BUBBLE RAP: Social-Based Forwarding in Delay-Tolerant Networks

Pan Hui, Jon Crowcroft, Eiko Yoneki

Presented By: Shaymaa Khater

Outline

- **Introduction.**
- Goals.
- Data Sets.
- Community Detection Algorithms
  - K-clique Community Detection .
  - Weighted Network Analysis (WNA).
- Heterogeneity in Centrality
- Different Forwarding algorithms
  - RANK Algorithm.
  - LABEL Algorithm.
  - BUBBLE Algorithm.
- DiBuBB.
- Conclusion.

Introduction

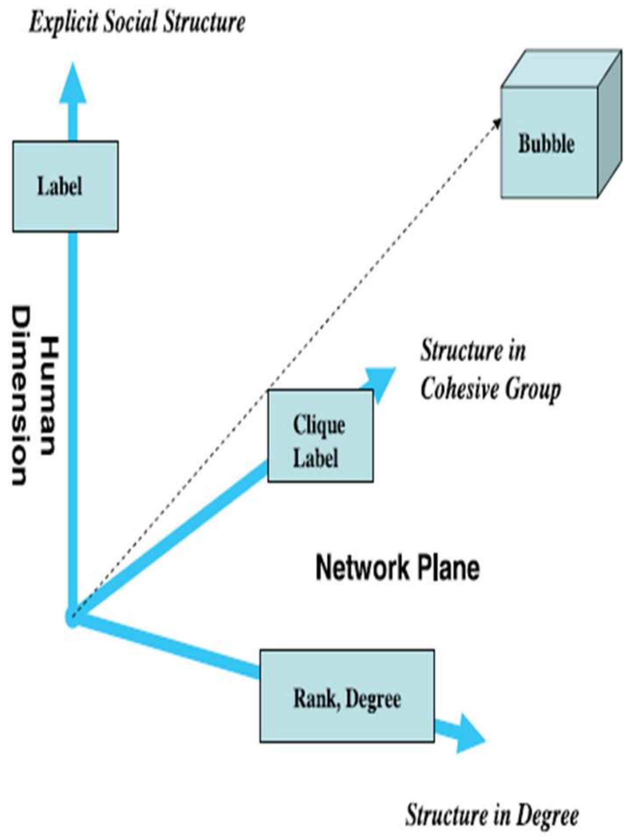
- **Pocket Switched Networks (PSN)** : is a type of Delay Tolerant Networks (DTN) that make use of the human mobility and local/global connectivity in order to transfer data between mobile users' devices.

- Some MANETs and DTN routing algorithms provide forwarding by building and updating routing tables whenever mobility occurs.

- Not cost effective for a PSN:
  - mobility is often unpredictable
  - topology changes can be rapid.

Introduction

- So rather than exchange much control traffic to create unreliable routing structures, it is better to search for some characteristics of the network which are less volatile than mobility.
- As PSN is formed by people, Those people's social relationships may vary much more slowly than the topology.
- So social metrics are important properties to guide data forwarding in human networks.



Outline

- Introduction.
- **Goals.**
- **Data Sets.**
- Community Detection Algorithms
  - K-clique Community Detection .
  - Weighted Network Analysis (WNA).
- Heterogeneity in Centrality
- Different Forwarding algorithms
  - RANK Algorithm.
  - LABEL Algorithm.
  - BUBBLE Algorithm.
- DiBuBB.
- Conclusion.

Goals

- To improve our understanding of human mobility in by focusing on two social metrics
  - Community
  - Centrality.

- Answer the following questions:
  - How variation in node popularity affects forwarding in a PSN?
  - Are communities detectable in PSNs?
  - What is the effect of social-based forwarding compared to other forwarding schemes in a real environment?
  - Can we devise a fully decentralized way for such schemes to operate?

Data Sets

- **Infocom05** - the iMotes were distributed to students attending the Infocom student workshop. They belong to *different social communities.*
- **Infocom06** - the same as in Infocom05 except that the scale is larger
- **Hong-Kong** - the people carrying the iMotes were chosen independently in a Hong-Kong bar to avoid any social relationship between them.
- **Cambridge** - the iMotes were distributed to students from University of Cambridge Computer Laboratory
- **Reality** – smart phones were deployed to students and staff at MIT over a period of 9 months.

Experimental data set	**Infocom05**	**Infocom06**	**Hong-Kong**	**Cambridge**	**Reality**
Device	iMote	iMote	iMote	iMote	Phone
Network type	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
Duration (days)	3	3	5	11	246
Number of Experimental Devices	41	98	37	54	97
Number of internal contacts	22,459	191,336	560	10,873	54,667
Average # Contacts pair day	4.6	6.7	0.084	0.345	0.024

iMote

Outline

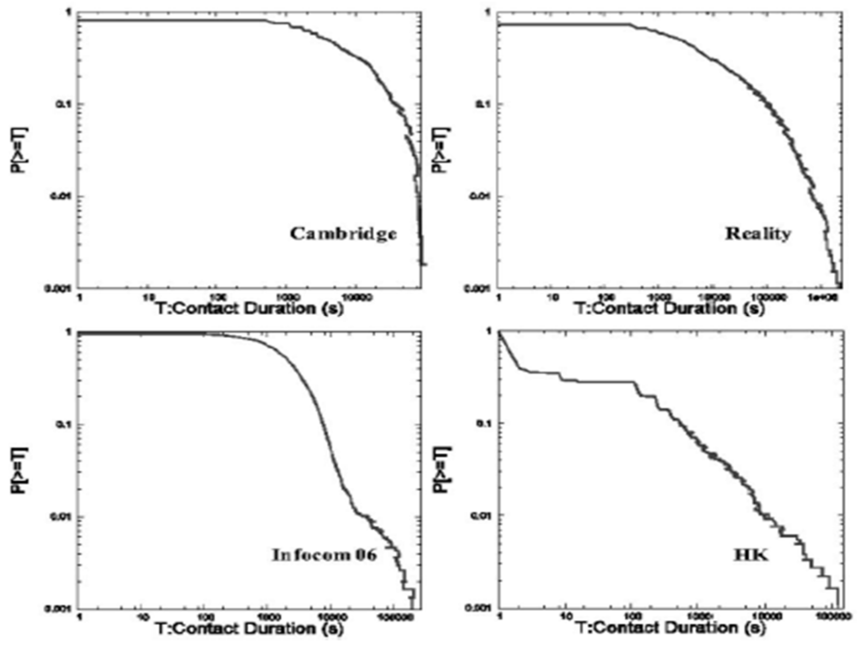
- Introduction.
- Goals.
- Data Sets.
- **Community Detection Algorithms**
  - **K-clique Community Detection .**
  - **Weighted Network Analysis (WNA).**
- Heterogeneity in Centrality
- Different Forwarding algorithms
  - RANK Algorithm.
  - LABEL Algorithm.
  - BUBBLE Algorithm.
- DiBuBB.
- Conclusion.

Are communities detectable in PSNs?

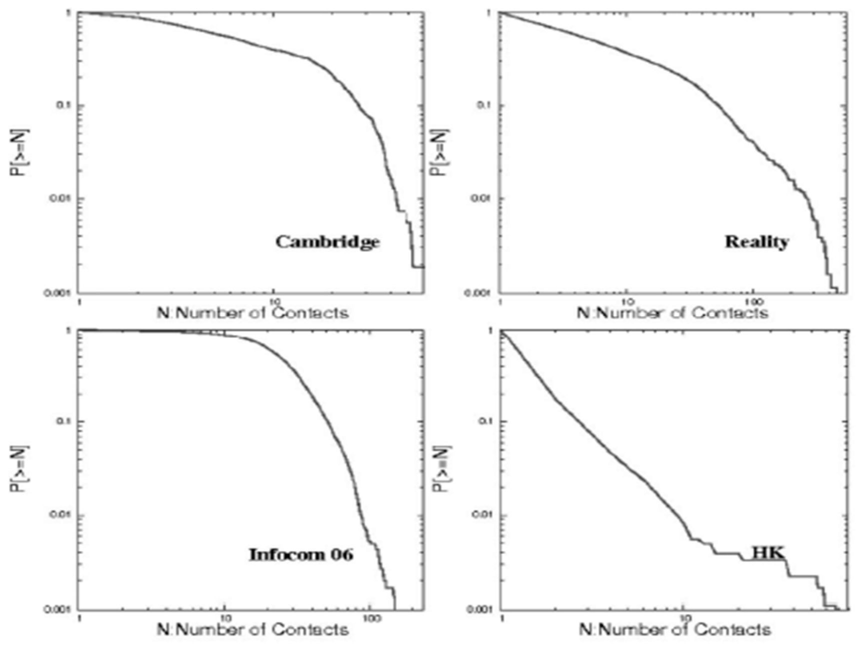
- This requires community detection algorithm
- Criteria for choosing the algorithm:
  - Ability to uncover overlapping communities
  - A high degree of automation (low manual involvement),

Contact graphs

- Convert human mobility traces into weighted contact graphs based on the number of contacts and the contact duration.
- Nodes represent the physical traces.
- Edges represents the contacts
- Weight of the edge are based on the specified metrics (contact duration, number of the contacts)



