ICN Scalability and Deployability

Akbar Rahman

IETF 84, July 29 - Aug. 3 2012

Scope

- Describe a new scalable name resolution and Distributed Hash Table (DHT) based routing approach for Information-Centric Networking (ICN)
  - Get technical feedback from ICNRG WG experts
  - Consider these techniques for potential inclusion in ICNRG WG Survey deliverable

Our Approach

- A name aggregation scheme: the prefix + the *digest of suffixes*
  - *Digest* is generated by Bloom filters
  - Reduce the size and update overhead of name resolution tables
  - Mitigate the suffix hole problem in traditional prefix-based aggregation.
  - Also propose to use *type-length-value* (TLV) coding for names
- Scalable Multi-level Virtual Distributed Hash Table (SMVDHT): A scalable name resolution and routing framework
  - Multi-level virtual DHTs with name aggregation
  - Constructed by exploiting the underlying intra- and inter-domain IP routing protocols
  - Multi-level DHT-based name resolution is an integrated part of routing and forwarding.
  - Improve scalability and deployability

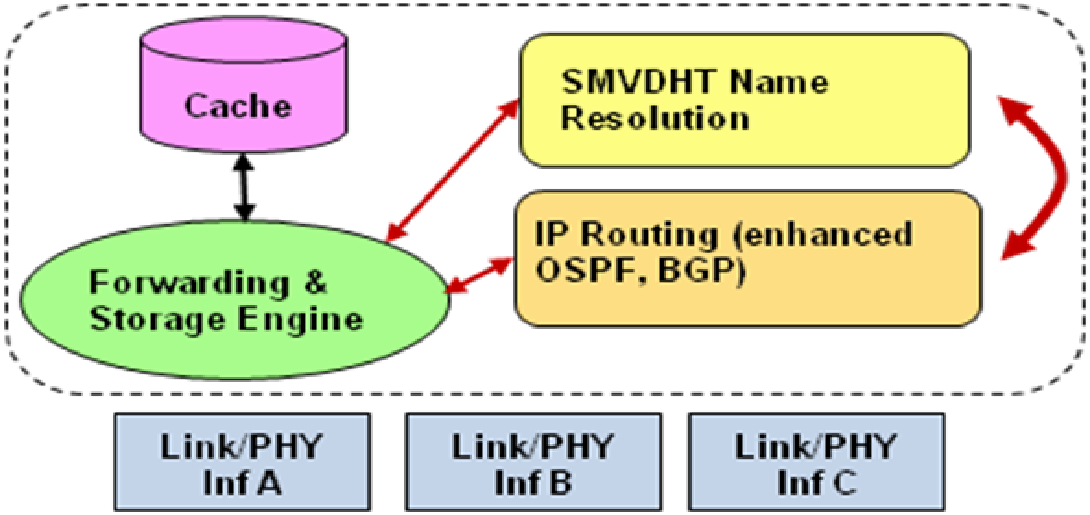
Proposed Aggregation Scheme (1)

- *Prefix+Digest* Aggregation: publish the prefix and the digest of suffixes [3], [4]
  - Bloom filters are used to generate the digest from the suffixes of the aggregated object names.
- A Content Router (CR) announces
  - *"the content objects with this prefix and digest value of the suffixes can be reached via me."*
    - more accurate information
  - other CRs only need to maintain one routing state per announced prefix
- Can be used for flat and hierarchical names

Proposed Aggregation Scheme(2)

- Flat names: *N* objects, *P:L* $_i$ *(i=1, 2, …N)*
  - ⇒ *Advertise a summary name (sOID),* ***P:digest(L)****,*
    digest(L) = Bloom-Filter{L_1*, …*L_N*}.*
- Hierarchical names: N objects with /example.com/movies/titles/segmentations
  - ***/example.com/movies/digest(titles/segmentations)***
  - ***/example.com/digest(categories/titles/segmentations)***
- Control aggregation degree based on the content object popularity or the distance to the content location
  - Balance between needed resources and routing information compression.
  - Adjust the prefix size, i.e. the number of non-aggregated elements or
  - Control # of aggregated elements that are added to a digest
- Relieve the suffix hole problem
  - while reducing the size and update overhead of name routing tables

