Circuit switching in the Internet

Ph.D. oral examination
Pablo Molinero-Fernández
21st May 2002

Traffic doubles every year

Optics removes bandwidth constraints
But we cannot buffer light!

Circuit switching
- Link bandwidth is reserved
- Need signaling
- No buffering
- No processing in data path

Packet switching
- Link bandwidth is shared
- Buffering
- Per-packet processing

Contributions
- The Internet needs circuit switching in the core
- TCP Switching: how to integrate circuit switching in the core
- Provisioning of fat pipes in an all-optical backbone
How we think the Internet is

Why the Internet uses packet switching
- Efficient use of expensive links:
  - “Circuit switching is rarely used for data networks, … because of very inefficient use of the links” - Bertsekas & Gallager ’92
- Resilience to failure of links & routers:
  - “For high reliability, … [the Internet] was to be a datagram subnet, so if some lines and [routers] were destroyed, messages could be … rerouted” - Tanenbaum ’96

What the Internet is really like today

SONET/SDH DWDM

What the Internet is really like today

Modems, DSL

Performance criteria

Users
- Response time
- Service Level Agreement (QoS)

Carriers
- Cost: Bandwidth efficiency
- Reliability and stability
- Low complexity

What users want

Users
- Response time
- Service Level Agreement (QoS)

Carriers
- Cost: Bandwidth efficiency
- Reliability and stability
- Low complexity
### Response time of packets and circuits

<table>
<thead>
<tr>
<th>Circuit sw</th>
<th>Packet sw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow bandwidth</td>
<td>1 Gb/s</td>
</tr>
<tr>
<td>Avg response time</td>
<td>0.51 sec</td>
</tr>
<tr>
<td>Max response time</td>
<td>1 sec</td>
</tr>
</tbody>
</table>

All but one of circuits finish earlier.

### Response time when blocking occurs

A big flow can kill CS if it blocks the link.

### Response time with flow rate limits

<table>
<thead>
<tr>
<th>Circuit sw</th>
<th>Packet sw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow bandwidth</td>
<td>1 Mb/s</td>
</tr>
<tr>
<td>Avg response time</td>
<td>109.9 ms</td>
</tr>
<tr>
<td>Max response time</td>
<td>10,000 sec</td>
</tr>
</tbody>
</table>

No difference between CS and PS in core.

### Analytical modeling

- Fluid model:
- Many independent arrivals $\Rightarrow$ Poisson
- Service policies:
  - Packets: Processor Sharing
  - Circuits: FCFS
- Service time distribution:
  - Flow size variance: Bimodal
  - Realistic flow size distrib: Pareto

### Fluid model: M/BiModal/N

- Flow rate limited by access link (1/N)
- Some response time regardless of flow size variance, if N is large
Fluid model: M/Pareto/N

Link hogging becomes very bad with heavy-tailed traffic, if ratio N=1

**Users see little difference in response time**

- Simulation of full networks
- N is large => same response time

**Service Level Agreements**

- Packet switching:
  - Algorithms (WFQ, DRR, ...) => not used
  - Thus we must overprovisioning => used and it works
- Circuit switching:
  - Simple QoS: guaranteed BW => no jitter

**What carriers want**

- Response time
- Service Level Agreement (QoS)
- Cost: Bandwidth efficiency
- Reliability and stability
- Low complexity

**Cost: Bandwidth efficiency**

- Argument: packets share all link BW => statistical multiplexing gain => more throughput with bursty traffic
- Reality:
  - Internet avg. link utilization: 5-20% [Coffman&Odyzyko02]
  - Phone avg. link utilization: ~33% (Odyzyko99)
  - There is a glut of BW in the core [WSS’00]
- Result:
  - Packets more efficient, but BW is no longer a scarce resource
Reliability and stability (I)

- **Argument:** because of the state, rerouting a circuit is more costly than with packets
- **Reality:**
  - Internet average availability: 1220 min/year down time [Labovitz’99]
  - Phone average availability: 5 min/year down time [Kuhn’97]

Reliability and stability (II)

- **Reality (cont.):**
  - IP recovers in about 3 min (median), sometimes it takes over 15 min [Labovitz’01]
  - SONET required to recover in less than 50 ms
- **Result:**
  - No evidence packet switching is more robust