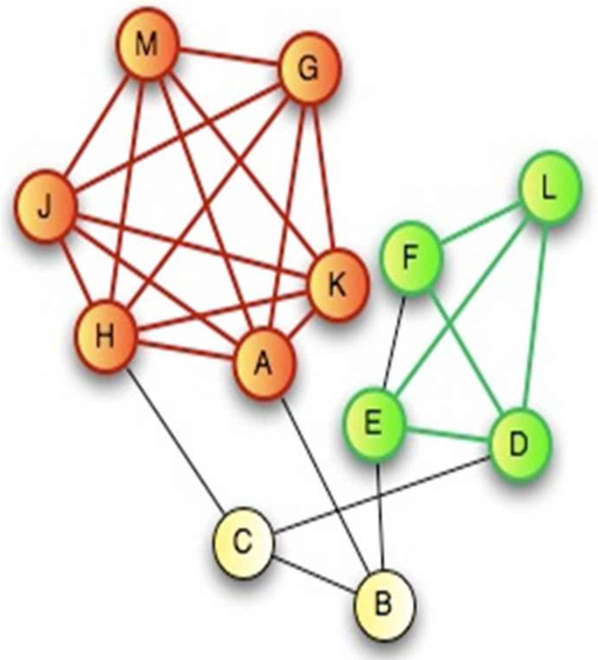
The distribution of pair-wise contact durations



The distribution of pair-wise number of contacts.

K-CLIQUE Community Detection

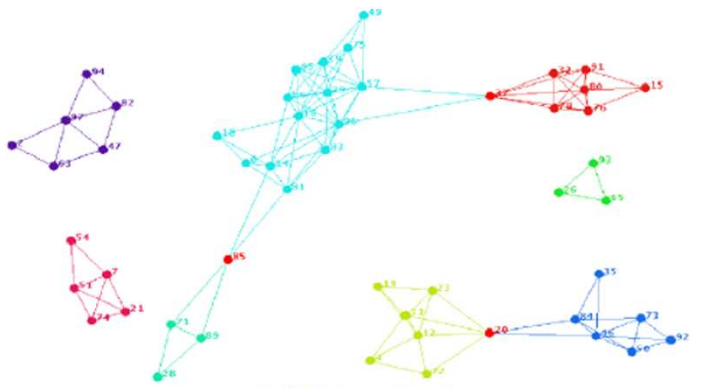
- Union of all adjacent k-cliques (complete sub graphs of size k) [Palla et al]
- Two *k*-cliques are adjacent if they share *k* – 1 nodes.
- Designed for binary graphs (undirected, unweighted)
- Satisfies the *overlapping feature* (a node can belong to several different *k*-clique clusters at the same time)

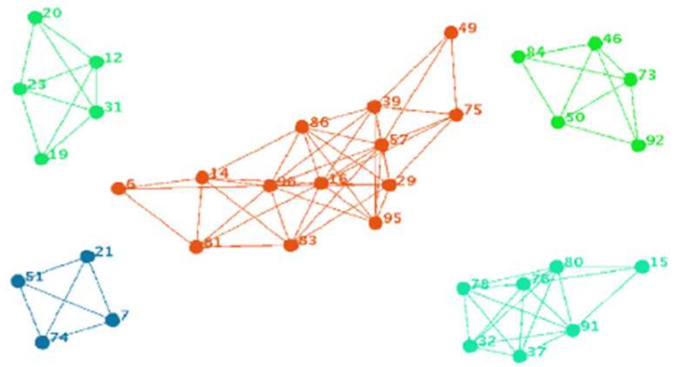


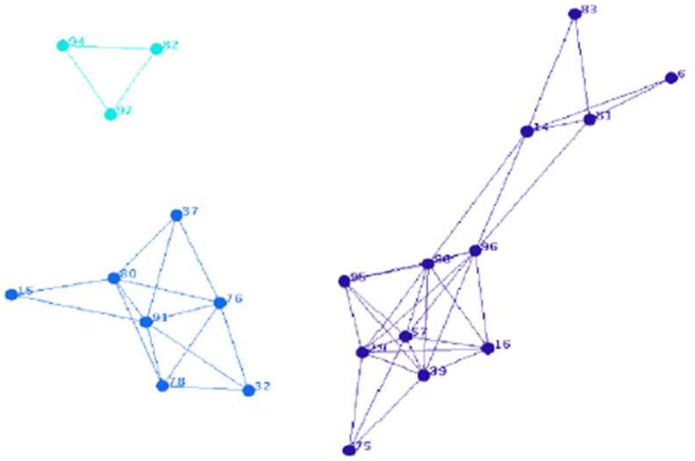
Cliques

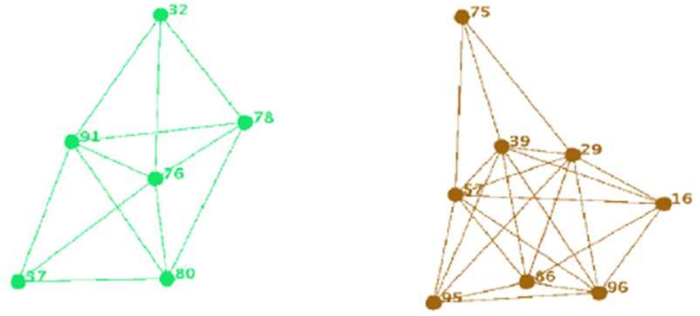
K-CLIQUE Community Detection

- Communities based on contact durations with weight threshold = 388800s (4.5days), 648000s (7.5days) and k=3,4 (*Reality)*


W388800, k=3


W388800, k=4


W648000, k=3


W648000, k=4

Weighted Network Analysis

- Use weighted modularity as a measurement of the community it detects.

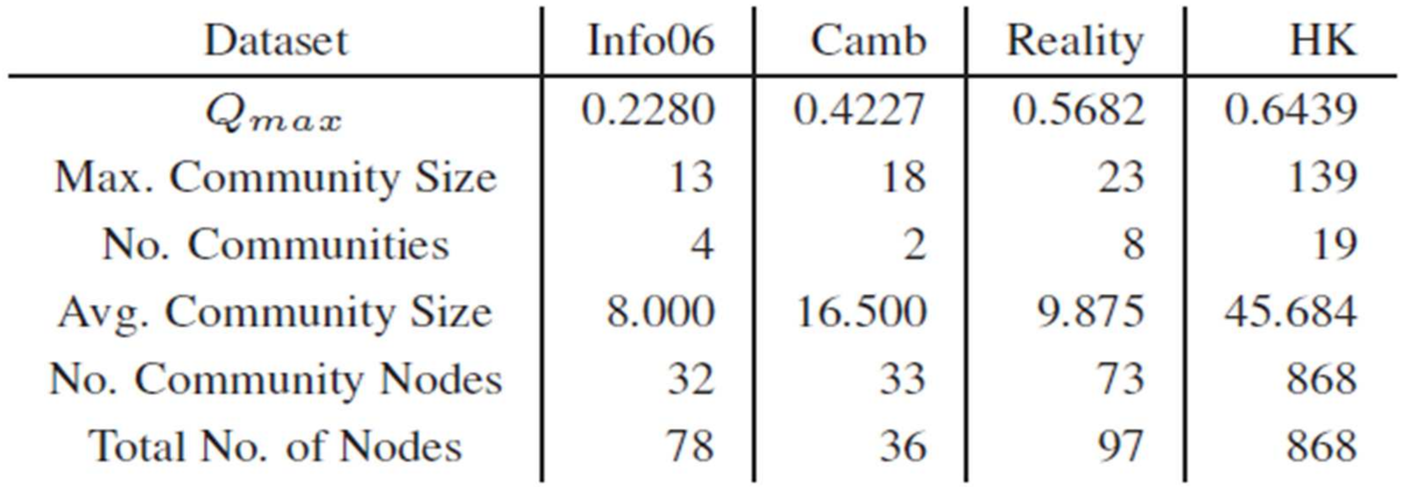
- For each community partitioning of a network, compute the corresponding modularity (Q):

$$Q = \sum_{vw} \left[\frac{A_{vw}}{2m} - \frac{k_v \, k_w}{(2m)^2} \right] \delta(c_v, c_w)$$

A_{vw}: weight of the edge between vertices v and w.
m = $\frac{1}{2} \sum_{vw} A_{vw}$
$\delta(i, j)$ = 1 if (i == j) , 0 otherwise.
k_v : degree of vertex v.
c_v: community of vertex v.

Weighted Network Analysis

- Communities detected by applying WNA on four datasets.



Dataset	Info06	Camb	Reality	HK
Q_{max}	0.2280	0.4227	0.5682	0.6439
Max. Community Size	13	18	23	139
No. Communities	4	2	8	19
Avg. Community Size	8.000	16.500	9.875	45.684
No. Community Nodes	32	33	73	868
Total No. of Nodes	78	36	97	868

- Infocom06 – Qmax is low, agrees with the fact that in a conference the community boundary becomes blurred.