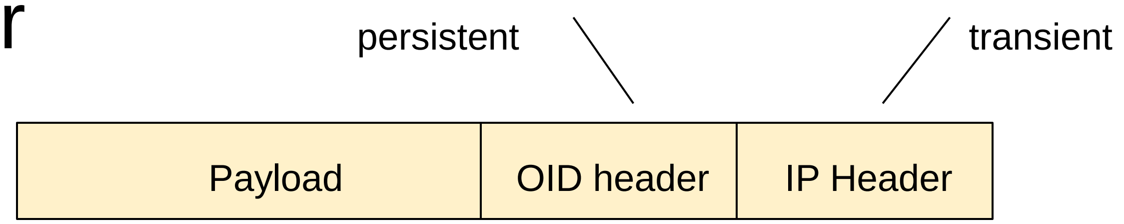
SMVDHT (1)



SMVDHT router model

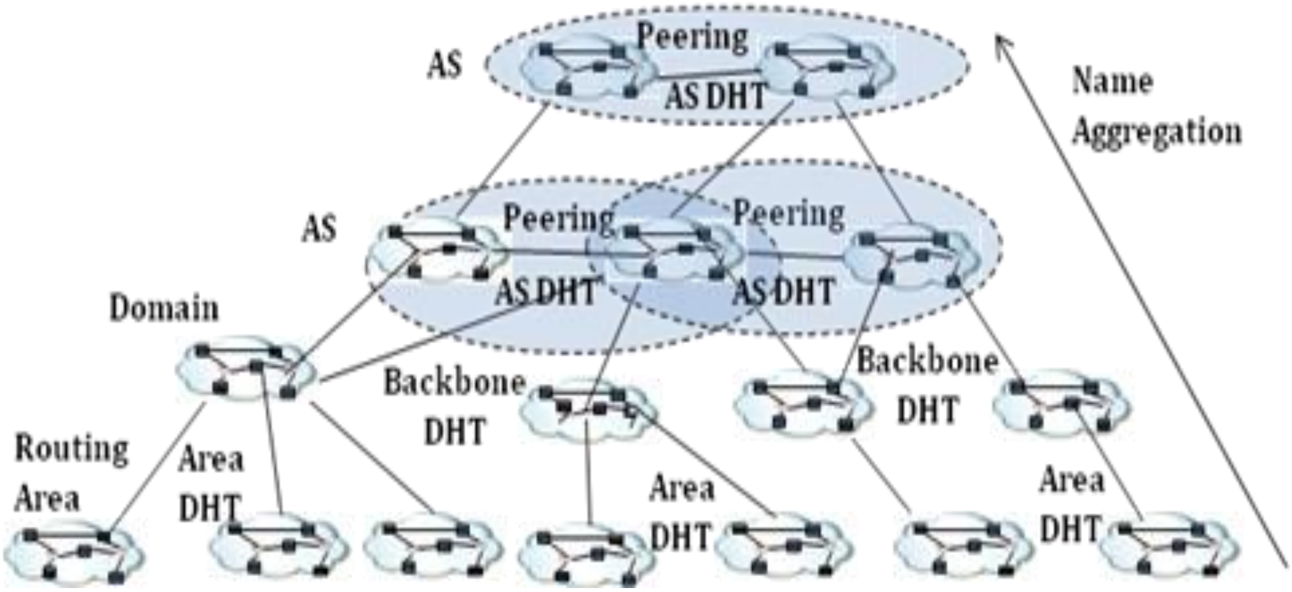
- A CR runs both IP routing and SMVDHT name resolution protocols,
- ICN services co-exist with other IP services such as traditional host-to-host communications.

