UCI University of California, Irvine

Closing the Floodgates with Stateless Content-Centric Networking

Cesar Ghali, **Gene Tsudik,** Christopher Wood
UC Irvine
{cghali,**gtsudik**,woodc1}@**uci.edu**

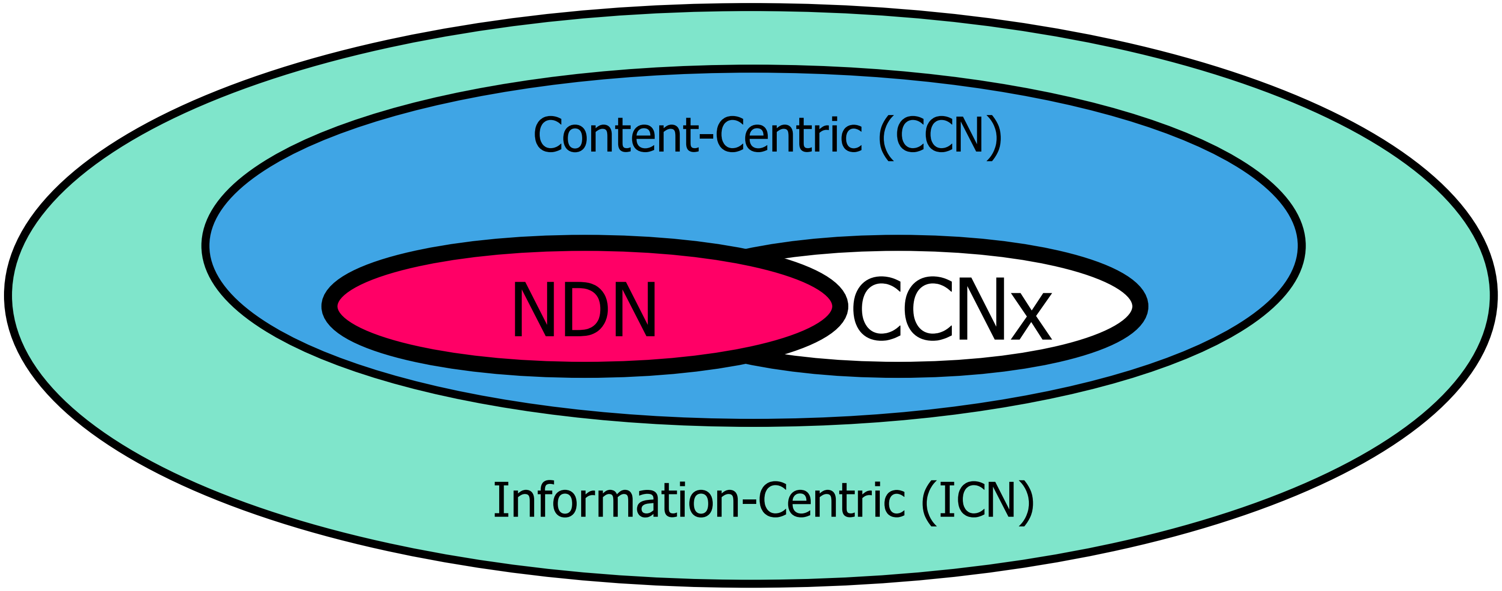
Ersin Uzun
PARC
ersin.uzun@parc.com

IEEE ICCCN 2017

Outline

- CCN Overview
- Stateful Forwarding Plane
- Stateless CCN
- Experimental Assessment
- Looking Ahead

ICN, CCN, etc.



- CCNx, NDN = very close relatives, branches of same tree
- Unification efforts underway

CCN Overview: Named Data

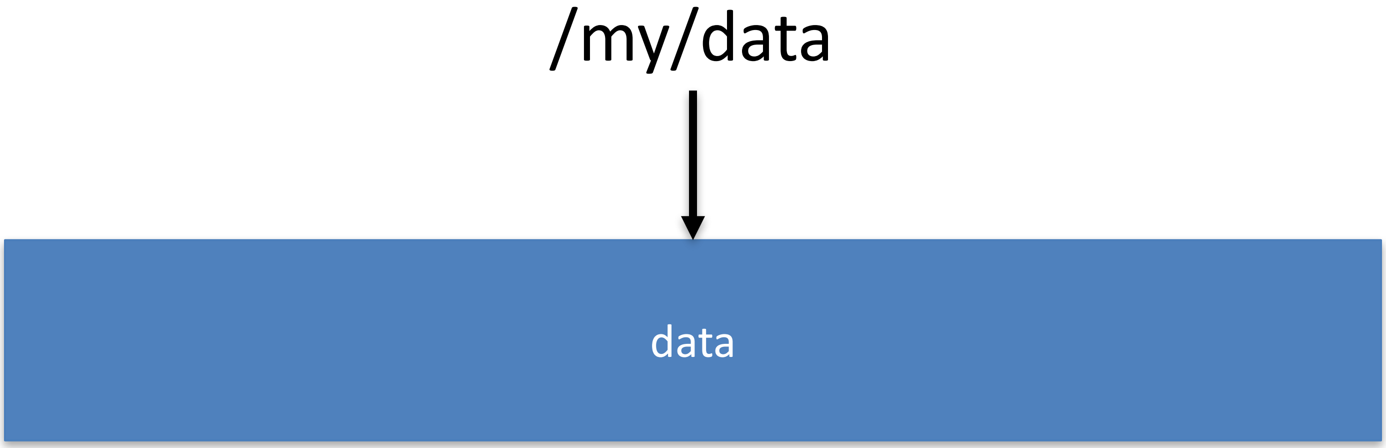
- Roles: Producers, Consumers, Routers
- Information=data=content treated as first-class object
- No explicit host or interface identifiers

data

Key elements:

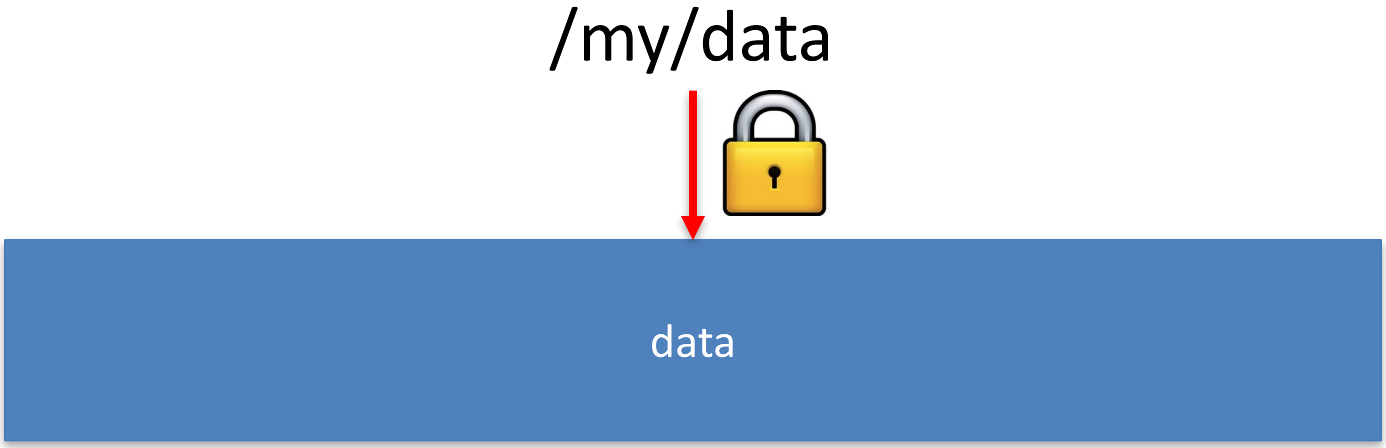
- Pending Interest Table (PIT)
- Cache = Content Store (CS)
- Cache lookups
- PIT lookups
- Interest collapsing

CCN Overview: Named Data



- Data is explicitly named
- Name can be is arbitrarily long, segments separated by “/”

CCN Overview: Named Data

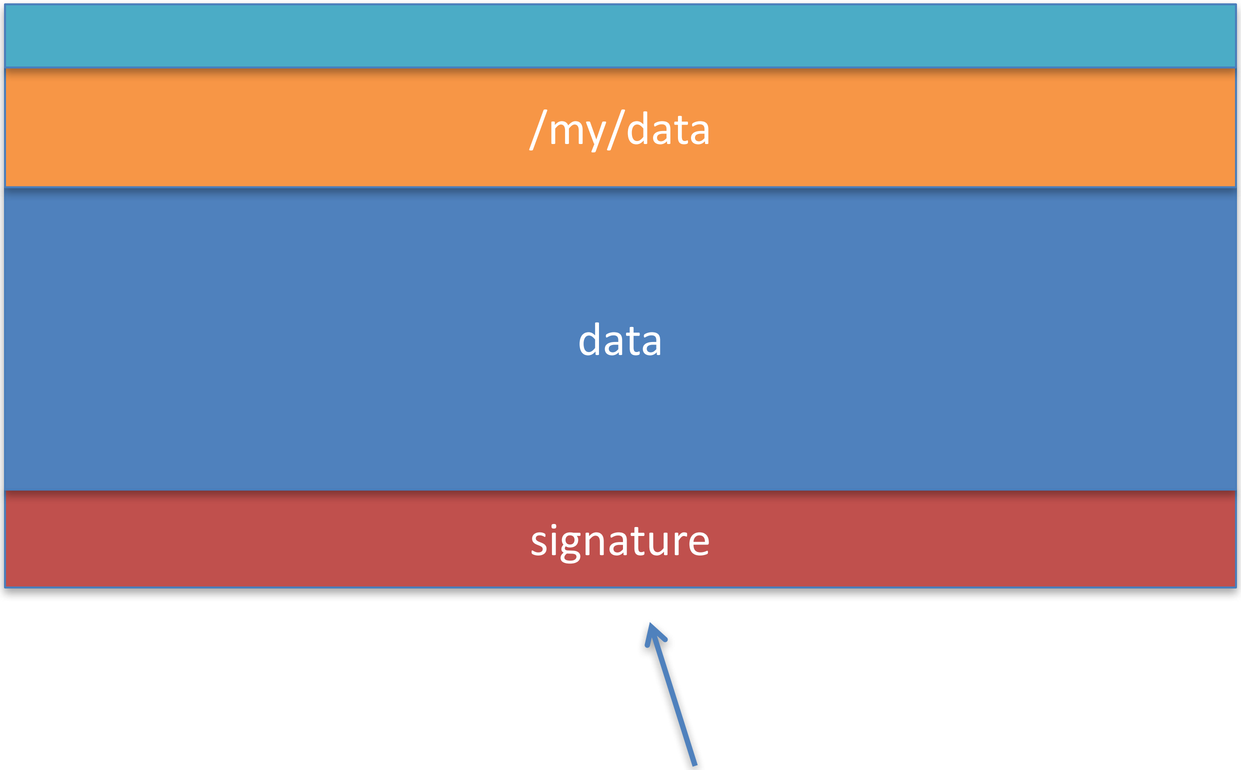


/my/data

data

- Signature: generated by producer; binds name to content via producer’s public key
- Public key: contained in a PKC
- PKC: special type of content; binds name prefix(es) to public key

