# Content-Centric Networking at Internet Scale through The Integration of Name Resolution and Routing

J. J. Garcia-Luna-Aceves

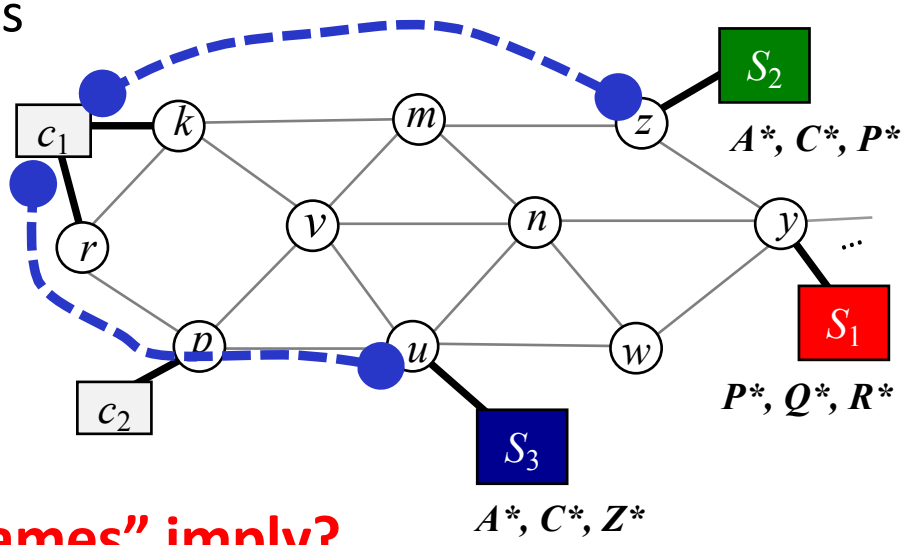
UCSC/Xerox PARC

# Overview

- Routing to names is not better than routing to addresses in a content-centric network
- Implications:
  Limitations of NDN and CCNx
- CCN-RAMP:  CCN based on routing to locations of name prefixes
- Performance comparison of CCN-RAMP and NDN

# Is Routing to Names Better than Routing to Addresses?

- A name may have multiple instantiations (multi-homing of names)
- **A route can exist only between an instantiation of a source and an instantiation of a destination**
- *e.g.*, route from E to B or from E to A in the example



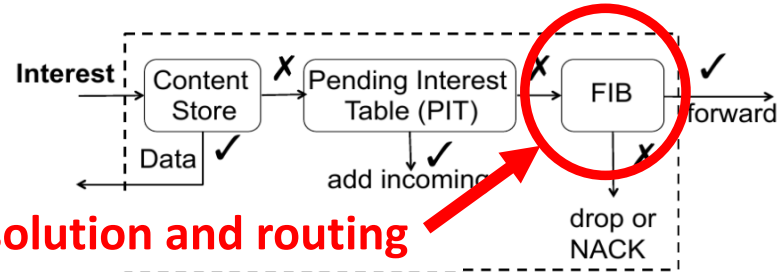

- **What does "routing to names" imply?**
- **What advantages does it provide vs. routing to addresses?**
- **How much does it cost?**
- **Is it a good tradeoff? Do benefits justify the cost?**

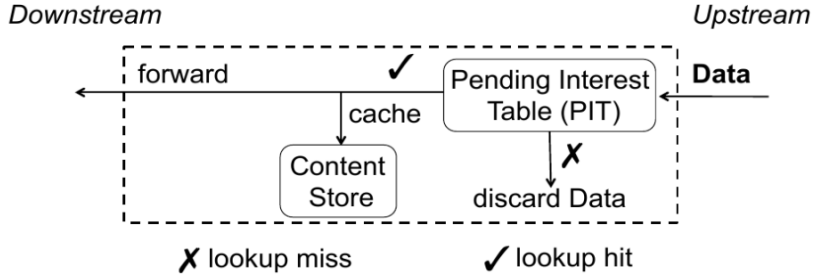
# What Routing to Names Implies

- No name-to-address mapping by special servers
- Name resolution is done at the same time as routing at each hop along the path from consumer to site advertising a name prefix
- **Benefits:**
  - No long delays due to DNS, no user or application involvement in name resolution, …
- **Cost:**
  - Very large FIB at each router; all name prefixes must be listed
  - A large PIT to keep reverse routes
  - **At least two orders of magnitude more complex than table lookup in IP routers:**

