The Network Layer

Chapter 4 from the Kurose and Ross textbook
The network layer

- The transport layer is responsible for application to application transport.
- The network layer is responsible for host to host transport.
- Unlike the transport and applications layers, there is a piece of the network layer in each host and router in the network.

Data and control planes

- The network layer can be decomposed into two interacting parts.
- The data plane controls how a datagram arriving at a router is forwarded.
- The control plane controls how a datagram is routed among routers along an end-to-one path from source to destination.
Router forwarding tables

- A router forwards a packet by examining the arriving packet’s header field to index into its forwarding table.

- The result indicates to which of the router’s link interface the packet is to be sent.*

*All well and good, but where did the forwarding table come from in the first place.

Traditionally, …

- … a routing algorithm is run in every router.

- The routing algorithm in one router communicate with those in other routers using a routing protocol.

- Well, at least until recently.
Software-defined networking (SDN)

- Software running in a separate remote controller computes and distributes forwarding tables to each router.

- The data-plane components are unchanged, but the control-plane functionally is physically separated from the router.

Network service models

- The network service model defines the characteristics of the end-to-end transport of data.

  - Guaranteed delivery?
  - Guaranteed delivery with bounded delay?
  - In-order packet delivery?
  - Minimal bandwidth?
  - Maximum jitter*?
  - Security?
## Service models to choose from

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Guarantees?</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bandwidth</td>
<td>Loss</td>
<td>Order</td>
</tr>
<tr>
<td>Internet</td>
<td>best effort*</td>
<td>none</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR**</td>
<td>constant rate</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR***</td>
<td>minimum</td>
<td>no</td>
</tr>
</tbody>
</table>

*Best-effort service appears to be a euphemism for no service at all.
**Constant Bit Rate – goal is to look like a dedicated copper or fiber connection.
***Available Bit Rate – best characterized as slightly-better-than-best-effort.

---

## What’s inside a router?

With a look at queuing algorithms
What’s inside a router?

Routing, management control plane (software)

Forwarding data plane (hardware)

Routing processor

*Today we focus mostly on the data plane. We’ll start the control plane next time.

Router architecture

At physical layer terminates incoming link

At link layer it does protocol decapsulation & at the network layer it does forwarding

Connects inputs to proper outputs

Reverse functions of input port
Traditionally:
Role of the router’s local processor

At physical layer terminates incoming link
At link layer it does protocol decapsulation & at the network layer it does forwarding
Connects inputs to proper outputs
Executed routing protocols to maintain forwarding tables
Reverse functions of input port

SDN:
Role of router’s local processor

At physical layer terminates incoming link
At link layer it does protocol decapsulation & at the network layer it does forwarding
Connects inputs to proper outputs
Communicates with remote controller to receive and install forwarding table entries. Performs network management functions
Reverse functions of input port
Data plane: Hardware implementation

A router’s input ports, output ports, and switching fabric are almost always implemented in hardware

While the router’s control functions are typically implemented in software

Input ports lookup & forwarding

- The forwarding table is either computed and updated by the routing processor or is received form a remote SDN controller, then copied to the line cards.

- With shadow copies at each line card, the input port simply does a linear search of the table for the longest network prefix match.
Life should be so simple

The problem is that the input port should be able to proceed at line speed.

For a 10 Gbps link with packets 64 bytes long, the input port has only 51 ns to process the datagram.

Not only that …

- Routing tables must be humongous
  - At one entry per possible destination each router would have billions and billions of entries.

- Table lookup and maintenance would more than a challenge.

- Fortunately, the Internet has a better way.
### Organize addressing into ranges

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010000 00000000 through 00010111 00010000 00000000</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000 00000000 through 00010111 00011000 00000000</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011001 00000000 through 00010111 00011001 00000000</td>
<td>2</td>
</tr>
<tr>
<td>11001000 00010111 00011111 00000000 through 00010111 00011111 00000000</td>
<td>3</td>
</tr>
</tbody>
</table>

### Then, match the longest prefix

- **Longest prefix match:**

<table>
<thead>
<tr>
<th>Prefix match</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 000111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

- For either technique to work, Internet addresses need to be assigned in contiguous blocks.
- Fragmentation rears its ugly head.
Typically,

- Ternary content addressable memories (TCAMs) are used for lookup. A 32-bit IP address is presented to memory, which returns the content of the forwarding table entry for that address in essentially constant time.

- The Cisco 6500 and 7600 series routers can hold upwards of a million TCAM forwarding table entries.

However it is done, ...

... we will probably need wait in line to get our boarding pass.
We have our ticket, now to find the gate

- Okay, we've made through the input queue. How do we find our way to the correct output port?

- How we get there is the job of the switching fabric.

Switching in memory*

- The simplest method is to put switching between input and output ports under the direct control of the routing processor.
- Many modern routers do just that, but the processors are located in the input cards. This looks a lot like shared memory multiprocessing.

*Input and output ports function as traditional I/O devices in the usual OS.
Switching via a bus

- With bus bandwidths over a gigabit per second, switching via a bus is often sufficient for routers that operate in access and enterprise networks.

- But, bus bandwidth is a rate limiter.

Switching via an interconnection network*

- A more sophisticated interconnection network may overcome the bandwidth limitation of a single, shared bus.

*Similar to the technique used in the past to interconnect processors in a multiprocessor computer architecture.
Finally through the switching network we reach the output port

- Output port provides data link protocol processing and line termination that interacts with the input port on the other end of the outgoing line.

But the real story is in the queuing buffer required when switch fabric bests the output link rate.

Queuing at the router
Output port problems

• We assume that all input and output line speeds are identical and that there are n input ports and n output ports.

• If the switching fabric speed is at least n time as fast as the input line speed, then no queuing can occur at the input ports.
• But what happens at the output ports?

A line forms at the output port

At the output queue a packet scheduler must choose one packet among those queued for transmission. May be FIFO or a some form of weighted fair queuing.
• If the switch fabric is not fast enough to transfer all packets through the fabric without delay, then packet queuing will also occur at the input ports.

• An input queue can grow to unbounded length under certain assumptions as soon as the packet arrival rate reaches only 58% due to something called Head-Of-the_Line (HOL) blocking.

HOL blocking at an input queued switch
Active queue management

• If there is not enough buffer, a decision must be made to,
  – either drop the arriving packet (drop-tail)
  – or remove one or more of the already-queued packets

• It may be advantageous to drop a packet before the buffer is full in order to provide a congestion signal to the sender.

Random Early Detection (RED)

• A weighted average is maintained for the length of the output queue.

• If the average is less than minth, the packet is enqueued.

• If the the queue is full or the average is greater than maxth, the packet is dropped.

• If average is in [minth, maxth] it is dropped with probability that is a function of the average queue length.
Packet scheduling: First In First Out

- Affectionately known as FIFO, this one is an old friend.

- If there is not sufficient buffering space to hold arriving packets, the queue’s packet-discard policy kicks in.

- Assuming sufficient space, packets join the back of the queue, remain in order, and are served when they reach the front.

FIFO in action

![Diagram showing FIFO in action](image)
Packet scheduling: Priority queues

- Packets arriving at the output queue are classified into priority classes.

- Each priority class has its own, usually FIFO, queue.

- The link chooses a packet from the highest nonempty queue.

Priority queues in action
Work-conserving round robin queues

Weighted fair queuing (WFQ)

- Similar to a work-conserving queuing round robin discipline.

- Arriving packets are classified and queued into per-class waiting areas and served in a round robin fashion.

- Unlike round robin, each class may receive a differential amount of service in any interval of time.