Switching forms and terms

Outline

History of switching
Switching forms
Packet and circuit switching
Variants of packet and circuit switching
A survey of definitions
Switch classifications
Marketing terminology: Bridges, switches routers
A survey of definitions

History of switching

1876 Bell is first to patent the telephone
1892 Strowger automated telephone switch
1937 Reeves invents Pulse Coded Modulation
1950s Research into switching networks (Clos, Batcher, etc)
early Packet switching invented by Baran, Davies & Kleinrock
1966 Bell System introduces the IESS (Electronic Switching System)
1973 Metcalfe invents Ethernet
1970s Internet & optical fibre transmission systems developed
1976 X.25 recommendation for public data networks
1978 OSI Reference Model
1982 Bell System introduces 5ESS switch
1984 Cisco (dominant router vendor) founded
1988+ ATM
Late MPLS, diffserv, photonic networks, peer-to-peer
1990s file-sharing systems
Examples of switching

- Data communications, integrated networks
- Telephone network
  - Gave rise to Clos networks, SS7 signalling, etc
- Interconnection networks for parallel processors
  - Strong parallels with structured space-division networks (e.g. Banyan)
- Sources of accessible analogies to help understand networking:
  - Vehicular traffic – railway switching yards, automotive traffic
  - Irrigation systems → fluid flow models & Hurst parameter
  - Utility networks (water, sewage, electricity, gas ...)

Outline

- Spectrum of switching forms
- Circuit switching
- Packet switching
- More state information → more dedicated resources
- Fast packet switching
- Soft-state flows
- Datagrams
- Virtual circuits
- Switched cell switching
- Permanent virtual circuits

Spectrum of switching forms

Another depiction

Unknown image source, probably referring to
M. De Prycker: Asynchronous Transfer Mode: Solution for Broadband
Packet vs circuit switching

**Packet switching**
- Bits of data to be switched are organised into discrete units of information, called "packets".
- Organising data into discrete units helps explain the name “packet” switching, but isn’t the distinguishing feature of packet switching, which is...
- Switching decisions are influenced by the content of the packets (e.g. whether particular bits have a value of 0 or 1).

**Circuit switching**
- Switching decisions are predetermined and are not influenced by the content of the information being switched.
- Examples of predetermining factors: port, frequency, time of arrival.
  - e.g. info from port $P_1$, frequency $F_1$ should go to output port $P_2$, frequency $F_2$

Simplistic analogies: postal service is packet switched; analog telephones are circuit switched

Comments on circuit switching

- Circuit switching can operate on discrete units of information, e.g.:
  - Time Division Multiplexing: “Slots” of time (corresponding to multiple bits) are switched in a predetermined manner.
  - Information carried over a circuit may be “packetised” for efficient reliable transfer
- “Circuit switching decisions are predetermined”:
  - How? Often by packet switching! (See “Establishing a circuit”)
  - For how long?
    - *Usually* until transfer ends
  - But some systems allow adjustment during transfer
- Deterministic ⇒ dedicated: Only one source can use predetermined access opportunities. Waste opportunity if not needed by that source.

Implications of circuit switching

Switch behaves predictably, independently of information being switched.

Circuit switching tends to (correlations, not requirements):
- Have little buffering within the network.
- Operate in a connection-oriented manner (more later)
- Assign a fixed capacity to the circuit for the duration of the transfer (rather than re-negotiating).
  ⇒ Source must be able to anticipate its requirements.
- Offer coarse granularity of transmission rates.
- Preserve sequence of transmitted information

Renaissance of circuit switching

1950s: Electromechanical circuit switching was popular for telephone circuits.
1960s: Electronic circuit switching for telephone circuits.
1990s: Volume of data sent across packet-switched networks increases exponentially over time
  ⇒ Photonic networks to provide higher speed links
  Such high speed that efficiency is no longer critical
  ⇒ Circuit switching is again viable, but in optical, rather than electromechanical, form.
  Circuit switching is well-suited to photonic networks: Buffering and processing (interpreting headers) are difficult in the photonic domain.
Establishing a circuit

Q: How is a switched programmed to know how to switch information (e.g. port/frequency/time mapping of input to output)?

A1: Source uses partial circuit spanning part-way to destination to signal to next switch. That switch hunts for an output that leads towards the destination.

  e.g. classic pulse-switched telephone: First digit controls first switch, last digit controls last switch.

