# S−38.180 Quality of Service in Internet

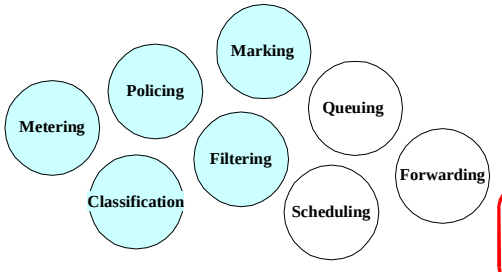
## Lecture I: Mechanisms for Traffic Management I

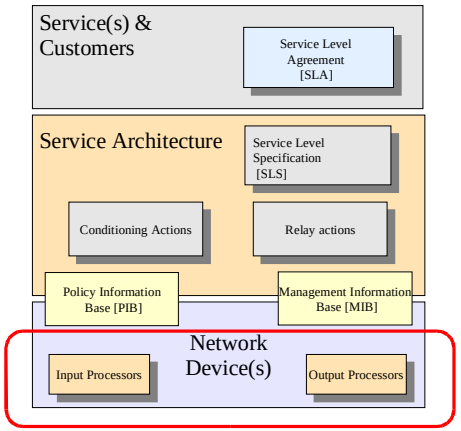
19.9.2002



# Today's Topic

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

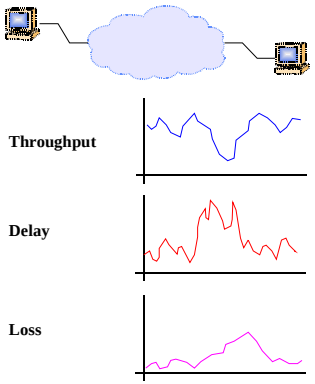





# Internet Service

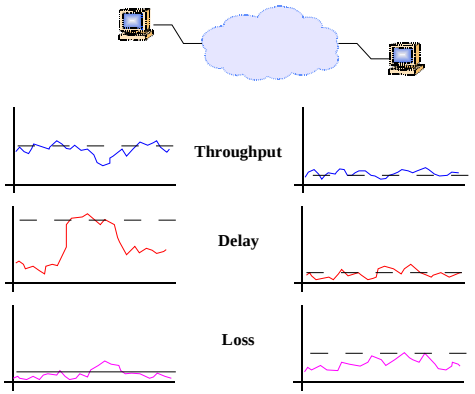
- Internet service is packet delivery service − pretty much like common snailmail service
  - Datagrams are delivered as individual items and they experience service which varies as a function of time






# Internet Service

- By adding QoS, we are aiming to provide differentiation to this situation
- Differentiation can be based on different criterion
  - Usage
  - Money
  - Status



# 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.



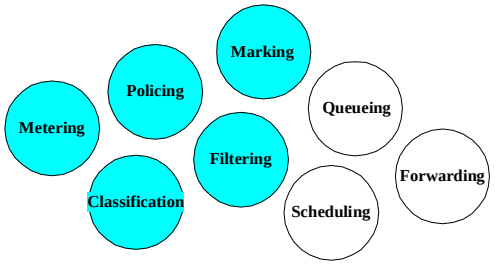
# Terminology

- **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



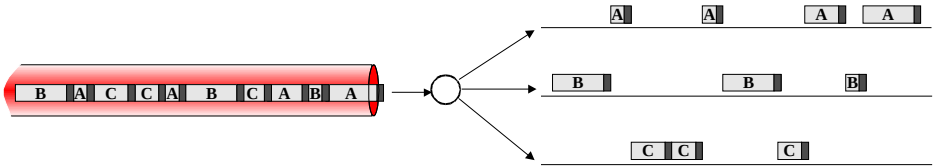



# Classification

- 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 class based operation the answer.
- This is used in Differentiated Services architecture.
- Class is based on static filters covering broad range of different connections i.e. aggregating connections to one logical unit

# Classification

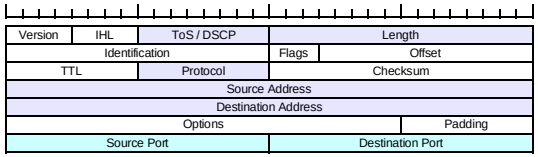
- Classification is process where packets in the packets stream are separated into ***n*** logically separate packet streams.
- These streams are then treated as separate entities for which different actions are performed
- Separation is based on filters which match packet content to the filtering rules.






