Opportunistic Context Based Networking in Future Internet

August 2008

Jisun An
Information and Communications University
Computer Distributed Systems and Networks

Table of Contents

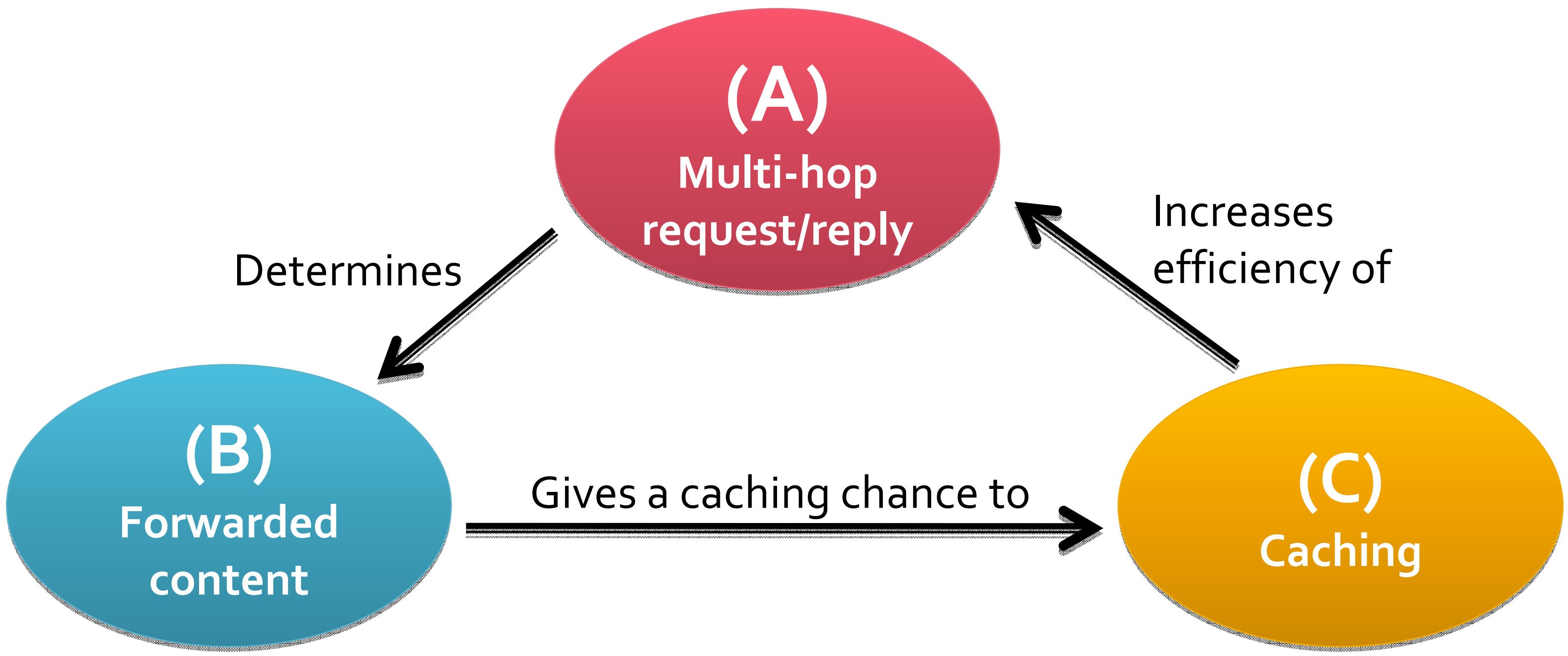
Mobile Social Software

- Mobile Social Software
  - To create and maintain social relationship
  - Common features
    - Request/reply
    - Content sharing
  - Existing works …
    - Restriction on communication range
  - Multi-hop comm. is necessary

Motivation

- Content sharing
  - Creating group based on shared interest
- Why people share content?
  - Sharing experience
- What content will be shared?
  - Interesting content
  - Popular content
- Possibility for predicting which content will be requested by whom?
  - Share content → share interest
  - Share interest → share content
  - Using application-level information, such as social-relation

Relationships



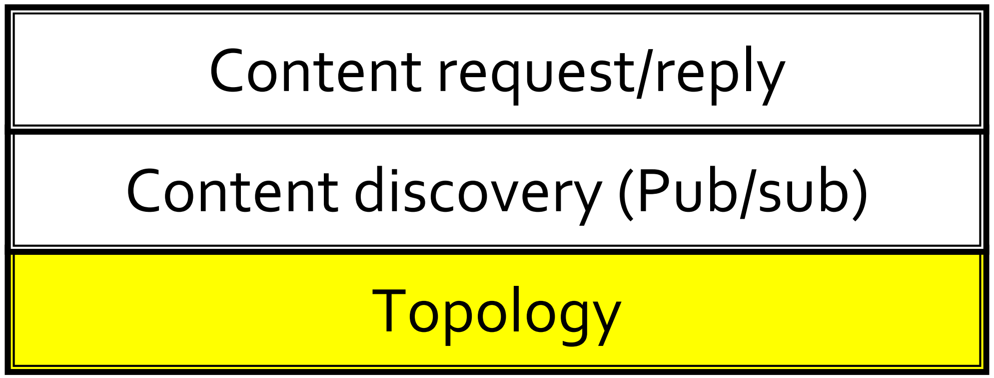
- Routing (A) and Caching (C) are *inter-related:*
  - Cross-layer approach
  - Social-related aware routing:
    - Using social-relation (interest) in routing path selection

Problem

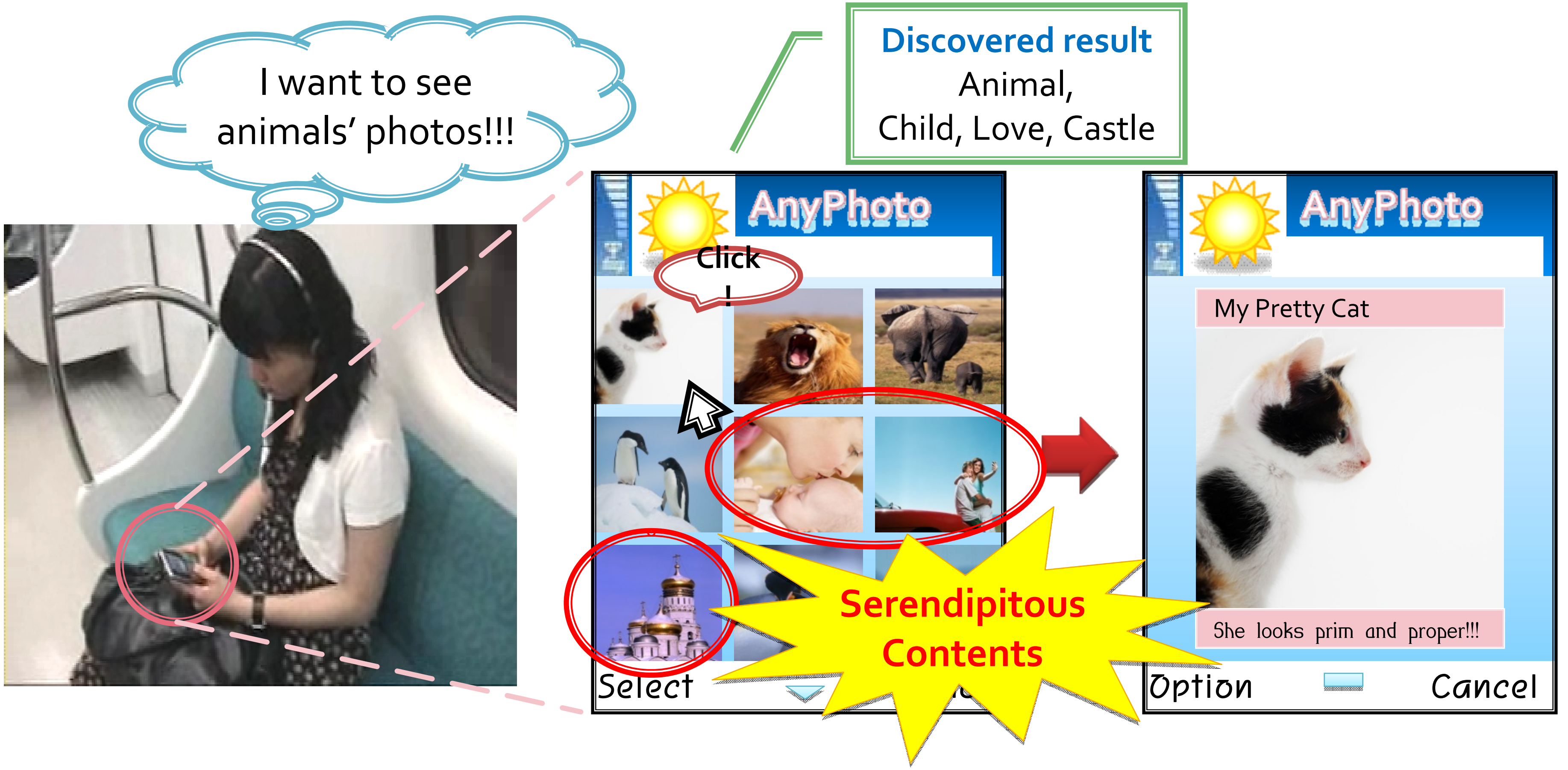
- In previous works
  - *Two different problems* at *different layers*
    - (1) Caching problem and
    - (2) Routing problem
  - *No cross-layer*: Application-level context (such as interest) is not revealed at network layer.
    - We cannot take advantage of the relationship between users derived by their social proximity

