Needle in a Haystack: Mitigating Content Poisoning in Named-Data Networking

Cesar Ghali, Gene Tsudik, and Ersin Uzun

NDSS Workshop on Security of Emerging Networking Technologies (SENT)
February 23, 2014

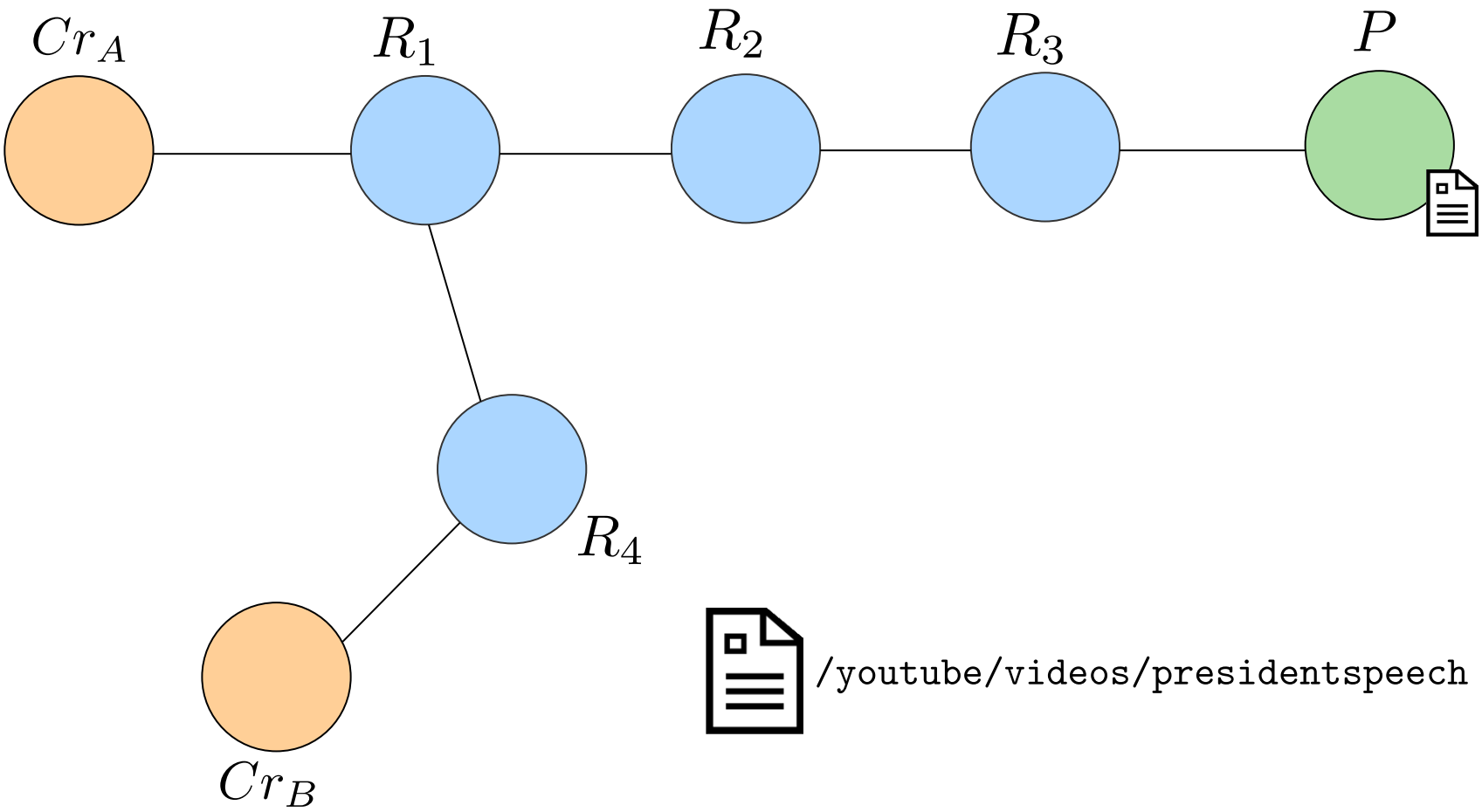
Outline

NDN Overview

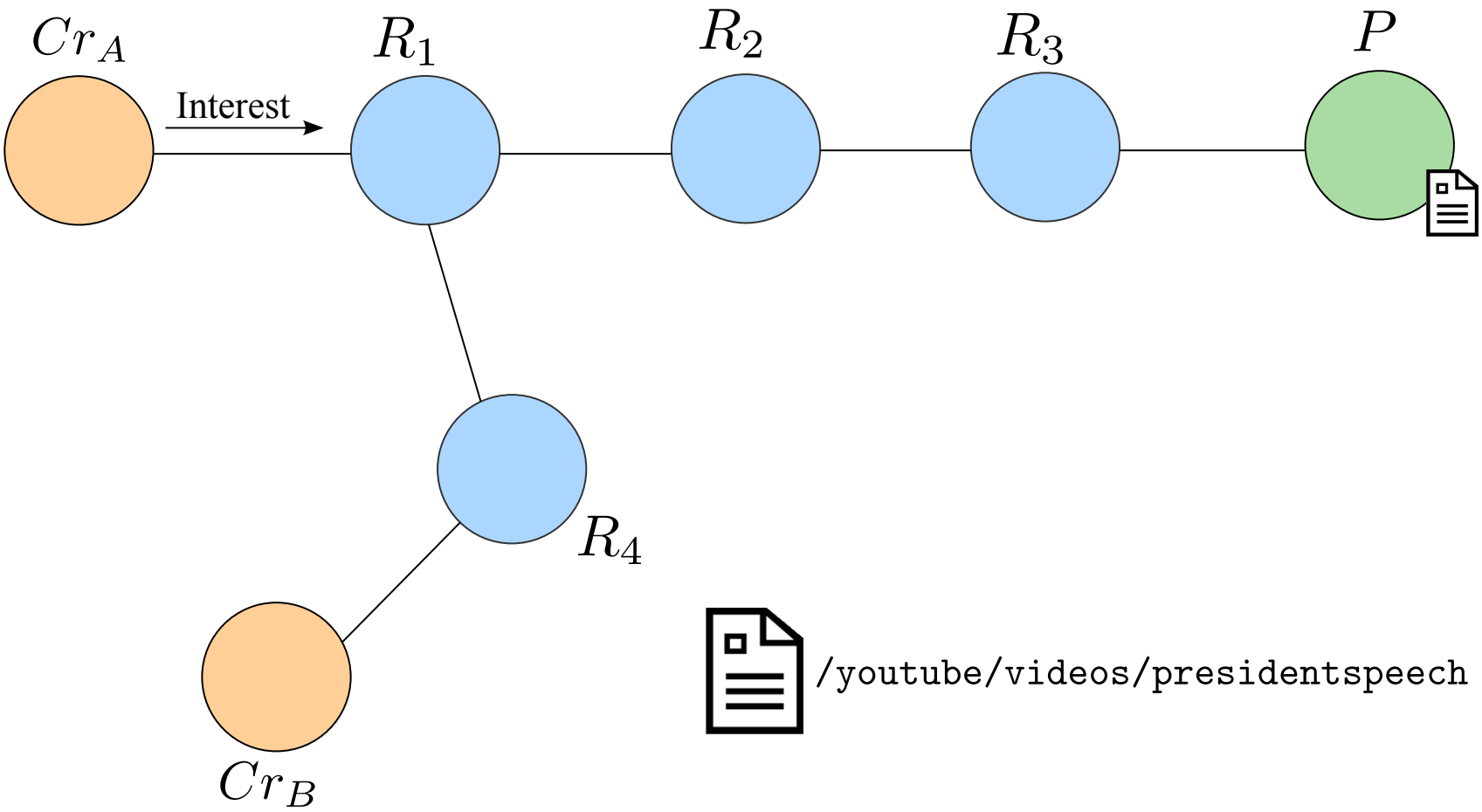
- Current Internet is designed
  - For point-to-point
  - Not content distribution
- Research efforts: Develop new Internet architecture

- Named-Data Networking (NDN):
  - Funded by NSF as part of FIA program
  - 10 US institutions
  - Security and privacy by design

