Réseaux,
Protocoles et applications de l'Internet

INF 586

Walid Dabbous

INRIA Sophia Antipolis

Contenu du cours

- Introduction: le téléphone et l’Internet.
- Les liens de communication et l’accès multiple
- Adressage et routage point à point dans l’Internet
- Contrôle de transmission
- Architecture de protocoles
- Communication de groupe
- Support de la qualité de service dans l’Internet

Références

- Cours inspiré (surtout) du livre de S. Keshav
- An Engineering Approach to Computer Networking, S. Keshav, Addison-Wesley, May 1997, 688 pages, ISBN 0-201-63442-2
- Plusieurs autres livres de référence
- Routing in the Internet, C. Huitema, Prentice-Hall, 1995, 319 pages, ISBN 0-13-132192-7
- Data and Computer Communications, W. Stallings, Prentice Hall International Editions, 6th edition, 2000, 810 pages, ISBN 0-13-086388-2
- 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: A Systems Approach, Larry L. Peterson, Bruce S. Davie, Morgan Kaufmann, April 1996, 500 pages, ISBN 1-55-860368-9
- Computer Networks, Andrew S. Tanenbaum, Prentice Hall International Editions, 3rd edition, March 1996, 814 pages, ISBN 0-13-394248-1
- Data Networks, Dimitri P. Bertsekas, Robert Gallager, Prentice Hall, 2nd edition, December 1991, 556 pages, ISBN 0-13-200916-1
- Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture, D. E. Comer, Prentice-Hall, 3rd edition, 1995, 613 pages, ISBN 0-13-216987-8

Références (suite)

- Computer Networks and Internets, D. E. Comer, Prentice-Hall, 3rd edition, 2001, 703 pages, ISBN 0-13-091449-5
- Systèmes multimédias communicants, W. Dabbous éditeur, Hermès Science Publications, 2001, 320 pages, ISBN 2-7462-0251-4
- Interconnections: Bridges and Routers, Radia Perlman, Addison-Wesley, May 1992, 400 pages, ISBN 0201563320
- Multicast Networking And Applications, Kenneth C. Miller, Addison-Wesley, 1999, 282 pages, ISBN 0-201-30979-3
- MobileIP: Design Principles and Practices, C. E. Perkins, Addison-Wesley, 1997, 275 pages, ISBN 0-201-63469-4
- TCP/IP Illustrated, Volume 1: The Protocols, W. Richard Stevens, Addison-Wesley, Published January 1994, 600 pages, ISBN 0201633469
- TCP/IP Illustrated, Volume 2: The Implementation W. Richard Stevens, Wright, Gary R., Addison-Wesley, Published January 1995, 832 pages, ISBN 020163354X
- TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the Unix Domain Protocols, W. Richard Stevens, Gary R. Wright, Addison-Wesley, Hardcover, Published January 1996, 325 pages, ISBN 0201634953
- Advanced programming in the UNIX environment, Richard Stevens, Addison-Wesley, 1992, 768 pages, ISBN 0-201-56317-7
- UNIX network programming, W Richard Stevens, Prentice Hall, 1998, 1240 pages, ISBN 0-13-490012-X

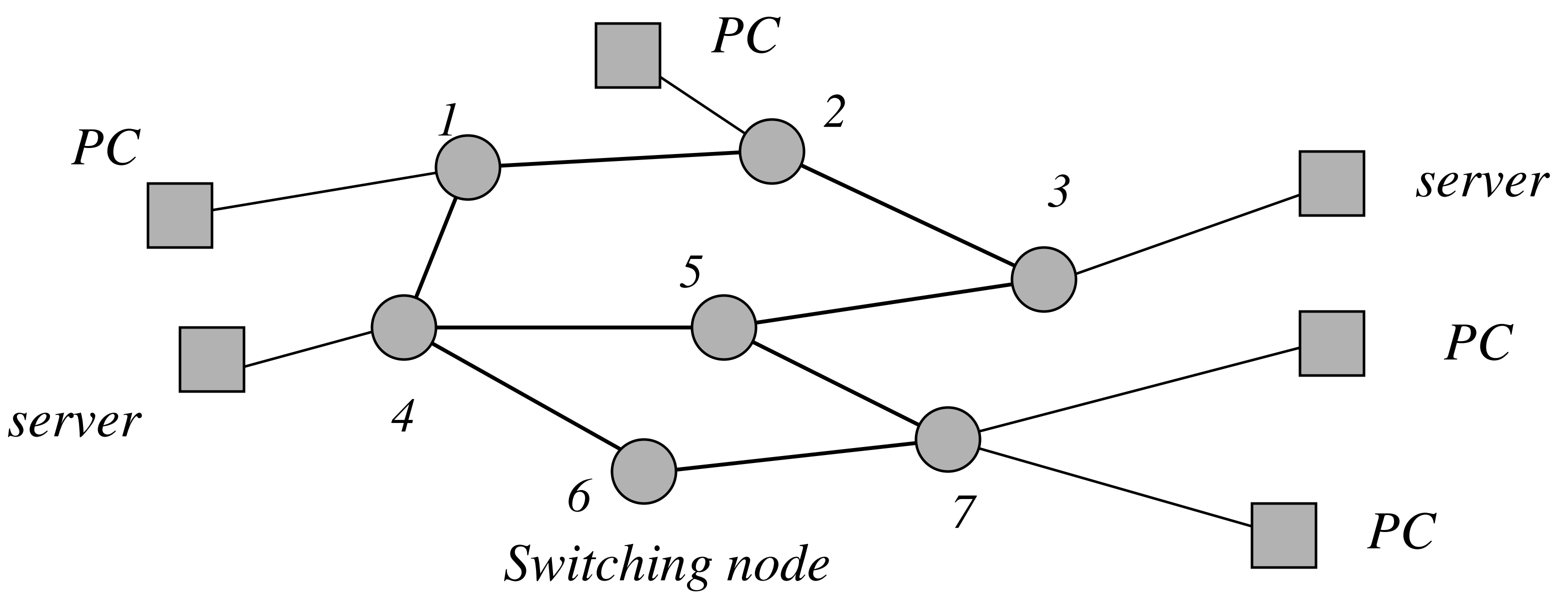
Plan du cours d'aujourd'hui - bloc 1
Le téléphone et l'Internet

