DDoS remediation with eBPF

Steven Simpson
Noor Shirazi
David Hutchison

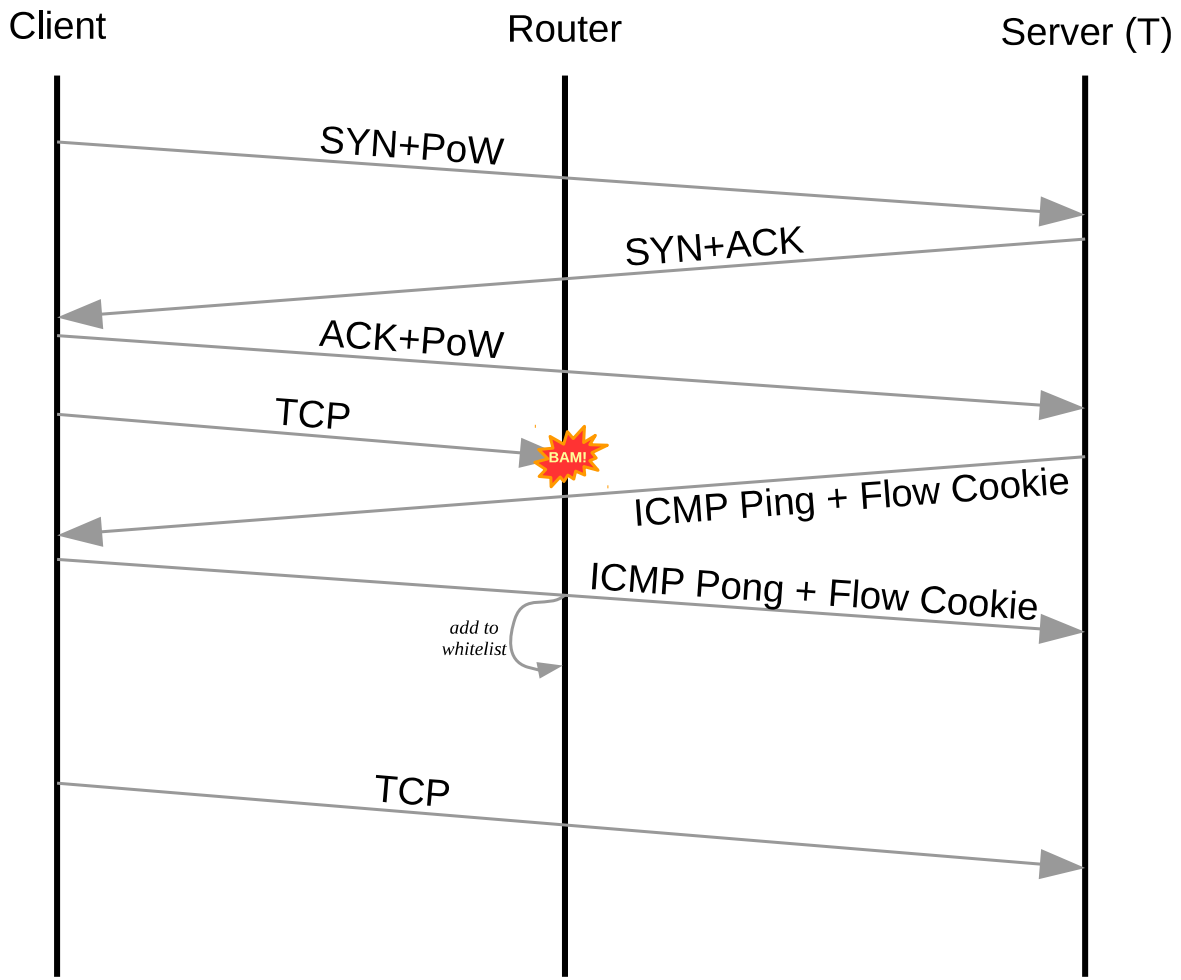
School of Computing and Communications
Lancaster University

