Circuit Switching

Behnam Momeni
Computer Engineering Department
Sharif University of Technology

Acknowledgments: Lecture slides are from Computer networks course thought by Jennifer Rexford at Princeton University. When slides are obtained from other sources, a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.
Circuit Switching
Circuit Switching (e.g., Phone Network)

- **Establish**: source creates circuit to destination
  - Node along the path store connection info
  - Nodes may reserve resources for the connection

- **Transfer**: source sends data over the circuit
  - No destination address, since nodes know path

- **Teardown**: source tears down circuit when done
Timing in Circuit Switching

Host 1 — Switch 1 — Switch 2 — Host 2

- **Circuit Establishment**
- **Transfer**
- **Circuit Teardown**

- Transmission delay

- Propagation delay between Host 1 and Switch 1
- Propagation delay between Host 1 and Host 2

Information

time
Circuit Switching With Human Operator
Circuit Switching: Multiplexing a Link

• Time-division
  – Each circuit allocated certain time slots

• Frequency-division
  – Each circuit allocated certain frequencies
Advantages of Circuit Switching

• Guaranteed bandwidth
  – Predictable communication performance
  – Not “best-effort” delivery with no real guarantees

• Simple abstraction
  – Reliable communication channel between hosts
  – No worries about lost or out-of-order packets

• Simple forwarding
  – Forwarding based on time slot or frequency
  – No need to inspect a packet header

• Low per-packet overhead
  – Forwarding based on time slot or frequency
  – No IP (and TCP/UDP) header on each packet
Disadvantages of Circuit Switching

• Wasted bandwidth
  – Bursty traffic leads to idle connection during silent period
  – Unable to achieve gains from statistical multiplexing

• Blocked connections
  – Connection refused when resources are not sufficient
  – Unable to offer “okay” service to everybody

• Connection set-up delay
  – No communication until the connection is set up
  – Unable to avoid extra latency for small data transfers

• Network state
  – Network nodes must store per-connection information
  – Unable to avoid per-connection storage and state
Virtual Circuits
Virtual Circuit (VC)

- Hybrid of packets and circuits
  - Circuits: establish and teardown along end-to-end path
  - Packets: divide the data into packets with identifiers

- Packets carry a virtual-circuit identifier
  - Associates each packet with the virtual circuit
  - Determines the next link along the path

- Intermediate nodes maintain state VC
  - Forwarding table entry
  - Allocated resources
Establishing the Circuit

• **Signaling**
  – Creating the entries in the forwarding tables
  – Reserving resources for the virtual circuit, if needed

• **Two main approaches to signaling**
  – Network administrator configures each node
  – Source sends set-up message along the path

• **Set-up latency**
  – Time for the set-up message to traverse the path
  – … and return back to the source

• **Routing**
  – End-to-end path is selected during circuit set-up
Virtual Circuit Identifier (VC ID)

- Virtual Circuit Identifier (VC ID)
- Source set-up: establish path for the VC
- Switch: mapping VC ID to an outgoing link
- Packet: fixed length label in the header
Swapping the Label at Each Hop

- Problem: using VC ID along the whole path
  - Each virtual circuit consumes a unique ID
  - Starts to use up all of the ID space in the network

- Label swapping
  - Map the VC ID to a new value at each hop
  - Table has old ID, and next link and new ID
Virtual Circuits Similar to IP Datagrams

- **Data divided into packets**
  - Sender divides the data into packets
  - Packet has address (e.g., IP address or VC ID)

- **Store-and-forward transmission**
  - Multiple packets may arrive at once
  - Need buffer space for temporary storage

- **Multiplexing on a link**
  - No reservations: statistical multiplexing
    - Packets are interleaved without a fixed pattern
  - Reservations: resources for group of packets
    - Guarantees to get a certain number of “slots”
Virtual Circuits Differ from IP Datagrams

- **Forwarding look-up**
  - Virtual circuits: fixed-length connection id
  - IP datagrams: destination IP address

