Crowdsourcing mobile networks

Katia Jaffrès-Runser
University of Toulouse, INPT-ENSEEIHT,
IRIT lab, IRT Team

Jiaotong University, Shanghai
Fête Nationale / Bastille day, July 14, 2015

N7 Engineering school, Toulouse

The smartphone phenomenon

- Multiple sensing and communication capabilities
  - Sensors, camera, GPS, microphone
  - 3G, WiFi, Bluetooth, etc.
  - Storage capabilities (several Gbytes)
  - Computing power

Mobile Traffic is growing constantly

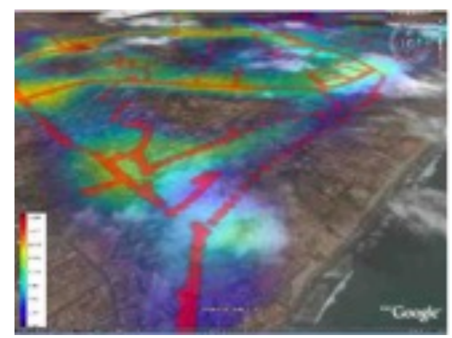
- Increasing volume of mobile data between 2014-2018
  - "…worldwide mobile data traffic will increase nearly **11-fold** over the next four years and reach **an annual run rate of 190 exabytes** (10^{18}) by 2018…"
  - 54% of mobile connections will be 'smart' connections by 2018

[Cisco VNI Global Mobile Data Traffic Forecast (2013-2018)]

+

In 2013, 4.1 billion users worldwide

=



Next Big Networking Challenge: meet traffic demand !

1. If data is not delay sensitive:
   - e.g. Videos, Application / system updates, music, podcasts, etc.

Leverage opportunistic encounters to route
or flood **delay tolerant** data hop by hop

Benefit: Reduce downloads from infrastructure wireless network

2. If several connectivity options exist:
   - e.g. 3G/4G, WiFi, Femto cells

Offload / Pre-fetch data using
the 'best' available connectivity, at the best time and location

Benefit: Load balancing between available infrastructures

Crowdsourcing (part of) this huge network

- This huge network of users is constantly active.
  - The context each user is evolving in is changing
  - The content each user is consuming / sending is evolving as well
- To provide the next intelligent data communications, ***we need to understand how this network evolves***
- How is this big dynamic network evolving?
  - Getting network traces
  - Model the interactions of this dynamic network to capture its evolution
- How to get network traces?
  - Network operator monitoring
  - **Crowdsourcing using smartphone capabilities**

Outline of this talk

1. Building a Mobile app for crowdsourcing

2. First statistics of Macaco Project and trace description

3. Use case exemple: Classifying social interactions from such contact traces

EU CHIST-ERA MACACO Project

Mobile context-Adaptive CAching for COntent-centric networking

www.macaco.inria.fr

INRIA (Paris), University of Toulouse, SUPSI (Lugano),
University College London, CNR-IEIIT (Torino), UFMG (Brazil)

Crowdsourcing Mobile app

Goal : Sample user context and content data

- Runs in background on **volunteer** phone users
  - Monitors different sensors periodically (5 mins)

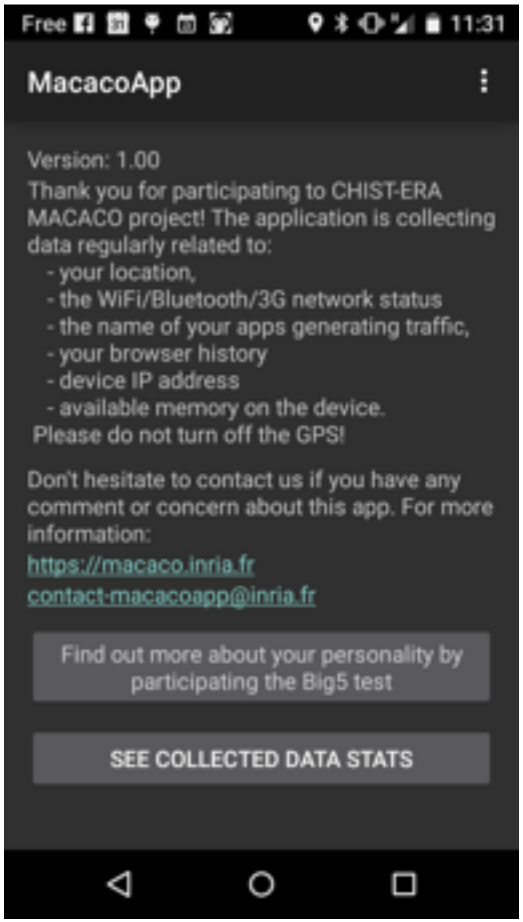
- Should be **seamless** with respect to regular phone usage
  - **Upload data** to our servers before memory is full
    - Full memory = no reactivity
    - But : does not ruin the 3G data plan !

