addison-Wesley Professional, 2003
10. Hardware Architecture of the Cisco 7500 Router (http://safari.ciscopress.com/1578701813/ch06), *Inside Cisco IOS Software Architecture (CCIE Professional Development)*, V. Bollapragada *et al.*, Cisco Press, 2000

Retrieved from "https://en.wikipedia.org/w/index.php?title=Forwarding_plane&oldid=774979398"

This page was last edited on 11 April 2017, at 21:25.