SPRINT: Social Prediction-Based Opportunistic Routing

Radu-Ioan Ciobanu, **Ciprian Dobre**, Valentin Cristea

Automatic Control and Computers Faculty,
University Politehnica of Bucharest

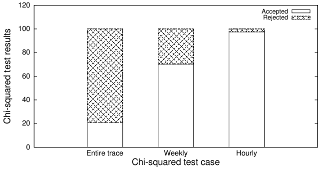
June 4, 2013

SPRINT

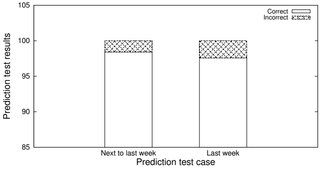
- Social PRedIction-based routing in opportunistic NeTworks
- introduces online social information about nodes (e.g. Facebook, Twitter, LinkedIn) as routing criterion
- in certain environment types, contacts between mobile devices are highly predictable
- prediction component approximates human mobility as a Poisson distribution when users follow rare events-based mobility patterns
- compare with BUBBLE Rap on real-life traces and mobility models

Contact Prediction [1]

- try to predict a node's future behavior
- number of encounters and contact duration in an academic environment are regular
- model contacts per hour in a day as a Poisson distribution
- prove the assumptions using the chi-squared test



(a) Chi-squared tests



(b) Prediction success

[1] Radu Ioan Ciobanu and Ciprian Dobre. 2012. Predicting encounters in opportunistic networks. In *Proceedings of the 1st ACM Workshop on High Performance Mobile Opportunistic Systems* (HP-MOSys '12).

The SPRINT Algorithm

- SPRINT nodes have data memory (messages) and cache memory (contact history)
- on contact, they exchange information about their messages:
  - hash of the content
  - source
  - destination
  - generation time
  - hop count
- each node A computes the utility of each message M and attempts to maximize its data memory:

$$u(M, A) = w_1 * U_1(M, A) + w_2 * U_2(M, A)$$

The SPRINT Algorithm (2)

$$U_1(M, A) = freshness(M) + p(M, A) * (1 - \frac{enc(M,A)}{24})$$

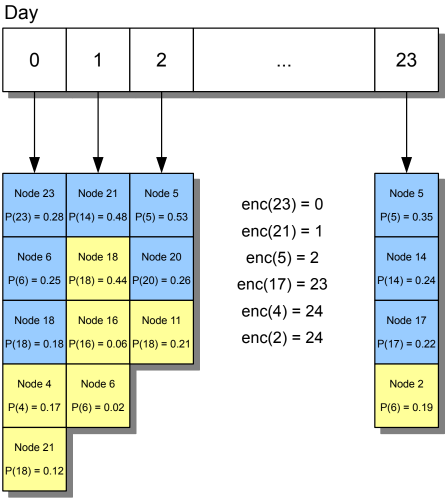
- $freshness(M)$ → favors newer messages (0 if message is older than one day, 0.5 otherwise)
- $p(M, A)$ → probability of node A being able to deliver message M:
  - count how many time each node was encountered
  - if the node was met in the same week-day and two-hour interval, increase value by 1
  - if the node is socially-connected, double the value
  - compute encounter probability as ratio between contacts with each node and total contacts
  - compute number of encounters N for each of the next 24 hours using Poisson (choose first N, λ → max likelihood)
- $enc(M, A)$ → time (h) until M's destination will be met by A

The SPRINT Algorithm (3)

$$U_2(M, A) = c_e(M, A) * \frac{s_n(M)+hop(M)+pop(A)+t(M,A)}{4}$$

- $c_e(M, A) \rightarrow 1$ if A is socially-connected with M's destination, or if it will encounter a node that has a social relationship with M in the next 24 hours, 0 otherwise
- $s_n(M) \rightarrow 1$ if M's source and destination are not socially-connected, 0 otherwise
- $hop(M) \rightarrow$ normalized number of nodes visited by M
- $pop(A) \rightarrow A$'s social popularity (e.g. number of Facebook friends)
- $t(M, A) \rightarrow$ normalized time spent by A in contact with M's destination

The SPRINT Algorithm (3)



Mobility Traces and Models

- UPB 2011:
  - faculty grounds
  - Bluetooth
  - 35 days, 22 participants
- UPB 2012:
  - faculty grounds
  - Bluetooth and WiFi
  - 64 days, 66 participants
- St. Andrews:
  - faculty grounds, around the surrounding town
  - Bluetooth
  - 79 days, 27 participants

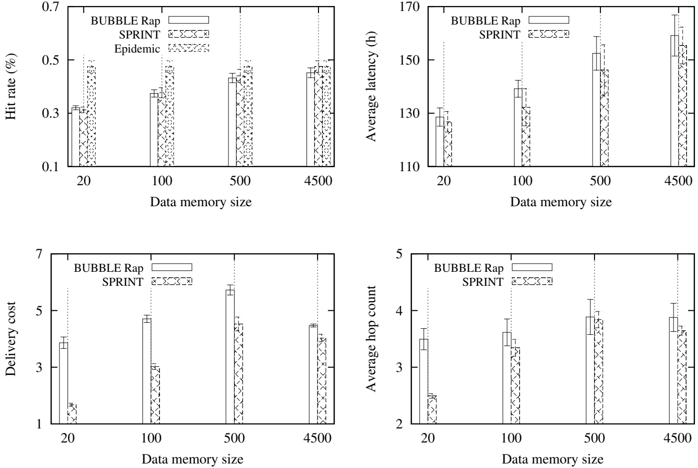
Mobility Traces and Models (2)