- Cambridge – the two communities exactly matched the two groups (1st year and 2nd year) of students selected for the experiment.

- Reality - Qmax is high, reflects the more diverse campus environment

Are communities of nodes detectable in PSN traces?

- *K-CLIQUE* method fits overlapping criteria
  - but was designed for binary graphs, thus we must threshold the edges of the contact graphs in order to use it and it is difficult to choose an optimum threshold manually

- Weighted network analysis (WNA) can work on weighted graphs directly
  - but it cannot detect overlapping communities

- Both *K-CLIQUE and WNA are used to complement each other*.

Centralized community detection algorithms

- Give us rich information about the human social clustering
- Useful for offline data analysis on mobility traces collected
- Useful for exploring structures in the data and hence design useful forwarding strategies, security measures

Outline

How does the variation in node popularity help us to forward in a PSN?

- Choose popular hubs as relays instead of unpopular ones as relays to help design more efficient forwarding strategies.

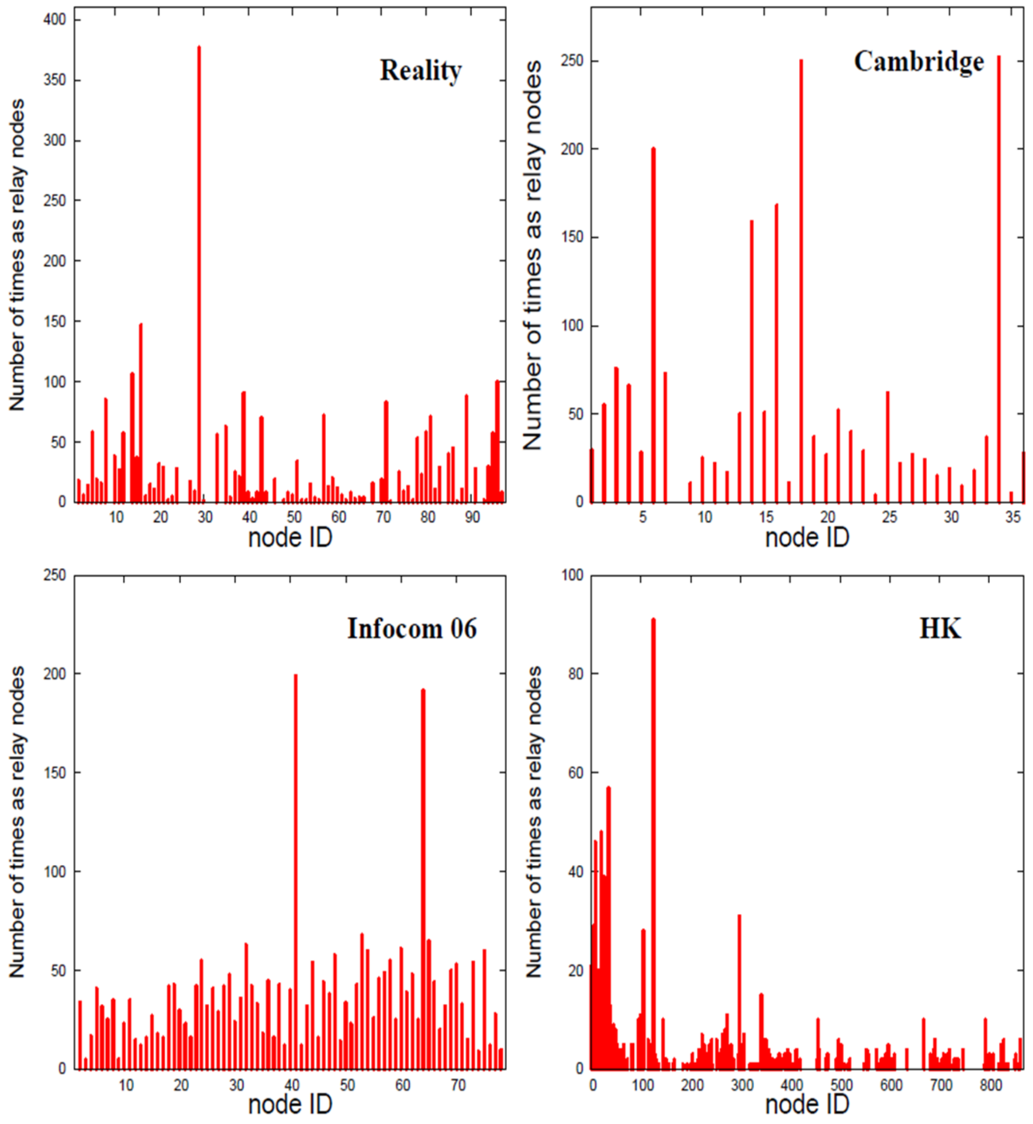
- Approach to calculate the centrality of each node:
  - Emulations of unlimited flooding with different distributed traffic patterns.
  - Count the number of times a node acts as a relay for other nodes on all shortest delay deliveries.
  - The number calculated (betweenness centrality) is normalized to the highest value.

- Use unlimited flooding since it can explore the largest range of delivery alternatives with the shortest delay.

Frequency of nodes as relays

- Shows the number of times a node falls on the shortest paths between all other node pairs (centrality of a node in the system)

- In order to design more efficient forwarding strategy we prefer to choose popular nodes as relays rather than unpopular ones



Outline

Evaluations of Different Forwarding Algorithms

- Comparison metrics:
  - **Delivery ratio** - the proportion of messages that have been delivered out of the total unique messages created
  - **Delivery cost** - the total number of messages (include duplicates) transmitted across the air. To normalize this, we divide it by the total number of unique messages created

Evaluations of Different Forwarding schemes

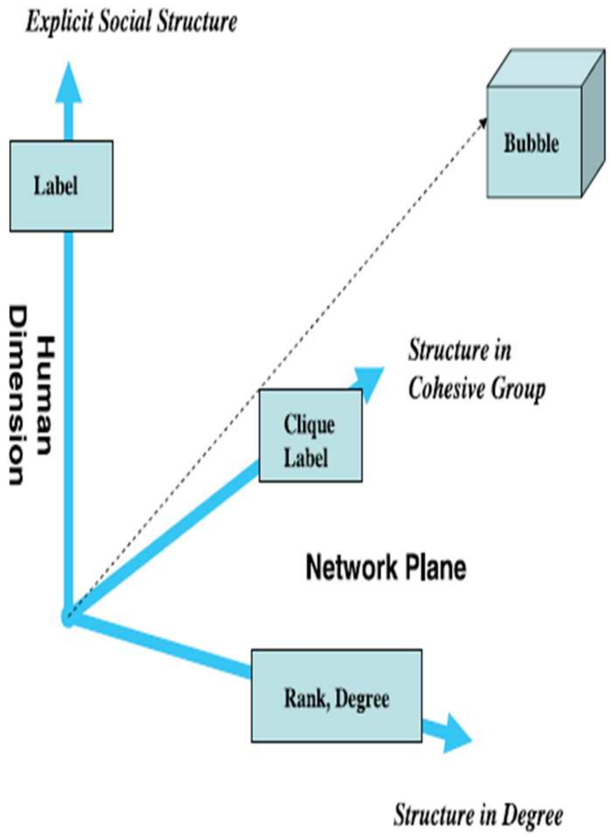
- WAIT - Hold on to a message until the sender encounters the recipient directly (paths with single hop)
  - lower bound for delivery and cost

- FLOOD - Messages are flooded throughout the entire system (length of the path is unlimited)
  - upper bound for delivery and cost

- MCP - Multiple-Copy-Multiple-Hop. We use 4-copy-4-hop MCP scheme in most of the cases
  - paths four hops are used (corresponding to a flooding algorithm with a Time-To-Live of 4 hops).
  - The most cost effective in terms of delivery and cost

Forwarding Algorithms

- LABEL - Messages are only forwarded to the nodes in the same community as the destination.
  - Done in human dimension.
- RANK – forwarding metric is the node centrality.
  - Done in the Network plane.
- Degree – forwarding metric is the node degree
  - Get the average of the degree of a node over certain time interval.
  - Used to select the forwarding nodes.

