ConEx Abstract Protocol

What's the Credit marking for?

draft-mathis-conex-abstract-mech-00.txt

apologies from **Bob Briscoe**, BT

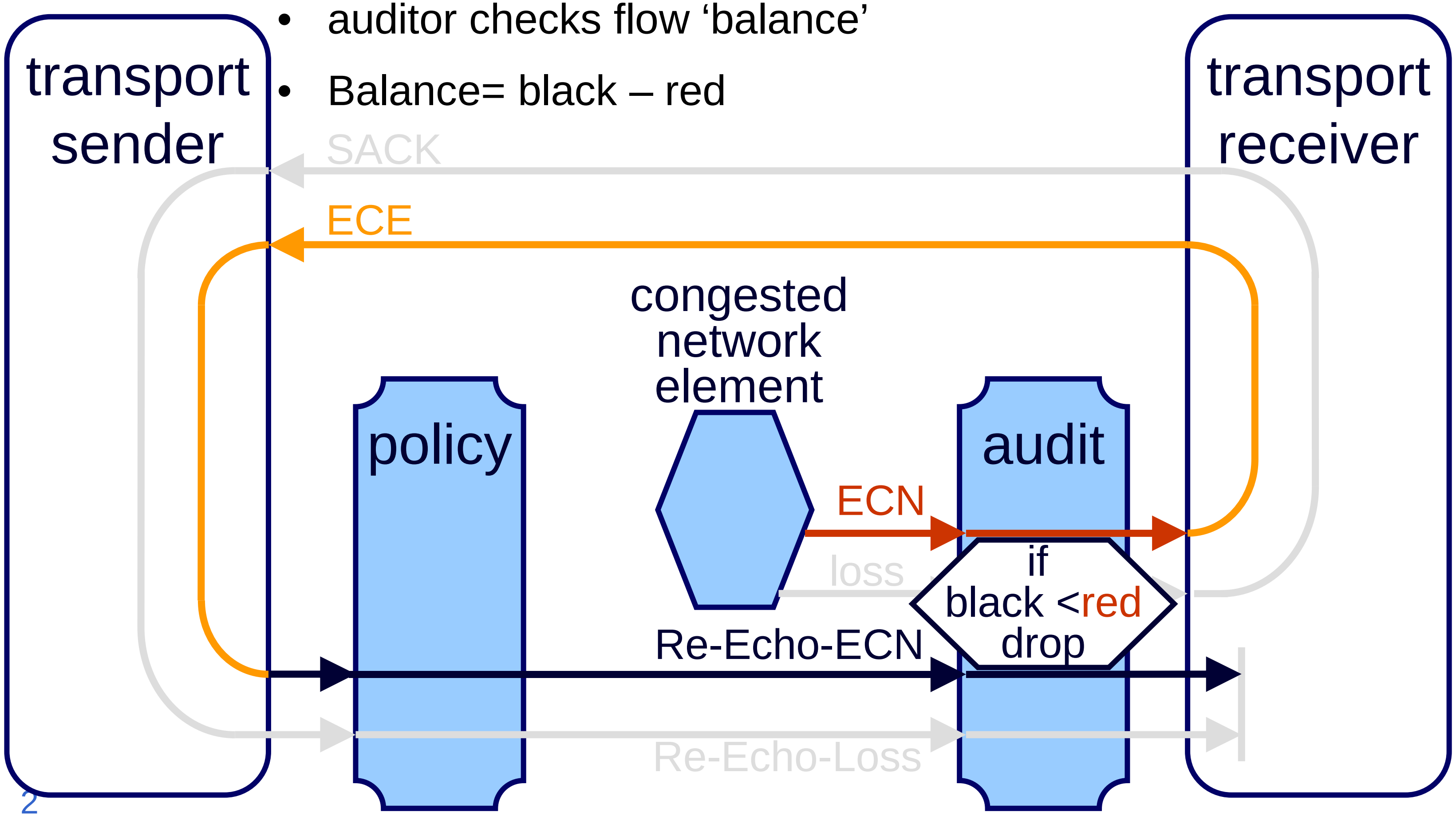
presented instead by Andrea Soppera, BT

IETF-79 ConEx Nov 2010

This work is partly funded by Trilogy, a research project supported by the European Community www.trilogy-project.org