- A name resolution layer on top of the IP layer



SMVDHT (2)

- Multi-level virtual DHTs mapped to the Internet hierarchy
  - No change to the current Internet hierarchy infrastructure as well as the relationship between enterprise domains and ISPs => simplifies deployment.



- Aggregation based on the content popularity and VDHT level.

- Conventional OSPF and BGP are used for IP routing with certain extensions, e.g. a router can advertise its name resolution capability in its IP routing dissemination.
- A host or a normal IP router can connect to an SMVDHT router as a client.

References

[1] "A Data-Oriented (and Beyond) Network Architecture," T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, I. Stoica, ACM SIGCOMM '07.

[2] "Networking Named Content," V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, CoNEXT '09.

[3] "A Multi-Level DHT Routing Framework with Aggregation," H. Liu, X. De Foy, D. Zhang, ACM SIGCOMM ICN'12.

[4] "A TLV-Structured Data Naming Scheme for Content-Oriented Networking," H. Liu, D. Zhang, IEEE FutureNet V, June, 2012.

[5] "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," I. Stoica, R. Morris, D. Karger, M. Kaashoek, H. Balakrishnan, ACM SIGCOMM 2001.

[6] "Canon in G Major: Designing DHTs with Hierarchical Structure," P. Ganesan, K. Gummadi, H. Garcia-Molina, IEEE ICDCS, March 2004.

[7] http://www.fp7-pursuit.eu/PursuitWeb/.

[8] http://www.psirp.org/.

[9] http://www.netinf.org/home/home/.

[10] Scalable and Adaptive Internet Solutions (SAIL). http://www.sail-project.eu/.

Backup Slides

Introduction

- Information-Centric Networking (ICN)
  - decouples identity from location at the networking level
  - retrieves an information object by its name (identifier), not by its storage location (IP address)
- Address limitations and inefficiency of IP networks
  - Content distribution, Mobility, Multi-homing, etc.
- Challenges:
  - Scalability
    - At least handle 10^{12} objects even based on the current web size
    - Increase by several orders of magnitudes considering sensor data, vehicular, Internet of things
    - Dynamic locations due to caching
  - Deployability
    - IP networks would not go away
    - ICN => next-gen CDN?
      - Built-in storage and computing power in network elements
      - No need for dedicated cache servers, proxies

Naming

- Naming scheme is critical in ICN
  - Used to identify, discover and retrieve content
  - Affect routing, scalability, and content security
- A content object name or ID (OID)
  - Uniqueness
    - globally unique to identify an object
  - Persistence
    - Independent of location and administrative domain
    - remain valid as long as the underlying object itself is available and not changed.
  - Trustworthiness
    - Secure the content rather than the communication path
    - End users and network elements should be able to authenticate the content
    - Binding between the user-friendly name and its corresponding ICN OID, and binding between the OID and content data
  - Scalability
    - Certain name aggregation

Existing Naming Schemes

- Flat OID [1]
  - P:L
    - P = hash of a public key of content owner/naming authority
    - L= a flat label or hash of the content data
  - Self-certifying
  - Difficult to aggregate
  - Need an external mechanism to map between user-friendly name and flat OID.
- Hierarchical OID [2]
  - Similar to binary encoded URL
    - /example.com/movies/title/format/segmentation
    - mapping between user-friendly name and corresponding ICN OID
    - But need mapping between OID and data (public key to authenticate data)
  - **Aggregation to improve scalability**

Prefix-Based Aggregation

- Hierarchical names with prefix-based aggregation (CCN) [2]
  - aggregate routing entries for the hierarchical OIDs with a common prefix
    - /example.com/movies/…./….
  - If all the content objects with a prefix of "/example.com/movies" stored in a node,
    - a single route announcement
    - maintain a single routing state for these objects in content routers
- Some issues with prefix-based aggregation
  - A caching node or content router (CR) may not have all the content objects with a given prefix.
    - If the prefix-based aggregation is used to reduce routing states and update overhead, a lot of information will be lost. Router has to advertise:
      - ***"some of the content objects with this prefix (/example.com/movies) may be reached via me."***