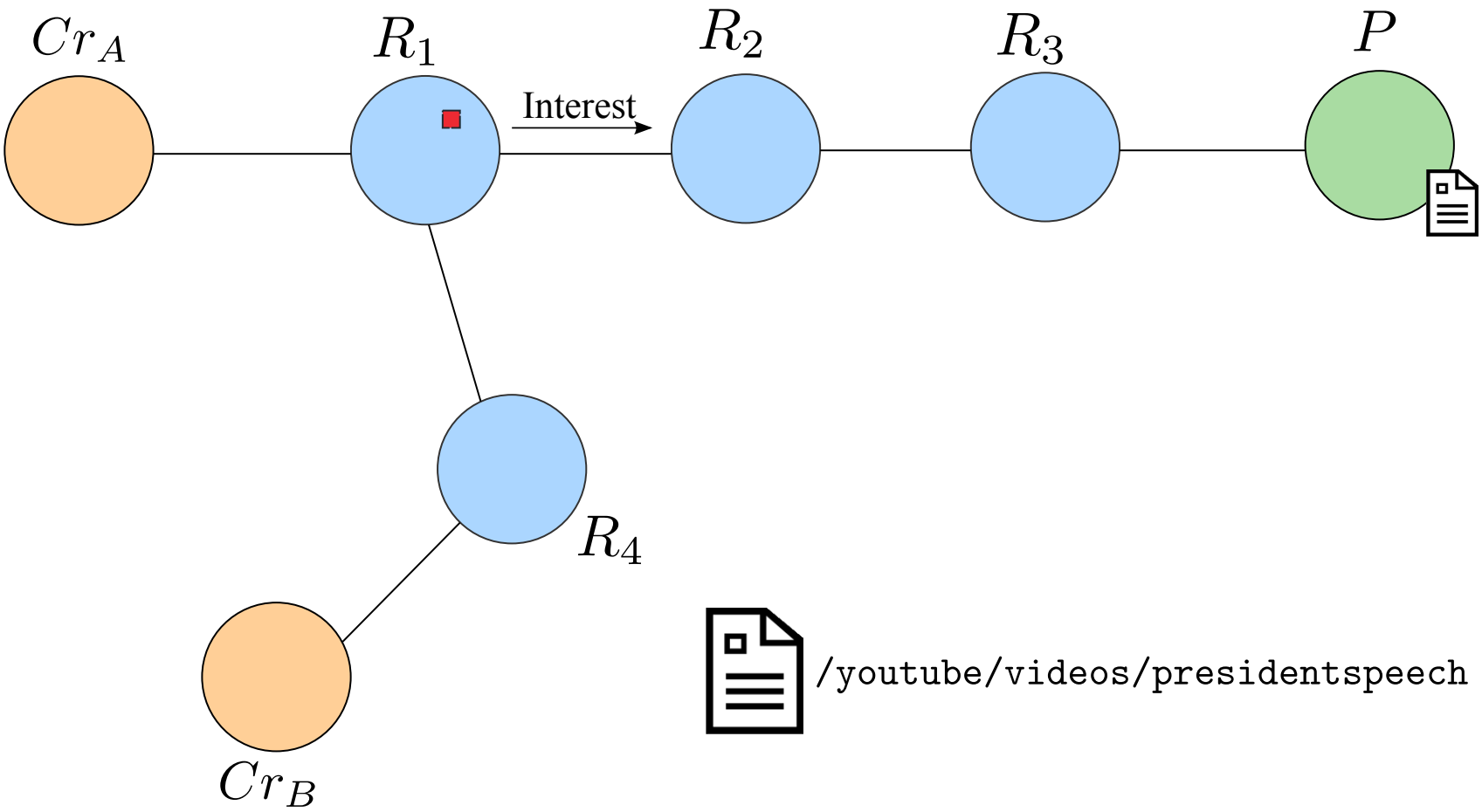
NDN Overview



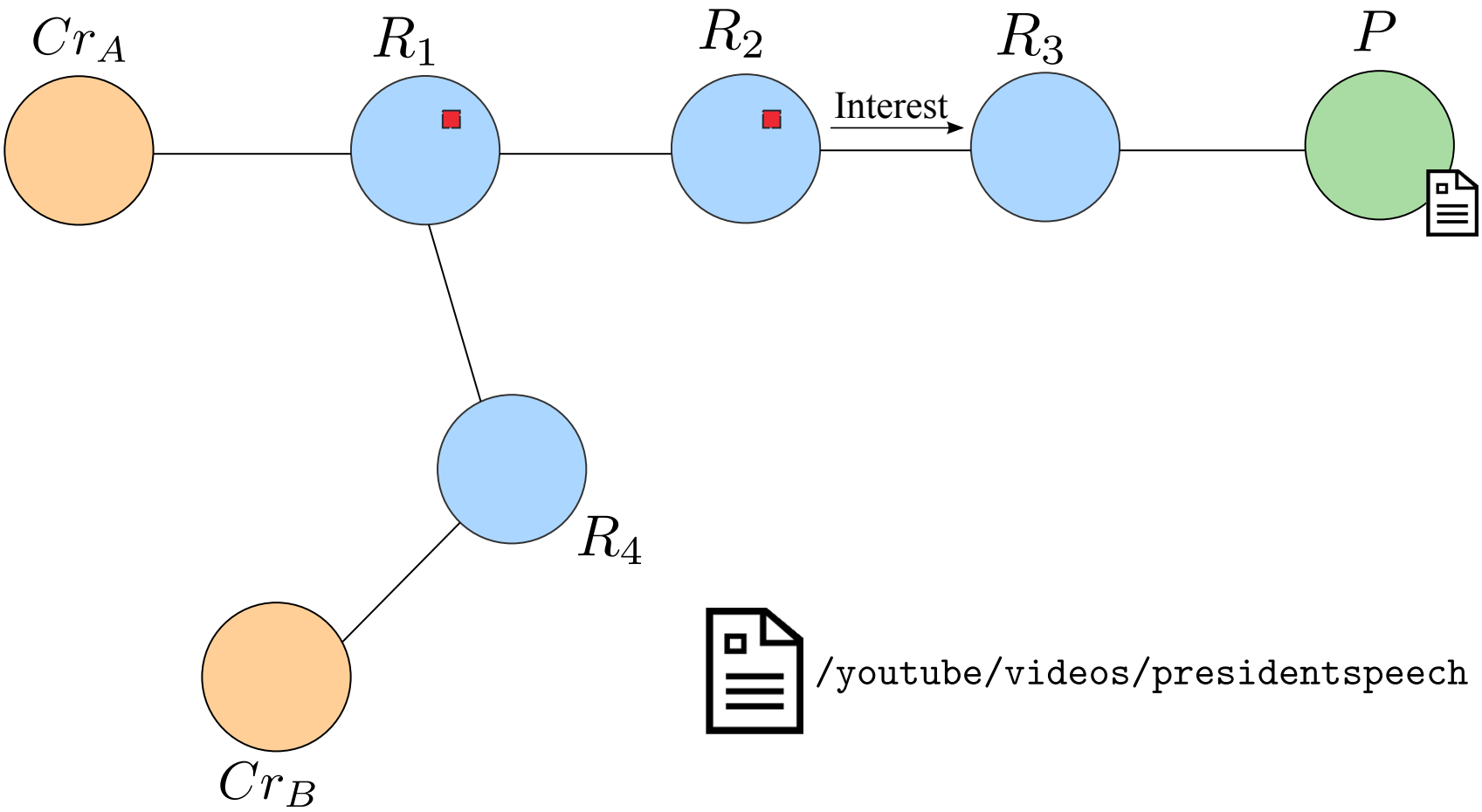
NDN Overview



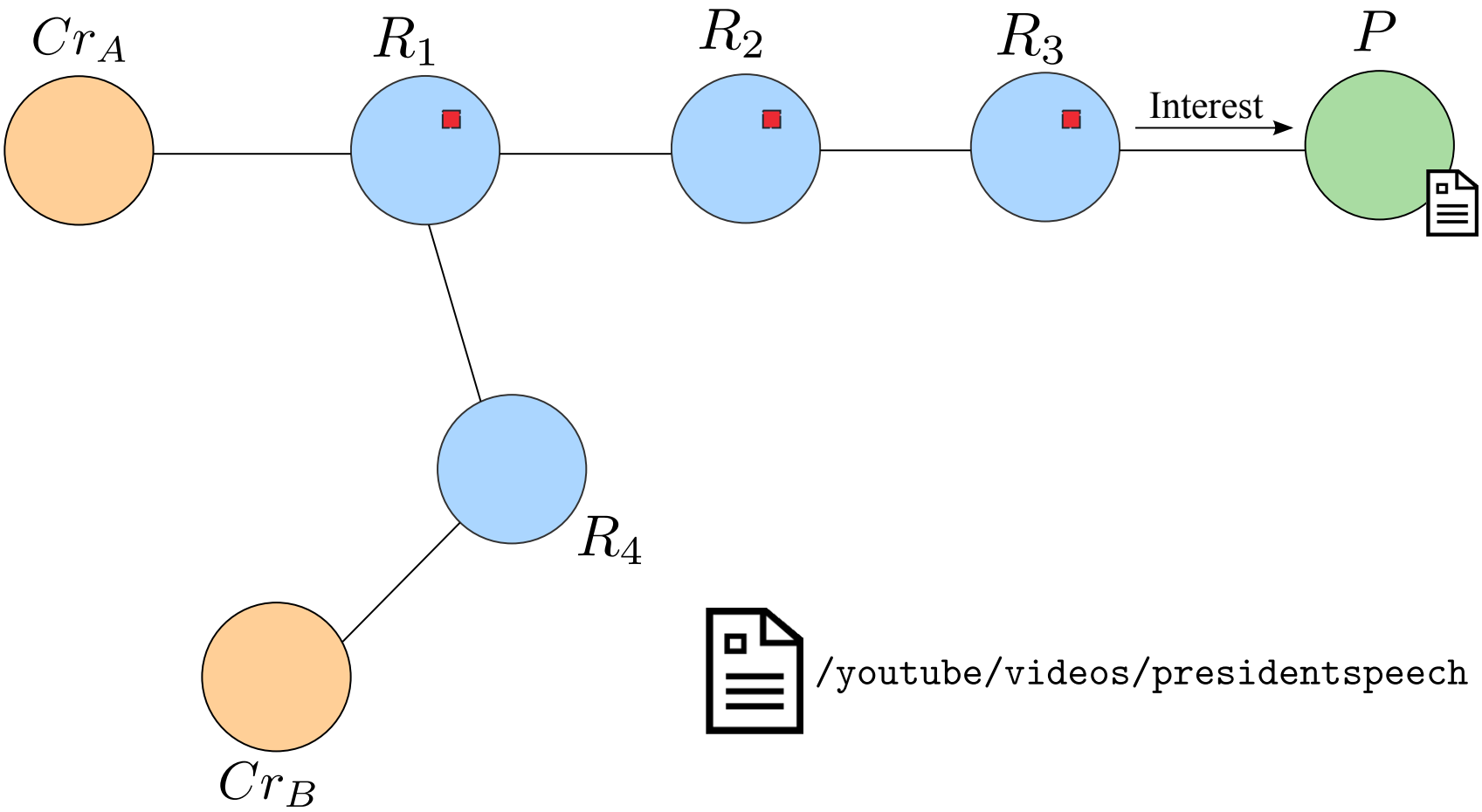
NDN Overview



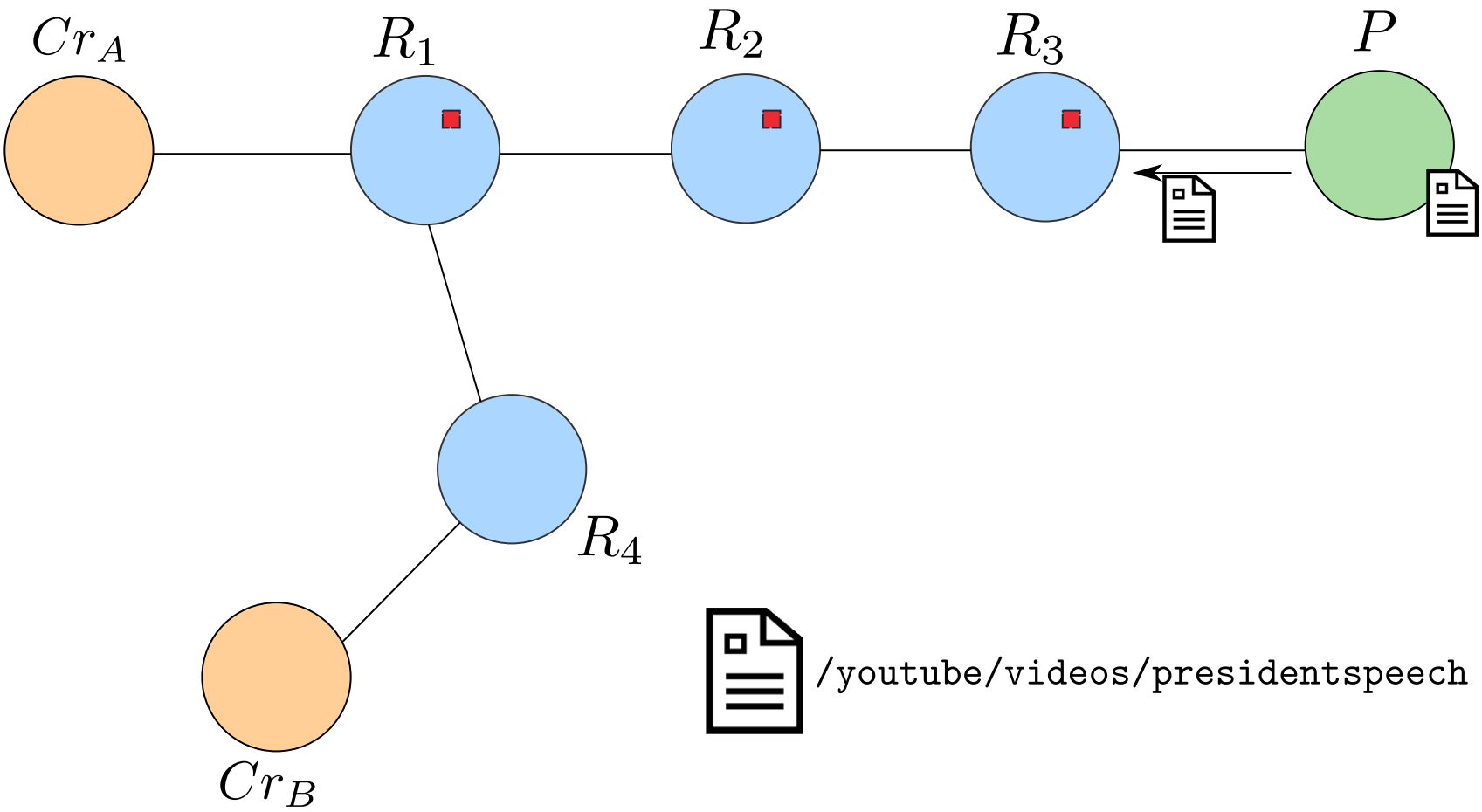
NDN Overview



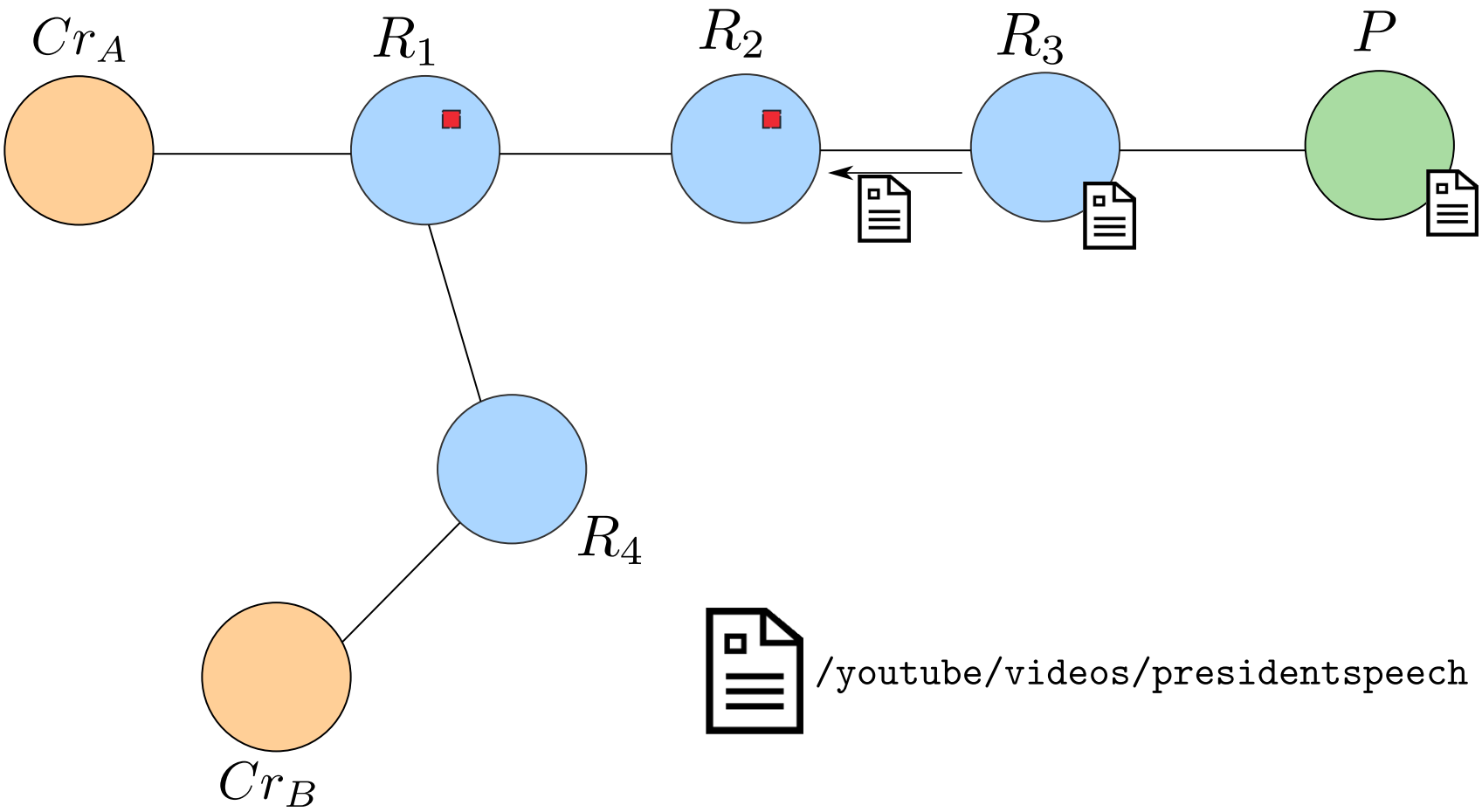
NDN Overview



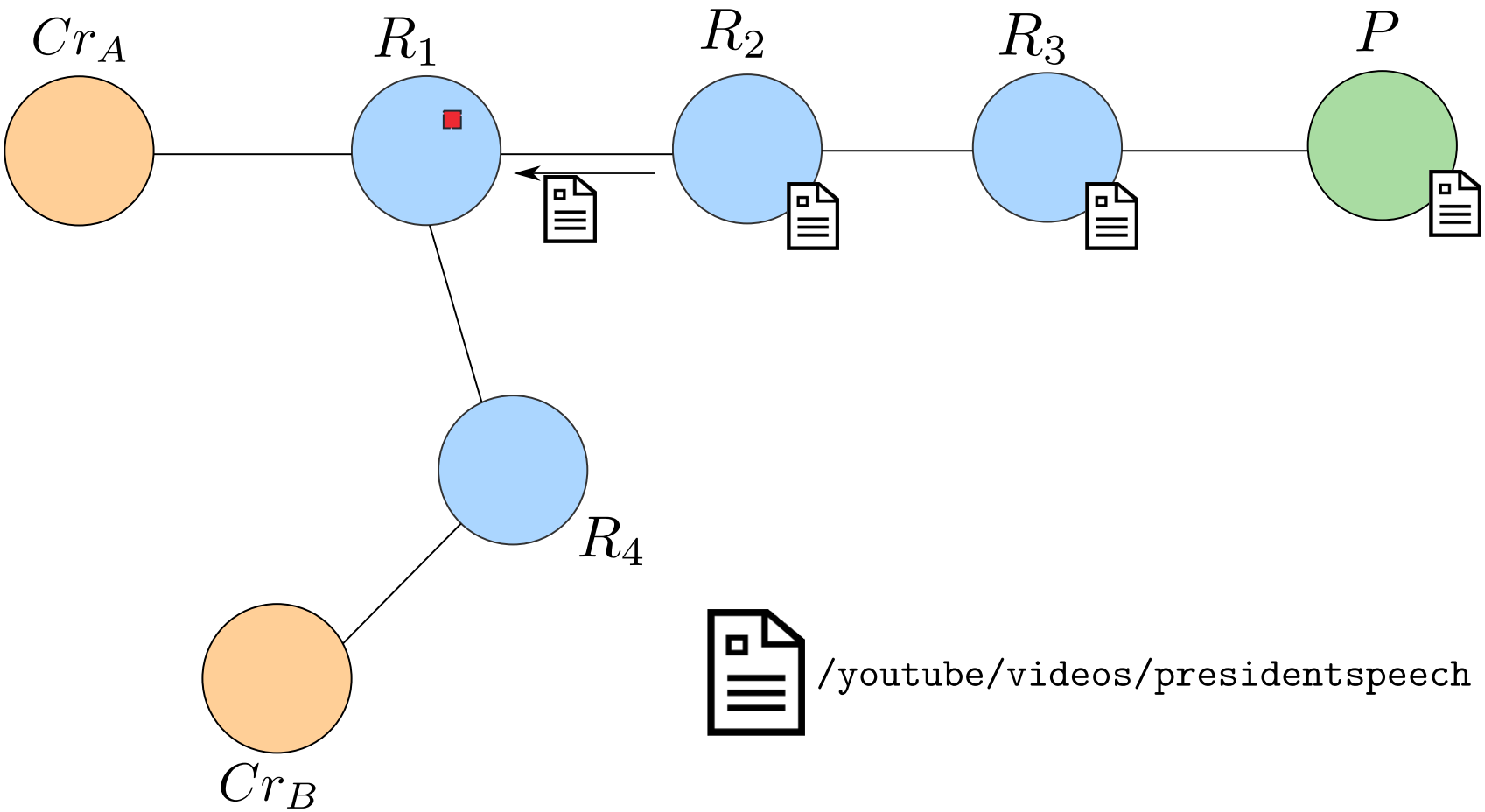
NDN Overview



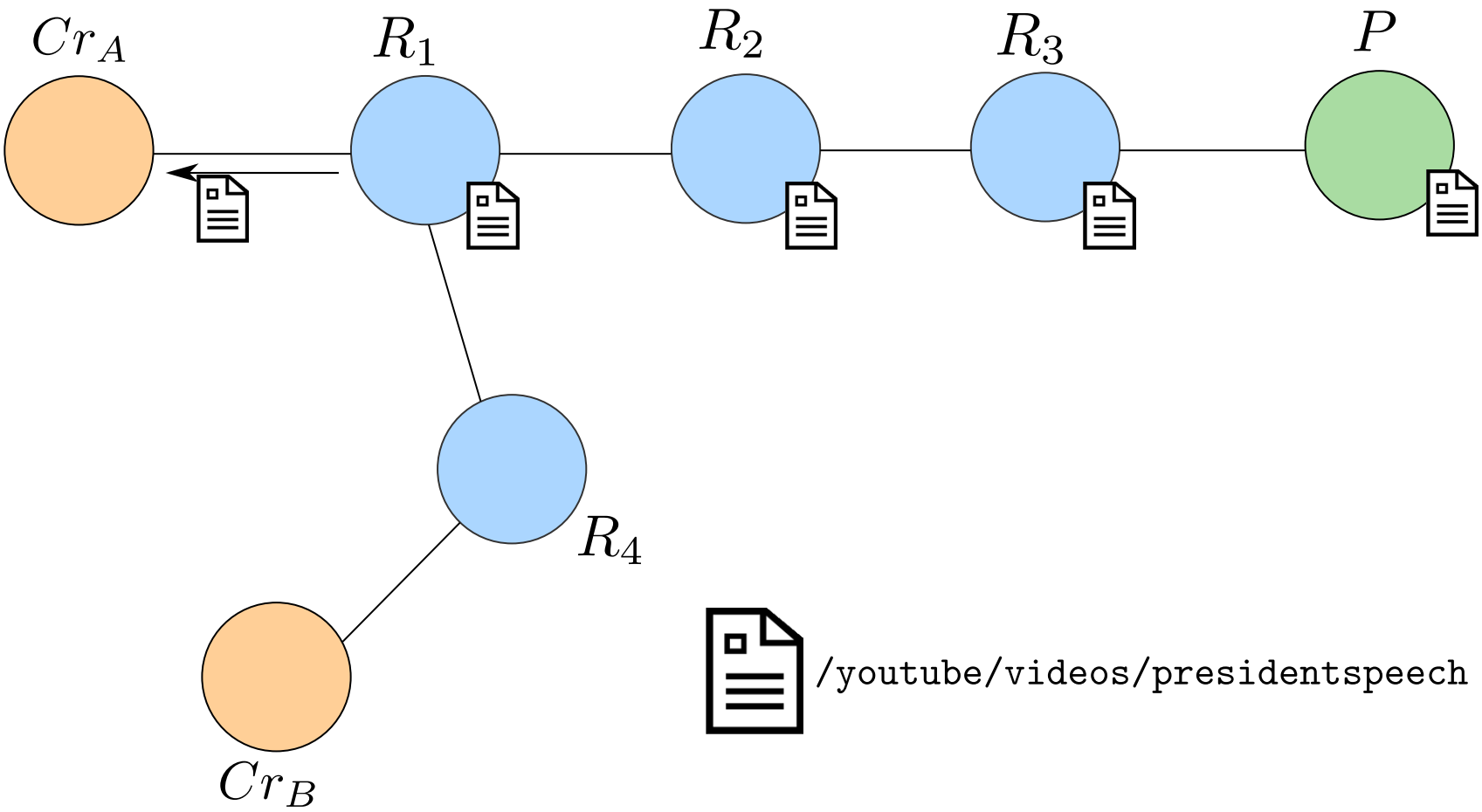
NDN Overview