- Les réseaux commutés
- Le réseau téléphonique
  - commutation de circuits
- Le réseau ATM
  - commutation de cellules - circuits virtuels
- L'Internet
  - commutation de paquets - datagrammes

Les réseaux commutés

Beyond local area networks

- End systems (stations) send data through a network of intermediate switching nodes
- Some nodes connect only to other nodes (routers, switches)
- usually the network is not fully connected
  - but more that one path from source to destination



Le réseau téléphonique

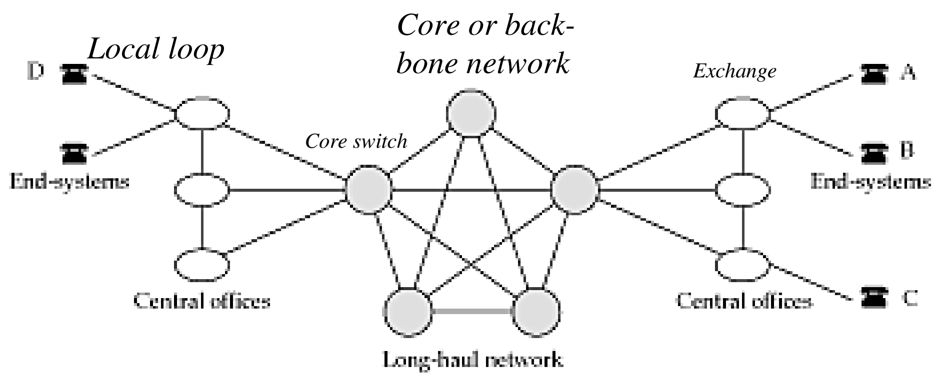
Is it a computer network?

- Specialized to carry voice (more than a billion telephones worldwide)
- But also carries
  - fax
  - modem calls
  - video
- Internally, uses digital *samples*
- Standard end-system/network interface
- Switches and switch controllers are special purpose computers
- Principles in its design apply to more general computer networks

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
  - telephone number is a hint about how to route a call
    - but not for 800/888 (toll-free) / 700 (AT&T Incoming call forwarding) / 900 (pay-per-call) numbers

The components of a telephone network

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

1. End-systems

- Transducers
  - key to carrying voice on wires
- Dialer
- Ringer
- Switchhook at central office interprets tones or pulses
  - place a call
  - or do call forwarding
  - sends ring signal
    - power for ringing provided by central office

Sidetone & Echo

- Transmission circuit needs two wires
- And so does reception circuit
- => 4 wires from every central office to home
- Can we do better?
- Use *same* pair of wires for both transmission and reception
- Two problems: sidetone and echo
  - Sidetone attenuation: balance circuit is required
  - (expensive) Echo cancellation for *long-distance* calls
- Lesson
  - keep end-to-end delays as short as possible

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 µs/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

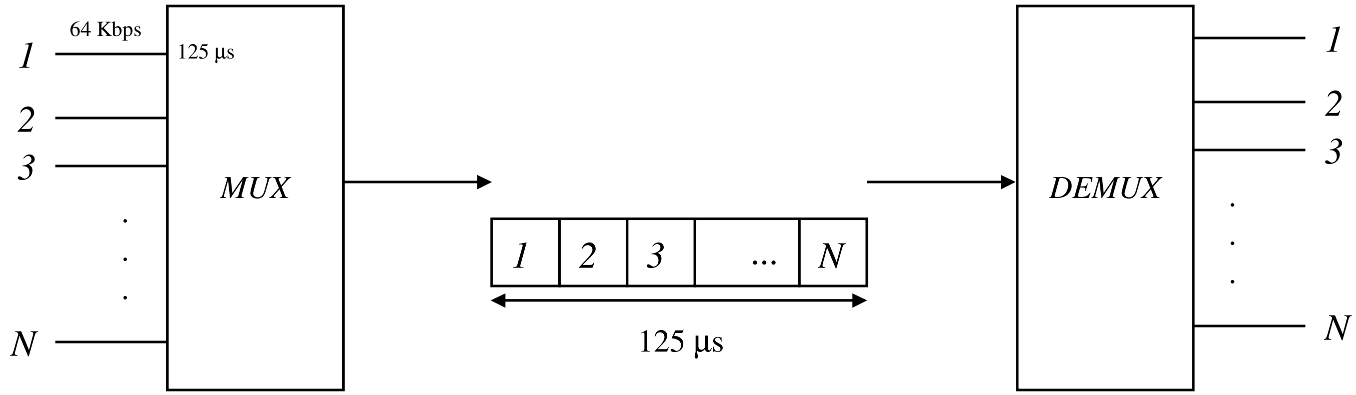
- *Trunks* between central offices carry hundreds of conversations
- Can't run thick bundles!
- Instead, send many calls on the same wire
  - *multiplexing*
- Analog multiplexing (FDM)
  - bandlimit call to 3.4 KHz and frequency shift onto higher bandwidth trunk
  - obsolete, the telephone network is 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 boudary)

