Networking — Network layer

Three concepts

• Naming
  • A way to identify the source/destination
  • E.g., house address

• Routing
  • Finding “how to” move towards the destination
  • E.g., which airplane should the stuff go on

• Forwarding
  • Actually “moving” towards the destination
  • E.g., Using airplane/truck/rail
Network layer — Forwarding
Network layer — Forwarding

Let's come up with an approach? Generalize Ethernet ideas?
Network layer — Forwarding
Network layer — Forwarding

Attempt 1: Broadcast
Network layer — Forwarding

Attempt 1: Broadcast

• Send to everybody
Network layer — Forwarding

Attempt 1: Broadcast

- Send to everybody
- Goods
Network layer — Forwarding

Attempt 1: Broadcast

• Send to everybody
• Goods
  • Oh, well, simplicity
Network layer — Forwarding

Attempt 1: Broadcast

• Send to everybody
• Goods
  • Oh, well, simplicity
• Not-so-goods
Network layer — Forwarding

Attempt 1: Broadcast

• Send to everybody
• Goods
  • Oh, well, simplicity
• Not-so-goods
  • Oh, well, everything else
Network layer — Forwarding

Attempt 1: Broadcast

• Send to everybody

• Goods
  • Oh, well, simplicity

• Not-so-goods
  • Oh, well, everything else
  • Bandwidth overheads
Network layer — Forwarding
Network layer — Forwarding

Attempt 2: Time division Multiplexing
Network layer — **Forwarding**

**Attempt 2: Time division Multiplexing**

- Each source-destination pair assigned a time slot
Network layer — Forwarding

Attempt 2: Time division Multiplexing

• Each source-destination pair assigned a time slot
  • Can send data only during that slot
Network layer — Forwarding

Attempt 2: Time division Multiplexing

• Each source-destination pair assigned a time slot
  • Can send data only during that slot

• Goods
Network layer — Forwarding

Attempt 2: Time division Multiplexing

• Each source-destination pair assigned a time slot
  • Can send data only during that slot

• Goods
  • No collisions
Network layer — Forwarding

Attempt 2: Time division Multiplexing

- Each source-destination pair assigned a time slot
  - Can send data only during that slot

- Goods
  - No collisions

- Not-so-goods
Attempt 2: Time division Multiplexing

• Each source-destination pair assigned a time slot
  • Can send data only during that slot

• Goods
  • No collisions

• Not-so-goods
  • Underutilization of resources
Network layer — Forwarding
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

• Each source-destination pair assigned a subset of resources
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

• Each source-destination pair assigned a subset of resources
  • Can use only “assigned” resources (e.g., bandwidth)
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

• Each source-destination pair assigned a subset of resources
  • Can use only “assigned” resources (e.g., bandwidth)

• Goods
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

• Each source-destination pair assigned a subset of resources
  • Can use only “assigned” resources (e.g., bandwidth)

• Goods
  • Predictable performance
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

• Each source-destination pair assigned a subset of resources
  • Can use only “assigned” resources (e.g., bandwidth)

• Goods
  • Predictable performance

• Not-so-goods
Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

• Each source-destination pair assigned a subset of resources
  • Can use only “assigned” resources (e.g., bandwidth)

• Goods
  • Predictable performance

• Not-so-goods
  • Underutilization of resources
Network layer — Forwarding
Network layer — Forwarding

Attempt 2 and 3: Circuit Switching
Attempt 2 and 3: Circuit Switching

- Source establishes connection
Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

• Source establishes connection
  • Resources along the path are reserved
Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

• Source establishes connection
  • Resources along the path are reserved

• Source sends data
Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

• Source establishes connection
  • Resources along the path are reserved

• Source sends data
  • Transmit data using the reserved resources
Attempt 2 and 3: Circuit Switching

• Source establishes connection
  • Resources along the path are reserved

• Source sends data
  • Transmit data using the reserved resources

• Source tears down connection
Network layer — *Forwarding*

**Attempt 2 and 3: Circuit Switching**

- **Source establishes connection**
  - Resources along the path are reserved
- **Source sends data**
  - Transmit data using the reserved resources
- **Source tears down connection**
  - Free resources for others to use
Network layer — Forwarding
Network layer — Forwarding