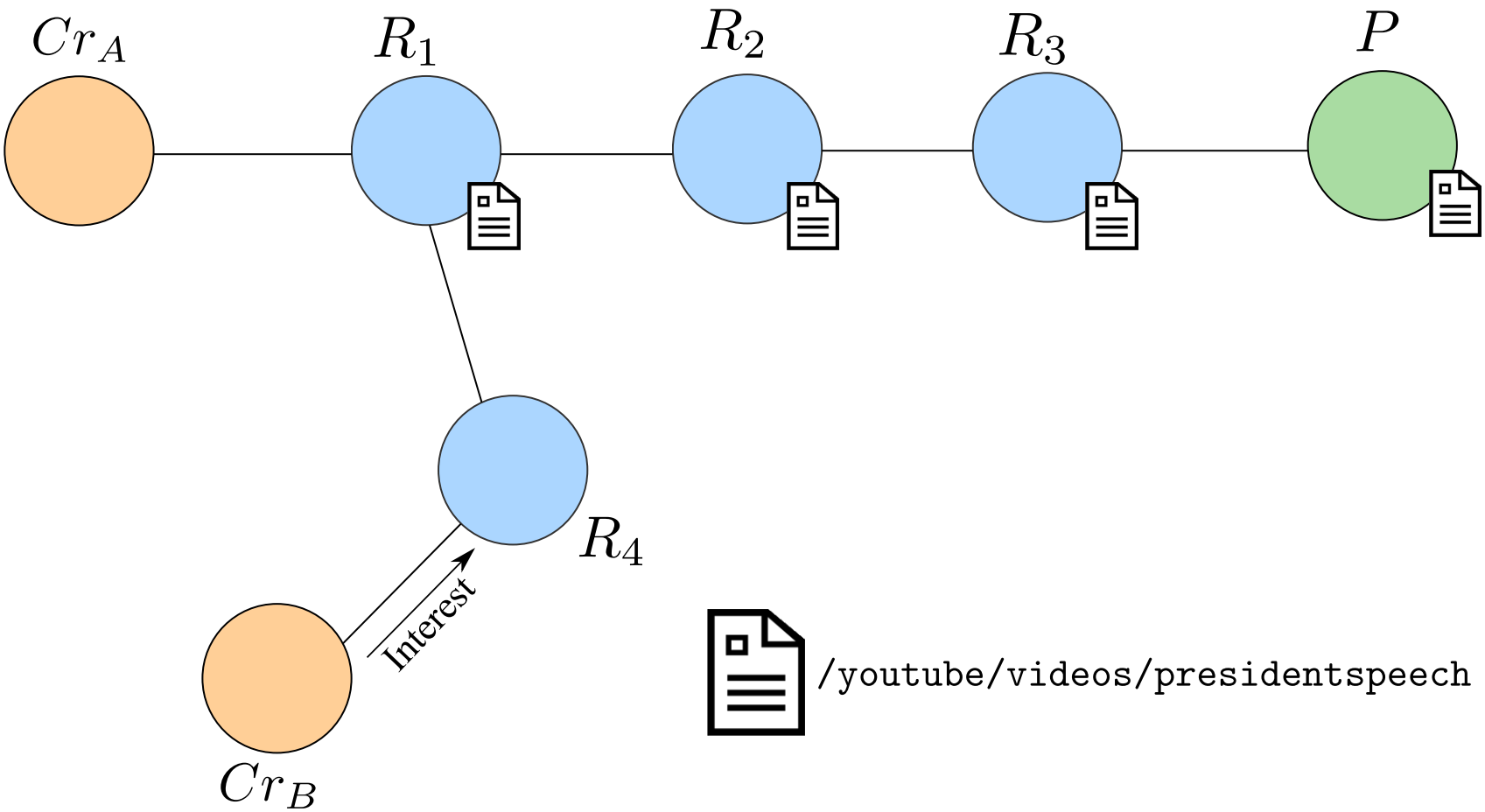
NDN Overview



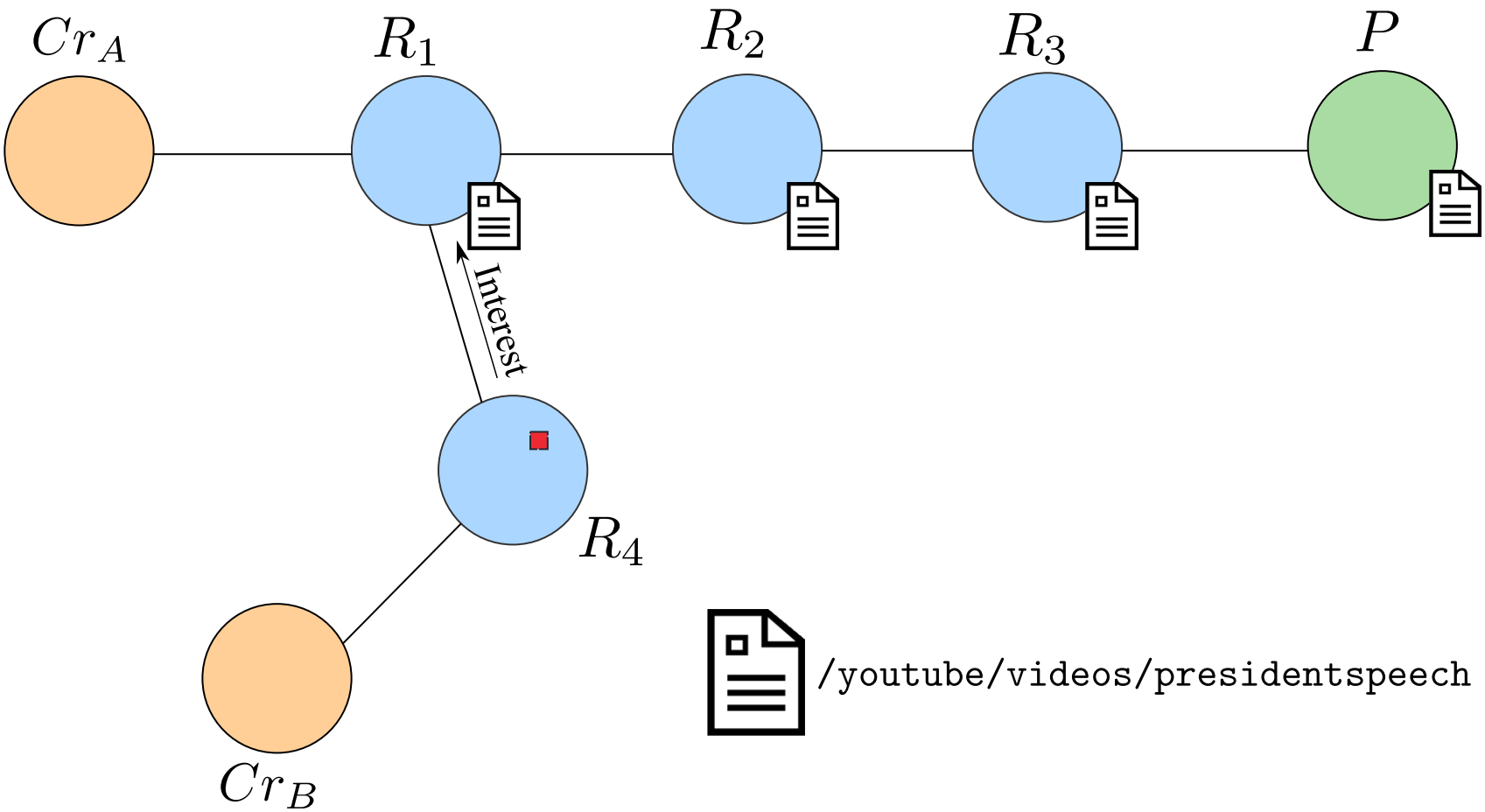
NDN Overview



NDN Overview



NDN Overview



Outline

Problem Definition

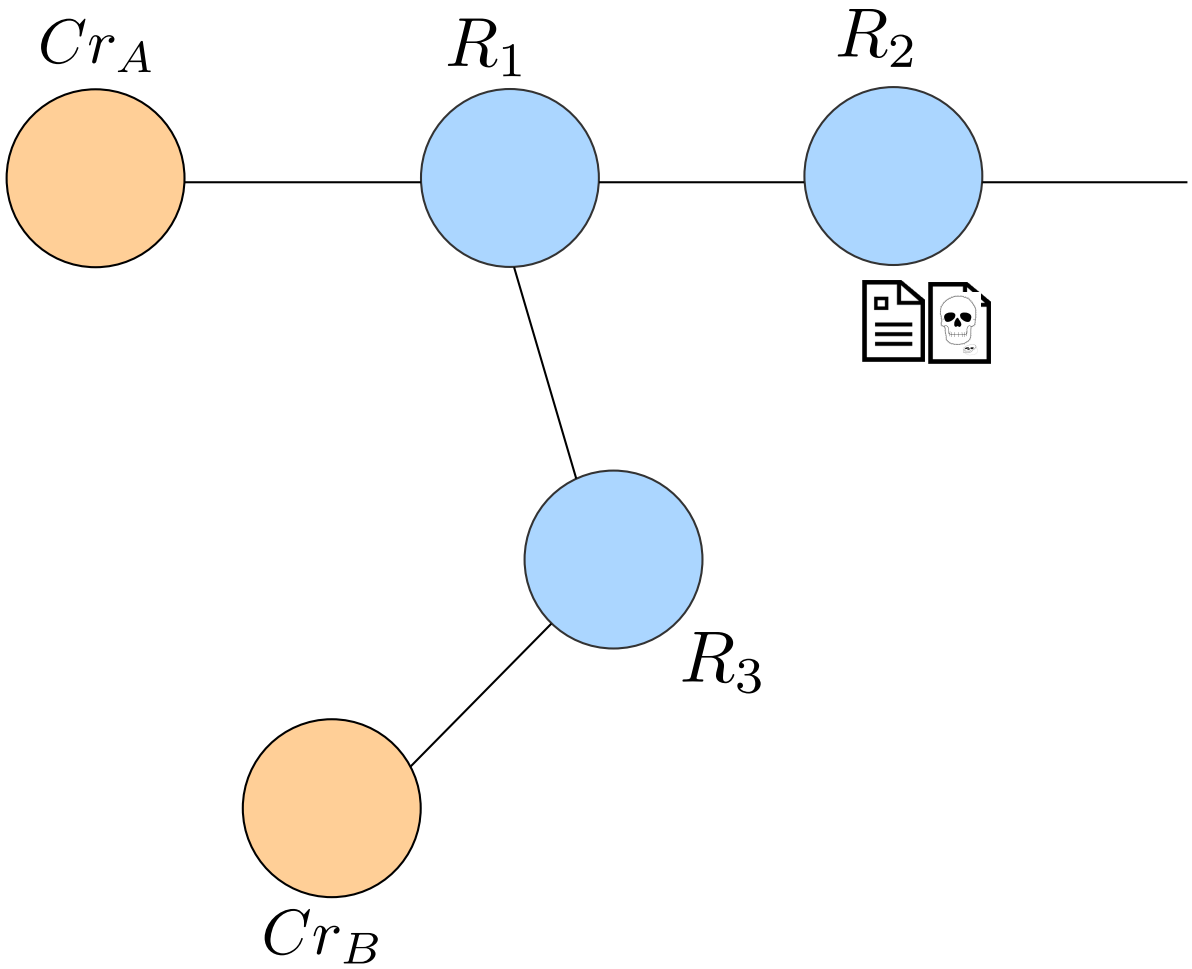
- NDN has built in security features
  - Producer signs content
  - Consumer verifies signature
- Verifying signatures in routers is expensive
- Fake content can be injected into router caches