Goal

- Framework that supports MoSoSo
  - Opportunistic content discovery
  - Content fetching by request/reply

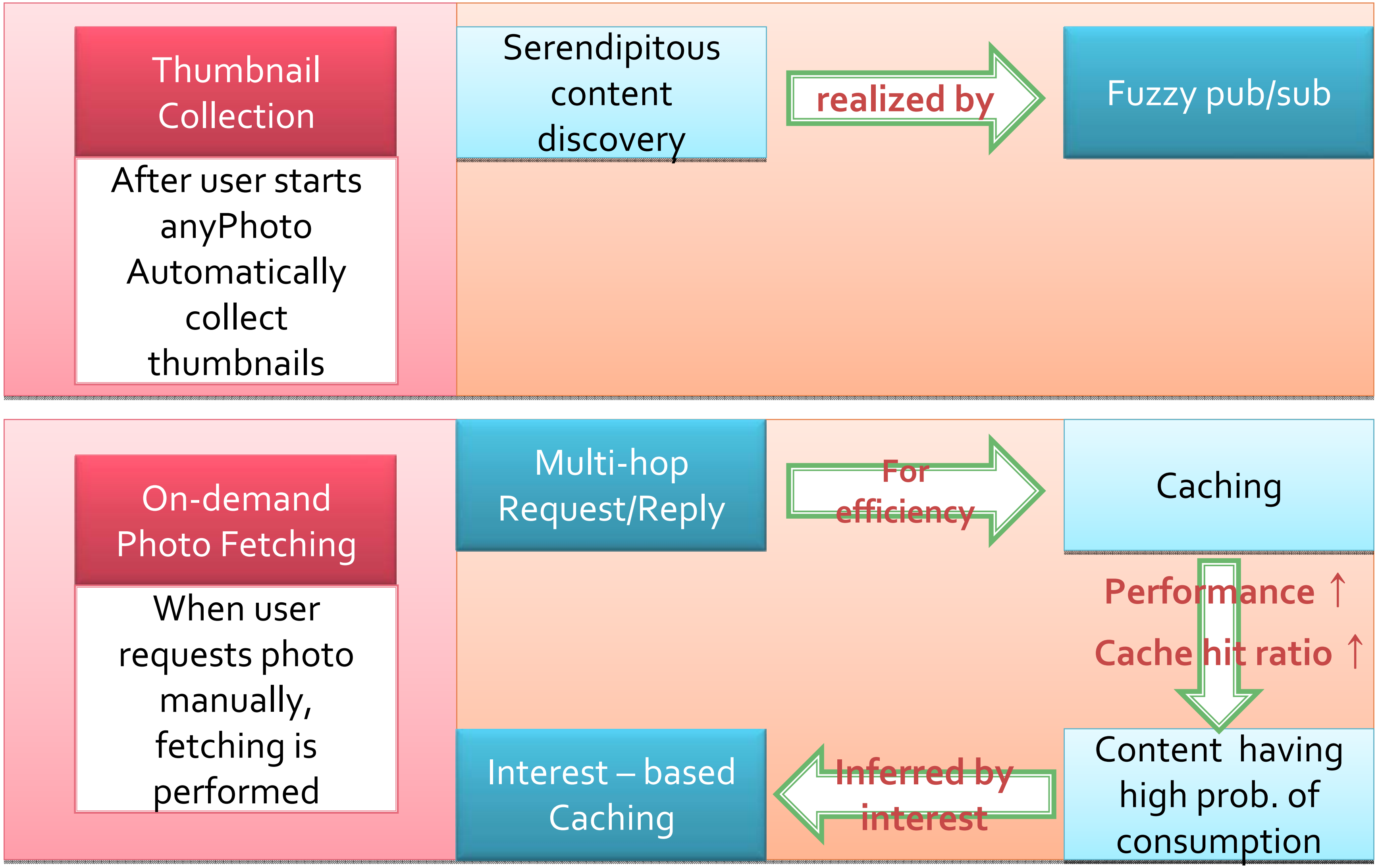


Scenario: Photo sharing in subway



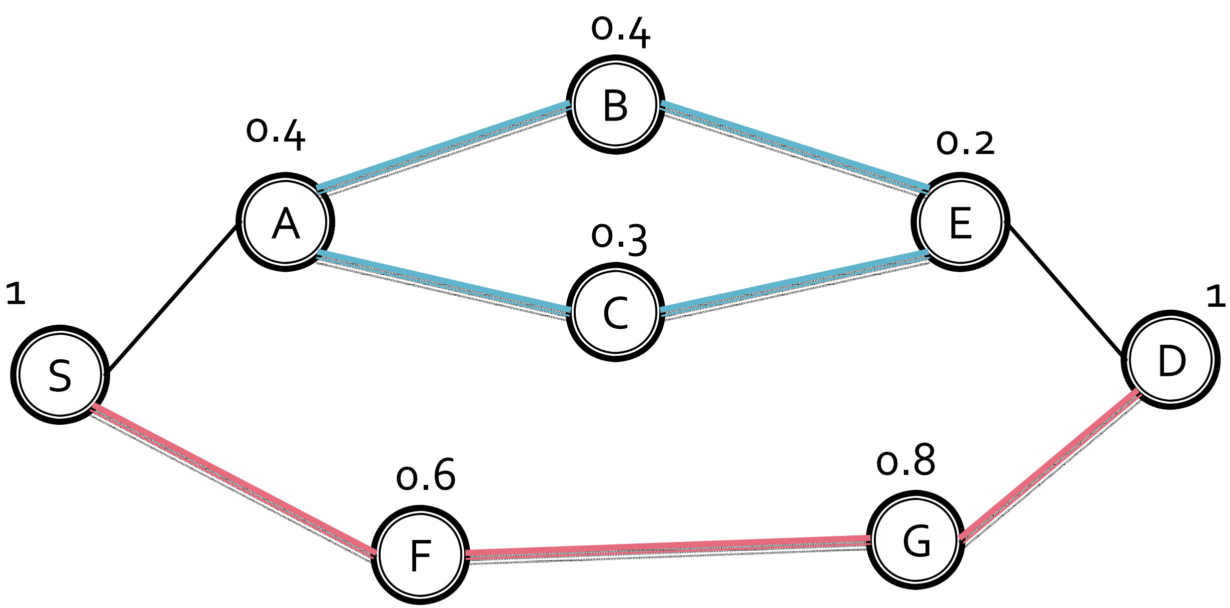
Start *anyPhoto* appl. → See discovered thumbnails → See requested photo

What do we need to implement ?

