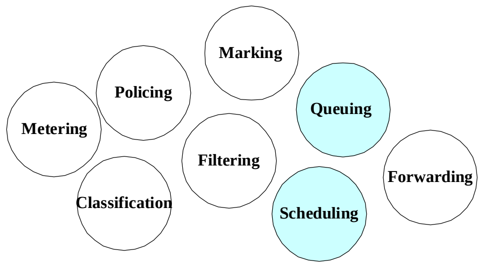
S−38.180 Quality of Service in Internet

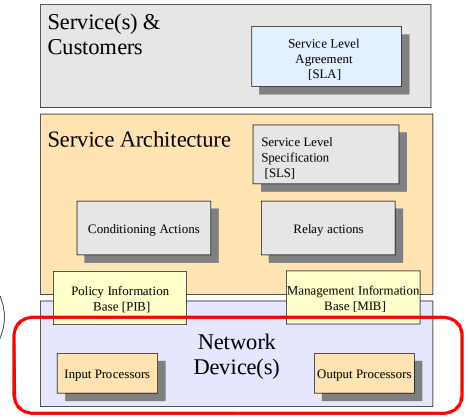
Lecture II: Mechanisms for Traffic Management II

19.9.2002

Today’s Topic

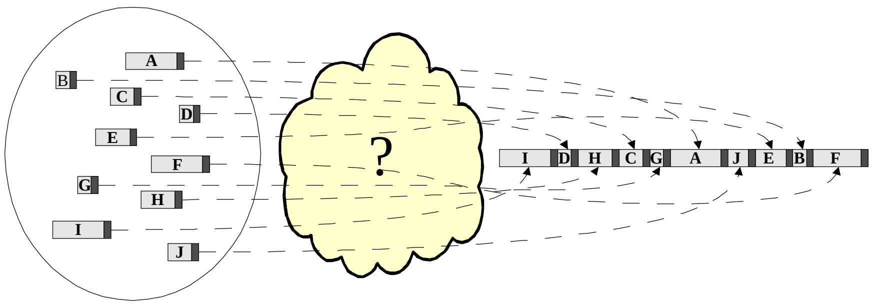
- First part of this lecture is about functional mechanisms which can be found from the output processors of network devices





Scheduling

- Task of a scheduler is to decide the order of packets which are transmitted from the queue

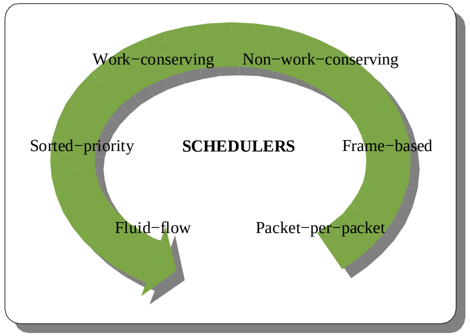


Scheduling

- Selecting the order of packets means that resource sharing is controlled with predefined policy.
- Policy defines the amount of resources which are allocated to the connections / classes for which single packets belong to.
- One end in this continuum is that predefined amount of resources are allocated to the connections.
- Other end is that no allocation is done and resources are shared on the basis of the need

Scheduling

- There are vast amount of schedulers developed for different purposes
- Generally they can be divided into categories of
  – Work−conserving vs non−work−conserving
  – Time−based vs frame−based
  – Continuous vs packetized
  – Priority vs no priority

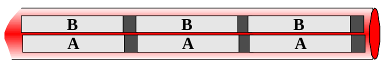


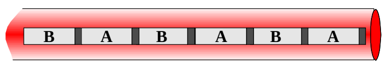
Scheduling

- **Conservation of work** means that scheduler is executing its task as long as it has some work to do.
- Technically this means that there are packets in the queue which has to be sent into the link before scheduler can take a break i.e. change to the idle state.
- Non−work conserving scheduler can idle even though it has packets in the queue.
- Why we would want to have non−work conserving scheduler ?
- Conservation of work means that packets are sent to the link even though receiver would prefere them to come a little bit later.
- This can happen with real−time applications which send packets with constant time intervals. However, network can multiplex them so that they form bursts. Non−work conserving scheduler may delay packets so that intervals structure is maintained throughout the network.