Circuit Switching
Network layer — Forwarding

Circuit Switching

• Goods:
Circuit Switching

• Goods:
  • Predictable performance
Network layer — Forwarding

Circuit Switching

• Goods:
  • Predictable performance
  • Reliable delivery
Network layer — Forwarding

Circuit Switching

• **Goods:**
  • Predictable performance
  • Reliable delivery
  • Simple forwarding mechanism
Network layer — Forwarding

Circuit Switching

• Goods:
  • Predictable performance
  • Reliable delivery
  • Simple forwarding mechanism

• Not-so-goods
Network layer — Forwarding

Circuit Switching

• Goods:
  • Predictable performance
  • Reliable delivery
  • Simple forwarding mechanism

• Not-so-goods
  • Resource underutilization
Network layer — Forwarding

Circuit Switching

• **Goods:**
  • Predictable performance
  • Reliable delivery
  • Simple forwarding mechanism

• **Not-so-goods**
  • Resource underutilization
  • Blocked connections
Network layer — Forwarding

Circuit Switching

• **Goods:**
  • Predictable performance
  • Reliable delivery
  • Simple forwarding mechanism

• **Not-so-goods**
  • Resource underutilization
  • Blocked connections
  • Connection set up overheads
Network layer — Forwarding

Circuit Switching

• **Goods:**
  • Predictable performance
  • Reliable delivery
  • Simple forwarding mechanism

• **Not-so-goods**
  • Resource underutilization
  • Blocked connections
  • Connection set up overheads
  • Per-connection state in switches (scalability problem)
Network layer — Forwarding
Network layer — Forwarding

Attempt 4: Packet Switching
Network layer — Forwarding

Attempt 4: Packet Switching

• Divide the message into packets
Attempt 4: Packet Switching

- Divide the message into packets
- Put destination address in the header of each packet
Network layer — Forwarding

Attempt 4: Packet Switching

- Divide the message into packets
- Put destination address in the header of each packet
  - Just like shipping stuff
Network layer — Forwarding

Attempt 4: Packet Switching

• Divide the message into packets
• Put destination address in the header of each packet
  • Just like shipping stuff
• Each device stores a “look-up table”
Network layer — Forwarding

Attempt 4: Packet Switching

• Divide the message into packets
• Put destination address in the header of each packet
  • Just like shipping stuff
• Each device stores a “look-up table”
  • What's the next hop towards the destination?
Network layer — Forwarding

Attempt 4: Packet Switching

- Divide the message into packets
- Put destination address in the header of each packet
  - Just like shipping stuff
- Each device stores a "look-up table"
  - What's the next hop towards the destination?
- Destination receives the packet(s)
Network layer — Forwarding

Attempt 4: Packet Switching

• Divide the message into packets
• Put destination address in the header of each packet
  • Just like shipping stuff
• Each device stores a “look-up table”
  • What’s the next hop towards the destination?
• Destination receives the packet(s)
  • And reconstructs the message
Network layer — Forwarding
Network layer — Forwarding

Packet Switched forwarding
Packet Switched forwarding

- Hop-by-hop forwarding
Network layer — Forwarding

Packet Switched forwarding

- Hop-by-hop forwarding
- Each router has a “look-up table” (forwarding information base)
Packet Switched forwarding

- Hop-by-hop forwarding
- Each router has a “look-up table” (forwarding information base)
  - What should be stored in this table?
Network layer — Forwarding

Packet Switched forwarding

• Hop-by-hop forwarding
• Each router has a “look-up table” (forwarding information base)
  • What should be stored in this table?
Packet Switched forwarding

- Hop-by-hop forwarding
- Each router has a “look-up table” (forwarding information base)
  - What should be stored in this table?
  - Prefix-based forwarding (longest-prefix matching)
Network layer — Forwarding

