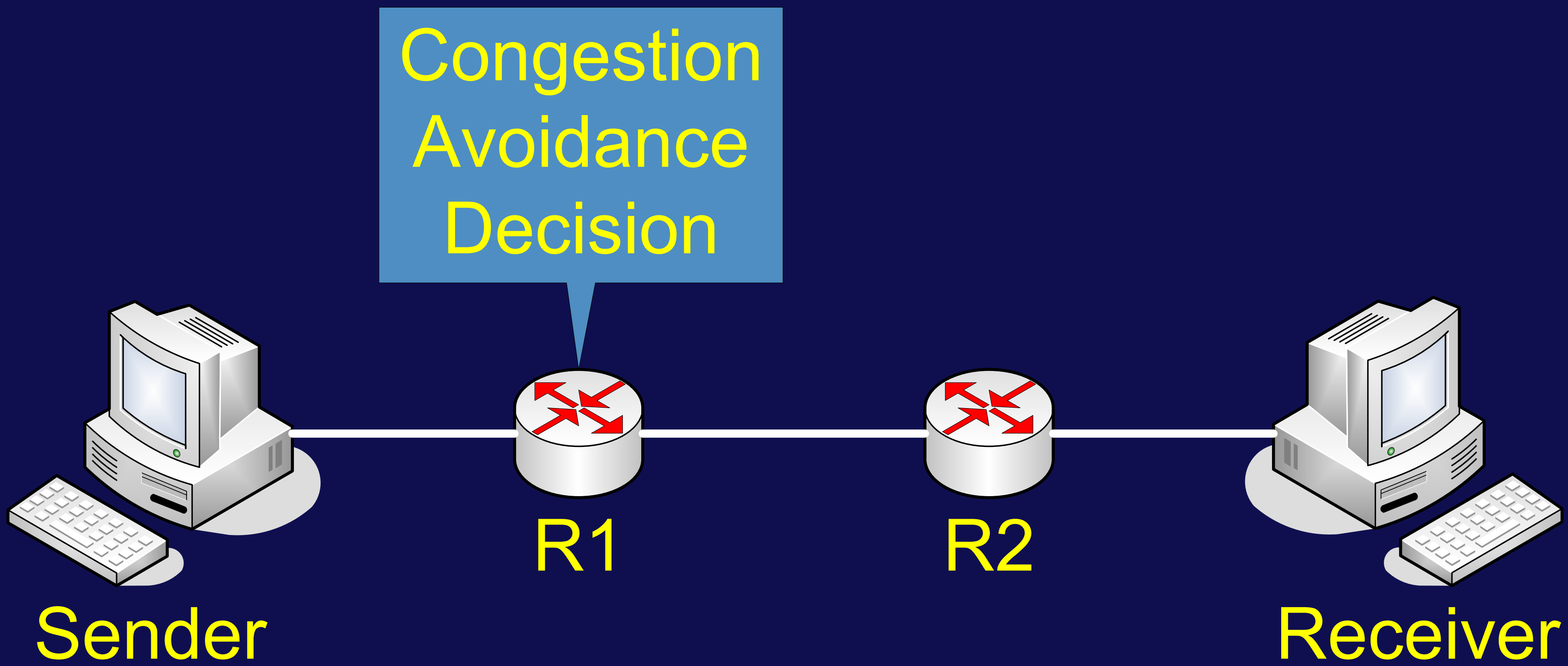
Congestion Avoidance with Future-Path Information