    **e.g., > $10^8$ registered domains**

    **< $10^6$ IP address ranges being routed**

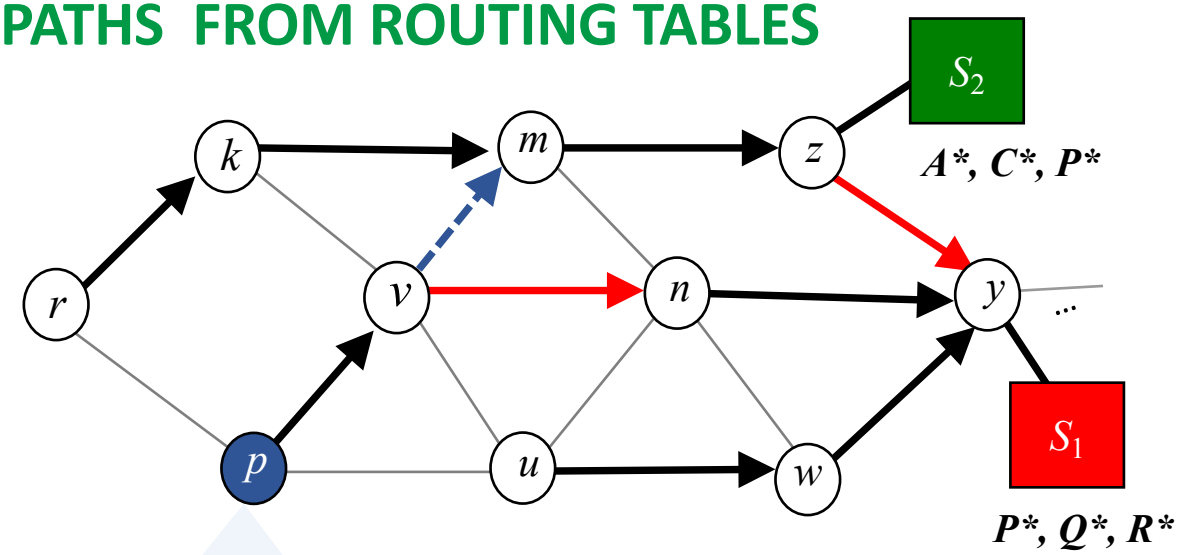



**Each forwarding router is a name resolver!**

# Routing to Names vs. Routing to Their Locations

**PATHS FROM ROUTING TABLES**




- Same routes!
- More entries must be looked up using names than using addresses

**This is important:**

- > $10^8$ registered domains
- < $10^6$ IP address ranges being routed

**or…**

$$\begin{array}{l}\{P^*, Q^*, R^*\}@\, y, \ \text{next} = v \\ \{A^*, C^*, P^*\}@\, z, \ \text{next} = v\end{array} = \begin{array}{l} y, \ \text{next} = v \\ z, \ \text{next} = v\end{array}$$

**Routes to all instances of a name**

$$\begin{array}{l}\{\, Q^*, R^*\}, \quad \text{next} = v \\ \{A^*, C^*, P^*\}, \ \text{next} = v\end{array} = \begin{array}{l} y, \ \text{next} = v \\ z, \ \text{next} = v\end{array}$$

**Routes to nearest instance of a name**

# Routing to Names vs. Routing to Their Locations

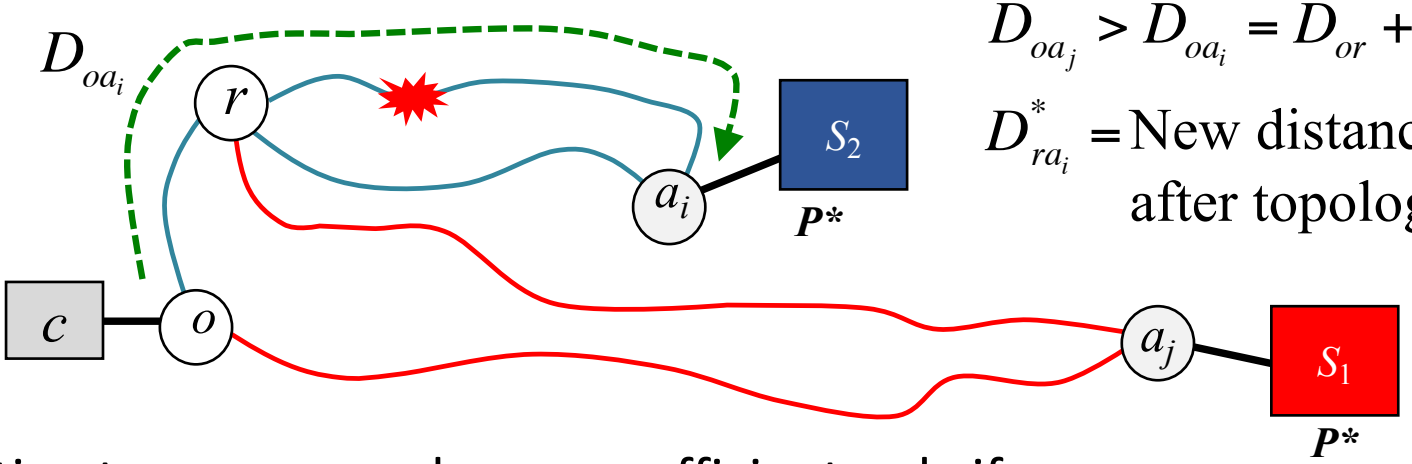
## INFORMATION USED IN ROUTING PROTOCOLS:

- All name-based content routing protocols provide routing information for both name prefixes and the routers where servers storing content for them are attached [**anchors**]
- **Link-state protocols:** Link state advertisements stating virtual links from anchor to name prefix must be used
- **Distance-vector protocols:** Distance to name prefix uses anchor of name prefix used as a label to avoid permanent loops
- **Stating an anchor of a name prefix must be done for a route to the name prefix to be valid.**
- **More efficient to send "updates about anchors" and "updates about mappings of names to anchors" separately**