Scheduling

- **Continuous time**
  – Scheduling decissions and calculations are done based on continuous time units
  – Fluid−Flow modeling − packets are infinitesimally small
  – Assumes that number of packets could be served on same time (not possible)
- **Packetized**
  – Scheduling decissions and calculations are based on packet per packet analysis
  – Distorts fluid flow model





Scheduling

- **Time based scheduling**
  – Uses either arrival time or finishing time as a criteria for ordering
  – Time may be virtual or real−time depending on scheduler time
  – Virtual time is usually finishing time in ideal scheduler i.e. scheduler which is not packetized
- **Frame based scheduling**
  – Uses fixed frame which is partitioned for the scheduled packets based on their weights.
  – During a rotation,
    - If there are enough tokens (partition + left overs), then packet is served.
    - Otherwise tokens are added for the next round.
  – A number of packets may be served from a single class if frame is big.

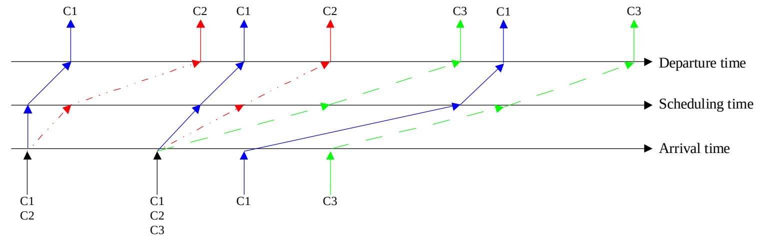
Scheduling

- Scheduling can happen:
  - **Within one queue**, sorting packets inside queue to appropriate transmission order
  - **Between several queues**, dispatching head of line packets from different queues
  - **Hierarchically over several schedulers**, combination of previous ones
- Many of scheduling algorithms can be used to produce QoS in each of these cases

Scheduling

- **First Come First Served** (FCFS) is prevalent scheduling method in routers.
- FCFS uses arrival time information as sorting criteria for packet dispatching.
- FCFS is not able to offer any QoS as arrival time is the only parameter that has influence to the order of packets.

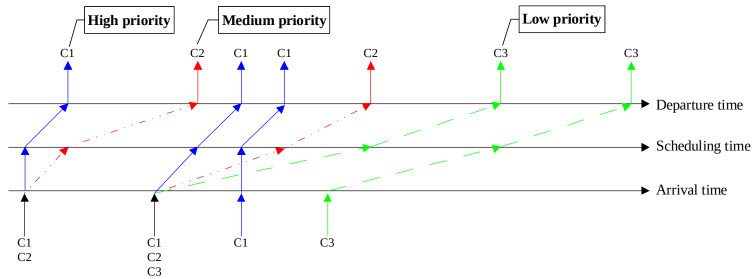
Class	Arrival Time	Service Time
C1	0	1
C2	0	3
C1	3	1
C3	3	3
C2	3	2
C1	5	1
C3	7	3



Scheduling

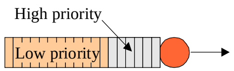
- **Simple priority** scheduler extends FCFS to be able to distinguish between more and less important traffic.
- Packets are ordered first based on their priority and second on their arrival time.

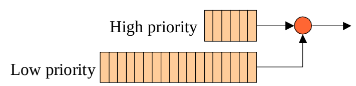
Class	Arrival Time	Service Time
C1	0	1
C2	0	3
C1	3	1
C3	3	3
C2	3	2
C1	5	1
C3	7	3



Scheduling

- Prioritized ordering may lead to starvation of resources in low priority classes if traffic in high priority classes is not limited.
- This can be accomplished by using
  - Connection admission control
  - Over provisioning
  - Rate control
  - Modifying priority scheduler to take class rates into account (token based operation)

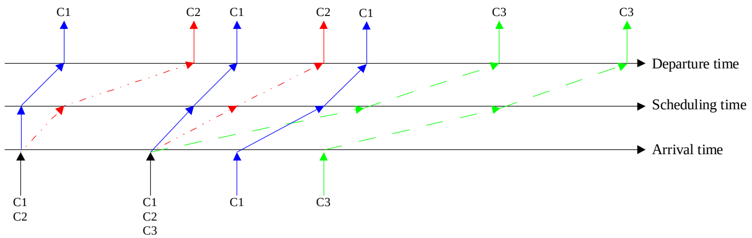




Scheduling

- **Deadline based** scheduling schemes (e.q. Earlies Due Date) are based on the calculation of finishing time (i.e. time when a packet would have been transmitted if it arrived to empty system).
- Packets are transmitted on the order of finishing times.