  - Consumers verify signature
  - No mechanism to cause removal of fake content from router caches

Counter-measures

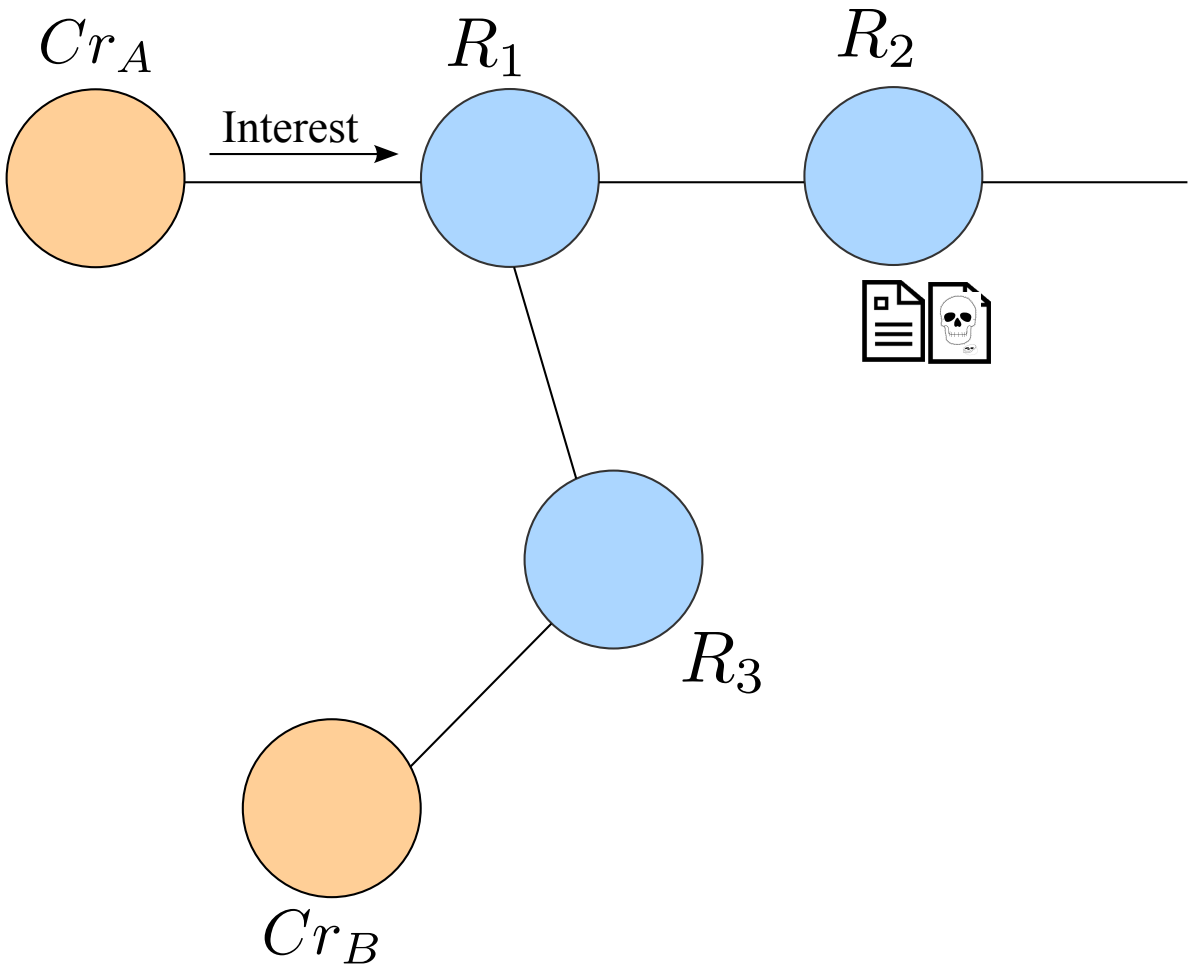
- Routers verifying signatures prevents poisoning
  - Expensive
  - Requires fetching, parsing and verifying public keys
  - Know trust context

- Light-weight content ranking approach
  - Observe consumer behavior when receiving fake content