Peter Jacko
Brunilde Sansò

December 7, 2007

The Picture



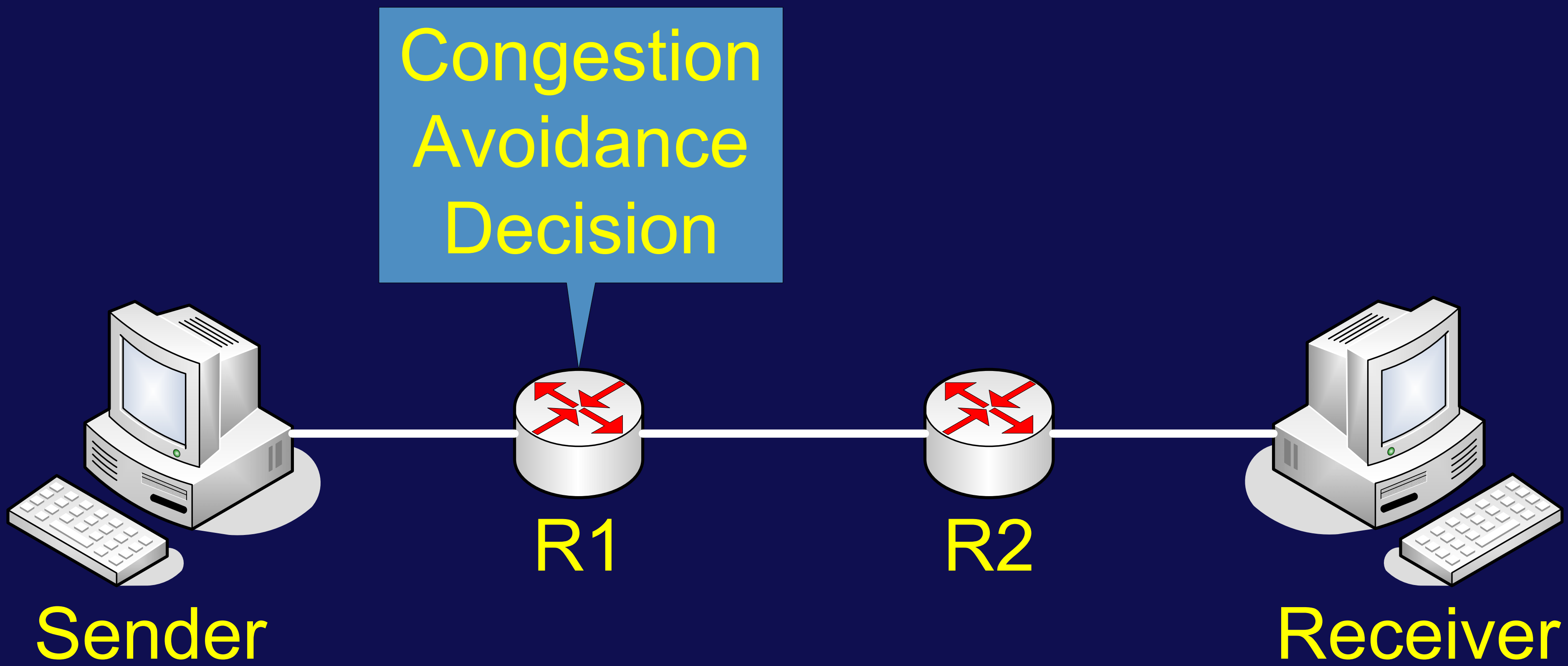
Motivation

- Network scarce resources: bandwidth and buffer space
- Dropping a packet on its route implies:
  - ▷ all scarce resources it has consumed so far are wasted
- Anticipating the loss of a packet
  - ▷ helps in economical allocation of scarce resources
  - ▷ avoids unnecessary packet losses
  - ▷ increases network throughput
- How to anticipate? ECN bits, plain drop estimation. . .

Congestion Avoidance Mechanisms

- 1990's: Reactive mechanisms: queue tail drop
- 2000's: Preventive mechanisms: RED, BLUE, etc.
  - ▷ based on router-based measures
    - – queue length, packet loss, link utilization
  - ▷ fairness from the sender's point of view
    - – each packet equally important
- 2010+ (?): Anticipative mechanisms:
  - ▷ fairness from the receiver's point of view
    - – packet's importance depends on the future path

Flow Rate Dependence



Restless Bandit Model

- Restless Bandit: binary-action MDP
- Appealing solution: index policy
  - ▷ e.g.: $c\mu$-rule, Gittins' index, Whittle's index
  - ▷ in general: marginal productivity index (MPI)
  - ▷ index captures an economic value of acting
- For resource allocation problems: priority index heuristic
  - ▷ nearly-optimal
  - ▷ easy to implement
  - ▷ easy to interpret

