Evolving Internet

Walid Dabbous & Chadi Barakat
INRIA Centre de Sophia-Antipolis Méditerranée
http://planete.inria.fr

General context

The Internet has been able to withstand rapid growth fairly well and its core protocols have been robust enough to accommodate numerous applications that were unforeseen by the original Internet designers.

How does this global network infrastructure work and what are the design principles on which it is based? In what ways are these design principles compromised in practice? How do we make it work better in today's world? How do we ensure that it will work well in the future in the face of future demands? What are the new protocols and services that have been proposed to enhance the Internet architecture? What are the tools and techniques to understand what is going on? These are some questions that we will grapple with in this course. The course will provide knowledge on these hot topics for both research and industrial interest.

Objectives & Content

To understand the state-of-the-art in network architecture, protocols, and networked systems and to study in depth some of the up-to-date networking problems.

Each course will consist in a lecture followed by somes exercices.

Content

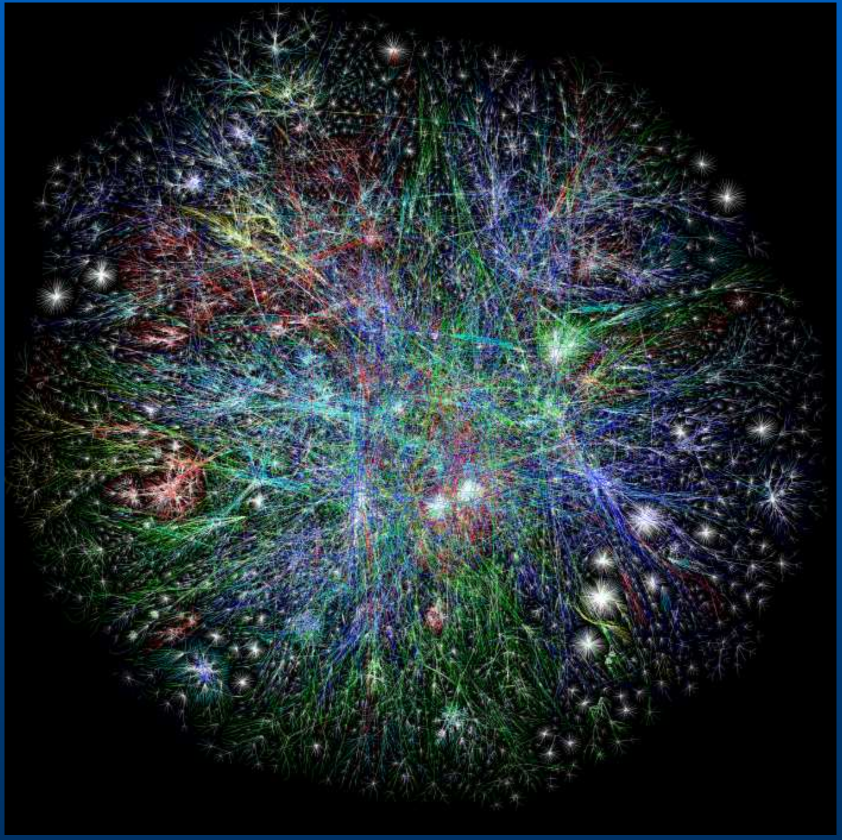
- Historical introduction (WD)
- MAC & link layer (WD)
- Addressing and Routing (WD)
  - Interdomain routing
- Mobility support (CB)
- Transmission control (CB)
  - Congestion control
- Quality of Service support (WD)
- Passive and active measurments (CB)