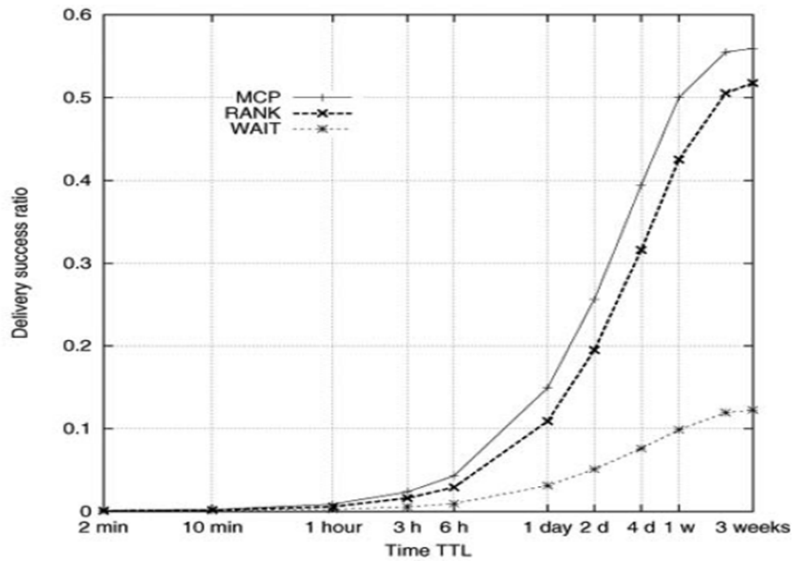


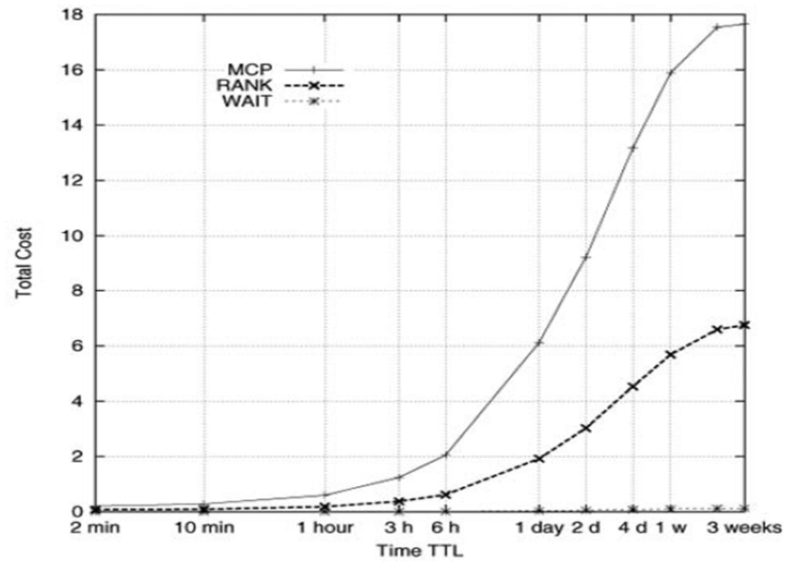
RANK Algorithm

- Each node knows only its own ranking and the rankings of those it encounters. Does not know the ranking of other nodes.
- It does not know which node has the highest rank in the system.
- keep pushing traffic on all paths to nodes which have a higher ranking than the current node, until either the destination is reached, or the messages expire.
- Works well in small and homogeneous systems.

Results of RANK Algorithm

- RANK is compared with MCP
- RANK performs as well as MCP for delivery.
- RANK performs at a cost 40% that of MCP, which represent marked improvement.
- The difference in cost between the two algorithms are not constant because they have different spreading mechanisms.





LABEL Algorithm

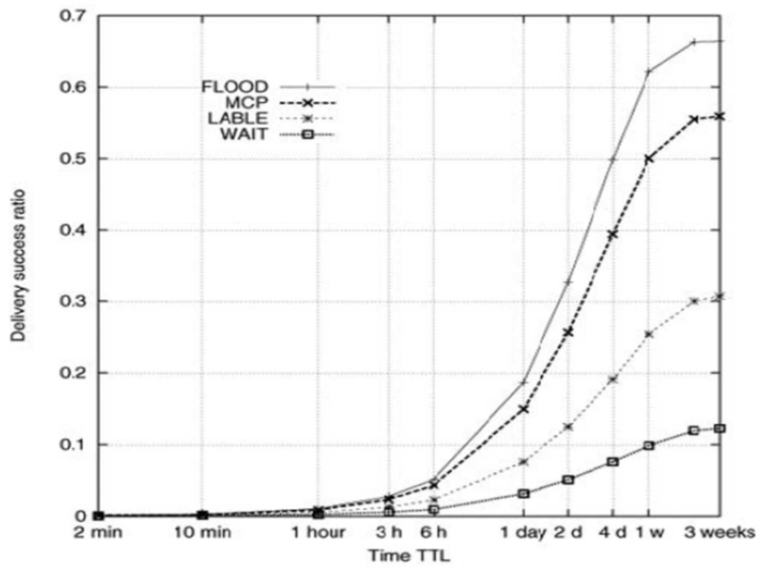
- Each node has a label that tells others its affiliation.
- Labels are used to forward messages to destinations.
- Next-hop nodes are selected if they belong to the same group (same label) as the destination.
- Significantly improves forwarding efficiency.

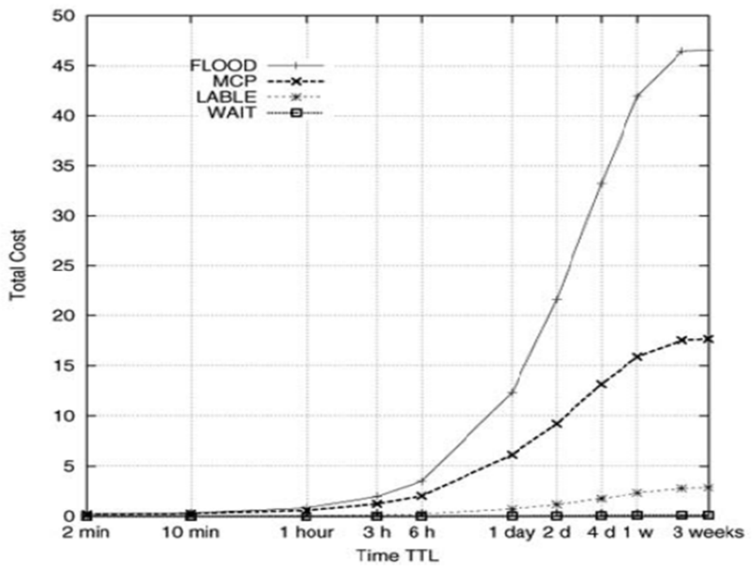
Limitation

- The lack of mechanisms to move messages away from the source when the destinations are socially far away (such as Reality).

Results of LABEL Strategy

- Evaluated on Reality Data set.
- Use the community information detected using K-CLIQUE algorithm to label the nodes.
- Achieves 55 % of the delivery ratio of the MCP strategy and only 45 percent of the flooding delivery although the cost is also much lower.
- However, it is not an ideal scenario for LABEL. In this environment, people do not mix as well as in a conference.





BUBBLE Algorithm

- BUBBLE combines the knowledge of community structure with the knowledge of node centrality to make forwarding decisions

- Two intuitions behind this algorithm:
  - People have varying roles and popularities in society, and these should be true also in the network
  - People form communities in their social lives, and this should also be observed in the network layer

BUBBLE Algorithm

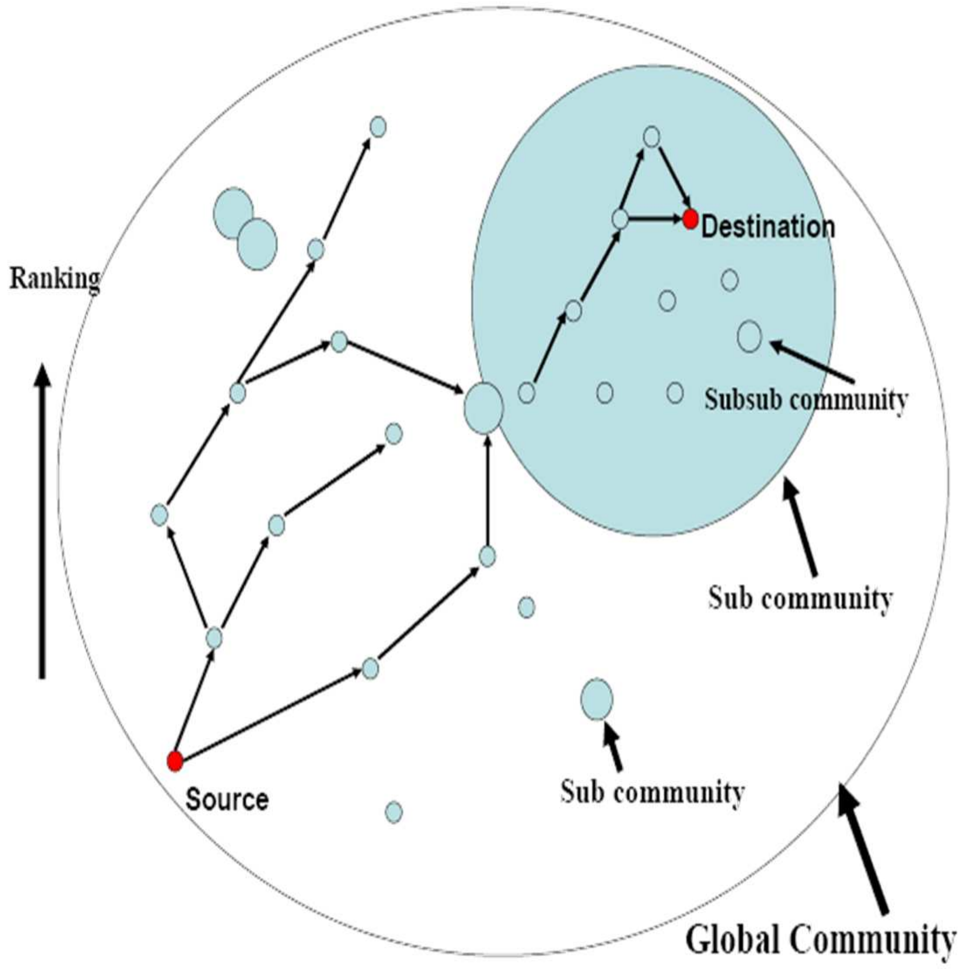
- BUBBLE is a combination of LABEL and RANK. It uses RANK to spread out the messages and uses LABEL to identify the destination community.