# Routing to Names vs. Routing to Their Locations

**ADAPTING TO NETWORK DYNAMICS**

Assume first router binds name to an anchor; routes are mostly the same even with topology changes




$$D_{oa_j} > D_{oa_i} = D_{or} + D_{ra_i}$$

$D^*_{ra_i}$ = New distance from $r$ to $a_i$ after topology change

Routing to names can be more efficient only if

$$D^*_{ra_i} < \infty \ \wedge \ D_{ra_j} < D^*_{ra_i} \quad \text{or} \quad D^*_{ra_i} = \infty \ \wedge \ > D_{ra_j} < D_{ro} + D_{oa_j}$$

**, i.e., rarely!**

# Routing to Names in NDN/CCNx

- FIBs list name prefixes
- PITs list reverse paths towards consumers allowing aggregation
- PITs support multicasting support w/o multicast routing protocol

# Routing to Names in NDN/CCNx

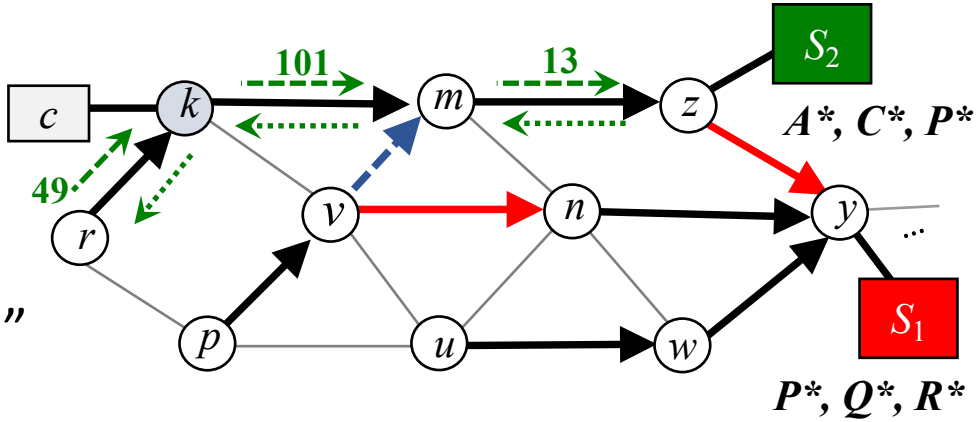
- **FIBs list name prefixes**
  - Forwarding Interests to name prefixes vs. forwarding Interests to addresses of anchors render similar paths with more overhead
- **PITs list reverse paths towards consumers allowing aggregation**
  - Interest aggregation can lead to undetected Interest loops **(see [13, 14])** "Waiting-to-infinity" problem (aggregation and routing-table loops)
  - On-path caching obviates the need for Interest aggregation **(see [8] )**
  - Opens network to new DDoS attacks (Interest flooding)
- **Multicasting support w/o multicast routing protocol**
  - Can be done without per-Interest forwarding state **(see [18])**

**WHY NOT USE LOCATORS RATHER THAN NAMES FOR INTEREST FORWARDING?**

# CCN-RAMP:
# Routing to Anchors Matching Prefixes

## GOAL 1: ELIMINATE LARGE FIBS

- Use Interests, data packets and NACKs as in NDN and CCNx
- Have ingress router [attached to consumer] bind name to an "anchor" (i.e., location of the name prefix)
- Forward Interests towards anchors after initial binding
- Use same routing protocol used in NDN/CCNx to populate routes to anchors and name-anchor mappings

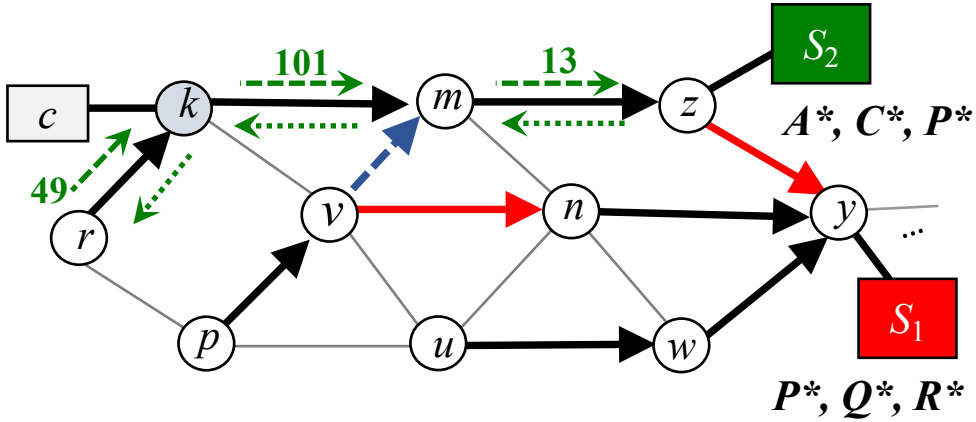



- **Forwarding routers just do routing to anchors**
- **Any content routing protocol can be used**
- **How do ingress routers function as name resolvers?**

# CCN-RAMP:
# Routing to Anchors Matching Prefixes

## GOAL 2: ELIMINATE UNDETECTED INTEREST FORWARDING LOOPS

- Use distance information in forwarding tables
- Add distance-to-anchor information in Interests



