Policing Congestion Response in an Internetwork using Re-feedback

Bob Briscoe[1,2]

Arnaud Jacquet[1], Carla Di Cairano-Gilfedder[1],
Alessandro Salvatori[1,3],

Andrea Soppera[1] & Martin Koyabe[1]

[1]BT Research, [2]UCL, [3]Eurécom

Based on authors' presentation on SIGCOMM'05
Presented for CS598PBG
by Haohui Mai
09/15/2009

the problem: policing with congestion response

- loss of throughput
- risk of repeated congestion collapses

from NCSA Visualization report on NSFNET

Catch me if you can!

- host response to congestion: *voluntary*
- Slow-start / ECN / XCP
  - I don't even use TCP
  - UDP flooding (e.g., VoIP, Skype)
- Fair Queue / RED / Throughputs throttling
  - Each flow gets the same amount of bandwidth
  - Create more flows (e.g., BitTorrent)
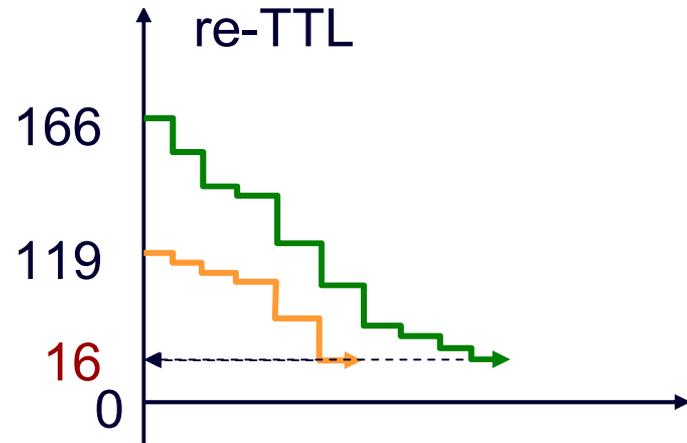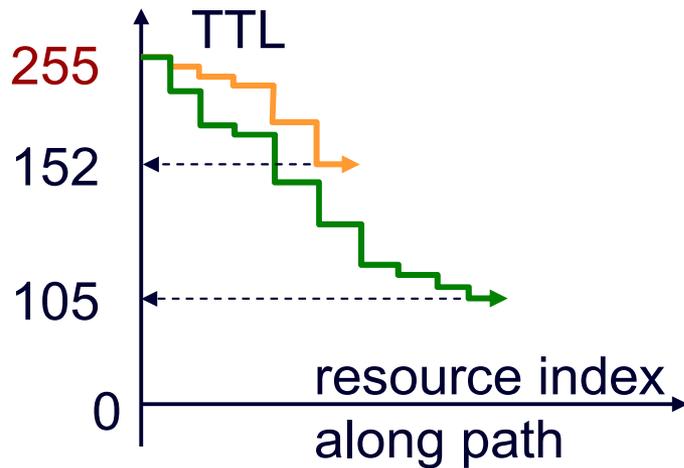- Whiten myself: change the MACs

A solution: Re-feedback

- Goal: Let every router on the path to know the downstream information
  - say, TTL, congestion
- Information propagates along the path
  - Kill suspicious packets