Bandit Machine

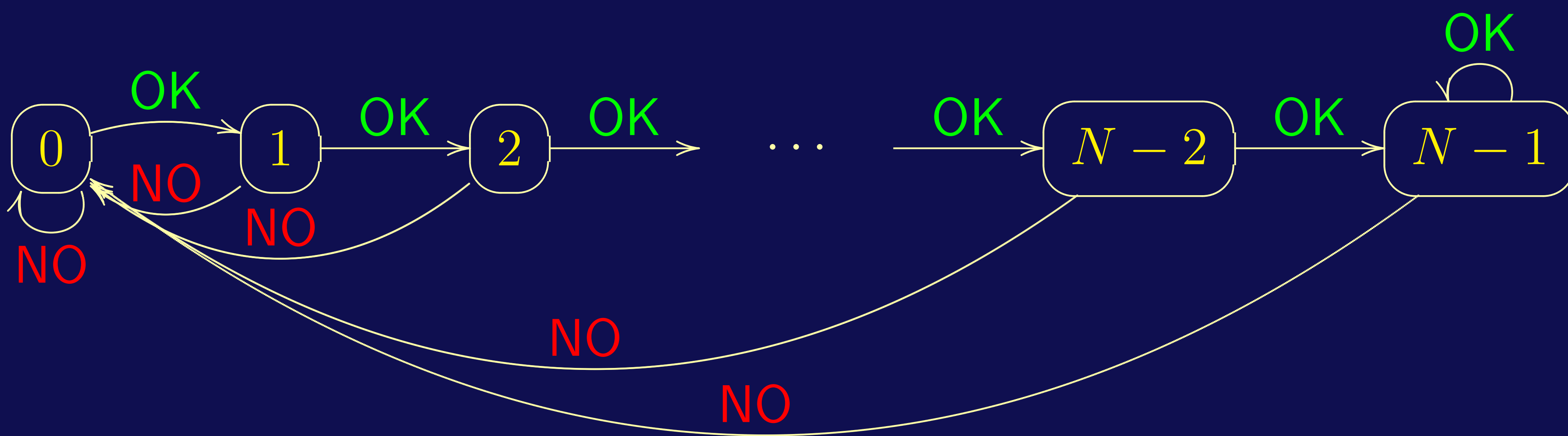
TCP Tahoe

- restarting-on-loss AI/MD transmission control protocol
- actualWindow — actual packet transmission rate
- Slow Start phase:
  - ▷ actualWindow starts at 1 packet per RTT
  - ▷ doubled every RTT with no packets lost
  - ▷ until reaching congestionThreshold
- Congestion Avoidance phase:
  - ▷ added 1 packet every RTT with no packets lost
  - ▷ until reaching advertisedWindow

TCP Tahoe as Restless Bandit

- States: $n \in \{0, 1, \ldots, N-1\}$
  - ▷ $n = 0$: transmission rate of 1 packet/RTT
  - ▷ $n = N-1$: maximum (advertisedWindow) rate
- Actions: OK (accept the flow), NO (reject the flow)

TCP Tahoe as Restless Bandit



- States: $n \in \{0, 1, \ldots, N-1\}$
  - ▷ w_n: buffer utilization (i.e., transmission rate)
  - ▷ r_n: receiver reward (e.g., expected goodput)
- Discrete time: period $=$ RTT

Optimization Problem

- Criteria: Total Discounted, Long-Run Average
- $\mathbb{R}_n^\pi$: expected reward under policy π starting from n
- $\mathbb{W}_n^\pi$: expected work under policy π starting from n
- ν: wage per unit of work (buffer utilization)
- Optimization problem