Class	Arrival Time	Service Time
C1	0	1
C2	0	3
C1	3	1
C3	3	3
C2	3	2
C1	5	1
C3	7	3

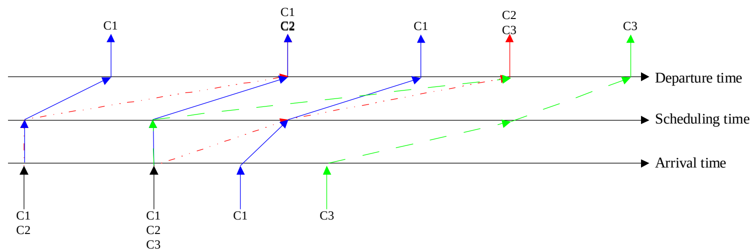


Scheduling

- **Generalized Processor Sharing** is ideal fair queueing algorithm which is based on fluid flow model.
- GPS provides service to the individual connections based on their weights.
- GPS is work conserving scheduler and thus distributes excess capacity to connections which are able to utilize it.

Class	Arrival Time	Service Time
C1	0	1
C2	0	3
C1	3	1
C3	3	3
C2	3	2
C1	5	1
C3	7	3

Weights are all equal



Scheduling

- Disadvantages of GPS are:
  - Departures from GPS are colliding which makes the use of GPS based scheduler impossible
    - However it may be used as backgroud scheduler if collisions are resolved in some manner
  - Heavy calculation of departure times
    - Departure time of every packet in scheduler changes whenever a packet arrives or departs the scheduler

Scheduling

- Advantages of GPS are:
  - Fairness which it provides for the sharing connections

$$\frac{|Service(t,\ t+\Delta t)|_i}{|Service(t,\ t+\Delta t)|_j} \geq \frac{Weight_i}{Weight_j}$$

  - Strict delay bound caused by scheduling when traffic is constrained by a token bucket of token rate *r* and bucket depth *b*

$$Service\ rate\ for\ connection\ i:\ r_i \geq \frac{Weight_i}{\sum_j Weighth_j} \cdot LinkRate$$

$$Delay\ for\ connection\ i:\ D_i \leq \frac{b_i}{r_i}$$

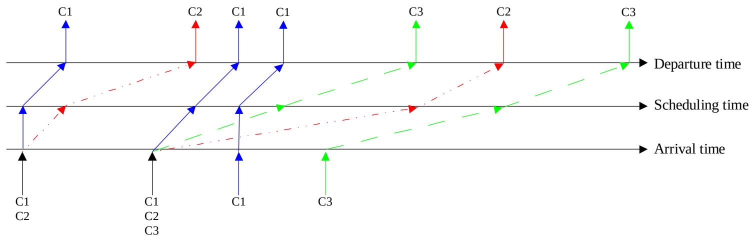
Remember these results were derived from the assumption that packets flow like fluid through the system i.e. there would be a dedicated link with capacity ***r*** between endpoints.

Scheduling

- **Packetized Generalized Processor Sharing** is packet per packet approximation of GPS scheduling.
- Most prevalent implementation of PGPS is weighted fair queueing (WFQ)
- WFQ uses calculation of finishing time in corresponging GPS system as a criteria for sorting the packets.

Class	Arrival Time	Service Time
C1	0	1
C2	0	3
C1	3	1
C3	3	3
C2	3	2
C1	5	1
C3	7	3

Weights are all equal



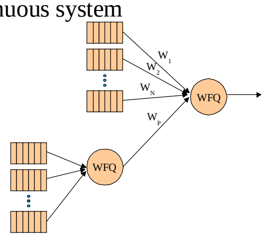
Scheduling

- Delay bound of WFQ system differs the one of GPS system with two extra components:
  - $\frac{(K-1)L_{max}}{r_i}$ which represents extra delay caused if packet arrives a moment later it would have been served in corresponding GPS system. L is the maximum packet length and K is the number of hops.
  - $\sum_{m=1}^{K}\frac{L_{max}}{R^m}$ which represents the fact that packets are served one by one. In backlogged system, packet must wait that previous packet is served, before it gets to be scheduled.

$$D_i \le \frac{b_i}{r_i} + \frac{(K-1)L_{max}}{r_i} + \sum_{m=1}^{K}\frac{L_{max}}{R^m}$$

Scheduling

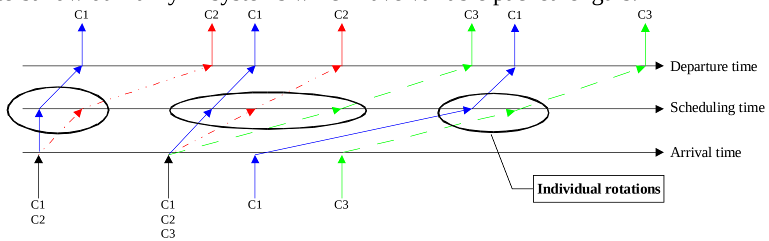
- WFQ scheduling has number of variant which aim:
  - Ease the calculation of finishing time in corresponding GPS system
    - By replacing the idle time function with the finishing time of packet which was in service when backlogging packet arrived to the system.
    - By replacing the time calculation with frame based operation
  - Make the fairness packetized system as good as continuous system
  - Allow hierarchical construction of service