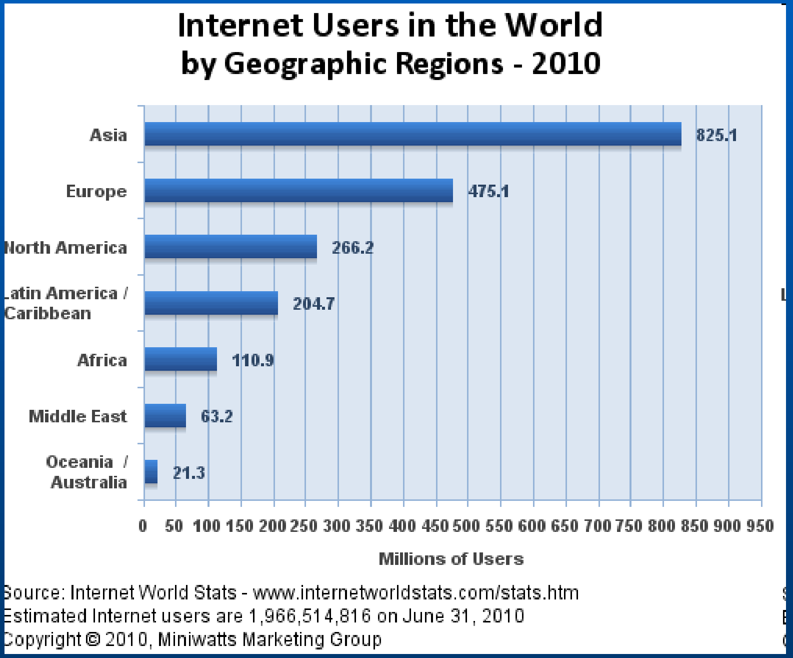
Reference books

- Computer Networks A systems approach, by Larry L. Peterson and Bruce S. Davie, (2007), ISBN-10: 0123705487, ISBN-13: 9780123705488.
- An Engineering Approach to Computer Networking, S. Keshav, Addison-Wesley, May 1997, 688 pages, ISBN 0-201-63442-2
- Routing in the Internet, C. Huitema, Prentice-Hall, 1995, 319 pages, ISBN 0-13-132192-7
- Computer Networking, A Top-Down Approach Featuring the Internet, J. Kurose, K. Ross, Pearson Education, 2001, 712 pages, ISBN 0-201-47711-4
- Computer Networks, Andrew S. Tanenbaum, Prentice Hall International Editions, 3rd edition, March 1996, 814 pages, ISBN 0-13-394248-1