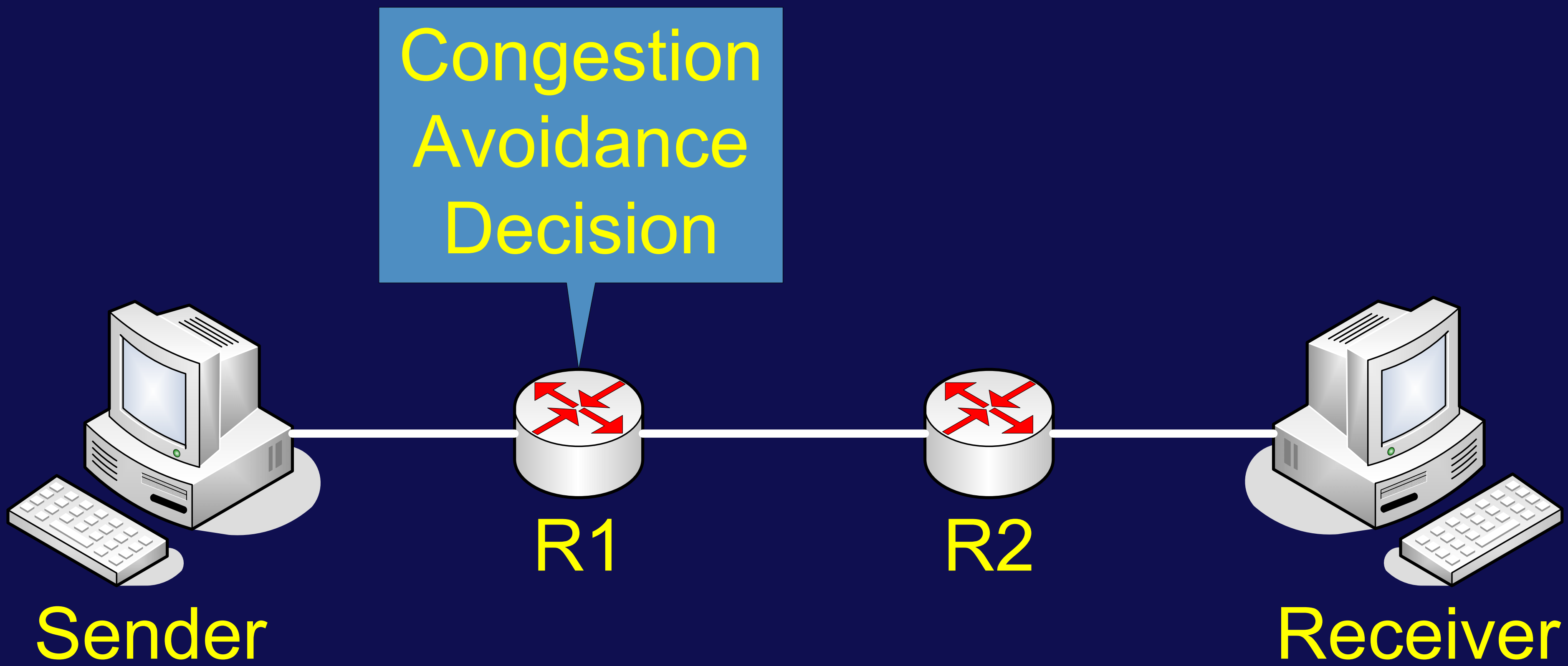
$$\max_{\pi} \ \mathbb{R}_n^\pi - \nu \mathbb{W}_n^\pi \tag{1}$$

Solution

- Concave Rewards Assum.: $r_n = r(w_n)$ for $r(\cdot)$ concave
- The optimal policy is a threshold policy
- The marginal productivity indices (MPI) exist
- The MPI for the long-run average criterion is