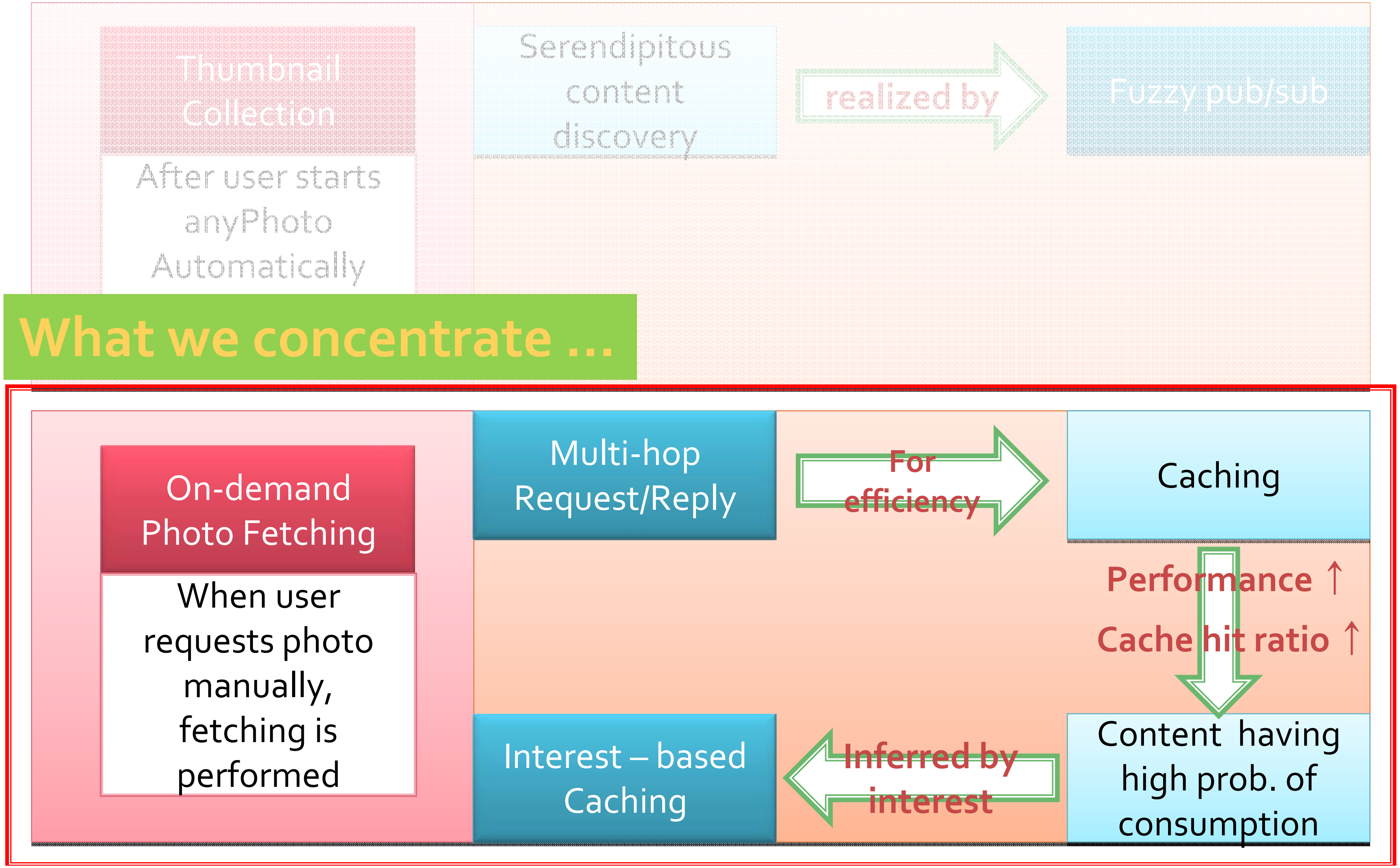


Fuzzy pub/sub

- Event forwarding
  - Forwards event not to all nodes
  - Event is forwarded based on social relationship



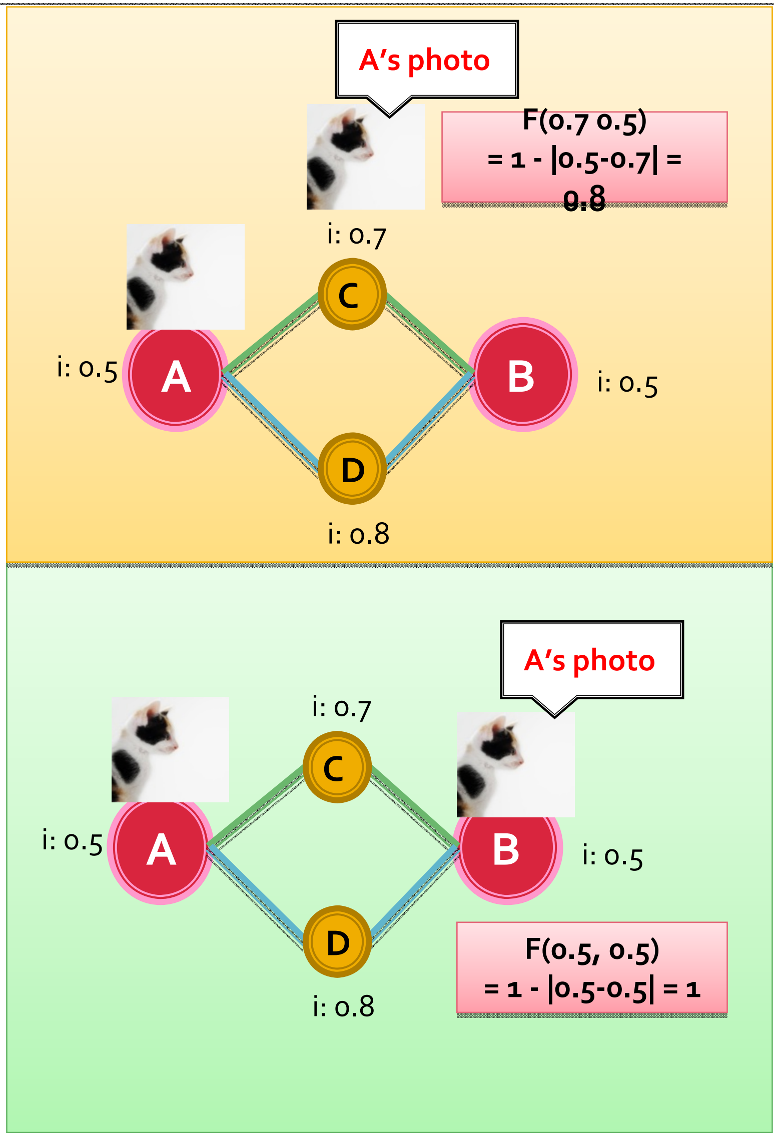
What do we need to implement ?



We assume that …

- Detecting a social-relationship
  - A user's interest is described in a user profile stored in his portable device.
  - Content is delivered with its meta information.
  - A formula calculating social relation is defined.
  - A user's interest and characteristic of content can be measured and mapped to scalar values (between 0 and 1) and these values are on the same dimension for calculating how much user is interested in content.
- User's interest can be expressed in one value.

Utilizing intermediate nodes as Cache



- When node receives a content …
  - decides to store the content depending on how likely it will consume the content in the future
  - calculates *expected cache hit ratio* using the below equation

$$F(i, c) = 1 - |c - i|$$

Social relation aware Routing table Construction

- A basic scheme is similar to DSDV except two things
  - No extended update scheme
  - **Utility value** for **routing metric**
    - Social relation is exploited as part of a utility value
    - A high utility value means that the nodes on a given route to a destination have high possibility to consume the contents sent by the destination of the path.

- Calculating **utility value**
  - Utility of a path

$$U(d) = \sum_{i \in I} (1 - e)^n F(i, d)$$

, where I is a set of intermediate node to d

Routing table entries

Dest	hopCnt	nextHop	Cumulated U(d)	interest	seqNum	installTime	flag
0	0	0	0	0.5	0	0.1	0

Advertised table entries

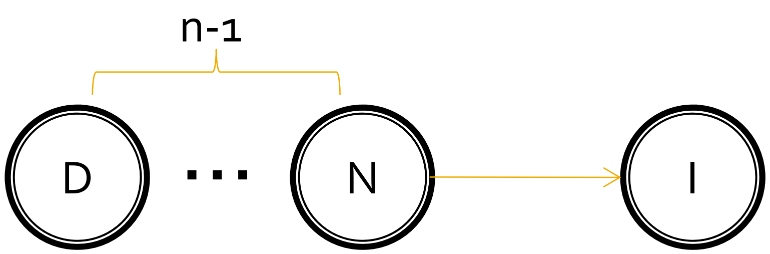
Dest	hopCnt	Cumulated U(d)	interest	seqNum
0	0	0	0.5	0

Scheme 1

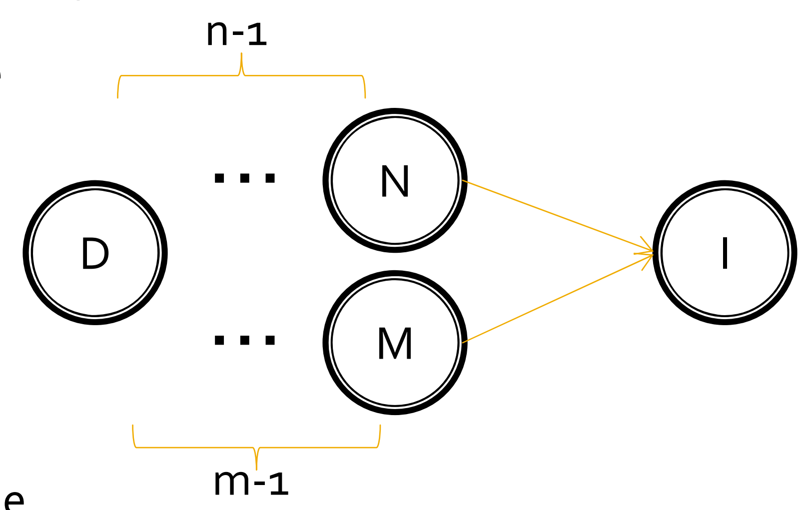
- Let's assume that node A and B are direct neighbors.
- Let's assume that all contents are equal sized.
  - Cost of transmission = sizeof(C_A) = 1
  - Let's assume that it is possible to calculate the benefit of participating in the path A → B and is only dependent on the fact that the content C is from node A. Then,
  - Benefit of node B's participating in path towards A = $F_B(A)$
- Let's assume that the error rate per hop is equal and denoted as e.
- Utility of the path (A → B) is given as
  - $U_B(A) = (1\text{-}e)F_B(A)$

Scheme 1

- Node N is a direct neighbor of node I. Node I receives a path information from node N to node D.
- Utility of the path (D → … → N → I) is given as
  - $U_I(D) = U_N(D) + (1-e)^n F_I(D)$
  - Since (1-e) > 0 and P >= 0, U is a monotonic increasing function. Thus, it is possible to apply existing shortest path algorithms.