- Two assumptions:
  - Each node belongs to at least one community. Here, we allow single node communities to exist.
  - Each node has a global ranking (i.e., global centrality) in the whole system and also a local ranking within its community. It may belong to multiple communities and, hence, may have multiple local rankings.

Centrality Meets Community(BUBBLE)

- Messages bubble up and down the social hierarchy, based on the observed community structure and node centrality, together with explicit label data

- Avoids the occurrence of dead ends encountered with global ranking schemes.



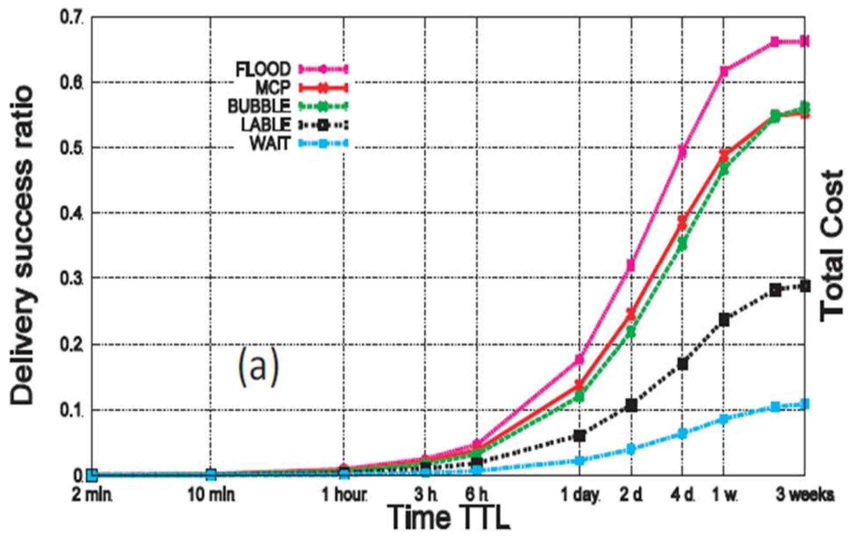
BUBBLE Algorithm

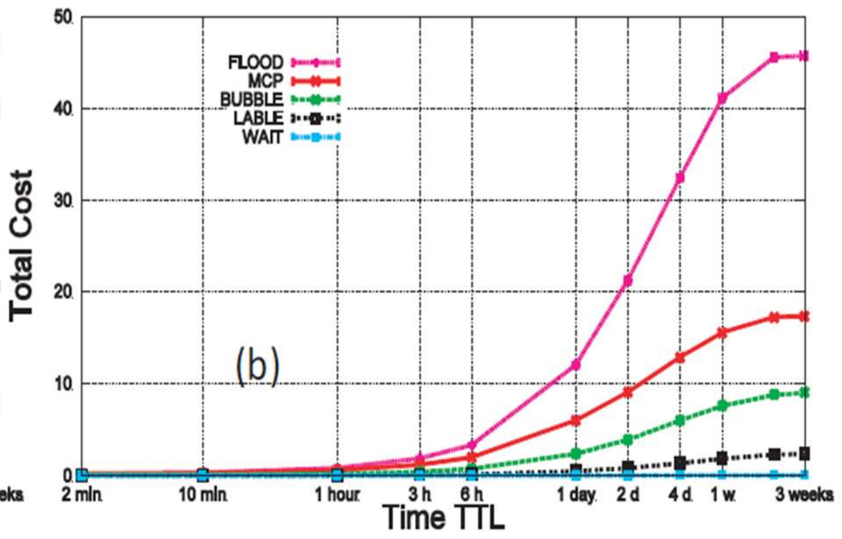
Forwarding is carried out as follows:

- The source node first bubbles the message up the hierarchical ranking tree using the global ranking, until it reaches a node which is in the same community as the destination node.
- The local ranking system is used instead of the global ranking, and the message continues to bubble up through the local ranking tree until the destination is reached or the message expires.
- Doesn’t require every node to know the ranking of all other nodes in the system.
- Require to be able to compare ranking with the node encountered.

Multiple-Community Case

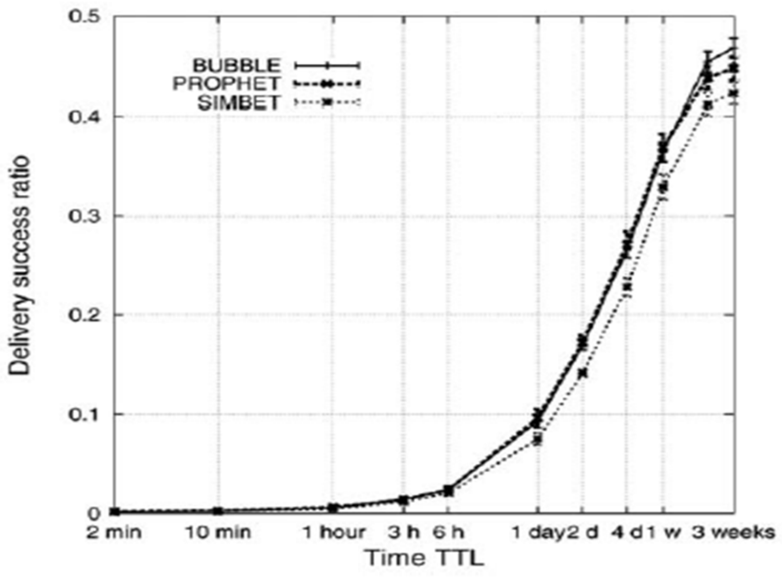
- Comparisons of several algorithms on Reality dataset
- flooding achieves the best for delivery ratio, but the cost is :
  - 2.5 times that of MCP
  - 5 times that of BUBBLE
- BUBBLE is very close in performance to MCP and even outperforms it when the time TTL of the messages is allowed to be larger than 2 weeks
- BUBBLE cost is only 50% that of MCP

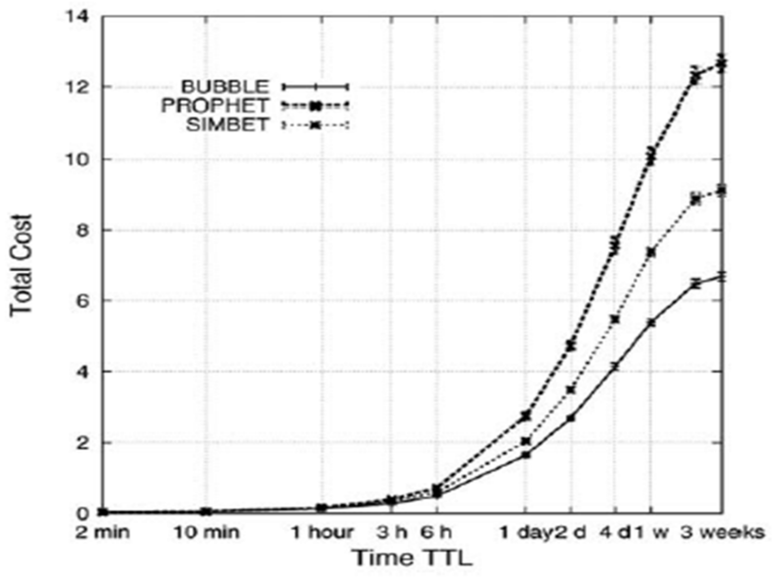




Comparisons of BUBBLE, PROPHET and SimBet (Reality dataset)

- PROPHET - Uses the history of encounters and transitivity to calculate the probability that a node can deliver a message to a particular destination
- BUBBLE achieves a similar delivery ratio to PROPHET, and 10 % better than the SimBet
- Only half of the cost of PROPHET and 70% of cost of SimBet.
- Similar significant improvements by using BUBBLE are also observed in other datasets, these demonstrate the generality of the BUBBLE algorithm





