# S-38.3180 Quality of Service in Internet

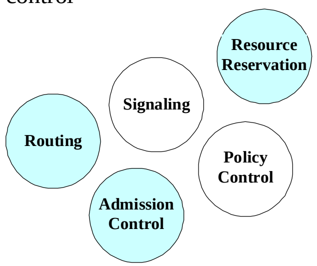
## Lecture II: Control Path Processing

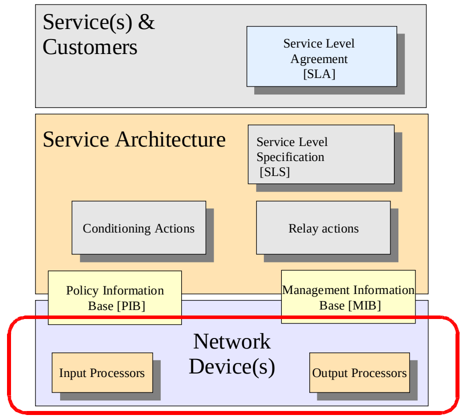
9.11.2006



# Today's Topic

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

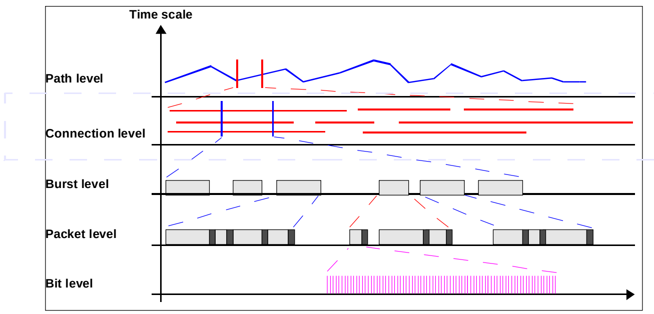





# Connection Admission Control

- **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
  - 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***
  - 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

**Dedicated resources**

**Shared resources**



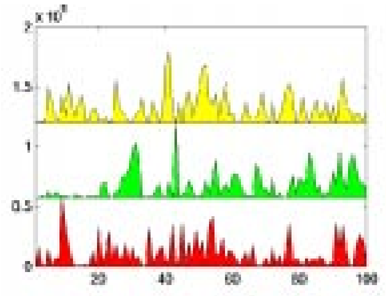
# CAC

- 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



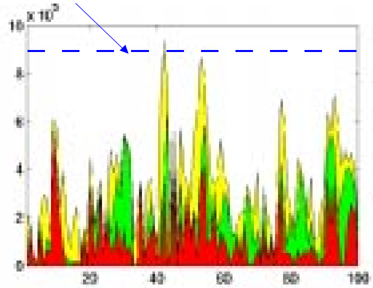
Aggregate capacity 1.8Mpbs -> there is no room for extra connections on this 2Mbps link



# 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

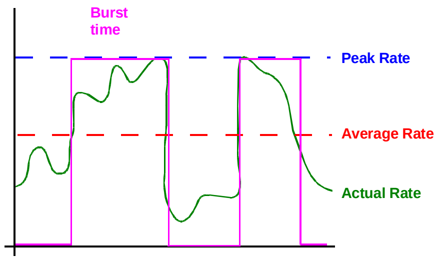
# CAC

- **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 ( $\boldsymbol{v}$ ) and rate of the new connection ( $\boldsymbol{r}$ )

$$v + r_\alpha < C$$

  - 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 $(\hat{\boldsymbol{v}})$
    - Corrects the error of false traffic parameters
      - Higher utilization

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

  - Note: measurement based estimate is prone with errors. Utilization target $(\delta)$ 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

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

    - Measured delay control for delay

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

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

$$\hat{C}_H + p_\alpha \le \delta C$$

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

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



# CAC

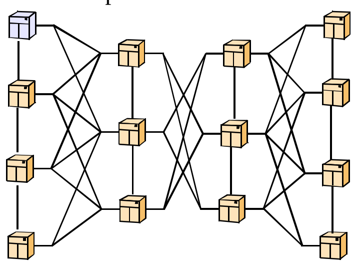
- **Hoeffding bound** gives upper bound for the tail distribution of random variables
  - Peak rates of the connections
- Quality of upper bound is controlled by risk factor
  - Packet loss probability in bufferless case
- Measured capacity is now compensated individual connections peak rates
  - Fluctuation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

