



Corso di QoS e sicurezza nelle reti

Lezione del 13/05/2015 – Configurazione
in Omnet++ Token Bucket e Algoritmo
RED

Ing. Cesare Sottile

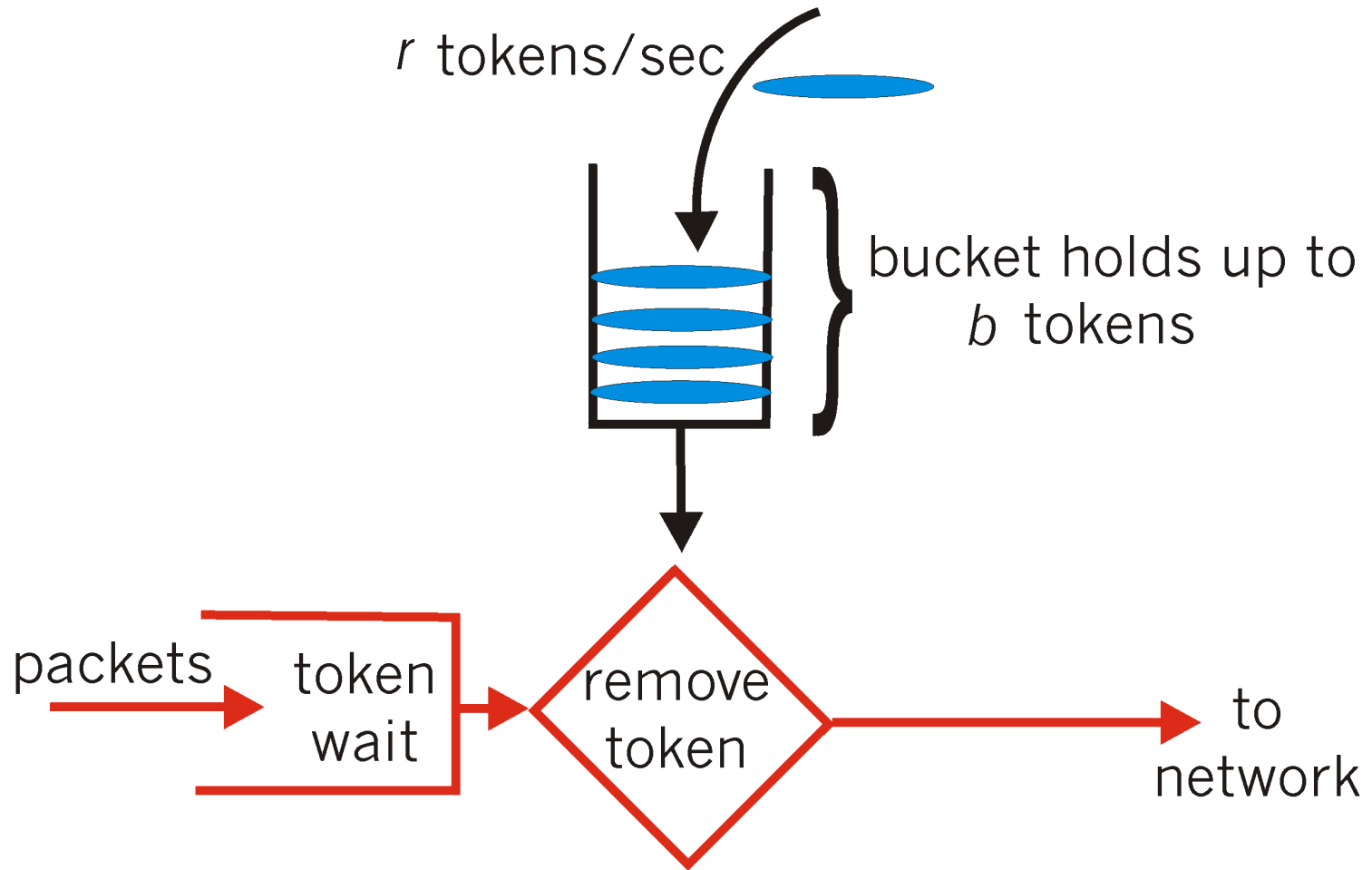
Ph.D Student

Cubo 41c – VI Piano

E-Mail : sottile@dimes.unical.it

Blog: <http://cesaresottile.wordpress.com/>

Token Bucket Regulator (Shaper)