Simon Jouet
Dimitris Pezaros

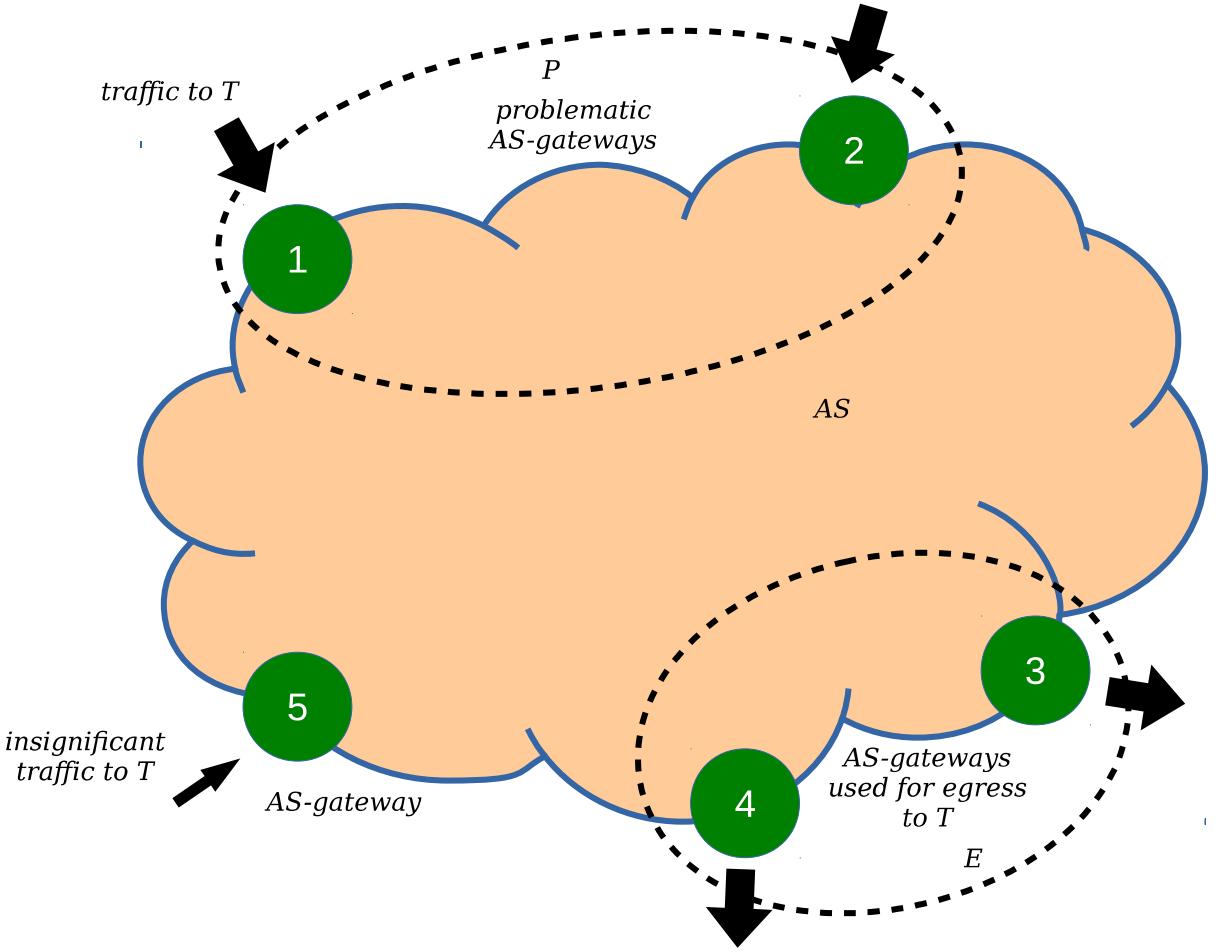
NETLAB
NETWORKED SYSTEMS RESEARCH LABORATORY
University of Glasgow | School of Computing Science

A Situation-Aware Information Infrastructure
(EPSRC EP/L026015/1)