Internet Evolution

- Huge numbers of nodes and users



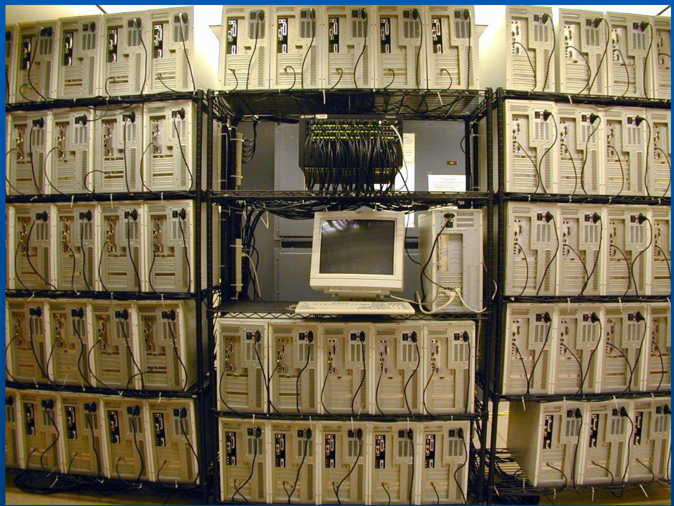


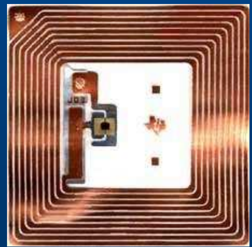
1,966,514,816

June 30, 2010

Internet Evolution

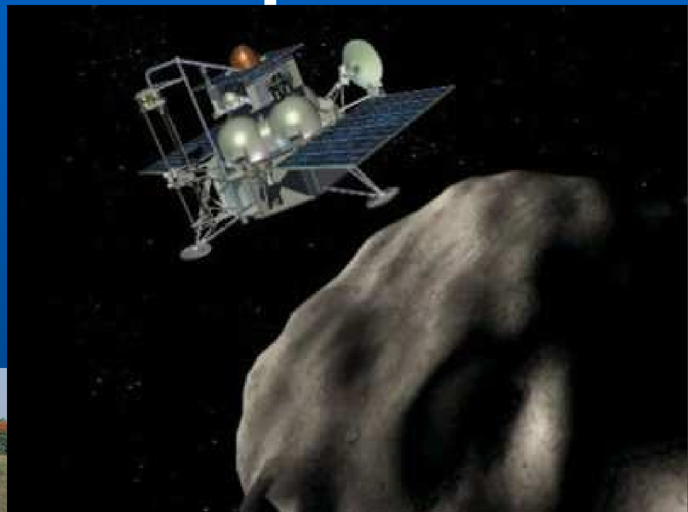
- **Increasing heterogeneity**





Internet Evolution

- **Mobility and episodic connectivity**

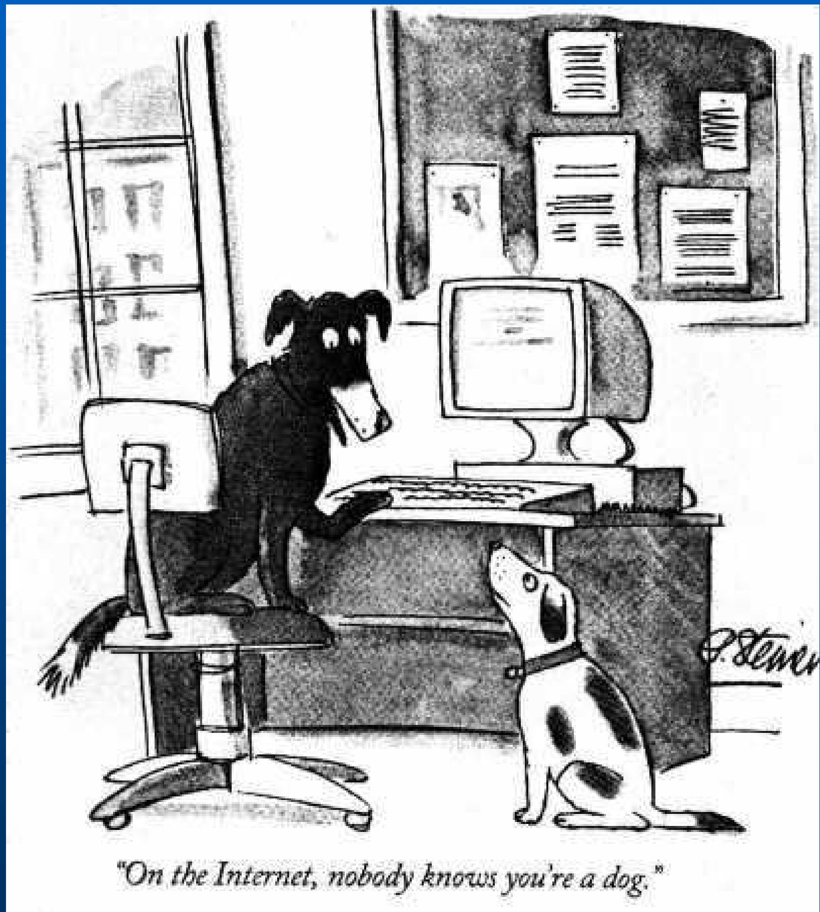


Internet Evolution

- **Dissemination of real time data**

Internet Evolution

- **Security issues**



Wonderful … But

- Ubiquitous wireless
- Devices connectivity
- Wealth of information

- Barely works ! (on a campus)
- Multiple (out of sync) devices
- Information related to hosts on which it resides

Point “patches” for ubiquitous problems

Networking History

- The Phone System
  - Focus on the wires
- The ATM network
  - Asynchronous transmission
- The Internet
  - Focus on the endpoints
- Tomorrow
  - Focus on the data

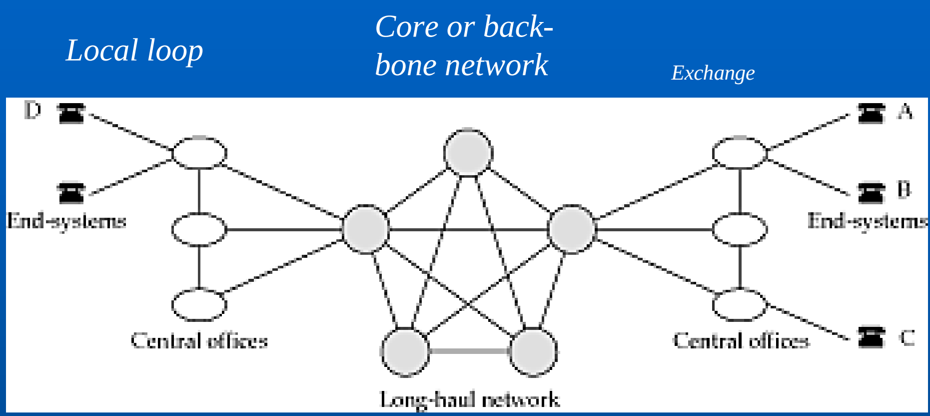
The Phone System

- Connecting wires to other wires
  - Utility depends on running wires to *every* home & office
  - Wires are the dominant cost
  - Revenue comes from path construction
- Not about making "calls"
  - For an operator
    - a call is a "circuit" not a "conversation"
    - a phone number is a <u>program</u> to build the path not the callee *<u>address</u>*
  - Business model based on making revenues from calls that are a "side effect"
  - Revolutionized communications!

Concepts

- Single basic service: two-way voice
  - low end-to-end delay
  - guarantee that an accepted call will run to completion
- Endpoints connected by a *circuit*
  - like an electrical circuit
  - signals flow both ways (*full duplex*)
  - associated with bandwidth and buffer *resources*

The big picture



- (nearly) Fully connected core
  - simple routing
  - hierarchically allocated telephone number space
  - (usually) a telephone number is a hint about how to route a call

The components of a telephone network

1. End systems
2. Transmission
3. Switching
4. Signaling

2. Transmission

- Link characteristics
  - information carrying capacity (bandwidth)
    - information sent as *symbols*
    - 1 symbol ≥ 1 bit (see next course)
  - propagation delay
    - time for electromagnetic signal to reach other end
    - light travels at 0.7c in fiber ~ 5 ms/km
    - Nice to Paris => 5 ms; London to NY => 27 ms ; ~250 ms for earth-sat-earth on GEO satellites
  - attenuation
    - degradation in signal quality with distance
    - long lines need regenerators
    - but recent links need regeneration each 5000 Km and optical amplifiers exist