Scheduling

- **Weighted Round Robin** is popular implementation of frame based fair queueing.
- WRR uses a rotation where each individual connection is served in relation of their weights.
- Service is usually based on packets, which causes WRR to be not able to distribute bandwidth fairly in systems which have variable packet lengths.

Class	Arrival Time	Service Time
C1	0	1
C2	0	3
C1	3	1
C3	3	3
C2	3	2
C1	5	1
C3	7	3



Scheduling

- **Deficit Round Robin** is extention of WRR which takes account the packet size
- DRR uses a rotation where a frame of *N* bits is divided to indivivual connections in relation to their weights (quantums).
- Quantums which individual connections receive serve packets
  - If the quantum is small, many rotations are required to serve backlogged connection
  - If the quantum is big, many packets can be served on one rotation
- DRR uses special counter for each backlogged connection which stores the information of received bits.
  - If connection gets to non backlogged state counter is cleared

Scheduling

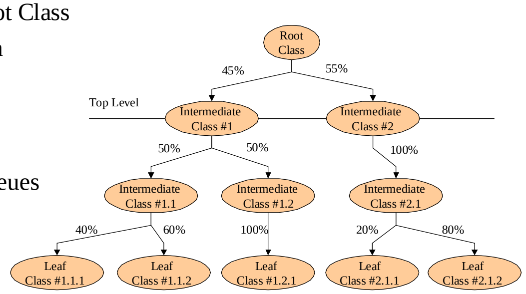
- **Class Based Queueing** is one form hierarchical scheduling
  - In CBQ scheduling is divided into two cases:
    - Unregulated: When a class is scheduled by **general scheduler**
    - Regulated: When a class is scheduled by **link share scheduler**
  - Class is regulated in situations when network is persistently contended and class has run over its limits
- Actual implementation of scheduling is uniform
  - Both schedulers manipulate HOL packets time to send information which is then examined by actual dispatcher.
- CBQ uses different variants of round robin schedulers as a general scheduler
- Link share scheduler is based on general rules supplied by user

Scheduling

- Advantage of CBQ is that scheduling during contention is easily manipulated to produce outcome which is not only based on time and priority information
- Disadvantage is that CBQ requires a lot of processing time when there are a lot of independent connections / classes

Scheduling

- Link sharing guidelines are based on tree like structure
  - Link resources are on Root Class
  - Intermediate Classes form logical groupings
    - Organisations
    - Protocols
  - Leaf classes are actual queues with distinct traffic

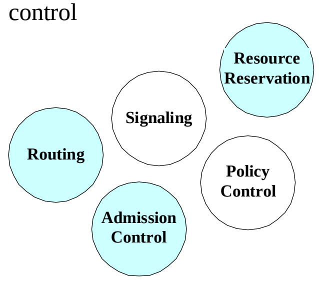


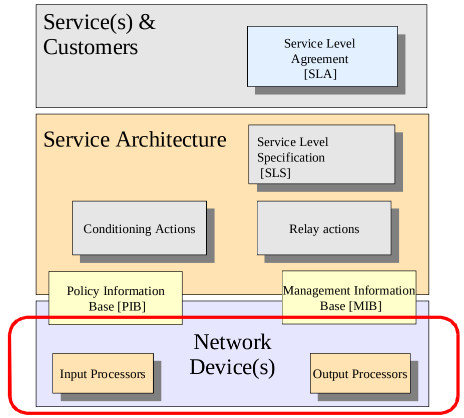
Scheduling

- CBQ has concept of **borrowing:**
  - If class has run over its limit but it has parent class which is not over its limit, it may borrow capacity from the parent
  - Borrowing may be limited to some level in link sharing tree (Top Level)
- Formal definition between regulated and un regulated follows from borrowing:
  - Class is unregulated if:
    - It is under its limit
      or
    - It has parent below Top Level which is under its limit