Packet Switched forwarding

• Hop-by-hop forwarding
• Each router has a “look-up table” (forwarding information base)
  • What should be stored in this table?
  • Prefix-based forwarding (*longest-prefix matching*)
  • Maps *prefixes* to the next-hop
Network layer — Forwarding
Network layer — Forwarding

Packet Switching
Network layer — Forwarding

Packet Switching

- Goods:
Network layer — Forwarding

Packet Switching

• Goods:
  • No resource underutilization
Network layer — Forwarding

Packet Switching

• Goods:
  • No resource underutilization
    • A source can send more if others don’t use resources
Network layer — **Forwarding**

**Packet Switching**

- **Goods:**
  - No resource underutilization
  - A source can send more if others don’t use resources
  - No blocked connection problem
Packet Switching

- Goods:
  - No resource underutilization
    - A source can send more if others don’t use resources
  - No blocked connection problem
  - No per-connection state
Network layer — Forwarding

Packet Switching

• Goods:
  • No resource underutilization
    • A source can send more if others don’t use resources
  • No blocked connection problem
  • No per-connection state
  • No set-up cost
Network layer — Forwarding

Packet Switching

• Goods:
  • No resource underutilization
  • A source can send more if others don’t use resources
  • No blocked connection problem
  • No per-connection state
  • No set-up cost

• Not-so-goods:
Network layer — Forwarding

Packet Switching

• Goods:
  • No resource underutilization
    • A source can send more if others don’t use resources
  • No blocked connection problem
  • No per-connection state
  • No set-up cost

• Not-so-goods:
  • Packet header overhead
Network layer — Forwarding

Packet Switching

• Goods:
  • No resource underutilization
    • A source can send more if others don’t use resources
  • No blocked connection problem
  • No per-connection state
  • No set-up cost

• Not-so-goods:
  • Packet header overhead
  • Network failures become a problem
Networking — Network layer

Three concepts

• Naming
  • A way to identify the source/destination
  • E.g., house address

• Routing
  • Finding “how to” move towards the destination
  • E.g., which airplane should the stuff go on

• Forwarding
  • Actually “moving” towards the destination
  • E.g., Using airplane/truck/rail
Network layer — Example
Network layer — Routing
Network layer — Routing

Let's come up with a routing scheme
Network layer — Routing
A wants to find a path to Dest. \{(A, 0)\}
A wants to find a path to Dest. $\{(A, 0), (B, 2)\}$

A wants to find a path to Dest. $\{(A, 0)\}$
A wants to find a path to Dest. \{(A, 0), (B, 2)\}

A wants to find a path to Dest. \{(A, 0), (C, 7)\}

A wants to find a path to Dest. \{(A, 0)\}
A wants to find a path to Dest. 
{(A, 0), (B, 2)}

Path to Dest. 
{(A, 0), (B, 2), (D, 3)}

A wants to find a path to Dest. 
{(A, 0), (C, 7)}
Network layer — Routing

A wants to find a path to Dest. 
{(A, 0), (B, 2)}

Path to Dest. 
{(A, 0), (B, 2), (D, 3)}

A wants to find a path to Dest. 
{(A, 0), (C, 7)}

Path to Dest. 
{(A, 0), (C, 7), (D, 1)}
A wants to find a path to Dest. 
{(A, 0), (B, 2)}

Path to Dest. 
{(A, 0), (B, 2), (D, 3)}

A wants to find a path to Dest. 
{(A, 0), (B, 2), (C, 1)}

Path to Dest. 
{(A, 0), (C, 7), (D, 1)}

A wants to find a path to Dest. 
{(A, 0), (C, 7)}
A wants to find a path to Dest. 
{(A, 0)}

A wants to find a path to Dest. 
{(A, 0), (B, 2)}

Path to Dest. 
{(A, 0), (B, 2), (D, 3)}

A wants to find a path to Dest. 
{(A, 0), (B, 2), (C, 1)}

Path to Dest. 
{(A, 0), (B, 2), (C, 1), (D, 1)}

A wants to find a path to Dest. 
{(A, 0), (B, 2), (C, 7)}

Path to Dest. 
{(A, 0), (C, 7), (D, 1)}
Network layer — Routing
Network layer — Routing