recap: audit function

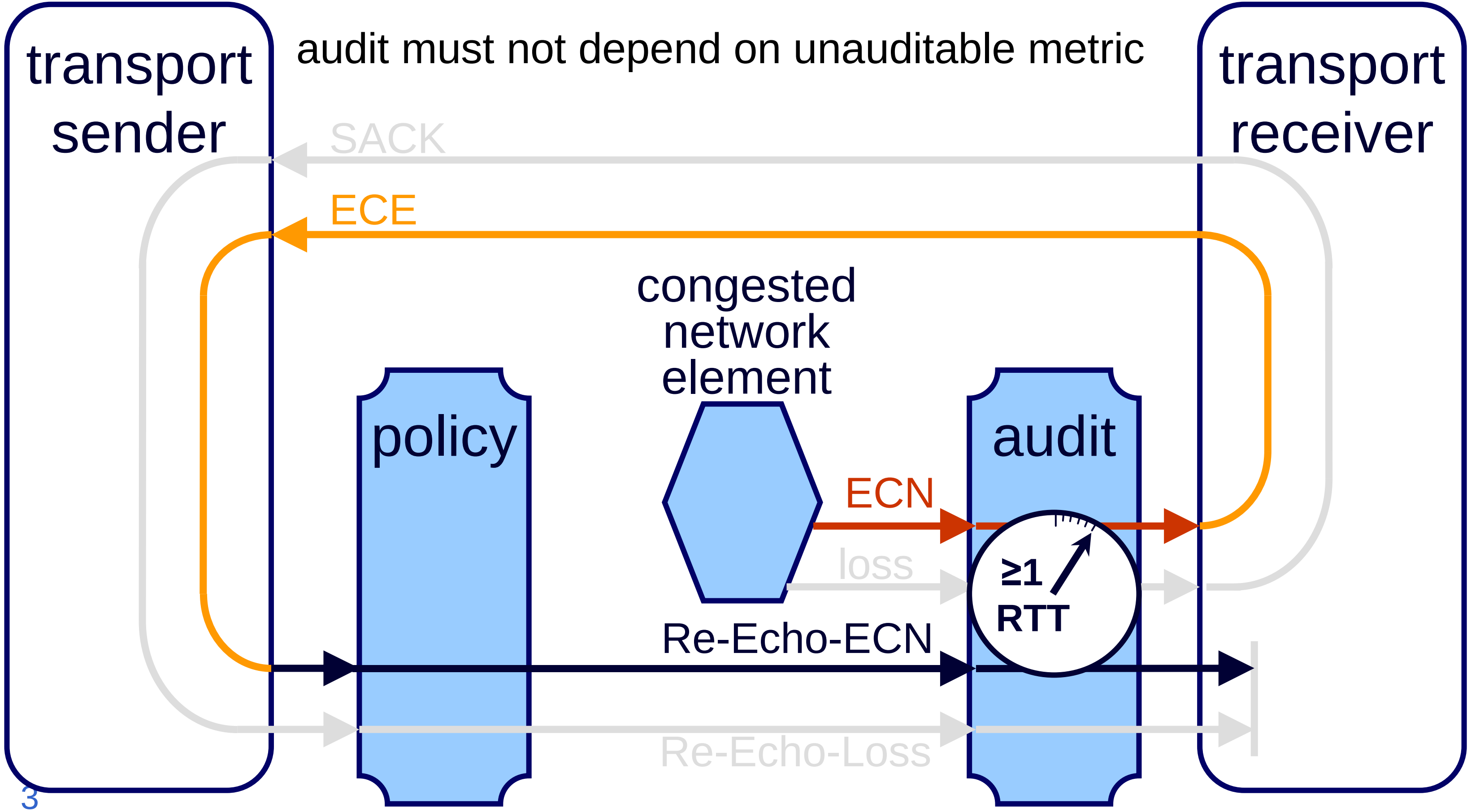
- ConEx signal from sender (black) can be checked against actual congestion signal (red)
  - auditor checks flow ‘balance’
  - Balance= black – red



how does audit handle inherent delay?