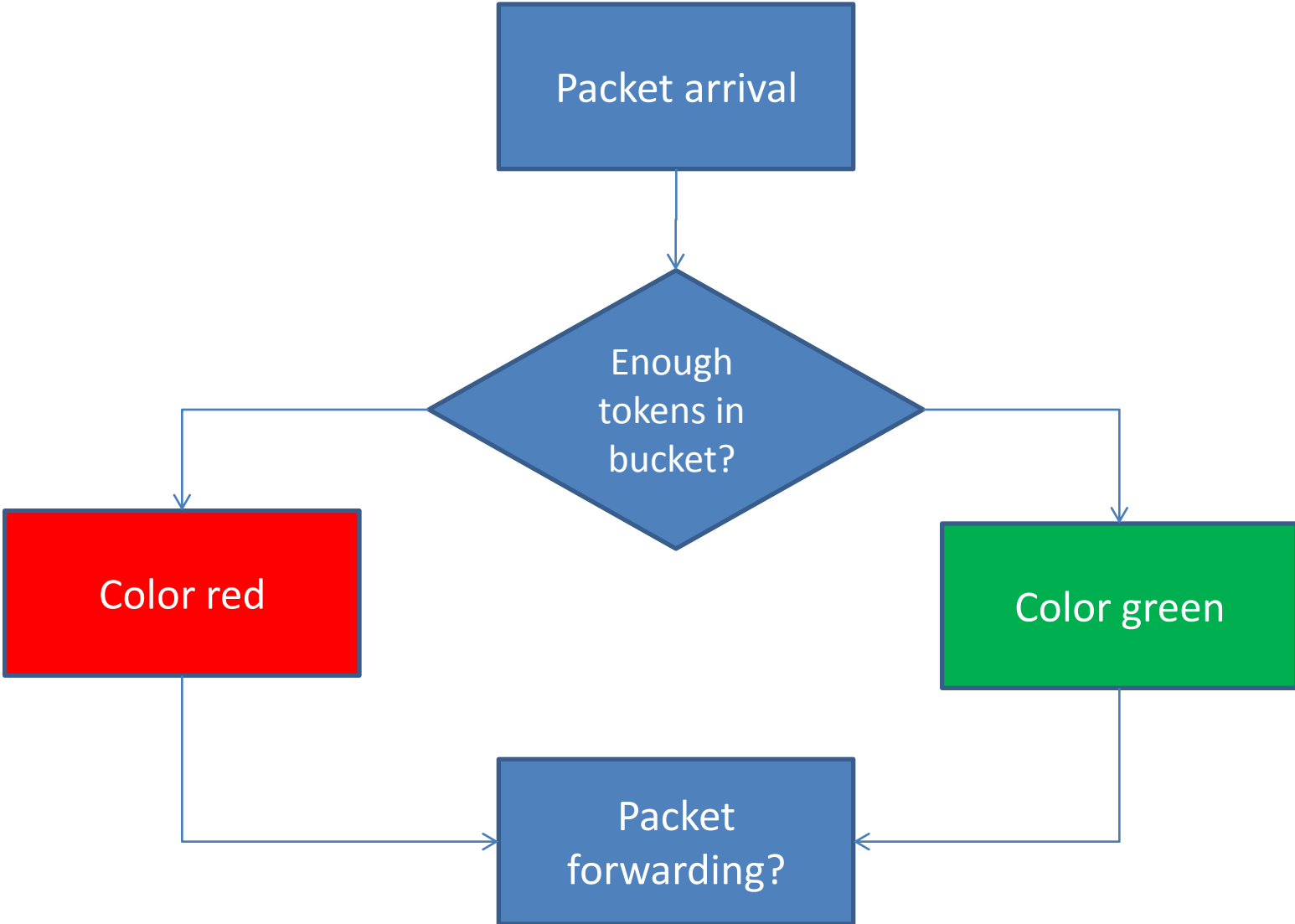
What is it good for?