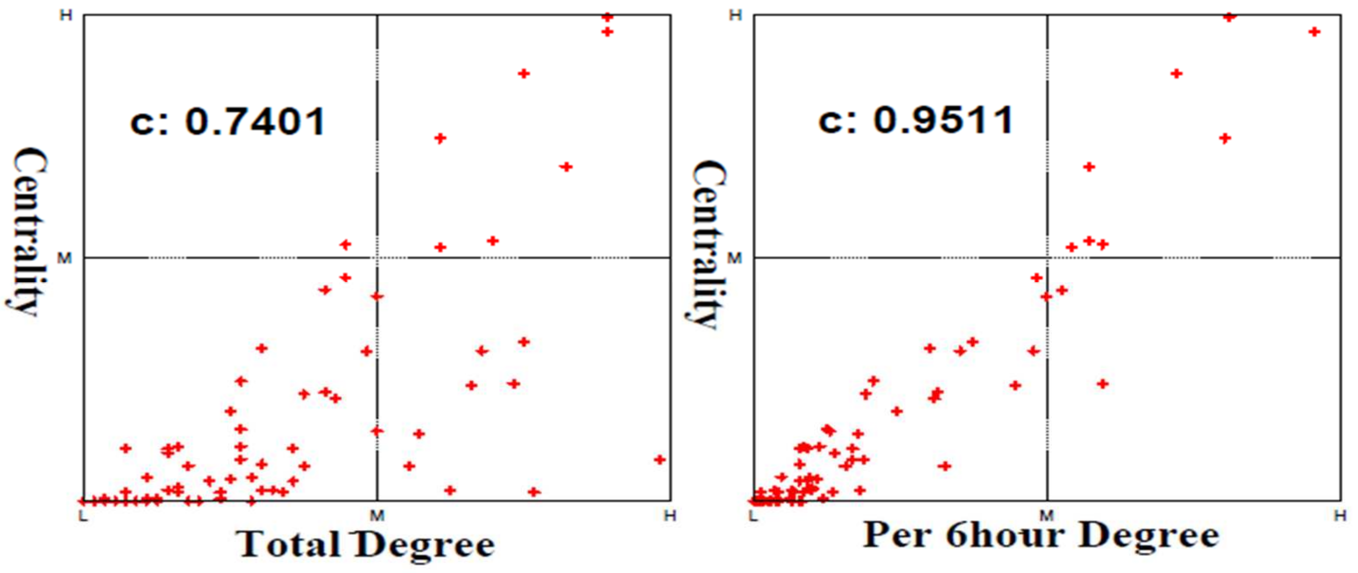
Outline

DiBuBB Algorithm

- For practical applications we need BUBBLE be implemented in a distributed way
  - Each device should be able to:
    - detect its own community
    - calculate its centrality values
- Use the distributed K-CLIQUE algorithm to detect local community (detecting accuracy up to 85% of the centralized one).
- Calculating centrality values (next slide).
- Besides that, it operate exactly like BUBBLE

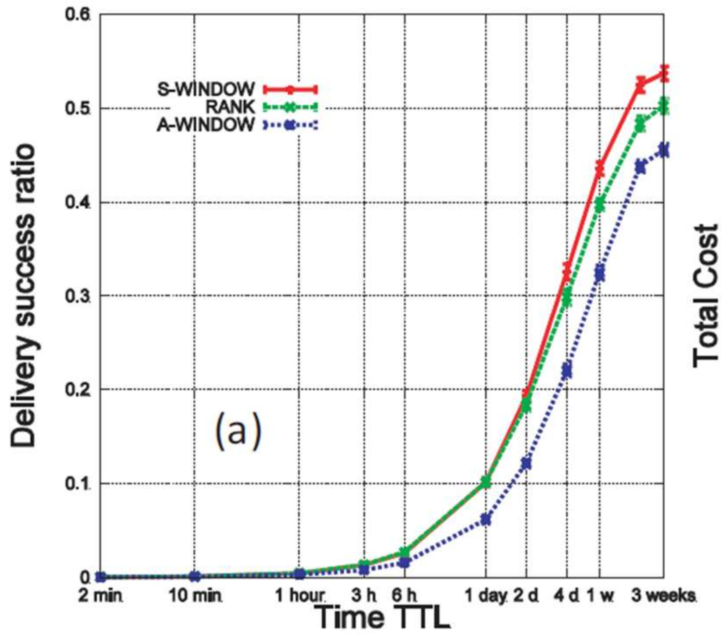
Distributed BUBBLE

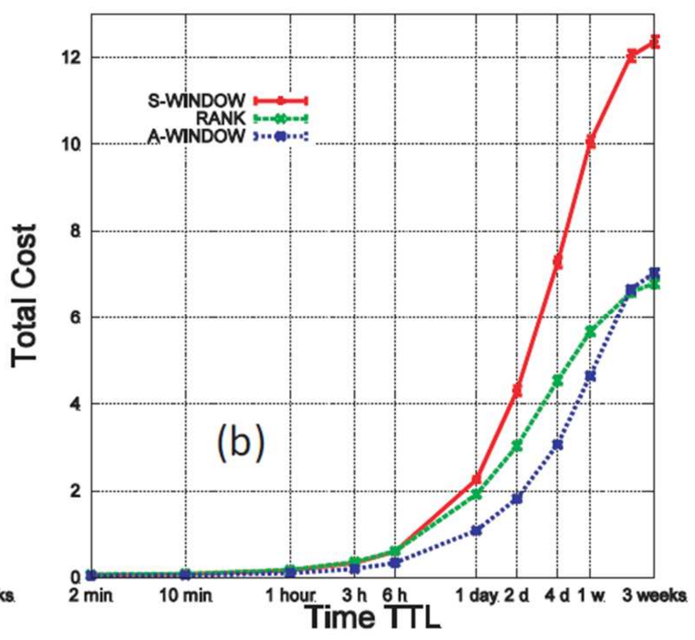
- Trace analysis conclusions:
  - Total degree (unique nodes seen by a node throughout the experiment period) is not a good approximation of the node centrality
  - The degree per unit time (for example the number of unique nodes seen per 6 hours) and the node centrality have a high correlation value



Approximating Centrality

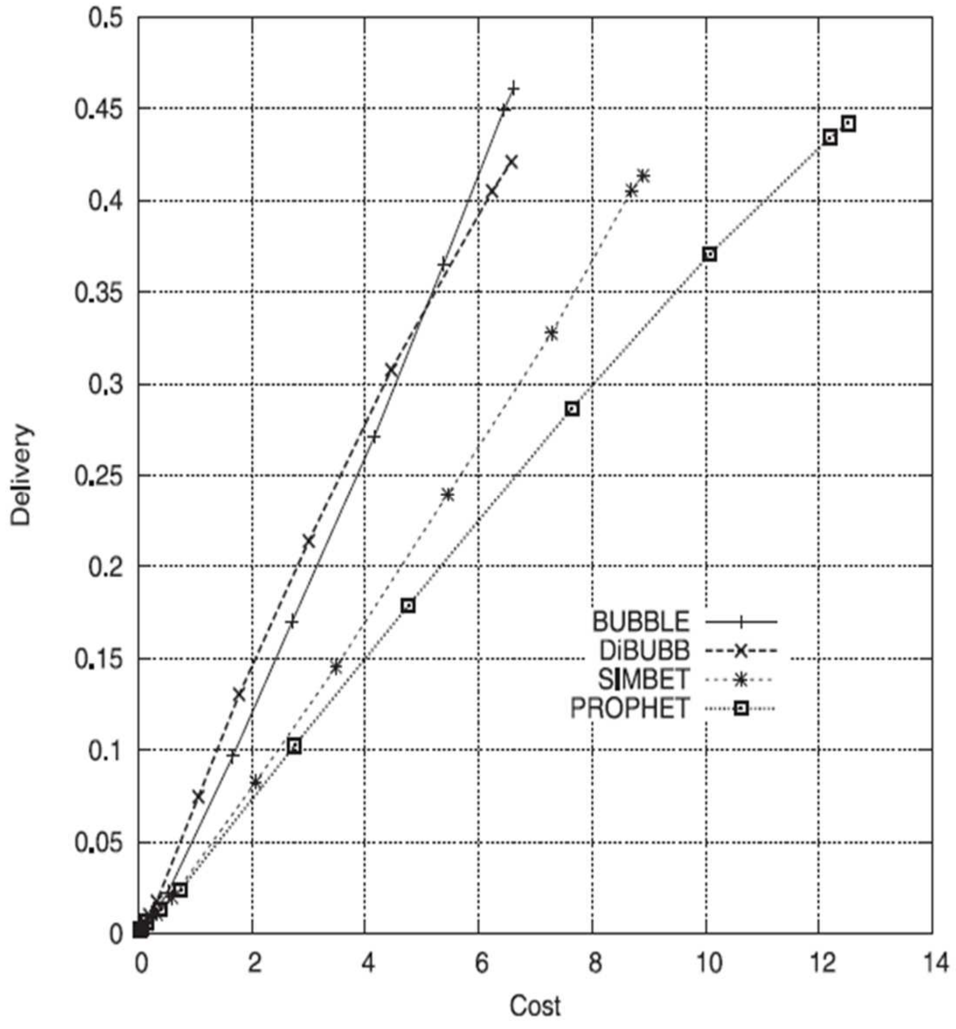
- S-Window achieves maximum of 4% improvement in delivery ratio than RANK, but at double the cost
- C-Window does not achieve as good delivery as RANK (not more than 10% less in term of delivery), but it also has lower cost





DiBUBB Algorithm

- Larger slope = more efficient algorithm
- DiBUBB is very close to BUBBLE in delivery per cost.
- Outperforms PROPHET and SimBet



