CATT: Cache Aware Target idenTification for ICN

Authored by,
Suyong Eum, Kiyohide Nakauchi, Yozo Shoji, Nozomu Nishinaga, Masayuki Murata

ICN outline

- It is estimated that by 2015, 90% of the traffic in the Internet will be multimedia and P2P contents
- Need for shift from location centric to information centric approach in the current architecture
- Information centric networking – emphasis on ‘name’ of the data rather than the location

Problem Statement

To efficiently distribute and locate contents in the ICN environment

Challenges

- To locate distributed contents considering Volatile behaviour of copies in caches
- Where to cache contents in the network

Proposed solution : CATT

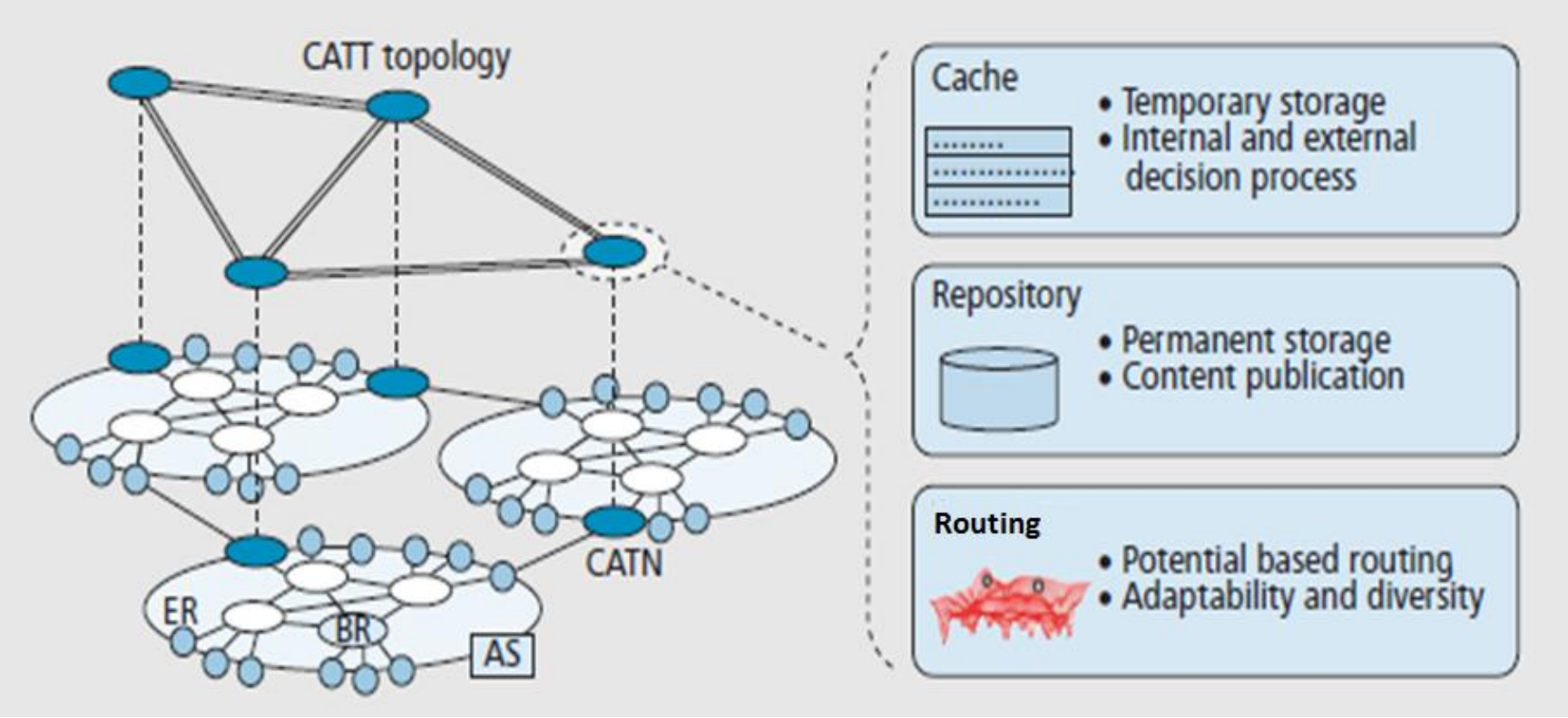
Cache Aware Target idenTification

- Based on the common route-by-name technique employed in most ICN
- '**Cache Aware**' – Identifies the best candidate source for distribution and retrieval of data
- Efficient distribution of content among caches (topology aware caching)
- Retrieval from both the original source and caching points (Potential Based Routing)
- Flat file naming scheme as in DONA

CATT Design Goals

- **Availability**: All distributed copies contribute to the retrieval process of a requested content
- **Adaptability**: Volatile behaviour of cached contents needs to be taken account in distributing and retrieving of contents
- **Diversity**: User request for a content not only based on proximity but also the qualities of the content or network condition.
- **Robustness**: To avoid single point of failure

Architecture



Architecture (contd..)