- **Initiating data transmission**
  - Virtual circuits: must signal along the path
  - IP datagrams: just start sending packets

- **Router state**
  - Virtual circuits: routers know about connections
  - IP datagrams: no state, easier failure recovery

- **Quality of service**
  - Virtual circuits: resources and scheduling per VC
  - IP datagrams: difficult to provide QoS
Quality of Service (QoS) on Virtual Circuits
Quality of Service

• Allocating resources to the virtual circuit
  – E.g., guaranteed bandwidth on each link in the path
  – E.g., guaranteeing a maximum delay along the path

• Admission control
  – Check during signaling that the resources are available
  – Saying “no” if they are not, and reserving them if they are

• Resource scheduling
  – Apply scheduling algorithms during the data transfer
  – To ensure that the performance guarantees are met
Admission Control

• Source sends a reservation message
  – E.g., “this virtual circuit needs 5 Mbps”

• Each switch along the path
  – Keeps track of the reserved resources
    • E.g., “the link has 6 Mbps left”
  – Checks if enough resources remain
    • E.g., “6 Mbps > 5 Mbps, so circuit can be accepted”
  – Creates state for circuit and reserves resources
    • E.g., “now only 1 Mbps is available”
Admission Control: Flowspec

- Flowspec: information about the traffic
  - The traffic characteristics of the flow
  - The service requested from the network

- Specifying the traffic characteristics
  - Simplest case: constant bit rate (some # of bits per sec)
  - Yet, many applications have variable bit rates
  - … and will send more than their average bit rate
Specifying Bursty Traffic

• Option #1: Specify the maximum bit rate
  – Maximum bit rate may be much higher average
  – Reserving for the worst case is wasteful

• Option #2: Specify the average bit rate
  – Average bit rate is not sufficient
  – Network will not be able to carry all of the packets
  – Reserving for average case leads to bad performance

• Option #3: Specify the burstiness of the traffic
  – Specify both the average rate and the burst size
  – Allows the sender to transmit bursty traffic
  – … and the network to reserve the necessary resources
Token Bucket Traffic Model

- **Parameterized Traffic Specification**
  - Token Rate \( (r) \)
  - Bucket Depth \( (d) \)
- **Each unit of data consumes a token to be sent**
  - Average traffic rate: \( r \)
  - Bursty traffic volume: \( d \)
  - Divide traffic to conforming and excessive traffics
  - Suitable for both traffic policing and traffic shaping
    - **Traffic Policing:** Excessive packets are dropped,
    - **Traffic Shaping:** Excessive packets are delayed to conform.
Token Bucket Traffic Model
Leaky Bucket Traffic Model

- **Parameterized Traffic Specification**
  - Token Leakage Rate \((r)\)
  - Bucket Depth \((d)\)
- **Each unit of data fills the bucket as much as one token**
- **Tokens leak with a fixed rate to empty space for new data**
  - Average traffic rate: \(r\)
  - Bursty traffic volume: \(d\)
  - Suitable for traffic shaping
  - In one variation, bucket is similar to a queue which holds traffic and sends it with leakage rate
    - Excessive data is dropped
    - Output rate is always equal or less than \(r\)
Leaky Bucket Traffic Model
Service Requested From the Network

• Variety of service models
  – Bandwidth guarantee (e.g., 5 Mbps)
  – Delay guarantee (e.g., no more than 100 msec)
  – Loss rate (e.g., no more than 1% packet loss)

• Signaling during admission control
  – Translate end-to-end requirement into per-hop
  – Easy for bandwidth (e.g., 5 Mbps on each hop)
  – Harder for delay and loss
  – … since each hop contributes to the delay and loss

• Per-hop admission control
  – Router takes the service requirement and traffic spec
  – … and determines whether it can accept the circuit
Ensuring the Source Behaves

• Guarantees depend on the source behaving
  – Extra traffic might overload one or more links
  – Leading to congestion, and resulting delay and loss
  – Solution: need to enforce the traffic specification

• Solution #1: policing
  – Drop all data in excess of the traffic specification

• Solution #2: shaping
  – Delay the data until it obeys the traffic specification