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

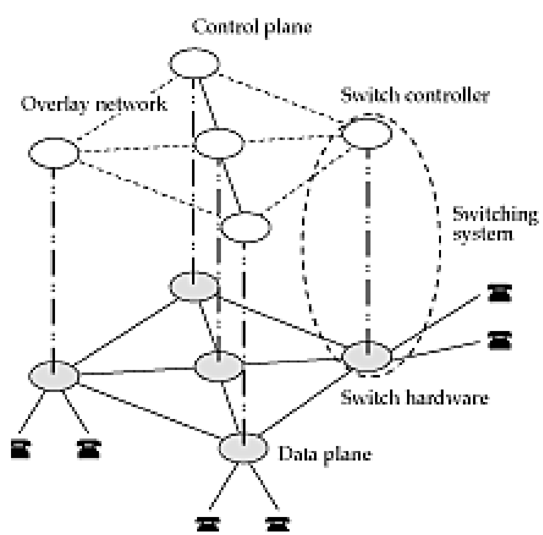
Digital Signal Number	Number of previous level circuits	Number of voice circuits	Bandwidth
DS0		1	64 Kbps
DS1 - T1	24	24	1.544Mbps
DS2	4	96	6.312 Mbps
DS3 - T3	7	672 = 28 T1	44.736 Mbps

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

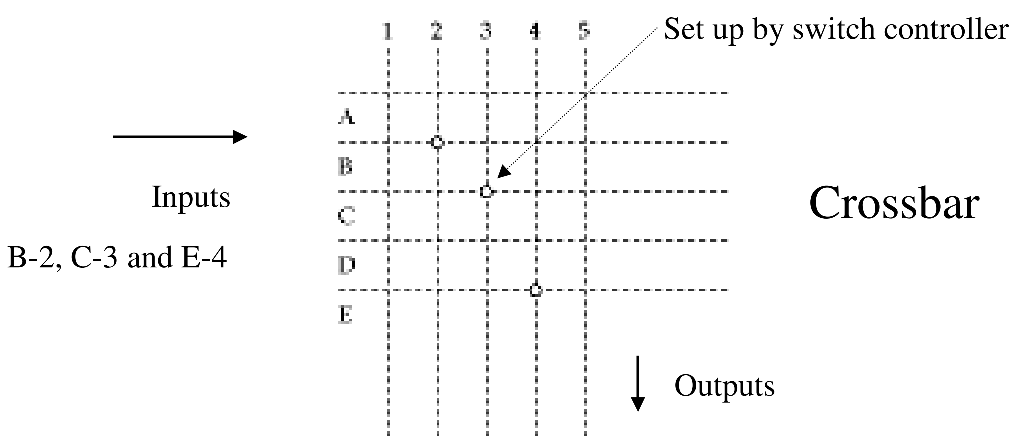
3. Switching

- 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



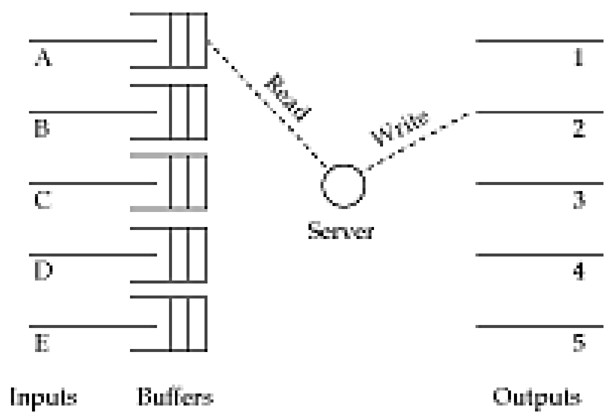
Switching: what does a switch do?

- 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 larger elements

More details: A circuit switch

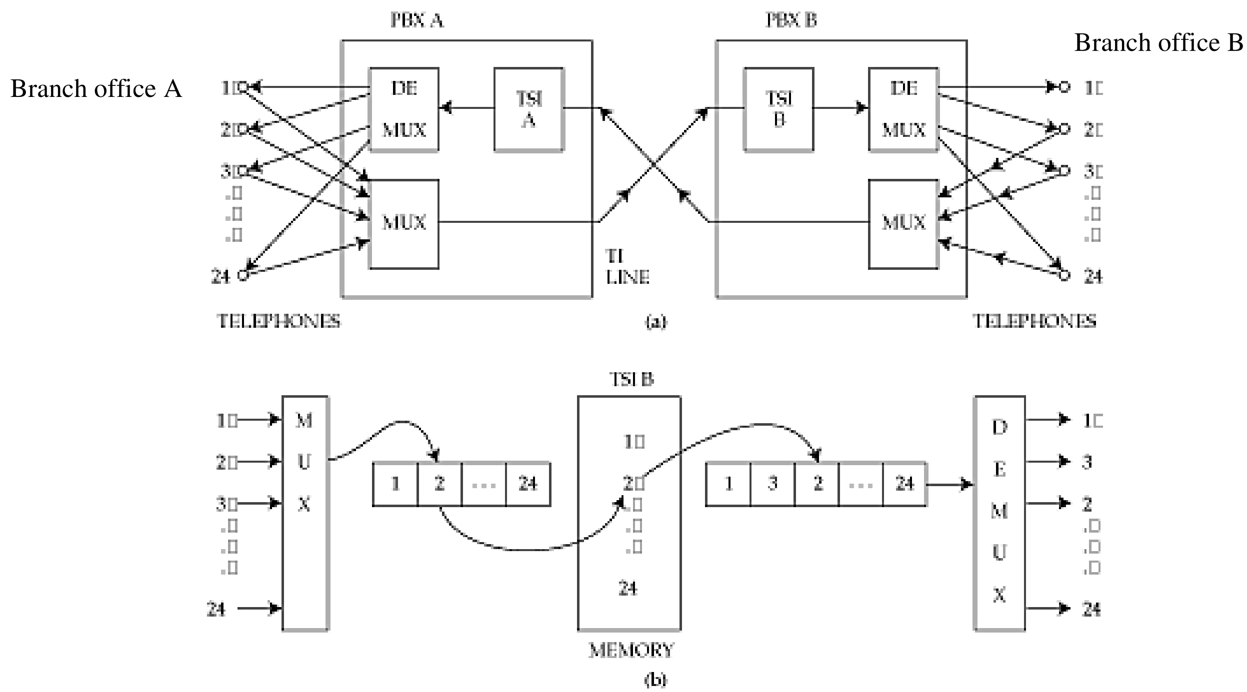
- A switch that can handle N calls has N logical inputs and N logical outputs
  - N up to 200,000
- In practice, input trunks are multiplexed
  - far fewer physical I/O lines
  - example: DS3 trunk carries 672 simultaneous calls
- Multiplexed trunks carry *frames* = set of samples
- Goal: extract samples from frame, and depending on position in frame, switch to output
  - each incoming sample has to get to the right output line and the right slot in the output frame
  - demultiplex, switch, multiplex