- Traffic shaper
 - Shape the traffic to conform to contract
- Traffic policer
 - Drop packets that do not conform
- Traffic marker
 - Color packets by conformance to contract
 - Rate estimator

Remarks

- When tokens are not available
 - Store packet (shaper)
 - Discard packet (policer)

Traffic Marker



IETF Packet Marking

- Two types of markers are available:
 - RFC 2697: A Single-Rate, Three-Color Marker
 - Committed Information Rate (CIR),
 - Committed Burst Size (CBS),
 - Excess Burst Size (EBS)
 - RFC 2698: A Dual-Rate, Three-Color Marker
 - Peak Information Rate (PIR)
 - Committed Information Rate (CIR),
 - Committed Burst Size (CBS),
 - Peak Burst Size (PBS)
- Suggested in the context of DiffServ

SLA

Class (aggr.)	CIR	PIR	EIR	CBS	EBS	fair thruput
Scenario A						
1	300K	2M	1.7M	18K	12K	817K
2	300K	2M	1.7M	18K	18K	817K
3	300K	1M	0.7M	18K	12K	514K
4	300K	1M	0.7M	18K	18K	514K
5	150K	1M	0.85M	18K	12K	410K
6	150K	1M	0.85M	18K	18K	410K
7	150K	0.5M	0.35M	18K	12K	257K
8	150K	0.5M	0.35M	18K	18K	257K
Total	1.8M	9M	7.2M			4M
$bn = 4M, ex(bn) = 2.2, AGG_{CIR} = 0.45, AGG_{EIR} = 3.27$						
Scenario B						
1	200K	4M	3.8M	18K	12K	833K
2	200K	4M	3.8M	18K	18K	833K
3	200K	2M	1.8M	18K	12K	500K
4	200K	2M	1.8M	18K	18K	500K
5	100K	2M	1.9M	18K	12K	416K
6	100K	2M	1.9M	18K	18K	416K
7	100K	1M	0.9M	18K	12K	250K
8	100K	1M	0.9M	18K	18K	250K
Total	1.2M	18M	16.8M			4M
$bn = 4M, ex(bn) = 2.8, AGG_{CIR} = 0.3, AGG_{EIR} = 6$						

TABLE I

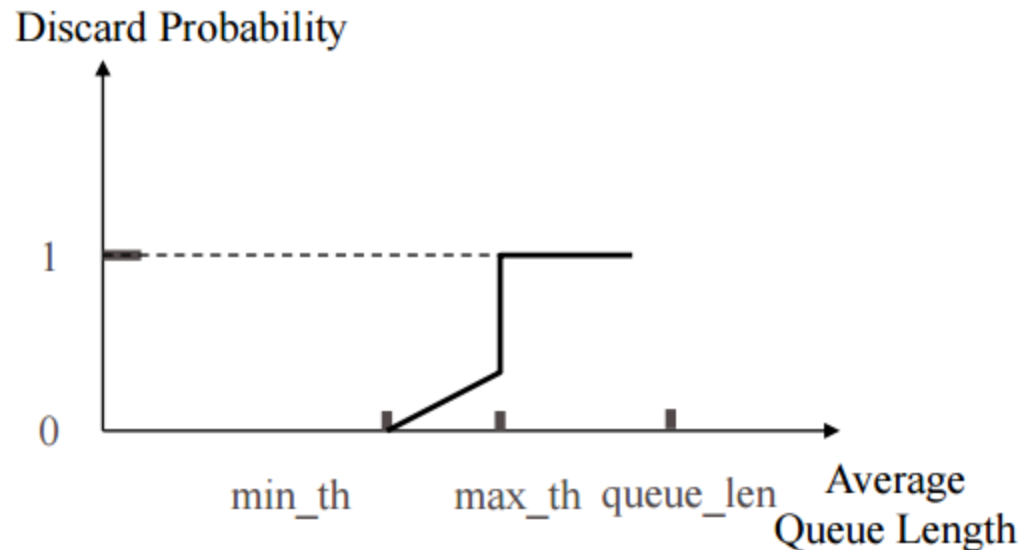
THE SLA PARAMETERS OF THE EIGHT AGGREGATES THAT COMPOSE SCENARIO A AND B. FOR EACH SET WE PRESENT THE EXCESS BOTTLENECK LINK AND THE *CIR* AND *EIR* AGGREGATION LEVELS.