Content Packet



/my/data

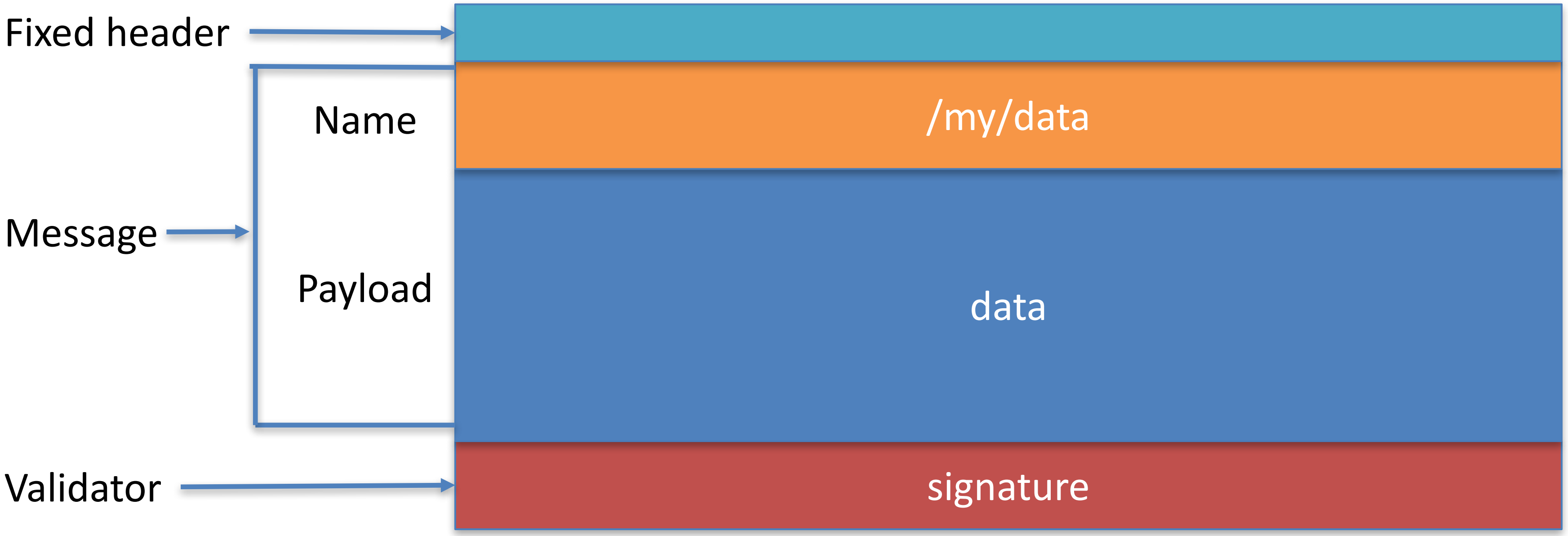
data

signature

refers to producer's public key

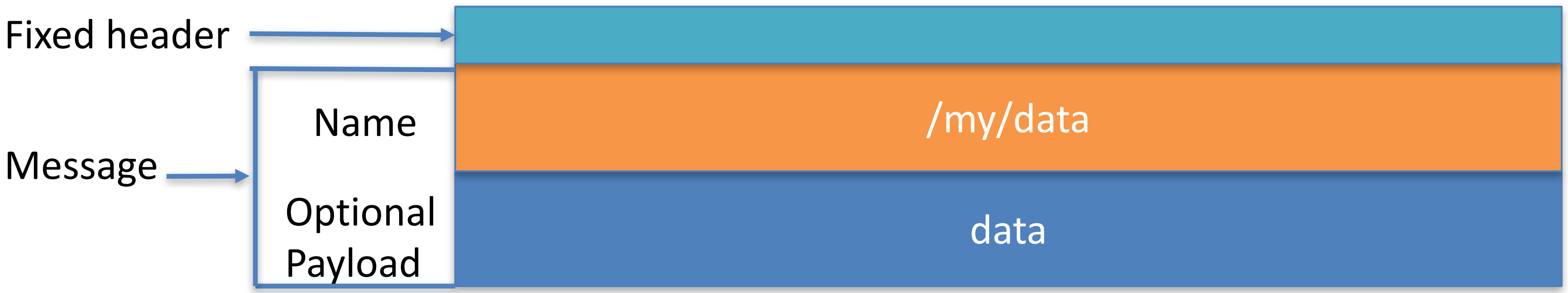
Content Packet

- **No destination address**
- **No source address**

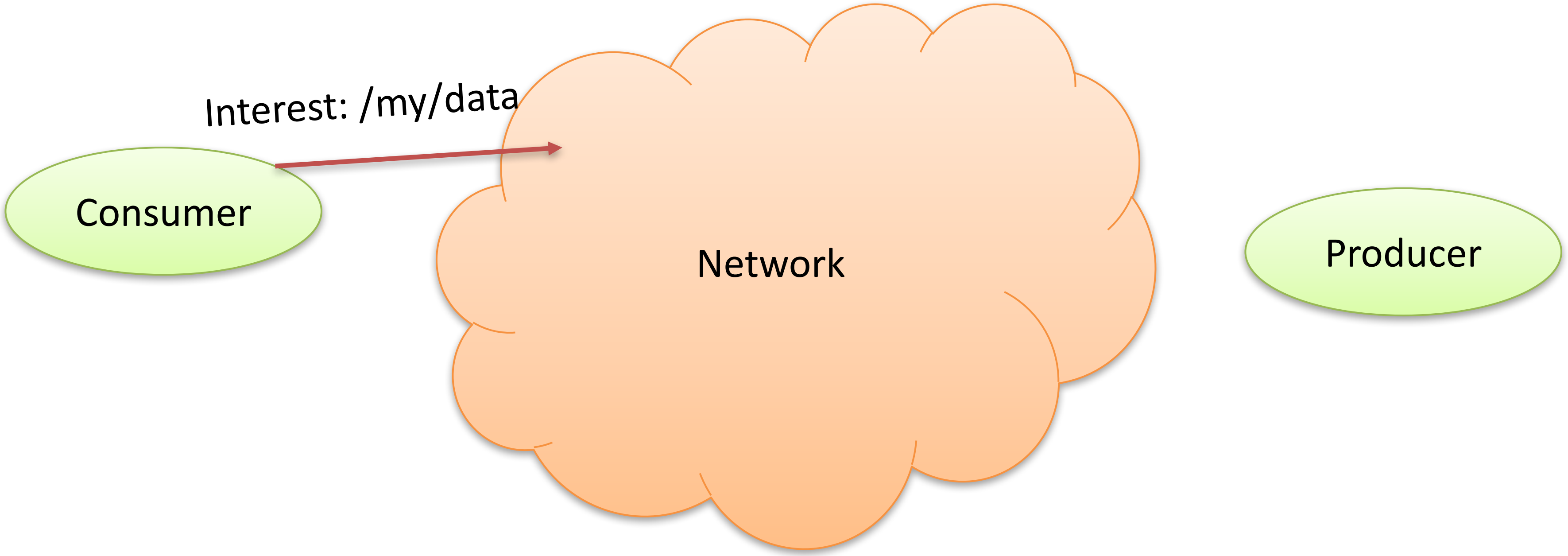


Interest Packet

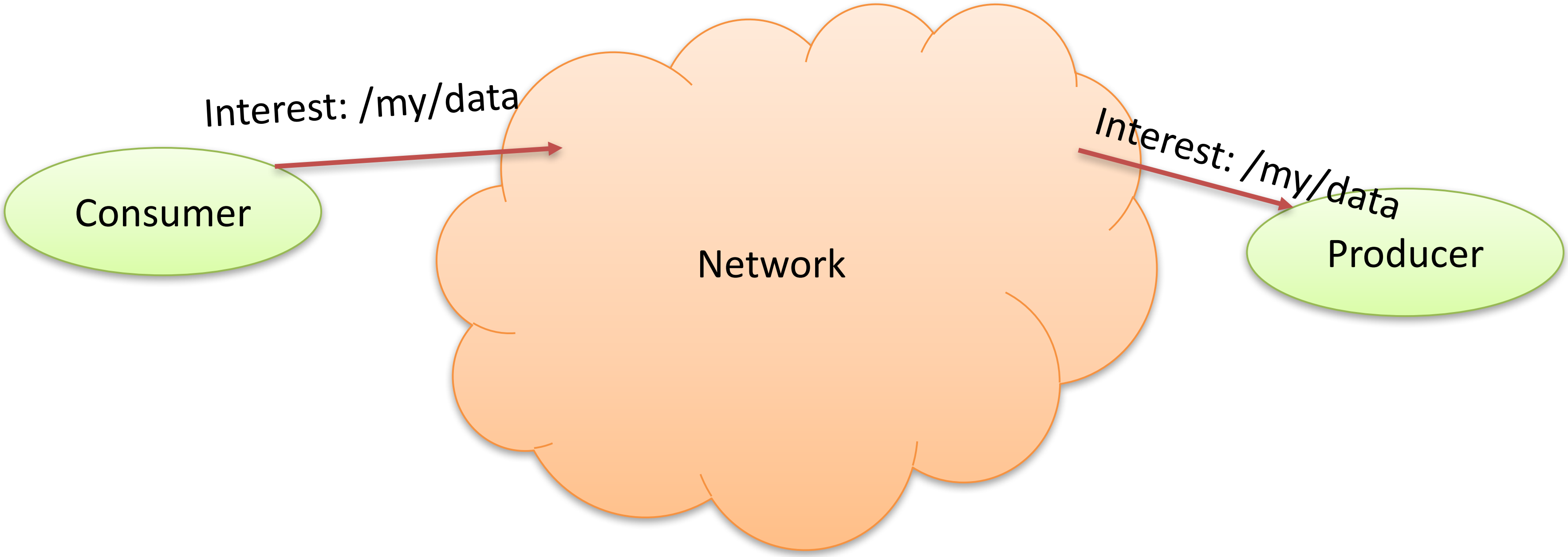
- **No destination address**
- **No source address**
- **No signature**



