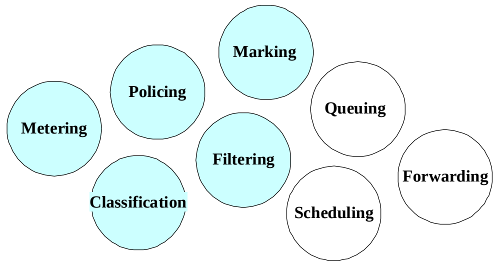
S-38.3180: Quality of Service in Internet

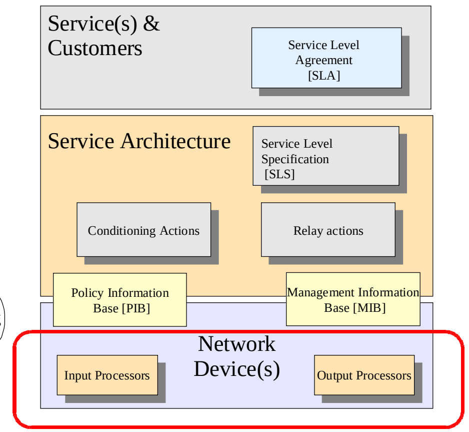
Lecture II: Ingress Traffic Processing

2.11.2006

Today's Topic

- This lecture is about functional mechanisms which can be found from the input processors of network devices





Terminology

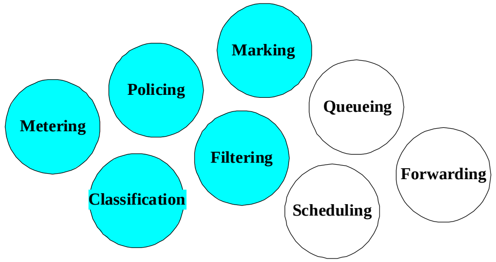
- **Connection**: is dynamically formed reservation of network resources for a period of time.
  - Connection requires a state to be formed inside the network
  - State is a filter defining packets which belong into particular connection and required reservation attributes
- **Flow**: is formed from arbitrary packets which fall within predefined filter and temporal behavior.
  - Packets from one source to same destination arrive to investigation point with interarrival time less than ***t*** seconds.

Terminology

- **Aggregate**: is a group of flows which have same forwarding characteristics and share link resources.
- **Class**: is a group of connections which share same forwarding characteristics.

Input processor

- Input processor of Internet router consists several mechanisms
  - Filtering
  - Classification
  - Metering
  - Policing
  - Marking
  - Shaping



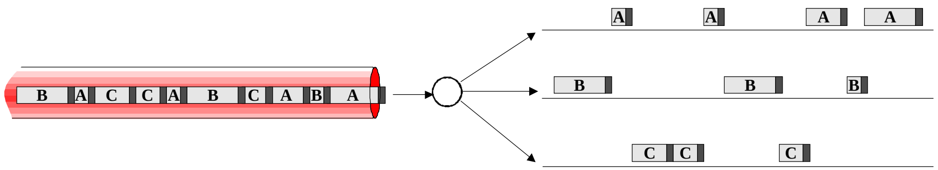
Classification

- Individual connections can be recognized by looking sufficient number of protocol fields.
- This is used in **Integrated Services** architecture.
- IntServ uses reservation protocol for informing the network about fields which should be examined.

- If per connection accuracy is not needed or can not be feasibly implemented is aggregate based operation the answer.
- This is used in **Differentiated Services** architecture.
- Aggregate is based on static filters covering broad range of different connections i.e. aggregating connections to one logical unit

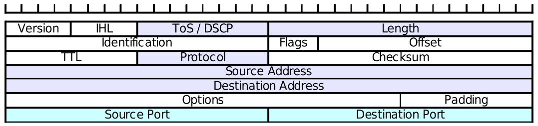
Classification

- Classification is process where packets in the packets stream are separated into ***n*** logically separate packet streams.
- These streams are then treated as separate entities for which different actions are performed
- Separation is based on filters which match packet content to the filtering rules.