- how long to wait from congestion to re-echo?
  - 1RTT? ~20RTT? ∞RTT? (TCP, RTCP,FEC)
  - how does a network node know the transport's RTT anyway?

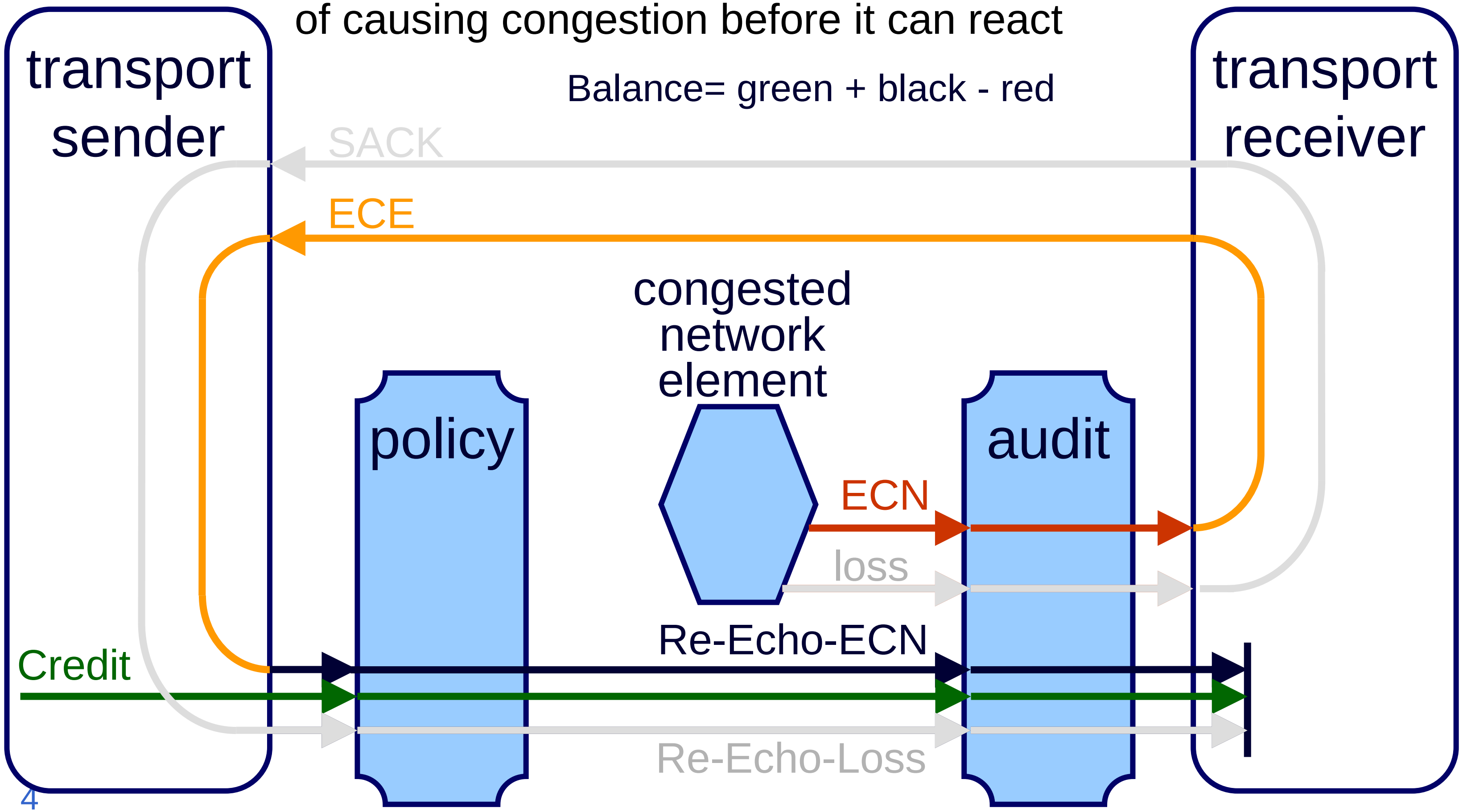
audit must not depend on unauditable metric