      ⇒ **suffix hole**

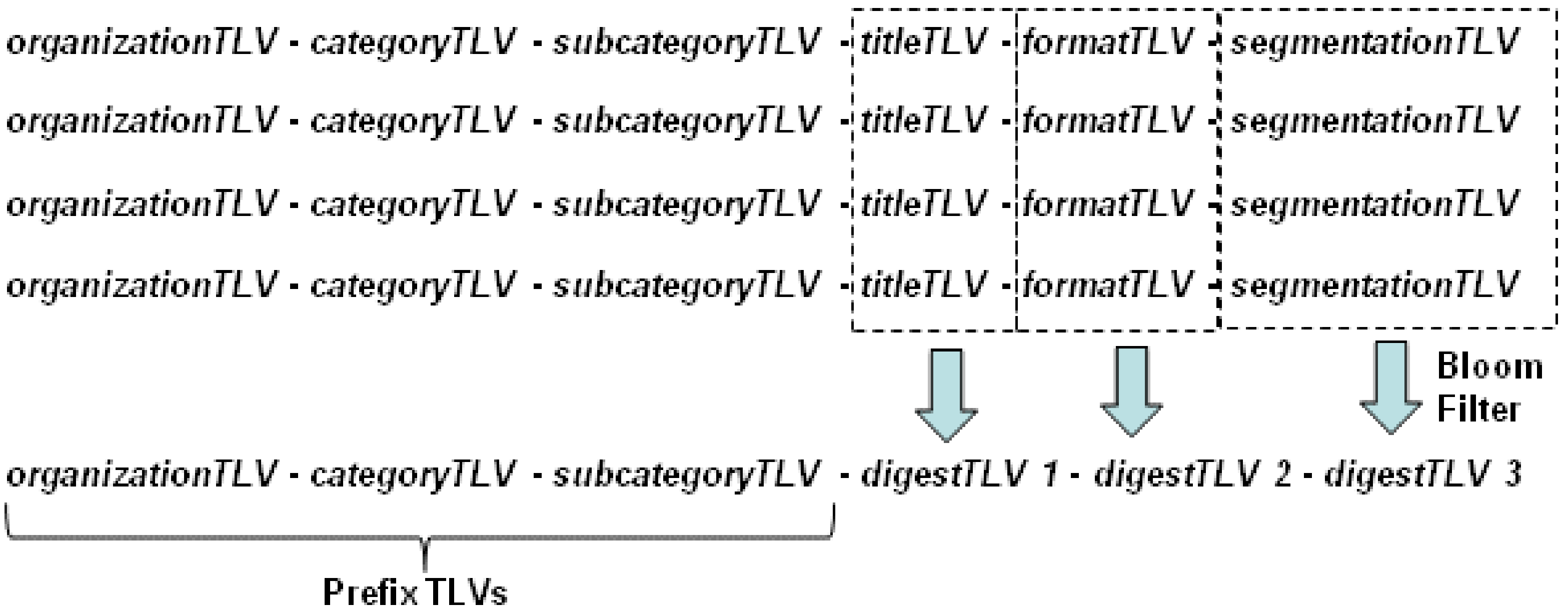
      ⇒ uncertainty in locating a particular content object and reduce routing efficiency
  - How to aggregate non-structured flat names?