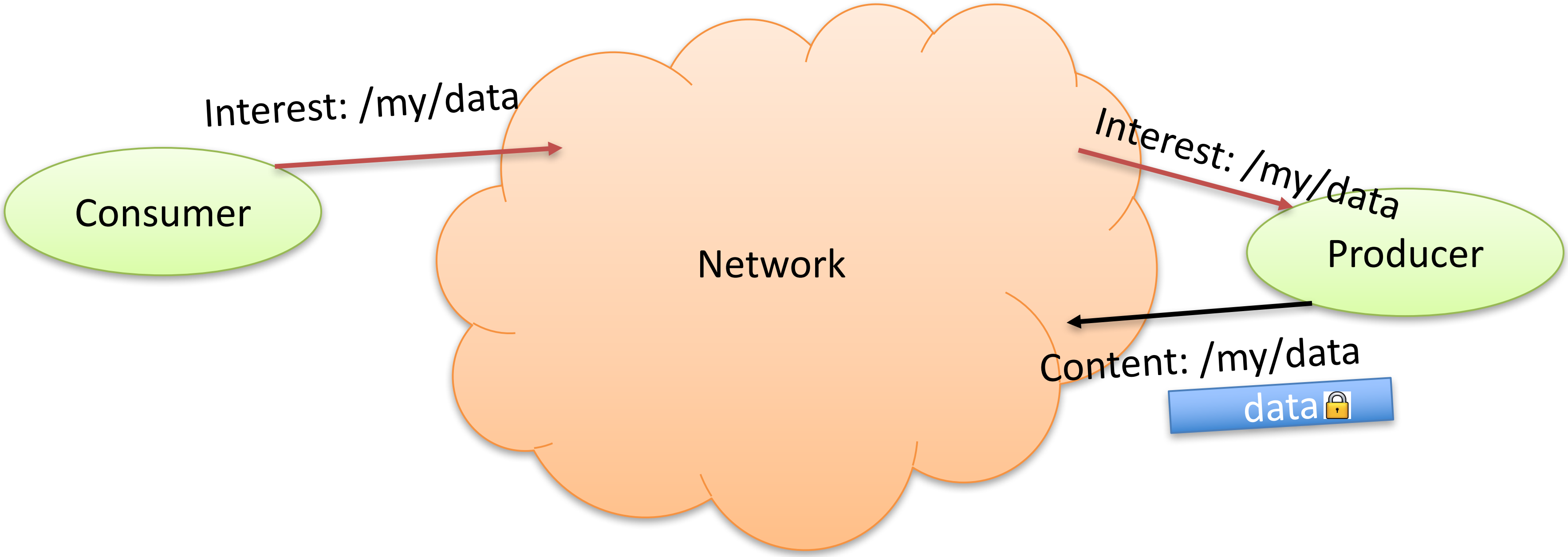
CCN Overview



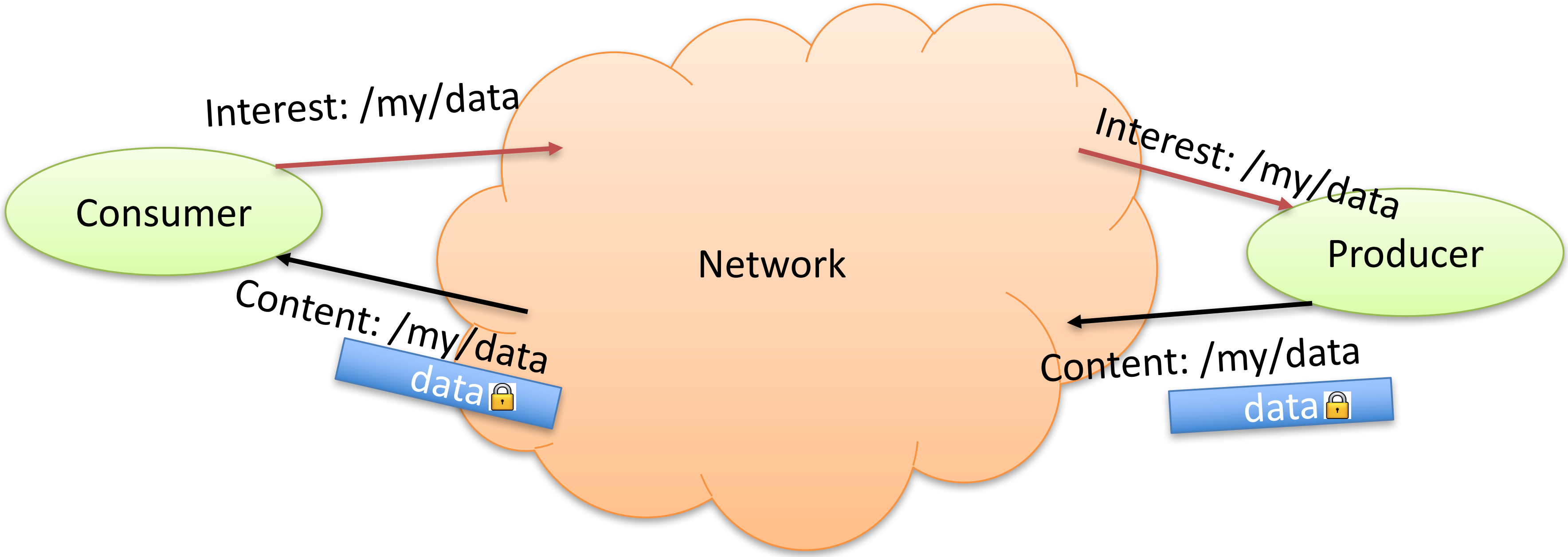
CCN Overview



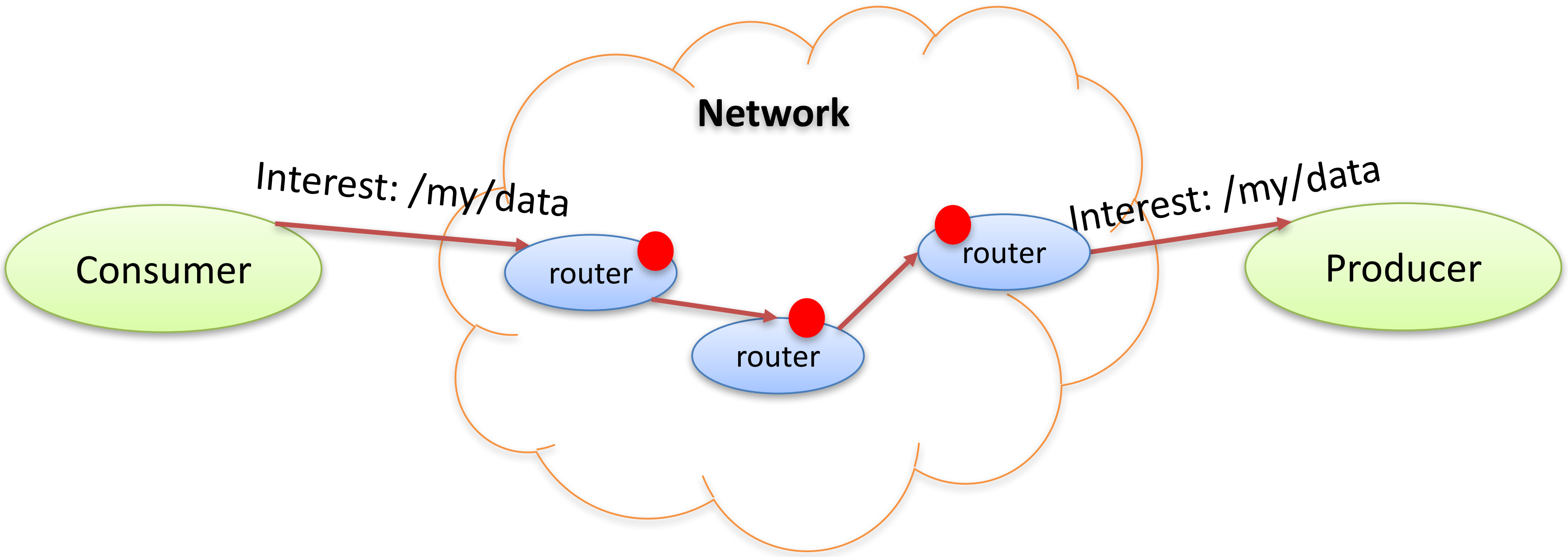
CCN Overview



CCN Overview

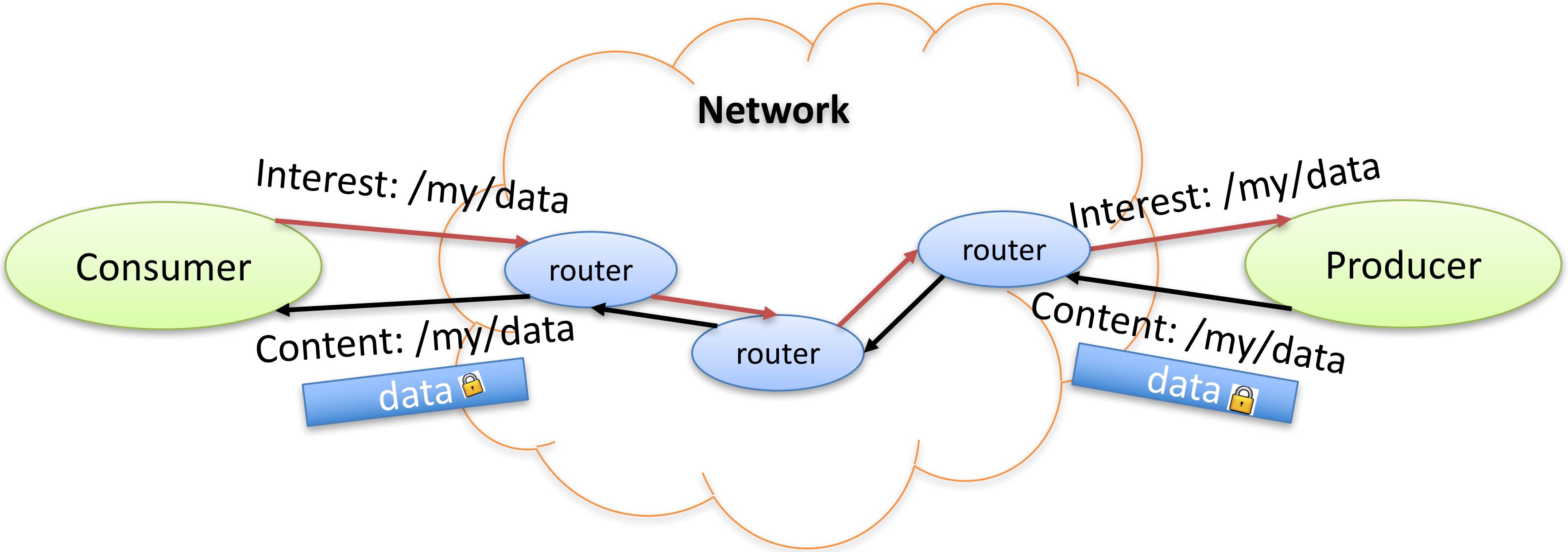


CCN Overview

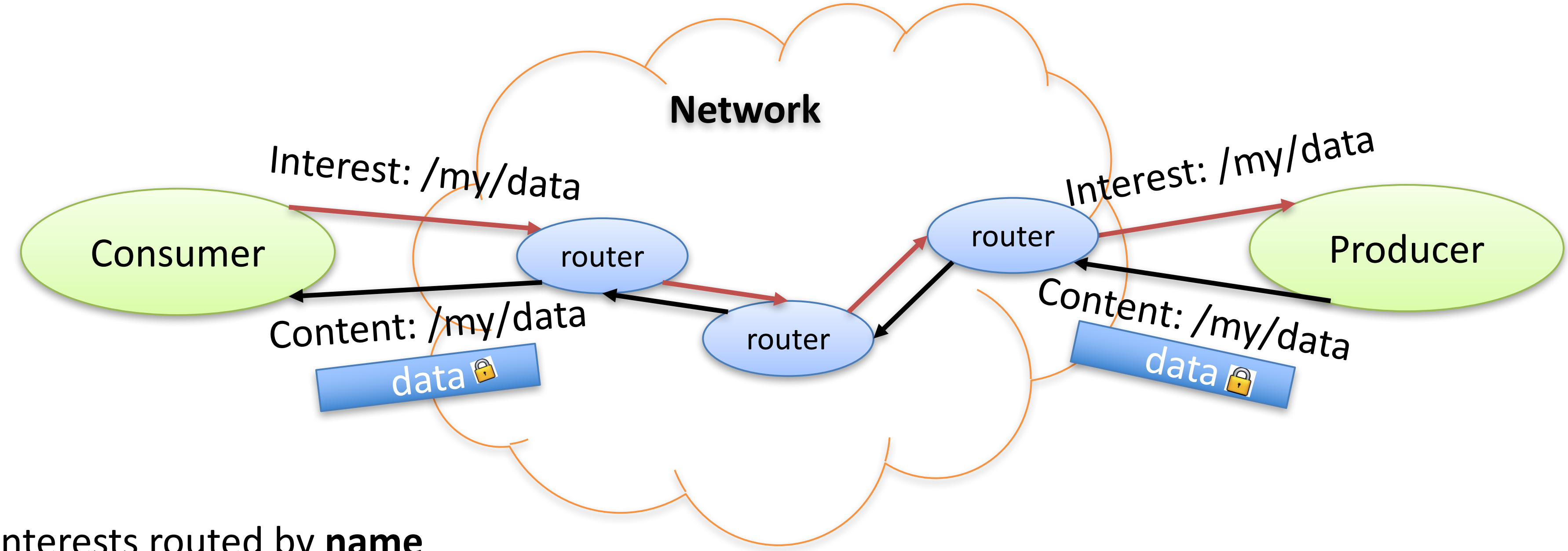


● Per-interest state = “pebble” = PIT entry

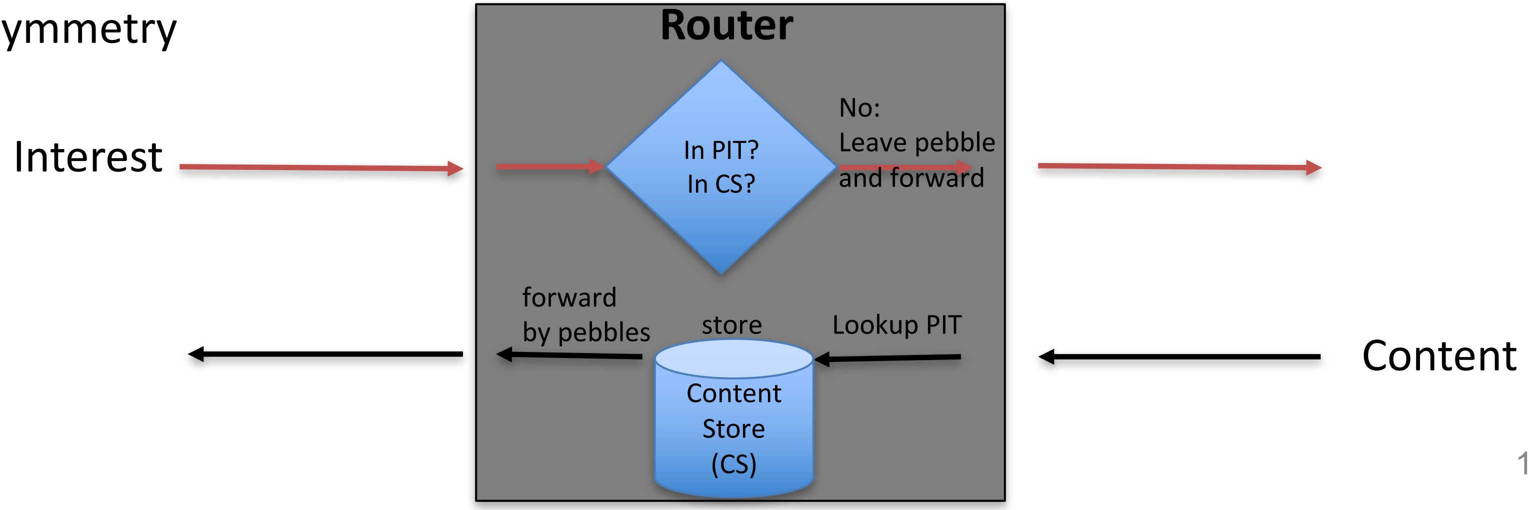
CCN Overview



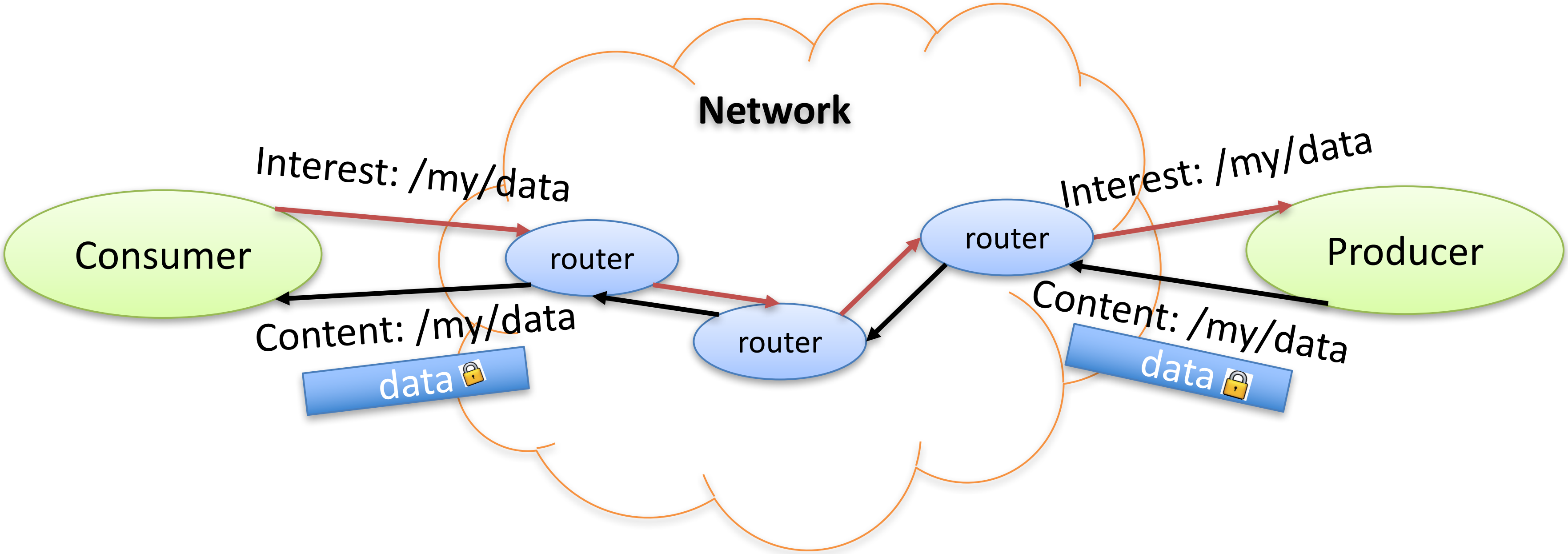
CCN Overview

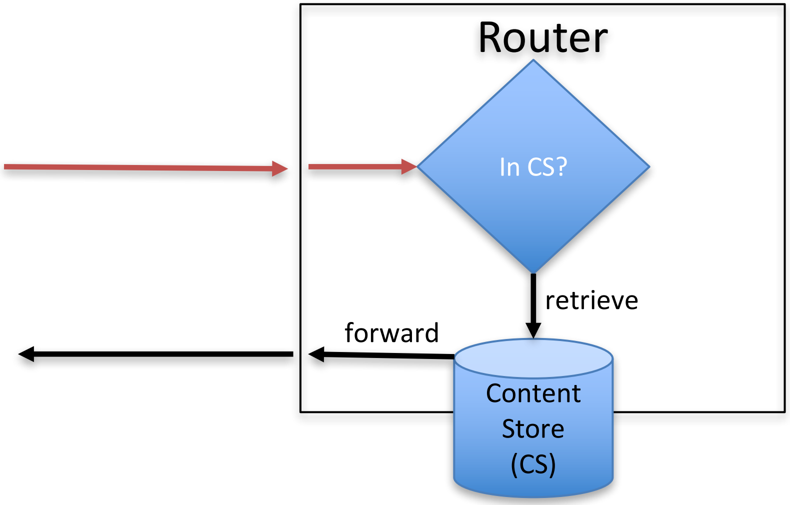


- Interests routed by **name**
- Content routed by pebbles: RPF
    - Path symmetry



CCN Overview





Interests are routed by **name**

Content is routed pebbles

Questions

Does “Content Centricity” imply the need for:

- Caches/CSs

and/or

- PITs = stateful forwarding

Is stateful forwarding beneficial?

- Always?
- Sometimes?
- Ever?

Stateful Forwarding Plane: Considerations

- Reverse-Path Forwarding
  - Is it a performance win?
- Flow and Congestion Control
  - Where does (or should) it take place
- Security
  - What attacks prevented and what prompted?
- Interest Collapsing
  - Does it (would it) occur often?