Transmission: Multiplexing

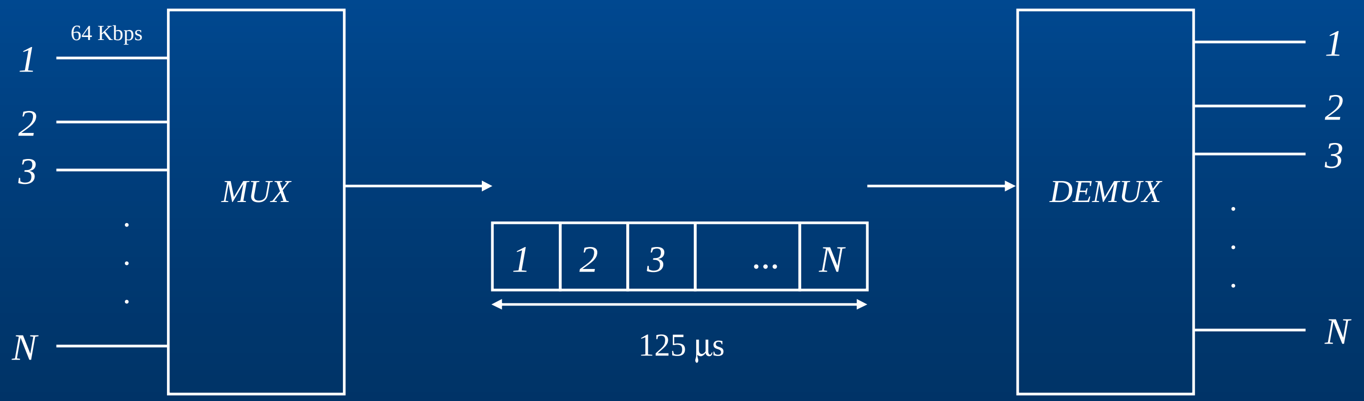
- *Trunks* between central offices carry hundreds of conversations
- Can't run thick bundles!
- Instead, send many calls on the same wire
  - *multiplexing*
- Analog multiplexing (FDM)
  - bandlimit call to 3.4 KHz and frequency shift onto higher bandwidth trunk
  - obsolete, the telephone network is becoming all-digital
- Digital multiplexing
  - first convert voice to *samples*
  - 1 sample = 8 bits of voice
  - 8000 samples/sec => call = 64 Kbps

Transmission: Digital multiplexing

- How to choose a sample?
  - 256 *quantization levels*
    - logarithmically spaced (better resolution at low signal levels)
    - sample value = amplitude of nearest quantization level
  - two choices of quantization levels (μ law (Japan and USA) and A law)
- Time division multiplexing (TDM)
  - (output) trunk carries bits at a faster bit rate than inputs
  - *n* input streams, each with a 1-byte buffer
  - output interleaves samples
  - need to serve all inputs in the time it takes one sample to arrive
  - => output runs *n* times faster than input
  - *overhead* bits mark end of *frame* (synchronize to frame boundary)

Multiplexors and demultiplexors

- Most trunks time division multiplex voice samples
- At a central office, trunk is demultiplexed and distributed to active circuits
- Synchronous multiplexor
  - N input lines (associated with a buffer to store at least one sample)
  - Output runs N times as fast as input



More on multiplexing

- Demultiplexor
  - one input line and N outputs that run N times slower
  - samples are placed in output buffer in round robin order
- Neither multiplexor nor demultiplexor needs addressing information (why?)
  - requires however accurate timing information
- Can cascade multiplexors
  - need a standard
  - example: DS hierarchy in the US and Japan

Digital Signaling hierarchy

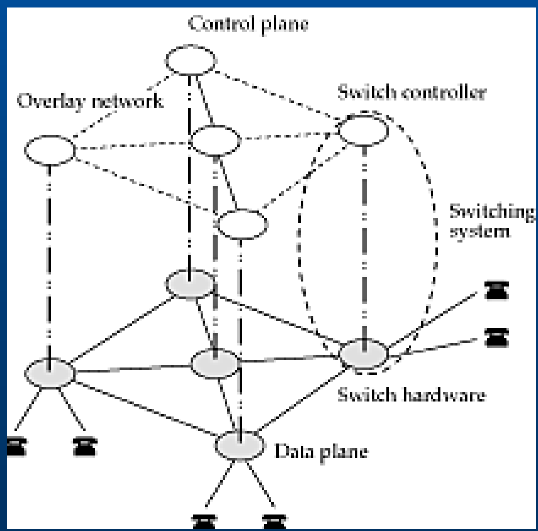
Digital Signal Number	Number of previous level circuits	Number of Voice circuits	Bandwidth
DS0		1	64 Kbps
DS1 - T1	24	24	1.544Mbps
DS2	4	96	6.312 Mbps
DS3 - T3	7	672 = 28 T1	44.736 Mbps