TLV-Structured Naming Scheme

- Type-Length-Value (TLV) Encoded OID [4]
  - consist of a set of variable-size information elements (IE), each IE encoded as a TLV
    - E.g. ***organizationTLV-categoryTLV-titleTLV-segmentationTLV***
  - The network imposes no restrictions to the OID assignment except the TLV structure
  - CRs do not have to know the meaning of types and values except certain *"well-known"* types.
    - use the length field to parse the TLV elements and treats the whole element as a binary number in publishing and routing
- Why:
  - Flexibility: Hierarchical or peer relationships
  - Can define **"well-known"** types, e.g. digest TLVs, signature TLVs
  - Extensibility: sub-TLVs
  - URLs or DNS names have their traditional semantics, somehow related to the location, e.g. example.com/video/WidgetA.mpg
    - the name becomes misleading if the administrative domain or location of the object is changed.

Aggregation

- To address suffix hole problem, use prefix TLVs + digest TLVs
  - apply Bloom filters on each column of aggregated TLV elements to generate digest elements.
  - a routing advertisement can express more accurate information



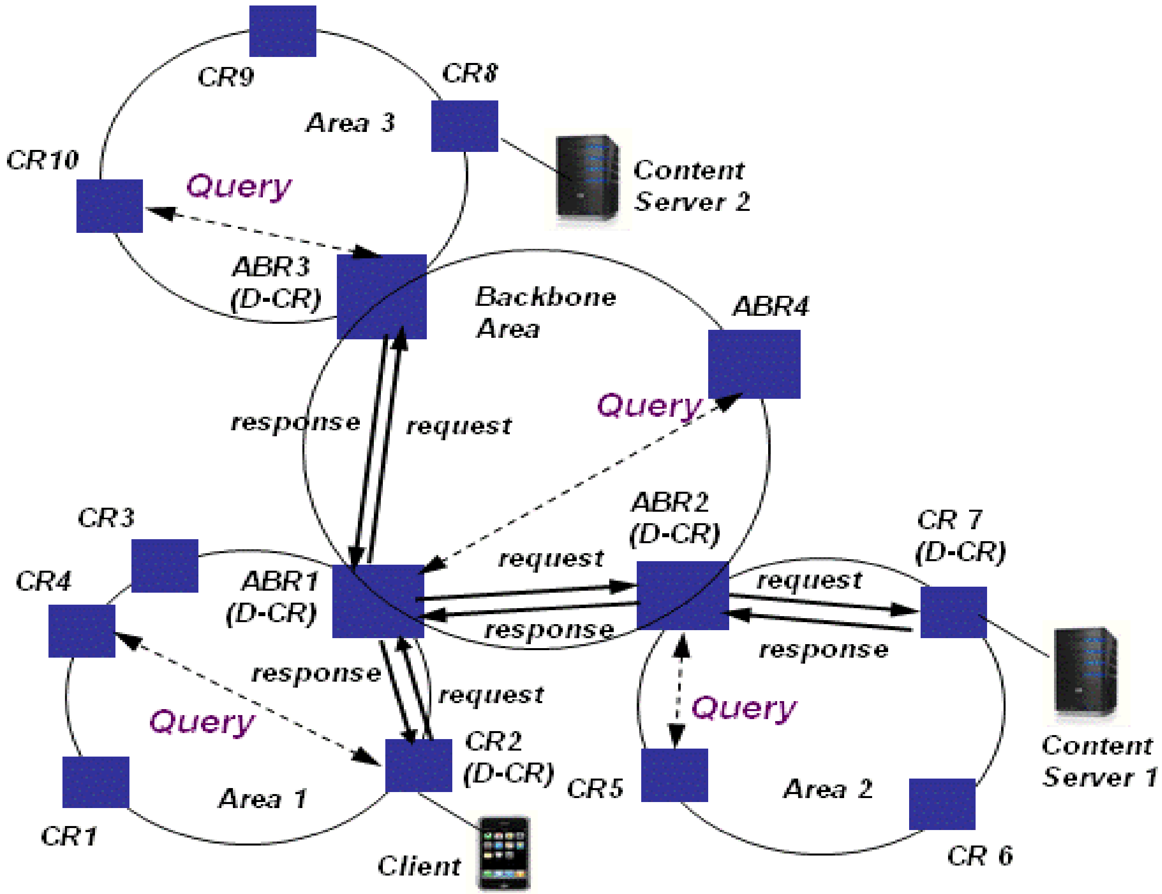
Type	Length	# of elements summarized	Digest

