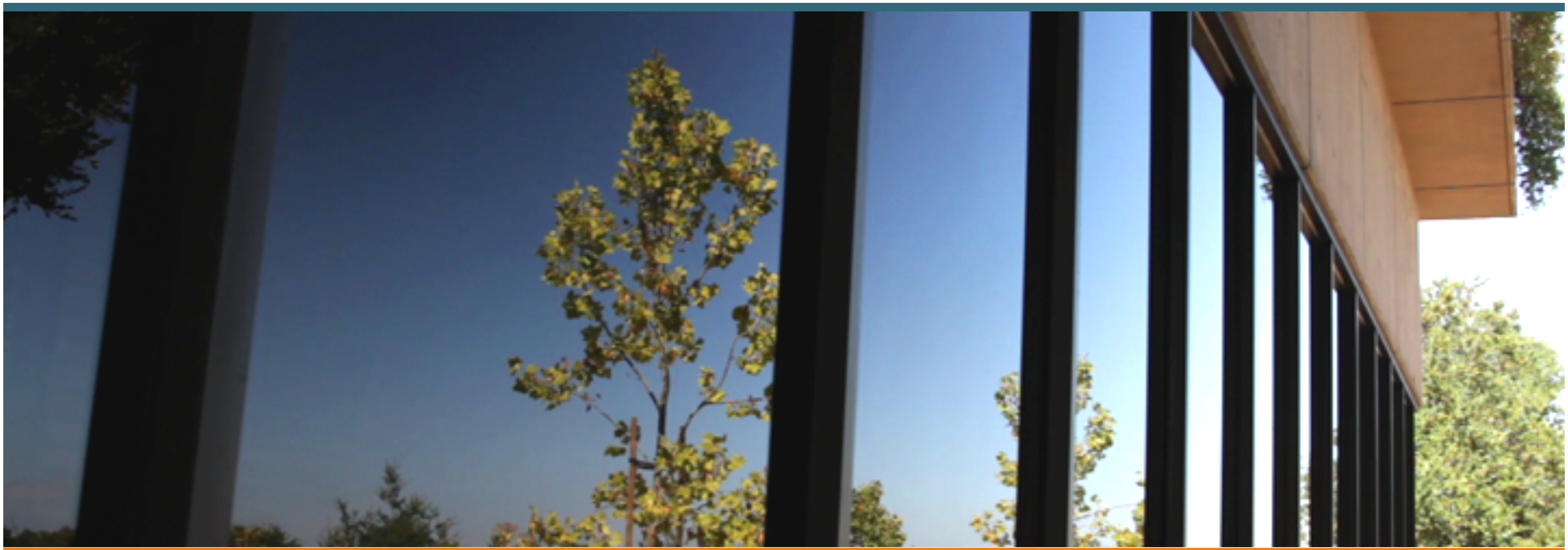
parc
A Xerox Company

(D)DoS in CCN: Evaluation & Countermeasures

September 12, 2012



Ongoing collaborative work between PARC, UCLA and UCI

Presenter: Ersin Uzun

What is DoS attack?

- Goal: Prevent legitimate resource usage
  - i.e., attack on availability

- Resources, e.g.,
  - Memory, CPU, Bandwidth, Storage etc.

- Distributed DoS (DDoS) attacks are common on today's Internet

(D)DoS in CCN: IP vs. CCN

- CCN is fundamentally different than IP
  - Not push based
    - Data transmission must be preceded by a request for that data.
    - Most DoS attacks in IP are possible because unsolicited data can be sent anywhere
  - Reliable Forwarding Plane
    - Interest and data follow the same path (i.e., immediate feedback to routers)
    - Easy to secure routing (remains challenging in BGP)
    - Better resiliency with multi-path routing
  - Most current DoS attacks on IP are not applicable to CCN.
- What about new DoS attacks, specific to CCN?

(D)DoS in CCN: Two Major Threats

- Content Poisoning:
  - Adversary introduces junk or fraudulent content
    - Pollutes router caches and consumes bandwidth
    - Invalid signatures or valid signatures by invalid producers
  - Not easy to implement: cannot unilaterally push content
    - there will likely be trust mechanisms to register namespaces, etc.
- Interest flooding:
  - Adversary injects a large number of spurious interests
    - Non-sensical distinct interests: not collapsible by routers
    - Consume PIT state in intervening routers as well as bandwidth
    - Legitimate CCN traffic suffers…
  - Easy to implement
  - Current CCNx has no countermeasures implemented