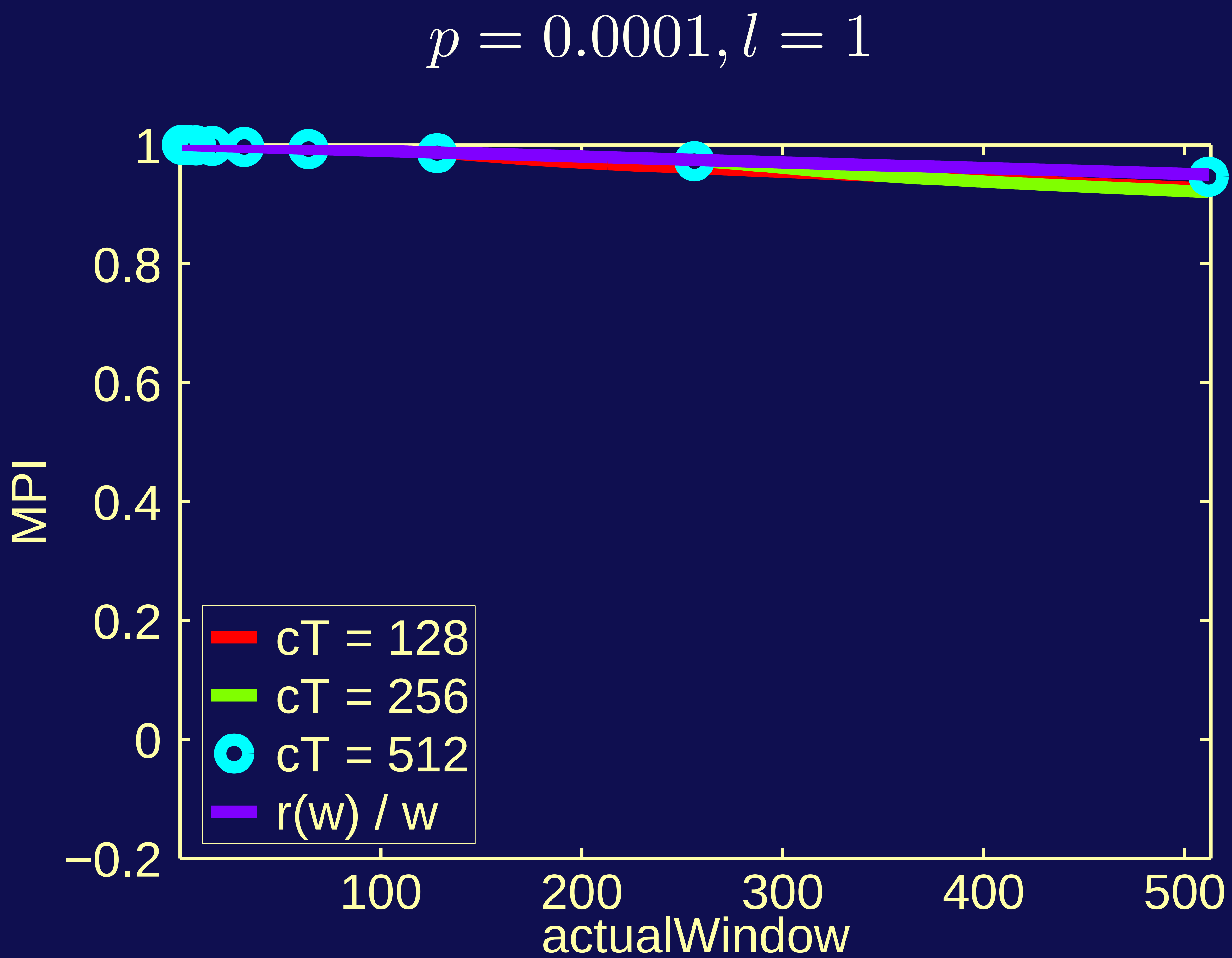
$$\nu_n = \frac{(n+1)r_n - \sum_{m=0}^{n-1} r_m}{(n+1)w_n - \sum_{m=0}^{n-1} w_m} \leq \frac{r_n}{w_n} \qquad (2)$$