Reverse-Path Forwarding

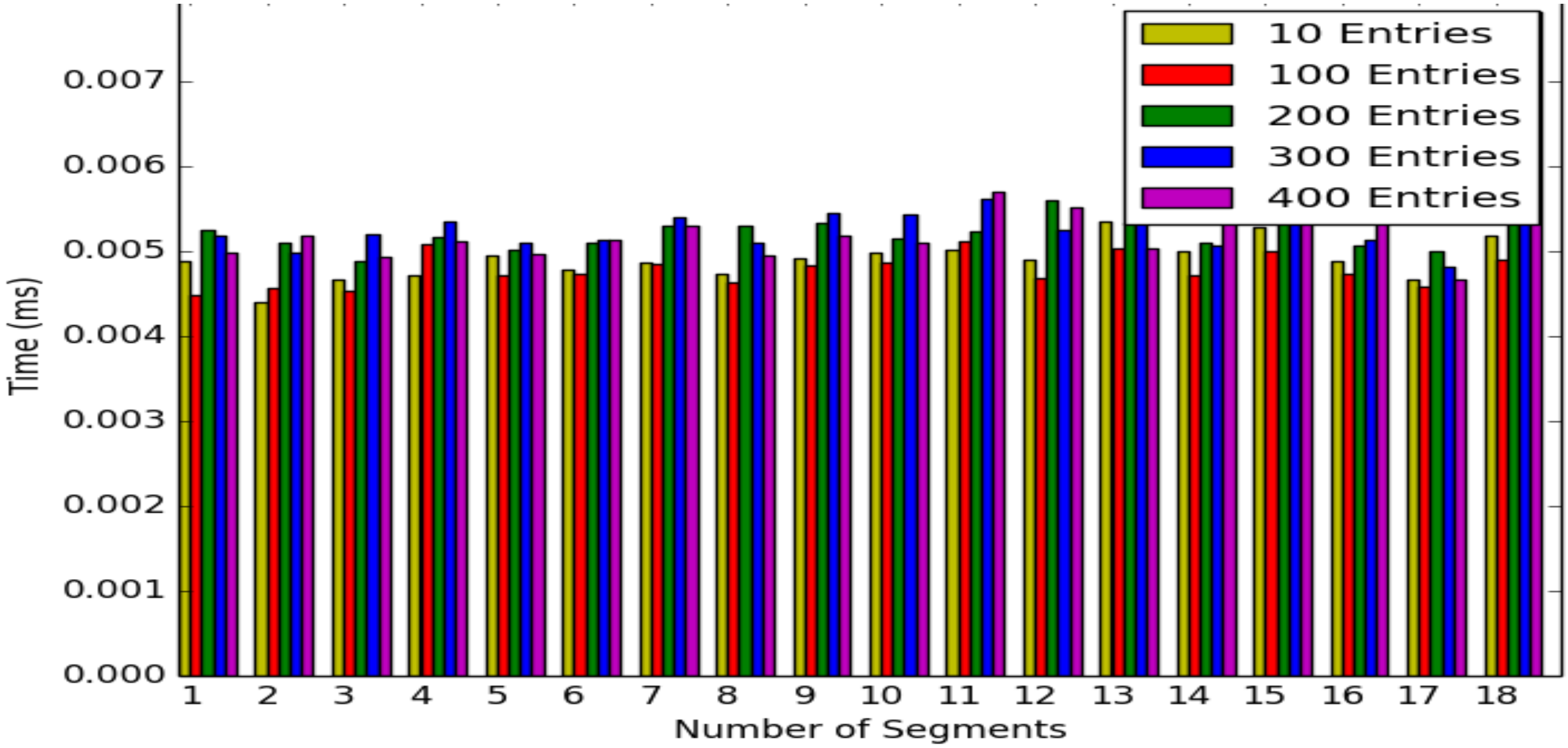
- Some applications are distributive, uni-directional
- Others are bidirectional
  - E.g., conferencing, remote login, p2p
  - FIB entries needed for both end-points
- FIB may subsume PIT state
  - E.g., for bidirectional communication
- Path symmetry not a given
  - Lower in core than at edges*

BTW:

- Total Cost: 2 PIT lookups + FIB lookup

- W. John et al., “Estimating routing symmetry on single links by passive flow measurements,” IWCMC, 2010.

PIT lookups are not free



- PIT lookup procedure for PARC Metis forwarder
- Random set of URIs generated from Cisco URI data set
- Added and removed PIT entries at varying rates to match desired steady state
- Average number of entries varies [10; 100; 200; 300; 400]
- HW/SW: 2.8 GHz Intel Core i7 CPU, 16GB of 1600 MHz DDR3 RAM, Ubuntu 14.04,
- **Removing PIT saves on avg about 4.5 microsecs**
- Weird: # of name segments doesn’t seem to matter much…

Flow and Congestion Control

- Increasingly moving towards the edge
  - Receiver-based alternatives
  - Per-link congestion info needn’t be stored in PIT

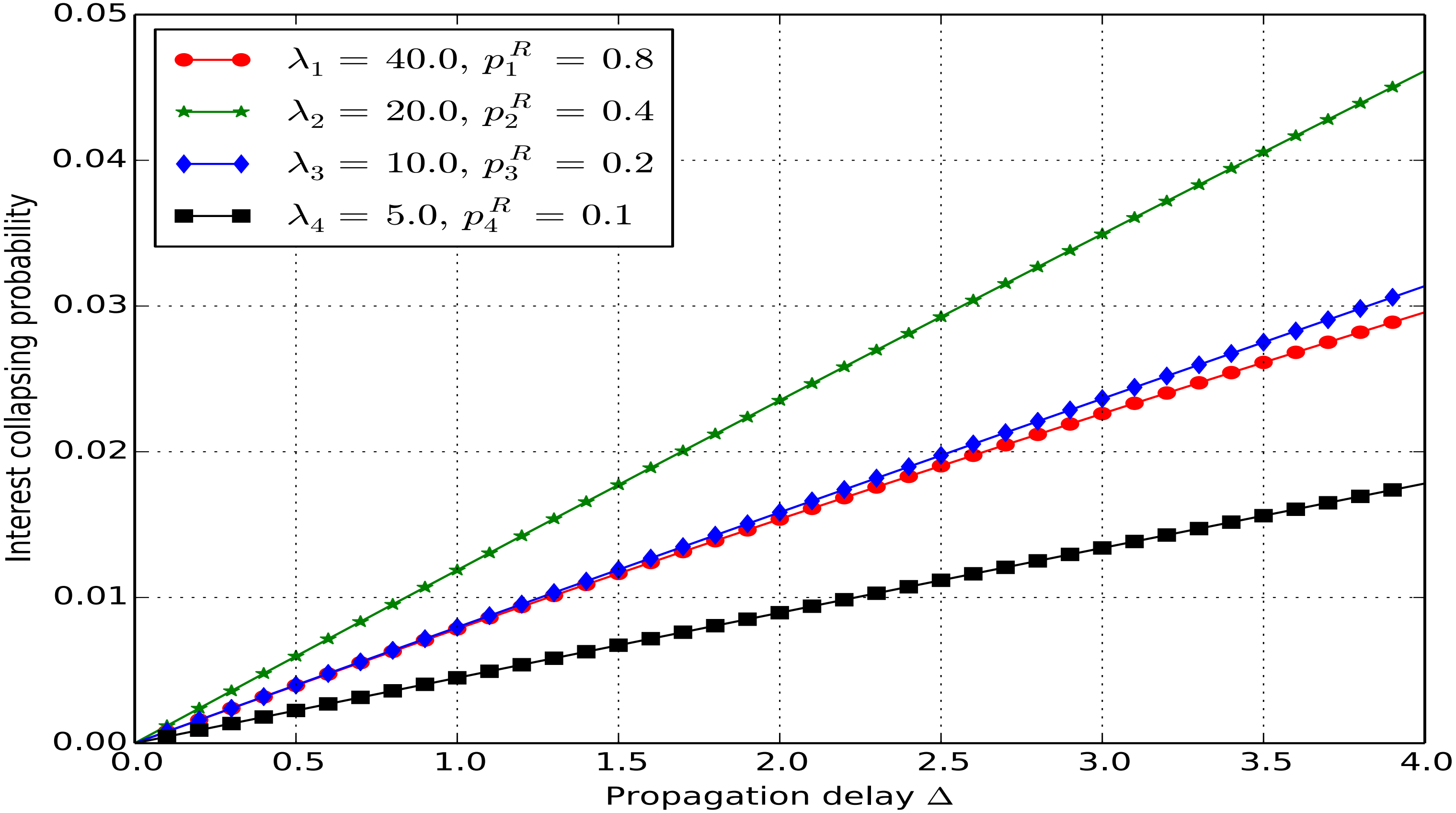
- Potentially substantial size disparity between interest and content packets

- G. Carofiglio et al., “Multipath congestion control in content-centric networks,” INFOCOM ICN WORKSHOP, 2013.
- S. Braun et al., “An empirical study of receiver-based aimd flow-control strategies for CCN,” ICCCN, 2013.
- L. Saino et al., “CCTCP: A scalable receiver-driven congestion control protocol for content centric networking,” ICC, 2013.

Security

- Each interest leaves state in a router
  - New PIT entry or new interface ID for existing one
- Reflection attacks are mitigated
  - Consumer can’t be hosed/DoS-d
- Any router state **can and will** be abused
  - Interest Flooding Attacks
    - PIT (and producers)
  - Content Poisoning Attacks
    - CS (and consumers)

Interest Collapsing



Interest collapsing probability for four content classes (<5% at best)

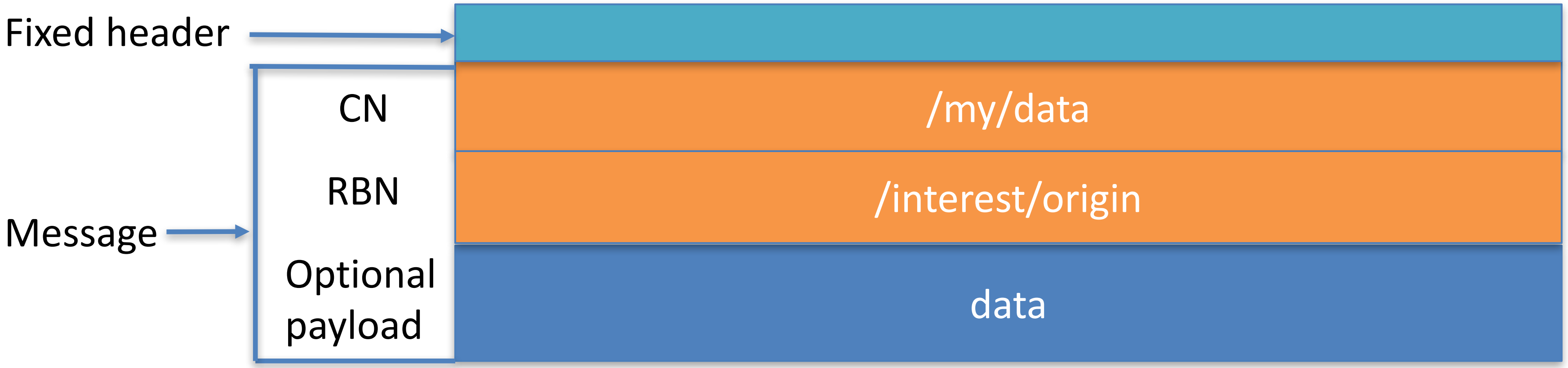
- Cache enabled
- Arrival rates and cache hit rates differ for different classes