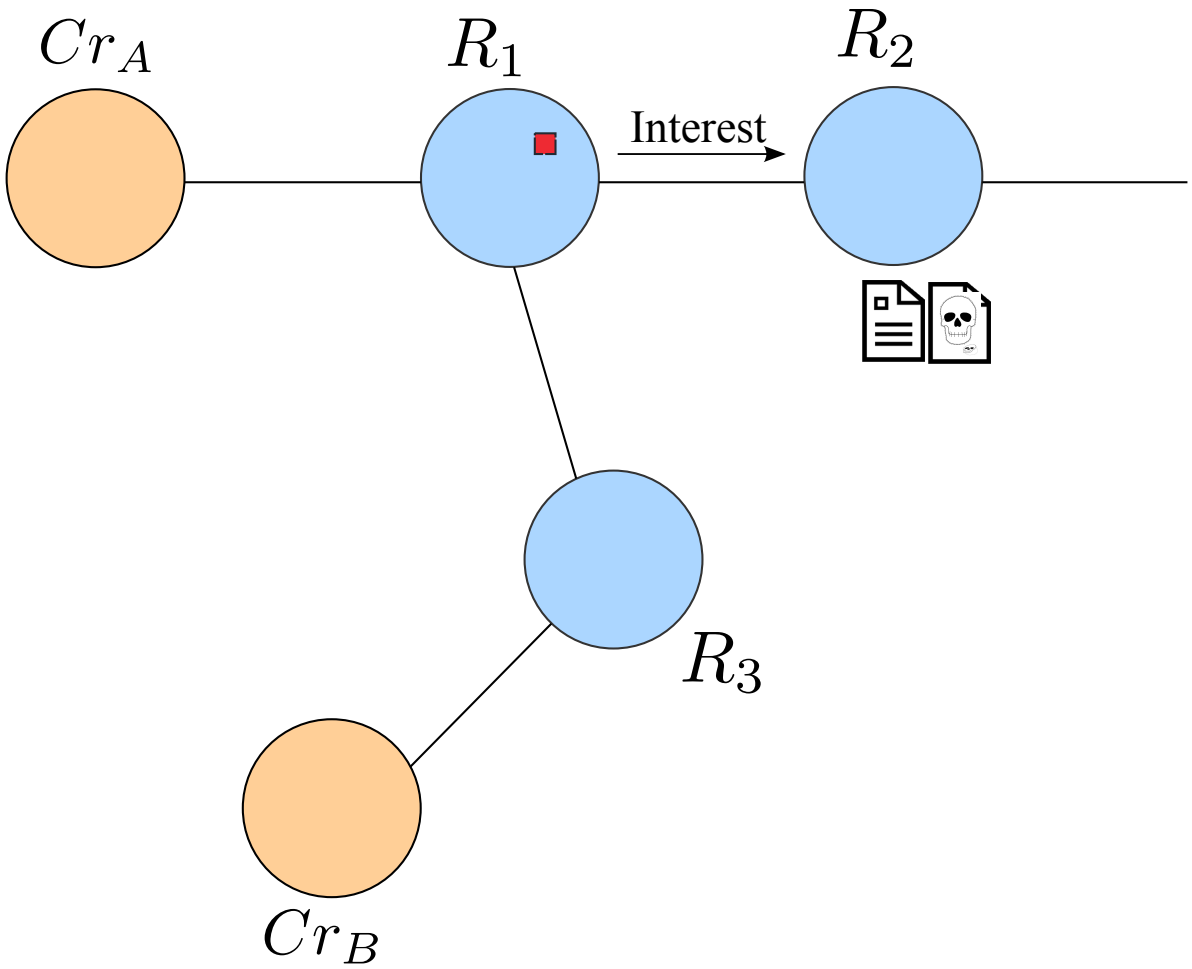
Counter-measures



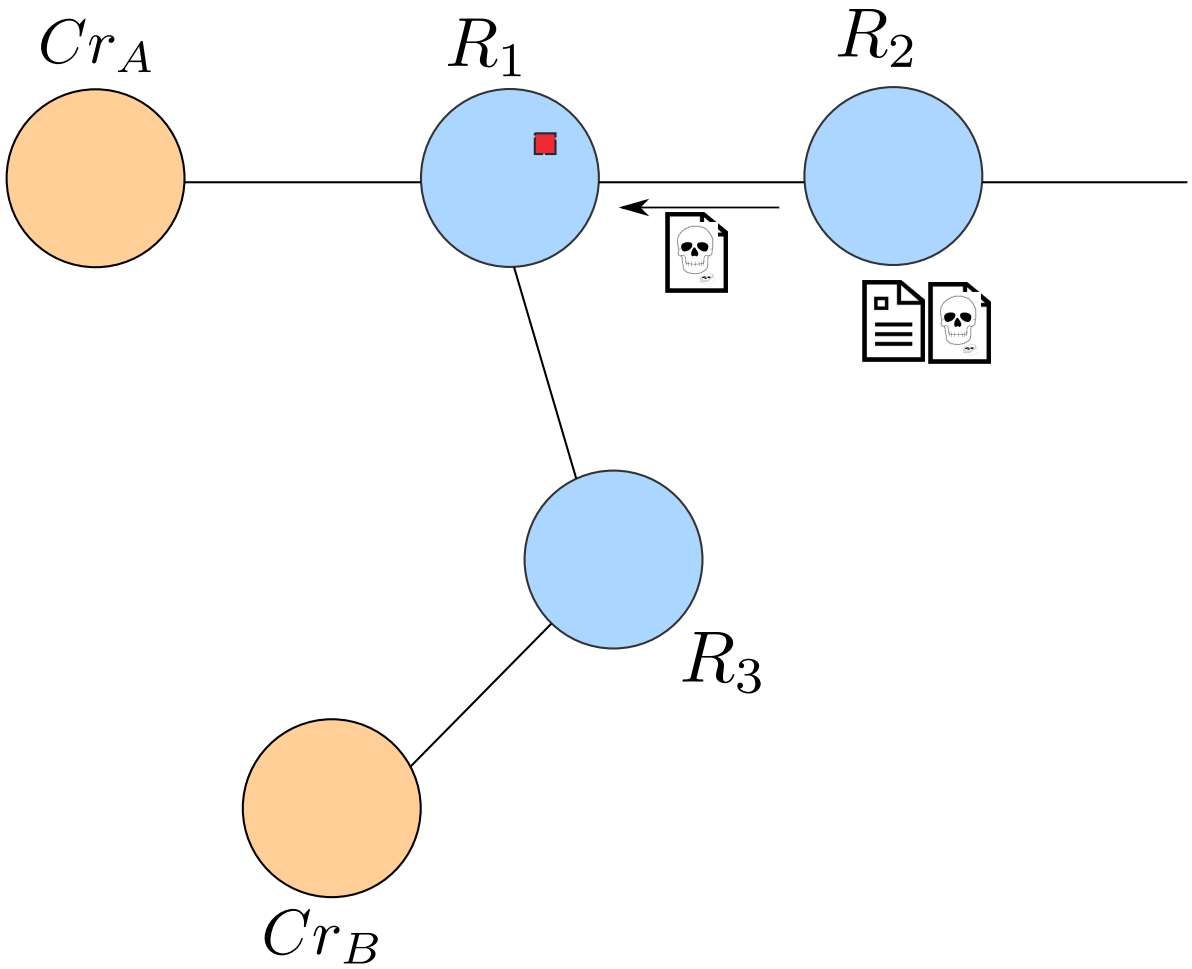
Counter-measures



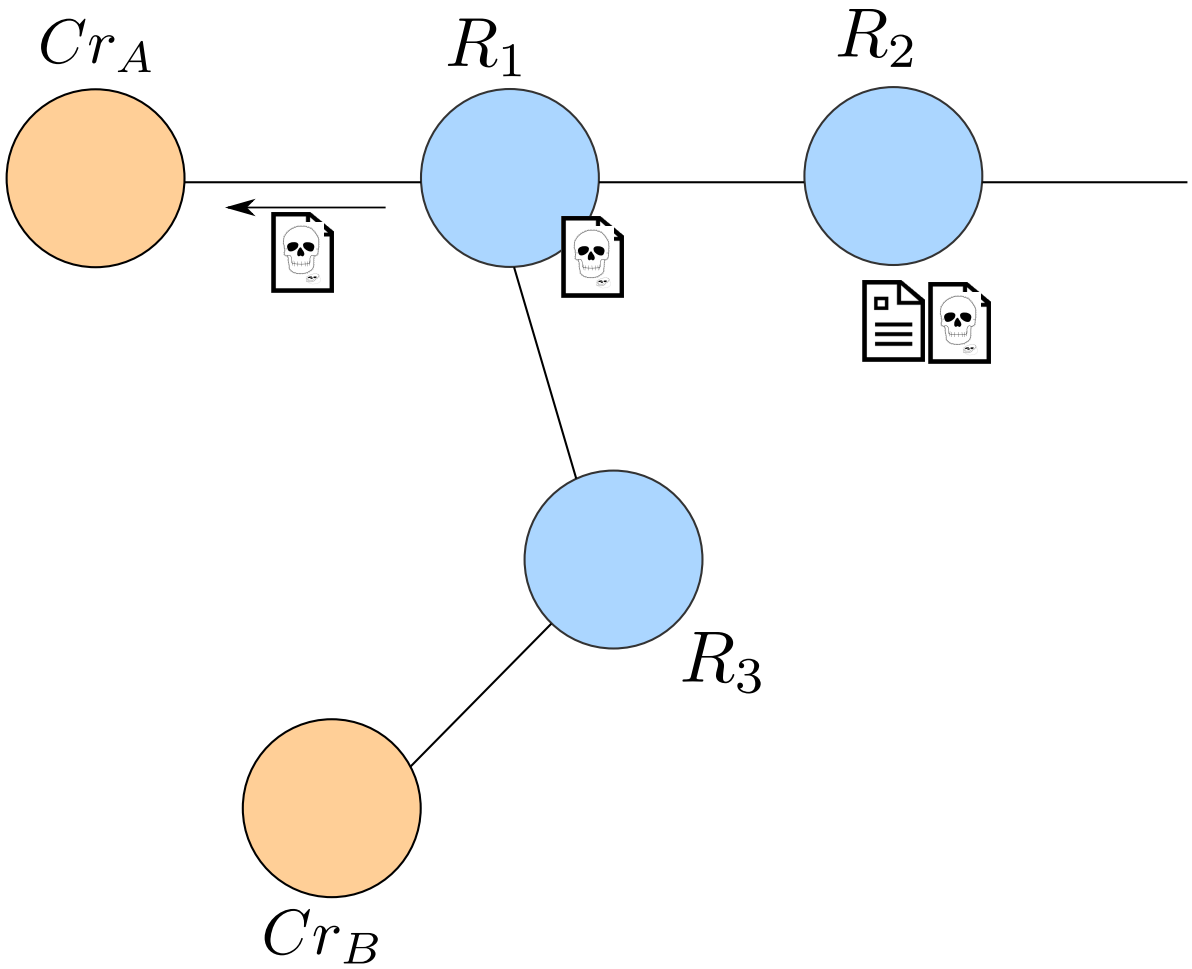
Counter-measures



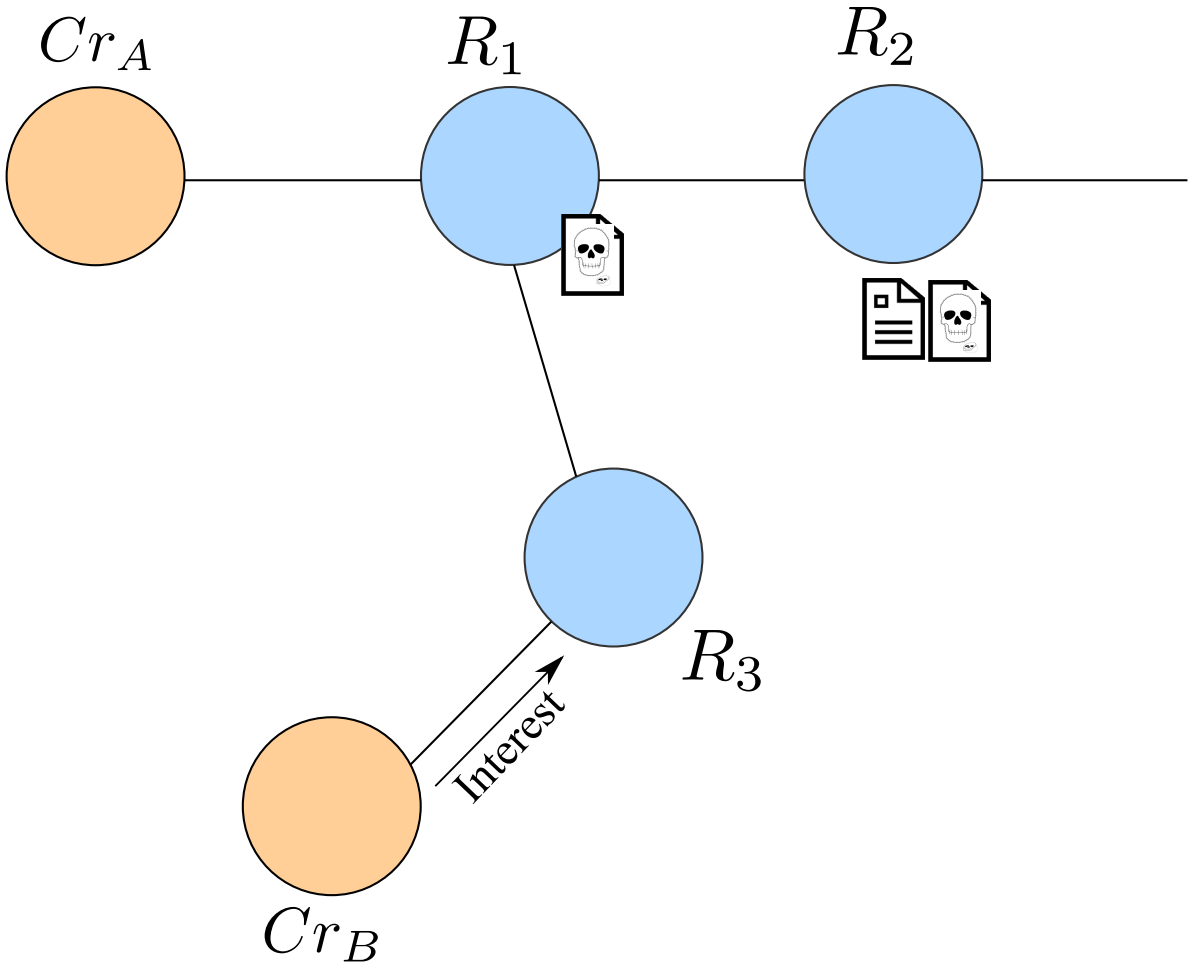
Counter-measures



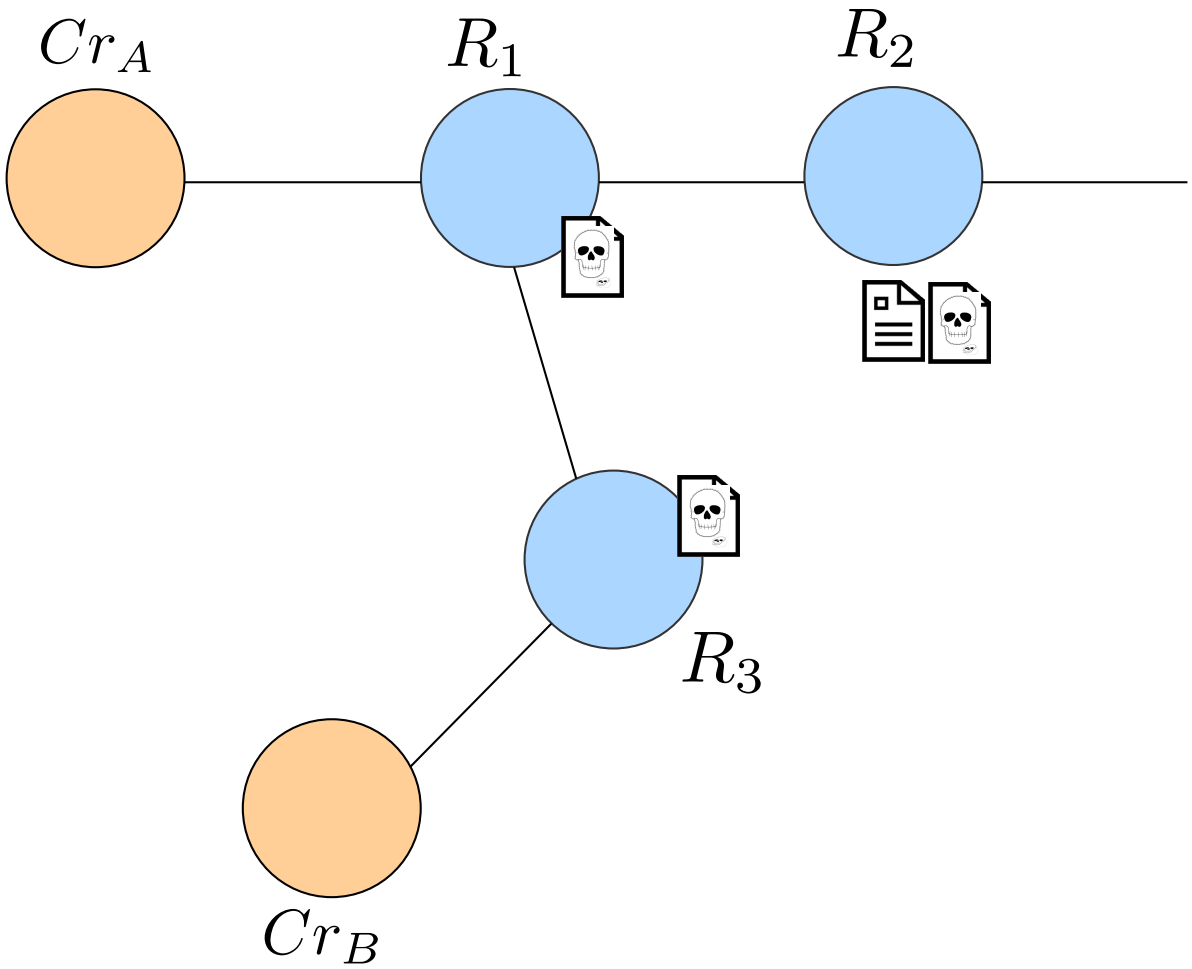
Counter-measures



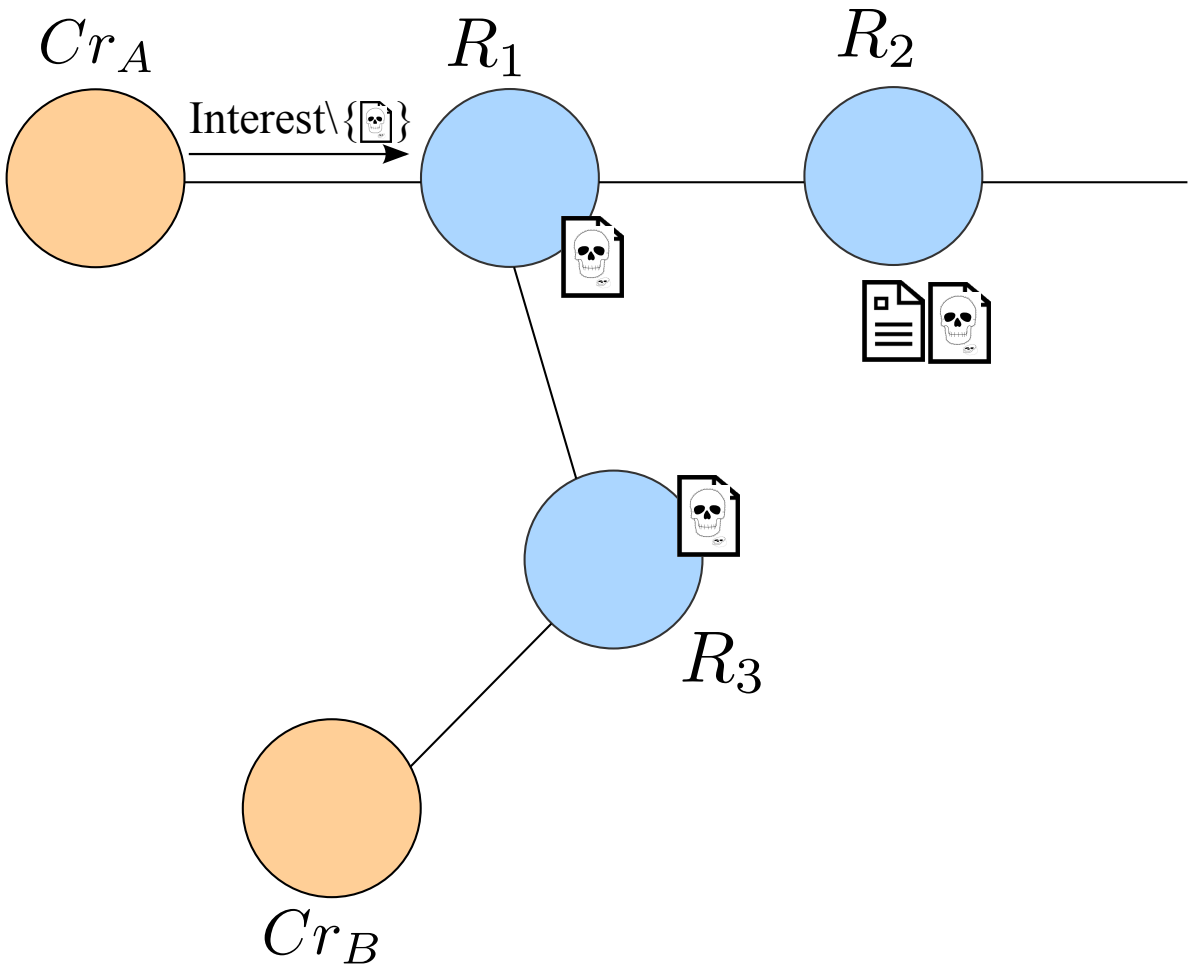
Counter-measures



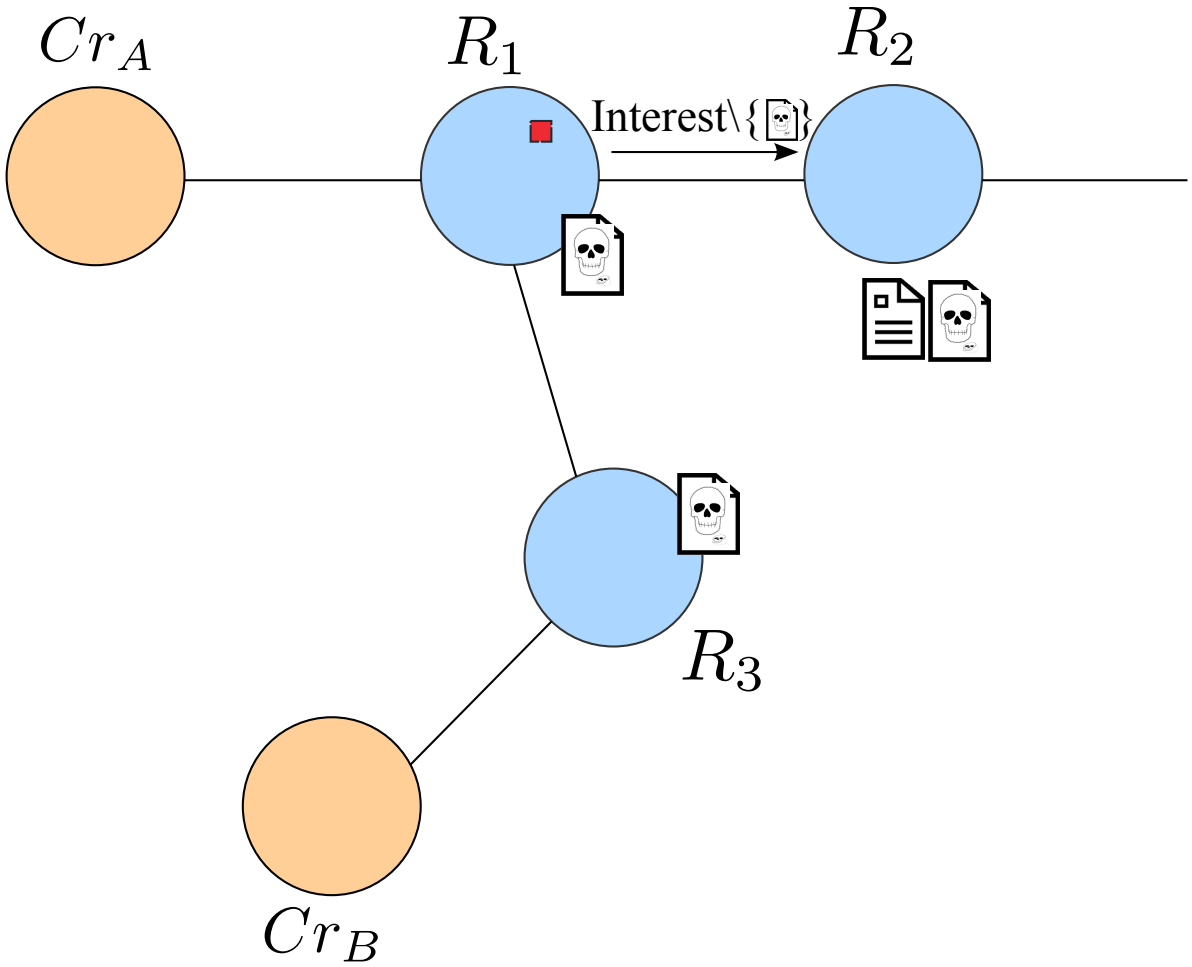
Counter-measures



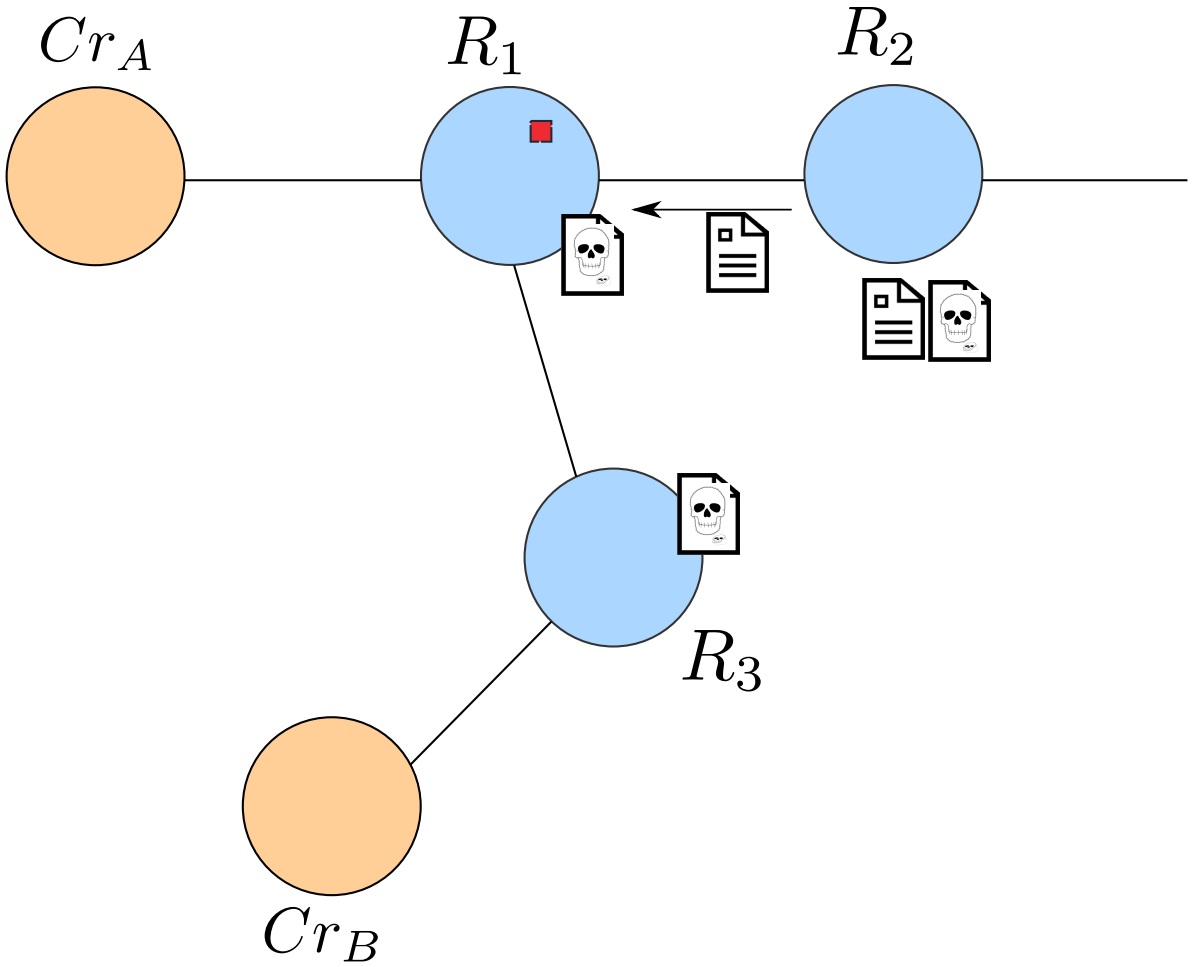
Counter-measures



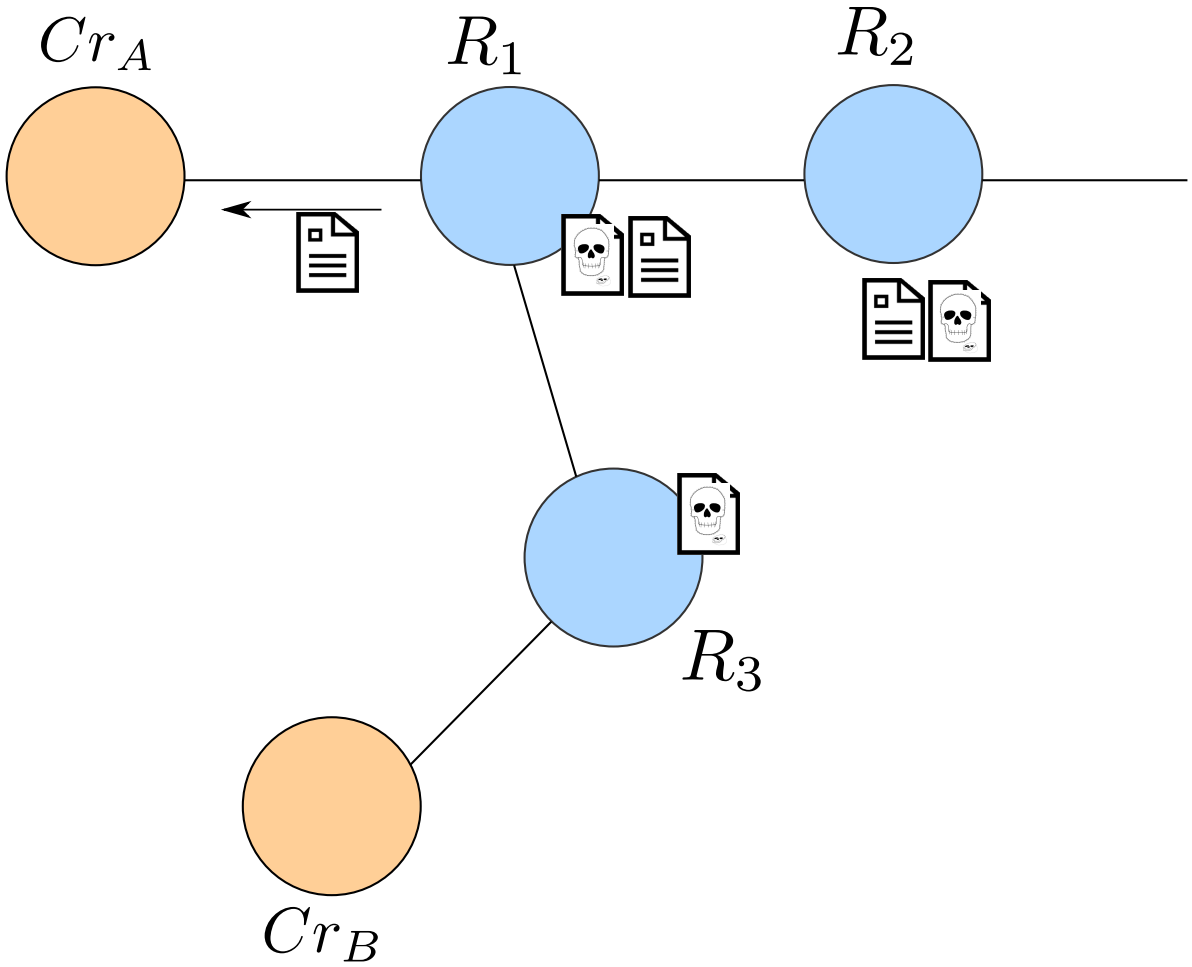