Inverse multiplexing : scatter/gather

- Takes a high bit-rate stream and scatters it across multiple trunks
- At the other end, combines multiple streams
  - resequencing to accommodate variation in delays
- Allows high-speed virtual links using existing technology
  - aggregate telephone channels to connect IP routers

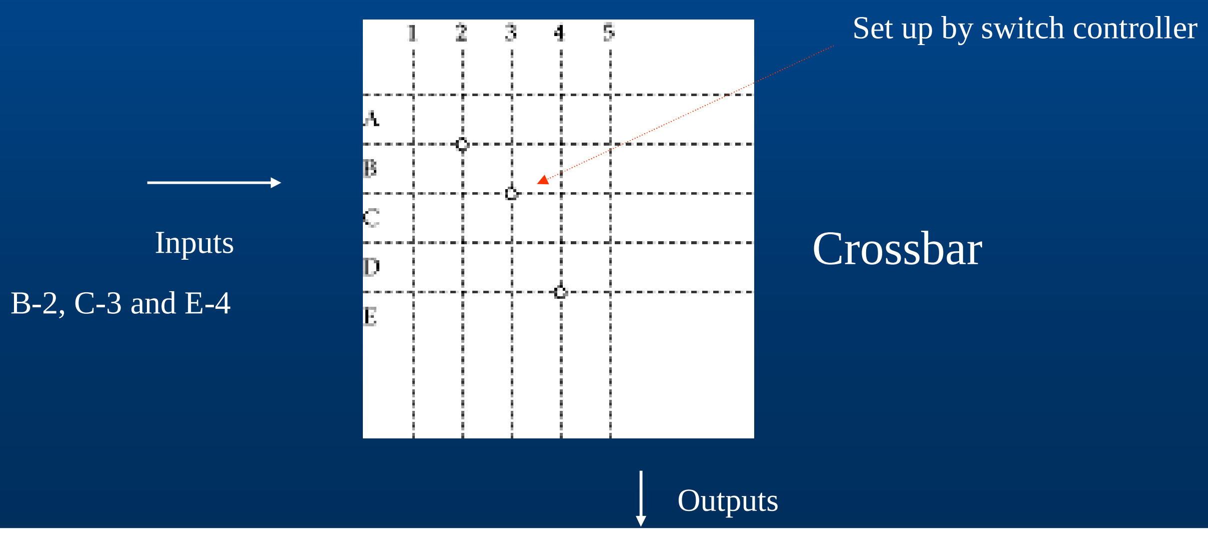
3. Switching

- Problem:
  - each user can potentially call any other user
  - can’t have direct lines!
- Switches establish temporary *circuits*
- Switching systems come in two parts: switch and switch controller



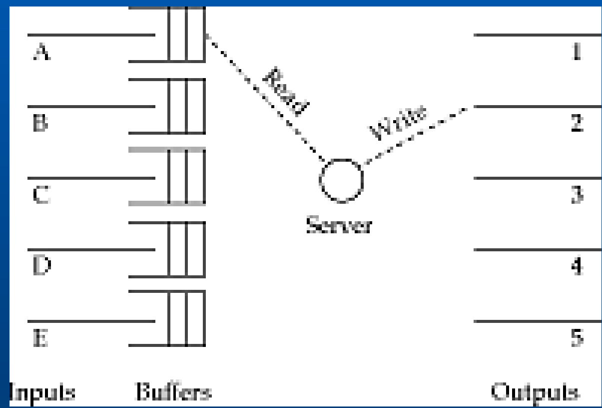
Switching: what does a switch do?

- Transfers data from an input to an output
  - many ports (up to 200,000 simultaneous calls)
  - need high speeds
- Some ways to switch:
  - First way*: space division* (data paths are separated in space)
    - *simplest space division switch is a "crossbar"*
    - if inputs are multiplexed, need a *schedule* (to rearrange crosspoints at each time slot)



Time Division Switching

- Another way to switch
  - *time division (time slot interchange* or *TSI)*
  - also needs (only) a schedule (to write to outputs in correct order)



- Inefficient if long pauses in conversations (idle slots are wasted)
- To build (large) switches we combine space and time division switching elements

Problems with STM

- Problems with STM
  - idle users consume bandwidth (STM is inefficient)
  - Arbitrary schedules result in complicated operation
    - links are shared with a fixed cyclical schedule => quantization of link capacity (corresponds to 64 Kbps circuits in telephone)
    - can’t ‘dial’ bandwidth e.g. 91 Kbps.
  - STM service is inflexible

Better than STM for data?