Stateless Forwarding Design

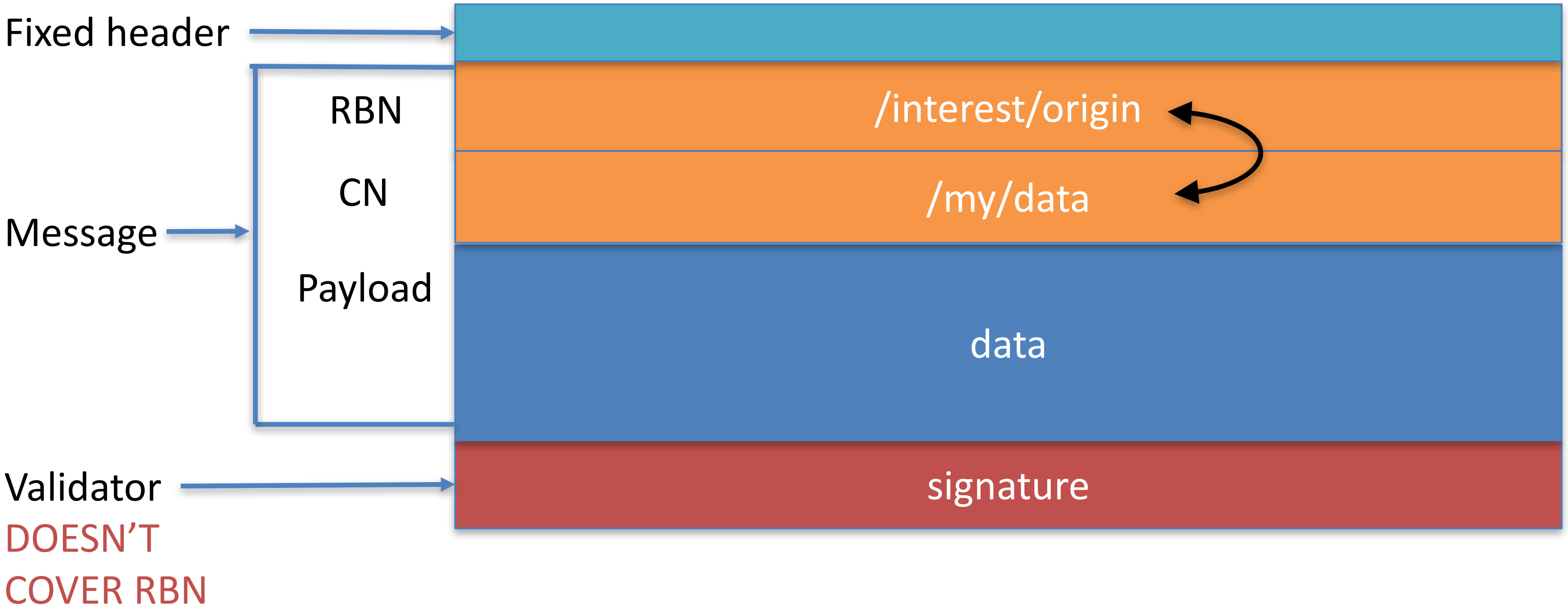
- Interests still carry **content names** (CNs)
- PIT state replaced with

  Routable Backwards Names (RBNs)
- Each packet carries both
  - Interest: (CN, RBN)
  - Content: (RBN, CN) → no need to re-sign, can still cache
- No PIT, CS optional (as it is now)
  - Interest flooding mitigated (mostly)
- Consumer becomes addressable
  - But, consumer DoS is back! ☺
  - BTW: consumer not required to have PK