CONCLUSIONS

- It is possible to detect characteristic properties of social grouping in a decentralised fashion from a diverse set of real world traces
- Demonstrated that community and centrality social metrics can be effectively used in forwarding decisions
- BUBBLE algorithm has similar delivery ratio, but much lower resource utilization than flooding, control flooding, and PROPHET and SimBet.
- C-Window is easy to implement in reality and has similar delivery and cost to RANK (pre-calculated centrality), which is why it was chosen for DiBuBB

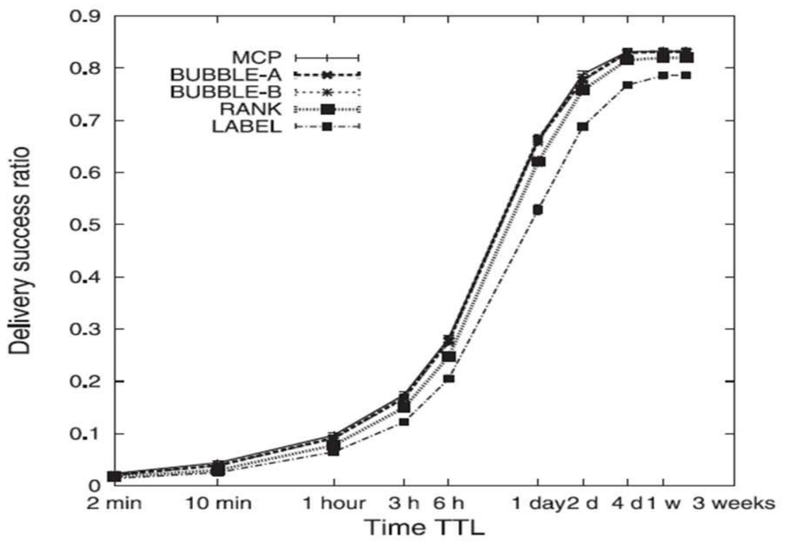
Thank you

Backup Slides

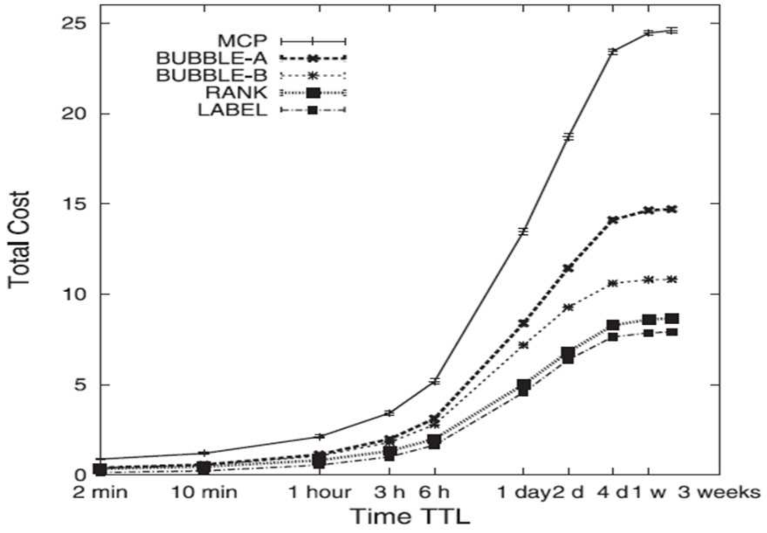
BUBBLE Algorithm

- A modified version of this strategy is that whenever a message is delivered to the community, the original carrier can delete this message from its buffer to prevent it from further dissemination.

- Assumes that the community member would be able to deliver this message.
- This protocol with deletion strategy BUBBLE-B, and the original algorithm introduced above BUBBLE-A.



(a)

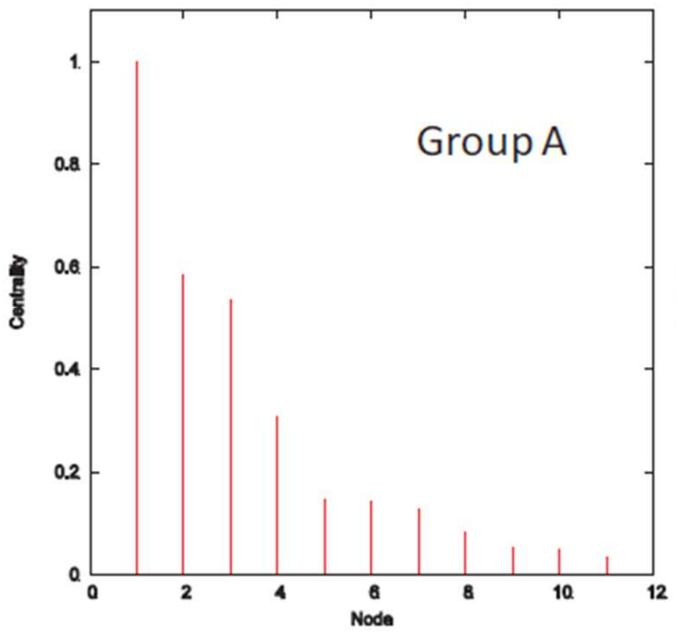


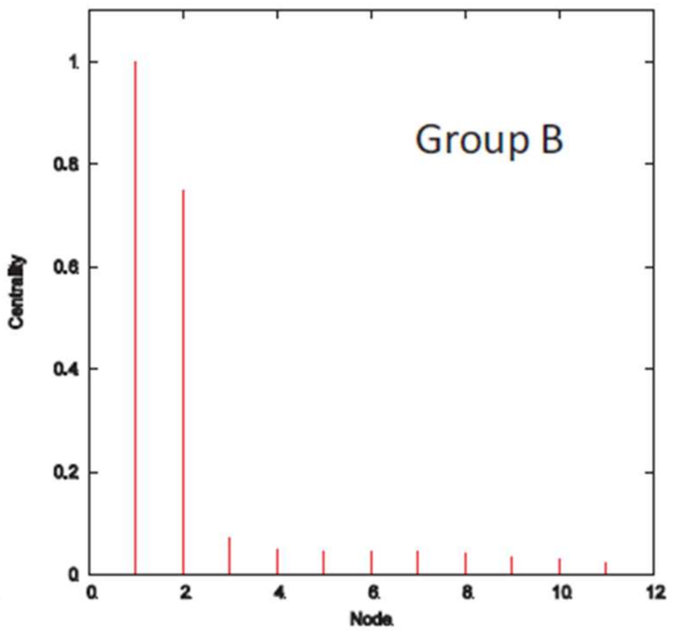
(b)

Two-Community Case

- Cambridge data can be divided into two communities :
  - undergraduate year 1 (Group A)
  - year 2 (Group B)

- Centrality of nodes within each group:
  - traffic is created only between members of the same community
  - only members in the same community are chosen as relays for messages

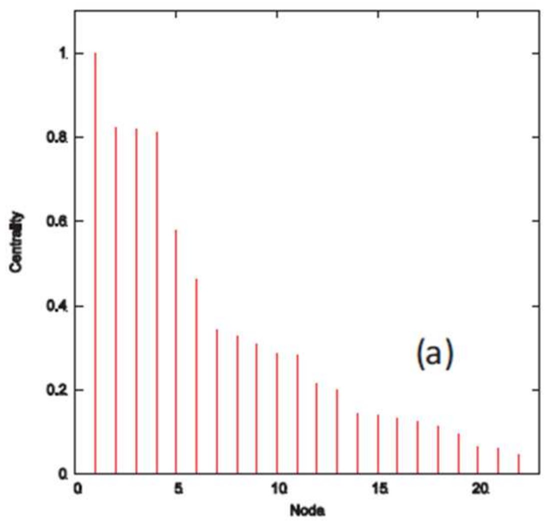


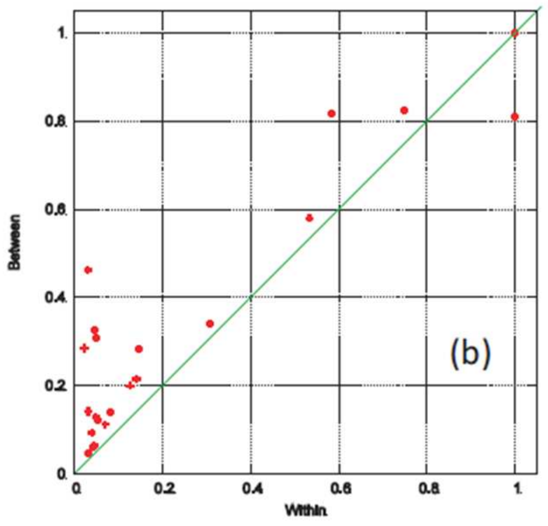


Two-Community Case

- Figure (a) shows the individual node centrality when traffic is created from one group to another
- Figure (b) shows the correlation of node centrality within an individual group and inter-group centrality (deliveries to other group, but not to its only group)