Filtering

- Commonly filters are based on IP packet / transport header information
  - IP addresses
  - Protocol information
  - DSCP-field
  - Port information
  - Length information

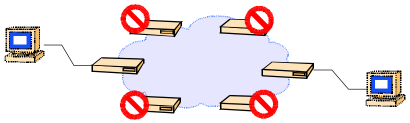


- Generally any fixed block of bits can be used as a filter

- Commonly used notion for filter ->Five tuple = (SourceIP, DestinationIP, Protocol, SourcePort, DestinationPort)

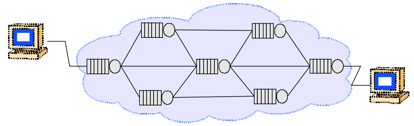
Service Level Management

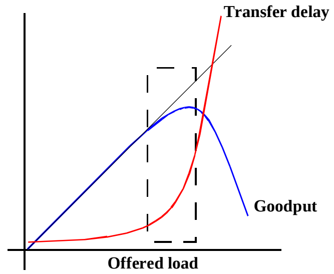
- QoS based networks need careful management
  - How to provision the network so that there will not be unnecessary queuing or packet loss
  - How to control the amount of traffic that gets into the network
- Network level
- Customer level / connection level
- Packet level



Service Level Management

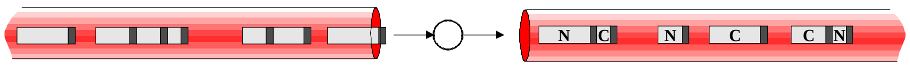
- Overall objective is to offer QoS and/or maximize network throughput
- This requires
  - Limiting user traffic to the level that individual links operate on optimal fashion
  - Individual links can not be fully utilized
    - Unequal capacities
    - Uncertainty of paths
    - Uncertainty of demands





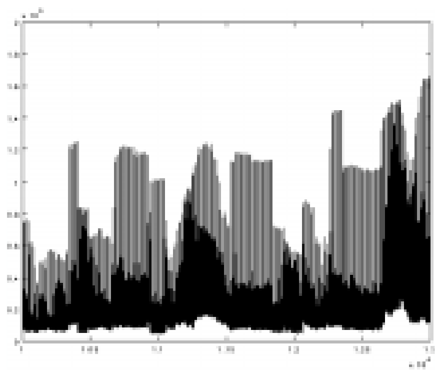
Rate Control

- Task is to decide which user packets should be delivered into the network and on what priority (mark)
  - They do not violate QoS management principles within the network by overloading the network
- Rate control operates in three levels
  - Measures the traffic
  - Compares the measured information to information in user / network policy
  - Executes policy based on comparison results
    - Marking
    - Dropping
    - Shaping



Rate Control

- User traffic process is largely dependent on application which is used.
  - Some applications produce constant traffic stream
    - Fixed size packets
    - Constant interarrival times
  - Other may produce bursts of packets
    - Variable size packets
    - Variable interarrival times



Rate Control

- **Objectives:**
  - **Simple**
    - Easy algorithm
    - Few parameters
  - **Accurate**
    - Actions are correct
    - Actions are transparent
    - Actions are immediate
  - **Predictable**
    - Action are consistent from time to time
- **Requires:**
  - Parametrization of user traffic
    - Either flow level
    - Or Aggregate level
  - This is bound to SLA made with the ISP

Metering

- Packet stream is measured to find out some of the following parameters:
  - *Peak rate* - maximum rate on which user is sending
  - *Sustained rate* - average rate on which user is sending
  - *Burst size* - maximum burst size which user sending on either with peak or average rate
- Actual measurement of information may be based on
  - Continuous time measurement
  - Discrete event analysis
  - Window based analysis

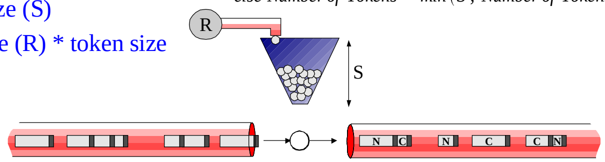
Token Bucket

- Produces information whether arrival rate is more or less than the threshold
- Algorithm is based on
  - Number of tokens in token bucket (in bytes)
  - Arrival time (T_{Now}, $T_{Last\ Arrival}$)
- **Two limiting parameters**
  - Bucket size (S)
  - Token rate (R) * token size

Initial condition:
Number of Tokens = S

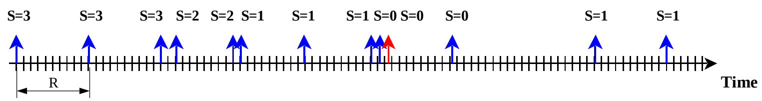
Upon each arrival:

$$Increment = TokenSize \cdot R \cdot (T_{Now} - T_{Last\ Arrival})$$
$$Decrement = PacketLength$$
$$Conformance = Number\ of\ Tokens + Increment - Decrement$$
$$if\ Conformance \geq 0$$
$$then\ Number\ of\ Tokens = \min(S, Conformance)$$
$$else\ Number\ of\ Tokens = \min(S, Number\ of\ Tokens + Increment)$$