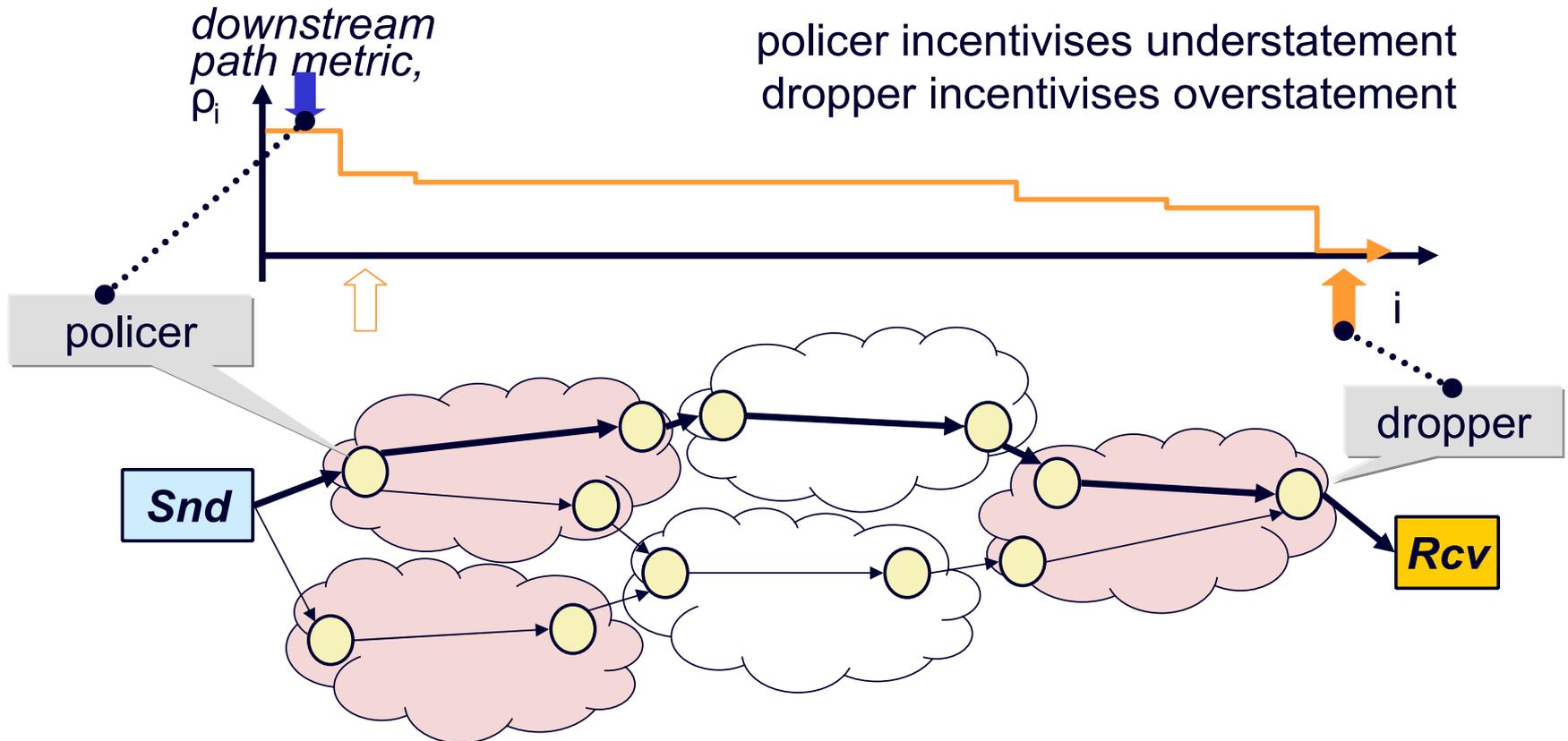- For each packet, source estimates a metric for downstream networks