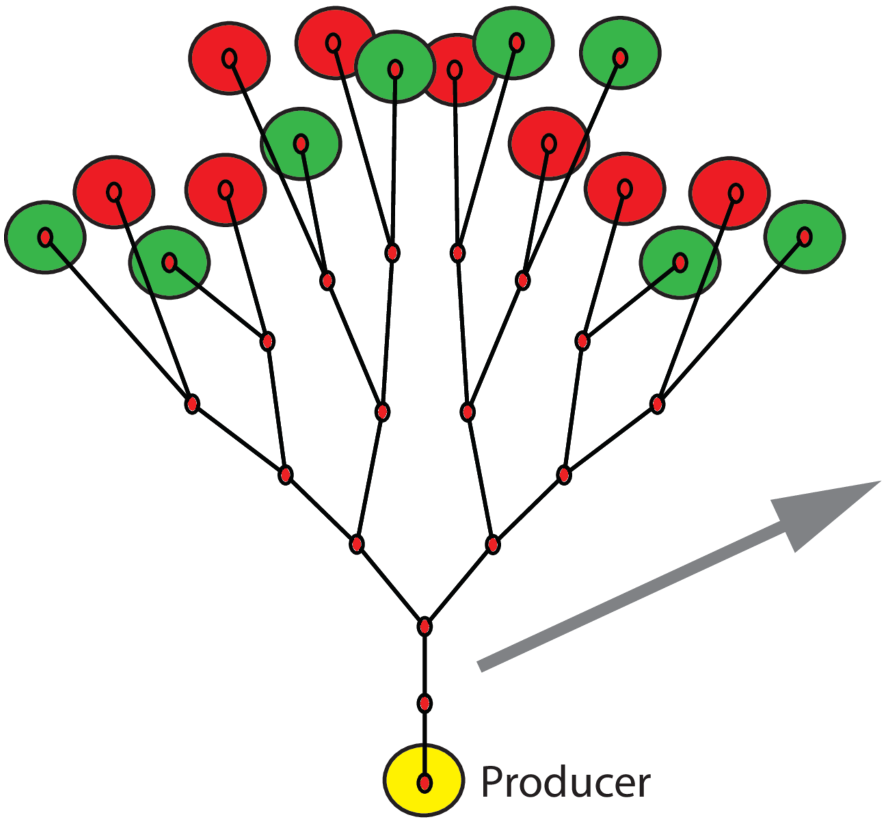
This talk is focused on interest flooding

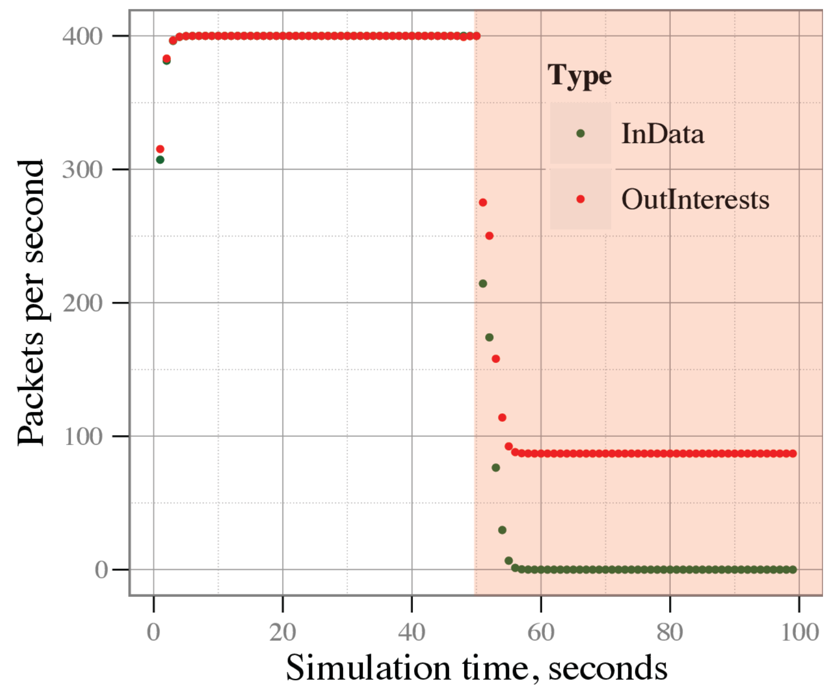
Interest Flooding Attacks

- Why interest packets could be used for DoS?
  - Interests are unsolicited
  - Each non-collapsible interest consumes state (distinct PIT entry) in intervening routers
  - Interests requesting distinct data cannot be collapsed
  - Interests (usually) routed towards data producer(s)

- Can such attacks be prevented?
  - Short Answer: Yes
  - Unlike IP routers, ccn routers maintain rich state information that can be used to detect and react to interest flooding