A2: Switching elements may be connected by a packet switched network. Circuit setup request propagates to switches through packet switched network, and when all switches are ready, circuit switching can commence.

  i.e. packet switching may be used for managing (establishing & releasing) circuits.

  e.g. SS7, ATM etc

Timing of circuit establishment

Using a partial circuit

Circuit usage over time

X1 hunts for link leading to D
X2 hunts for link leading to D
(Feedback to source)

Data transfer

Using a packet-switched network

Circuit usage over time

Packet-switched setup makes tentative reservations
Setup and returns full link to confirm reservations
Data transfer

Separate setup and transfer phases ⇒ notion of “connection-oriented”

Outline

Packet switching

Motivation 1: Statistical multiplexing

Varieties

  Possible packet header information

  Datagram switching

  Source routing

  [State information]

  Virtual Circuit Switching

  Label swapping

Multiplexing example

Offered load: 3 sources with total load of 8 units/second

4/sec

1/sec

3/sec

Separate circuits: 4 units per second each (capacity of 12 units/sec)

4/sec

1/sec

3/sec

Multiplexed on one channel (capacity of 8 units/sec)

4 units/second

When arrivals are deterministic, we can time-division multiplex in a circuit-switched manner. But arrivals are likely to be bursty...
Statistical multiplexing

Bursty network traffic ⇒
- Dedicated channel may often be wasted
  ⇒ circuit switching may be inefficient
    (may not be important for optical links)
- Non-dedicated channel may occasionally be overloaded
  ⇒ Packet-switches need buffering and may drop packets
due to congestion.

Outline

Possible packet header information

Address of end-system: Allows packets to be self-contained “datagrams”.
- “best effort” service = “I’ll do my best but make no guarantees”
- “best effort” ? better than other service.
  Prototypical examples: Ethernet, Internet Protocol.

Addresses of intermediate systems = “source routing”

Labels/tags: Distinguish this flow of packets from others, e.g. IPv6
  FlowID, ATM VCI/VPI
  - Switch consults table as to what to do with the flow.
  - Switches can swap tags ⇒ label on a packet may change as it
    propagates.

Datagram Switching

No connection setup phase
Each packet forwarded independently
Sometimes called connectionless model
Analogy: postal system
Each switch maintains a forwarding (routing) table

[Based on material from L. Peterson]
Internet Protocol packet *headers*

version 4:

```
+---------------+---------------+---------------+---------------+---------------+
<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
+---------------+---------------+---------------+---------------+
| Identification | Flags | Fragment Offset |
| 0         | 0   |                |
+---------------+---------------+---------------+---------------+
| Time to Live | Protocol | Header Checksum |
| 0         |        |                |
+---------------+---------------+---------------+---------------+
| Source Address | Destination Address |
| 0         | 0      |
+---------------+---------------+---------------+---------------+
| Options | Padding |
| 0         | 0      |
+---------------+---------------+---------------+---------------+
```

version 6:

```
+-----+---------- +-------+--------+--------- +-------+--------- +------------- +-----------+
| Ver | Diffserv | Flow  | Length | NextHdr | Hop  | Source  | Destination | (Options) |
|     |          | label |        |         | Limit | Address | Address    |           |
| 4b  |    8b    |  20b  |   16b  |    8b   |   8b  |   128b  | 128b    |           |
+-----+---------- +-------+--------+--------- +-------+--------- +------------- +-----------+
```

Outline

Source Routing

Packet header provides incremental “directions” to reach destination, rather than globally identifying destination.

Examples of directions:
- Which port to leave each switch.
- Identifiers that are specific to the switch (e.g. north, south, east, west) (as per example on next slide). Identifiers can be small.
- Globally unique identifier (as per IP)
- Which switch to pass through.
- Identifiers must be globally unique ⇒ large.
- Useful when multiple ports connect one switch to another.
- Control of port helps load balancing and dealing with line faults.

Example from L. Peterson
Types of source routing

**Loose source routing**: Must pass through specified addresses, but can pass through others, e.g. want it to pass through a particular node to measure network access from that node.

**Strict source routing**: Must only pass through specified addresses e.g. don’t pass classified military info through enemy networks.

These terms aren’t important. They are raised because the widespread Internet Protocol offers these as separate services, and hence applications (e.g. traceroute) may mention them, but source routing is rarely used.

Switch’s role in source routing

Each switch needs to determine which part of the directions relate to it. Arriving packets may have propagated different distances ⇒ can’t rely on field in fixed position.

