DoS and DDoS resiliency of NDN architecture

Alexander Afanasyev (UCLA), Ilya Moiseenko (UCLA), Lixia Zhang (UCLA), Ersin Uzun (PARC), Priya Mahadevan (PARC)

Motivation

- (Distributed) Denial of Service (DoS) attacks are significant threats to the current Internet
- NDN is fundamentally different than IP
  - No push: request needed for Data transmission
  - Interest/Data flow balance
    - Interest and data follow the same path
    - immediate feedback to routers
    - simple multi-path forwarding
- Most current (D)DoS attacks on IP are not applicable to NDN.
  - no spoofing or reflector attacks
  - not as easy to target a specific host
- New NDN-specific attacks?

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
    - consumes PIT state in intervening routers as well as bandwidth
    - legitimate NDN traffic suffers…
  - Easy to implement
  - Current CCNx has no countermeasures

Current primary focus

Interest flooding attacks

- Why Interests can 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 routed towards data producer(s), thus can cause DoS at or near producer(s)
- Can such attacks be prevented?

Yes

- Unlike IP routers, NDN routers maintain rich state information that can be used to detect and react to interest flooding
  - don't accept more than can be served
  - try not to accept bad Interests

Exploring the solution space

- Simulation-based small experimentations
  - ndnSIM modular NDN simulator
    - **http://ndnsim.net**
  - different scale topologies
    - binary trees (3, 31, 127 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 25% attacker population
    - only client nodes can be malicious, routers are not compromised and fully cooperate
- Emulation-based verification of the simulator on DETER testbed
  - small-scale topology (3-node binary tree)
- Large scale simulations for promising mitigation techniques

Physical (bandwidth) limits

- Current CCNx code does not currently limit the PIT size, or the # of Pending interests for any interface
  - downstream can send more interests than physically possible to satisfy.

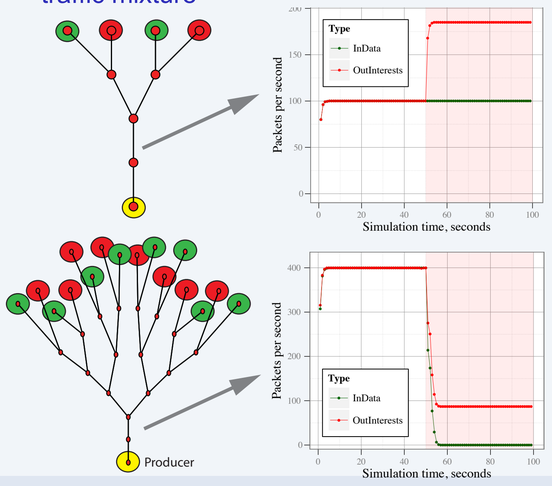
NDN architecture provides opportunities to set limits

- NDN 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:

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

The limit alone is not sufficient

- In small topologies, prevents attackers from injecting excessive # of interests
- Does not work in larger topologies
  - as the number of hops between clients and producers increases, it becomes impossible to identify good and malicious Interests in the traffic mixture

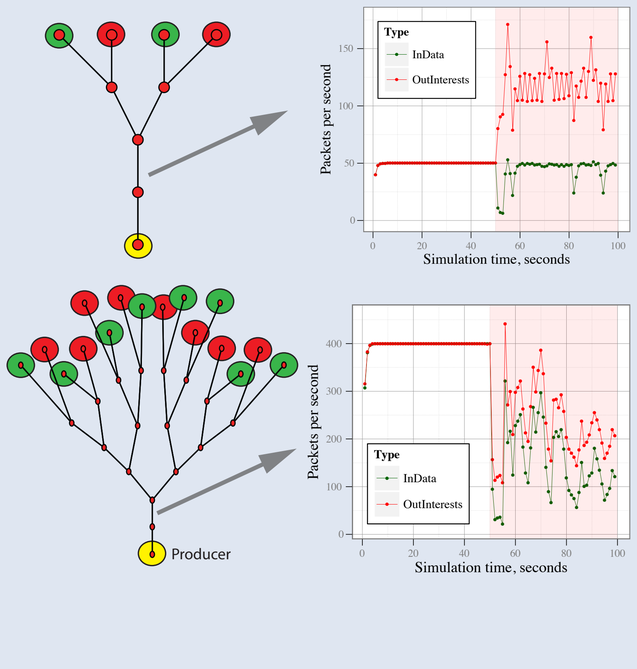


Utilizing the state information

- Theoretically, NDN routers have all the information needed to be able to differentiate good Interests from malicious ones
  - to be most effective in DoS, malicious Interests need to be insuppressible and request non-existing content
    - malicious interests ~ unsatisfiable
  - on the other hand, good Interests will likely be satisfied with a content packet
    - good interests ~ satisfiable
  - experimenters with more sophisticated attacker strategies are underway
- Keep per incoming interface, per prefix (FIB entry) interest satisfaction statistics in routers
- Use the statistics to detect and control malicious traffic