Counter-measures



Counter-measures



Counter-measures



Content Ranking

- Assign a rank to each in-router cached content
- Ranges in $[0, 1]$
- Starts with 1, and decreases with time

- Depends on:
  - Number of exclusions
  - Freshness of exclusion
  - Number of excluding interfaces

Content Ranking

- Assign a rank to each in-router cached content
- Ranges in $[0, 1]$
- Starts with 1, and decreases with time

- Depends on:
  - Number of exclusions
  - Freshness of exclusion
  - Number of excluding interfaces

$$rank = e^{\frac{-t}{f(\#\text{ of exclusions},\ \alpha_0)\ \cdot\ \text{freshness}\ \cdot\ \text{interfaces ratio}}}$$

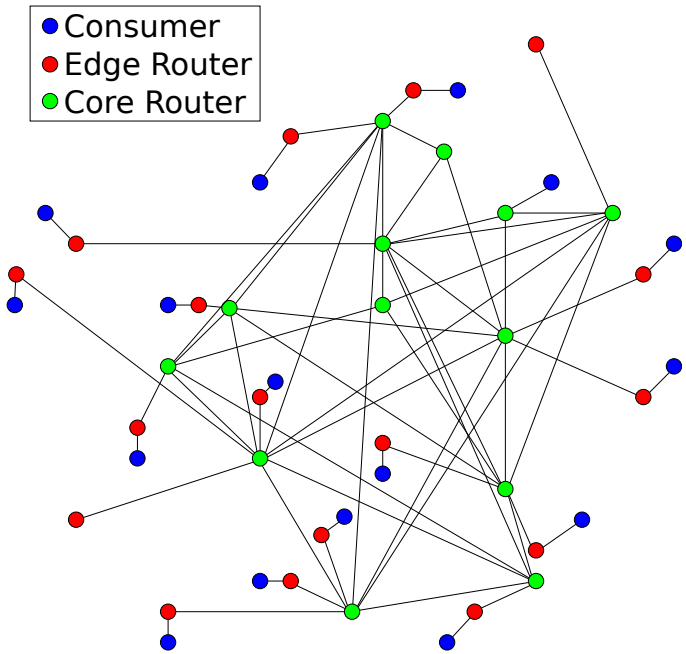
ndnSIM Experiments

- We used ndnSIM to simulate content ranking algorithm

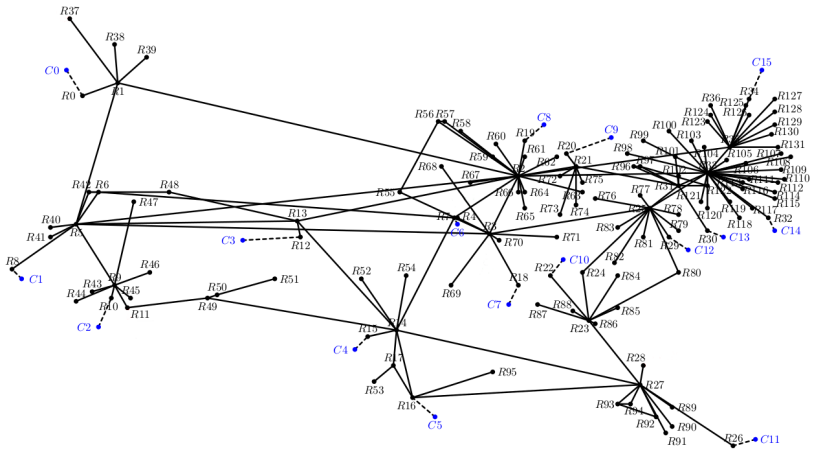
- Experimental setup:
  - Adversary model:
    - Pre-populate router cache
    - Malicious consumers
  - Different rates of pre-populated fake content
  - Different rates of malicious consumers
  - Benign consumers stop after receiving valid content