An example: TTL
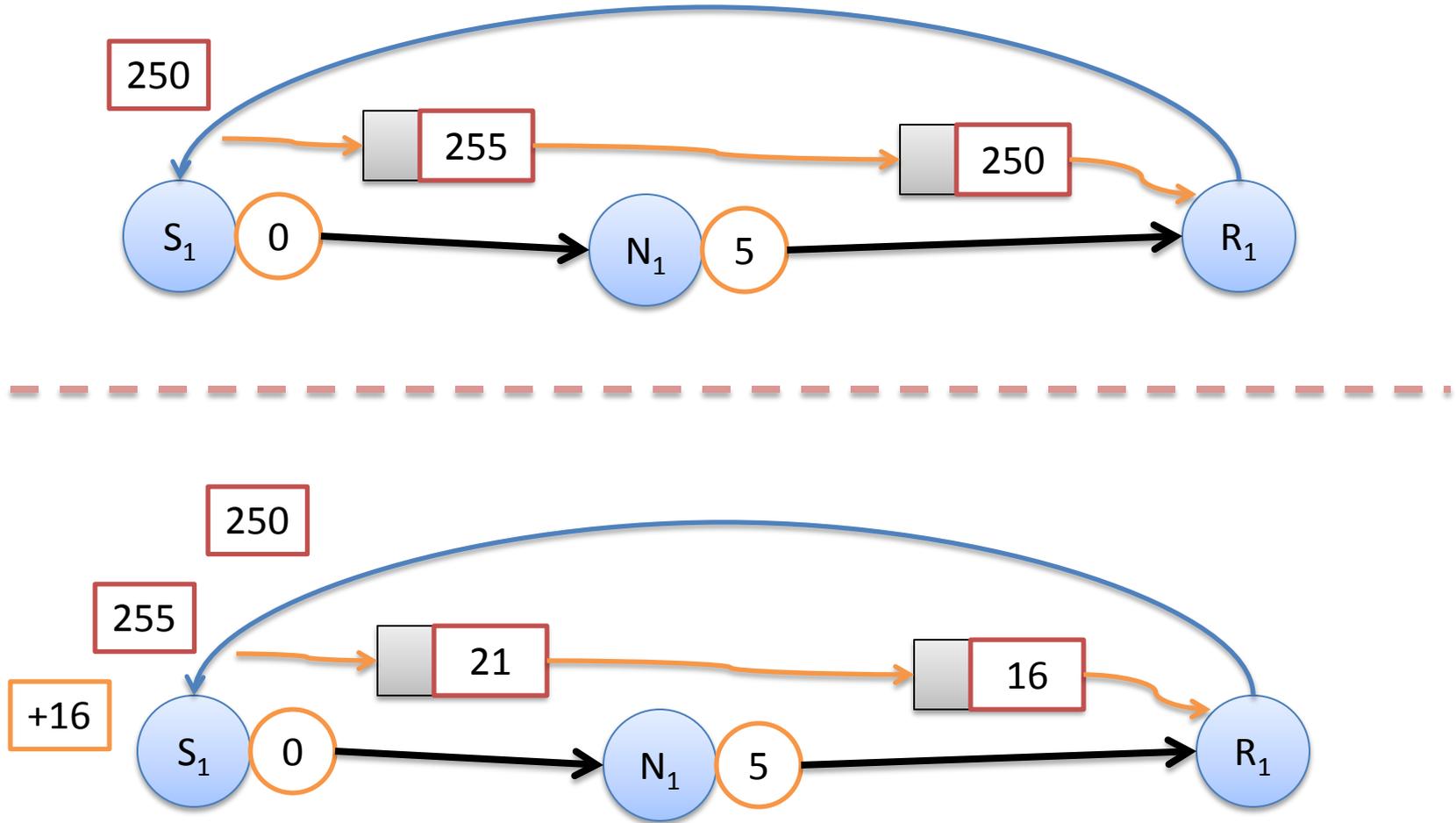
downstream path characterization

Let every router on the path
to know the downstream
information

Incentive frameworks

- A field estimating congestion rate
- Introduce Policer / Dropper

downstream path metric, ρ_i

policer incentivises understatement
dropper incentivises overstatement

policer

dropper

Snd

Rcv

i

# Lying on estimating congestion rates

- Understatement to pass through policer quickly
  - Packet might be dropped by dropper
- Overstatement to pass the dropper
  - Policer slows down the transmission
- The reasonable choice
  - To tell the truth