Marker Locations and Size

Type of Marking	# of Bits	Bits Location
IP Precedence	3	Three most significant bits of TOS byte in IPv4 and IPv6 headers
Differentiated Services Code Point (DSCP)	6	Six most significant bits of TOS byte in IPv4 and IPv6 headers
MPLS Experimental (EXP) Bits	3	Part of 20 bit MPLS label
Ethernet CoS Bits	3	ISL or 802.1q/p header
ATM CLP Bit	1	ATM Cell header
Frame Relay DE Bit	1	Frame Relay header

RED Algorithm

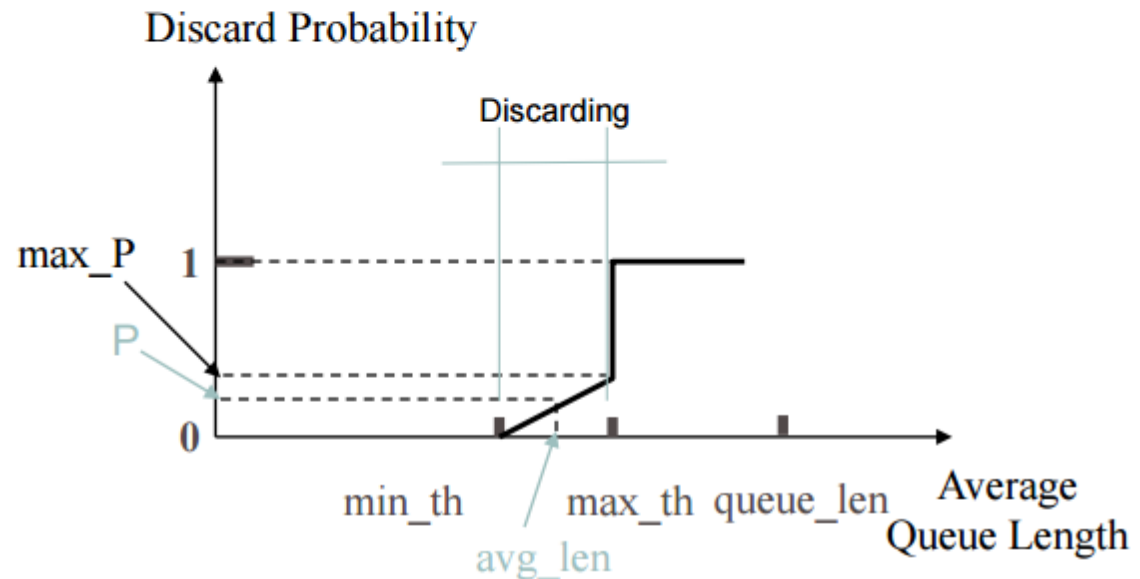
- FIFO scheduling
- Buffer management:
 - Probabilistically discard packets
 - Probability is computed as a function of **average** queue length (why average?)



RED Algorithm

- Setting the discard probability P :

$$P = \max_P \frac{\text{avg_len} - \text{min_th}}{\text{max_th} - \text{min_th}}$$



RED Algorithm

$$\text{avg} = (1 - w_q) \times \text{avg} + w_q \times q$$

Where:

q is the current length of the queue

w_q is the forgetfulness factor ($0.001 \leq w_q \leq 0.0042$)

we setted $w_q = 0.002$

they recommend that \max_p never be greater than 0.1

For more details see the paper:

Random Early Detection Gateways for Congestion Avoidance

Sally Floyd and Van Jacobson

<http://www.utdallas.edu/~jjue/cs6390/papers/red.pdf>

Esempio Token Bucket / RED

Progettare il token bucket (H1, H2, H3, H5) tale per cui si possa abbassare l'utilizzazione dei link R4-R2 minimizzando al massimo il dropping dei pacchetti. Questo permetterà di effettuare un traffic shaping che porti ad un miglioramento delle performance sulla rete, in particolare considerando l'utilizzazione dei link e il delay accumulato sulla rete tra sorgente e destinazione.