Interest Format

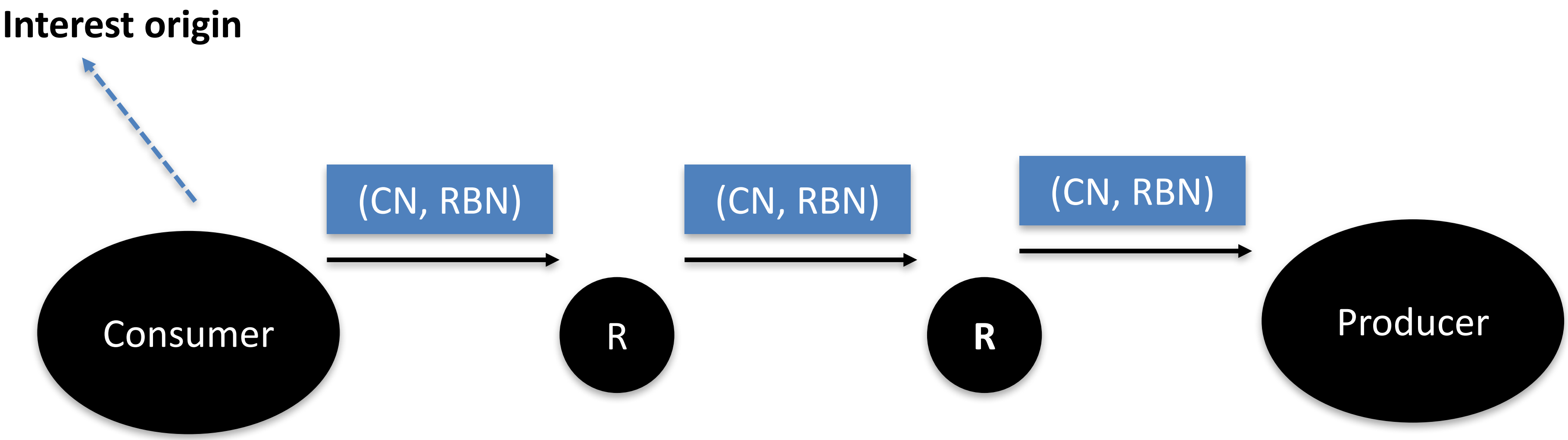


Content Format

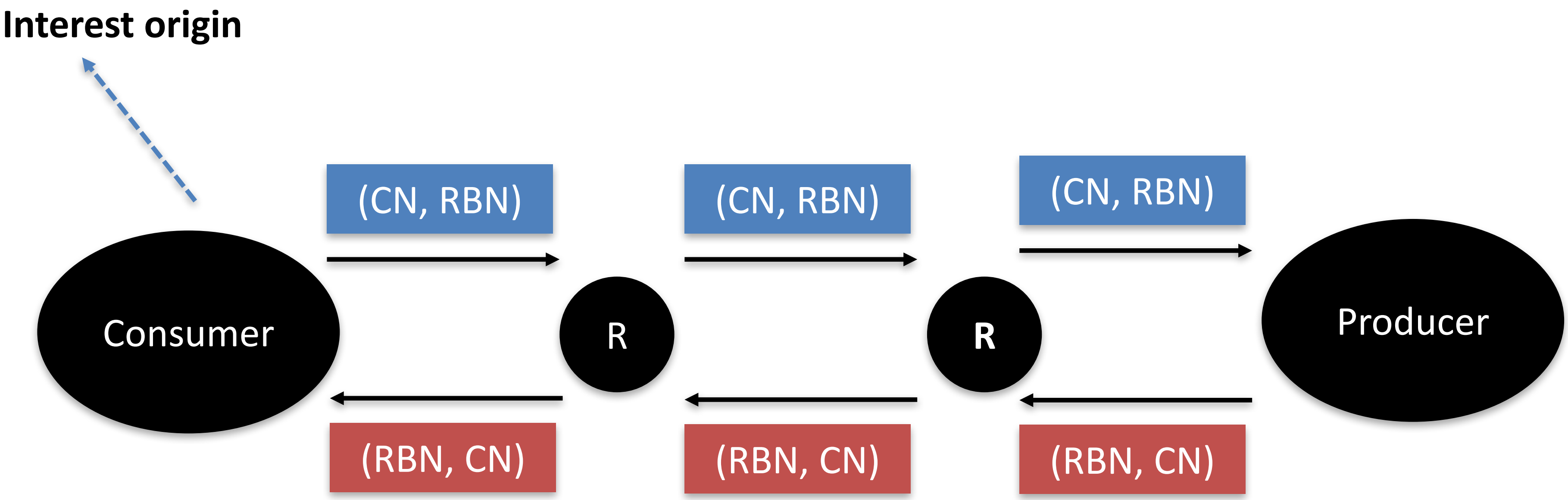


DOESN'T
COVER RBN

Stateless Data Plane

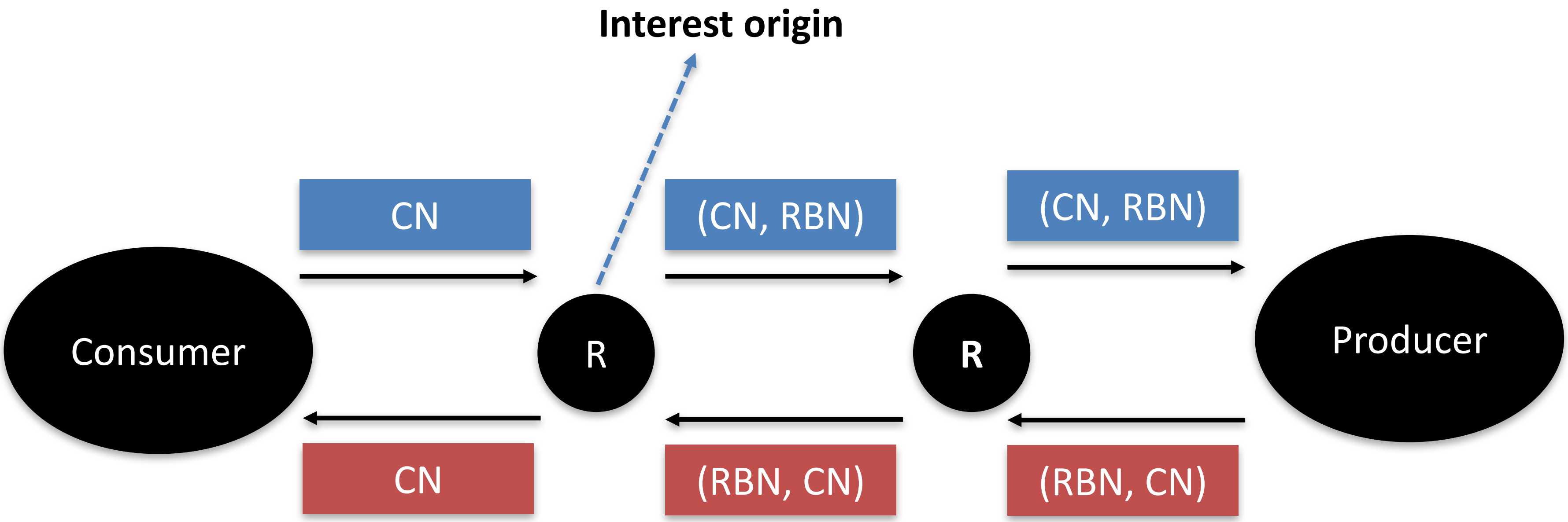


Stateless Data Plane



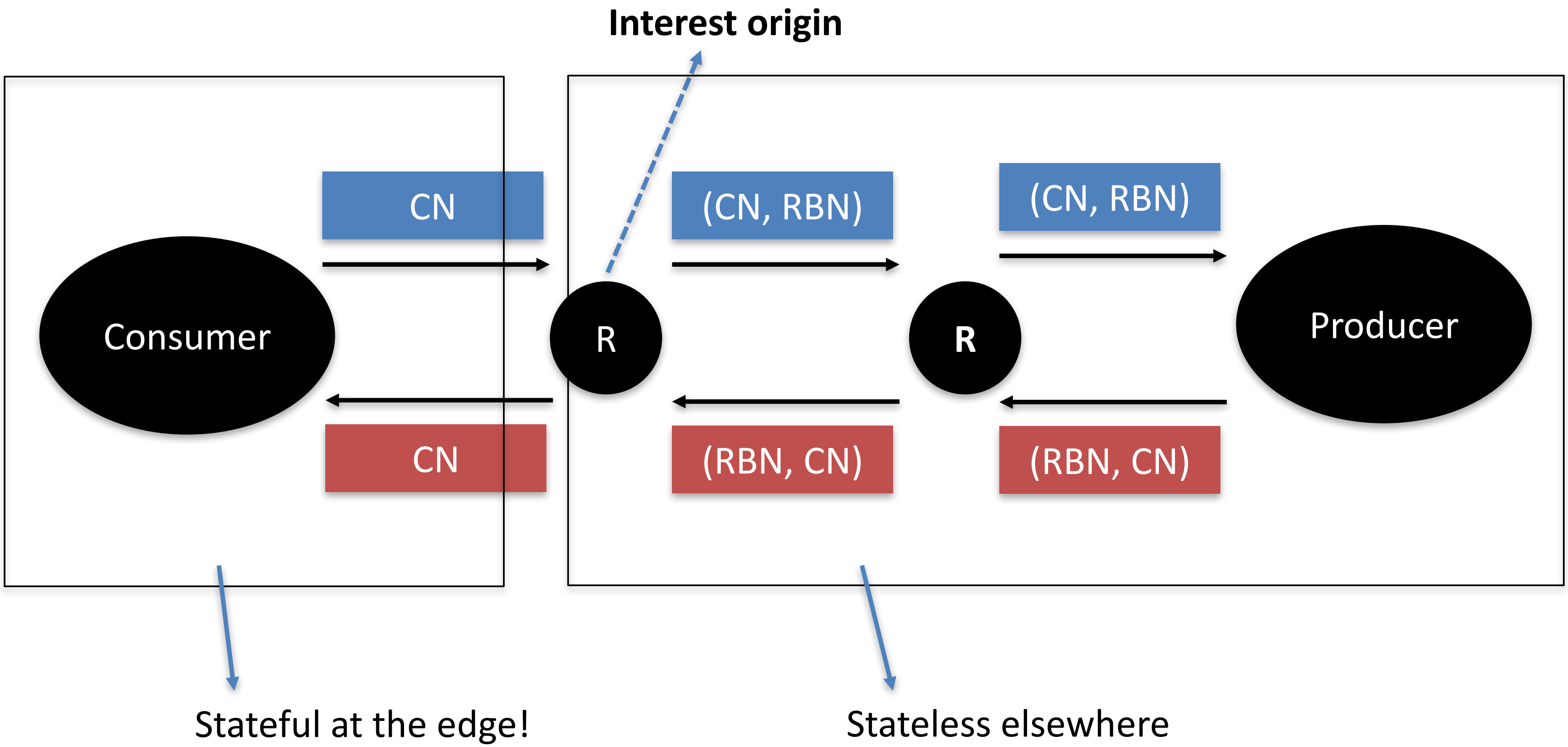
NOTE: interest path might differ from content path