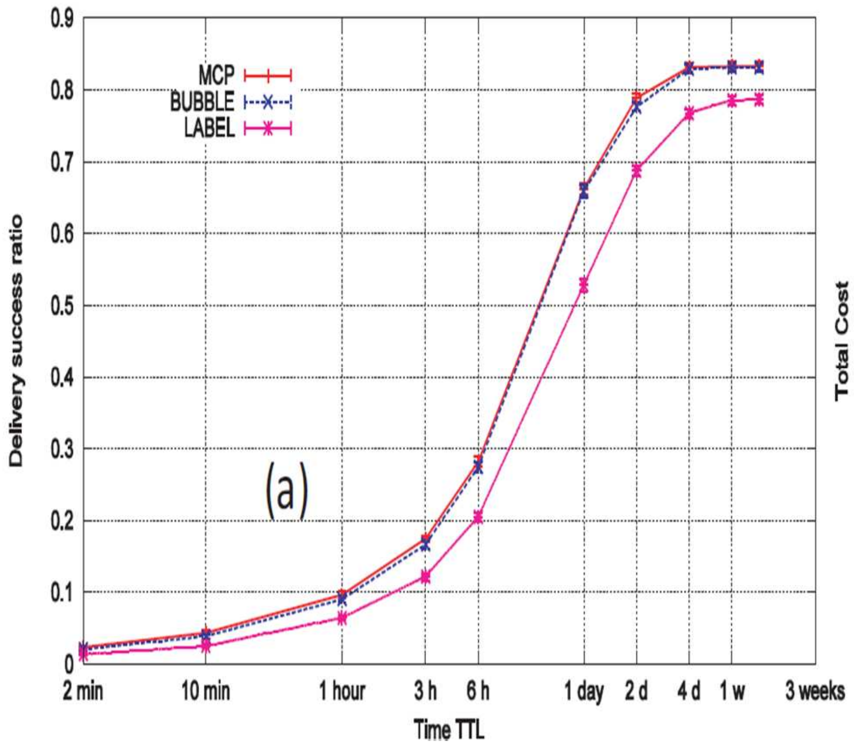
- Points lie more or less around the diagonal line:
  - the inter- and intra- group centralities are quite well correlated
  - active nodes in a group are also active nodes for inter-group communication

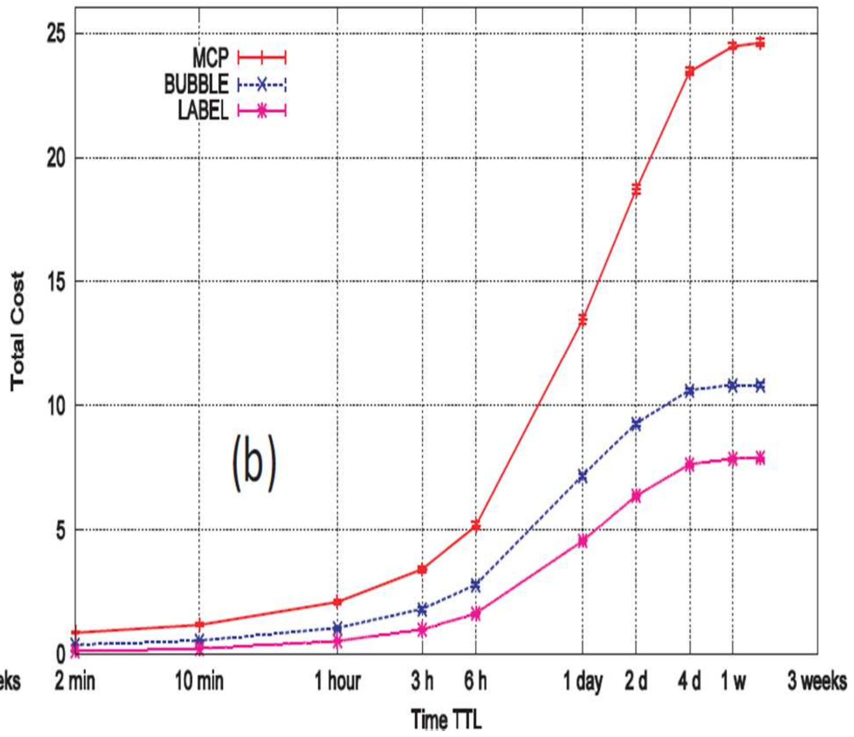




Two-Community Case

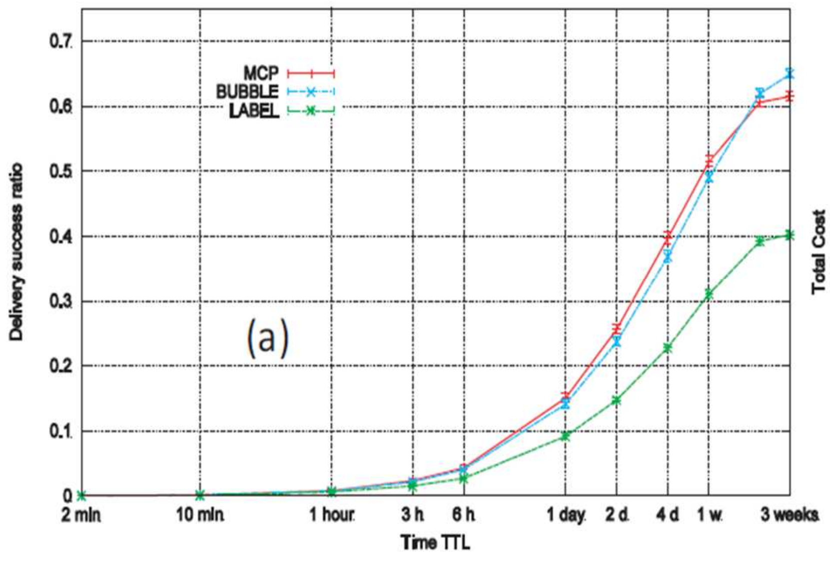
- Comparisons of several algorithms on Cambridge dataset, delivery and cost:

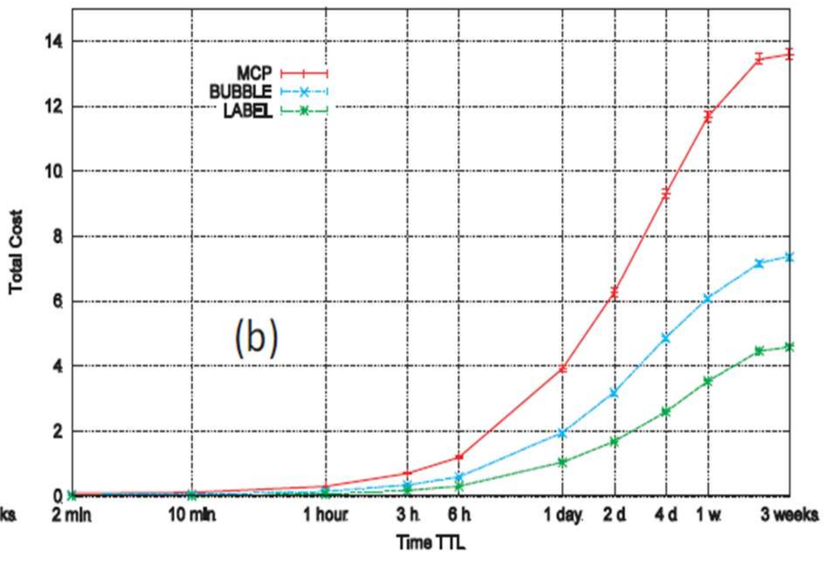




Multiple-Community Case

- Use the Reality dataset
  - There is a total 8 groups within the whole dataset
  - Within each individual group, the node centralities demonstrate diversity similar to the *Cambridge case*
  - First isolate just one group, consisting of 16 nodes, single group case:
    - BUBBLE performs very similarly to MCP most of the time and even outperform MCP when the time TTL is set to be longer than 1 week (delivery success ratio)
    - BUBBLE only has 55% of the cost of MCP





Difference between PROPHET and BUBBLE

- PROPHET relies on encountering history and transient delivery predictability to choose relays. This can efficiently identify the routing paths to the destinations,
- but the dynamic environment may result in many nodes having a lot of slightly fluctuation of probabilities.
- This results in more redundant nodes being chosen as relays, which can be reflected from the delivery cost.
- BUBBLE uses social information and, hence, filters out these noises due to the temporal fluctuations of the network.

Difference between BUBBLE and SimBet

- SimBet can successfully leverage social context, but it fails in identifying the sequence of using betweenness and similarity.
- BUBBLE explicitly identifies centrality and community, and first uses centrality metric to spread out the messages and then uses community metric to focus the messages to the destinations.
- This approach effectively guarantees a high delivery ratio and a low delivery cost.

Approximating Centrality

- C-Window is easy to implement in reality and has similar delivery and cost to RANK (pre-calculated centrality), which is why it was chosen for DiBuBB
- S-Window, and C-Window can approximate the pre-calculated centrality quite well
- Running a set of RANK emulations on more datasets, but using the centrality values of the Multiple-Community Case showed that the delivery ratio and cost of RANK on the new datasets is as good as in the original dataset
- These results imply some level of human mobility predictability, and show empirically that past contact information can be used in the future