- When node I receives two alternative path information,
  - Through node N,
    - $U_I(D) = U_N(D) + (1-e)^n F_I(D)$
  - Through node M,
    - $U_I(D) = U_M(D) + (1-e)^m F_I(D)$
  - Node I will choose the path through which U values is bigger.
  - If $F_x(y)$ is equal for any x and y, then the routing is shortest path routing.



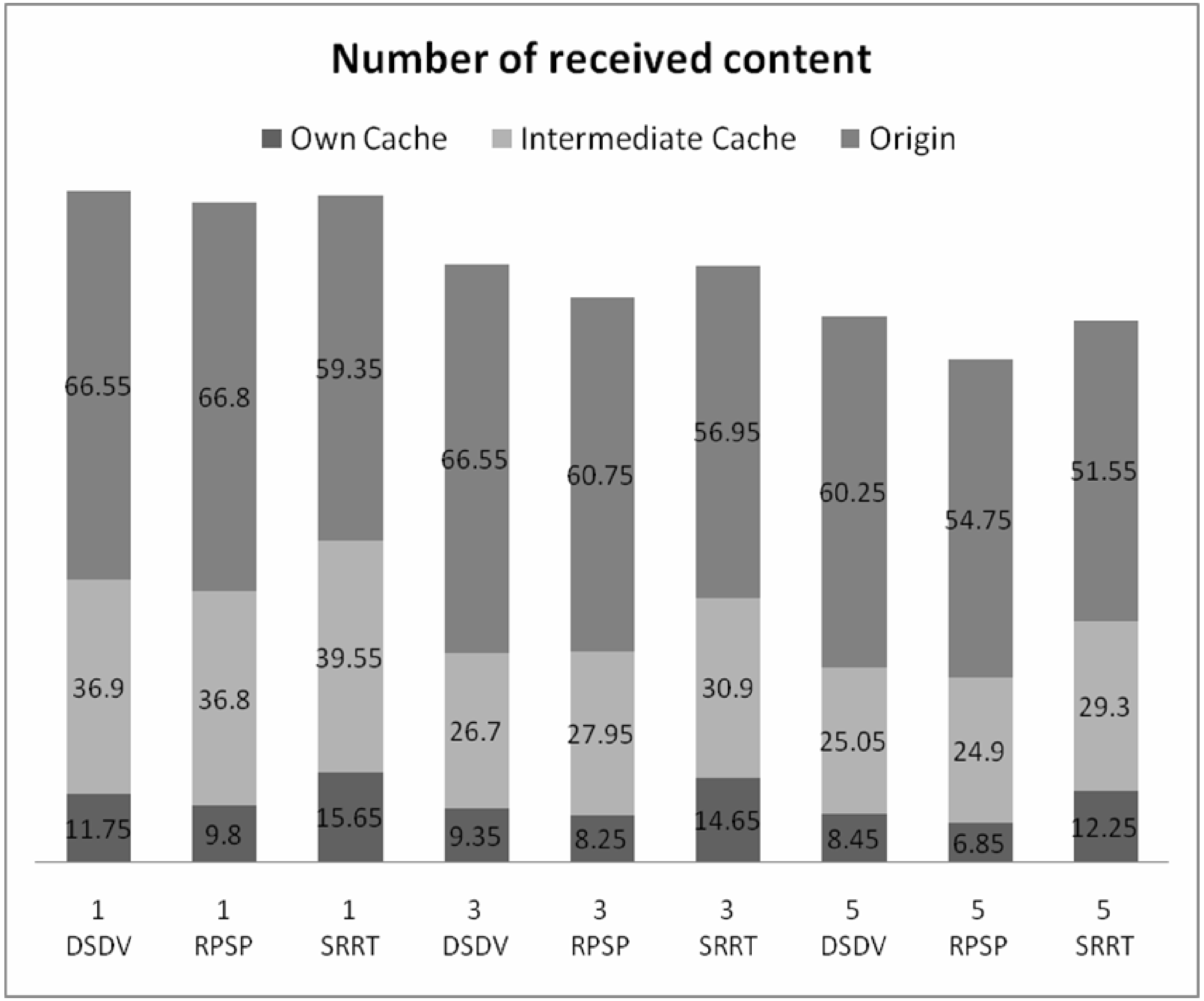
Evaluations

- Simulator : ns-2

- Space: 1000x1000m
- Number of nodes : 40
- Communication range : 250m
- Simulation time : 550s
- Routing table exchange time : 15s
- Mobility
  - Nodes move randomly
  - Maximum speed of 1~5 m/s with 3 second pause

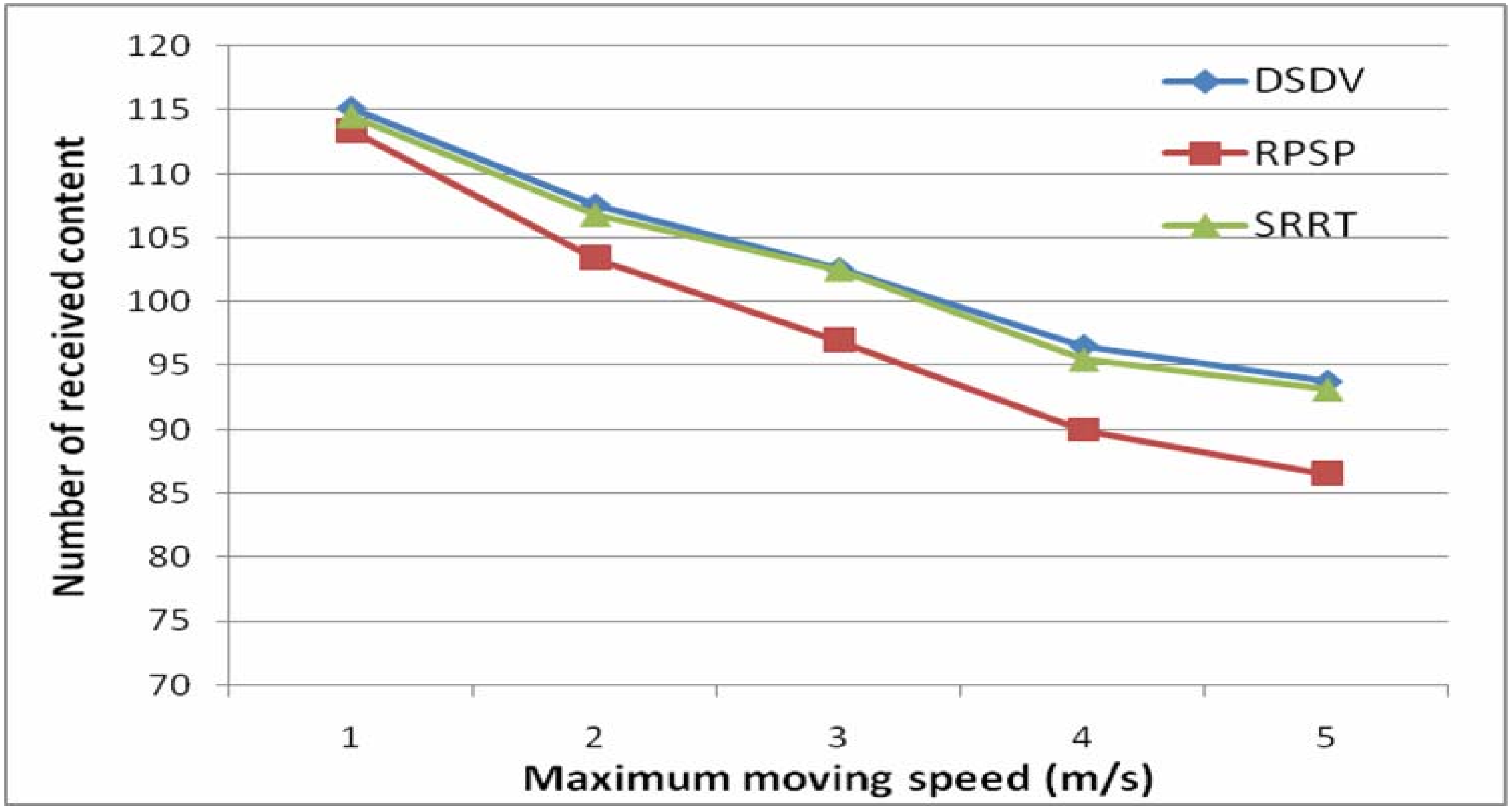
- Buffer size : 20
- Simple request/reply application
  - Request preferred content
  - Number of request : 160
  - Size of content : 1000B