solution

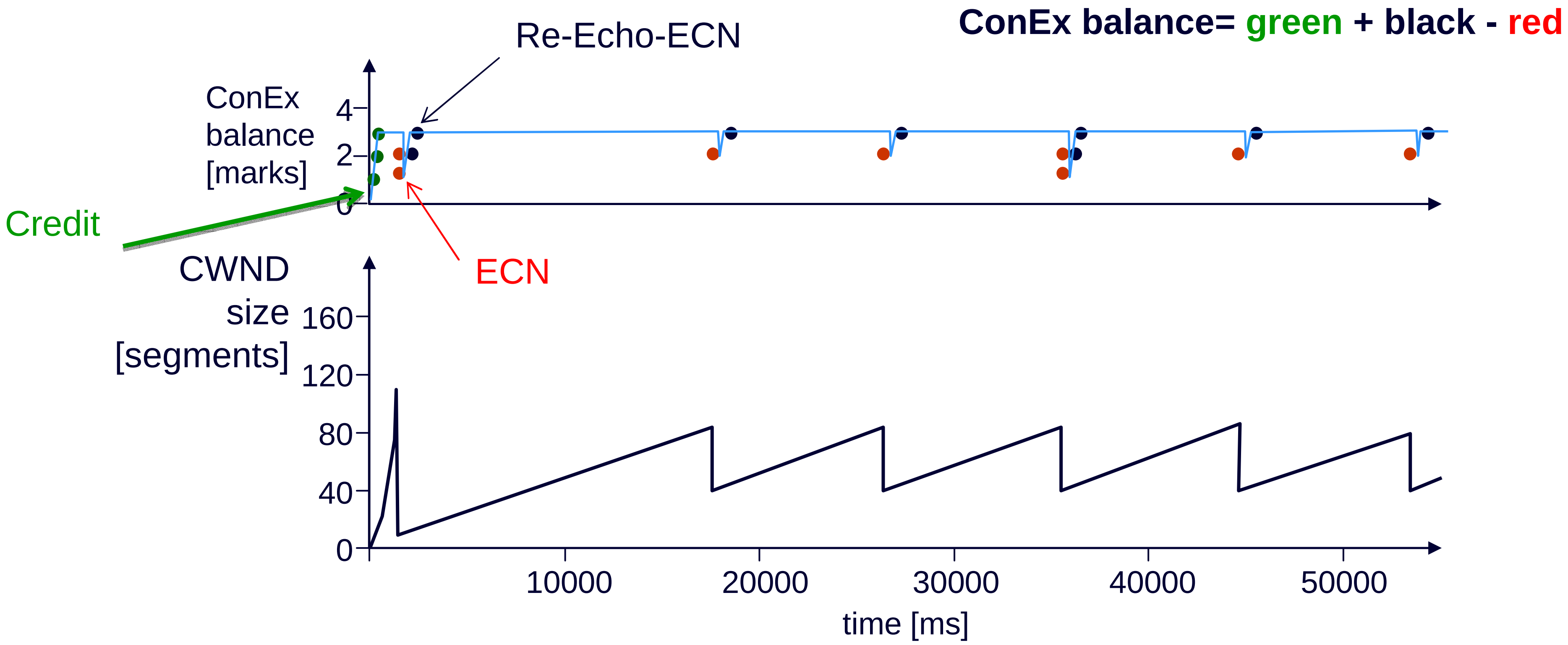
hold transport responsible for delay

- transport must pre-load Credit (green)into loop
  - sufficient Credit (green)marks for expected congestion during delay
  - makes transport accountable for risk
    of causing congestion before it can react