The Picture

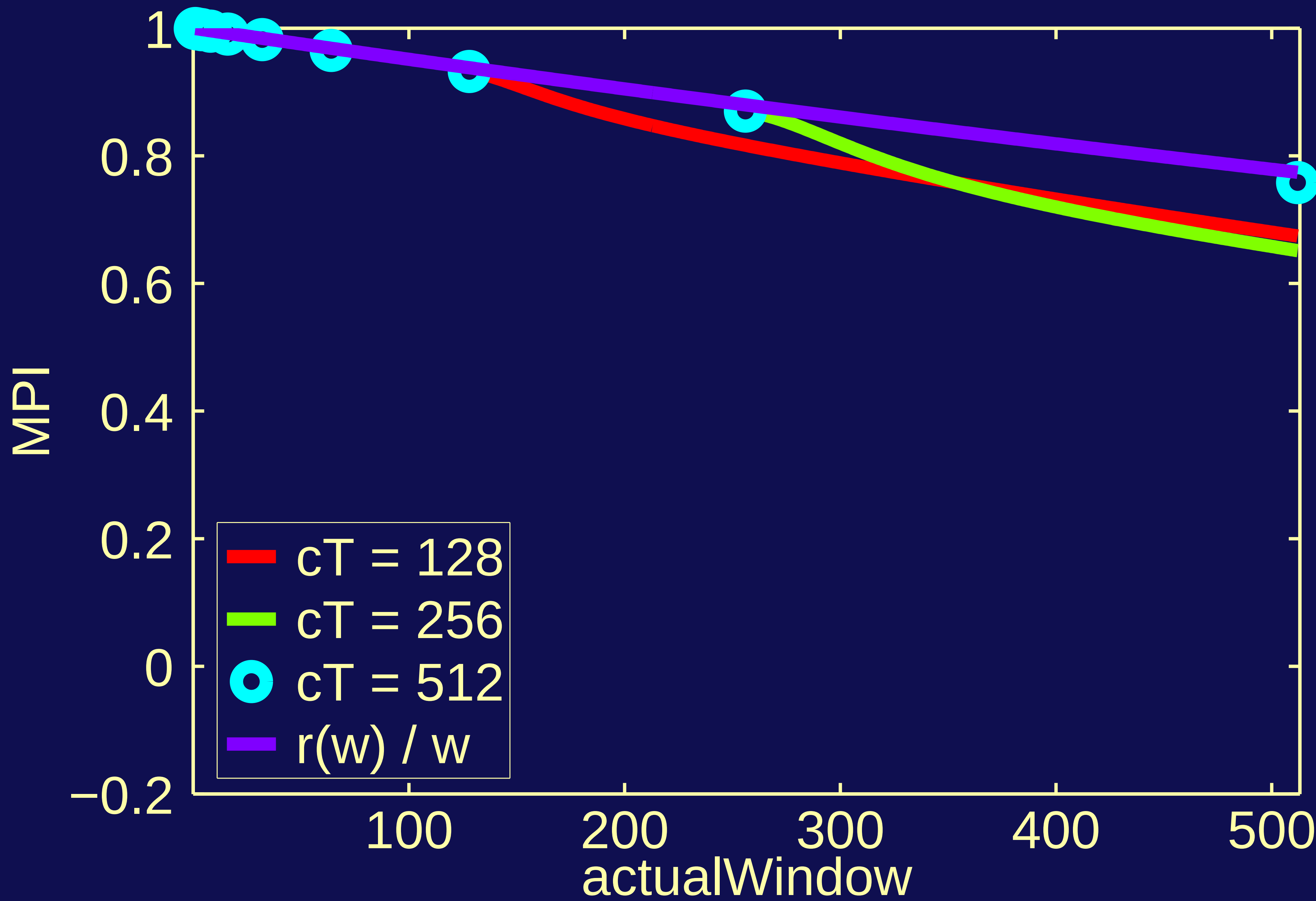


Example

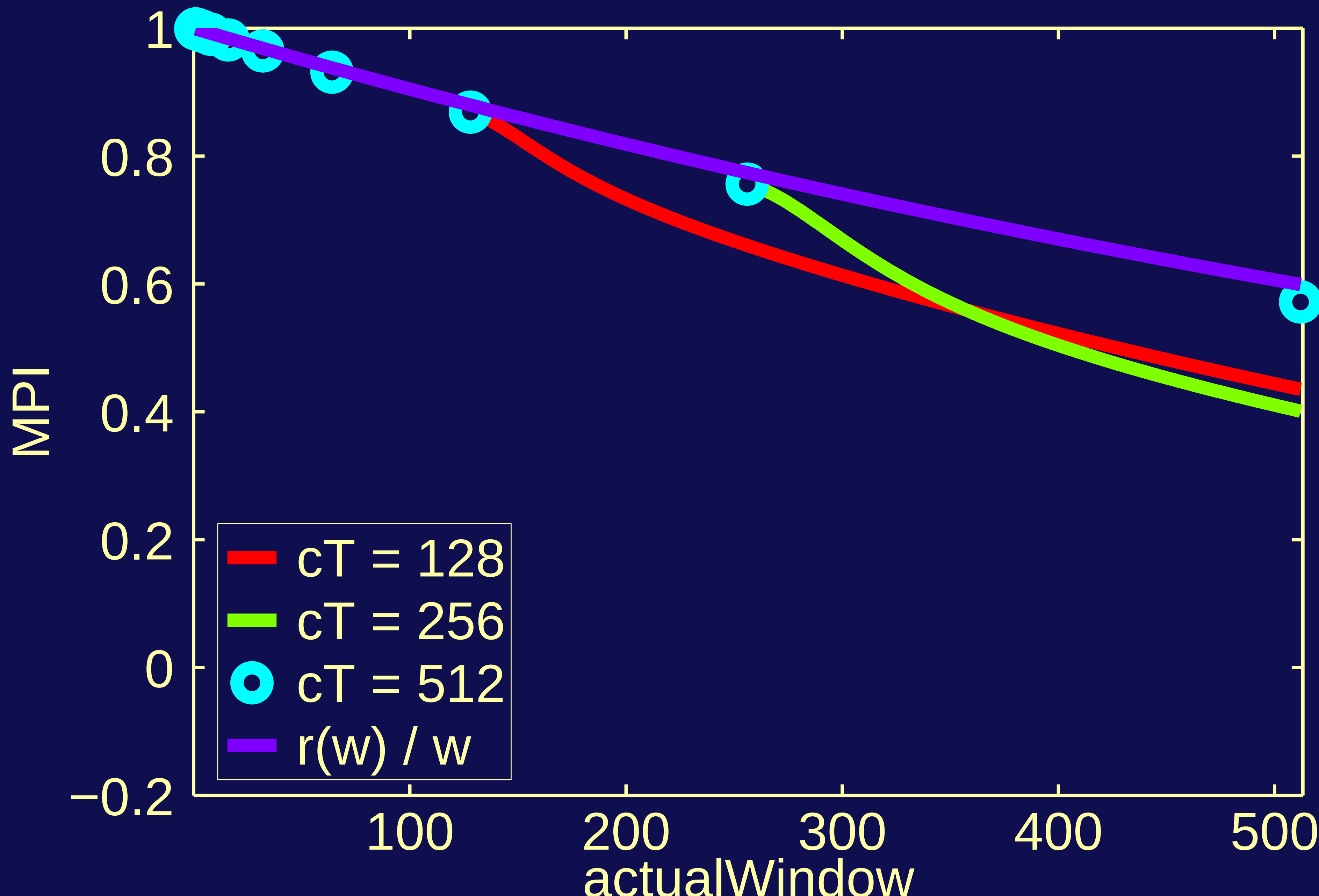
- Connection with l links, dropping probability p
- TCP Tahoe:
  - ▷ advertisedWindow $= 2^A = 512$ packets/RTT
  - ▷ congestionThreshold $= 2^C = 128/256/512$
- Expected Goodput: $r(w) = (1-p)^{lw} ws$
  - ▷ reward w, if all packets arrive
  - ▷ reward 0, if any packet is lost
  - ▷ s is useful size in Bytes of each packet (let $s = 1$)


$p = 0.0001, l = 1$
1
0.8
0.6
0.4
0.2
0
−0.2
MPI
100
200
300
400
500
actualWindow
cT = 128
cT = 256
cT = 512
r(w) / w

