



**S-38.180 Palvelunlaatu Internetissä**  
**S-38.180 Quality of Service in Internet**  
Harjoitus 2: Rate Control  
Exercise 2: Rate Control



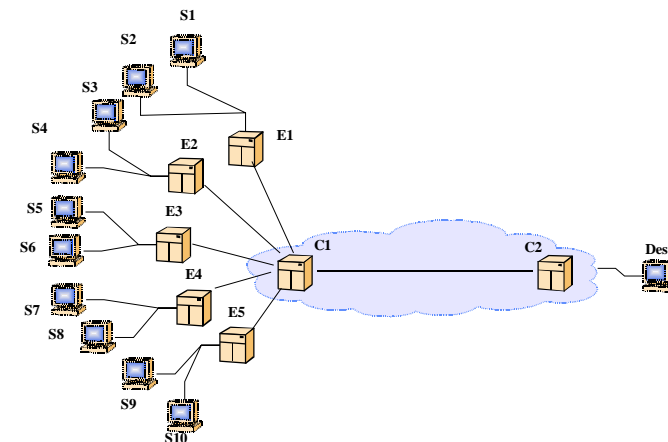
## Exercise

- Comparison of different rate control methods in Internet.
- Methods which are used are:
  - Token bucket
  - Time sliding window



## Reporting

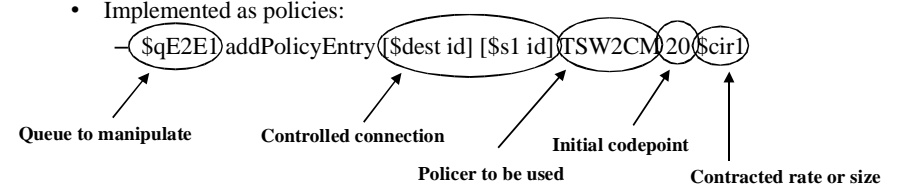
- Due date: 25.10.2002 at 0900 hours.
- Reports: delivered to course locker in G-wing 2nd floor.



## Task

- Investigate the operation of different metering and marking algorithms
- Loose and strict conformance rules
  - Loose: excess packets are accepted with lower priority
  - Strict: excess packets are discarded
- Contracts:
  - All clients equal contract
  - Some clients substantially higher contract
  - Some clients zero contract
- Different SLAs
  - Homo and heterogeneous SLAs
- SLA1:
  - 25kbps UDP
  - 150kbps TCP
- SLA2:
  - 100kbps UDP
  - 500kbps TCP
- Different round trip times
  - Vary round trip times between different TCP connections

## Rate control

- Implemented as policies:
    - `$qE2E1 addPolicyEntry ($dest id) [$s1 id] TSW2CM 20 ($cir1)`
- 

## Policers

- TSW2CM:
  - Time Sliding Window Meter with Two Color Marker
  - `Queue addPolicyEntry [Destination id] [Source id] TSW2CM Initial Codepoint CIR`
  - Packets are marked to lower precedence probabilistically when CIR is exceeded
- TSW3CM:
  - Time Sliding Window Meter with Three Color Marker
  - `Queue addPolicyEntry [Destination id] [Source id] TSW3CM Initial Codepoint CIR PIR`
  - Packets are marked to medium precedence probabilistically when CIR is exceeded
  - Packets are marked to low precedence probabilistically when PIR is exceeded

## Policers

- Token Bucket:
  - `Queue addPolicyEntry [Destination id] [Source id] tokenBucket Initial Codepoint CIR CBS`
  - Packets are marked to lower precedence when bucket is empty
  - $CIR = \text{Token rate} * \text{Token size}$
  - $CBS = \text{Size of the token bucket}$

## Policers

- srTCM:
  - Single Rate Meter with Three Color Marker
  - *Queue addPolicyEntry [Destination id] [Source id] srTCM Initial Codepoint CIR CBS EBS*
  - Two token buckets in cascade
  - Packets are marked to medium precedence when first bucket is empty but second is not
  - Packets are marked to low precedence when both buckets are empty

## Policers

- trTCM:
  - Two Rate Meter with Three Color Marker
  - *Queue addPolicyEntry [Destination id] [Source id] trTCM Initial Codepoint CIR CBS PIR PBS*
  - Two token buckets in cascade
  - Packets are marked to medium precedence when first bucket is empty but second is not
  - Packets are marked to low precedence when both buckets are empty

## Policers

- Action of policers is defined

- `$qE1E2 addPolicerEntry TSW2CM 10 11`

Queue to manipulate

Policer type

Initial codepoint

Degraded codepoint

## Forwarding

- Each packet with assigned codepoint need some forwarding action to be associated
- This is done by selecting queue and precedence based on the codepoint

- `$qE1E2 addPHBEntry 10 0 1`

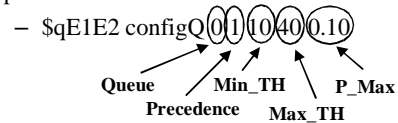
Codepoint

Queue

Precedence (virtual queue)

## Forwarding

- Each virtual queue (precedence) is associated a RED algorithm with own parameters:



## Sources

- UDP source is defined by
  - Protocol source
    - Packet size
  - Protocol destination
    - Null
  - Application
    - Packet size
    - Sending rate

```
set udp1 [new Agent/UDP]
$ns attach-agent $s1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packet_size_ $packetSize
$udp1 set packetSize_ $packetSize
$cbr1 set rate_ $rate1
set null1 [new Agent/Null]
$ns attach-agent $dest $null1
$ns connect $udp1 $null1
```

## Sources

- TCP source is defined by
  - Protocol source
    - Packet size
  - Protocol destination
    - Sink
  - Application

```
set tcp10 [new Agent/TCP]
$ns attach-agent $s10 $tcp10
set ftp10 [new Application/FTP]
$ftp10 attach-agent $tcp10
$tcp10 set packetSize_ $packetSize
set sink10 [new Agent/TCPSink]
$ns attach-agent $dest $sink10
$ns connect $tcp10 $sink10
```

## Simulator output

- Event file where all
  - Enqueue (+)
 

```
r 6.6938 4 5 tcp 1000 ----- 1 0.0 2.0 657 1301
+ 6.6938 5 2 tcp 1000 ----- 1 0.0 2.0 657 1301
```
  - Dequeue (–)
 

```
- 6.6938 5 2 tcp 1000 ----- 1 0.0 2.0 657 1301
```
  - Drop (d)
 

```
r 6.694155 1 4 tcp 1000 ----- 2 1.0 3.0 10 1356
+ 6.694155 4 5 tcp 1000 ----- 2 1.0 3.0 10 1356
```
  - Receive (r)
 

```
d 6.694155 4 5 tcp 1000 ----- 2 1.0 3.0 10 1356
r 6.6946 0 4 tcp 1000 ----- 1 0.0 2.0 684 1357
```

are presented in tabulated fashion