Picture of an (interest flooding) attack





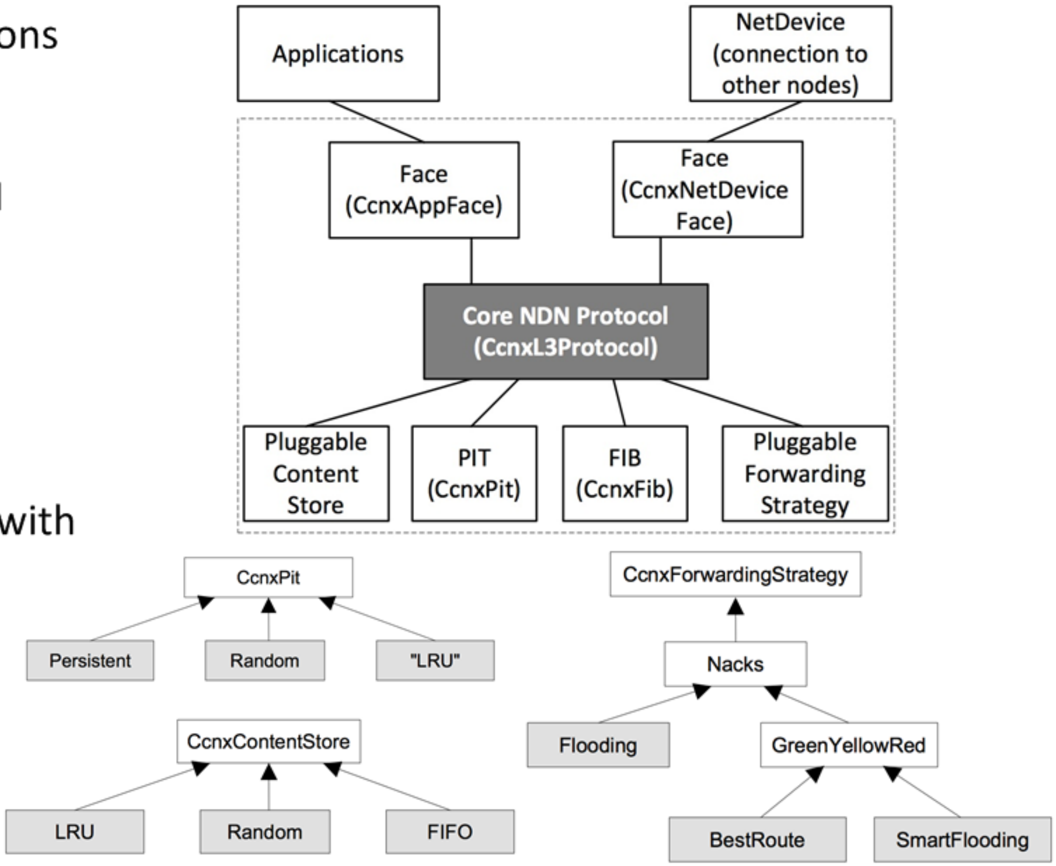
Exploring the solution space

- Simulation-based small experimentations
  - ndnSIM modular NDN simulator
    - http://ndnsim.net
  - different scale topologies
    - binary trees (3, 31, 128 nodes)
    - 10Mbps links
    - propagation delays randomized from range 1-10ms
    - No caching (worst case scenario)
  - simple attacker model
    - Sends targeted interests (common prefix) for non-existing content
    - up to 50% attacker population
  - various mitigation techniques
- Emulation-based verification
- Large scale simulations for promising mitigation techniques

Brief intro to ndnSIM

ndnSIM: NS-3 based NDN simulator

- Simulated basic NDN operations
- Modular, easy extensible architecture
  - C++ classes for every NDN component
    - Face
    - PIT
    - FIB
    - ContentStore
- Packet-level interoperability with CCNx implementation
- Pluggable modules
  - Forwarding strategy
  - PIT
  - Content Store