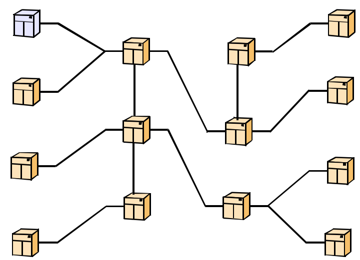
$$\hat{C}_H = \hat{v} + \sqrt{\frac{\ln(1/\epsilon)\sum_{i=1}^{n}(p_i)^2}{2}}$$



# Routing

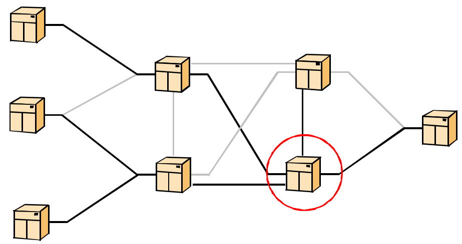
- **Routing** is a matter of finding a path (usually shortest possible) between any two networks in the whole Internet
  - Finding a path means that mess of networks is organised in to **tree** like structure representing neccesary links to reach all possible networks from the point of interest





# 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 <u>hello</u> packets
    - Adress and cost information is gathered
    - Heartbeat of particular link (failure detection)
  - **Two:** <u>Link-state advertisement</u> (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 <u>topology database</u>
  - **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



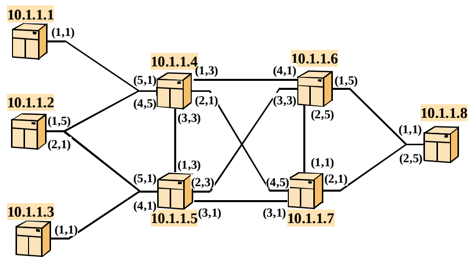
# OSPF

- **Topology database** is <u>initially</u> copied from one of the adjacent neighbours
- <u>Updates</u> 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

| | Unicast QoS routing | |
|---|---|---|
| | **Basic routing problems** | **Composite routing problems** |
| Link constrains:<br>• Capacity<br>• Buffer space | **Link optimization routing (bandwidth optimization)**<br>**Link constrained routing (bandwidth constrained)** | **Link constrained link optimization routing (bandwidth constrained - buffer optimization)**<br>**Link constrained path optimization routing (bandwidth constrained - least delay)**<br>**Multilink constrained routing (bandwidth and buffer constrained)**<br>**Link constrained path constrained routing (bandwidth and delay constrained)** |
| Path constrains:<br>• Delay<br>• Cost | **Path optimization routing (least cost)**<br>**Path constrained routing (delay constrained)** | **Path constrained link optimization routing (delay constrained bandwidth optimization)**<br>**Path constrained path optimization routing (delay constrained least cost)**<br>**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



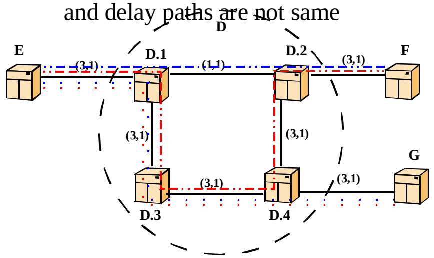
# Routing Strategies

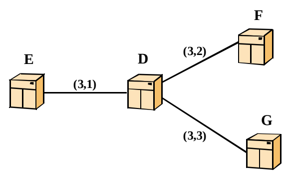
- **Centralized routing:**
  - Routing is performed outside the network
    - In centralized server
    - In distributed route servers
  - Routes are distributed into the network as a static routes
- Biggest problem:
  - Local failures have long impact time to routes
    - Network should fallback to conventional distributed routing in case of local failures



# 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

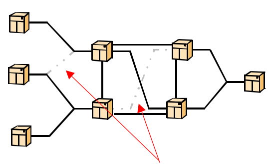


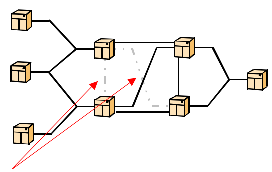






# Pruning

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

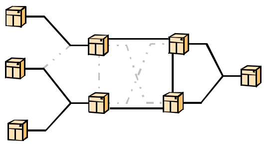




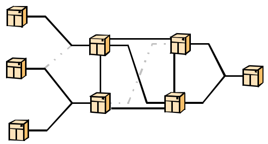
Links which do not have resources to fulfill constraints of the metric are removed (pruned) from the graph

# 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