# Filtering

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

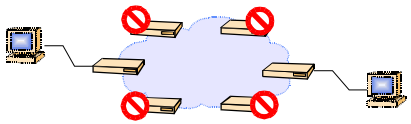



- Generally any fixed block of bits can be used as a filter



# Service Level Management

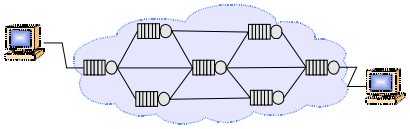
- QoS based networks need careful management
  - How to provision the network so that there will not be unnecessary queuing or packet loss
  - How to control the amount of traffic that gets into the network
- Network level
- Customer level / connection level
- Packet level

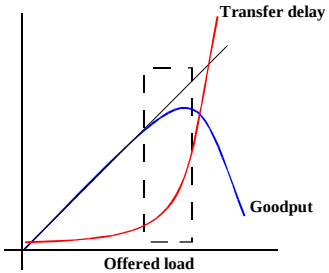




# Service Level Management

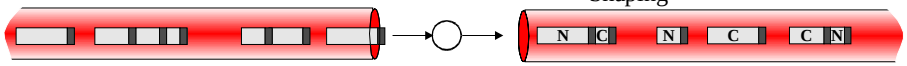
- Overall objective is to offer QoS and/or maximize network throughput
- This requires
  - Limiting user traffic to the level that individual links operate on optimal fashion
  - Individual links can not be fully utilized
    - Unequal capacities
    - Uncertainty of paths
    - Uncertainty of demands





# Rate Control

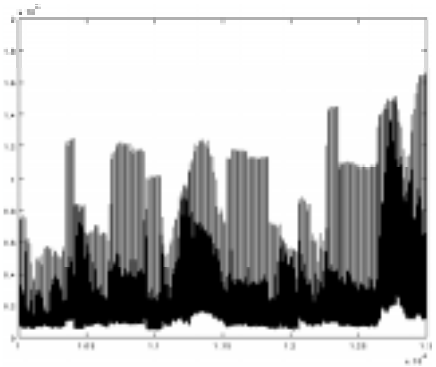
- Task is to decide which user packets should be delivered into the network and on what priority (mark)
  - They do not violate QoS management principles within the network by overloading the network
- Rate control operates in three levels
  - Measures the traffic
  - Compares the measured information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

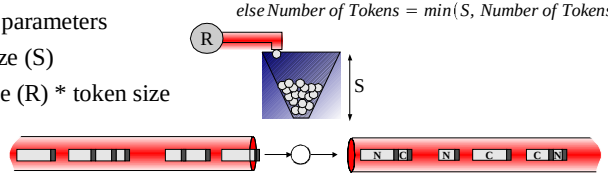
- Packet stream is measured to find out some of the following parameters:
  - Peak rate − maximum rate on which user is sending
  - Sustained rate − average rate on which user is sending
  - Burst size − maximum burst size which user sending on either with peak or average rate
- Actual measurement of 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:***
$Number\ of\ Tokens = S$

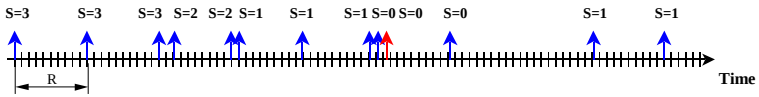
***Upon each arrival:***
$Increment = TokenSize \cdot R \cdot (T_{Now} - T_{Last\ Arrival})$
$Decrement = PacketLength$
$Conformance = Number\ of\ Tokens + Increment - Decrement$
$if\ Conformance \geq 0$
$then\ Number\ of\ Tokens = min(S, Conformance)$
$else\ Number\ of\ Tokens = min(S, Number\ of\ Tokens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






# 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

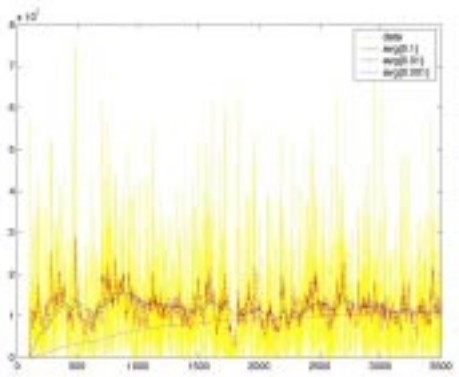





# Packet per packet EWMA meter

- Measures packet stream by using exponentially weighted moving average filter.
  - Tunable by parameter
    - Memory ($\epsilon$)