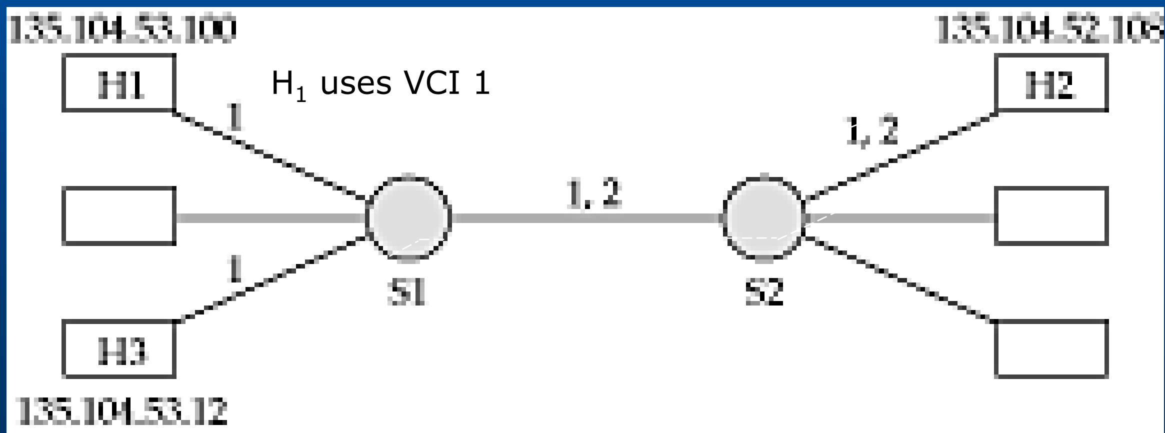
- STM is easy to overcome
  - use *packets* instead
  - meta-data (header) indicates src/dest
    - allows to store packets at switches and forward them when convenient
    - no wasted bandwidth (identify cell by source address not only order in frame) - more *efficient*
    - arbitrary schedule (cells of same source can occur more than once in frame) - more *flexible*
- Two ways to use packets
  - carry only an identifier (The ATM network)
  - carry entire destination address in header (IP)

ATM networks

1. Virtual circuits
2. Fixed-size packets (*cells*)
3. Small packet size
4. Statistical multiplexing
5. Integrated services

Virtual circuits

- Identifiers save on header space
- But need to be pre-established
- We also need to switch Ids at intermediate points
  - VCIs are allocated locally
- Need *translation table* (for VCI swapping) and *connection setup*



Features of virtual circuits

- All packets must follow the same path
  - if any switch along the route fails -> the VC fails
- Switches store per-VC state (entry in translation table)
  - can also store QoS information (priority, reserved bandwidth)
- Call set-up (or signaling) => separation of *data* and *control*
  - control in software over slow time scale, data transfer in hardware
- Virtual circuits do not automatically guarantee reliability
  - possible packet loss
- Small Identifiers can be looked up quickly in hardware
  - harder to do this with IP addresses

More features

- Setup must precede data transfer
  - delays short messages
- Switched vs. Permanent virtual circuits
- Ways to reduce setup latency
  - preallocate a range of VCIs along a path
    - *Virtual Path*
    - *reduces also the size of the translation table*
  - dedicate a VCI to carry datagrams, reassembled at each hop

2. Fixed-size packets

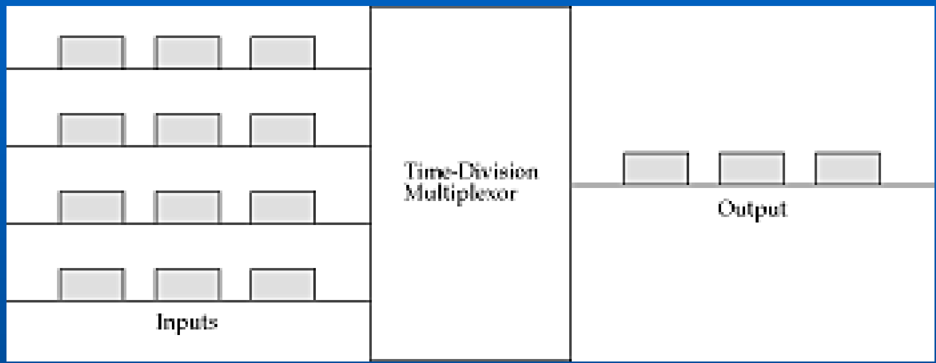
- Pros
  - Simpler buffer hardware
    - packet arrival and departure requires us to manage fixed buffer sizes (easier, no memory fragmentation)
  - Simpler line scheduling
    - each cell takes a constant chunk of bandwidth to transmit -> harder to achieve simple ratios with variable size packets
  - Easier to build large *parallel* packet switches
    - input buffers, parallel switch fabrics, output buffers -> *maximum parallelism if same packet size*
- Cons
  - If the chosen size < ADU => overhead
  - segmentation and reassembly cost
  - last unfilled cell after segmentation wastes bandwidth