ndnSIM Experiments – Topologies

DFN

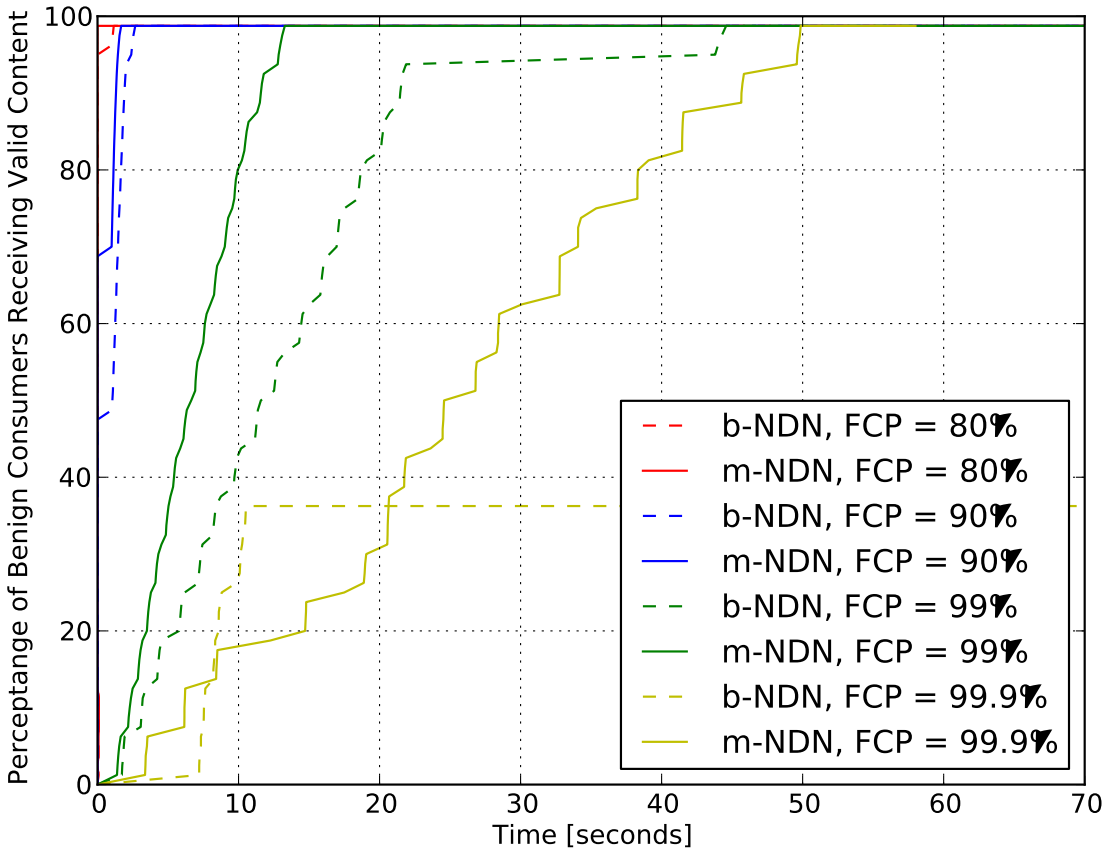


AT&T



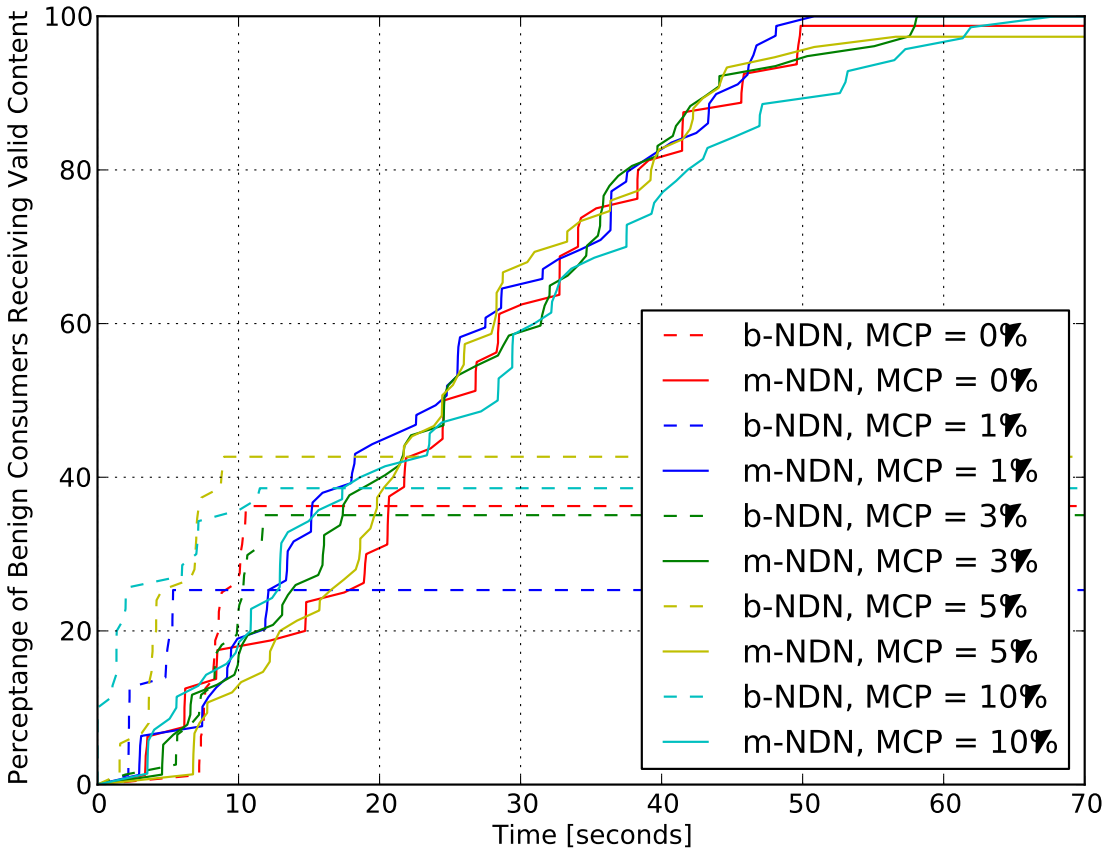
ndnSIM Experiments - DFN

- Different pre-population rate & benign consumers

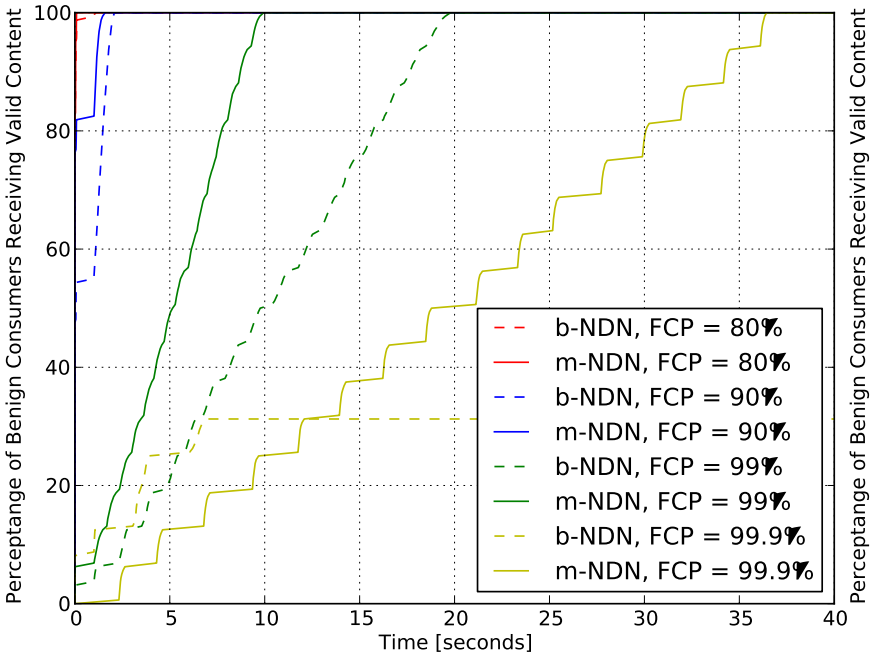


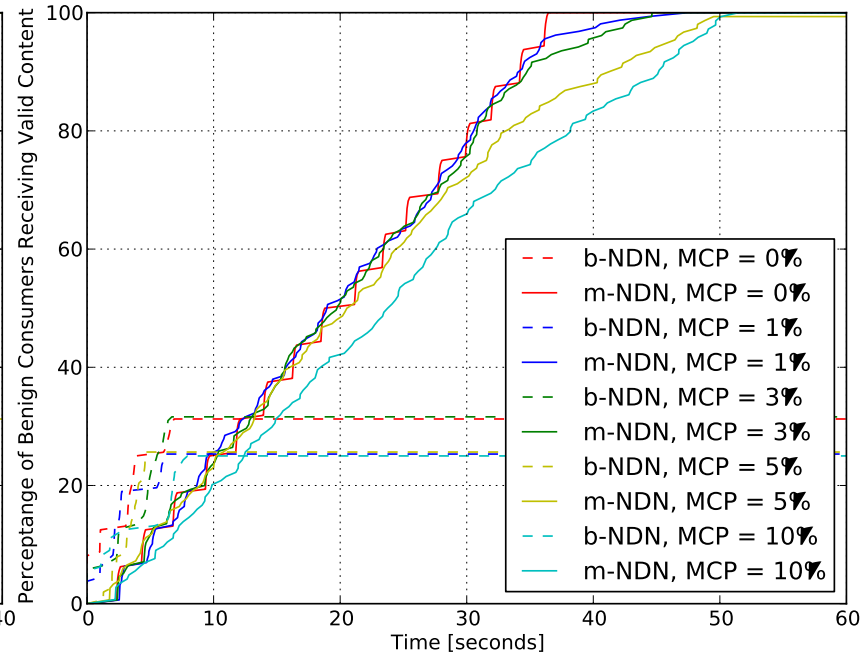
ndnSIM Experiments - DFN

- 99.9% pre-population rate & benign and malicious consumers



ndnSIM Experiments - AT&T





Outline

Conclusion

- Content poisoning is a threat in current NDN design
- Our approach: content ranking is based on observing exclusion patterns
- Encouraging results up to 10% malicious consumers
- Future: ranking algorithm in active adversary model

Thank you!

Questions?

Adversary Model

- **Fake** content object:
  - invalid signature,
  - valid signature generated with the wrong key,
  - or, malformed `Signature` or `KeyLocator` field
- **Valid** content object – verifiable signature generated with correct key
- **Adversary** – NDN entity that can inject fake content
- **Content poisoning** – injects fake content

Content Ranking

A. *Number of Exclusions:*

- The more exclusions the less the weight
- Define
  - $n|H(C)$ – content object
  - $R_{n|H(C)} = E_{n|H(C)}/Q_n$ – exclusion rate
  - r_{to} – rank of $n|H(C)$ when expires
  - α_{to} – makes rank equal to r_{to} when content expires
- Assign higher rank to content excluded less

$$\alpha = \alpha_{to} - \left(R_{n|H(C)} \times \alpha_{to}\right)$$

Content Ranking

B. *Time Distribution of Exclusions:*

- Give more weight to newer exclusions
- Define
  - $i_{n|H(C)}$ – exclusion influence

$$i_{n|H(C)}(t_e) = 1 - e^{\frac{-t_e}{\beta}}$$

  - t_e – time elapsed since last exclusion
  - β – determines influence degradation pattern
  - t_{mw} – time elapsed before minimally weighting $n|H(C)$
- Can calculate β by setting:
  - $t_e = t_{mw}$
  - $i_{n|H(C)} = 1$

Content Ranking

C. *Excluding Interfaces Ratio:*

- Penalize content excluded on multiple interfaces
- Define
  - f_n – # of router interfaces
  - $f_e \in [0, f_n]$ – # of interfaces on which exclusion is received for $n|H(C)$
  - $f_s \in [1, f_n]$ – # of interfaces on which $n|H(C)$ has been served
  - $e_{n|H(C)} \in [0, 1]$ – excluding interfaces ratio

$$e_{n|H(C)} = \begin{cases} \frac{f_s - f_e}{f_s} & \text{if } f_s \geq f_e \\ 1 & \text{otherwise} \end{cases}$$

Content Ranking

- Based on previous definitions

$$rank = e^{\frac{-t}{f(\#\text{ of exclusions},\ \alpha_0)\ \cdot\ \text{freshness}\ \cdot\ \text{interfaces ratio}}}$$

- When content object has never been excluded
  - interfaces ratio $= 1$,
  - freshness $= 1$,
  - and, $\#$ of exclusions $= 0$

$$rank = e^{\frac{-t}{f(\alpha_0)}}$$

Content Ranking

- Based on previous definitions

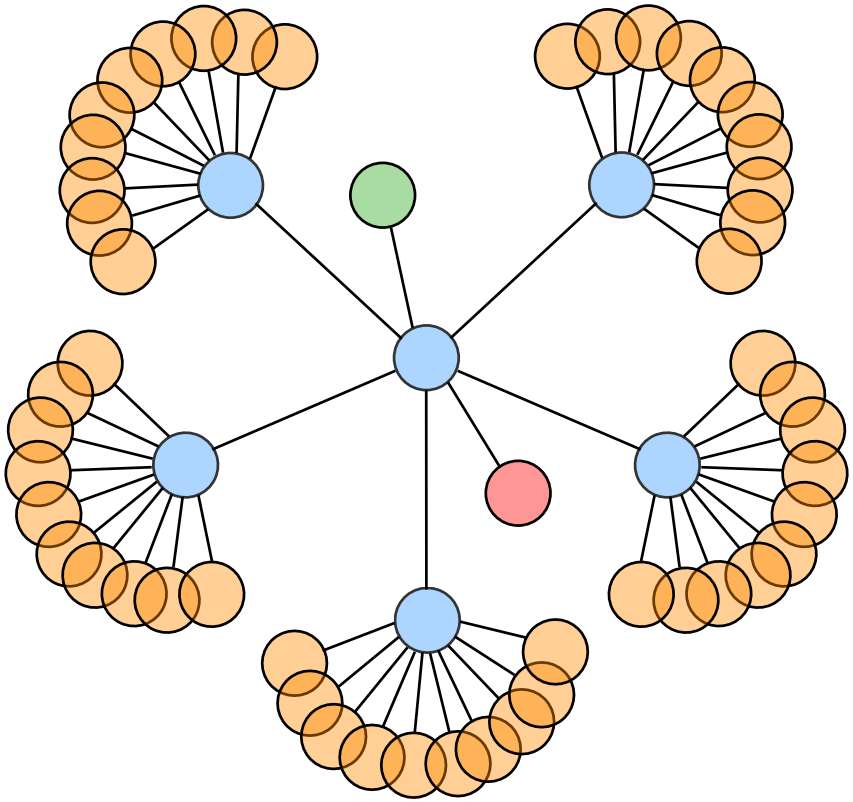
$$r_{n|H(C)}(t) = e^{\frac{-t}{e_{n|H(C)} \times i_{n|H(C)}(t_e) \times \left[\alpha_{to} - \left(R_{n|H(C)} \times \alpha_{to}\right)\right]}}$$

- When $n|H(C)$ has never been excluded
  - $e_{n|H(C)} = 1$,
  - $i_{n|H(C)}(t_e) = 1$,
  - and, $R_{n|H(C)} = 0$

$$r_{n|H(C)}(t) = e^{\frac{-t}{\alpha_{to}}}$$

ndnSIM Experiments

1. *Tree Topology:*



ndnSIM Experiments

1. *Tree Topology:*