Attempt 1: Dynamic Source Routing
Network layer — Routing

Attempt 1: Dynamic Source Routing

- Broadcast a Route Request Packet for destination d
Network layer — **Routing**

**Attempt 1: Dynamic Source Routing**

- Broadcast a Route Request Packet for destination $d$
  - Put source ID in the packet header
Attempt 1: Dynamic Source Routing

- Broadcast a Route Request Packet for destination d
  - Put source ID in the packet header
- At each router
Network layer — Routing

Attempt 1: Dynamic Source Routing

- Broadcast a Route Request Packet for destination d
  - Put source ID in the packet header
- At each router
  - If a path not known to the destination
Network layer — Routing

Attempt 1: Dynamic Source Routing

• Broadcast a Route Request Packet for destination d
  • Put source ID in the packet header

• At each router
  • If a path not known to the destination
    • Put its {ID, cost} in the packet header
Attempt 1: Dynamic Source Routing

- Broadcast a Route Request Packet for destination d
  - Put source ID in the packet header

- At each router
  - If a path not known to the destination
    - Put its \{ID, cost\} in the packet header
    - Broadcast the Route Request Packet
Network layer — Routing

Attempt 1: Dynamic Source Routing

- Broadcast a Route Request Packet for destination $d$
  - Put source ID in the packet header

- At each router
  - If a path not known to the destination
    - Put its {ID, cost} in the packet header
    - Broadcast the Route Request Packet
  - Else
Attempt 1: Dynamic Source Routing

- Broadcast a Route Request Packet for destination d
  - Put source ID in the packet header

- At each router
  - If a path not known to the destination
    - Put its {ID, cost} in the packet header
    - Broadcast the Route Request Packet
  - Else
    - Respond with a Route Reply packet
Attempt 1: Dynamic Source Routing

• Broadcast a Route Request Packet for destination d
  • Put source ID in the packet header

• At each router
  • If a path not known to the destination
    • Put its {ID, cost} in the packet header
    • Broadcast the Route Request Packet
  • Else
    • Respond with a Route Reply packet
    • Put known path in the packet header
Network layer — Routing

Attempt 1: Dynamic Source Routing

• Broadcast a Route Request Packet for destination d
  • Put source ID in the packet header

• At each router
  • If a path not known to the destination
    • Put its {ID, cost} in the packet header
    • Broadcast the Route Request Packet
  • Else
    • Respond with a Route Reply packet
    • Put known path in the packet header

• Challenge?
Network layer — Routing
Network layer — Routing

{(A-B, 0), (A-C, 7)}
Network layer — Routing

\{(A-B, 2), (B-C, 1), (B-D, 3)\}

\{(A-B, 0), (A-C, 7)\}
Network layer — Routing

{[(A-B, 2), (B-C, 1), (B-D, 3)]

{(A-B, 0), (A-C, 7)}

{(A-C, 7), (B-C, 1)}

Dest.
Network layer — Routing

\{(A-B, 2), (B-C, 1), (B-D, 3)\}

\{(A-B, 0), (A-C, 7)\}

\{(A-C, 7), (B-C, 1)\}

\{(B-D, 3), (C-D, 1)\}
Network layer — Routing

- \{(A-B, 0), (A-C, 7)\}
- \{(A-B, 2), (B-C, 1), (B-D, 3)\}
- \{(A-C, 7), (B-C, 1)\}
- \{(B-D, 3), (C-D, 1)\}
Network layer — Routing