Today's Topic

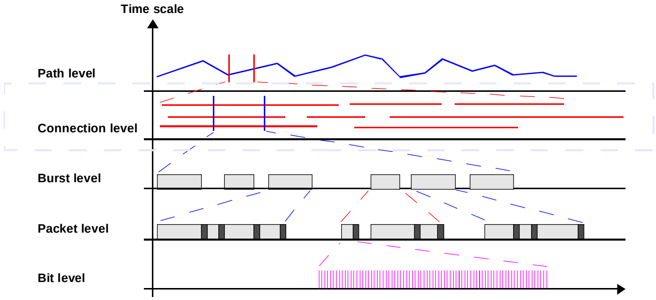
- Second part of this lecture is about control plane of Internet routers − especially routing and admission control





Connection Admission Control

- **Connection Admission Control (CAC)** is a functionality which controls the usage of resources from the **connection** level
  - Accepting and rejecting **connections**



CAC

- **CAC is applicable only if we have connection oriented network protocol**
  - IP is inherently **not**
  - IP **can be** used also in connection oriented manner
    - Requires **signaling** system to transfer connection parameters from the end system to the network
    - If the goal is **resource reservation** (IntServ), we need to define per connection
      - QoS requirements
      - Traffic profile
    - Otherwise just the **awareness** of a new traffic flow is indicated

CAC

- Why CAC

Dedicated resources

- – If resource reservation is used, we need to have ability to decide whether or not there is **enough resources** for the new connection
  - Link has capacity ***c***
  - Previous connections have used capacity ***x***
  - New connection requires capacity ***y***
  - Is $x+y<c$

Shared resources

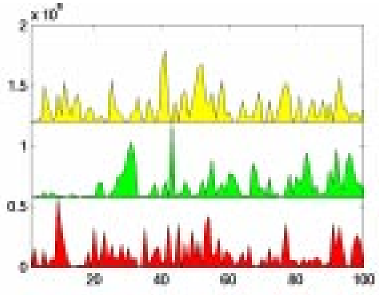
- – Even if there is no reservation we may want to **limit the number** of connections on the link
  - Link is provisioned for 250 simultanious connections
  - More than that would lower the experienced quality

CAC

- CAC operates usually on the **end to end** manner (hop by hop)
  - – **Path** for the communication is determined
    - On demand computation
    - Pre computed routes for each destination
  - – Decission logic requires information about
    - **Available capacities** on the links
      - – Link capacity
      - – Reserved capacity
  - – **Resource requirements** of the new connection
    - QoS requirements
    - Traffic profile

CAC

- **Resource reservation** is partiotioning of link and buffer capacity to individual connections
- Reservation can be based on lossless **peak rate allocation**
  - – Link capacity is fragmented to pieces size of the maximum capacity of the connection
  - – Overall utilization of the link is poor when variation on the sending rate is high

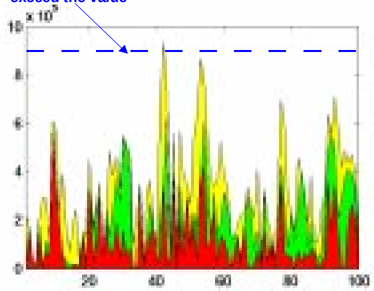


Aggregate capacity 1.8Mpbs –> there is no room for extra connections on this 2Mbps link

CAC

- Reservation based on **statistical multiplexing** of original traffic streams yields much lower consumption of resources
  - – Risk of overload causes potential packet loss
    - Actual capacity exceeds the equivalent capacity
    - Packet losses can be eliminated with buffering
  - – Higher loss probability (p)
    - Lower equivalent capacity
    - Higher **multiplexing gain**

Equivalent capacity is the aggregated capacity of multiplexed connections with probability *p* to exceed the value



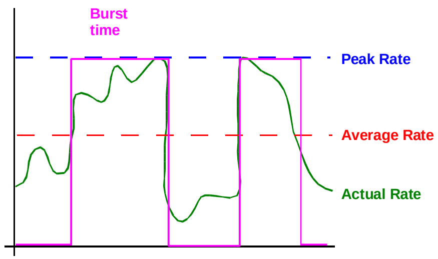
Aggregate capacity 0.95Mpbs –> there is room for extra 3–5 connections on this 2Mbps link

CAC

- **Problem is to know how statistically multiplexed traffic streams interact**
  - What is the probability of overflow i.e. probability that traffic streams are in their peak rate mode simultaneously
  - Chicken and egg problem
    - How to know traffic profile well enough to make appropriate resource reservation when traffic is not even yet began to be generated

CAC

- **Traffic profile** is usually declared in form of token bucket filter
  - Morning lecture slide 17
- CAC algorithms are generally use
  - Token generation rate (*r*)
  - Peak rate (*p*) calculated from the token bucket parameters
    - Worst case expectation
    - on/off bursts from zero to bucket size divided by burst time