Low complexity (I)

- **Argument:** No per-flow state => packet switching is simpler
- **Reality:**
  - PS: 8M lines of code in core router [Cisco’s IOS ’00]
  - CS: 18M lines of code in telephone switch [AT&T/Lucent SESS ’96]
  - CS: 3M lines of code in transport switch [’01]
- **Result:**
  - Packet switching does not seem inherently less complex than circuit switching

Functions in a packet switch

Functions in a circuit switch
Low complexity (II)

- **Argument:** IP does not have the signaling of circuits switches => Routers go faster
- **Reality:**
  - IP does almost same operations on every packet as a circuit switch on the circuit establishment
  - CS has no work to do once circuit is established
- **Result:**
  - The fastest commercially-available circuit switches [Ciena '01, Lucent '01] have 5x the capacity of the fastest routers [Cisco '01, Juniper '02]

Network architecture

- Use packet switching
- Better response time (ratio N small)
- Efficient use of the spectrum
- If metro-to-access BW ratio (N) is small => use packets
- Otherwise use what costs less
- Use circuit switching
- More capacity, reliability
- Similar response time & QoS

Contributions

- The Internet needs circuit switching in the core
- TCP Switching: how to integrate circuit switching in the core
- Provisioning of fat pipes in an all-optical backbone

Integration of circuits and packets

- Create a separate circuit for each user flow
- IP controls circuits
- Optimize for the most common case
  - TCP (90-95% of traffic)
  - Data (9 out of 10 pkts)

TCP Switching exposes circuits to IP

TCP "creates" a connection

*Source* Router Router Router *Destination*

**SYN**

**DATA**

**Packets**

**Packets**

**Packets**

**Packets**

**SYN-ACK**
Let TCP leave state behind

- Source
- Boundary TCP-SW
- Core TCP-SW
- Boundary TCP-SW
- Destination

Create circuit
Create circuit
SYN-ACK

Packets
One Circuit
Packets

What is a typical flow?

- Most traffic are TCP connections:
  - Taking less than 10 s, 12 packets and 4 KBytes
  - Obtaining less than 100 Kbps
  - ~40% of the flows continue transmitting ACKs after sending a FIN (asymmetrical closures)

State management feasibility

- Amount of state
  - Minimum circuit = 56 Kb/s.
  - 178,000 circuits for OC-192.
- Update rate
  - About 51,000 entries per sec for OC-192
- Implementable in hardware or software.

TCP Switching can be implemented in software

TCP Switching boundary router:

- Kernel module in Linux 2.4 1GHz PC
- Forwarding latency
  - Forward one packet: 21μs.
  - Compare to: 17μs for IP.
  - Compare to: 95μs for IP + QoS.
- Time to create new circuit: 57μs.

Source: Byung-Gon Chun ’01

Bandwidth inefficiencies

- IP Switching
  - Uses ATM virtual circuits (i.e. packets)
  - Became MPLS (but no longer user flows)
- Generalized Multi-Protocol Label Switching (GMPLS)
  - Coarse circuits
  - Heavy weight signaling

Related work
Contributions

- The Internet needs circuit switching in the core
- TCP Switching: how to integrate circuit switching in the core
- Provisioning of fat pipes in an all-optical backbone

New networking scenario

- IP routers
- Optical Switches

New networking scenario

- IP Linecards
- Optical Crossconnect

Circuit creation is slow

- We need a safeguard to avoid running out of BW => inefficiency
  - A slow signaling requires a larger BW safeguard

Controlling coarse circuits with user flows

- Should use the fastest optical switching elements
- Should avoid ACKs => no RTT

Conclusions

- Circuits should be used in the core, packets in the edges
- TCP Switching integrates circuits and packets in an evolutionary way
- User flows can be used to control an all-optical network
Papers

- PMF, Nick McKeown, "TCP Switching: Exposing Circuits to IP", IEEE Micro, 2002
- PMF, NM, "TCP Switching: Exposing Circuits to IP", Hot Interconnects, 2001
- PMF, NM, "Study of routing behavior through traffic analysis and traceroute measurements", ANET Times, 2001
- PMF, NM, Hui Zhang, "Is IP going to take over the world (of communications)?", submitted