- CATN – CATT node
- AS – Autonomous system
- CATNs strategically placed at edge of each AS
- CATN – cache, repository and routing
  - Cache – selective caching done internally (what to cache) / externally (where to cache)
  - Repository – permanent storage for content published
  - Routing – based on content file quality (PBR)

Potential Based Routing

- Potential : scalar value associated with individual network elements
- Routing of query or traffic based on potential value calculated using multiple factors
- Factors include proximity, quality and volatility of the content
- State and capacity of the caches are also considered

Calculating Potential Value

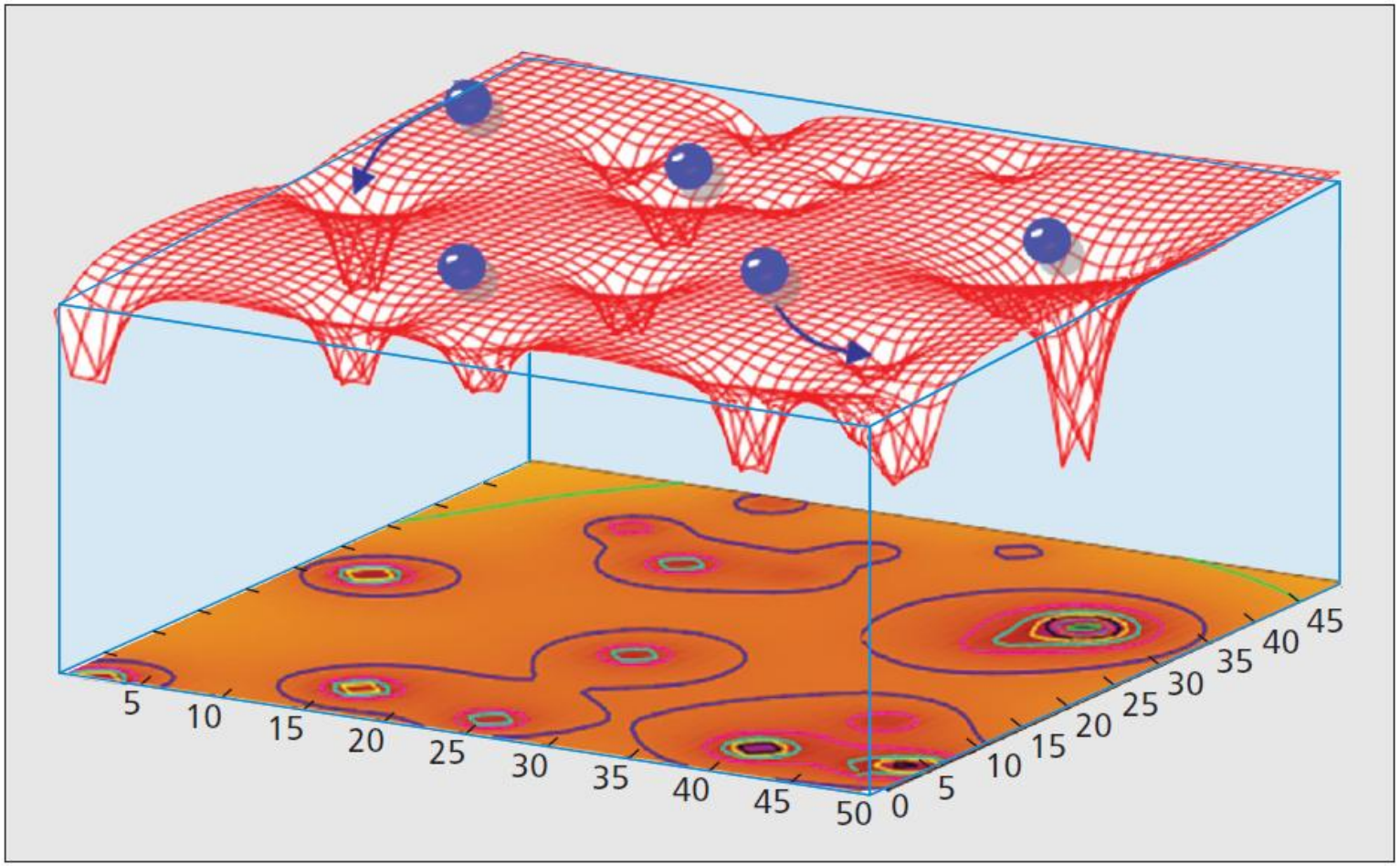
- Define an initial potential value at the provider node containing the file
- Flood this value using an advertising packet containing a field for hop count
- The hop count increases as the packet travels away from the provider node
- Each node which receives the advertising packet calculates its own version of the content file's potential value based on hop count, link costs, geographical distance, etc.