Call blocking

- Can’t find a path from input to output (reject blocked calls)
- Internal blocking
  - slot in output frame exists, but no path through the switch
- Output blocking
  - no slot in output frame is available (compete for the same output)
- *Line* switch : connect a specific input to a specific output
- *Transit* switch: connect an input to one *several* outputs
- Internal and output blocking is reduced in transit switches
  - need to put a sample in one of *several* slots going to the desired next hop
  - a transit switch acheives same blocking probability as a line switch with less hardware

More on Time division switching

- Key idea: when demultiplexing, position in frame determines output trunk
- Time division switching interchanges sample position within a frame: time slot interchange (TSI)

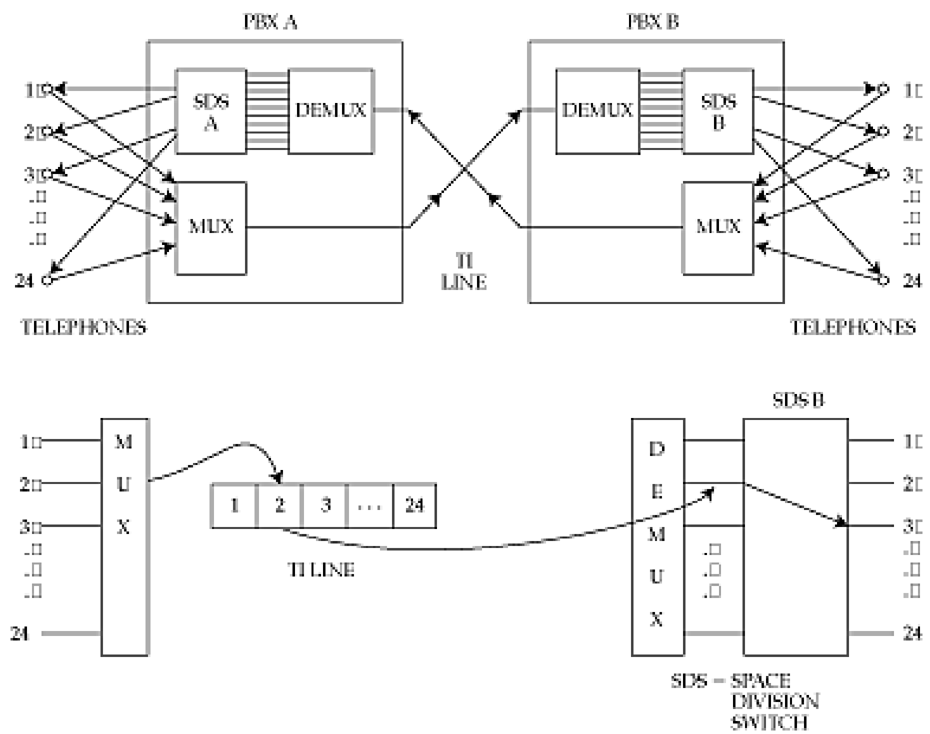


How large a TSI can we build?

- Limit is time taken to read and write to memory
- For 120,000 circuits
  - need to read and write memory 120,000 times every 125 μs (slot duration)
  - each operation takes around 0.5 ns => impossible with current technology
  - with 40-ns memory => 1500 circuits!
- Need to look to other techniques

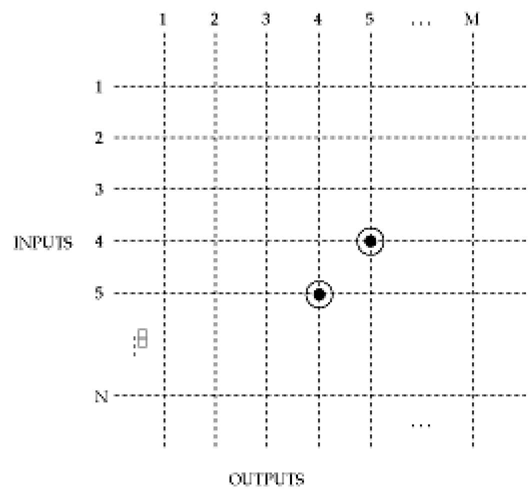
Space division switching

- Each sample takes a different path through the switch, depending on its destination