Number of received contents

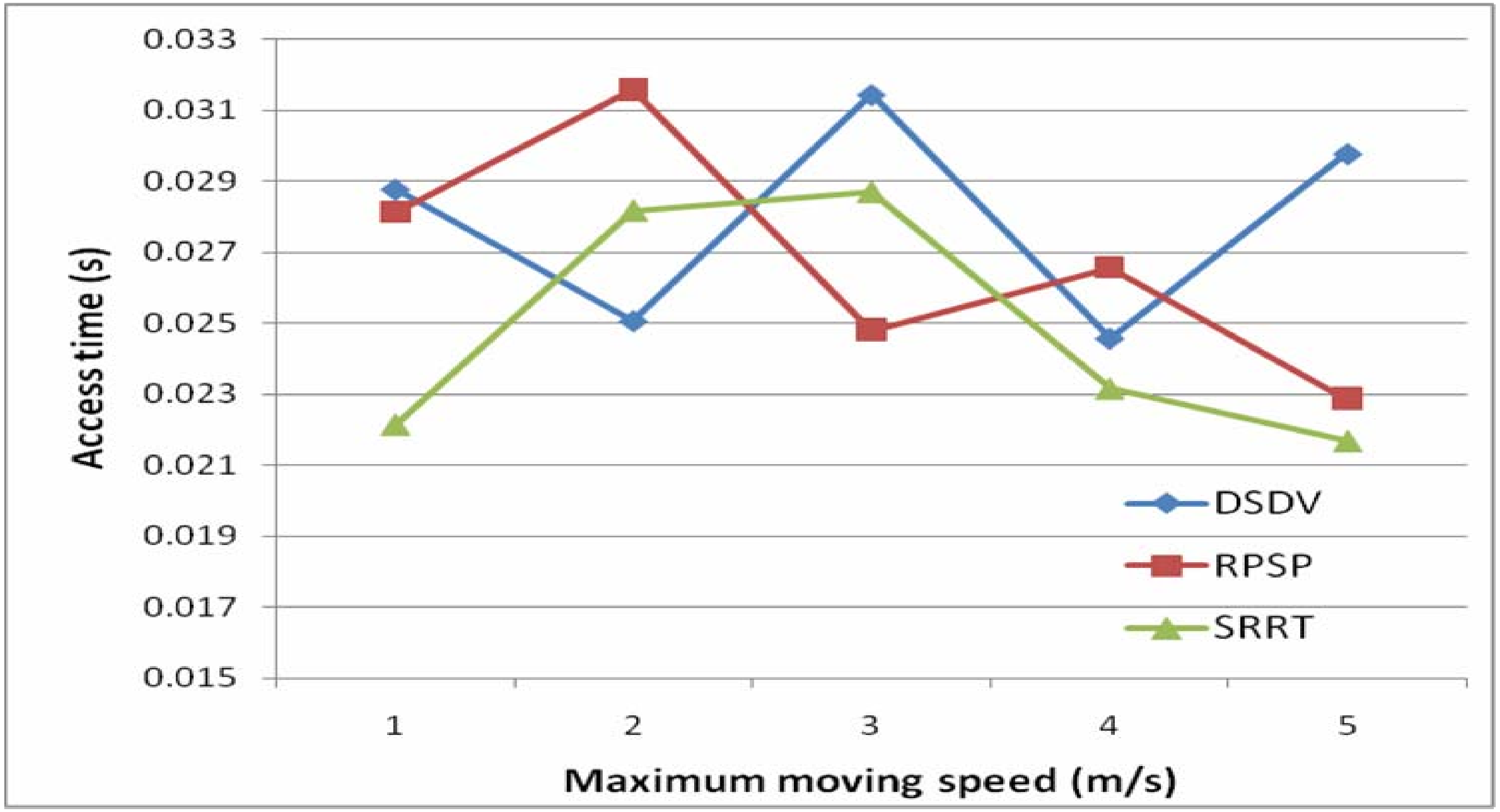


Shows that users receive more requested content from either their own caches or intermediate nodes' caches which results in smaller performance degradation as mobility increases

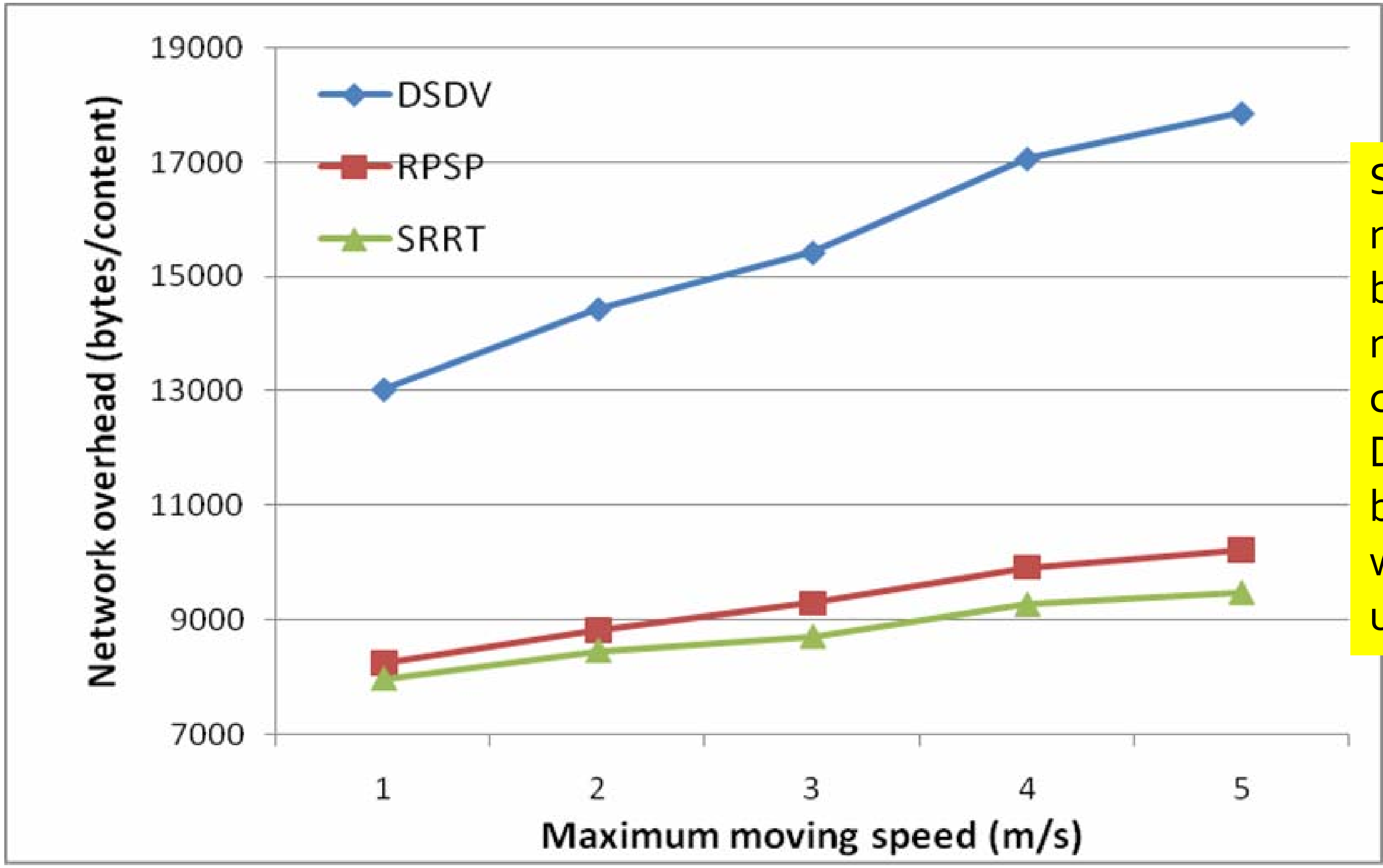
Number of received contents



Access time


0.033
0.031
0.029
0.027
0.025
0.023
0.021
0.019
0.017
0.015
Access time (s)
DSDV
RPSP
SRRT
1
2
3
4
5
Maximum moving speed (m/s)