3. Small packet size

- At 8KHz, each byte is 125 microseconds
- The smaller the cell, the less an endpoint has to wait to fill it
  - *packetization delay*
- The smaller the packet, the larger the header overhead
- EU and Japan: reduce cell size (32 bytes cell, 4 ms packetization delay)
- US telcos: reduce header cost (existing echo cancellation equipment) (64 bytes cell, 8ms packetization delay)
- Standards body balanced the two to prescribe 48 bytes + 5 byte header = 53 bytes
  - => ATM maximal efficiency of 90.57%

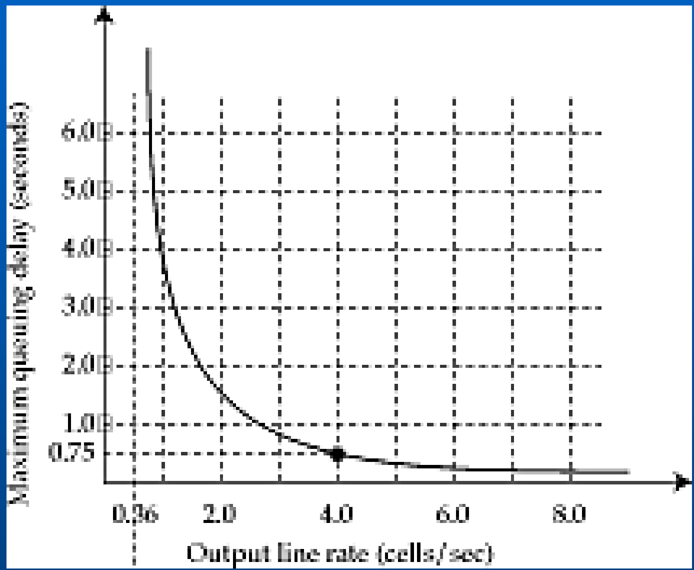
IETF TShirsts

4. Statistical multiplexing



- output rate: 4cells/s. queuing delay <= 3/4s.
- Suppose cells arrive in bursts
  - each burst has 10 cells evenly spaced 1 second apart
  - mean gap between bursts = 100 seconds (average rate = 0.0909 cell/s)
- What should be service rate of output line?
  - No single answer (4c/s? 0.36c/s? 1c/s?)

Statistical multiplexing



- We can trade off *worst-case delay* against *speed of output trunk*
- Statistical Multiplexing Gain = sum of peak input/output rate
  - A cell switch exploits SMG in the same way as a TD multiplexor.
- Whenever long term average rate *differs* from peak, we can trade off service rate for delay (requires buffers for zero loss)
  - key to building packet-switched networks with QoS

Generalized SMG

- *n* bursty source that have *p* peak rate and *a* average rate
- Worst case: simultaneous arrivals -> conservatively serve at *n.p*
- To reduce cost, can serve at *r* with $n.a < r < n.p$
  - Requires buffering -> higher delays
- SMG = $n.p/r$
- general principle:
  - if long-term average rate < peak rate; trade-off service rate for mean delay
- ATM cells can be stored & long distance BW expensive
  - -> SMG applicable
- Not if average rate close to peak rate

5. Integrated services?

- Traditionally, voice, video, and data traffic on separate networks
- Integration
  - easier to manage
  - innovative new services (e.g. Vconferencing)
- How do ATM networks allow for integrated service?
  - lots of (switching) capacity: hardware-oriented switching
  - support for different traffic types
    - signaling for call set-up
    - admission control, Traffic descriptor, policing
    - resource reservation
    - requires intelligent link scheduling for voice/data integration (more flexible than telephone because of headers)

Problems with Connection Oriented approaches for data

- Path construction is non-local and encourages centralization and monopoly (know/control resources)
  - Scheduling is NP hard
- *System* reliability goes down exponentially with scale
  - Requires high reliability *elements*
- Requires a path set-up phase
  - Not efficient for data (especially for large BDP: 100ms at Gbps is 12 MB!)

Datagrams

- A different style of communication
  - Change of Point of view : Focus on endpoints
  - The wires are already there!
- Data is sent in independent chunks of reasonable size with the destination address
  - Fairly share the path
- Simple relaying/routing of datagrams
  - "Connecting" adjacent hops
  - Based on addresses

Datagrams (contd)

- Internet was built on top of the phone system
  - Using the wires differently
- Speed agnostic
  - No set up phase
- But for operators it was:
  - Just an inefficient way to use their network!
  - A pure overlay
- Delivery technology agnostic
  - Could be overlaid over "everything"
  - Phone, Ethernet, satellite, radio, etc.