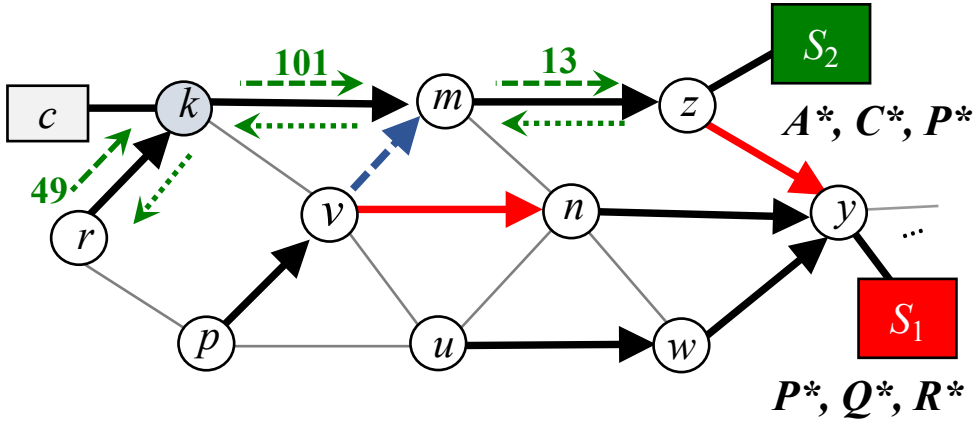

- **Works with any routing protocol**
- **Works under any assumption of topology changes or forwarding-table inconsistencies**
- **Works with or w/o Interest aggregation (i.e., PITs)**

# CCN-RAMP:
# Routing to Anchors Matching Prefixes

## GOAL 3: ELIMINATE PITS

- Maintain same level of Interest anonymity as in NDN/CCNx
- Send back COs using forwarding tables that are O(N)

  [N is number of routers, rather than number of pending Interests)
- Rely on:
  - On-path caching obviates the need for Interest aggregation **[8]**
  - Data-plane multicasting can be done without per-Interest forwarding state **[18]**

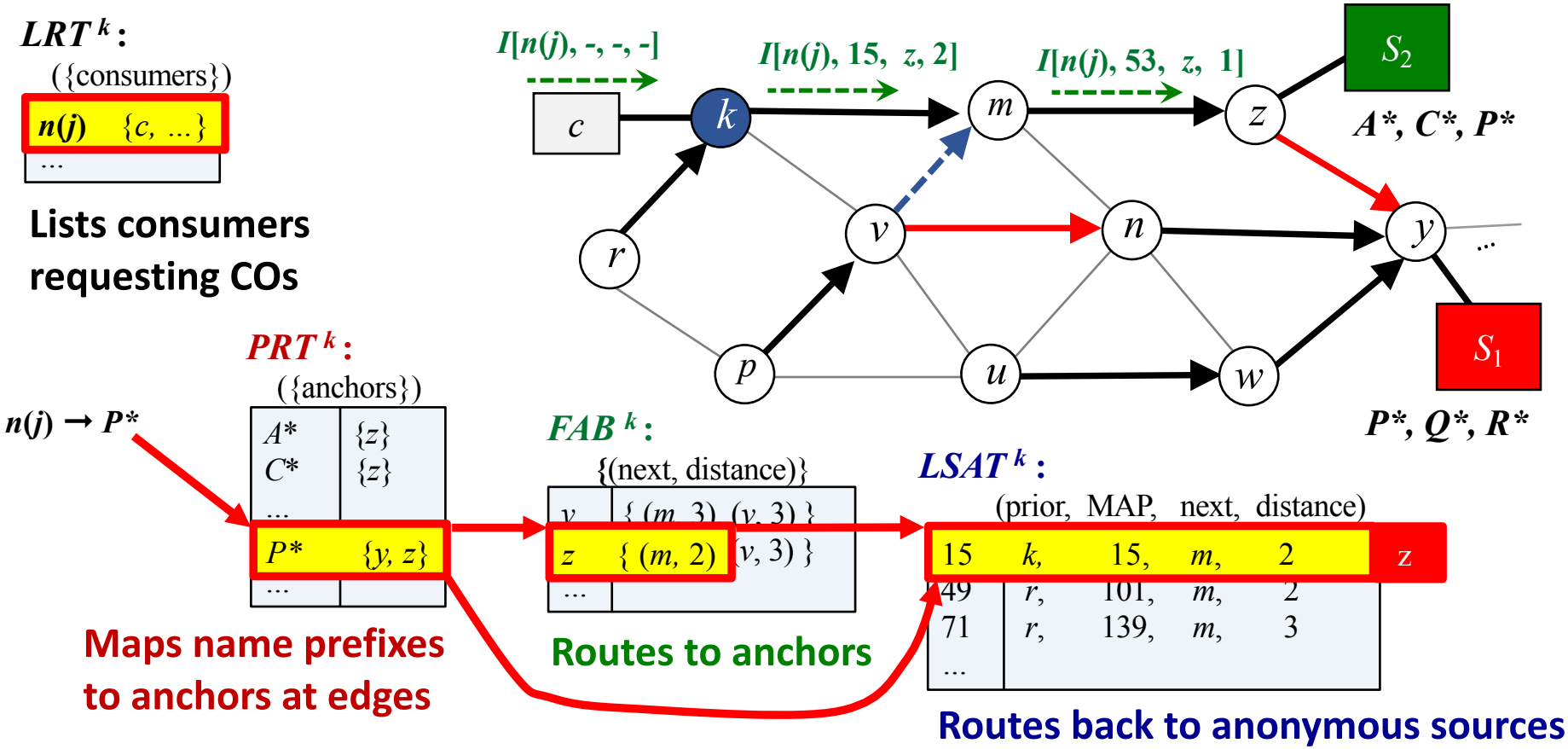



- **What forwarding state should routers maintain?**
- **How can "pseudo-anonymous Interest forwarding" be done without PITs?**