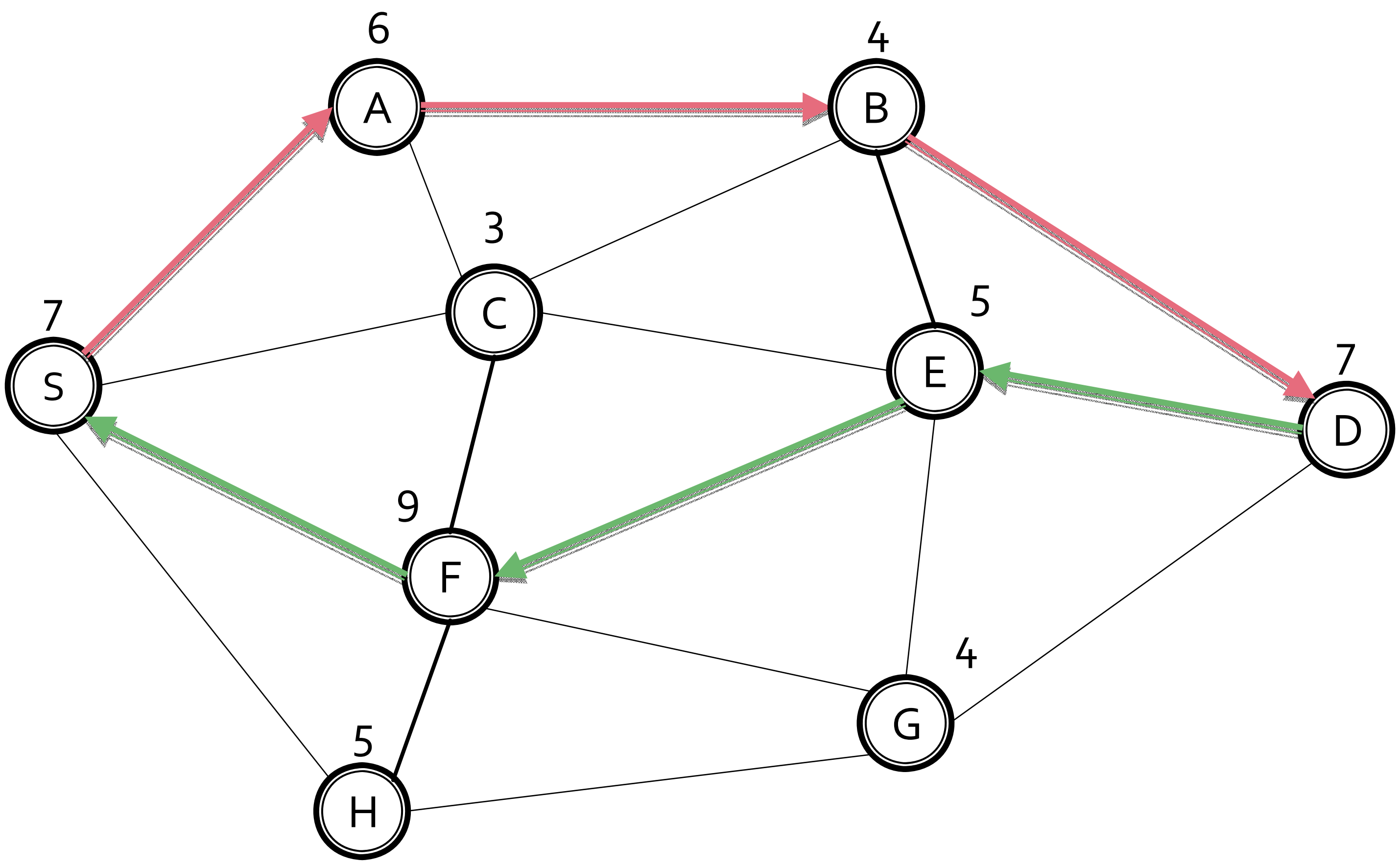
Network Overhead



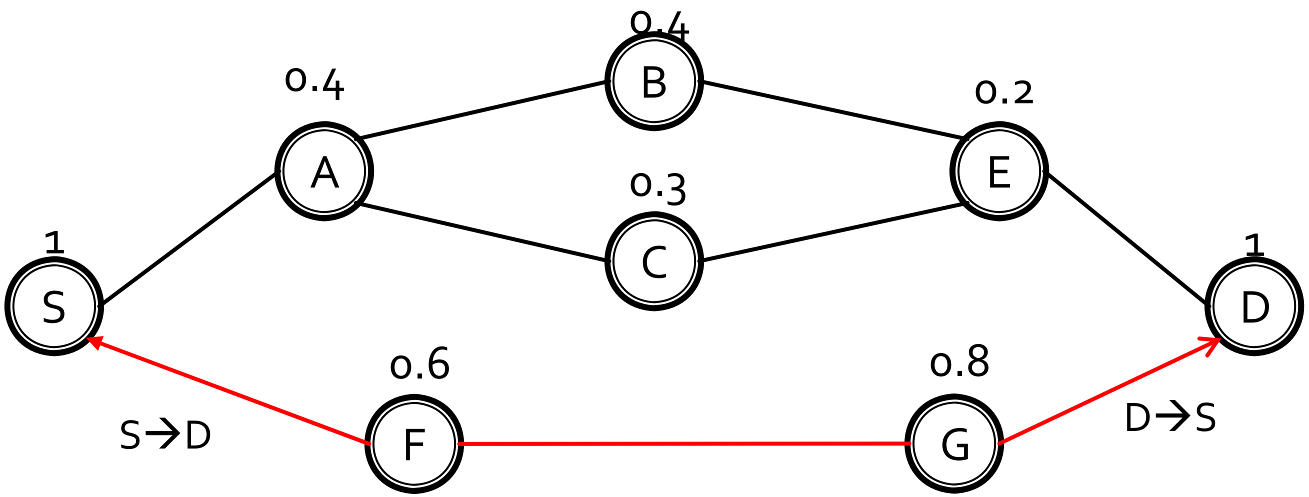
SRRT reduces network overhead by utilizing caches more frequently. In comparison with DSDV, DSDV burdens a network with its triggered update algorithm.

Analysis: D-SRRT

- Path is selected based on destination's interest
- If two paths have same utility values, faster path will be chosen.

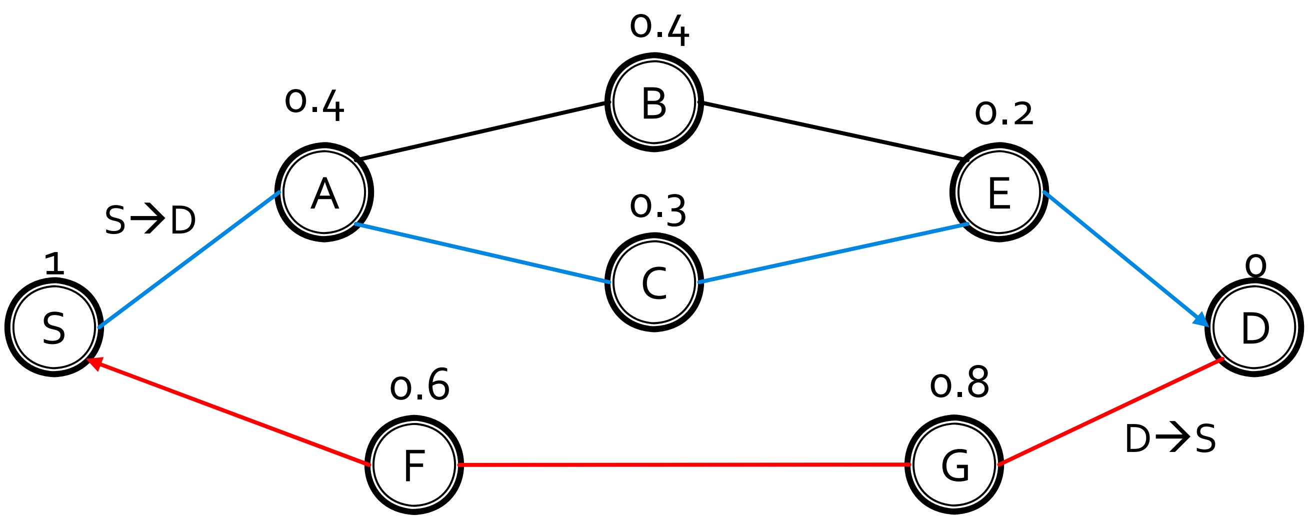


Analysis: D-SRRT



- Abbreviation
  - I = interest
  - RC = requested content
  - HC = holding content
- Destination-initiated algorithm
  - RC == I == HC

Issues – D-SRRT



- What if 'RC == I != HC'?
  - Path for request might be different from path for reply
  - We need source-initiated path selection