- Content:
  - locations around a city
  - Bluetooth
  - 25 days, 36 mobile and 18 fixed participants
- Infocom 2006:
  - scientific conference
  - Bluetooth
  - 4 days, 78 mobile and 20 stationary participants
- HCMM:
  - synthetic mobility model
  - simulated an academic environment: 400 $\times$ 400 meters grid with 10 $\times$ 10 meters cells, transmission radius of 10 meters
  - 3 days, 33 participants

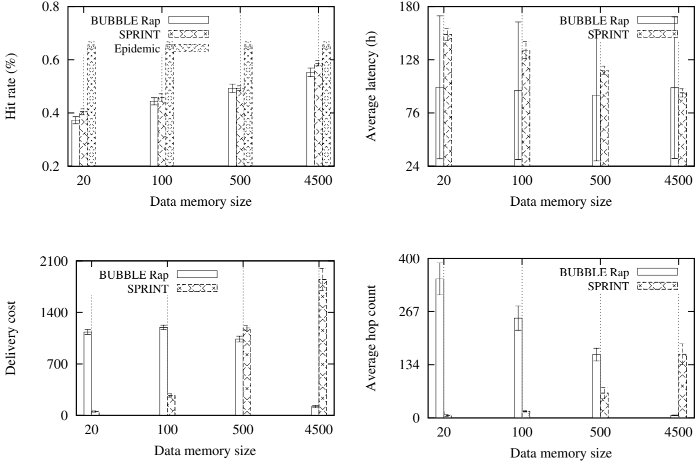
Experimental Setup

- compare with distributed BUBBLE Rap (with k-CLIQUE and C-window) and Epidemic
- 30 messages per weekday (Zipf distribution with exponent 1)
- the time of generation selected according to encounter periods
- fixed cache size (40), varied data memory size (from 20 to 4500)
- 95% confidence level
- use social network information where available, or k-CLIQUE otherwise
- measure hit rate, delivery latency, hop count and delivery cost

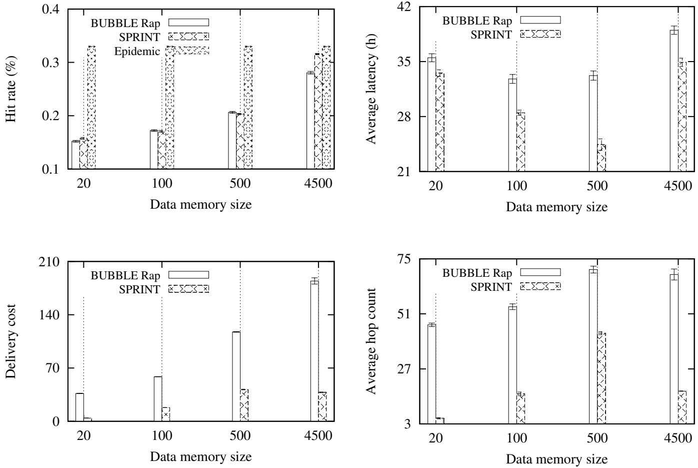
Results (UPB 2011)



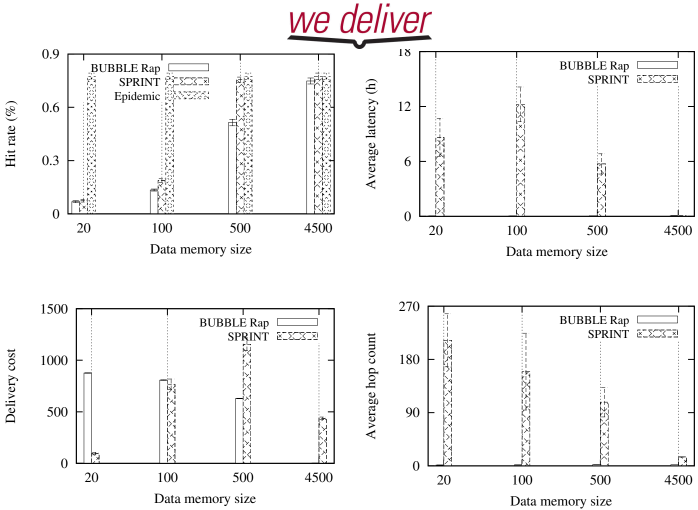
Results (St. Andrews)



Results (Content)



Results (HCMM)



Selfish Nodes Detection [2]

- selfish nodes → nodes that don't want to participate in the routing process for various reasons
- propose a novel social-based collaborative content and context-based selfish node detection algorithm
- characteristics:
  - an incentive mechanism that rewards active nodes and punishes selfish ones
  - based on gossiping
  - context-based: social knowledge, battery level, etc.
  - content-based: message content-based decisions

[2] Radu Ioan Ciobanu, Ciprian Dobre, Mihai Dascalu, Stefan Trausan-Matu and Valentin Cristea. 2013. Collaborative Selfish Node Detection with an Incentive Mechanism for Opportunistic Networks. In *Proceedings of the 5th International Workshop on Management of the Future Internet* (ManFI '13).

Conclusions

- SPRINT → routing algorithm for ONs that uses information about a node's social connections, contacts history, and predictions of future encounters
- it outperforms existing algorithms for various mobility traces and models
- improves hit rate, delivery latency, hop count and delivery cost
- distribution of contacts in certain scenarios is highly predictable and can be approximated as Poisson