Flow Policer
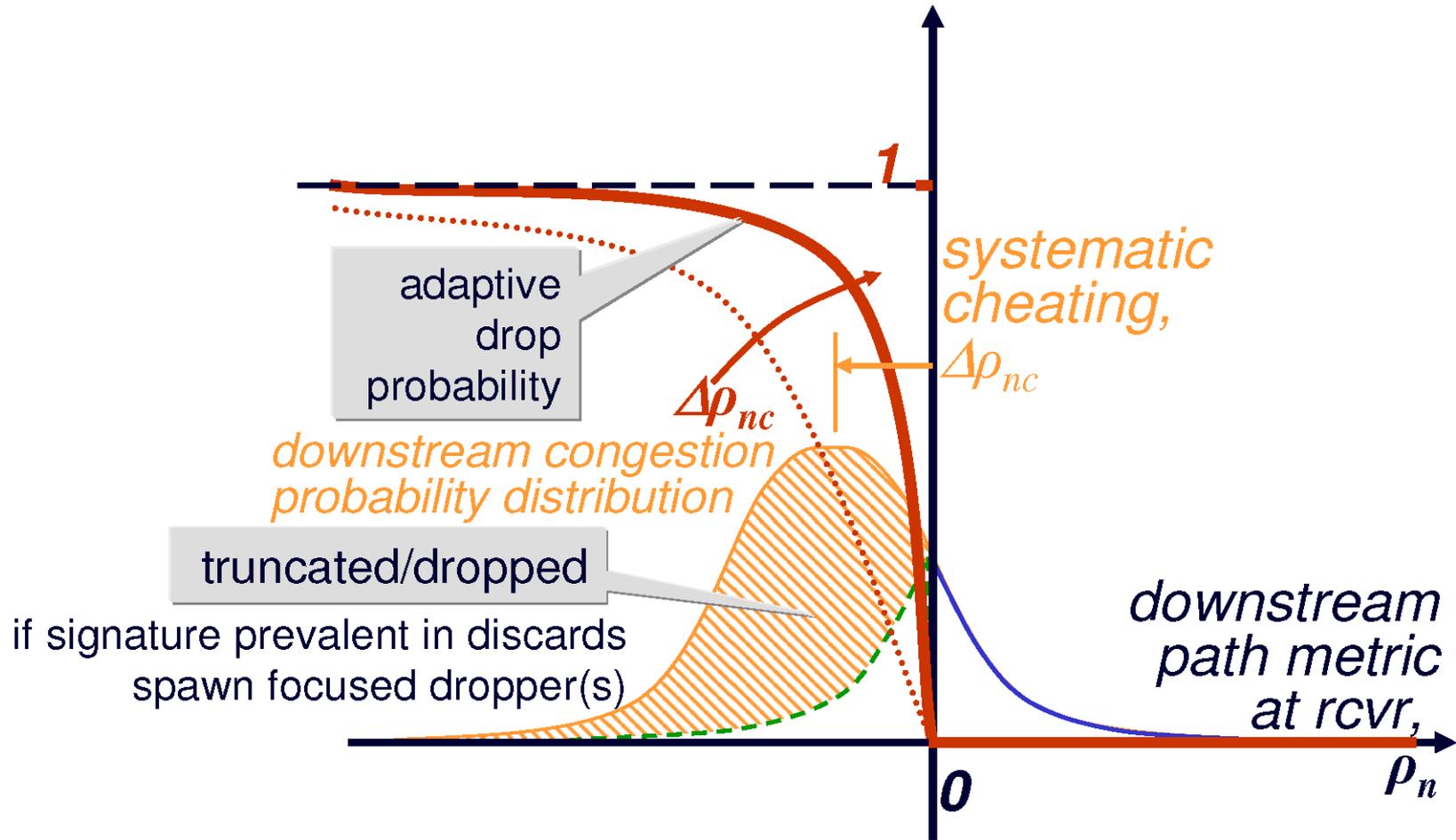
- each packet header carries prediction of its own downstream path