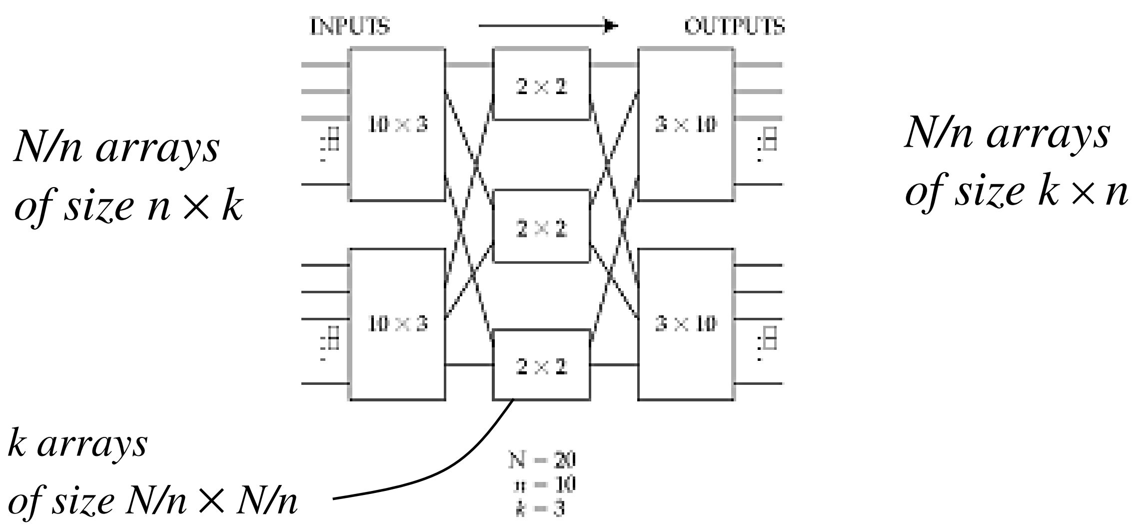
Crossbar

- Simplest possible space-division switch
- *Crosspoints* can be turned on or off
- For multiplexed inputs, need a switching *schedule*
  - as different samples may have different destinations
- Internally nonblocking
  - vulnerable to single faults (only one path between given input output pair)
  - time taken to set crosspoints grows quadratically with N
  - need N^2 crosspoints
- Small switches 8×8 or 64×64



Multistage crossbar

- In a crossbar during each switching time only one crosspoint per row or column is active
- Can *save crosspoints* if a crosspoint can attach to more than one input line
- This is done in a multistage crossbar

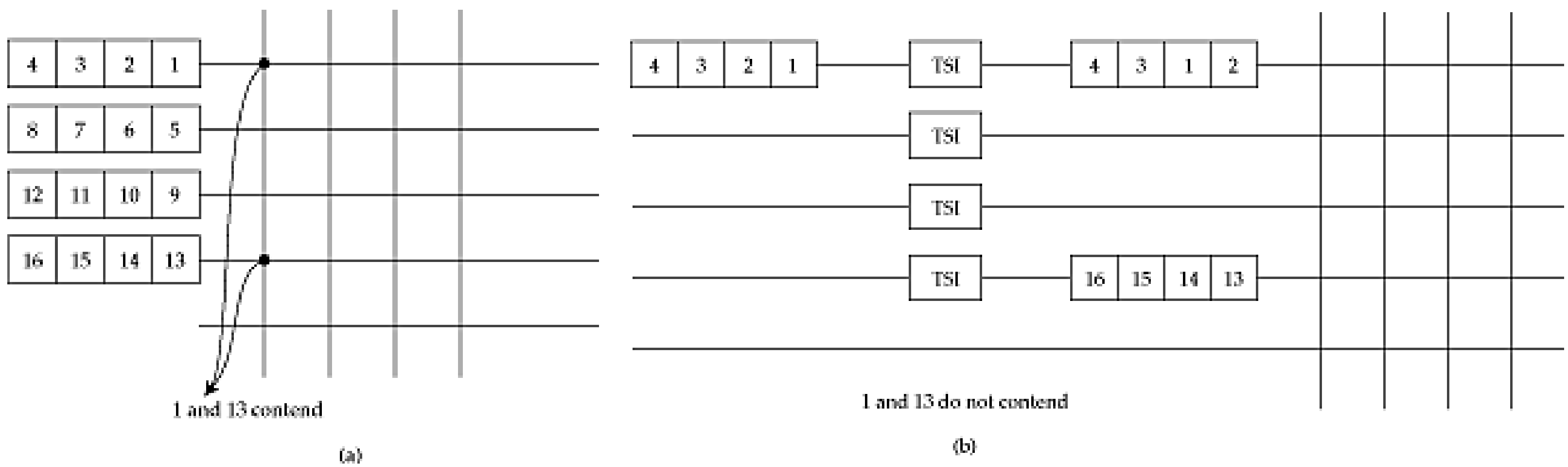


Multistage crossbar

- Can suffer internal blocking
  - unless sufficient number of second-level stages ($k > 2n - 2$)
  - but requires rearranging existing connections as a new call arrives
  - Clos network: *rearrangably nonblocking* switch
- Number of crosspoints < N^2
  - minimize crosspoints for n ~ SQRT(N)
- Finding a path from input to output requires a depth-first-search
  - path stored in switch schedule
- Scales better than crossbar, but still not too well
  - 120,000 call switch needs ~250 million crosspoints
- Unless we accept blocking
  - trade-off between blocking probability and switch cost