Token Bucket

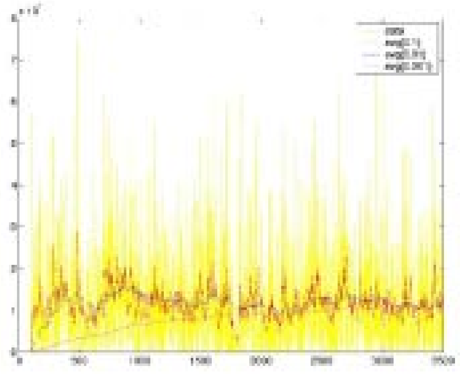
- **In ideal situation**
  - Packets arrive with intervals of token generation rate (R)
  - Packets are size of token
  - Variation of arrivals is compensated with bucket size (S)
    - Allows bursting
- Example:
  - R=10
  - S=3



Packet per packet EWMA meter

- Measures packet stream by using exponentially weighted moving average filter.
  - **Tunable by parameter**
    - Memory (ϵ)

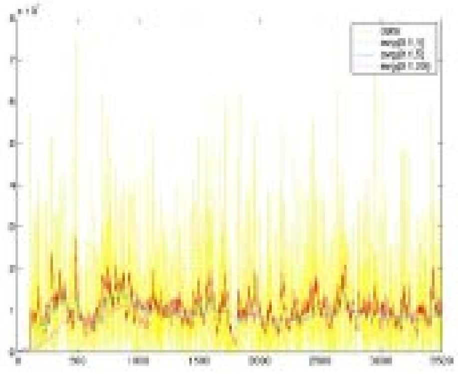
Initial condition:
$avg(0) = 0$
After every packet arrival
$$avg(n+1) = (1-\epsilon)\cdot avg(n) + \epsilon \cdot \frac{PacketLength}{t_{n+1} - t_n}$$



Windowed EWMA meter

- Measures packet stream by using exponentially weighted moving average filter with sampling window.
  - **Tunable by parameters**
    - Memory (ϵ)
    - Sampling interval (ΔT)

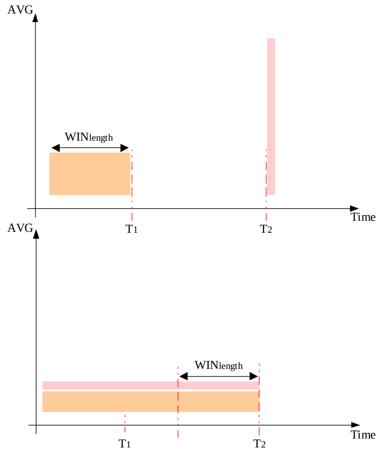
Initial condition:
$avg(0) = 0$
After every ΔT time units
$$avg(t_{n+1}) = (1-\epsilon)\cdot avg(t_n) + \epsilon \cdot bytes\ during\ [t_{n+1}, t_n]$$



Time Sliding Window Meter

- TSW is memory based, windowed average rate estimator
- **Tunable by parameter**
  - Window length

Initial condition:
$avg(0) = 0$
$Win_{length} = C$
$T_{front} = 0$
After every packet arrival:
$Bytes_{TSW} = avg(n)\cdot Win_{length}$
$New_{bytes} = Bytes_{TSW} + PacketLength$
$$avg(n+1) = \frac{New_{bytes}}{T_{now} - T_{front} + Win_{leght}}$$
$T_{front} = T_{now}$



Metering

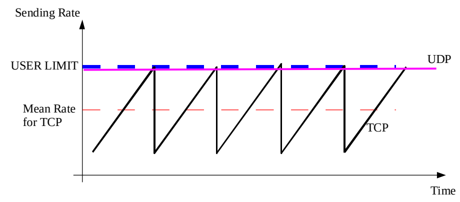
- Based on the measured information a conformance statement is declared
- **Conformance is the observation whether the measured variable is within predefined boundaries.**
  - Customer has contracted rate of ***X*** bps with variation of ***x*** bps
  - Customer has contract of average rate ***X*** bps and peak of ***Y*** bps. He is allowed to send bursts of ***Z*** kB in peak rate.

Conformance algorithms

- **Strict conformance**
  - Packets exceeding contracted rate are marked immediately as non-conforming
- **TSW conformance**
  - Packets exceeding 1.33 times contracted rate are marked as non-conforming
- **Probability conformance**
  - Packets exceeding contracted rate are marked as non-conforming with increasing probability

Rate Control Problems

- Two parallel transport protocols with contradicting control:
  - UDP - with no control
  - TCP - with additive increase exponential decrease rate control
- **Problem:** Metering system cannot easily offer fair service to both TCP and UDP clients in the same system.



Marking

- Marker is used to attach conformance / class information to every packet.
- Marker uses IPv4 TOS/DSCP field to convey information for other processing elements in the network.
  - TOS
    - Prec: 3 bit priority
    - TOS: user preference for routing
  - DSCP
    - Class and precedence