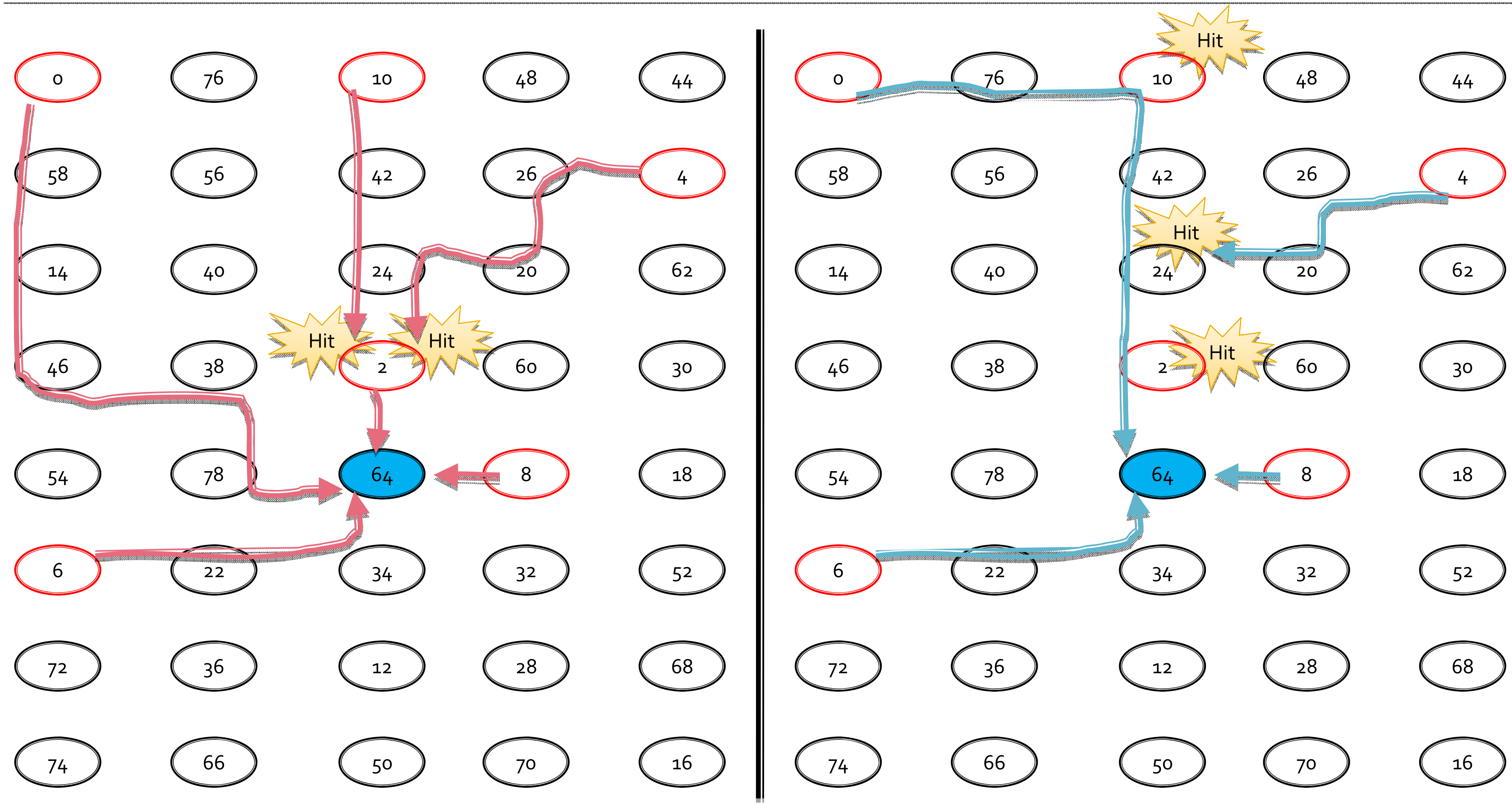
Source-initiated path selection

- Assumption
  - I == RC != HC
- Environment
  - Small network, small group
- Proactive vs. Reactive
  - On-demand method such as, AODV and DSR, occurs path discovery latency
  - Proactive algorithm is preferred.
- Algorithm
  - Not very different from D-SRRT except, entries of routing table

Advertised table entries

hopCnt	seqNum	Dest	Dest's Interest	IN1	IN1's interest	IN2	IN2's interest
3	0	D	1	G	0.8	F	0.6

DSDV vs. S-SRRT (Example)

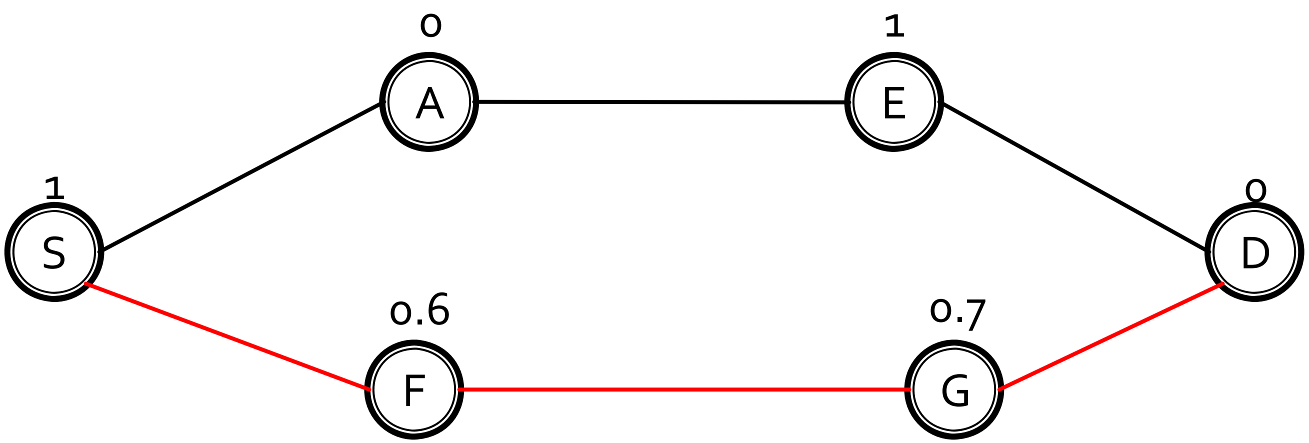


Request: 7 times → 36 hops

Request: 7 times → 26 hops

Issues for S-SRRT

- Is "User's interest is expressed in one value." is possible?
  - E.g. car, cat, baseball, ...
- Utility value: sum of expected cache hit ratio



Issues for S-SRRT

- Is "User's interest is expressed in one value." is possible?
  - Define interest!
    - User's interest model
    - Content metadata model
    - Similarity calculation model
  - Multi-interest → size of routing table
  - Multi-path routing protocol
    - Per-request source-initiated path selection
- Utility value: sum of expected cache hit ratio
  - Content access pattern model

We assume that …

- Detecting a social-relationship
  - A user's interest is described in a user profile stored in his portable device.
  - Content is delivered with its meta information.
  - ~~A formula calculating social relation is defined.~~
  - ~~A user's interest and characteristic of content can be measured and mapped to scalar values (between 0 and 1) and these values are on the same dimension for calculating how much user is interested in content.~~
- ~~User's interest is expressed in one value.~~
- I == RC != HC

Related works – Interest model

- User's interest representation and
  - Interest-based personalized search [1]
    - Construct ontology using Open directory project (ODP)
    - Figure out where user's interest is located in ontology
  - Bloom filter [2]
    - Compression method for various interest
- Similarity calculation between two users
  - Semantic matching [3]
  - An information theoretic approach to ontology-based interest matching [4]

Interest and content metadata model

- Interest model
  - Using ontology, define common N categories
  - User can have multi-interest
  - Each interest is one of elements in particular category
  - User interest is represented by N-dimension vector
- Content metadata model
  - Content is also represented by N-dimension vector for matching with user`s interest
- Similarity calculation model
  - Distance between two interests in ontology = D(i, j)
  - Sum of distances

Interest model

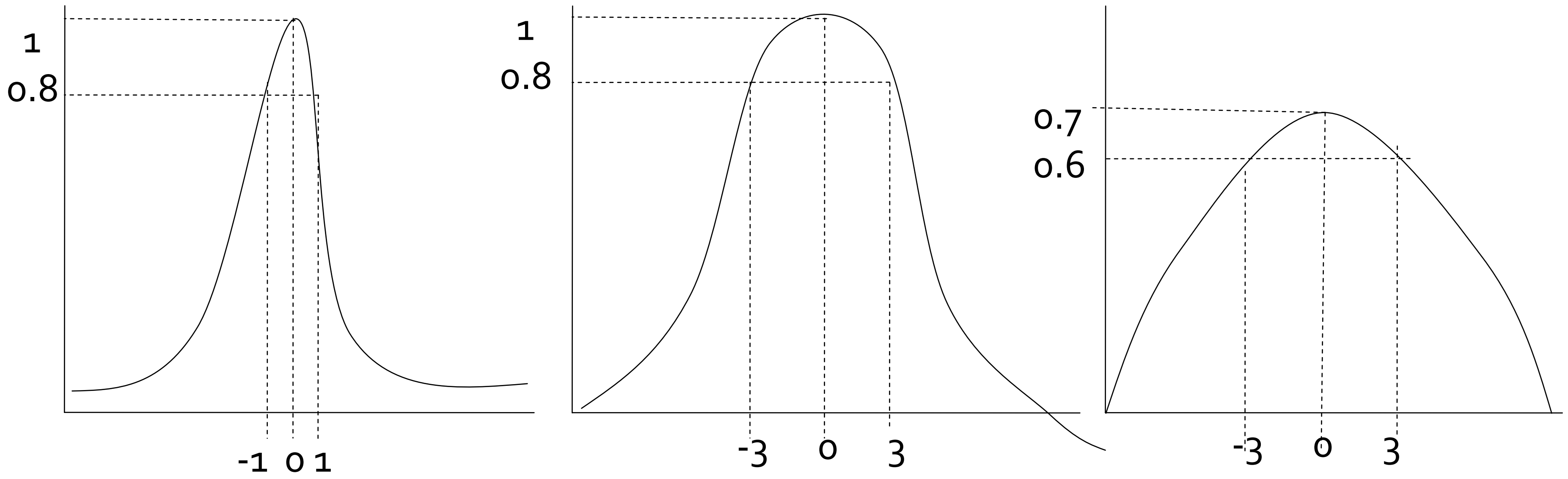
I1= (i11, i12, i13, i14, …)
I2= (i21, i22, i23, i24, …)
C1= (c11, c12, c13, c14, …)