- Throughput drops if the sender has a higher - estimate on downstream congestion

rate

downstream congestion, ρ_i

Adaptive Dropper

downstream congestion
probability distribution
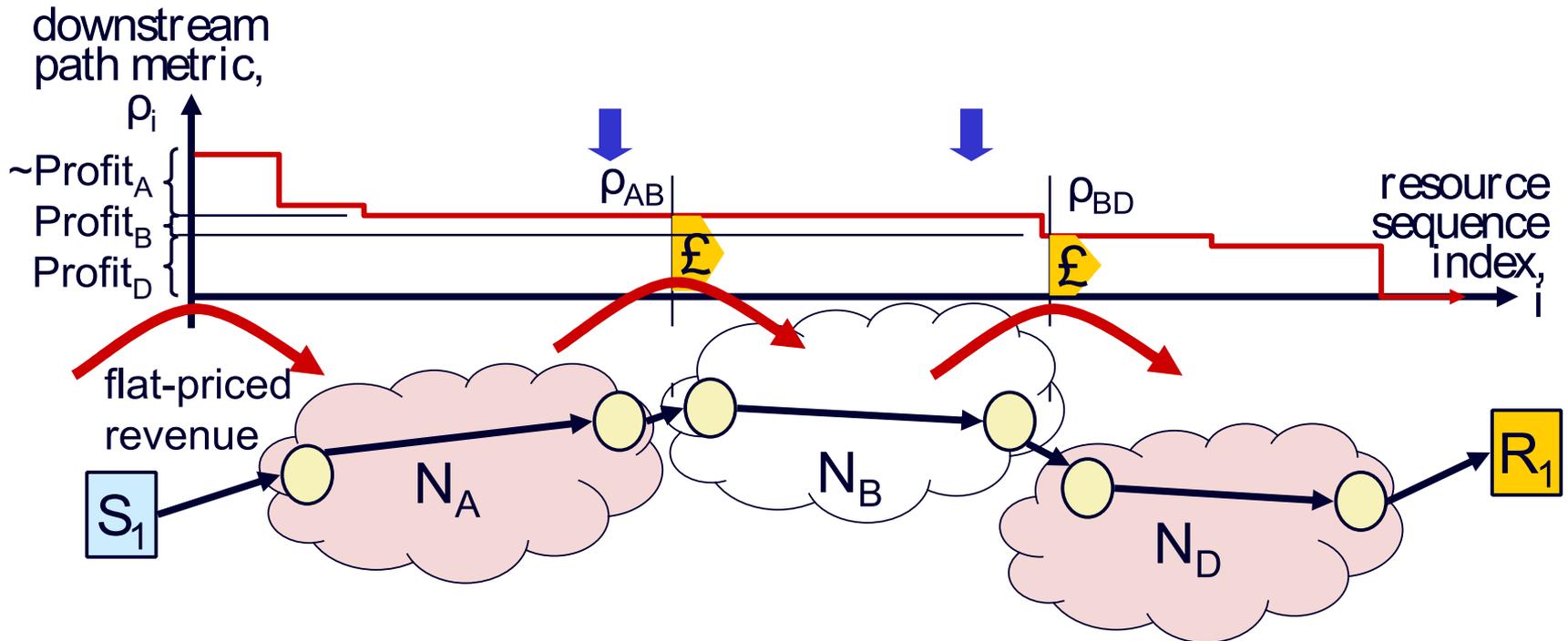
downstream
path metric
at rcvr,