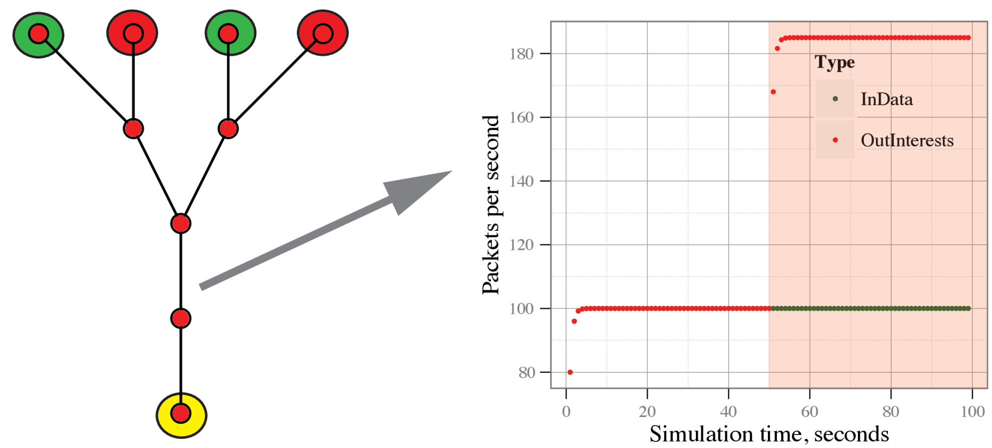
Respecting physical (bandwidth) limits

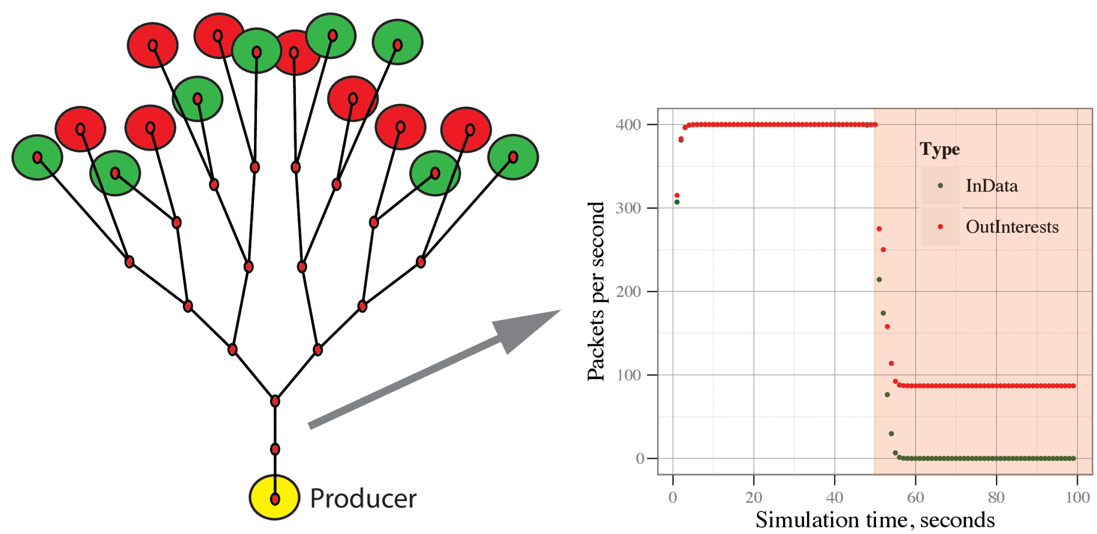
- Current CCNx code does not limit the PIT size, or the # of Pending interests for any interface
  – Downstream can send more interests than physically possible to satisfy.
- CCN has balanced flow between Interests & Data
  – Number of Interests defines upper limit on Data packets
- The number of pending Interests to fully utilize a link with data packets is:

$$\text{Interest limit} = \text{delay}(\text{s}) \cdot \frac{\text{bandwidth (Bytes/s)}}{\text{avg data packet size (Bytes)}} + \varepsilon$$

That limit alone is not sufficient

- In small topologies, prevents attackers from injecting excessive # of interests.
- As expected, it does not work in big topologies
  - No differentiation between good and bad traffic.





Utilizing the state information in routers

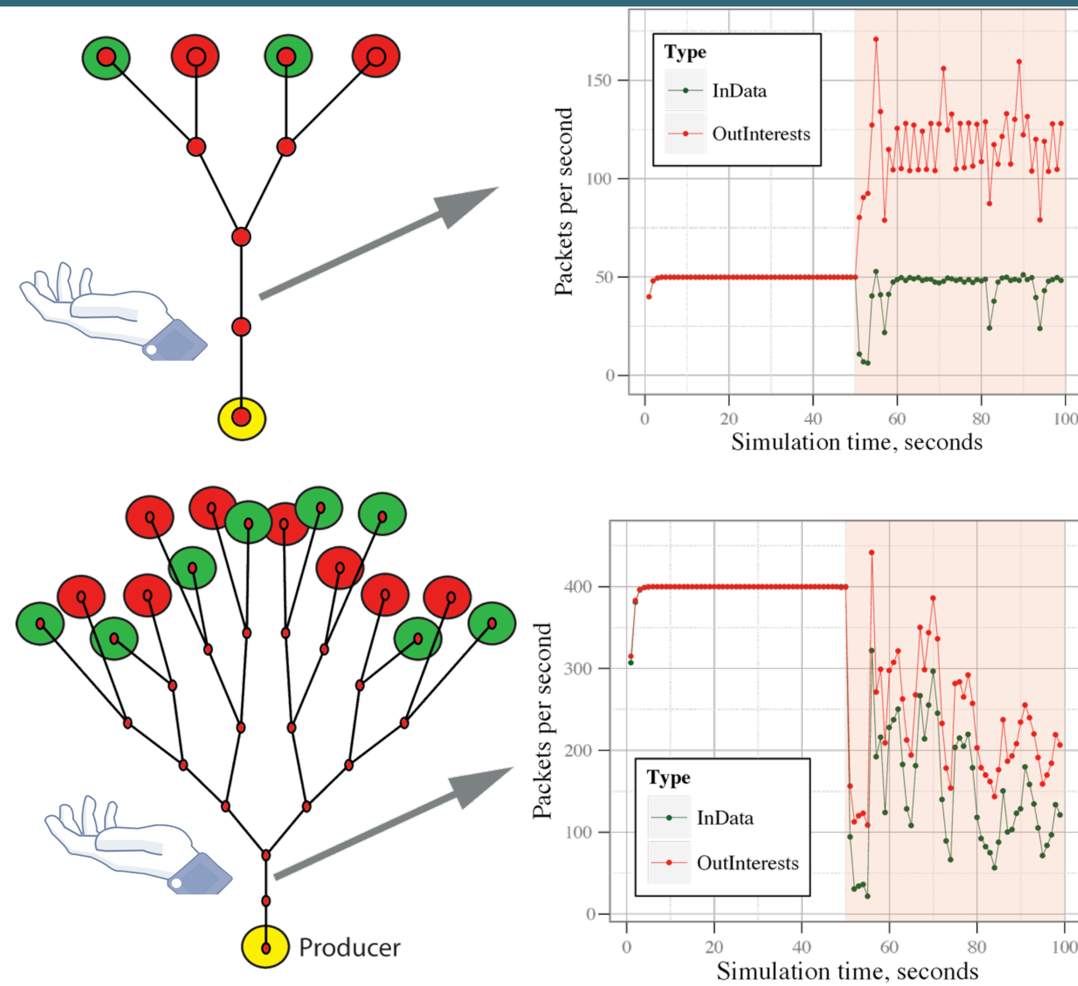
- Theoretically, CCN routers have all the information needed to be able to differentiate good interests from bad ones.
  - To be effective in DoS, bad interests need to be insuppressible and requesting non-existing content.
  - On the other hand, good interests will likely be satisfied with a content
- Keep per incoming interface, per prefix (FIB entry) interest satisfaction statistics in routers
- Use the statistics to detect and control bad traffic.