CAC

- CAC algorithms are based on the different types of approximations for the actual event
  - **Distribution models** for the traffic
    - On/Off
    - Fluid flow
  - **Limit theorems** used to calculate boundary conditions (loss probabilities)
  - **Methods to update aggregate** traffic profile
    - Stored individual connection profiles
    - Measured estimate for the aggregate

CAC

- Most of the CAC algorithms are based on the usage of equivalent capacity
  - Ways to estimate or calculate this capacity differ
    - **Parametric** approaches uses declared traffic parameters
      - Token bucket parameters
        - » Generation rate (*r*)
        - » Bucket size (*b*)
    - **Measurement based** approaches use sampling of actual link utilization with some memory
      - Exponentially weighted moving average
      - Time sliding window

CAC

- **Simple sum**
  - Algorithm uses parametric representation of offered traffic
  - Admission is based on sum of existing connections rate parameters (v) and rate of the new connection (r)

$$\boxed{v + r_\alpha < C}$$

  - Note: link is loaded completely by rate parameters. Bursts taken care with buffering in each stage of the network

- **Measured sum**
  - Algorithm uses measurement based estimate for capacity $(\hat{v})$
    - Corrects the error of false traffic parameters
      - Higher utilization

$$\boxed{\hat{v} + r_\alpha < \delta C}$$

  - Note: measurement based estimate is prone with errors. Utilization target (δ) is used to allow room for measurement errors.

CAC

- **Bounded capacity and delay**
  - Combination of rate and delay admission control
    - Measured sum control for rate

$$\boxed{\hat{v} + r_\alpha < \delta C}$$

    - Measured delay control for delay

$$\boxed{\hat{D} + \frac{b_\alpha}{C} < D}$$

- **Equivalent bandwidth**
  - Combines Hoeffding upper bound $(\hat{C}_H)$ estimate on bandwidth and peak rate of a new connection (p_α)

$$\boxed{\hat{C}_H + p_\alpha \leq \delta C}$$

  - Peak rates are derived from the token bucket parameters by spreading the bursts over fixed time interval (t)

$$\boxed{p_\alpha = r_\alpha + \frac{b_\alpha}{t}}$$

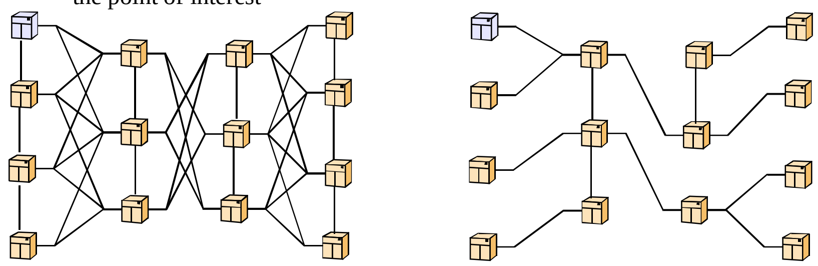
CAC

- **Hoeffding bound** gives upper bound for the tail distribution of random variables
  - Peak rates of the connections
- Quality of upper bound is controlled by risk factor
  - Packet loss probability in bufferless case
- Measured capacity is now compensated individual connections peak rates
  - Fluctuation of the estimate is not so critical

$$\boxed{\hat{C}_H = \hat{v} + \sqrt{\frac{\ln(1/\epsilon)\sum_{i=1}^{n}(p_i)^2}{2}}}$$

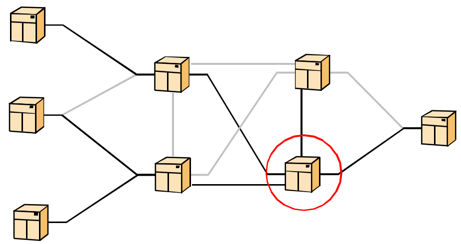
Routing

- **Routing** is a matter of finding a path (usually shortest possible) between any two networks in the whole Internet
  - Finding a path means that mess of networks is organised in to **<u>tree</u>** like structure representing neccesary links to reach all possible networks from the point of interest



Conventional IP routing

- Nature of conventional shortest path algorithms cause traffic to be aggregated to lowest cost links
  - Centralises traffic into hot spots in the network
  - Large amount of links are left to idle while few are overloaded

