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
1960s
1965  Bell System introduces the 1ESS (Electronic Switching System)
1973  Metcalfe invents Ethernet
1970s Internet & optical fibre transmission systems developed
History of switching (continued)

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, sewerage, electricity, gas ...)

Figure 1-8 from A. Tanenbaum and M. v. Steen: 'Distributed Systems: Principles and Paradigms', 2002
Copyright © 2005 Tim Moors
Outline
Spectrum of switching forms

Switching

Circuit
- actual circuits
- circuit emulation

Packet
- extent of state information
  - Fast packet switching
  - Soft-state flows
- Datagrams
  - source
  - transparent
    - routing
      - strict
      - loose

Virtual circuits
- permanent
- switched

Cell switching

More state information
- More dedicated resources

Copyright © 2005 Tim Moors
Another depiction

Figure 4.2  Spectrum of Switching Techniques (based on [PRYC93])

Unknown image source, probably referring to
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

† Packet switching may also be influenced by other predetermined factors, e.g. a router discarding packets that have a source address that is invalid for the port from which they arrived.
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/“private”: 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.

✅ Good for providing guaranteed service

✖ Bad for efficient link use

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 predict 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 msg returns from dest to confirm reservations
Data transfer

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

Packet switching
  Overview
  Possible packet header information
Datagram switching
  Source routing
  [State information]
Virtual Circuit Switching
  Label swapping
Packet switching: Overview

Characteristics:

• Payload is segmented into discrete units of information, called packets *(not a distinguishing feature).*
• Each packet contains a header, indicating how switches should process it.

Some benefits: Header makes data unit more self-contained (independent of timing, possibly independent of other data units)

• Packets can arrive asynchronously ⇒ statistical multiplexing ⇒ efficient link utilisation
• May be routed independently of other packets belonging to the same flow ⇒ robustness
Possible packet header information†

**Address of end-system**: Allows packets to be self-contained “datagrams”.

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 an internal table to determine what to do with packets from the flow.

† for routing. Headers may contain other info to indicate type of service, and for other functions.
Datagram Switching

No connection setup phase

“best-effort service”

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:

```
+---------------------------------------------+
| Version | IHL | Type of Service | Total Length |
+---------------------------------------------+
+---------------------------------------------+
|       | Flags | Fragment Offset |
+---------------------------------------------+
| Time to Live | Protocol | Header Checksum |
+---------------------------------------------+
| Source Address |
+---------------------------------------------+
| Destination Address |
+---------------------------------------------+
| Options | Padding |
+---------------------------------------------+
```

## version 6:

```
<table>
<thead>
<tr>
<th>Ver</th>
<th>Diffserv</th>
<th>Flow label</th>
<th>Length</th>
<th>NextHdr</th>
<th>Hop Limit</th>
<th>Source Address</th>
<th>Destination Address</th>
<th>(Options)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4b</td>
<td>8b</td>
<td>20b</td>
<td>16b</td>
<td>8b</td>
<td>8b</td>
<td>128b</td>
<td>128b</td>
<td></td>
</tr>
</tbody>
</table>
+---------------------------------------------+
```
Outline
Source Routing

A way of using datagrams in which the source determines the route, e.g. rather than switches/routers.

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

Examples of directions:
• Which \textit{port} to leave each switch.
  • Identifier that is specific to the switch (e.g. north, south, east, west) (as per coming examples). Identifiers can be small.
  • Globally unique identifier (as per IP)
• Which \textit{switch} to pass through.
  • Identifiers must be globally unique $\Rightarrow$ large.
  • Useful when multiple ports connect one switch to another.

Switch determination of port helps load balancing and dealing with line faults.

‡ or, in the case of loose source routing, \textit{helps determine}
Source routing with address pointers

Packet format:
- next addr ptr, # of addresses, address list, data (not shown)
- e.g. 1,3,EES = next switch should examine 1st of 3 addresses (East, East, South), and then increment next addr ptr

In real life (IP), directions = address of interface on next router
Source routing with address rotation

Packet format:
# of addresses, address list, data (not shown)
e.g. 3,NSE = next switch should forward north, and rotate address list to SEN
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:

• Header contains a pointer to the instructions for the next switch, and switches increment this pointer. (as per IP)
• Rotate header s.t. instructions for next switch are always at front (and instructions for previous switch at end). (as per example)
• 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.
Source Routing: Advantages

- Switches can be simple and fast

- Path is determined at one point (source), so can work when switches have inconsistent views of the network topology (e.g. while topology updates are propagating).

- Sources can influence routing, e.g. on basis of price, security, etc, e.g.
  - MPLS
  - UNSW/NICTA research project on routing with Byzantine robustness

- No need for switches to proactively learn routes for various destinations. Important for ad hoc networks, where learning consumes energy. May prefer that source learns route on-demand, e.g. Dynamic Source Routing (DSR)

- Can be used for network troubleshooting, e.g.
  1. User complains to helpdesk that they can’t communicate with a target
  2. Helpdesk needs to check what happens to packets sent from user’s computer to a target
  3. Helpdesk can use loose source routing to force packet from helpdesk to target through user’s computer
Source Routing: Disadvantages

- Source routing decision is made at one point (source);
  - may not have current local information available in distributed routing, e.g. datagram.
  - can’t reroute packet after it has been launched

- Requires extra room in packet header

- Only really supports unicast.
  Difficult to support multicast & anycast.