Integration of Name Resolution and IP Routing

- Content Publishing Techniques: flooding or DHT
  - Each has its pros and cons.
- ICN is different from P2P
  - Infrastructure nodes relatively stable, but content locations change frequently due to cache replacements
- IP routing (flooding) provides infrastructure topology
  - help design more scalable and efficient name resolution mechanisms and improve deployability
- Build scalable multi-level virtual one-hop DHTs (SMVDHT) using IP routing [3]
  - Simplified network management and more efficient than conventional hierarchical DHTs such as Chord [5] and Canon [6]
    - No need for DHT bootstrapping and maintenance (IP does the job)
  - Corresponds to the Internet hierarchy and optimize forwarding path
- Name resolution is an integral part of routing and forwarding

Name Resolution and Routing Procedures

- Integrated name resolution and routing protocol: ***delegated CRs*** perform local look-up and forwarding decision in hop-by-hop



- a content request is forwarded to the best or closest host(s) of the requested object by a set of delegated CRs.
- A response carrying the content data or an instruction to establish the content retrieval session is forwarded back to the requester along the same shortest path as the request travels
  - en-route caching can be performed by intermediate CRs.

Summary

- Scalability and deployability are critical
- Need to work/integarte with IP, not to replace IP
- Better aggregation:
  - prefix + digest of suffix
- SMVDHT:
  - exploit IP routing for efficient and scalable name resolution

Some More Details

CR Forwarding Process (1)

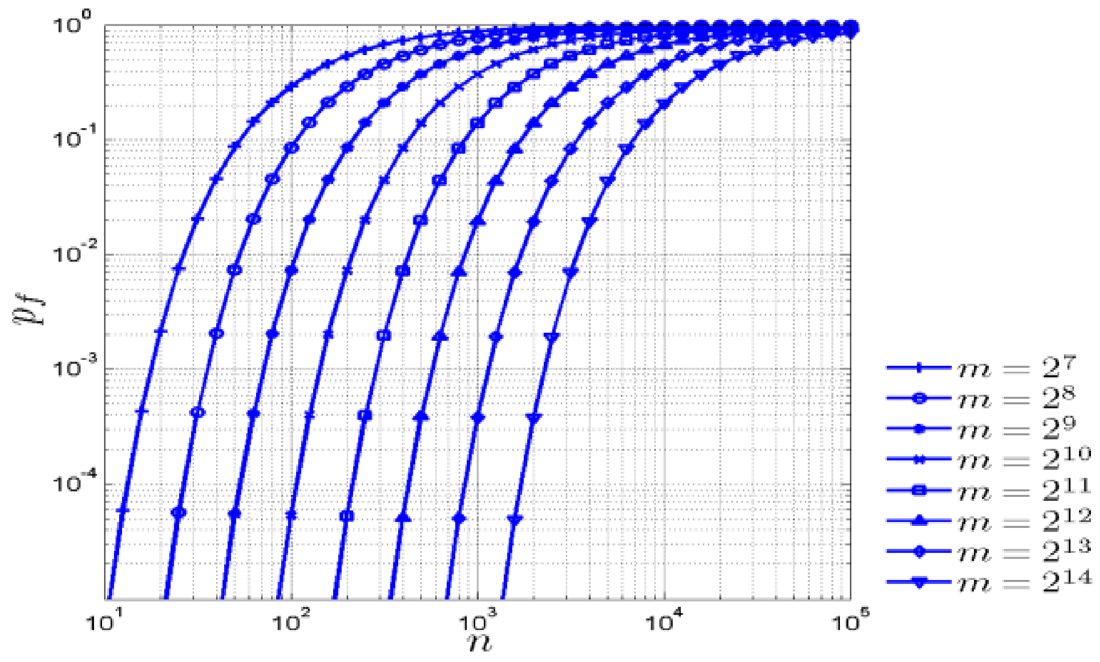
- To resolve a request to a publishing node, there should be an entry match between the requested OID and a published sOID,
  - the corresponding prefix should be exactly the same
  - the digest in the sOID should give a positive match to indicate that the corresponding suffix element in the requested OID is likely to be present.
- With Bloom filters, false positives are possible, but false negatives are not.
  - the error probability can be controlled
    - designing appropriate filters
    - limiting the number of elements that are added to a digest.