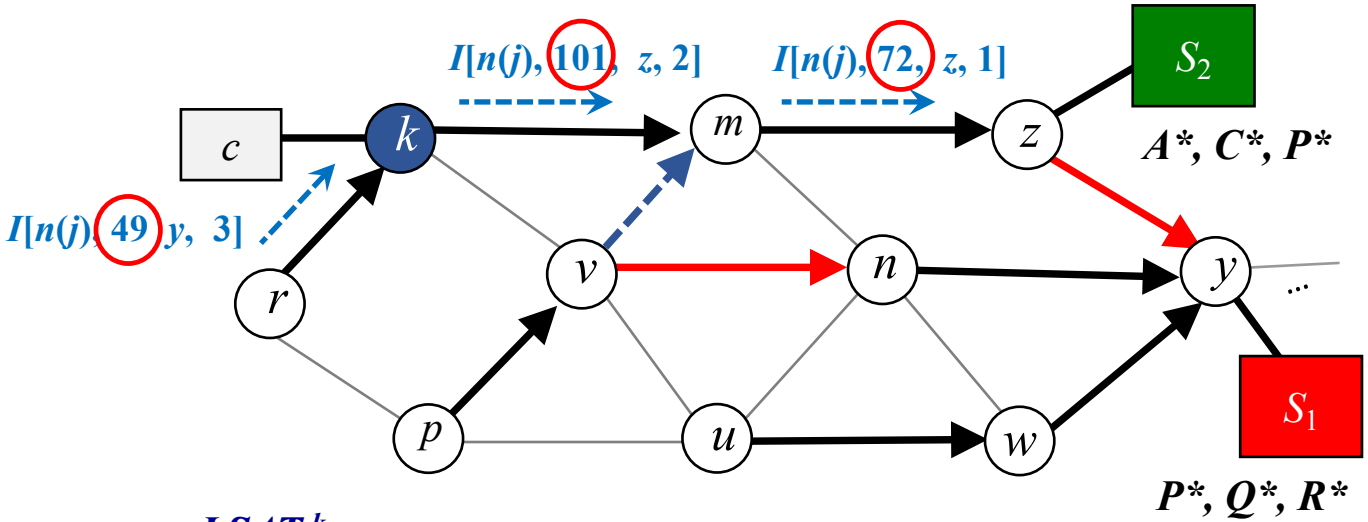
# CCN-RAMP

- **Interest** $I[n(j), AID^I(k), a^I(k), D^I(k)]$ forwarded by router $k$ states
  - **Content name $n(j)$,** an **anonymous identifier $AID^I(k)$,**
  - **anchor $a^I(k)$ of prefix for $n(j)$, distance to anchor $D^I(k)$,**
- **Data packet** $DP[n(j), AID^R(k), sp(j)]$ forwarded by router $k$ states
  - **Name $n(j)$, anonymous identifier $AID^R(k)$, and security payload $sp(j)$**
- Three forwarding tables:
  - **PRT (prefix resolution table):** Used by name resolvers
  - **FAB (forwarding to anchors base):** Used by routers to forward Interests
  - **LSAT (label swapping with anchors):** Used to routers to forward COs back
- **CS (content store)** for edge caching or on-path caching
- **LRT (local resolution table)** for handling Interests from local consumers.

# Interest Forwarding in CCN-RAMP



# Interest Forwarding in CCN-RAMP




***FAB*** $^{k}$ **:**

{(next, distance)}

| | |
|---|---|
| *y* | { (*m*, 3), (*v*, 3) } |
| *z* | { (*m*, 2), (*v*, 3) } |
| ... | |

**Routes to anchors**

***LSAT*** $^{k}$ **:**

(prior, MAP, next, distance)

| | | | | | |
|---|---|---|---|---|---|
| 15 | *k*, | 15, | *m*, | 2 | |
| 49 | *r*, | 101, | *m*, | 2 | z |
| 71 | *r*, | 139, | *m*, | 3 | |
| ... | | | | | |

**Routes back to anonymous sources**

# Data Forwarding in CCN-RAMP

***LRT*** $^{k}$ **:**

({consumers})