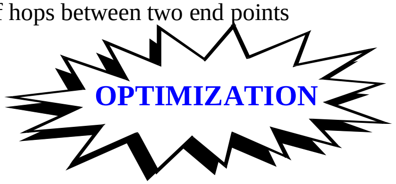


Conventional IP routing

- Construction of routing tables is responsibility of routing protocols
- Routing protocols can be divided based on their usage (scalability):
  - Interior Gateway Protocols: Running inside one autonomous system
    - OSPF, IS−IS, RIP, IGRP ...
  - Exterior Gateway Protocols: Running between autonomous systems
    - BGP, IDPR
- Routing protocols implement neccesary optimization algorithms to find shortest paths between end points:
  - Distance vector (RIP, IGRP, BGP)
  - Link−state (OSPF, IS−IS)

Interior Gateway Protocols

- Possibility to **full knowledge** of domain characterstics
  - Capacities
  - Delays
  - Offered traffic
  - Preferences
- Routing normally based on the shortest path
  - Least amount of hops between two end points



Exterior Gateway Protocol

- Domain characteristics relatively **unknown**
  - Knowledge is based on agreements and policies
  - Real−time data is rarely distributed
  - Reachability information (distance vector features)
  - Support for QoS ???

OSPF

- Operation goes through four phases:
  - **One:** Neighbours are aquired and maintained in adjacency by hello packets
    - Adress and cost information is gathered
    - Heartbeat of particular link (failure detection)
  - **Two:** Link−state advertisement (LSA) packets are formed based on information gathered by hello packets
  - **Three:** LSA packets are flooded into the network and received from the network to construct topology database
  - **Four:** Least cost routes are calculated to every other router in the network

OSPF

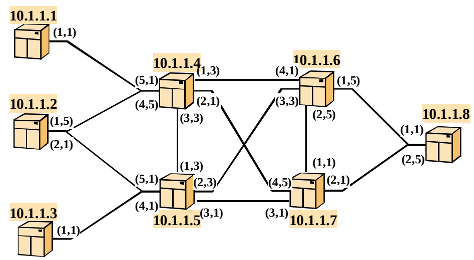
- Link−state advertisement packet contains:
  - Header part identifying
    - Advertising router
    - LSA type
      - Certain LSA types may have additional header information
  - LSA information part (depending on LSA type)
    - Link information and metrics
    - Network information and attached routers

OSPF

- **Topology database** is initially copied from one of the adjacent neighbours
- Updates to initial database are received and sent by flooding
  - Every adjacent neighbour receives flooded LSAs and process them to topology database.
  - After processing LSAs are repacked and flooded ahead
  - Every router in the net receives a copy of original LSA
- ’Full’ knowledge of network devices and links
- Calculation of **routes** is based on Dijkstra algorithm and information in topology database

OSPF

- Metric used in route computation is based on information received in LSAs
  - It set by
    - Network administrator to indicate preference of particular link
    - Automatically as a form of computational intelligence in a router



Routing in general

- **Optimize**
  – Find best possible solution to the problem in hand
    - Minimum cost
    - Shortest path
    - Maximum bandwidth
  – Optimal
  – One solution
  – Full depth search
- **Constrain**
  – Find possible solution to the problem in hand
    - Delay less than X
    - Free capacity larger than Y
  – Usually suboptimal
  – Many possible options
  – Limited search

Routing

- Conventional IP routing is based on connectionless network philosophy
  – Each packet is independent and complete unit
  – Routing is decoupled from the packet streams
  – Pure optimization problem
- Differentiated Services is based on connectionless network philosophy
  – Routing is decoupled from the packet streams
  – Multi variable constraint and optimization problem
- Integrated Services is based on connection oriented network philosophy
  – Path is coupled into the packet streams through state information in the routers
  – Multivariable constraint problem
- Multiprotocol label switching is based on connection oriented philosophy
  – Path is coupled into packet streams through state
  – Multivariable constraint problem

QoS Routing problems

Link constrains:
- Capacity
- Buffer space

Path constrains:
- Delay
- Cost

Unicast QoS routing	
Basic routing problems	**Composite routing problems**
Link optimization routing (bandwidth optimization)	Link constrained link optimization routing (bandwidth constrained – buffer optimization)
Link constrained routing (bandwidth constrained)	Link constrained path optimization routing (bandwidth constrained – least delay)
	Multilink constrained routing (bandwidth and buffer constrained)
	Link constrained path constrained routing (bandwidth and delay constrained)
Path optimization routing (least cost)	Path constrained link optimization routing (delay constrained bandwidth optimization)
Path constrained routing (delay constrained)	Path constrained path optimization routing (delay constrained least cost)
	Multipath constrained routing (delay and delay jitter constrained)