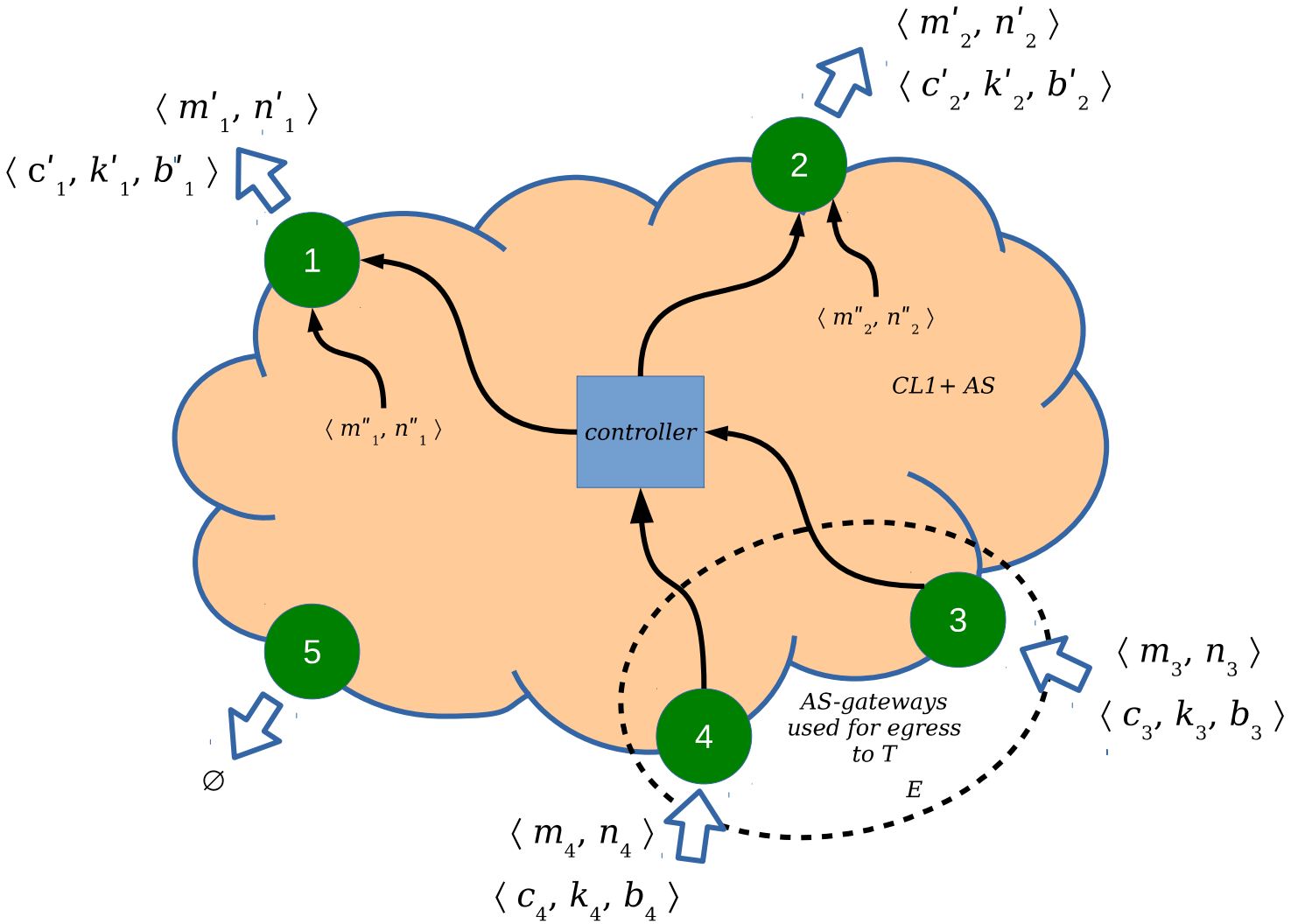
Proof-of-work & flow cookies



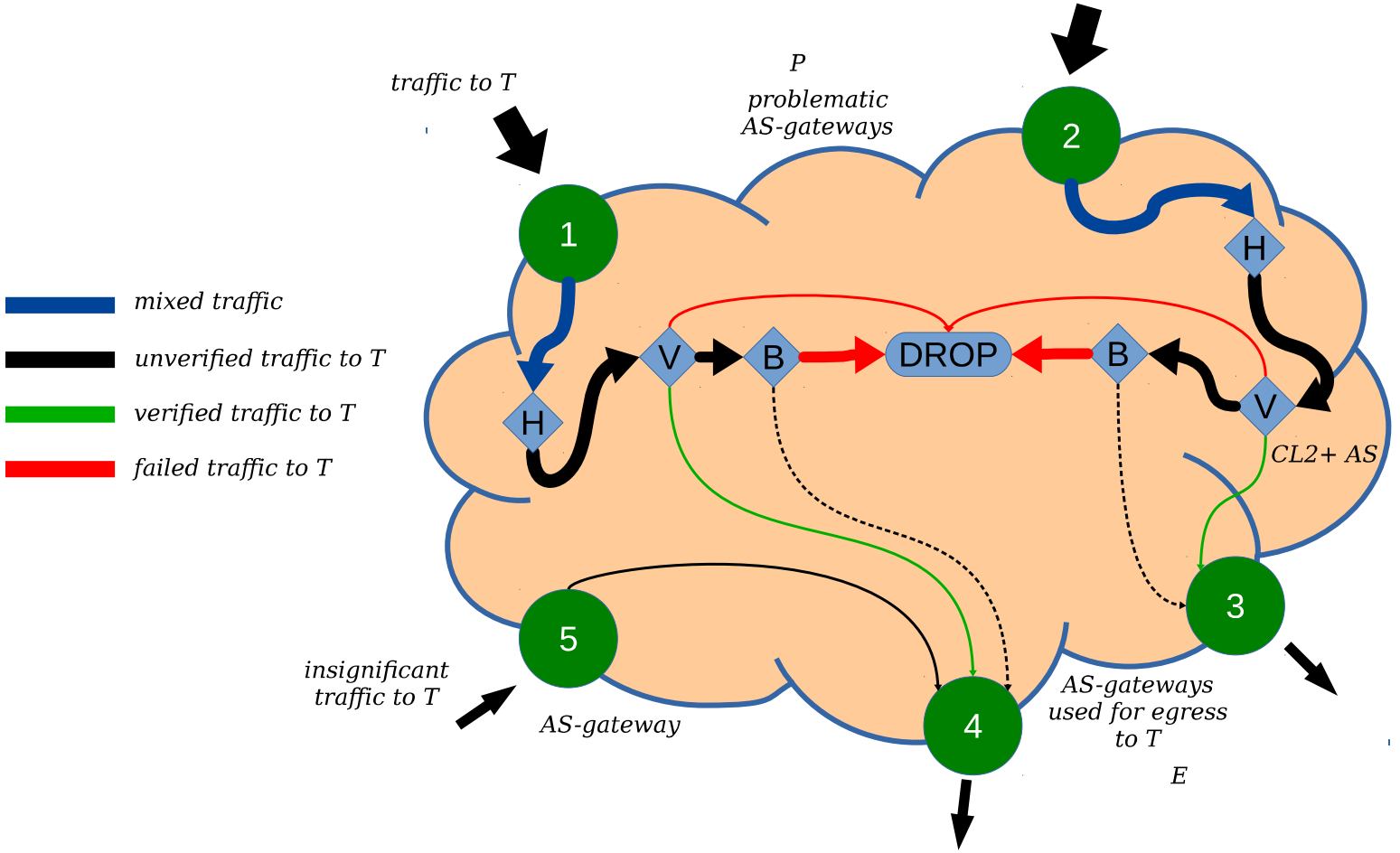
AS categorization



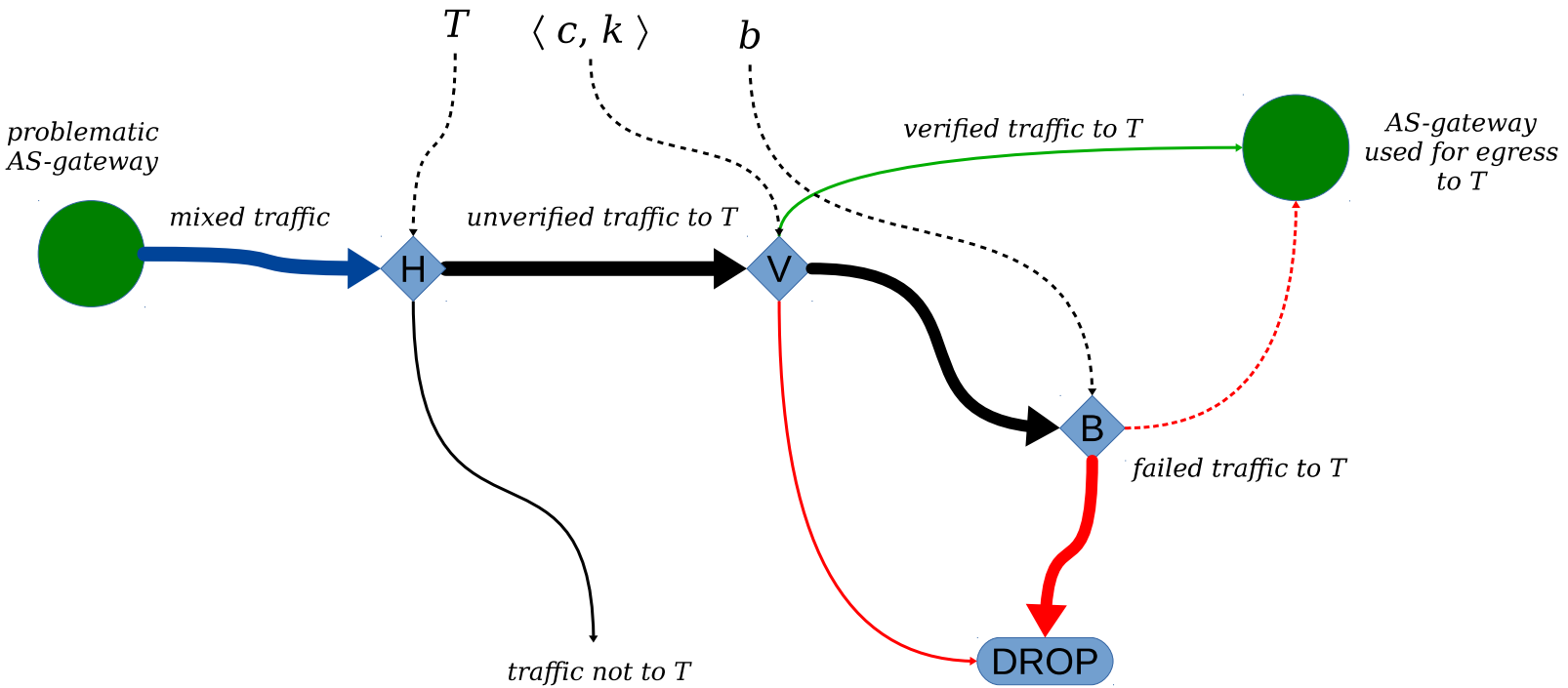
AS information flow



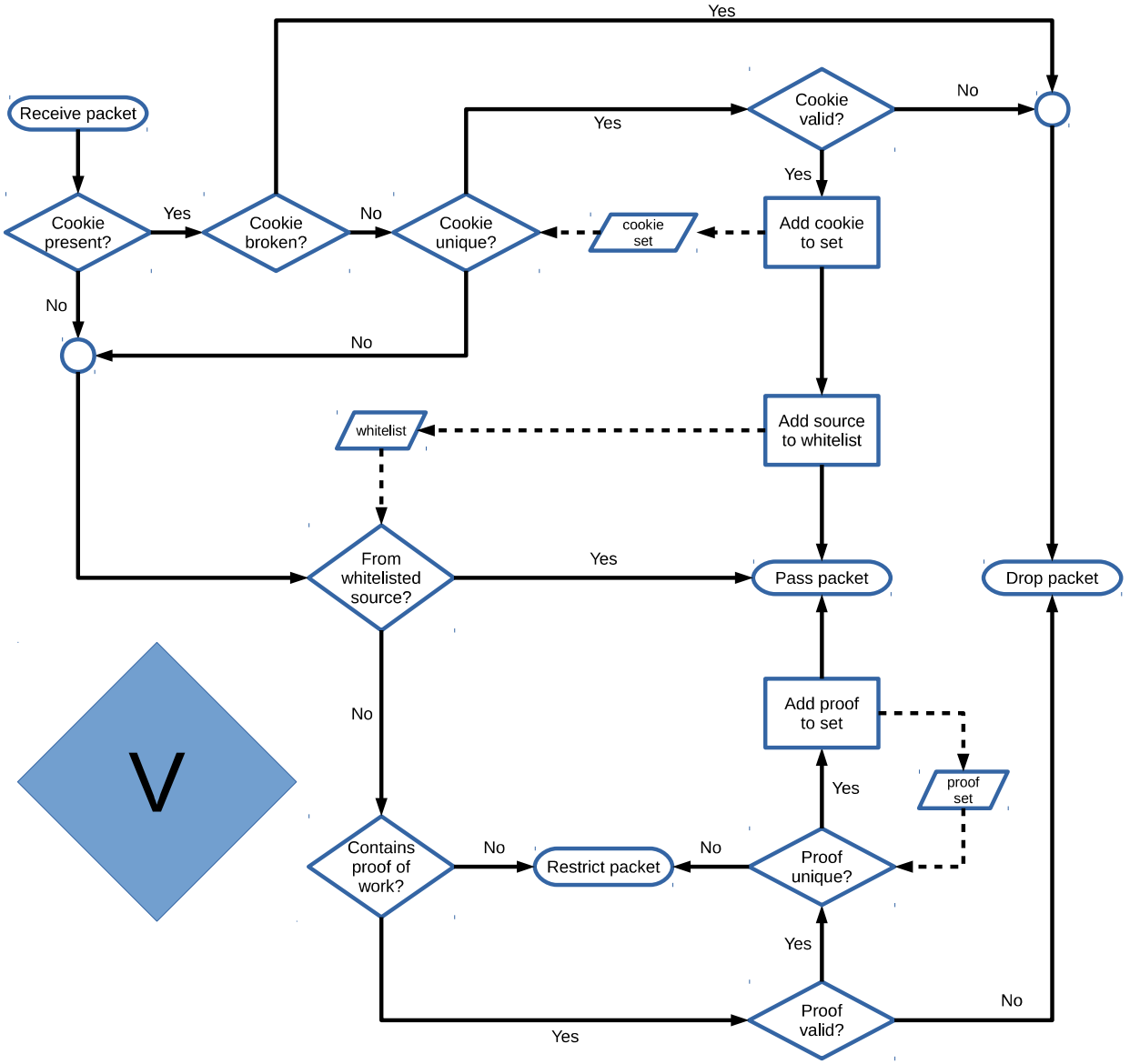
Filtering in AS



Filtering in AS



Verification filter



Verification filter

- Check for cookie before whitelist
  - Need to see refreshing cookies
- Check whitelist before proofs-of-work
  - PoW not useful to us if already whitelisted
  - Still useful downstream
- New attack vectors
  - Flood with invalid cookies
    - Requires asymmetric signature verification per packet
  - Flood with invalid proofs
    - Requires hash computation per packet

Signature verification in hardware

- NetFPGA implementation?
  - Translate from C
    - Restricted language
      - No unbounded loops
  - Tricky to set up
    - for our needs

- eBPF implementation
  - Translate from C
    - Still restricted
    - Push complex functions to edge of EE
      - Hashing
      - Signature verification
  - Validate on software switches
  - Switch between alternative eBPF EEs
    - Future NetFPGA implementation of eBPF?
    - Operational portability?

Outstanding questions

- How fast can we go?
  - In each EE
- Still issues with the DDoS plan
  - Can we avoid flooding the network with PoW parameters?
    - Restrict to areas with clients?
    - Find another distribution system?
  - Deliver parameters/credentials with BGP?
  - Needed? Is spoofing still a problem?
- What h/w-assisted functions can we expect in real environments?
  - Hashing?
  - Signature verification?
  - Can we use this application to drive the design of future h/w and SDN functionality?

Thanks!
Hmm?