Several options:

- **Pop off own address** (no longer needed). Rarely used, since often prefer to keep list of addresses s.t. destination can reply along reverse route.
- **Rotate header** s.t. instructions for next switch are always at front (and instructions for previous switch at end). (as per example)
- **Header contains a pointer** to the instructions for the next switch, and switches increment this pointer. (as per IP)

Source Routing: Evaluation

- Switches can be simple and fast
- Sources can influence routing, e.g. on basis of price, security, etc
  - Source routing decision is made at one point (source); may not have current local information available in distributed routing, e.g. datagram.
  - Only really supports unicast. Difficult to support multicast & anycast.

Outline
State information

State information: "memory in the system used to influence future behaviour" [Keshav, p. 108]
- Switch examples: route, type of service (e.g. delay requirements), profile of permissible flow, what resources are assigned to this flow.
- End-system examples: what has been acknowledged (for reliable transfer)

Soft and hard state

Soft state vs hard state
- Hard state is explicitly established and released
- Soft state is established on demand (e.g. when a packet arrives) and released after timeout (if not refreshed before then)

State management

Connection-oriented communication: 3 phases:
1. Explicitly establish state information
2. Transmit
3. Release state information

c.f. connectionless communication

Signalling for switching

Signalling (signaling): The process of managing state information, e.g. establish & release connection

Tends to be implemented over a packet-switched or broadcast/select network

Examples:
- PSTN uses packet-switched SS7 signalling network
- ATM broadcasts initial signalling message on specified VPI/VCI
- optical circuit-switched network may have ancillary electronic packet-switched network for signalling.
Benefits of state information

- **Performance**: Perform complicated functionality (e.g. route determination) once during setup, then quick reference to state during transfer.
  
  Switches are often differentiated from routers by virtue of having more state information, presumably operating faster.

- **Preserve sequence**: Simplifies end-systems

- **Guarantees**: Reserve resources to guarantee “Quality of Service” to particular connections. If resources aren’t available, then the call “blocks”.

Costs of state information

- **Complexity**
- **Delay during setup**
- **Robustness**: In state lies fate…

Recovery of network state

Recovery of state information inside the network is theoretically possible, e.g. problem of switch in midst of network failing & neighbouring switches reconfiguring around it.

But:
- Is complicated due to monitoring, clearing established state etc (and the more complicated the system, the more likely it is to fail in the first place)
- Needs to be achieved rapidly if outage is to be masked from users.

Bottom line: Possible, but so complicated and slow that it is rarely implemented. (exception: SS7)
Virtual Circuit Switching

A form of packet switching: Switches inspect headers to determine where (& how) to switch packets.

Unlike datagram switching, headers don’t identify destination, but identify a “connection” (aka “virtual circuit”). The “label” within the header used for identification is often called a “Virtual Circuit Identifier”.

Switches have state information (e.g. a table indexed by VCI) installed indicating where (& how) to switch packets belonging to established connections.

State information must be explicitly established (e.g. using datagrams) before communication can proceed ⇒ connection oriented.

† Called “Virtual Channels” or “Virtual Paths” in the Asynchronous Transfer Mode

VCs compared to datagrams

May be more efficient than datagram switching:

- Complex processing (e.g. to decide optimal route) need only be done once, at connection setup.
- Identifiers need only be unique locally, not globally:
  - No complicated global coordination of identifier usage
  - Smaller headers saves bandwidth
- Connection setup overhead
- Reliability depends on switches maintaining state information

While this figure shows one table per switch, most switches implementations divide the content of this table into one table per port. This reduces the size of what has to be looked up (the port is implicit), simplifying the “packet classification” (see later lecture). It also facilitates parallelism, allowing packet classification for different ports to proceed concurrently, without the need to access a table shared between ports.

† Based on material from L. Peterson

Virtual Circuits

Virtually a “circuit” because:

- All packets follow the same path ⇒ maintain sequence (like a circuit)
- Setup phase may also reserve resources for the connection ⇒ service guarantees (like a circuit)

But not a real circuit because:

- Switches must inspect data, and
- Reservation is possible, but it is not necessary:
  - e.g. VC may reserve space in switch forwarding tables, but not reserve buffer space or link bandwidth
Label swapping

Tables in each switch:

<table>
<thead>
<tr>
<th>W</th>
<th>1</th>
<th>E</th>
<th>2</th>
</tr>
</thead>
</table>

If a packet comes in from the west with a label of 1, send it east with a label of 2.

Each packet sent over a connection has the same label. But the packet label may change as it propagates through the network.