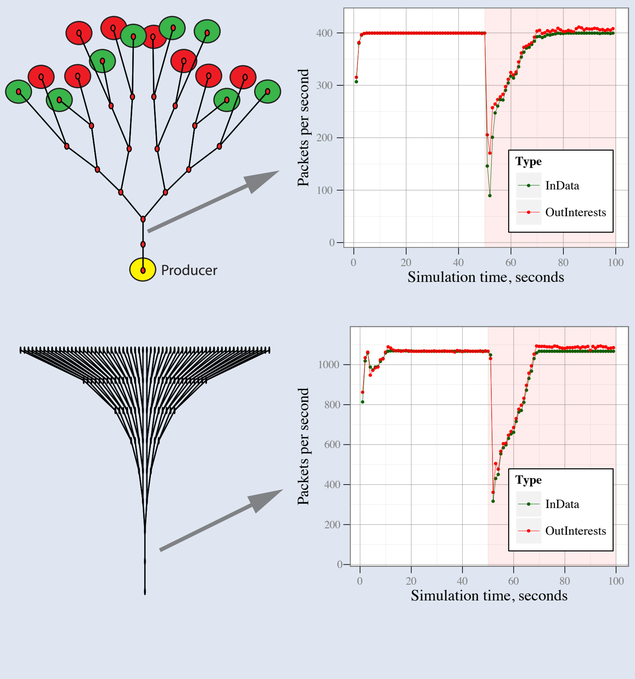
Weighted round-robin on Interest queues

- Interest processing algorithm
  - 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
- Performance: **subpar**
  - more fair sharing of resources
  - not 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 investigation)



Probabilistic Interest acceptance

- Interest processing algorithm
  - "accept" if the outgoing face is utilized under a threshold
  - otherwise, accept with probability proportional to the satisfaction ratio on this face
  - 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
- Performance: **reasonable**
  - parameter selection is important but may not be easy due to topology variances.
  - may result in link under-utilization
  - might perform better with NACKs (more accurate statistics)

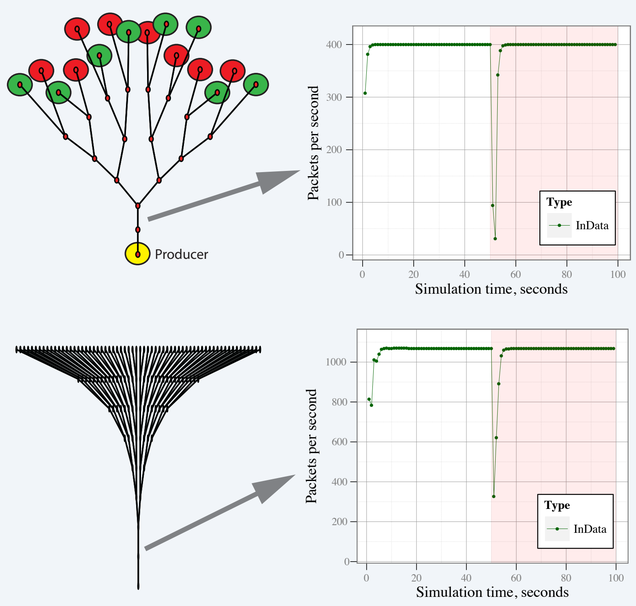


Dynamic Interest limit adjustments

- Incorporate "active" PIT management
  - Limit # of pending Interests per incoming face
    - periodically for every FIB prefix for all faces announce limit for # of pending Interests proportional to the satisfaction ratio
  - Do not over-limit
    - the sum of all announced limits is at least equal to the sum of output limits

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

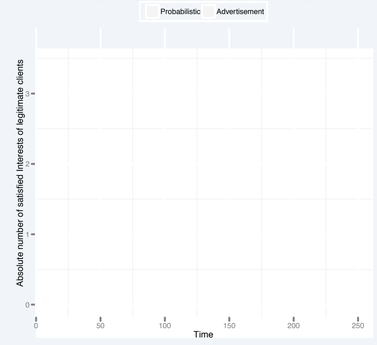
- Performance: **effective on all tested topologies**
  - Does not require much parameter tweaking
  - Response time depends on several parameters
    - limit announce period
    - statistics averaging and time-decaying factors
  - Optimization possible to reduce control overhead
    - announce only when limits change



Large scale experimental setup

- Rocketfuel Sprint topology (and others in future e.g. ATT)
- Sprint is 7337 routers and 10 000 links
- Only adjacency = no link characteristics info
- Extract
  - 535 backbone routers
  - 3339 gateway routers
  - 3463 customer routers
- Realistic topology approximation
  - **Backbone** <-> **Backbone** links are 100Mb with 70ms delay
  - **Backbone** <-> **Gateways** links are 10Mb with 20ms delay
  - **Gateway** <-> **Customer** links are 1Mb with 20ms delay

5% of malicious clients



10% of malicious clients