| ***n(j)*** | ***{c, …}*** |
|---|---|
| … | |

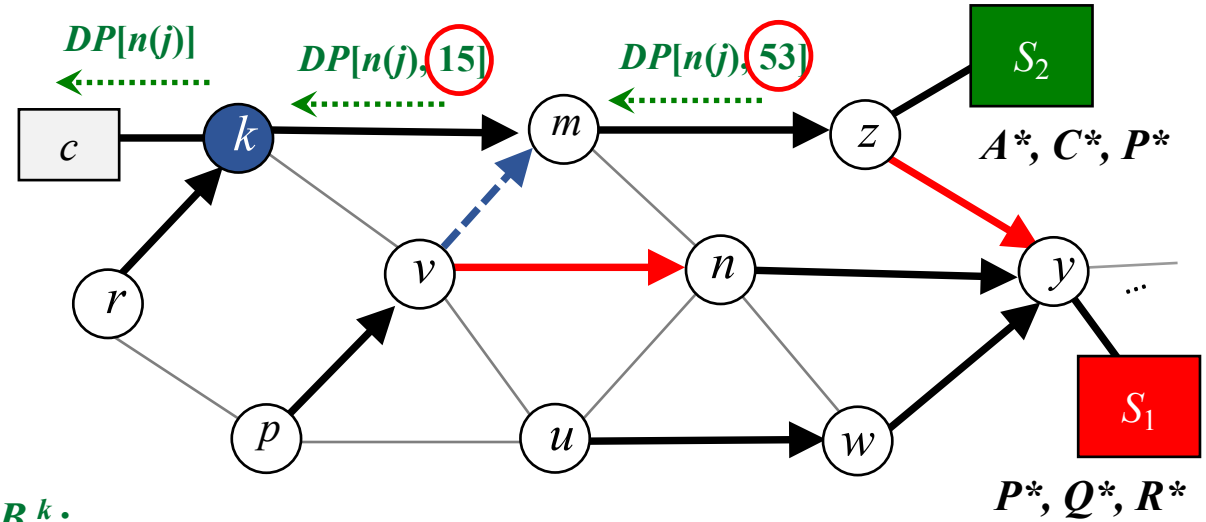
**Lists consumers requesting COs**

***PRT*** $^{k}$ **:**

({anchors})

| | |
|---|---|
| *A** | {*z*} |
| *C** | {*z*} |
| … | |
| *P** | {*y, z*} |
| … | |

**Maps name prefixes to anchors at edges**




***FAB*** $^{k}$ **:**

{(next, distance)}

| | |
|---|---|
| *y* | { (*m*, 3), (*v*, 3) } |
| *z* | { (*m*, 2), (*v*, 3) } |
| … | |

**Routes to anchors**

***LSAT*** $^{k}$ **:**

(prior, MAP, next, distance)

| | | | | | |
|---|---|---|---|---|---|
| 15 | *k*, | 15, | *m*, | 2 | z |
| 49 | *r*, | 101, | *m*, | 2 | |
| 71 | *r*, | 139, | *m*, | 3 | |
| … | | | | | |

**Routes back to anonymous sources**

# CCN-RAMP: Preventing Forwarding Loops

- FAB maintains next hop and distance to each anchor known by router
- $S^i_a$ = set of next hops to anchor $a$ in $FAB^i$
- $D^i\,[AID^I(k)]$ = distance to $AID^I(k)$ in $LSAT^i$
- $D(\,i, a, v\,)$ = distance from neighbor $v$ to anchor $a$ in $FAB^i$

**Rule (ALF): Router *i* can forward Interest from *k* if :**

$$AID^I(k) \notin LSAT^i \wedge \exists\, v \in S^i_a\ (\,D^I(k) > D(\,i, a, v\,)\,)$$

$$AID^I(k) \in LSAT^i \wedge \exists\, v \in S^i_a\ (\,D^I(k) > D^i\,[AID^I(k)]\,)$$