Asynchronous Transfer Mode (ATM)

Later, we’ll spend a whole lecture on ATM, but in a nutshell:

- ATM was designed (in the 90s) to provide a “Broadband Integrated Services Digital Network” (B-ISDN).
- Integrated Services ⇒ need reservations to provide guarantees ⇒ used virtual circuits.
- Used short packets to limit voice packetisation delay ⇒ fixed-length packets called “cells” (5B header + 48B payload).
- Simple packet classification (VCI indexes a lookup table).
- ATM’s downfall was difficult interworking with connectionless IP systems. MPLS applies many ATM principles to IP.

Permanent Virtual Circuits (PVCs) vs Switched Virtual Circuits (SVCs)

Both identify virtual circuits (indeed, traffic for both is switched)

Difference is in how the VC is managed:

**PVCs**:
- Expected to have long life (e.g. months)
- e.g. used to provide guaranteed connectivity between physically separated offices (like leased line)
- Often established/released manually
- Stored in durable memory in switches (e.g. flash) s.t. VC is re-established each time the switch boots

**SVCs**:
- Established and released on demand from users.
- Signalling is complicated ⇒ SVC services may appear later than PVC services (as was the case with Frame Relay & ATM).
- Stored in RAM. If switch suffers an outage, users will lose connectivity through VC and re-establish another VC if desired. Other VC may be through other switches (established while original switch is still out) or through same switch (when switch returns online)

Spectrum of switching forms

| Is it affected by the content being switched? | No | Yes |
| Is a physical channel dedicated to the flow? | Yes | No |
| Are packets self-contained? | Yes | Perhaps |
| Is the transmission transparent? | Yes | No |
| Are packets of fixed size? | Yes | Perhaps |
| Who determines the route? | Source switches | Other switches |
| Is a physical channel dedicated to the flow? | Yes | No |
| Are packets self-contained? | Yes | Perhaps |
| Is the transmission transparent? | Yes | No |
| Are packets of fixed size? | Yes | Perhaps |
| Who determines the route? | Other switches | Source switches |
Defining ‘circuit’ and ‘packet’ switching

Many sources dodge the issue of defining circuit and packet switching, and instead give examples of each mode, i.e. they describe properties (consequences), rather than characteristics that are root causes.

Main themes that recur throughout definitions:

- **Circuit**: 
  - Dedicated/reserved resources
  - Set up before transfer
- **Packet**: 
  - Meta-data/addresses
  - Self-contained
  - Discrete blocks
  - Buffering in switches

Keshav [K]

p. 159: “packet switches ... switch packets that contain both data and descriptive meta-data. In contrast, telephone switches are circuit switches that switch voice samples that contain no meta-data.”

Not bad, but doesn’t define what meta-data is, or what it will be used for.

p. 160: “Each packet is self-contained in that it contains sufficient information to be routed to its destination.”

Strictly, “packet” here should be “datagram”

The important feature of packet-switched networks is that the nodes in such a network send discrete blocks of data to each other.

Packet-switched networks typically use a strategy called store-and-forward. In contrast, a circuit-switched network first establishes a dedicated circuit across a sequence of links and then allows the source node to send a stream of bits across this circuit to a destination node.

A counter-argument is that Time-Slot Interchange circuit switches also send discrete blocks of data and store the information before forwarding it. "Typically" suggests exceptions, which should exist for a clean definition.

In circuit-switched networks, the resources needed along a path (buffers, link transmission rate) to provide for communication between end systems are reserved for the duration of the communication session. In packet-switched networks, these resources are not reserved; a session’s messages use the resources on demand, and as a consequence, may have to wait (that is, queue) for access to a communication link.

Dedicated resources.

Connection-oriented circuit switching originated in the public telephone network. Early telephone networks dedicated a physical circuit for an electrical signal.

“Baran and his research team ... articulated the concept of packet switching ... The solution was to segment a longer message into many smaller pieces and then wrap routing and protocol information around each of these pieces, resulting in a string of data called a packet.”

Connection-oriented circuit switching originated in the public telephone network. Early telephone networks dedicated a physical circuit for an electrical signal.
circuit switching: 1. A method of routing traffic through a switching center, from local users or from other switching centers, whereby a connection is established between the calling and called stations until the connection is released by the called or calling station. 2. A process that, on demand, connects two or more data terminal equipments (DTEs) and permits the exclusive use of a data circuit between them until the connection is released.
def 1: = connection-oriented
def 2: “exclusive use”

packet switching: The process of routing and transferring data by means of addressed packets so that a channel is occupied during the transmission of the packet only, and upon completion of the transmission the channel is made available for the transfer of other traffic.

[http://www.atis.org/tg2k/]