Routing information

- Each node will create a routing entry for each content file

- Routing entry – 3 fields
  - **Content Id** : content file Identification
  - **Potential value** : Potential value of the content from the node's perspective
  - **Next hop** : hop to the neighbor with the lowest potential

Potential field – ball and valley analogy

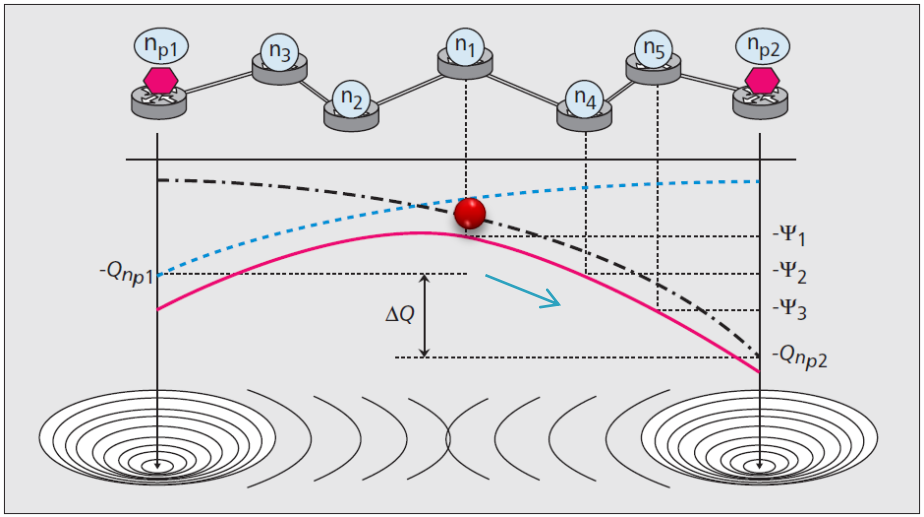


PBR : how to create a potential field?

-Potential values of np1, np2 are defined as np1 and np2.

-Assuming the potential values are increased as they travels along its neighbor nodes

-The solid line represents the sum of both potential values which individual nodes n1, n2, n3, n4, n5 use for routing a client request.