Hybrid Data Plane

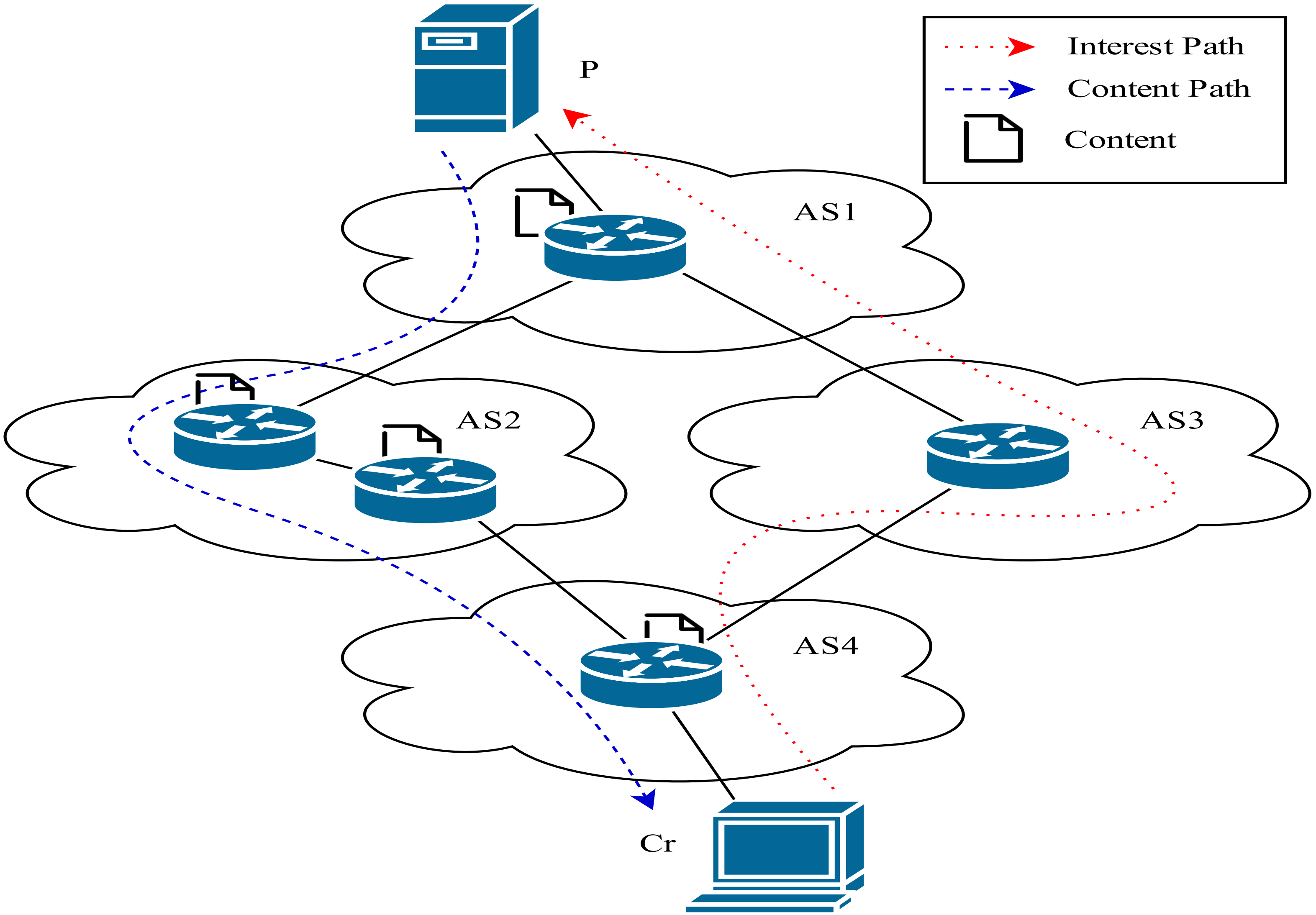


NOTE: interest path might differ from content path

Hybrid Data Plane



Hybrid: Asymmetry in the middle



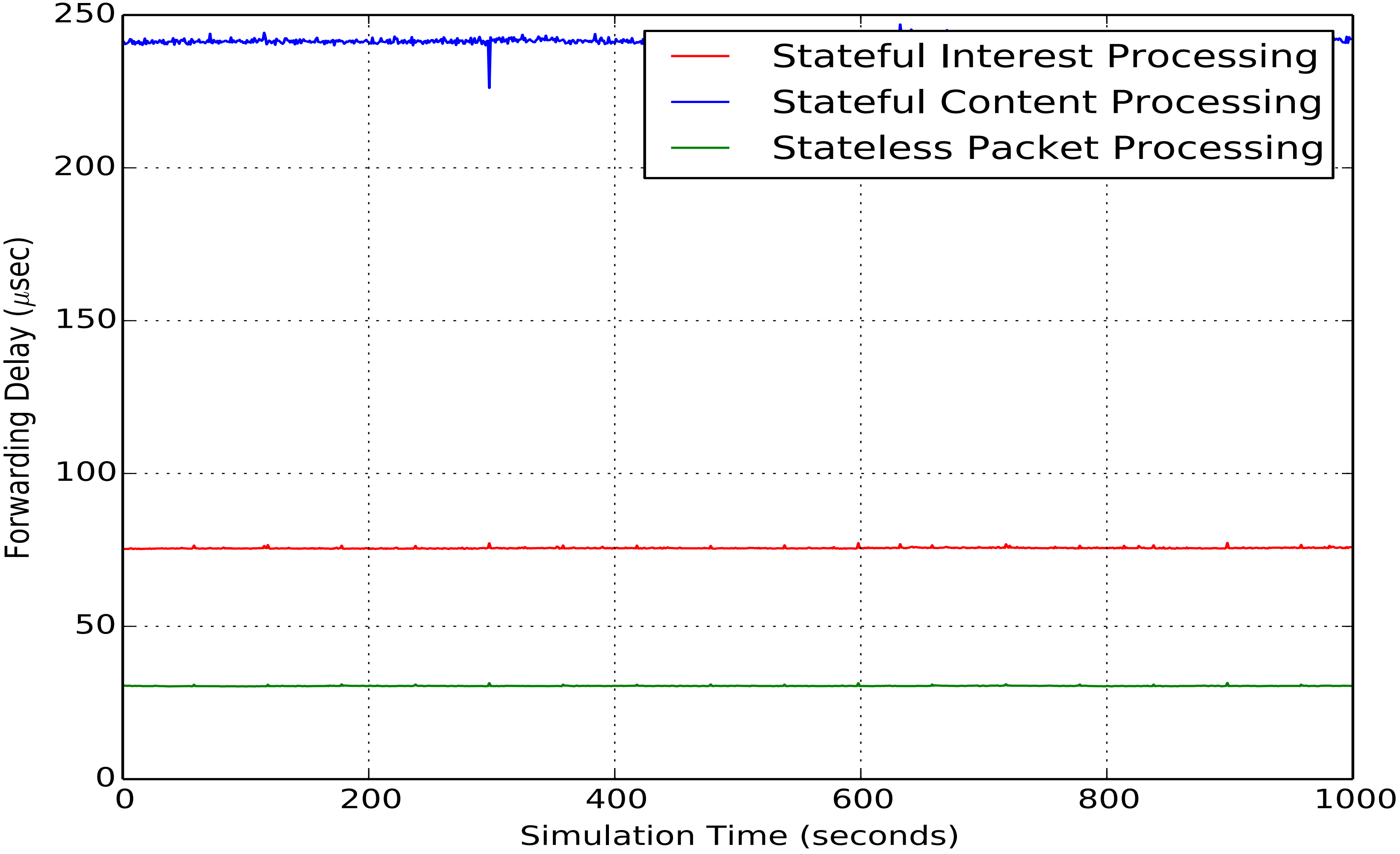
Alternative: Separate but Equal

- Consumer (or router?) selects stateful vs stateful operation per application or even per interest

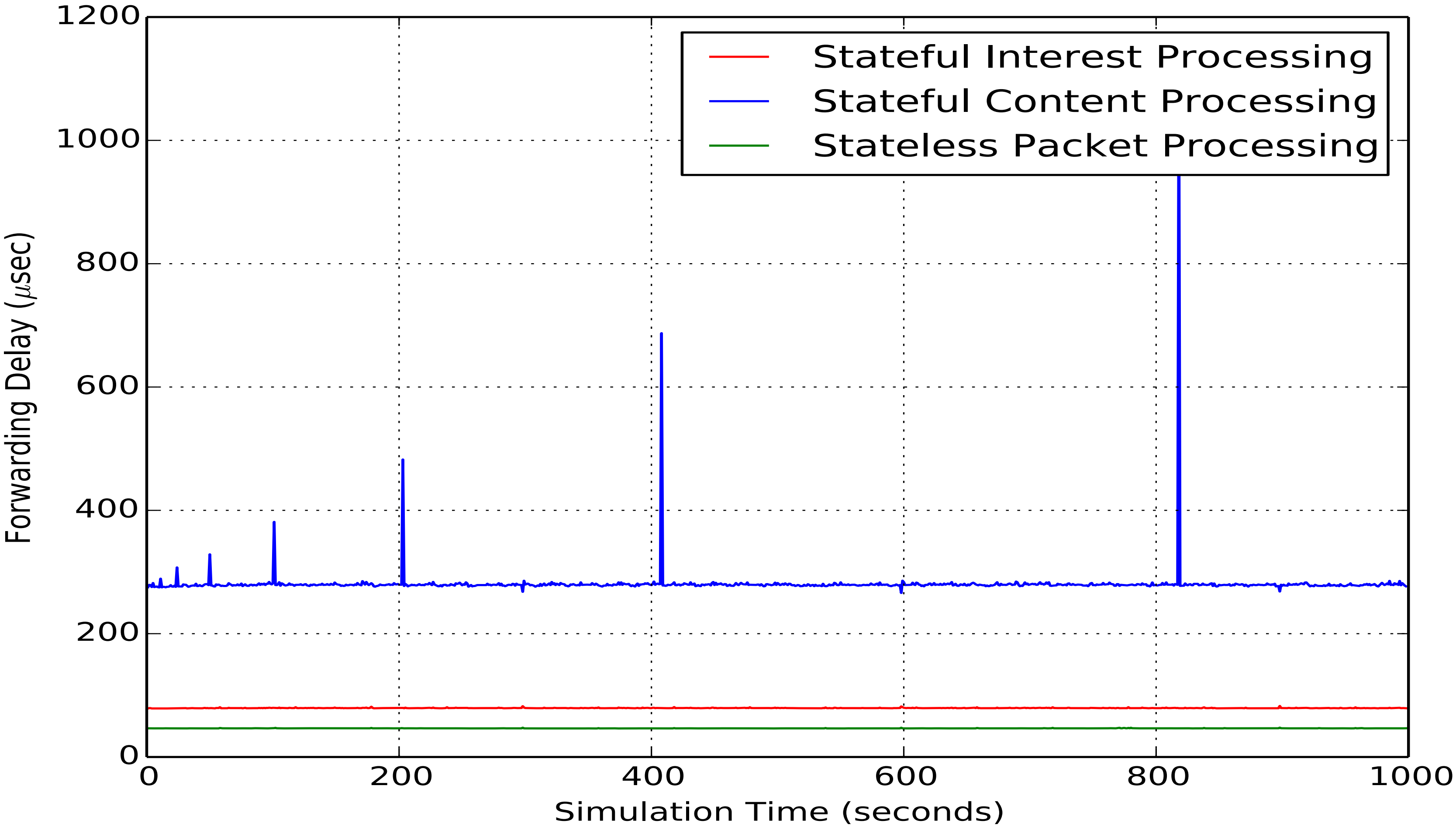
Experimental Assessment

- Question: does stateless design slow things down? If so, by how much?
- Metrics:
  - Forwarder overhead: FIB vs PIT lookup for content
  - End-to-end latency
- Approach:
  - Modified ndnSIM 2.1 to support stateless operation
  - NDN Forwarding Daemon (NFD) forwards interests and content based on CNs and BRNs
  - Topologies based on Deutsches ForschungsNetz (DFN) and AT&T core network
  - Each consists of 160 consumers, a single producer multiple (>30) routers
  - Each consumer generates 10 interests/sec, with a random suffix (to avoid cache hits)

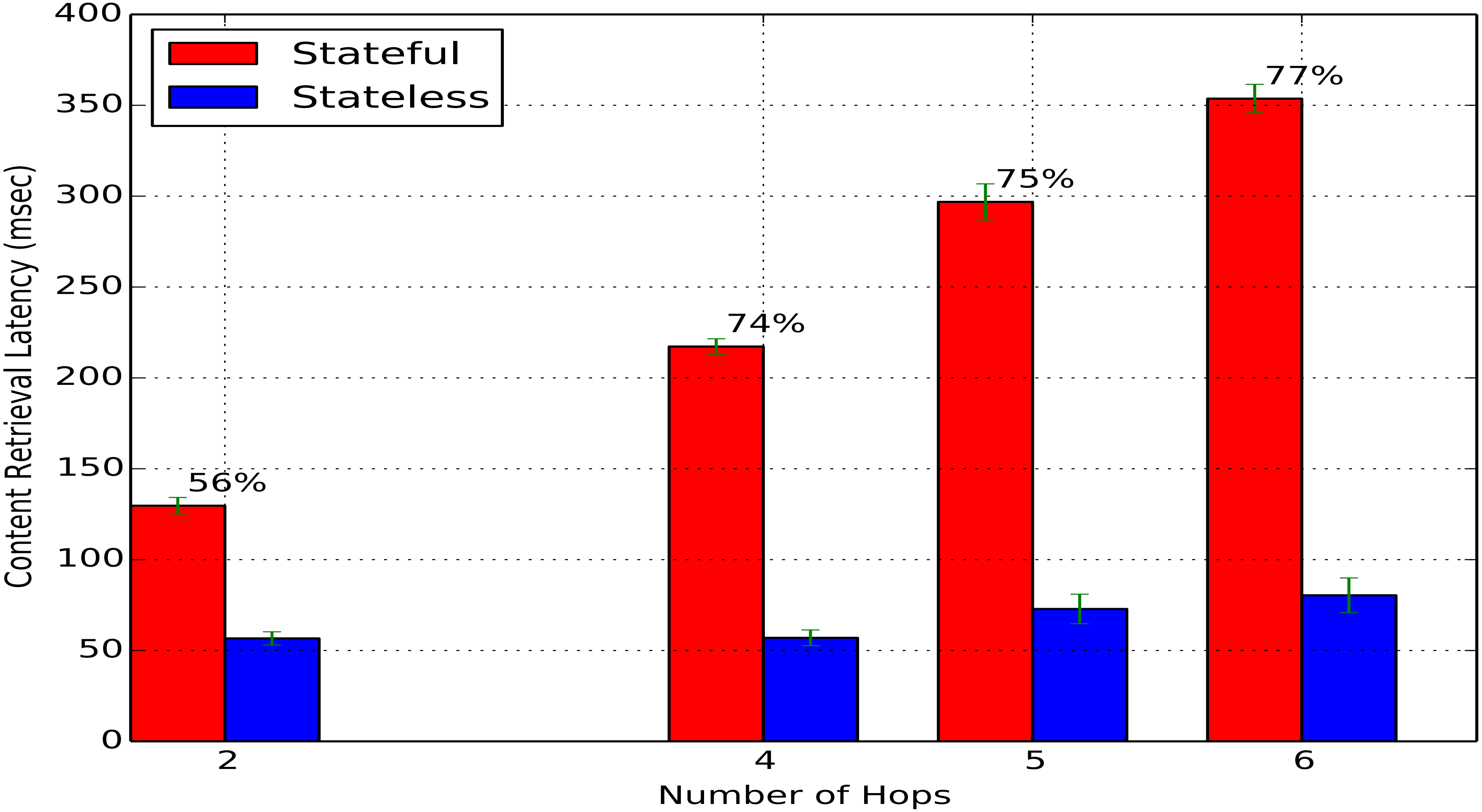
Overhead w/out Caching



Overhead with Caching



End-to-End Latency



Wrapping Up

- Stateless variant addresses **some** security and scalability issues of stateful design
  - But, triggers some new ones
- Hybrid deployment: PITs at the edge with cache-/PIT-less routers in the core
- Or, purely cache-less operation for some traffic, and cache-/PIT-ful for others
- Low overhead wrt stateful variant

- Lots and lots of unchartered territory remains

Questions?

Thanks!