• Solution #3: marking
  – Mark all data in excess of the traffic specification
  – … and give these packets lower priority in the network
Enforcing Behavior

- Applying a leaky bucket to the traffic
  - Simulating a leaky bucket \((r, d)\) at the edge
  - Discarding, delaying, or marking packets accordingly

- Ensures that the incoming traffic obeys the profile
  - So that the network can provide the guarantees

- Technical challenge
  - Applying leaky buckets for many flows at a high rate
Link Scheduling: FIFO

• First-in first-out scheduling
  – Simple to implement
  – But, restrictive in providing guarantees

• Example: two kinds of traffic
  – Video conferencing needs high bandwidth and low delay
    • E.g., 1 Mbps and 100 msec delay
  – E-mail transfers are not that sensitive about delay

• Cannot admit much e-mail traffic
  – Since it will interfere with the video conference traffic
Link Scheduling: Strict Priority

• **Strict priority**
  – Multiple levels of priority
  – Always transmit high-priority traffic, when present
  – .. and force the lower priority traffic to wait

• **Isolation for the high-priority traffic**
  – Almost like it has a dedicated link
  – Except for the (small) delay for packet transmission
    • High-priority packet arrives during transmission of low-priority
    • Router completes sending the low-priority traffic first
Link Scheduling: Weighted Fairness

• Limitations of strict priority
  – Lower priority queues may starve for long periods
  – … even if the high-priority traffic can afford to wait

• Weighted fair scheduling
  – Assign each queue a fraction of the link bandwidth
  – Rotate across the queues on a small time scale
  – Send extra traffic from one queue if others are idle

50% red, 25% blue, 25% green
Link Schedulers: Trade-Offs

• Implementation complexity
  – FIFO is easy
    • One queue, trivial scheduler
  – Strict priority is a little harder
    • One queue per priority level, simple scheduler
  – Weighted fair scheduling
    • One queue per virtual circuit, and more complex scheduler

• Admission control
  – Using more sophisticated schedulers can allow the router to admit more virtual circuits into the network
  – Getting close to making full use of the network resources
  – E.g., FIFO requires very conservative admission control
Routing in Virtual Circuit Networks

• Routing decisions take place at circuit set-up
  – Resource reservations made along end-to-end path
  – Data packets flow along the already-chosen path

• Simplest case: routing based only on the topology
  – Routing based on the topology and static link weights
  – Source picks the end-to-end path, and signals along it
  – If the path lacks sufficient resources, that’s too bad!
Quality-of-Service Routing

• QoS routing: source selects the path intelligently
  – Tries to find a path that can satisfy the requirements

• Traffic performance requirement
  – Guaranteed bandwidth $b$ per connection

• Link resource reservation
  – Reserved bandwidth $r_i$ on link $i$
  – Capacity $c_i$ on link $i$

• Signaling: admission control on path $P$
  – Reserve bandwidth $b$ on each link $i$ on path $P$
  – Block: if $(r_i + b > c_i)$ then reject (or try again)
  – Accept: else $r_i = r_i + b$
Source-Directed QoS Routing

- New connection with $b = 3$
  - Routing: select path with available resources
  - Signaling: reserve bandwidth along the path ($r = r + 3$)
  - Forward data packets along the selected path
  - Teardown: free the link bandwidth ($r = r - 3$)
QoS Routing: Link-State Advertisements

• Advertise available resources per link
  – E.g., advertise available bandwidth \((c_i - r_i)\) on link \(i\)
  – Every \(T\) seconds, independent of changes
  – … or, when the metric changes beyond threshold

• Each router constructs view of topology
  – Topology including the latest link metrics

• Each router computes the paths
  – Looks at the requirements of the connection
  – … as well as the available resources in the network
  – And selects a path that satisfies the needs

• Then, the router signals to set up the path
  – With a high likelihood that the request is accepted
Virtual Circuit Realizations