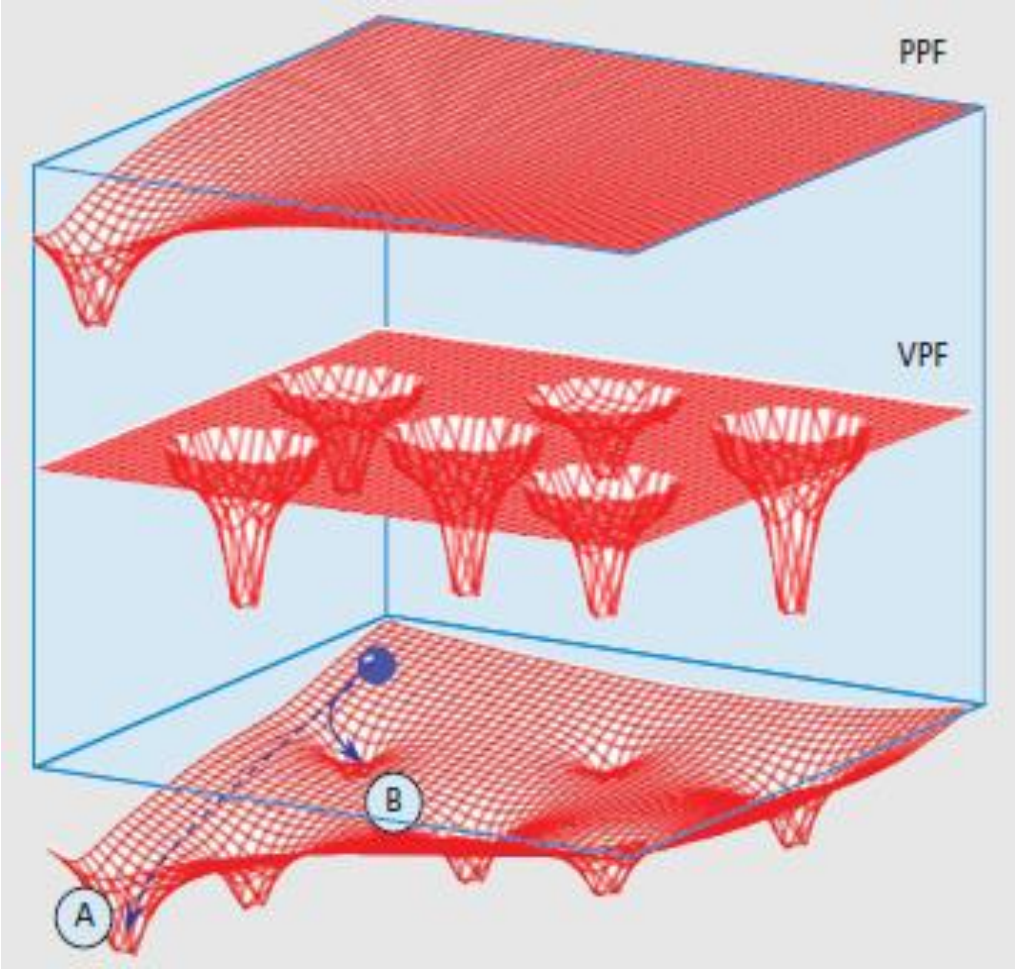
Topology Aware Caching

- External caching strategy
- On-path caching mechanism
- Content file is cached along the downloading path at certain nodes decided by the content provider
- Only the request message is routed and the response message is forwarded along the trail left by the request message

PBR as main routing algorithm

- Two different potential fields
  - Permanent potential field
  - Volatile potential field
- PPF – preferred for static content files (repository), not many updates to the potential value
- VPF – preferred for dynamic or volatile content files, regularly replaced by newly arriving copies which redefines the potential value frequently
- Linear combination of the two methods is the most preferred