0

ρ_n
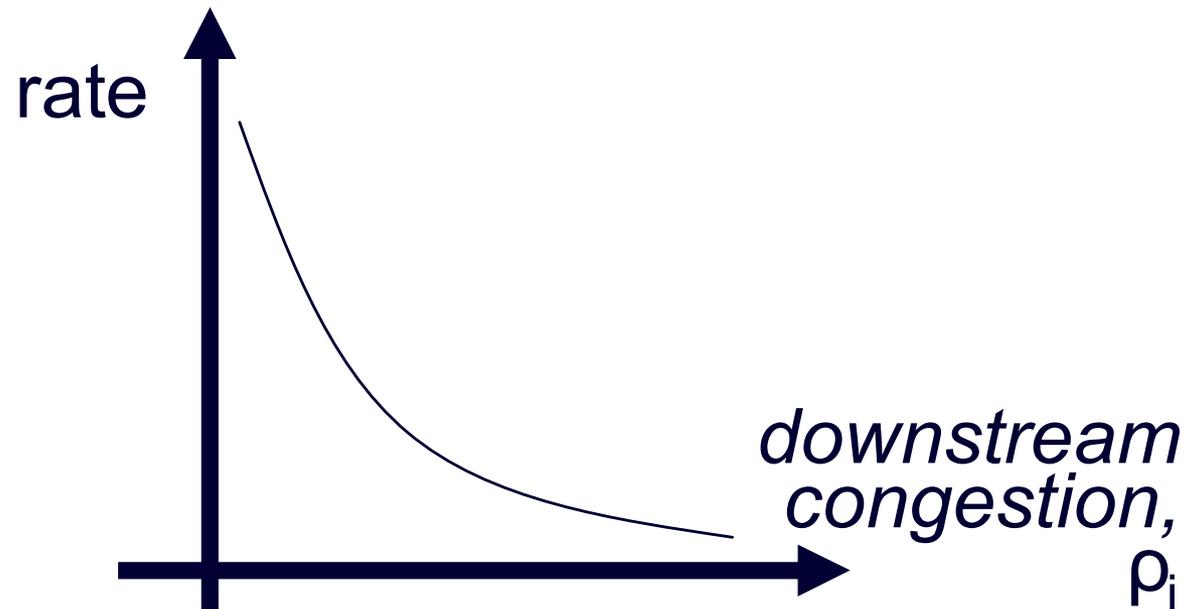