Closer according to FAB

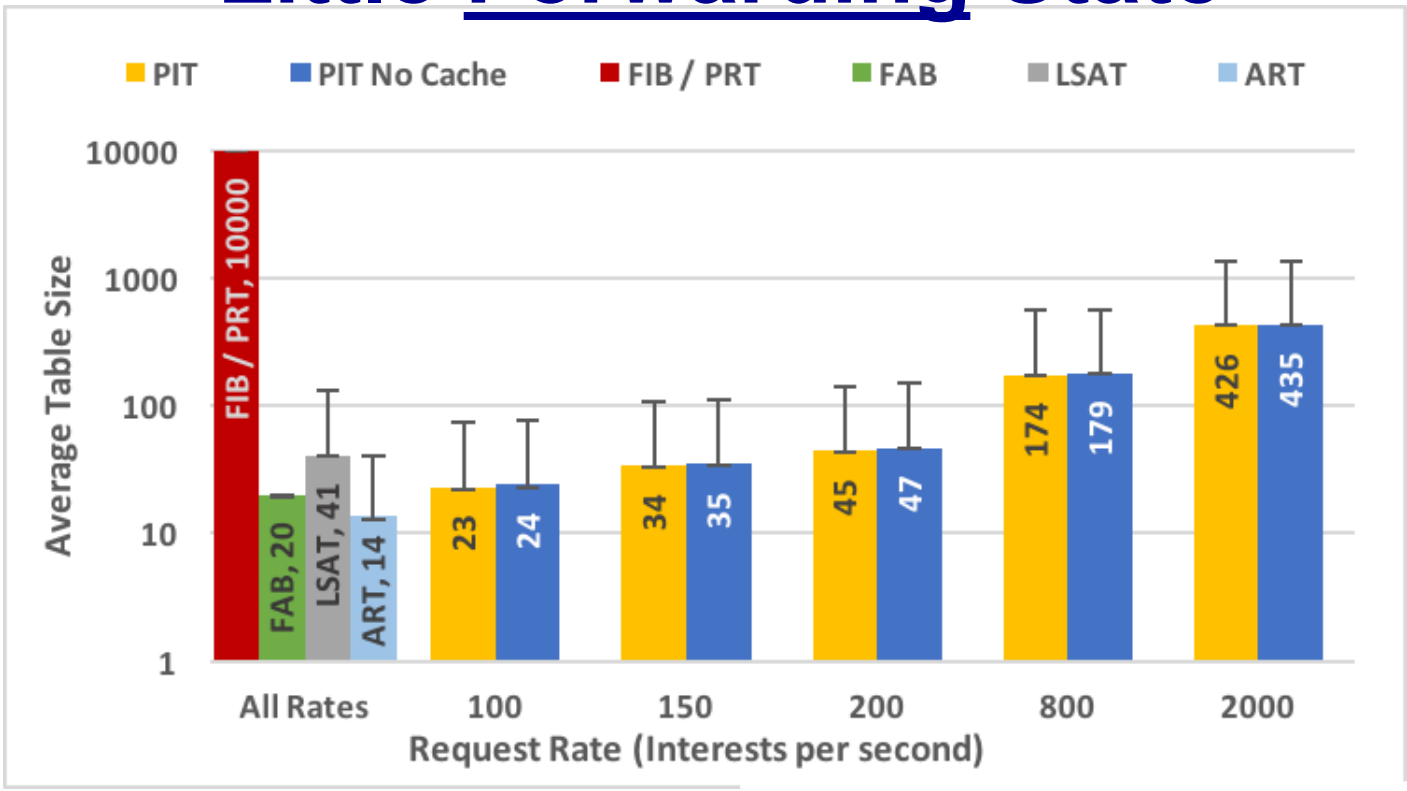
Closer according to LSAT

See Theorem 2 in paper

# Performance Comparison

- Used ndnSIM tool with NDN implementation and implemented CCN-GRAM based on description in ICN 2016 paper
- Used AT&T network topology: 153 nodes and 184 point-to-point links with 30 ms delay
- On-path caches can store up to 1000 content objects (CO)
- Total number of COs is $10^7$, with 1000 COs per name prefix.
- 20 nodes selected randomly to be anchors of 500 different name prefixes each, and each name prefix has a single anchor
- 70 nodes selected randomly to have local consumers
- Consumers generate Interests requesting COs from all name prefixes following a Zipf distribution with parameter $\alpha = 0.7$