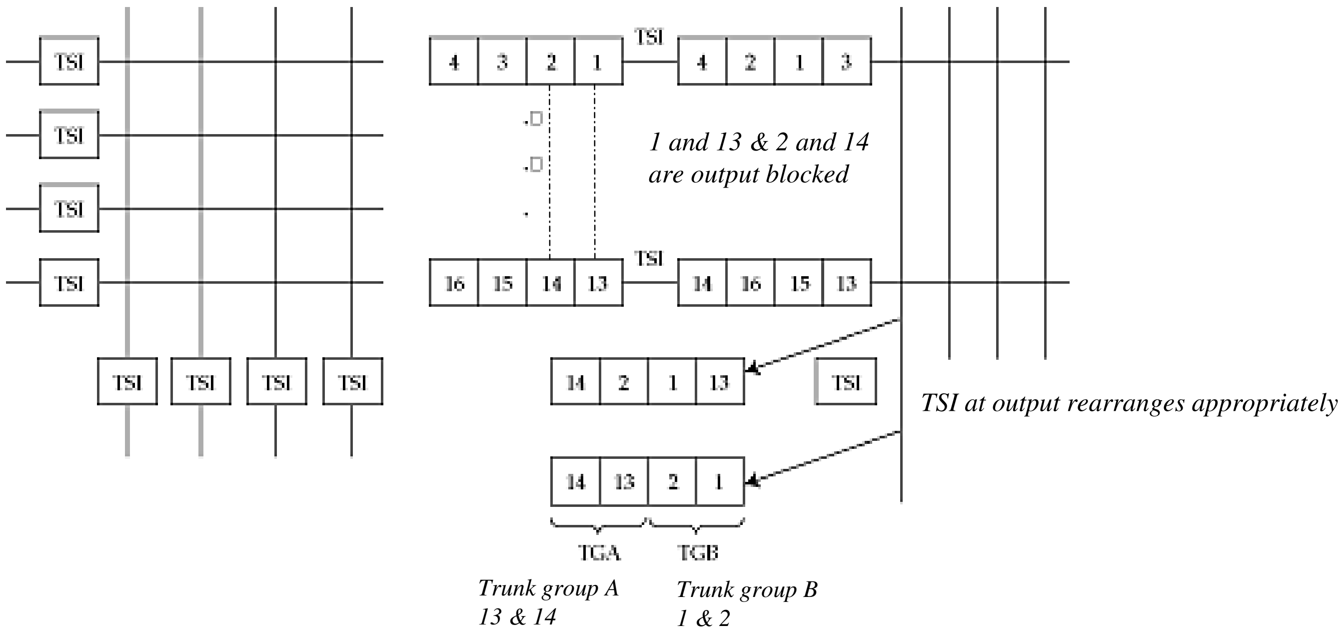
Time-space switching

- Precede each input trunk in a crossbar with a TSI
- "Delay" samples so that they arrive at the right time for the space division switch's schedule
- Allows to build non blocking SDS with fewer crosspoints than a Clos switch



Time-space-time (TST) switching

- Allowed to flip samples both on input and output trunk
- Gives more flexibility => lowers call blocking probability



4. Signaling

- Recall that a switching system has a switch and a switch controller
- Switch controller is in the *control* plane
  - does not touch voice samples
- Manages the network
  - call routing (collect *dialstring* and forward call)
  - alarms (ring bell at receiver)
  - billing
  - directory lookup (for 800/888 calls)

Challenges for the telephone network

- Multimedia
  - simultaneously transmit voice/data/video over the network
  - people seem to want it
  - existing telephone network can't handle it
    - bandwidth requirements
    - *burstiness* in traffic (TSI can't skip input)
      - either peak rate service or very large buffers
    - change in statistical behavior with regard to voice
      - decades of experience for telephone engineers
- Backward compatibility of new services
  - huge existing infrastructure
  - "advantage" of developing countries
- Regulation
  - monopoly stifles innovation

Challenges

- Competition
  - future telephone networks will no longer be monopolies
    - end to good times
  - how to manage the transition?
    - be more responsive to technological innovations
    - at the expense of long term thinking!
- Inefficiencies in the system
  - an accumulation of incompatible systems and formats
  - special-purpose systems of the past (assembly language parts)
  - ‘legacy’ systems
  - need to change them without breaking the network

Les réseaux ATM

Why ATM networks?

- Different information types require different qualities of service from the network
  - stock quotes vs. USENET
- Telephone networks support a single quality of service
  - and is expensive to boot
- ATM networks are meant to support a range of service qualities at a reasonable cost

Design goals

- Providing “end-to-end” quality of service
- High bandwidth
- Scalability
- Manageability
- Cost-effectiveness

How far along are we?

- Basic architecture has been defined
- But delays have resulted in ceding desktop to IP
- Also, little experience in traffic specification, multicast, and fault tolerance
- We will never see “end-to-end” ATM
  - but its ideas continue to influence design of next-generation Internet - see block 7 (Scheduling)
  - Internet technology + ATM philosophy -- will it work ?
- Note--two standardization bodies
  - ATM Forum
  - International Telecommunications Union-Telecommunications Standardization Sector (ITU-T)

Concepts

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

Together

can carry *multiple* types of traffic

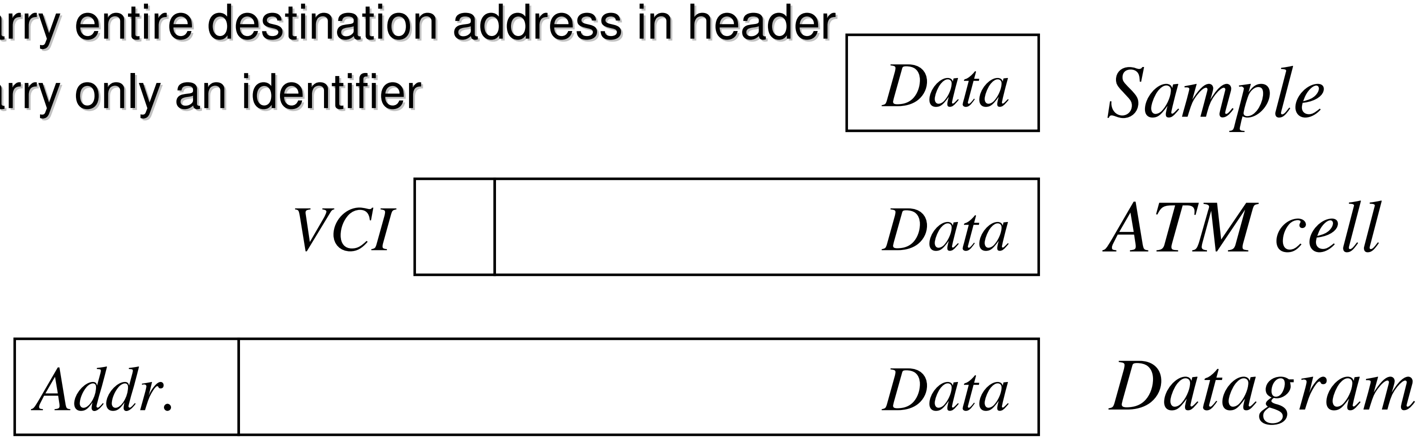
with (ATM) end-to-end quality of service