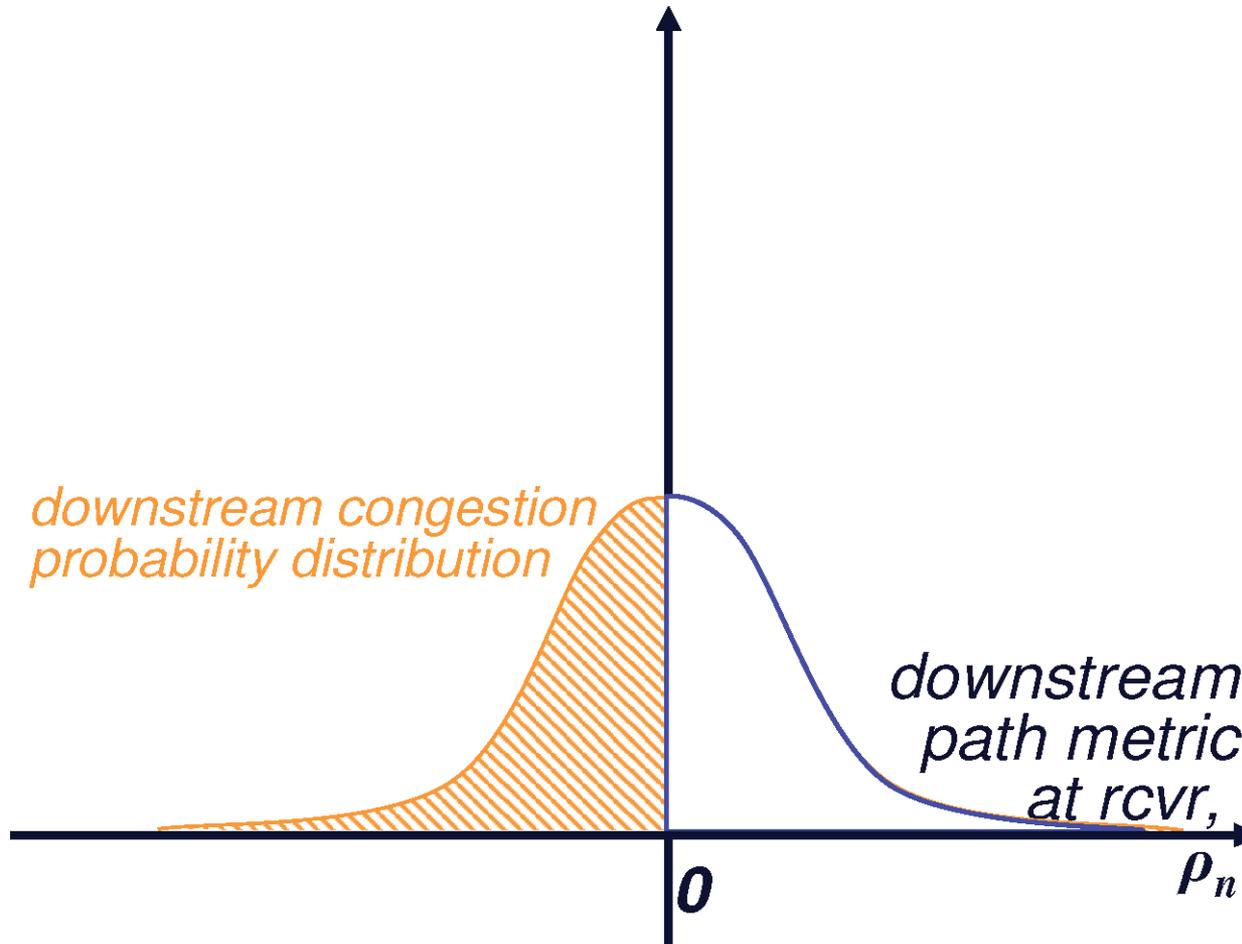
Adaptive Dropper