ConEx balance of a TCP connection at a audit device

What would a ConEx signal look like?



auditor needs flow state in network ☹

…but don’t forget

- ConEx only needs flow state to check correctness of *information*

- ConExdoes not embed rules in the network on how flows *behave* unlike many other traffic management approaches such as:
  - flow-state aware routers
  - deep packet inspection (DPI)
  - and other like this…

Summary

What is a credit signal?

- expectation of the worst congestion that a sender is going to contribute to before it can re-echo
- credit is speculative congestion exposure while re-echo reflects actual
- the number of credit that a sender is going to signal will depend on the aggressiveness of the congestion control it uses
    - create correct incentives not to be aggressive
- This presentation is focused on credit signals for auditing - the signal is also useful in other cases but out of scope here

status & plans

- rationale for Credit signal to be added to draft-01
- normative text on design constraints for audit devices
  - Mathis & Briscoe close to agreeing text to add to draft-01
  - informational, but we don’t have a better charter milestone for this
- an audit device design has been implemented
  - resisted various simulated attacks proposed by research community
  - can never prove anything is secure until its broken
  - plan to prepare I-D as a ConEx ‘experience report’

ConEx Abstract Protocol

What’s the Credit marking for?

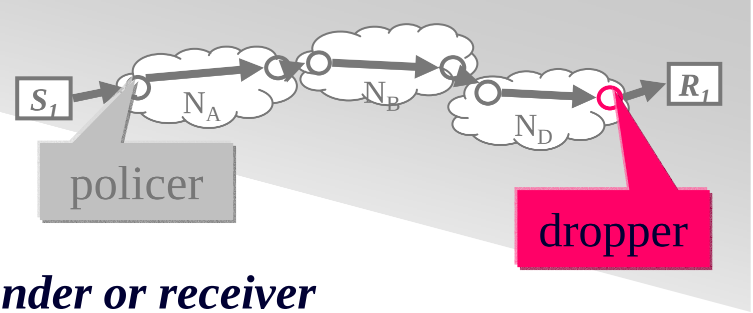
draft-mathis-conex-abstract-mech-00.txt

Q&A

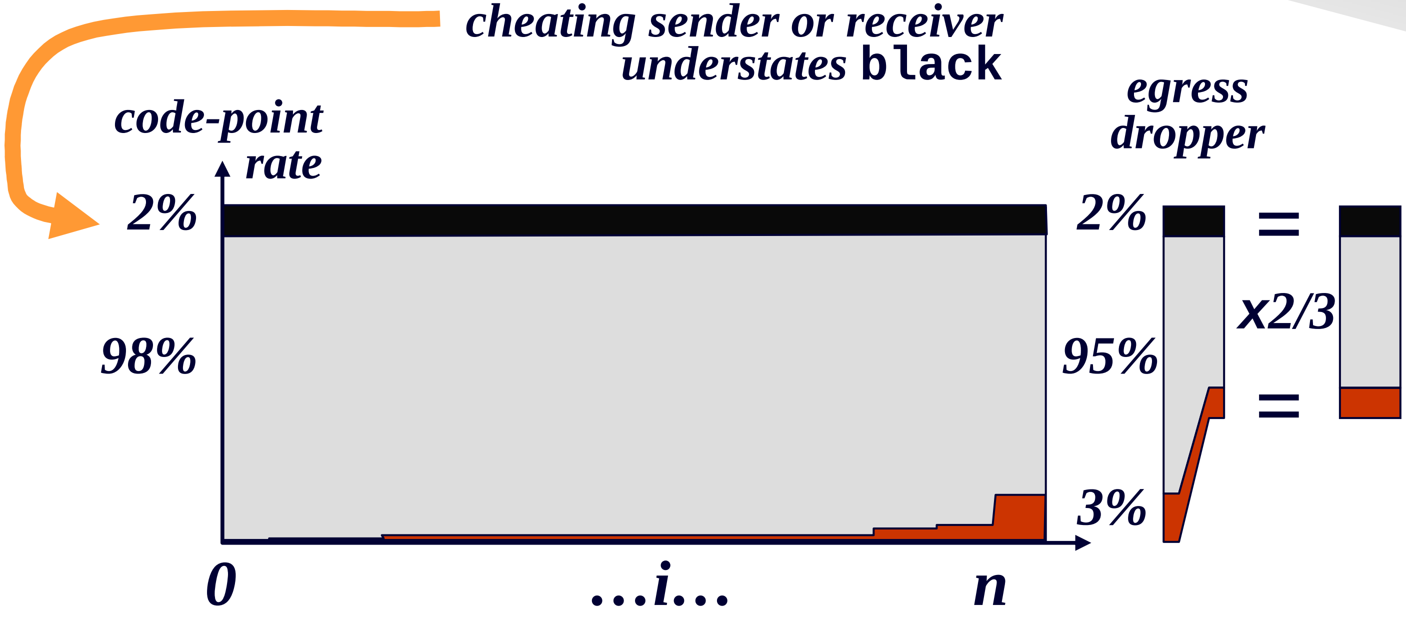
define ‘flow’?