Weighted round-robin on interest queues

- when an Interest arrives
  - If (per-prefix/per-face) pending Interest limit is not reached
    - accept Interest and create PIT entry
  - If limit is reached
    - “buffer” Interest in per-outgoing face/prefix queue (within per-incoming face sub-queue)
    - set weight for per-incoming face sub-queue proportional to observed interest satisfaction ratio
  - when new PIT slot becomes available
    - accept and create PIT entry for an Interest from queues based on weighted round robin sampling

Weighted round-robin results

- **Partially works**
  - more fair share of resources
  - Not very effective at differentiating bad and good traffic (no-cache scenario)
  - Setting queue sizes and lifetime can get tricky
  - Will most likely improve if supplemented with NACKS (under testing)

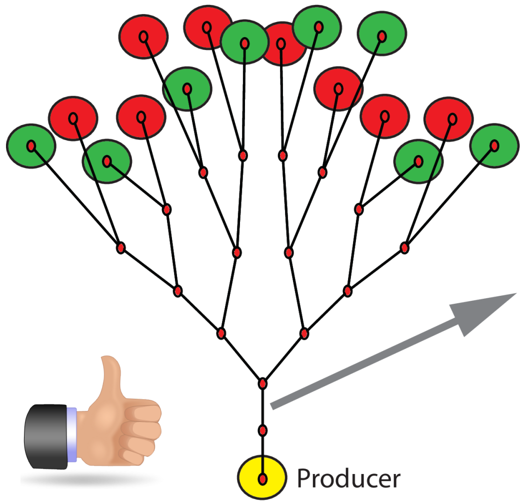


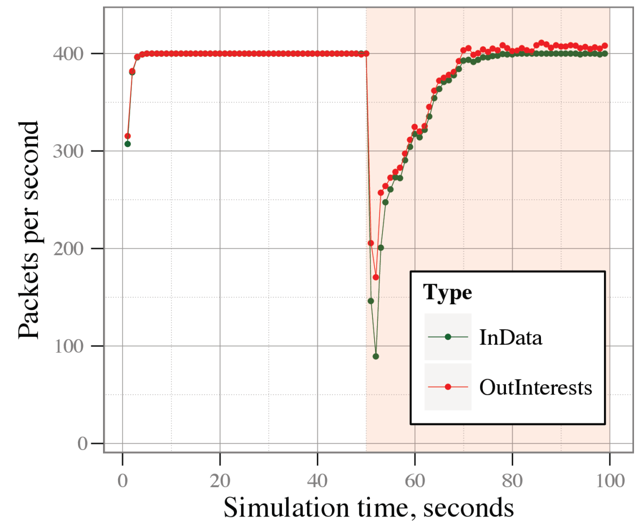
Probabilistic Interest acceptance/drops

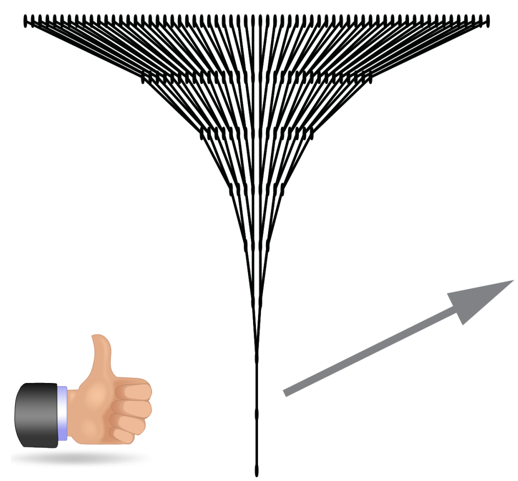
- When an Interest arrives
  - "accept" if the outgoing face is utilized under a threshold
  - Otherwise, accept with probability proportional to the satisfaction ratio for Interests on this face and per-prefix
  - Even if satisfaction ratio is 0: "accept" with a low ("probe") probability
- All "accepted" Interests are still subject to (per-prefix/per-face) pending Interest limit

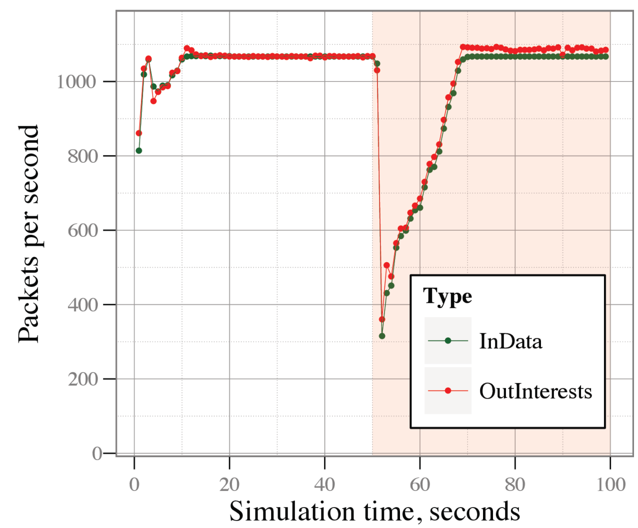
Probabilistic Interest acceptance Results