CR Forwarding Process (2)

- Given a filter, a publishing CR can flexibly control the aggregation degree based on the popularity of the content objects or the distance to the content location
  - e.g. no aggregation performed for the content objects residing in the local network domain.
  - a domain gateway router publishes the summary OIDs of its content objects to outside domains.
  - the number of suffix elements added to a digest can be limited to control the error probability.
  - When the number of elements exceeds the limit, the elements are divided into groups.
    - Each group generates a digest.
- Balance between the network resources needed for maintaining routing states and the false positive probability.
- Mitigate the suffix-hole problem while achieving good routing scalability
- One learns from the sOID that the requests for the content objects with this prefix and digest may be served by this domain.

Bloom Filter Performance

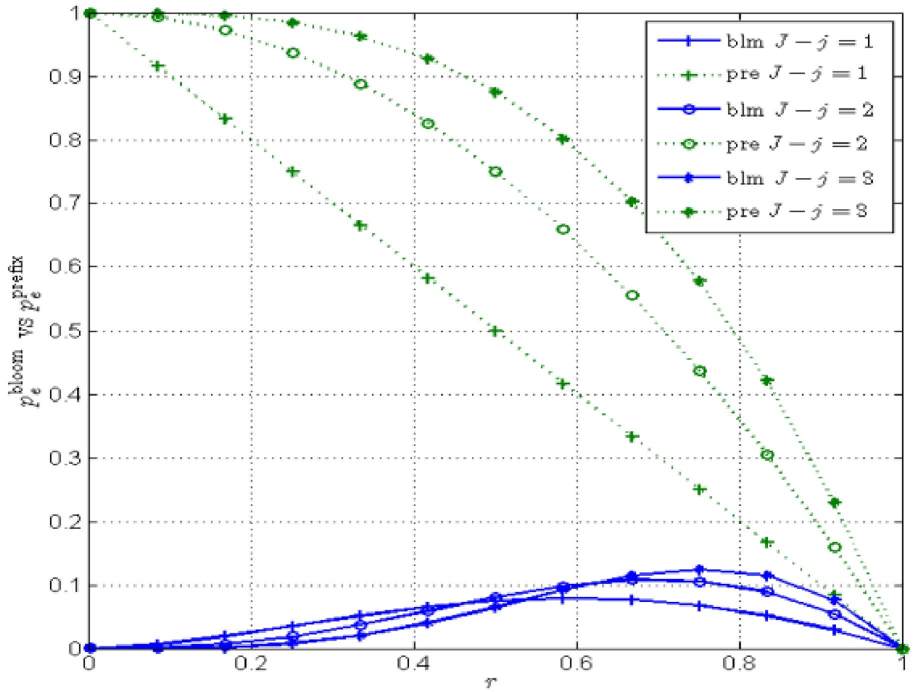
- False positive probability p_f as a function of the number of aggregated elements n in the digest and the filter size m, assuming an optimal value of k is used.



- Design Bloom filters to meet the requirements

Bloom-Filter Aggregation vs. Prefix-Based Aggregation

- Routing resolution error probabilities of Bloom-filter and prefix-based aggregations



- Bloom filter aggregation greatly outperforms the conventional prefix-based aggregation

- n = # of elements added to a digest
- u = total # of potential elements
- r = n/u