# Little Forwarding State

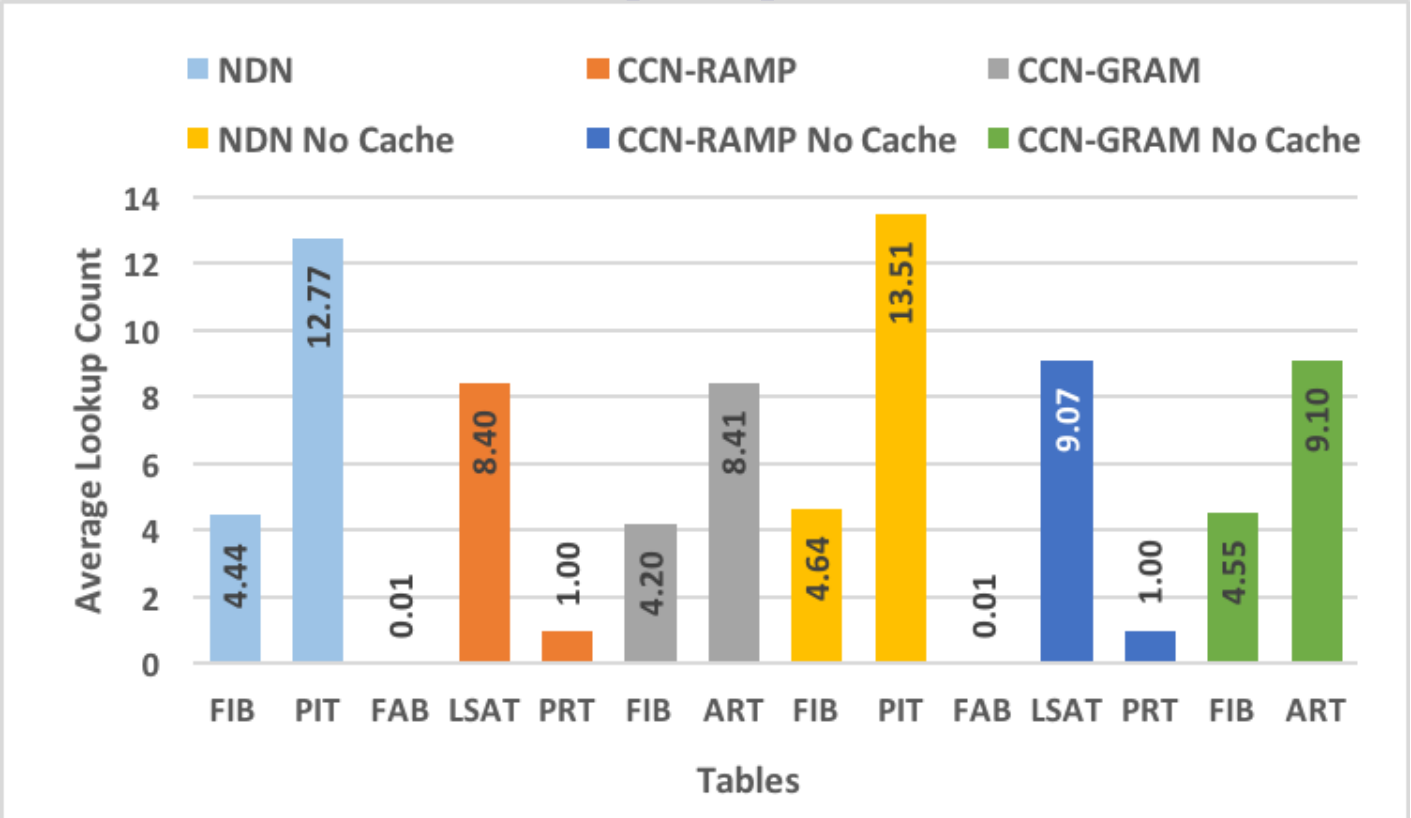



**FABs are orders or magnitude smaller than FIBs. LSATs can be orders of magnitude smaller than PITs**

**PRTs are same order as name-prefix FIBs.**

**Used only at ingress routers; can be improved.**

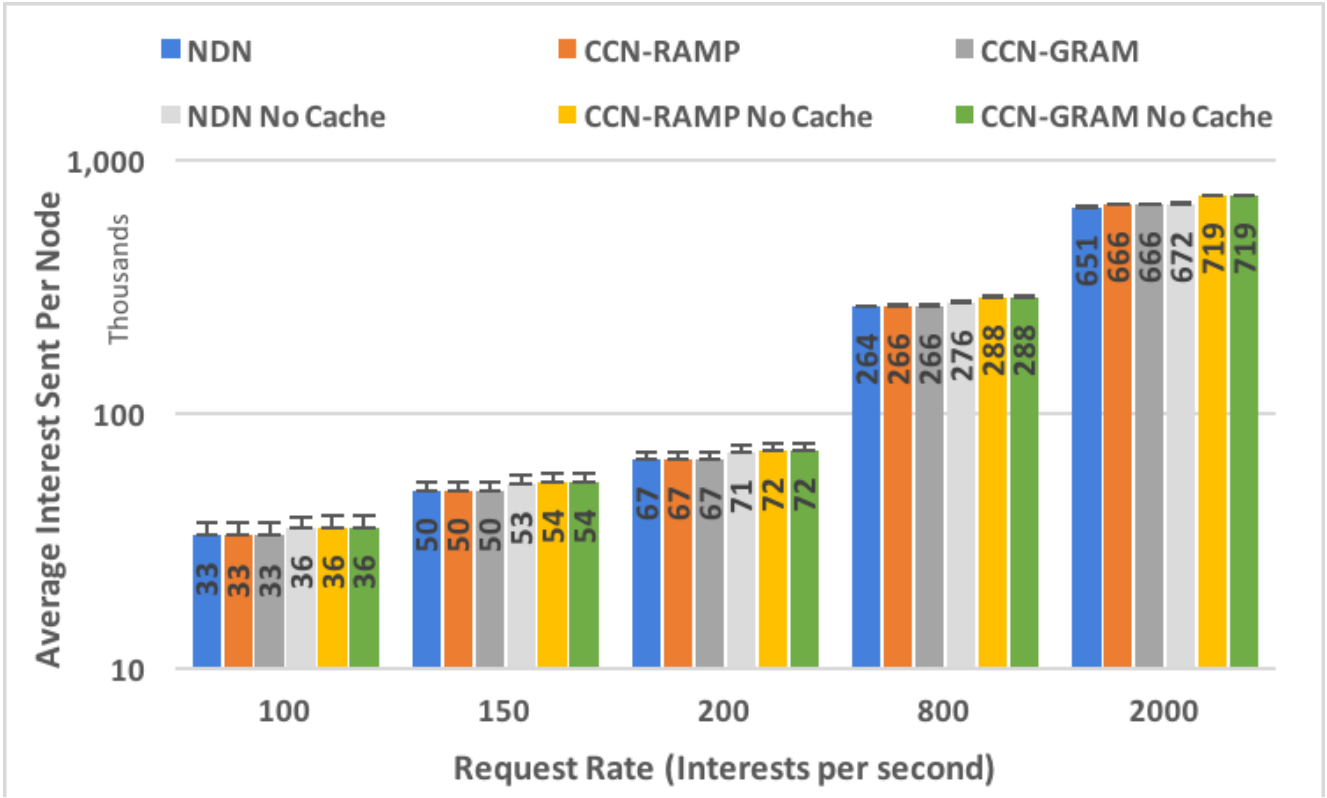
# Fewer Table Lookups per Content Object




**Only ingress routers look up PRTs ("replace role of DNS").**

**FAB and LSAT lookups much faster than FIB and PIT lookups.**

# Similar Number of Interests Forwarded




- Expected, in-network caching works
- Additional benefit: CCN-RAMP is immune to Interest flooding attacks

# Conclusions

- **Content-centric networking ≠ Routing to named data**
  - Using names does not really improve over using addresses, on the contrary
  - Indirection makes forwarding much simpler
  - Directories and routing to addresses need not increase signaling overhead vs. name-based routing
- **Next (a sample):**
  - **If you love PITs:** Smaller FIBs and loop-free Interest forwarding can be applied to NDN/CCNx while keeping PITs
  - **If you hate PRTs everywhere:** Not every router needs to be a name resolver even it it has local consumers
  - **If you love IoT**: Benefits of content-centric networking w/o expensive/complex routers embedded in things