- Parameter selection is important but may not be easy due to topology variances.
- May result in link under-utilization
- Works in general,
- Might perform better with NACKs (more accurate statistics)









Dynamic Interest limit adjustments

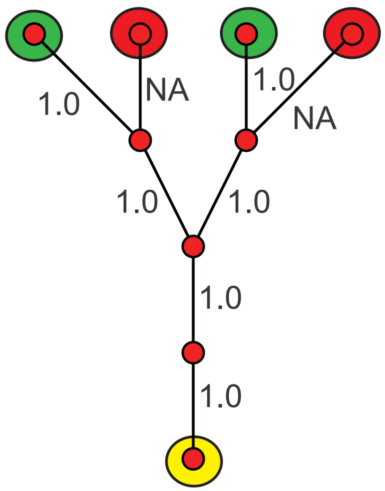
- Incorporate "active" PIT management
  - Periodically
    - for every FIB prefix
      - for all faces
        - » Announce NoPI limit proportional to the satisfaction ratio
  - Min limit is 1 and sum of all announced limits is at least equal to sum of output limits

$$\sum_{face}(ratio_{face} \times Limit_{out}) \geq Limit_{out}$$

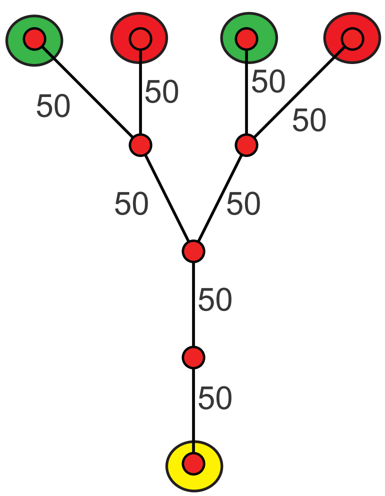
Illustration of Dynamic limits

Just before the attack

Satisfaction rates

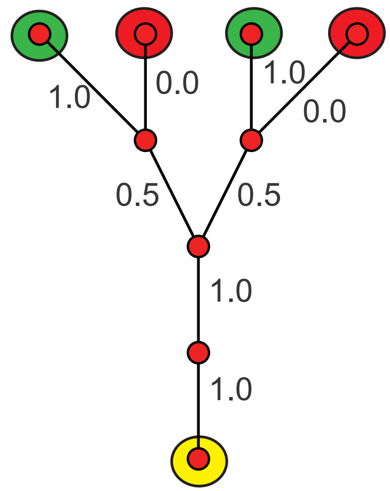


Announced limits

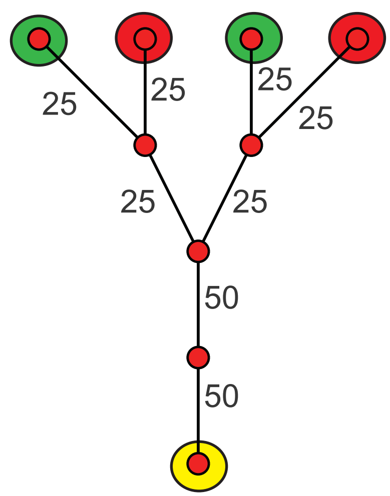


Immediately after attack starts

Satisfaction rates

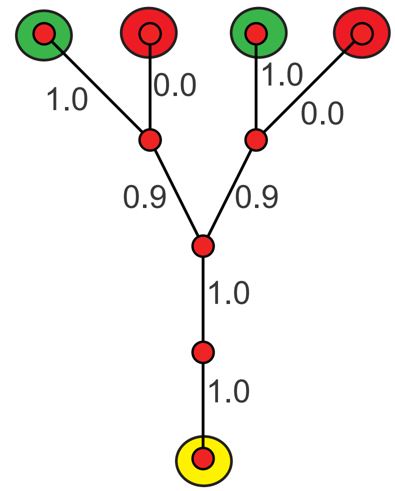


Announced limits

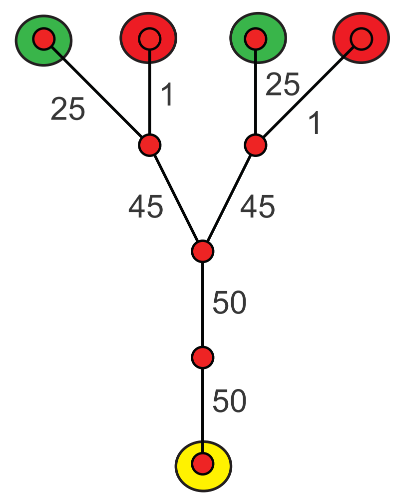


Stabilized after the attack

Satisfaction rates

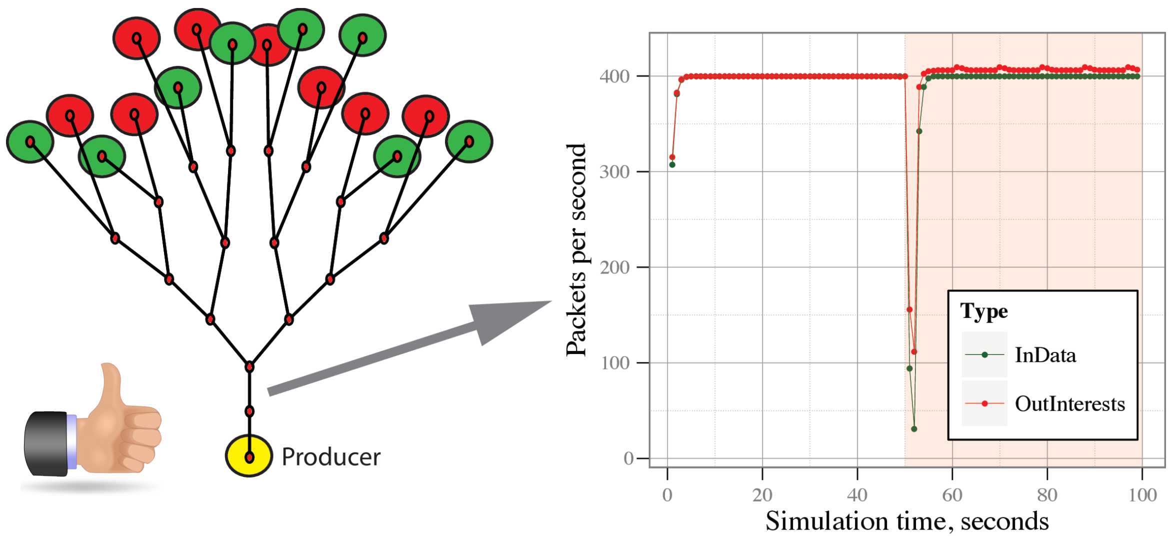


Announced limits

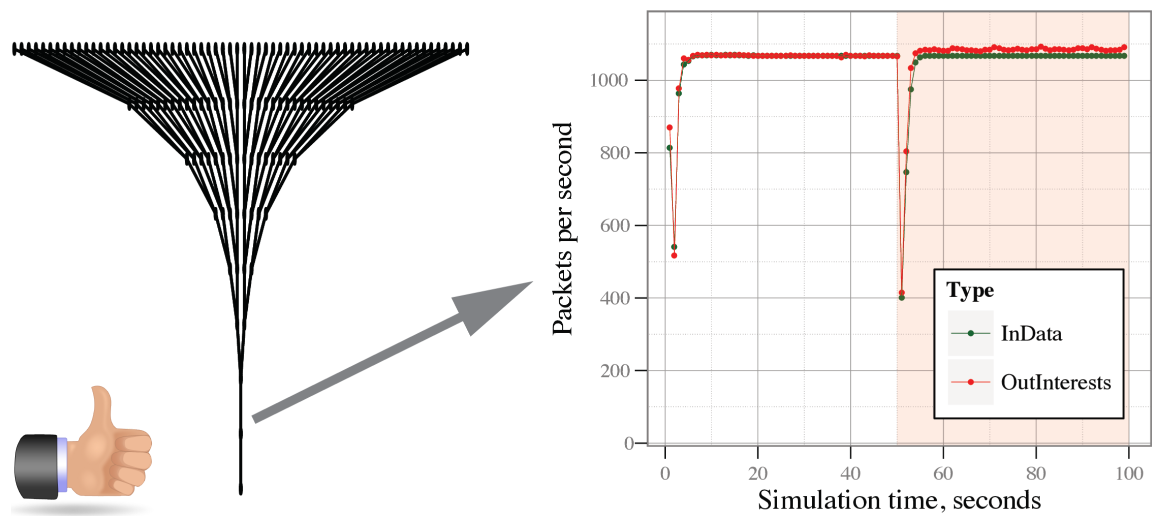


Dynamic limits results