{(A-B, 0), (A-C, 7)}

{(A-B, 2), (B-C, 1), (B-D, 3)}

{(B-D, 3), (C-D, 1)}

{(A-C, 7), (B-C, 1)}

{(A-B, 0), (A-C, 7)}

{(B-D, 3), (C-D, 1)}
Network layer — Routing

{((A-B, 0), (A-C, 7))}

{((A-B, 2), (B-C, 1), (B-D, 3))}

{((B-D, 3), (C-D, 1))}

{(A-C, 7), (B-C, 1)}

{(A-C, 7), (B-C, 1)}

{(B-D, 3), (C-D, 1)}

{(A-B, 0), (A-C, 7)}

{(A-C, 7), (B-C, 1)}
Network layer — Routing
Network layer — Routing

Attempt 2: Link State Routing
Network layer — Routing

Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
Network layer — Routing

Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
• Each router periodically “floods” its LS
Network layer — Routing

Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
• Each router periodically “floods” its LS
  • And forwards all the LS received from other routers
Network layer — Routing

Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
• Each router periodically “floods” its LS
  • And forwards all the LS received from other routers
• At one point
Network layer — Routing

Attempt 2: Link State Routing

- Each router maintains its local “link state” (LS)
- Each router periodically “floods” its LS
  - And forwards all the LS received from other routers
- At one point
  - Every router knows the entire topology
Network layer — Routing

Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
• Each router periodically “floods” its LS
  • And forwards all the LS received from other routers
• At one point
  • Every router knows the entire topology
• Run a shortest path algorithm (e.g., Dijkstra) locally
Network layer — Routing

Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
• Each router periodically “floods” its LS
  • And forwards all the LS received from other routers
• At one point
  • Every router knows the entire topology
• Run a shortest path algorithm (e.g., Dijkstra) locally
  • Find path to the destination
Attempt 2: Link State Routing

• Each router maintains its local “link state” (LS)
• Each router periodically “floods” its LS
  • And forwards all the LS received from other routers
• At one point
  • Every router knows the entire topology
• Run a shortest path algorithm (e.g., Dijkstra) locally
  • Find path to the destination
  • More importantly, find next-hop to the destination
Network layer — Routing

Attempt 2: Link State Routing

- Each router maintains its local “link state” (LS)
- Each router periodically “floods” its LS
  - And forwards all the LS received from other routers
- At one point
  - Every router knows the entire topology
- Run a shortest path algorithm (e.g., Dijkstra) locally
  - Find path to the destination
  - More importantly, find next-hop to the destination
- Challenge?
Network layer — Routing
Network layer — **Routing**

**Attempt 3: Distance Vector Routing**
Network layer — Routing

Attempt 3: Distance Vector Routing

• Each router
Network layer — Routing

Attempt 3: Distance Vector Routing

• Each router
  • maintains its “current distance to destination”
Network layer — Routing

Attempt 3: Distance Vector Routing

- Each router
  - maintains its “current distance to destination”
  - Periodically announces it to all its neighbors
Network layer — **Routing**

**Attempt 3: Distance Vector Routing**

- Each router
  - maintains its “current distance to destination”
  - Periodically announces it to all its neighbors
  - Update its local table
Network layer — Routing

Attempt 3: Distance Vector Routing

• Each router
  • maintains its “current distance to destination”
  • Periodically announces it to all its neighbors
  • Update its local table
    • \[d(A, \text{dest}) = \min\{d(A, \text{neighbor}) + d(\text{neighbor, dest})\}\]
Network layer — Routing

Attempt 3: Distance Vector Routing

• Each router
  • maintains its “current distance to destination”
  • Periodically announces it to all its neighbors
  • Update its local table
    • \( d(A, \text{dest}) = \min\{d(A, \text{neighbor}) + d(\text{neighbor}, \text{dest})\} \)
    • \{\text{dest} — distance, \text{neighbor}-that-minimizes-distance\}
Attempt 3: Distance Vector Routing

• Each router
  • maintains its “current distance to destination”
  • Periodically announces it to all its neighbors
  • Update its local table
    • \[ d(A, \text{dest}) = \min\{d(A, \text{neighbor}) + d(\text{neighbor}, \text{dest})\} \]
    • \{\text{dest} — \text{distance, neighbor-that-minimizes-distance}\}
  • Broadcast to all its neighbors