1. Virtual circuits

- Some background first
- Telephone network operates in *Synchronous Transfer Mode*
  - the destination of a sample depends on where it comes from. Knowing *when* it came is suffcient, no need for a descriptive header
  - example--shared leased link to the same destination
- Problems with STM
  - idle users consume bandwidth
  - 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

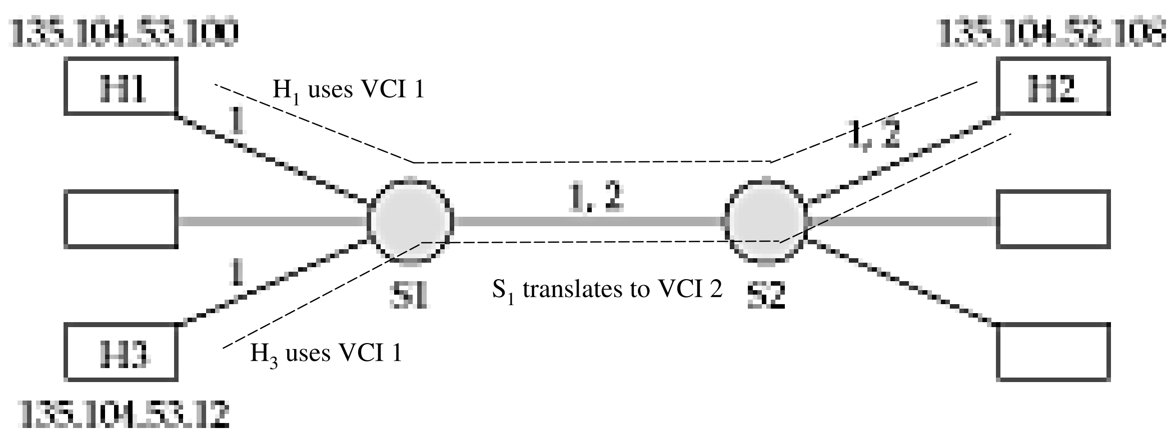
Virtual circuits (contd.)

- 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 entire destination address in header
  - carry only an identifier



Virtual circuits (contd.)

- 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

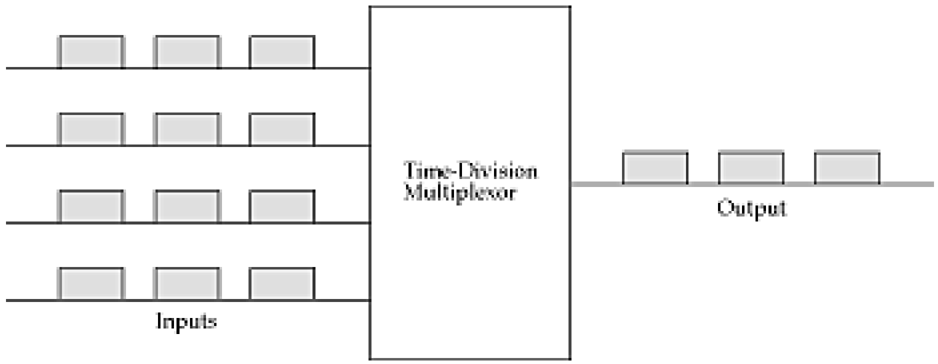
- Pros
  - Simpler buffer hardware
    - packet arrival and departure requires us to manage fixed buffer sizes (easier, no memory fragmentation)
  - Simpler line scheduling
    - each cell takes a constant chunk of bandwidth to transmit -> harder to achieve simple ratios with variable size packets
  - 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%

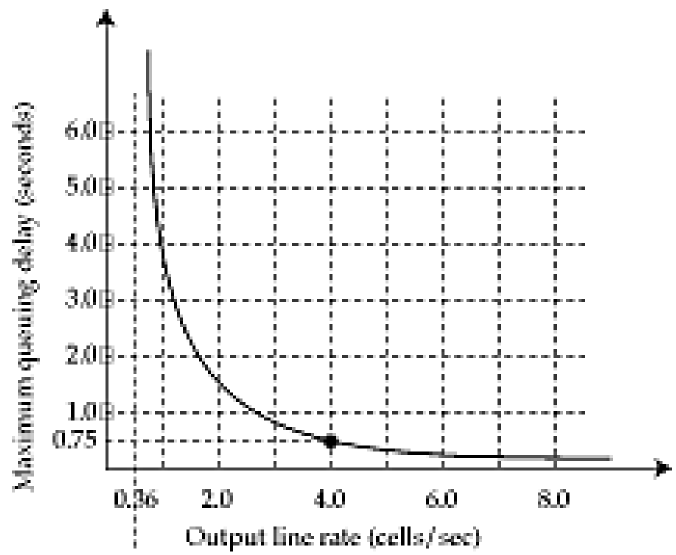
IETF TShirsts

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.g*
  - 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 service

- Traditionally, voice, video, and data traffic on separate networks
- Integration
  - easier to manage
  - innovative new services (Vconferencing, Venvironments)
- 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)

Challenges

- Quality of service
  - defined, but not used!
  - still needs research
- Scaling
  - little experience
- Competition from other LAN technologies
  - FDDI
  - 100Mbps Ethernet
- Standardization
  - Political (ATM forum is not the IETF)
  - slow

Challenges

- IP
  - a vast, fast-growing, non-ATM infrastructure
  - interoperation is a pain in the neck, because of fundamentally different design philosophies
    - connectionless vs. connection-oriented
    - resource reservation vs. best-effort
    - different ways of expressing QoS requirements
    - routing protocols differ
  - ATM serves as a “leased line” service between IP routers

L'Internet

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, 700 to 900 million expected
- Soon, 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
```

A closer example