- Does not require much parameter tweaking



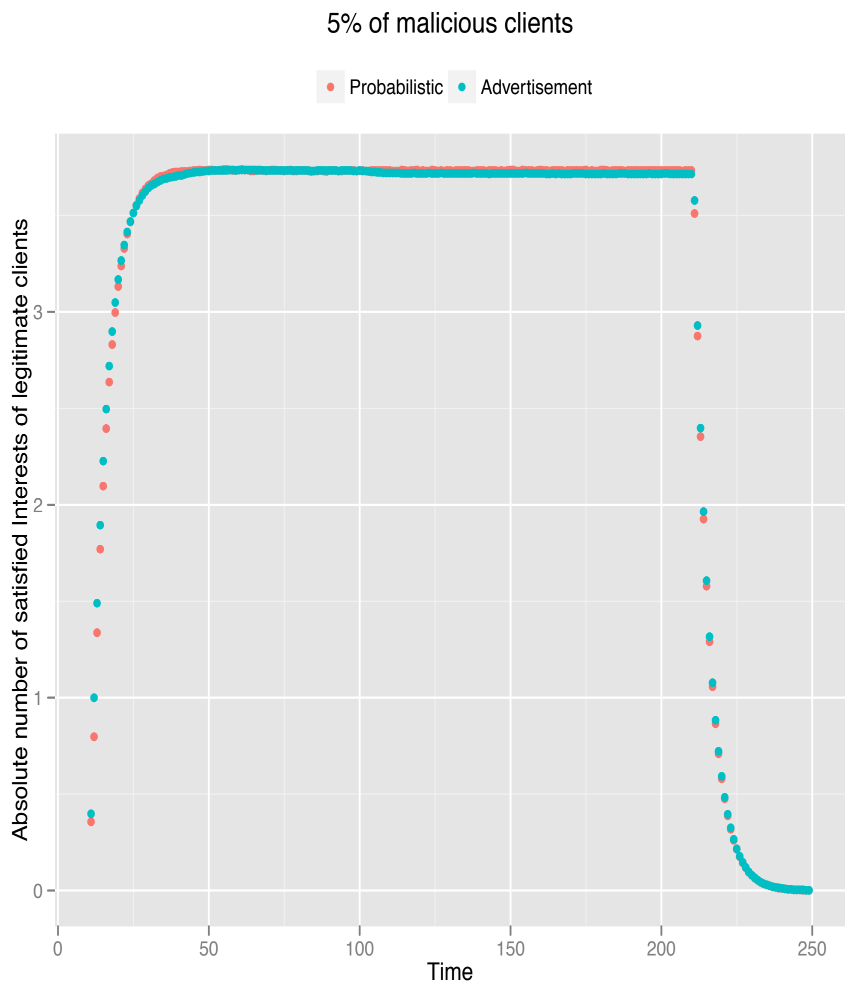
- Works with all topologies tested

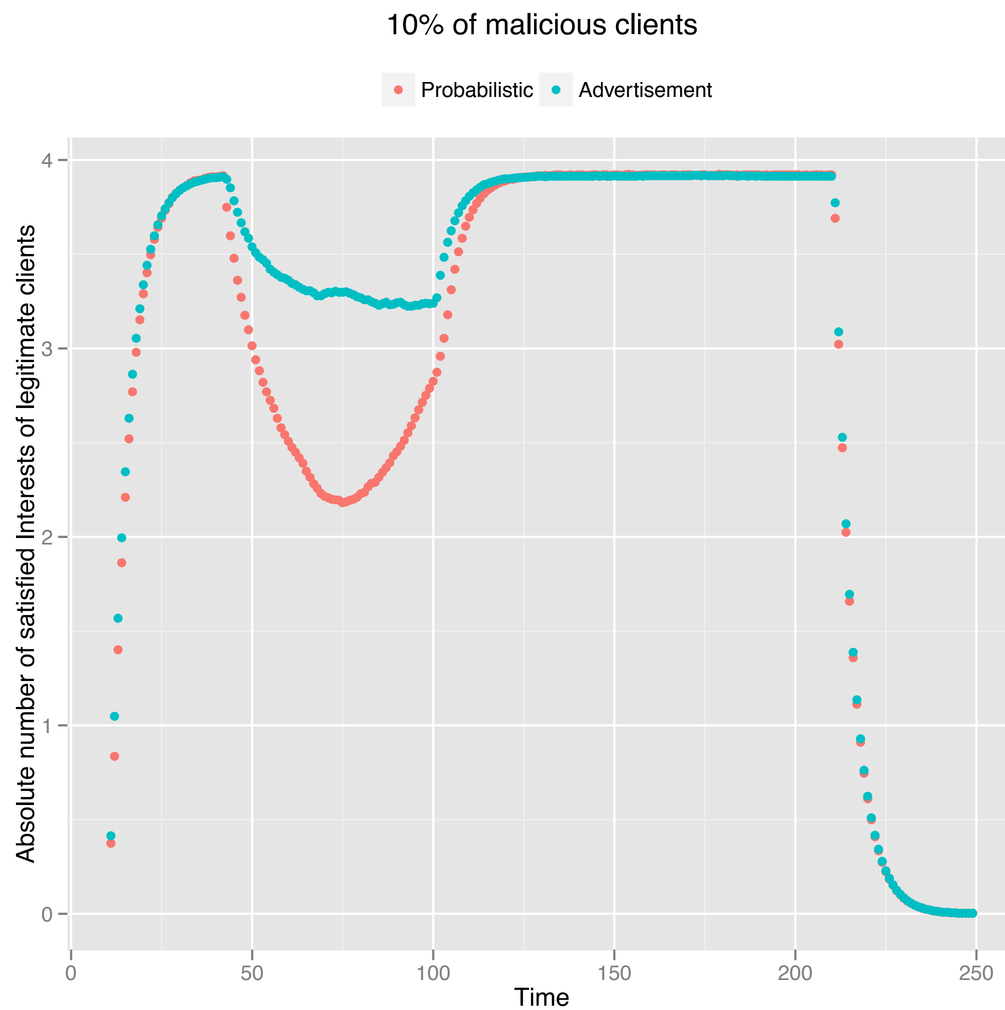


Large scale experimental setup

- Rocketfuel Sprint topology
- 7337 routers and 10 000 links
- Only adjacency = no link characteristics info
- Extract
  - 535 backbone routers
  - 3339 gateway routers
  - 3463 customer routers
- Backbone <-> Backbone links are 100Mb with 70ms delay
- Backbone <-> Gateways links are 10Mb with 20ms delay
- Gateway <-> Customer links are 1Mb with 20ms delay

Large scale results





Thanks!

- Questions?