- **Asynchronous Transfer Mode (ATM)**
  - Small and fixed sized packets (53 bytes)
  - IP packets can be fragmented in **ATM cells** and reassembled later
  - ATM switches (layers 1, 2, and 3) / Preferred to work end-to-end
  - Virtual Paths (VP)

- **Frame Relay**
  - Designed for the context of ISDN networks (layers 1 and 2)
  - Permanent Virtual Circuit (PVC)
  - Switched Virtual Circuit (SVC)

- **Synchronous Optical Networking (SONET)**
  - Can be used to carry ATM cells (layer 1)
  - Virtual Container (VC)
Inferring the Need for a Virtual Circuit

• Which IP packets go on a virtual circuit?
  – All packets in the same TCP or UDP transfer?
  – All packets between same pair of end hosts?
  – All packets between same pair of IP subnets?

• Edge router can infer the need for a circuit
  – Match on packet header bits
    • E.g., source, destination, port numbers, etc.
  – Apply policy for picking bandwidth parameters
    • E.g., Web traffic get 10 Kbps, video gets 2 Mbps
  – Trigger establishment of circuit for the traffic
    • Select path based on load and requirements
    • Signal creation of the circuit
    • Tear down circuit after an idle period
Grouping IP Packets Into Flows

- Group packets with the “same” end points
  - Application level: single TCP connection
  - Host level: single source-destination pair
  - Subnet level: single source prefix and dest prefix

- Group packets that are close together in time
  - E.g., 60-sec spacing between consecutive packets
Challenges for IP Over ATM

• Many IP flows are short
  – Most Web transfers are less than 10 packets
  – Is it worthwhile to set up a circuit?

• Subdividing an IP packet into cells
  – Wasted space if packet is not multiples of 48 bytes

• Difficult to know what resources to reserve
  – Internet applications don’t specify traffic or QoS

• Two separate addressing schemes
  – IP addresses and ATM end-points

• Complexity of two sets of protocols
  – Supporting both IP and ATM protocols
ATM Today

• Still used in some contexts
  – Some backbones and edge networks
  – But, typically the circuits are not all that dynamic
  – E.g., ATM circuit used as a link for aggregated traffic

• Some key ideas applicable to other technologies
  – Huge body of work on quality of service
  – Idea of virtual circuits (becoming common now in MultiProtocol Label Switching)
Differentiated Services
Differentiated Services in IP

- Compromise solution for QoS
  - Not as strong guarantees as per-circuit solutions
  - Not as simple as best-effort service

- Allocate resources for classes of traffic
  - Gold, silver, and bronze

- Scheduling resources based on ToS bits
  - Put packets in separate queues based on ToS bits

- Packet classifiers to set the ToS bits
  - Mark the “Type of Service” bits in the IP packet header
  - Based on classification rules at the network edge
Example Packet Classifier

• Gold traffic
  – All traffic to/from John Adam’s IP address
  – All traffic to/from the port number for DNS

• Silver traffic
  – All traffic to/from academic and administrative buildings

• Bronze traffic
  – All traffic on the public wireless network

• Then, schedule resources accordingly
  – E.g., 50% for gold, 30% for silver, and 20% for bronze
Real Guarantees?

- It depends…
  - Must limit volume of traffic that can be classified as gold
  - E.g., by marking traffic “bronze” by default
  - E.g., by policing traffic at the edge of the network

- QoS through network management
  - Configuring packet classifiers
  - Configuring policers
  - Configuring link schedulers

- Rather than through dynamic circuit set-up
Example Uses of QoS Today

• Virtual Private Networks
  – Corporate networks interconnecting via the Internet
  – E.g., IBM sites throughout the world on AT&T backbone
  – Carrying VPN traffic in “gold” queue protects the QoS
  – Limiting the amount of gold traffic avoids overloads
  – Especially useful on the edge link to/from customer

• Routing-protocol traffic
  – Routing protocol messages are “in band”
  – So, routing messages may suffer from congestion
  – Carrying routing messages in the “gold” queue helps

• Challenge: end-to-end QoS across domains… 😞