Cool things: charge for congestion

- Today: 95th percentile bandwidth charging
- ρ is the sum downstream congestion metric
- metered between domains by single bulk counter
- automatically shares congestion revenue across domains

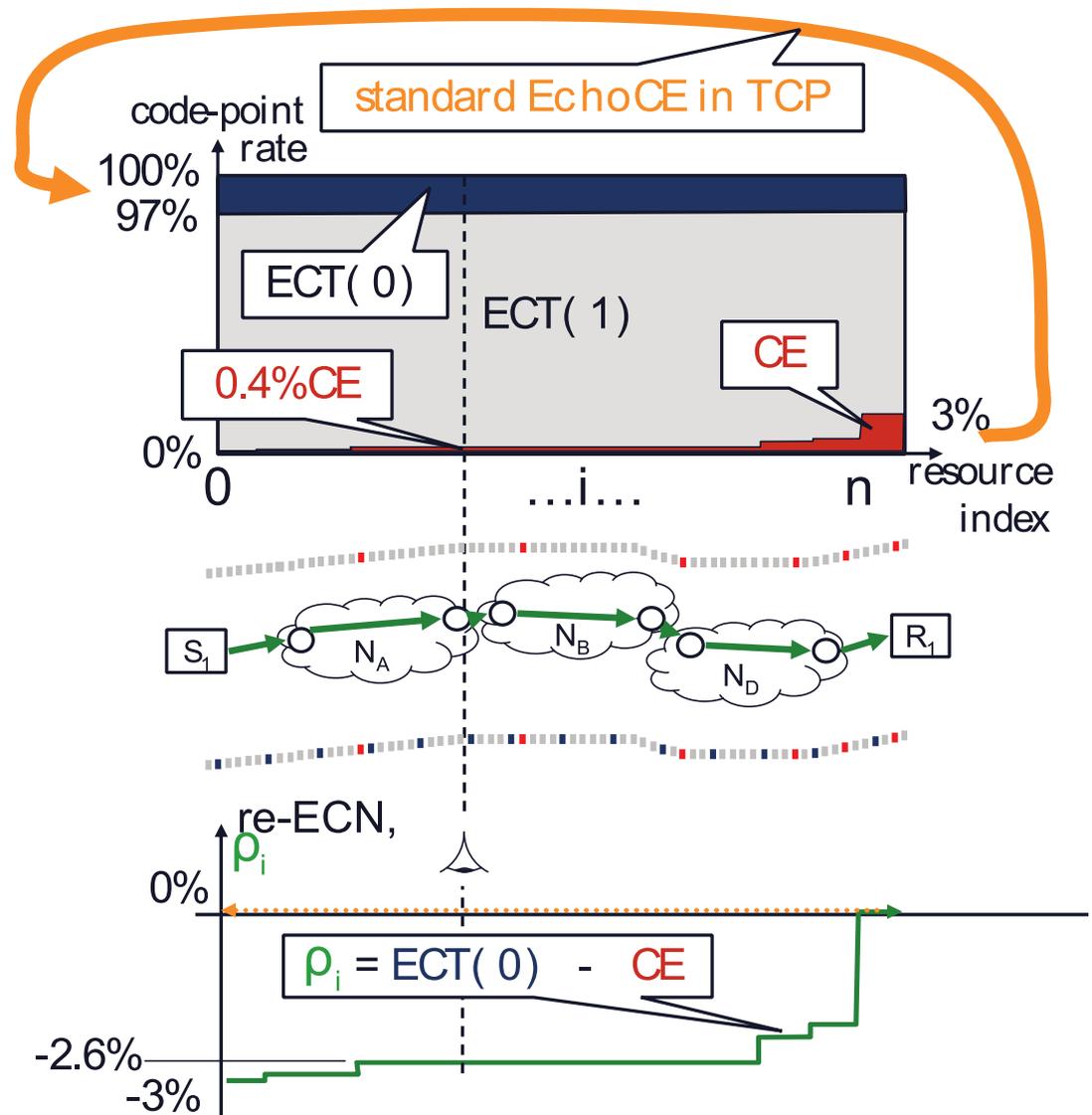# More cool things

- Once timely truthful path visible…
  - Differentiated QoS
  - DDoS mitigation

Protocol Engineering: re-ECN

- on every EchoCE from TCP, set ECT(0)

- at any point on path, diff between rates of ECT(0) & CE is downstream congestion

- works with unchanged routers

standard EchoCE in TCP

code-point rate

100%
97%

ECT(0)

ECT(1)

0.4%CE

CE

3%

0%

0 …i… n resource index

S_1 N_A N_B N_D R_1

re-ECN, ρ_i

0%

ρ_i = ECT(0) - CE

-2.6%
-3%

re-feedback summary

- reinsert feedback to align path characterizations at receiver
- packets arrive at each router predicting downstream path
- arranged for dominant strategy of all parties to be honesty
- a simple idea for the Internet's accountability architecture
- democratizes path information
  - either network or source can control (control requires timely information)
  - designed for tussle

Discussion

- Based on market equilibrium
  - What if market fails?
  - Monopoly paths
- deliberate dilemma: downstream metric during flow start?

initial value of metric(s) for new flows?

- undefined – deliberately creates dilemma
  - if too low, may be dropped at egress
  - if too high, may be deprioritised at ingress
- without re-feedback (today)
  - if congested: all other flows share cost equally with new flow
  - if not congested: new flow rewarded with full rate
- with re-feedback
  - risk from lack of path knowledge carried solely by new flow
  - creates slow-start incentive
  - once path characterized, can rise directly to appropriate rate
  - also creates incentive to share path knowledge
  - can insure against the risk (see differentiated service)