NP

Routing Strategies

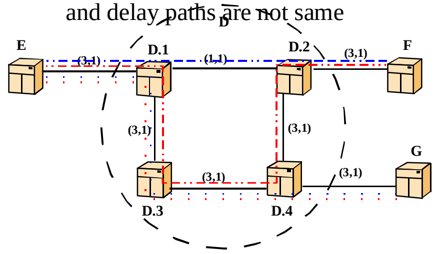
- **Source routing:**
  – Centralized routing decission
    - Source computes route through the network
  – Biggest problems:
    - Knowledge of the global state is only approximate (communication delay)
    - Size of the state base is huge (all links and nodes and their attributes)
- **Distributed routing:**
  – Path computation is distributed to all routers between source and destination (distance vectors)
  – Biggest problems:
    - State change in the network may cause loops which can not be easily solved
    - Construction of distributed heuristics for multiple attributes is not straight forward

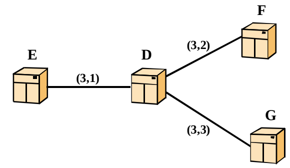
Routing Strategies

- **Hierachical routing:**
  - State base is shrinked with clustering and aggregation
    - Network is partitioned to clusters reflecting areas of common policy
    - State of the clusters is aggregated at the boundaries
  - Approximates distributed source routing
    - Each cluster is individually source routed
- Biggest problems:
  - Aggregation causes imprecision which causes paths to be only semi−optimal
  - Formation of aggregate metrics is not straight forward

Problems with multiple metrics

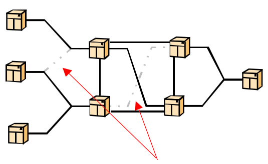
- Metric aggregation:
  - E−>G is correct as largest bandwidth path is equal to lowest delay path
  - E−>F is incorrect as bandwidth and delay paths are not same
- Path selection:
  - Link which do not qualify by link constraint should be <u>pruned</u> before optimization with path constraint

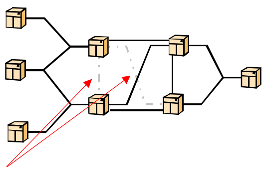




Pruning

- Metric 1: Free capacity greater than X bps
- Metric 2: Delay less than Y ms

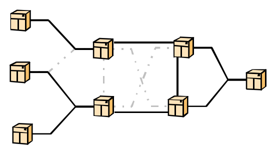


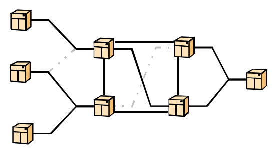


Links which do not have resources to fulfill constraints of the metric are removed (pruned) from the graph

Constraint based routing

- Optimization is used to find path from the reduced topology
  - Cost
  - Delay
- Optimization can be done straight after pruning of first metric
- Lowest delay path is searched
- Requires check whether delay constraint is held





What is the difference

- Pruning constraint 1: Capacity
- Pruning constraint 2: Delay
- Optimization with <delay>

Delay is path constraint which has very little meaning on link by link basis. Therefore it has to be broken down to link constraints.

Easily NP complete problem...

- Pruning constraint 1: Capacity
- Optimization with delay
- Sanity check
  - Delay less than constraint 2

QoS support in OSPF

- Traditional QoS support for OSPF is based on Type of Service paradigm
  - IPv4 TOS makes possible to indicate routing preference
    - Normal service (0000)
    - Minimize monetary cost (0001)
    - Maximize reliability (0010)
    - Maximize throughput (0100)
    - Minimize delay (1000)
  - OSPF TOS has 8 bit numerically encoded QoS support
- IPv4 TOS offers selection of one routing attribute
- OSPF uses separate routing table for every TOS value
- Routing table is calculated from the subset of topology database indicating only links capable of offering service defined by TOS

But nobody uses TOS so there is no actual support for it in the network !!!

Extended QoS Support for OSPF

- Generalisation of QoS concept
  - QoS routing is decoupled from the TOS values of the IP packet
    - Routing decission is done in a **connection oriented way** –> signaling
  - Metrics are selected to reflect dynamic nature of network
    - Link available bandwidth: Current available bandwidth meaning unallocated bandwidth
    - Link propagation delay: Makes possible to differentiate between satellite and terrestial links

This is matters of Integrated Services !!!

Extended QoS Support for OSPF

- Middle way:
  - QoS routing is coupled to the DSCP values of the IP packet
  - Metrics are selected to reflect dynamic nature of network
    - Link available bandwidth: Current available bandwidth meaning subtraction of measured average link utilisation from the link capacity
    - Link propagation delay: Makes possible to differentiate between satellite and terrestial links