$$\text{Similarity}(i_1, i_2) = \sum_{i=1}^{N} D(i_{1i}, i_{2i})$$

$$\text{Similarity}(i_1, C_1) = \sum_{i=1}^{N} D(i_{1i}, C_{1i})$$

Access pattern model

- Similarity between content and user's interest might be a metric that determine whether user will access to content or not.
- X:distance, Y: probability that access to content



- Utilized by caching decision and utility value calculation

Per-request source-initiated path selection

- Utility value calculation
  - Weighted sum

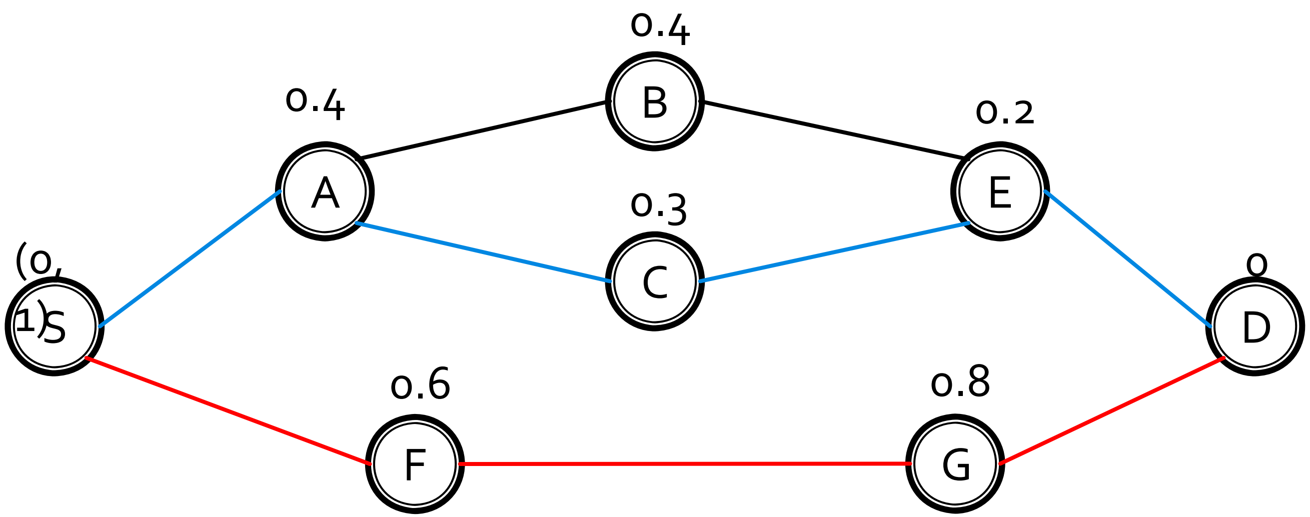
$$U(d) = \sum_{i \in I} (1 - e)^n W_f F(i, d)$$

, where i is intermediate node, d is destination node, n is number of hops, W is weighting value, e is link error rate

  - Weighting value
    - Determined by content access pattern & similarity
    - High similarity means high probability to access

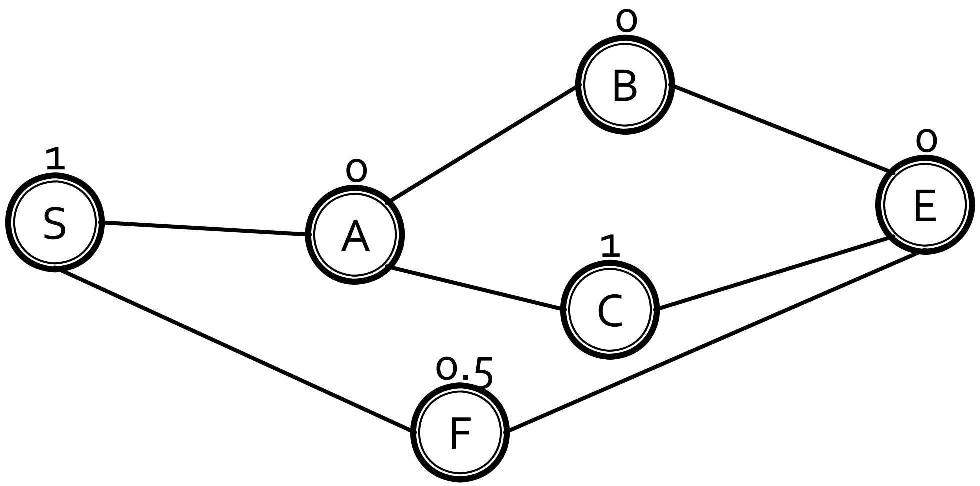
Per-request source-initiated path selection

- Path selection
  - Keep all paths toward a destination including intermediate nodes
  - Determine path per request
    - Calculate utility value for each path per request
  - *Determine next hop when data is actually forwarded*

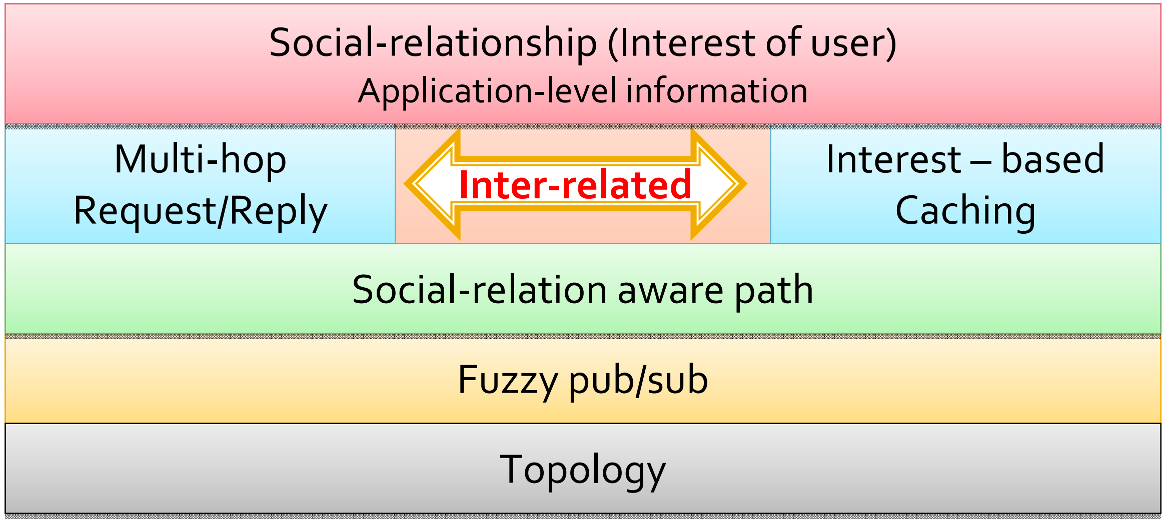


Issue – P-SRRT

- Keep all path?
  - Huge routing table
  - Network overhead
  - Takes long time for determining path
- Possible solution?
  - Keep all received path & select one or two paths
  - Accumulation of paths occurs wrong decision.



Summary & Conclusions



- Summary
  - MoSoSo → Content sharing → Sharing experience → Social network construction
  - Framework → content discovery (fuzzy pub/sub) & content dispatching (SRRT)
  - Caching & Routing are interrelated
  - Framework uses application semantic (social relation) for improving efficiency.
- Contribution
  - <u>Our scheme can help (not just support) application to improve its performance by cross-layering.</u>

References

[1] Z. Ma, G. Pant, and O. R. L. Sheng. Interest-based personalized search. ACM Trans. Inf. Syst., 25(1):5, 2007

[2] Burton H. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM, v.13 n.7, p.422-426, July 1970

[3] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. S-Match: An Algorithm and an Implementation of Semantic Matching. ESWS 2004, 61--75

[4] Koh, W., and Mui, L. An information-theoretic approach for ontology-based interest matching. IJCAI'01 Workshop on Ontology Learning (Seattle, WA, Aug. 2001).