- Can’t implement policy-based routing
  - Where a “subnet” (part of the Internet) may only be willing to carry particular traffic (e.g. DA≠competitor)
  - “Subnet” may not publicise policies (as needed for sources)
  - “Subnet” can’t rely on sources to abide by policies.
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**: The exchange of information for the purpose of managing a flow of information. In particular, to establish and release connections.

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.

† Often spelled “signaling” by Americans
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)
Outline
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 (VCI)”†.

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.

“virtually a circuit: almost entirely like a circuit, but not a circuit.
† Called “Virtual Channels” or “Virtual Paths” in the Asynchronous Transfer Mode
VC e.g. 1: Fixed VCI (“label switching”)

Tables in each switch:

If a packet comes in with a label of 1, send it east.

VCI space must distinguish all connections through the network ⇒ reasonably large.

While this figure shows one table per switch, most switches implementations divide the content of this table into one table per port. This facilitates parallelism, allowing packet classification for different ports to proceed concurrently, without the need to access a table shared between ports.
VC e.g. 2: VCI/label *swapping*

Tables in each switch:

\[
\begin{array}{c|c|c}
\text{N} & \text{in} & \text{out} \\
\text{W} & \text{E} & \text{E} & \text{W} \\
\text{S} & \text{E} & \text{W} & \text{W} \\
\end{array}
\]

*†* label

\[\begin{array}{c|c|c}
\text{W} & \text{1} & \text{E} & \text{2} \\
\end{array}\]

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

Packet label is *swapped*

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

✓ Labels need only be unique on each link ⇒ can be smaller

Label swapping is a particular type of (VC) label switching used in ATM, MPLS
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 (⇒ packet switching), 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 transmission capacity
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 transmission capacity
  ✓ Simple (fast) classification by table lookup
 ✗ Connection setup overhead
 ✗ Reliability depends on switches maintaining state information
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

• Broadband ⇒ operate at high speed ⇒ simple processing ⇒
  • 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 \(\Rightarrow\) 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)
Confusion about “Virtual Circuits”

The term “virtual circuit”

- Is usually used in the context of the link or network layer, refers to the passage through the network
- Is sometimes used in the context of higher layers (e.g. transport,) to refer to an end-to-end connection that has circuit-like properties (e.g. delivers information in sequence, possibly even reliable, e.g. as provided by TCP but not by UDP.

e.g.

1. D. Clark: The Design Philosophy of the DARPA Internet Protocols
   “The traditional type of service is the bi-directional reliable delivery of data. This service, which is sometimes called a "virtual circuit" service, is appropriate for such applications as remote login or file transfer.”

2. Domain Name System (DNS): Virtual Circuit means use TCP (instead of UDP) to communicate to server.
   e.g. Microsoft’s nslookup program has a commend “set [no] vc” to control whether DNS lookups “always use a virtual circuit”
Spectrum of switching forms

Switching
- Is it affected by the content being switched?
  - No
  - Yes

Circuit
- Is a physical channel dedicated to the flow?
  - Yes
  - No: logical part is dedicated

Packet
- extent of state information
  - Are packets self-contained?
    - No
    - Yes

Datagrams
- Fast packet switching
  - Are packets of fixed size?
    - Perhaps
    - Yes

Soft-state flows
- Who determines the route?
  - Source
  - Switches

Virtual circuits
- permanent
  - more state information

more dedicated resources
- switched

Cell switching
- strict
- loose
References in Keshav

Datagrams and virtual circuits, pp. 48-53, 175-6
State information pp.108-9
Outline

A survey of definitions of circuit and packet switching
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: [K] [P&D] [K&R] [T] [M] [TG]
  • 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.

Kurose and Ross [K&R]

pp. 14-5: “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.

Tanenbaum [T]  
4th edition, p. 147-50

Doesn’t define circuit switching, but describes examples of circuit

“When you ... places a telephone call, the switching equipment within the telephone system seeks out a physical path all the way from your telephone to the receiver’s telephone. This technique is called circuit switching”

“The [circuit switching] model shown in ... is highly simplified, of courses, because parts of the physical path between two telephones may, in fact, be microwave or fiber links onto which thousands of calls are multiplexed. Nevertheless, the basic idea is valid: once a call has been set up, a dedicated path between both ends exists and will continue to exist until the call is finished.”

Packet switching: “the alternative to circuit switching is packet switching ... With this technology, individual packets are sent as need be, with no dedicated path being set up in advance. It is up to each packet to find its way to the destination on its own.”

Comments:

• Path set up: Virtual Circuits are not real circuits (i.e. are packet switched) and also require a path to be set up before sending data.
• Dedicated resources: VCs can also dedicate resources to particular sources.
• “on its own”: Alluding to datagrams
McDysan [M]

“Connection-oriented circuit switching originated in the public telephone network. Early telephone networks dedicated a physical circuit for an electrical signal” p. 4

“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.”

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/]