Esempio Token Bucket / RED








Verificare l'andamento della rete con e senza algoritmo RED attivato sul router R4 . I vincoli da considerare, per il disegno dell'algoritmo RED, sono i seguenti:

Ricavare il tasso di Drop attuato dal RED;

Il buffer deve essere utilizzato al massimo fino all'80% considerando una capacità pari a 150 frame (Vedi file "*omnet.ini*");

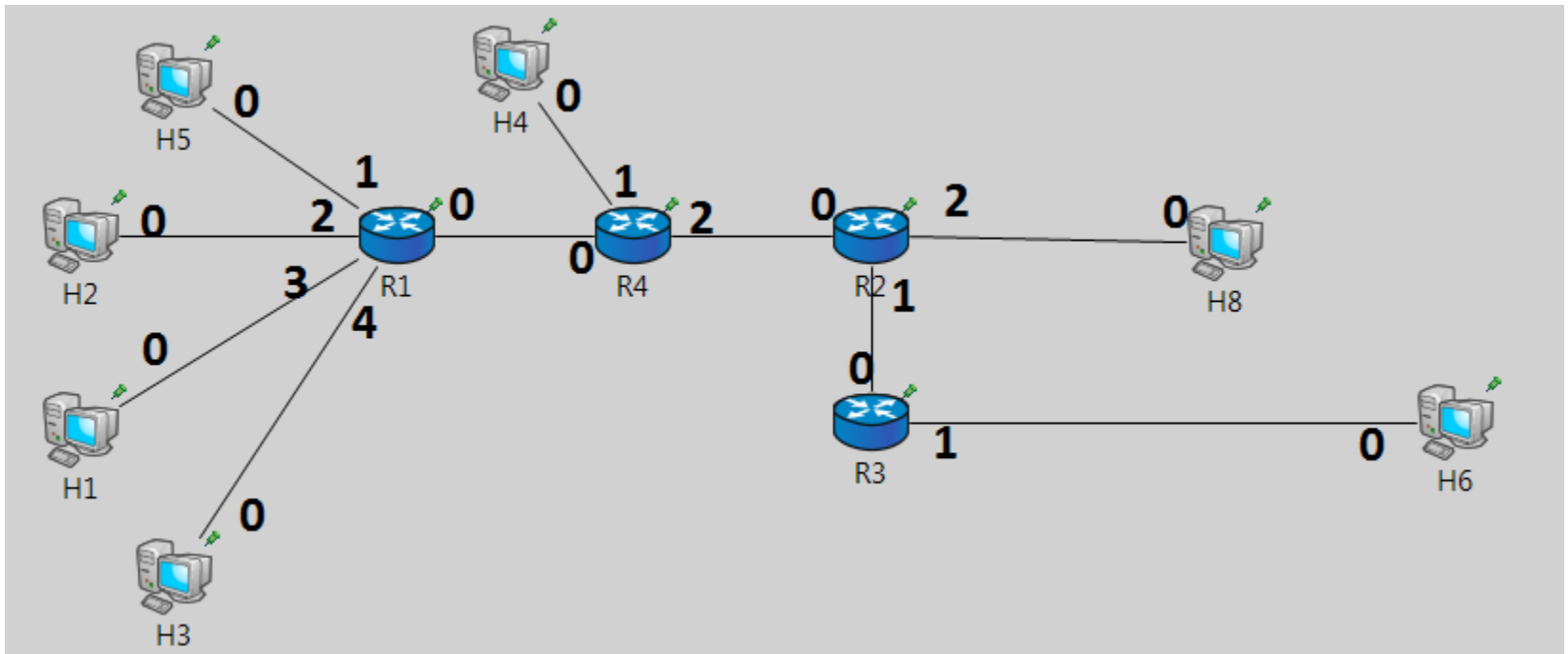
Minimizzare il ritardo E-2-E;