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

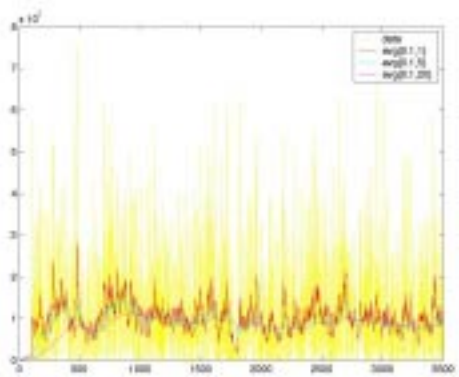




# Windowed EWMA meter

- Measures packet stream by using exponentially weighted moving average filter with sampling window.
  - Tunable by parameters
    - Memory ($\epsilon$)
    - Sampling interval ($\Delta T$)

***Initial condition:***
$avg(0) = 0$
***After every $\Delta T$ 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:***

$avg\{0\} = 0$

$Win_{length} = C$

$T_{front} = 0$

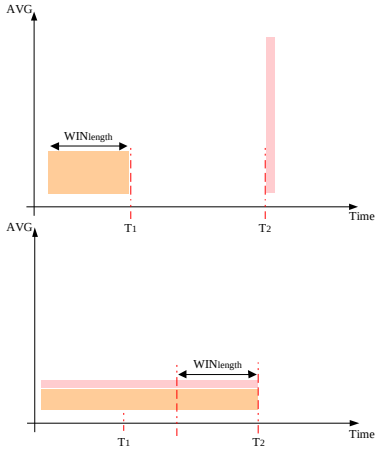
***After every packet arrival:***

$Bytes_{TSW} = avg\{n\} \cdot Win_{length}$

$New_{bytes} = Bytes_{TSW} + PacketLength$

$$avg\{n+1\} = \frac{New_{bytes}}{T_{now} - T_{front} + Win_{leght}}$$

$T_{front} = T_{now}$






# 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.



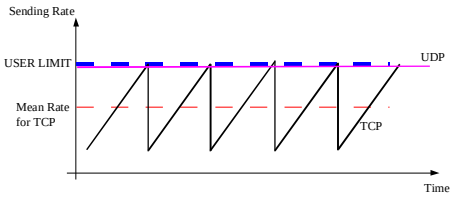
# Conformance algorithms

- 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



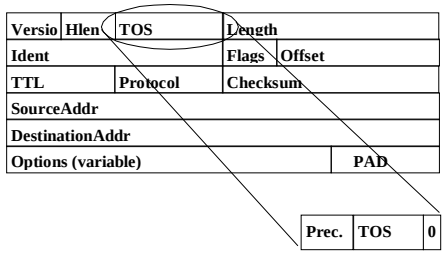
# Rate Control Problems

- 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

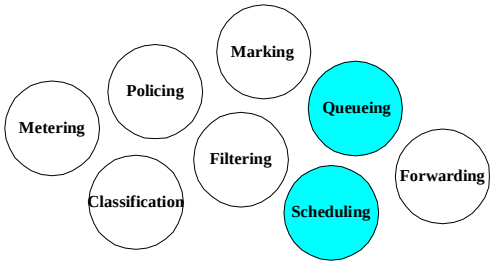
- Marker is used to attach conformance / class information to every packet.
- Marker uses IPv4 TOS/DSCP field to convey information for other processing elements in the network.
  - TOS
    - Prec: 3 bit priority
    - TOS: user preference for routing
  - DSCP
    - Class and precedence






# Output processor

- Output processor of a Internet router consists following elements
  - Queues and their management algorithms
  - Scheduling






# Queues

- Queues are used to store **contending** packets
  - Contention is temporary event rising from statistical multiplexing
    - Packets from different input links of a router attempt to the same output link at certain time
    - Packets from a higher speed link arrive temporarily too fast for a slow speed link
- If contention is permanent queues overflow i.e. network is **congested**
- Difference:
  - Contention − packets are not lost only delayed
  - Congestion − packets are not only delayed but also lost



# Queues

- Congestion situations demand **queue management** to decide
  - When packets should be discarded
  - Which are the packets that should be discarded
- Prevalent solutions
  - Tail Drop
  - Random Early Detection (RED)
  - Random Early Detection In/Out (RIO)

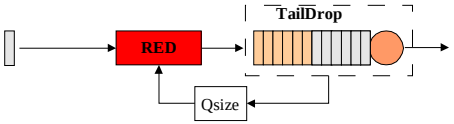
# Tail Drop

- Simple algorithm:
  - If arriving packets sees a full queue it is discarded
  - Otherwise it is accepted to the queue
- Problem:
  - Poor fairness in distribution of buffer space
  - Unable to accommodate short transients when queue is almost full
    - Bursty discarding leading global syncronisation
- Global syncronisation is a process where large number of TCP connections syncronise their window control due to concurrent packet losses.
  - Packet losses are bursty, therefore window decreases to one and halts the communication



