

Table of Contents

<u>Tracing a Packet From Network Ingress to Egress, or "The Life of a Packet"</u>	1
<u>Document ID: 13713</u>	1
<u>Introduction</u>	1
<u>Prerequisites</u>	1
<u>Requirements</u>	1
<u>Components Used</u>	1
<u>Conventions</u>	1
<u>The Process</u>	1
<u>Related Information</u>	3

Tracing a Packet From Network Ingress to Egress, or "The Life of a Packet"

Document ID: 13713

Introduction

Prerequisites

Requirements

Components Used

Conventions

The Process

Related Information

Introduction

This document describes the life of a packet.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Conventions

For more information on document conventions, refer to Cisco Technical Tips Conventions.

The Process

Cisco low-end routers include the 1xxx, 25xx, 26xx, 3600, 3800, 4000, 4500, and 4700 series. With these routers, a message is sent on a wire by someone and received by a controller in another system. That controller, on most systems, stores the packet directly into buffer memory. When the message is fully received, the controller stores some pointer information so the ingress driver can find it, then presents a receive interrupt.

Note: If the controller has no buffers into which to store the message, it logs an "ignore" at this point and doesn't receive the message.

At some point in the future, usually measured in microseconds, a driver becomes active. The driver first checks the transmit queue and, later, checks the receive queue. For now, we will focus on receive queue

processing. The driver checks the necessary aspects of the queue, decides that there is a message, removes the message from the queue, and replenishes the controller's receive buffer list. It then decides to what forwarder, such as IP or Internet Packet Exchange (IPX), to hand the message.

For this explanation, assume it is an IP message. The driver will want to hand the IP message to the configured IP Fast Path. It first removes the link layer header from the message and then determines whether a fast path is configured on the ingress interface. If none is configured, the packet is placed (marked) in the "input hold queue" and a counter is inspected. If the counter is zero, the "input hold queue" is exhausted and the packet is dropped. If the counter is not zero, it is decremented and the message is enqueued to the process path.

Note: The "input hold queue" is not a queue in the literal sense. It is a set of packets that has been received on an interface and is not fully processed (either by forwarding the message to an egress interface or by releasing the buffer). However, if there is a configured fast path, and there usually is, the message is handed to the fast path.

The fast path now validates the message and applies some set of features to the unrouted message. This step includes decrypting or decompressing, or both (if necessary), performing Network Addresses Translation (NAT), applying input committed access rate (CAR) tests, applying policy routing tests, and so on.

Policy routing, if used, actually selects the egress interface. If policy routing is not used, the next step is to look up the destination address in the route cache, a process called "switching the packet." Depending on the fast switching mode, the structure and content of that cache varies. In standard fast switching, the cache contains either destination prefix routes or destination host routes that have been recently used, and occasionally does not have a cache entry. In this case, the message is bumped to process level, again marked as being in the "input hold queue." In Cisco Express Forwarding switching, the cache (called a Forwarding Information Base, or FIB) is a complete route table, so that process does not occur.

If a route is found, the route cache entry (or "adjacency," as it is called in Cisco Express Forwarding) indicates the output software, and hardware interface, and the header to place on the message (next relevant hop). For multiplexed interfaces, this would be the next relevant virtual circuit or virtual channel.

On that software interface, there are a number of issues that might apply. For example, the interface might have been configured with a Maximum Transmission Unit (MTU) smaller than the size of the message. Since there is no fragment in the fast path, this particular event would be a reason to "bump" to process level. Additionally, the interface might be configured for NAT processing, output CAR, and so on. At this point in the processing of the message, those features would be applied. Finally, the output interface's link layer header is affixed to the message, and it is handed to the output driver.

The message is presented to the fast-send routine in the driver which does the following steps.

1. The driver asks a number of questions, such as "Do I have to copy the message to a new buffer before transmitting it?"
2. It determines whether traffic shaping is active. If traffic shaping is active, it compares the message arrival rate to the message transmission rate for the indicated class of messages. If either a shaping queue is forming on the subinterface or the queue is not present but the rate has now been exceeded, it places the message into a queue on the software interface.
3. If traffic shaping is not active, does not apply to this message, or the rate has not been exceeded, the driver now asks whether the depth of the output controller's transmit queue is below the tx-queue-limit. If it is below the limit, the driver simply queues the message for transmission. A message that followed this path is accounted as having been fast-switched on input and fast-switched on output.

4. If it cannot be fast-switched, however, the driver diverts the message into a software queue, which is generically referred to as the "output hold queue". Examples of such hold queues include First In, First Out (FIFO) queuing, Priority Queuing, Custom Queuing, and Weighted Fair Queuing (WFQ).

Since this is the destination of messages that follow the process path as well, such messages are accounted as having been fast-switched on input and process-switched on output. Note that they were not, in fact, process-switched. The switching decision was made when the packet was switched in the fast path. However, the message was diverted to a queue, which is shared with the process path. It is accounted as being process-switched at a later time when the message is removed from the hold queue and enqueued to the transmitting controller.

Process switching is what happens when the message cannot be transmitted in the fast path. It means that the message was sent to this system and will eventually (ideally) be consumed by a routing process, a link maintenance process, a network management process, and so on. However, some traffic indeed goes via the process path, such as traffic that is using Link Fragmentation and Interleaving (LFI) to interleave voice between segments of a jumbogram, X.25 traffic, traffic requiring fragmentation, and traffic for which there was no fast path route entry. The processing on the process path is conceptually identical to the fast path, but differs in implementation for various reasons. One of the differences is that on output, the "input hold queue" flag is cleared and the counter on the input interface incremented (removing the message from the input hold queue), and the message is enqueued into the output hold queue. Then, an interrupt is simulated, which may trigger the transmission of the message on the output interface. It is slower than fast switching because of the process overhead; there may be some other process running when such a message is received, and there are some more complex data structures that have to be dealt with.

Related Information

- [IP Routed Protocols Support Page](#)
- [IP Routing Support Page](#)
- [Technical Support – Cisco Systems](#)

All contents are Copyright © 1992–2005 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Aug 08, 2005

Document ID: 13713