$p = 0.0001, l = 5$

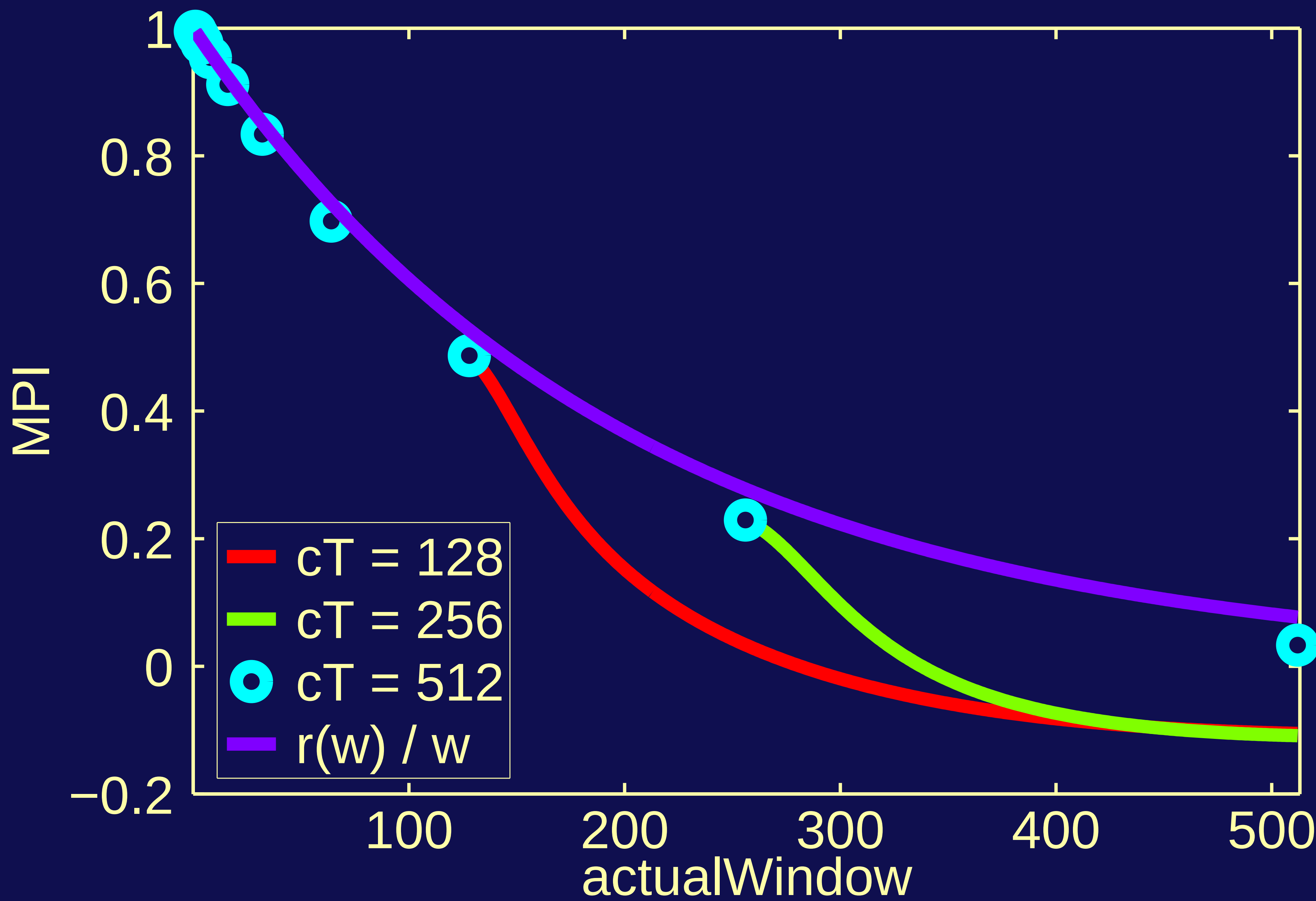

1
0.8
0.6
0.4
0.2
0
−0.2
MPI
100
200
300
400
500
actualWindow
cT = 128
cT = 256
cT = 512
r(w) / w

$$p = 0.001, l = 1$$



1
0.8
0.6
0.4
0.2
0
−0.2
MPI
100
200
300
400
500
actualWindow
cT = 128
cT = 256
cT = 512
r(w) / w

$$p = 0.001, l = 5$$



MPI
actualWindow
1
0.8
0.6
0.4
0.2
0
−0.2
100
200
300
400
500
cT = 128
cT = 256
cT = 512
r(w) / w

Implementation of MPI as Priority Indices

- Into any congestion avoidance mechanism that randomly drops packets on arrival
- Packet i with MPI ν_i and useful size s_i Bytes:
  - ▷ let it be dropped with probability p_i
- What should be dropping probability of packet j?
- Equalling the expected economic loss of dropping:

$$\nu_j s_j p_j = \nu_i s_i p_i \tag{3}$$

Implementation of MPI as Priority Indices

- Equalling the expected economic loss of dropping:

$$p_j = \frac{\nu_i s_i}{\nu_j s_j} p_i \tag{4}$$

- Only works for very small p's and similar-sized packets
- Alternatively:

$$p_j = 1 - (1 - p_i)^{\frac{\nu_i s_i}{\nu_j s_j}} \tag{5}$$

- Roughly equivalent for small p's, well defined for larger

Conclusions

- Optimal index policy for TCP Tahoe
- Implementation of indices into congestion avoidance
- Limitations
  - ▷ TCP Tahoe: no fast recovery, no fast retransmit
  - ▷ Restless bandit model: only two possible actions
  - ▷ Practice: not applicable into the Internet of today
- Though, a good starting point
- Extensions under development

Thank you for your attention!