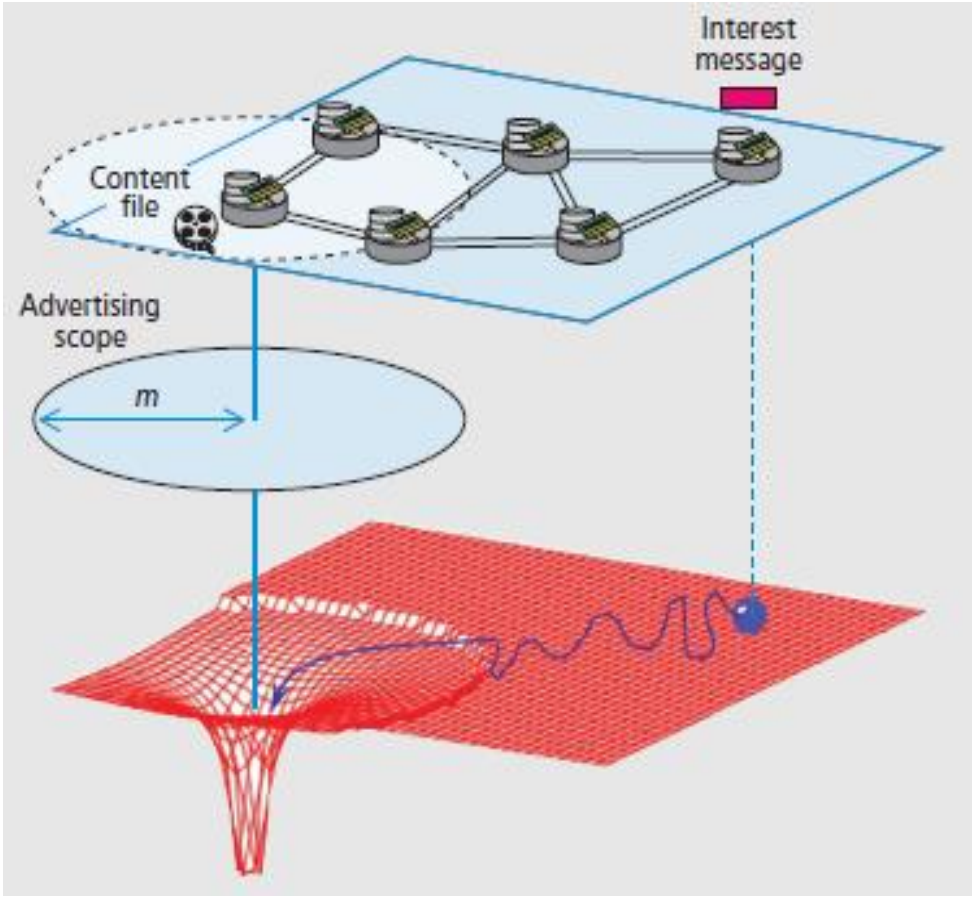
PBR as main routing algorithm



PBR as subsidiary best effort routing algorithm

- Overcomes the storage requirement drawback when using PBR as main routing algorithm
- Request originally routed towards the main repository of the content file
- Caching nodes which possess copies of the content file flood an advertisement message within a **limited scope**
- Potential values are created at these caching nodes(within the scope) which attract user requests
- Similar to breadcrumb routing (but active advertisement)

PBR as subsidiary best effort routing algorithm

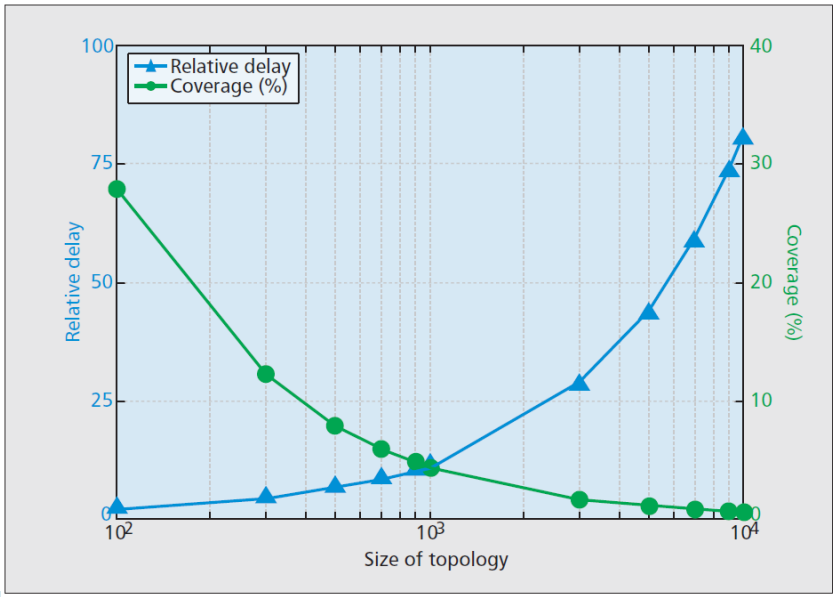


Performance Evaluation

- Event driven simulator was developed for evaluation of CATT
- Dijkstra algorithm was used to compare PBR and shortest path routing (OSPF)
- Various sizes of Autonomous System level topologies used for simulation

PBR with a Random Walk in Various-Scale Topologies:

- **Metrics:**
  - Relative Delay: delay caused by random walk
  - Coverage: size of expected routing table
- **Results:**
  - As size of topology increases user tends to experience more delay since potential field is defined in relatively small area