- auditor checks flow ‘balance’
  - should be non-negative at any granularity of identifiers
- microflow granularity may not be visible to auditor
  - due to NATs, tunnelling, etc
- can audit at any level of granularity
  - tunnel, src-dst pair, etc
  - if negative balance, go finer if possible
- finer (and closer to destination) always better

egress dropper (sketch)



cheating sender or receiver understates **black**



- drop enough traffic (black immune) to make fraction of **red** = **black**
- goodput best if rcvr & sender honest about feedback & re-feedback

flow bootstrap

- at least one`green` packet(s) at start of flow or after >1sec idle
  - means “feedback not established”
  - ‘credit’ for safety due to lack of feedback
  - a `green` byte is ‘worth’ same as a **`black`** byte
- a different colour from black
  - distinguishes expected congestion based on experience from based on conservatism
  - gives deterministic flow state mgmt (policers, droppers, firewalls, servers)
  - rate limiting of state set-up
  - congestion control of memory exhaustion

- `green` also serves as state setup bit [Clark, Handley & Greenhalgh]
  - protocol-independent identification of flow state set-up
  - for servers, firewalls, tag switching, etc
  - don’t create state if not set
  - may drop packet if not set but matching state not found
  - firewalls can permit protocol evolution without knowing semantics
  - some validation of encrypted traffic, independent of transport
  - can limit outgoing rate of state setup
- to be precise`green` is ‘idempotent soft-state set-up codepoint’

flow state in network?

three separate reasons for avoiding network flow state

a) pins flow to path ⇐not an issue
b) state attacks ⇐ not an issue
c) memory cost ⇐auditingcannot avoid this ☹

a) auditor’s flow state is soft

- if flow moves, ConEx markings recreate state in another auditor

b) auditor requires credit marking before allocating flow state

- ingress policers can then limit influx of credit markings
- flow state exhaustion attacks (incl. SYN attacks) thwarted at source
- servers/firewalls under stress can also prefer new flows with credit marking

c) cannot avoid memory cost

- only need full per-flow auditing once, at egress of internetwork
- clever hardware implementers may design better scaling

discussion

is Credit / Re-Echo distinction worth 2 codepoints?

- for w-g to discuss/decide
  - depends how much space we find for encoding
- more benefits than mentioned so far
  - distinguishes actual vs. speculative congestion exposure
    - useful for bulk monitoring as well as per-flow mechanisms
  - benefits of Credit as a flow state set-up flag
    - hook for e2e session congestion control
    - hook for link layer cut-through optimisations (cf. tag switching)
    - etc