Esempio Token Bucket / RED

 H1	Sorgente	UDPBasicApp messageLength = 800B
 H2	Sorgente	UDPBasicApp messageLength = 720B
 H3	Sorgente	UDPBasicApp messageLength = 1024B
 H4	Sorgente	UDPBasicApp messageLength = 720B
 H5	Sorgente	UDPBasicApp messageLength = 720B
 H6	Destinazione	H3,H5
 H8	Destinazione	H1, H2, H4

Esempio Token Bucket / RED

Definizione delle interfacce / gate sui vari dispositivi della rete



Esempio Token Bucket / RED

Definizione delle applicazioni per generare il traffico da immettere sulla rete

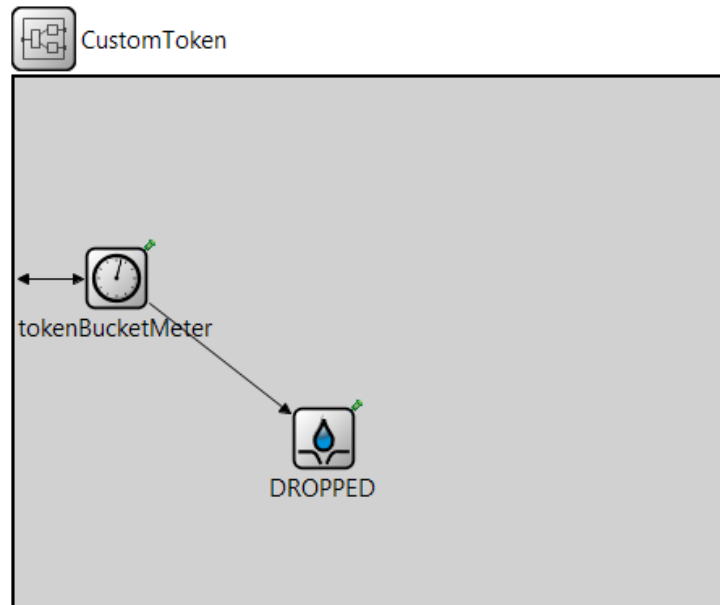
```
** .H2.numUdpApps = 1
** .H2.udpApp[0].typename = "UDPBasicApp"
** .H2.udpApp[0].destPort = 1002
** .H2.udpApp[0].startTime = uniform(1s,2s)
** .H2.udpApp[0].stopTime = 1200s
** .H2.udpApp[0].sendInterval = 5ms
** .H2.udpApp[0].messageLength = 720B
** .H2.udpApp[0].destAddresses = "H8"
```

```
** .H8.numUdpApps = 3
** .H8.udpApp[0].typename = "UDPSink"
** .H8.udpApp[0].localPort = 1001
** .H8.udpApp[1].typename = "UDPSink"
** .H8.udpApp[1].localPort = 1002
* .H8.udpApp[2].typename = "UDPSink"
* .H8.udpApp[2].localPort = 1004
```

Esempio Token Bucket / RED

Settaggio dei parametri relativi al Token Bucket
Committed Information Rate (cir), *Committed Burst Size (cbs)*,

```
*.R1.ppp[0].egressTCType      = "CustomToken"  
*.R1.ppp[0].**.tokenBucketMeter.cbs = 3264B  
*.R1.ppp[0].**.tokenBucketMeter.cir = "70%"  
*.R1.ppp[0].**.tokenBucketMeter.colorAwareMode = true
```



Esempio Token Bucket / RED

Settaggio dei parametri relativo al RED

```
** .R4.ppp[*].queueType = "RedQueue"  
* .R4.ppp[*].*.redDropper.minths = "10.0"  
* .R4.ppp[*].*.redDropper.maxths = "100.0"  
* .R4.ppp[*].*.redDropper.sampleTime = 1  
* .R4.ppp[*].*.redDropper.wq = 0.002  
* .R4.ppp[*].*.redDropper.maxps = "0.08"
```



RedQueue