Topology Aware Caching with PBR

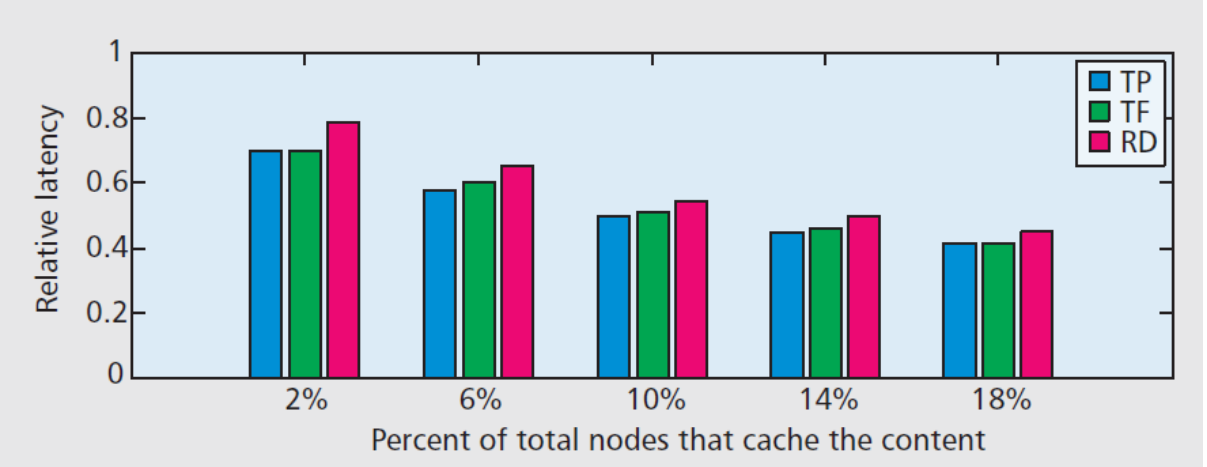
Simulation :

• one content file is initially published in the center of the network based on
its node betweeness-centrality calculation.

• the content file is cached on another point of the network using the TP method whenever a query to the content file is made from a randomly selected user and hits either the original content file or its

Relative Delay

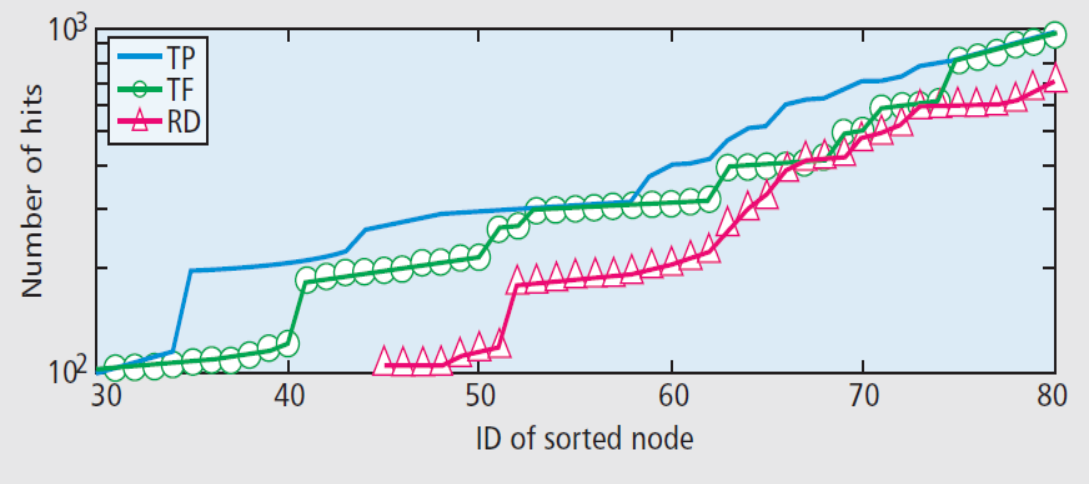
The average latency that users experience compared to the case where there is not a copy except the original content file.



•One content file is cached on 10% of the total nodes using the three caching policies.
•Some caching nodes are expected to serve users' requests more frequently than the others.
•**Load balancing**: users' requests are desired to be distributed evenly in an ideal scenario.

Cache Hits

The result demonstrates that TP seems to be preferable to TF and RD.



Limitations

- Delay: introduced by random walk algortihms
- Scalability: limited scope of the potential field
- Complexity: expanding the network to include more AS, then external decision making process becomes harder
- Overhead Traffic: Flooding of advertising packets among neighbours, especially in VPF

Conclusion

- CATT is founded on the Potential Based Routing (PBR) and the topology aware caching policy.
- CATT architecture achieved several design goals
  - **Availability** is achieved by incorporating an original content file published in the repository and all copies in caches into the retrieval process.
  - **Adaptability:** it takes into account copies in caches that tend to have a high volatile behavior due to its replacement for cache management.
  - PBR provides a mechanism to select a content file based on proximity and also on the quality of the content, which makes the selection process rather **diverse**.
  - Fully distributed algorithm, it is robust against a **single point of failure** scenario.
- TP is more preferable in terms of implementation perspective