```
# traceroute ultralix.polytechnique.fr
traceroute to ultralix.polytechnique.fr (129.104.11.15), 30 hops max, 38 byte packets
 1  t4-gw.inria.fr (138.96.32.250)  0.550 ms  0.270 ms  0.263 ms
 2  nice.cssi.renater.fr (195.220.98.117)  8.779 ms  6.381 ms  2.391 ms
 3  nio-n1.cssi.renater.fr (195.220.98.101)  19.744 ms  24.804 ms  17.490 ms
 4  nio-n1.cssi.renater.fr (193.51.206.5)  21.975 ms  17.592 ms  13.758 ms
 5  jussieu.cssi.renater.fr (194.214.109.6)  18.938 ms  21.357 ms  15.002 ms
 6  epp-jussieu.cssi.renater.fr (193.51.12.82)  25.117 ms  29.762 ms  21.258 ms
 7  129.104.63.1 (129.104.63.1)  23.580 ms  20.993 ms  25.804 ms
 8  129.104.63.13 (129.104.63.13)  21.973 ms  16.783 ms  23.964 ms
 9  ultralix.polytechnique.fr (129.104.11.15)  19.174 ms *  25.052 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

More details : joining the Internet

- How can people talk to you?
  - get an IP address from your administrator
- How do you know where to send your data?
  - if you only have a single external connection, then no problem
  - otherwise, need to speak a routing protocol to decide next hop
- How to format data?
  - use the IP format so that intermediate routers can understand the destination address
- Decentralized and distributed
  - No single authority for addressing
  - No coordination for routing
  - Connectionless IP service
  - scales to millions of hosts

What lies at the heart?

- Two key technical concepts
  - packets
  - store and forward

Packets

- Self-descriptive data
  - packet = data + metadata (header)
- Packet vs. sample
  - samples are not self descriptive
  - to forward a sample, we have to know *where* it came from (in fact order in frame)
  - can't store it!
  - hard to handle bursts of data

Store and forward

- Metadata allows us to forward packets when we want
- E.g. letters at a post office headed for main post office
  - address labels allow us to forward them in batches
- Efficient use of critical resources
  - allows to share the cost of expensive transmission link
- Three problems
  - hard to control delay within network
  - switches need memory for buffers
  - convergence of flows can lead to congestion

Key features of the Internet

- Addressing
- Routing
- Endpoint control

Addressing

- Internet addresses are called IP addresses
- Refer to a *host interface*: need one IP address per interface
- Addresses are structured as a two-part hierarchy
  - network number
  - host number

135.105.53	*100*

An interesting problem

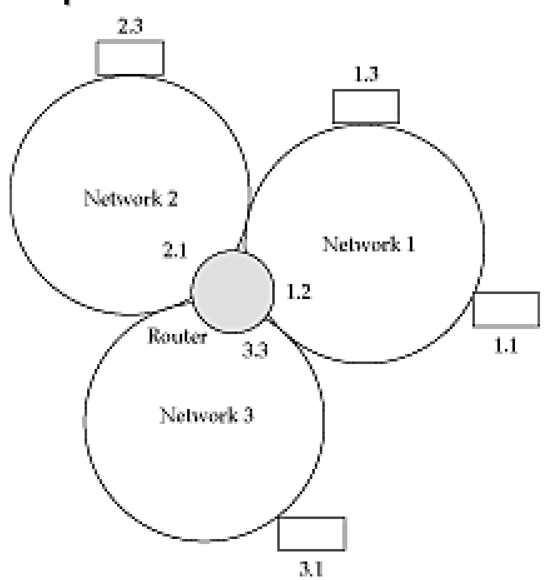
- How many bits to assign to host number and how many to network number?
- If many networks, each with a few hosts, then more bits to network number
- And *vice versa*
- But designer’s couldn’t predict the future
- Decided three sets of partitions of bits
  - class A: 8 bits network (in fact 7), 24 bits host
  - class B: 16 bits networks (in fact 14), 16 bits host
  - class C: 24 bits network (in fact 21), 8 bits host

Addressing (contd.)

- To distinguish among them
  - use leading bit
  - first bit = 0=> class A, range 1-126 (127 is loopback)
  - first bits 10 => class B, range128-191
  - first bits 110 => class C, range 192-223
- Problem
  - if you want more than 256 hosts in your network, need to get a class B, which allows 64K hosts => wasted address space
- Solution
  - associate *every* address with a *mask* that indicates partition point
  - *CIDR (Classless InterDomain Routing)*
- What about IPv6?

Routing

- How to get to a destination given its IP address?
- We need to know the next hop to reach a particular network number
  - this is called a *routing table*
  - computing routing tables is non-trivial (distributed routing protocol)
- Simplified example



Default routes

- Strictly speaking, need next hop information for every network in the Internet
  - > 80,000 now
- Instead, keep detailed routes only for local neighborhood
- For unknown destinations, use a *default* router
- Reduces size of routing tables at the expense of non-optimal paths

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
    - cellular phone introduction does not require fixed telephones upgrade

Challenges

- IP address space shortage
  - because of free distribution of inefficient Class B addresses
  - decentralized control => hard to recover addresses, once handed out
- Decentralized control
  - allows scaling, but makes *reliability* next to impossible
  - cannot "guarantee" that a route exists
  - Corrupted routing messages can cause a major disaster
  - Non-optimal routing
    - each administrative makes a locally optimal decision

Challenges (contd.)

- Decentralized control (contd.)
  - hard to guarantee security
    - end-to-end encryption is a partial solution
    - requires scalable and efficient key distribution scheme
  - no equivalent of white or yellow pages
    - hard to reliably discover a user's email address
  - no uniform solution for accounting and billing
    - can't even reliably identify individual users
    - password based identification does not "scale"
    - -> flat rate billing

Challenges (contd).

- Multimedia
  - requires network to support quality of service of some sort
    - hard to integrate into current architecture
    - store-and-forward => shared buffers => traffic interaction => hard to provide service quality
  - requires endpoint to signal to the network what it wants
    - but Internet does not have a simple way to identify streams of packets
    - nor are routers required to cooperate in providing quality
    - and what about pricing!
  - However, basic Internet multimedia applications exist today