# Random Early Detection

- RED is an active queue management algorithm (AQM), which aims to
  - Prevent global syncronisation
  - Offer better fairness among competing connections
  - Allow transient burst without packet loss
- Algorithm operates on the knowledge of current Qsize
  - Updated on every arrival and departure from the actual queue






# RED

- Qsize is used to calculate average length of the queue:

***Initial condition:***

$avg(0) = 0$

$Count = -1$

***When Qsize=0:***

$T_{idle} = T_{now}$

***After every packet arrival:***

***if*** $Qsize(n) > 0$***:***

$avg(n+1) = (1-\epsilon) \cdot avg(n) + \epsilon \cdot Qsize(n)$

***else:***

$avg(n+1) = avg(n) \cdot (1-\epsilon)^{f(T_{now} - T_{idle})}$

**If queue is empty, averaging is done based on the assumption that N packets have passed the algorithm before actual packet arrival.**
**-> Decay of average during idle times**

- Packets are discarded based on the average queue length:

***if*** $avg(n+1) < min_{th}$***:***

$Count = -1$

***else if*** $min_{th} \le avg(n+1) < max_{th}$***:***

**Stochastic packet discard**

$count = count + 1$

$P_b(n+1) = max_p \cdot \frac{avg(n+1) - min_{th}}{max_{th} - min_{th}}$

$P_a(n+1) = \frac{P_b(n+1)}{1 - count \cdot P_b(n+1)}$

***With probability*** $P_a(n+1)$***:***

*Discard packet*

$Count = 0$

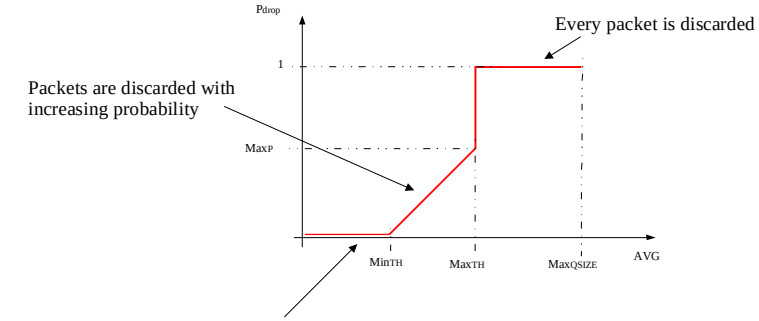
***else if*** $max_{th} \le avg(n+1)$

*Discard packet*

$Count = 0$



# RED



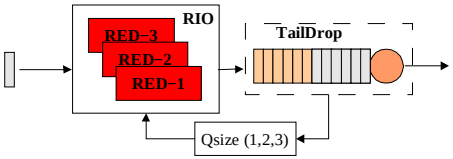
# Achievements of RED

- Some packets are discarded even before overflow of the actual buffer
  - Is it good or bad ?
    - Bad: A part of buffer space is in some occasions wasted
    - Good: A signal is sent to co−operating sources that they should decrease their sending rate or congestion will occure
- On the average early packet discards will hit connections which use more than their fair share of capacity in contending link
  - Is it good or bad ?
    - Bad: Makes differentiation impossible
    - Good: Is consistent policy and withing the goal of conventional Best Effort model



# RED In/Out − WRED

- When we aim for differentiation of resources we must also allow different shares of resources in contending link or buffer
- One way to do it is to use RED with several parallel algorithms and thresholds
  - RED In/Out −> RIO or WRED
  - Popular implementations use two or three parallel algorithms
- This requires that packets are marked
  - One algorithm is responsible of one or several marks






# RIO

- Operation is usually based on following idea:
  - Customer has contracted capacity of X bps
  - He sends packets with rate Y bps
  - If Y is greater than X, some packets are marked as out of profile.
    - Out of profile packets usually experience harsh treatment on contending situations
- Calculation of the average queue length is modified to take into accout number of packets with different markings:
  - In (green): Only green packets
  - In/Out (yellow): Green and yellow packets
  - Out (red): All packets in the queue



# Parameters in WRED

- All parameres are independent for different markings
  - More dimensions in creating differentiation
- Some parameters are common for different markings
  - Less dimensions but more understandable