"TCP/IP"

- Reliability increases exponentially with the system size
- No call setup
  - Higher efficiency
- Distributed inter-domain routing
  - Works on any topology (No scheduling)
  - Tends to spread load
  - Network repairs from failures and
  - "hooks itself up" initially (due to the use of *explicit* address) – a big democratization
- Great for getting ubiquitous communication infrastructure

My how you’ve grown!

- The Internet has doubled in size every year since 1969
- In 1996, 10 million computers joined the Internet
- By July 1997, 10 million more have joined
- By Jan 2001, 100 million hosts
- By March 2002, 400 million users
- By 2004, 800 million users
- By June 2008, 1.46 billion users
- By September 2009, 1.73 billion users
- By June 2010, 1.97 billion users
- Now, everyone who has a phone is likely to also have an email account

What does it look like?

- Loose collection of networks organized into a multilevel hierarchy
  - 10-100 machines connected to a *hub* or a *router*
    - service providers also provide direct dialup access
    - or over a wireless link
  - 10s of routers on a *department backbone*
  - 10s of department backbones connected to *campus backbone*
  - 10s of campus backbones connected to *regional service providers*
  - 100s of regional service providers connected by *national backbone*
  - 10s of national backbones connected by *international trunks*

Example of message routing

 # traceroute parmesan.cs.wisc.edu (three probes at each TTL value)
traceroute to parmesan.cs.wisc.edu (128.105.167.16), 30 hops max, 38 byte packets

```
 1  t4-gw.inria.fr (138.96.32.250)  0.314 ms  0.271 ms  0.332 ms
 2  nice.cssi.renater.fr (195.220.98.117)  7.953 ms  10.770 ms  2.018 ms
 3  nio-n1.cssi.renater.fr (195.220.98.101)  17.489 ms  22.218 ms  14.136 ms
 4  nio-i.cssi.renater.fr (193.51.206.14)  14.080 ms  23.882 ms  18.131 ms
 5  opentransit-nio-i.cssi.renater.fr (193.51.206.42)  22.554 ms  15.353 ms  15.653 ms
 6  P3-0.PASCR2.Pastourelle.opentransit.net (193.251.241.158)  25.020 ms  16.662 ms  20.514 ms
 7  P11-0.PASCR1.Pastourelle.opentransit.net (193.251.241.97)  18.202 ms  15.704 ms  16.216 ms
 8  P12-0.NYKCR2.New-york.opentransit.net (193.251.241.134)  90.137 ms  90.190 ms  89.799 ms
 9  P6-0.NYKBB3.New-york.opentransit.net (193.251.241.238)  96.411 ms  97.740 ms  96.006 ms
10  BBN.GW.opentransit.net (193.251.250.138)  112.554 ms  116.028 ms  110.994 ms
11  p3-0.nycmny1-nbr2.bbnplanet.net (4.24.10.69)  119.815 ms  113.583 ms  108.599 ms
12  * p15-0.nycmny1-nbr1.bbnplanet.net (4.24.10.209)  115.725 ms  115.237 ms
13  so-6-0-0.chcgil2-br2.bbnplanet.net (4.24.4.17)  115.999 ms  124.484 ms  119.278 ms
14  so-7-0-0.chcgil2-br1.bbnplanet.net (4.24.5.217)  116.533 ms  120.644 ms  115.783 ms
15  p1-0.chcgil2-cr7.bbnplanet.net (4.24.8.106)  119.212 ms  117.684 ms  117.374 ms
16  a0.uwisc.bbnplanet.net (4.24.223.22)  123.337 ms  119.627 ms  126.541 ms
17  r-peer-WNMadison-gw.net.wisc.edu (216.56.1.18)  123.403 ms  127.295 ms  129.175 ms
18  144.92.128.226 (144.92.128.226)  124.777 ms  123.212 ms  131.111 ms
19  144.92.128.196 (144.92.128.196)  121.280 ms  126.488 ms  123.018 ms
20  e1-2.foundry2.cs.wisc.edu (128.105.1.6)  132.539 ms  127.177 ms  122.419 ms
21  parmesan.cs.wisc.edu (128.105.167.16)  123.928 ms *  124.471 ms
```

What holds the Internet together?

- Addressing
  - how to refer to a machine on the Internet
- Routing
  - how to get there
- Internet Protocol (IP)
  - what to speak to be understood at the "inter-network" level

Endpoint control - the end2end argument

- Key design philosophy
  - do as much as possible at the endpoint
  - dumb network
  - exactly the opposite philosophy of telephone network
- Layer above IP compensates for network defects
  - Transmission Control Protocol (TCP)
- Can run over any available link technology
  - but no quality of service
  - modification to TCP requires a change at every endpoint
  - telephone network technology upgrade transparent to users

Next courses

- MAC & Link layer
- Addressing and routing
  - Interdomain routing
- Quality of service support