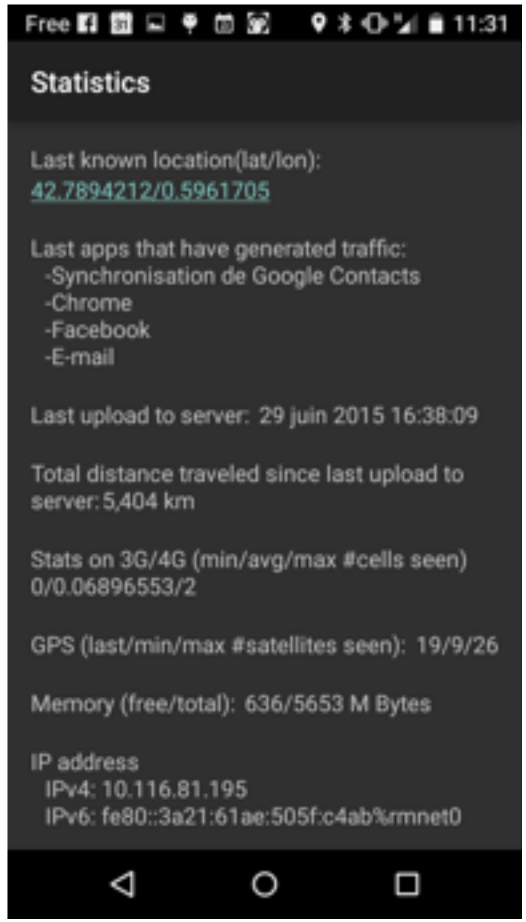
      Favor uploads on WiFi

  - **Energy** constraint !!
    - Monitoring all sensors is costly

The macaco App

www.macaco.inria.fr



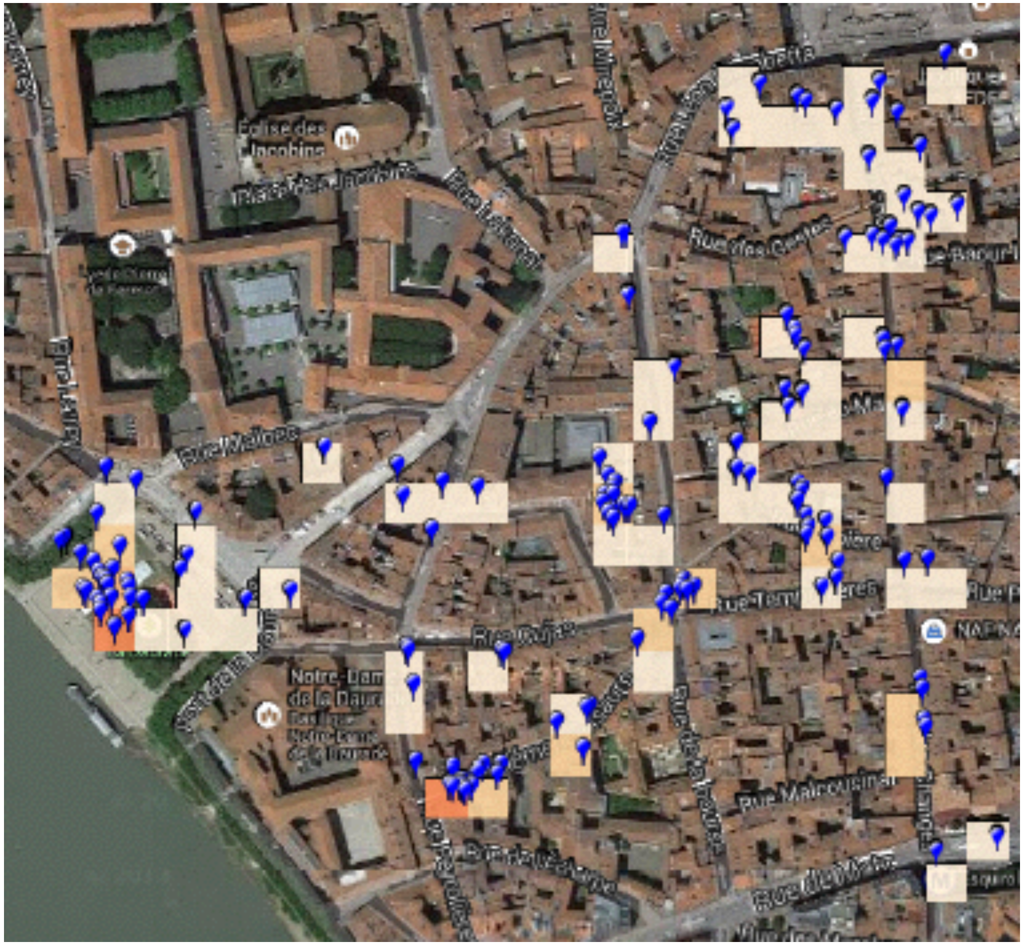


Macaco App

macaco

Measured data every minute :

- Context data
  - Location (GPS, Internet)
  - WiFi connectivity
  - Bluetooth connectivity
  - Cellular network towers
  - Battery discharge
  - Accelerometer
  - Big 5 personality test



- Content data
  - Name of applications that have generated traffic
  - Browser history
  - Name of applications run

Main issue: getting volunteers :-)

- **Privacy** issues (discussion with CNIL)
  - Keep data within project partners,
  - Have data anonymized (hashed IMEI - location)
  - Limit storage duration of non-anonymized data use
  - Option to remove its own data from the collection

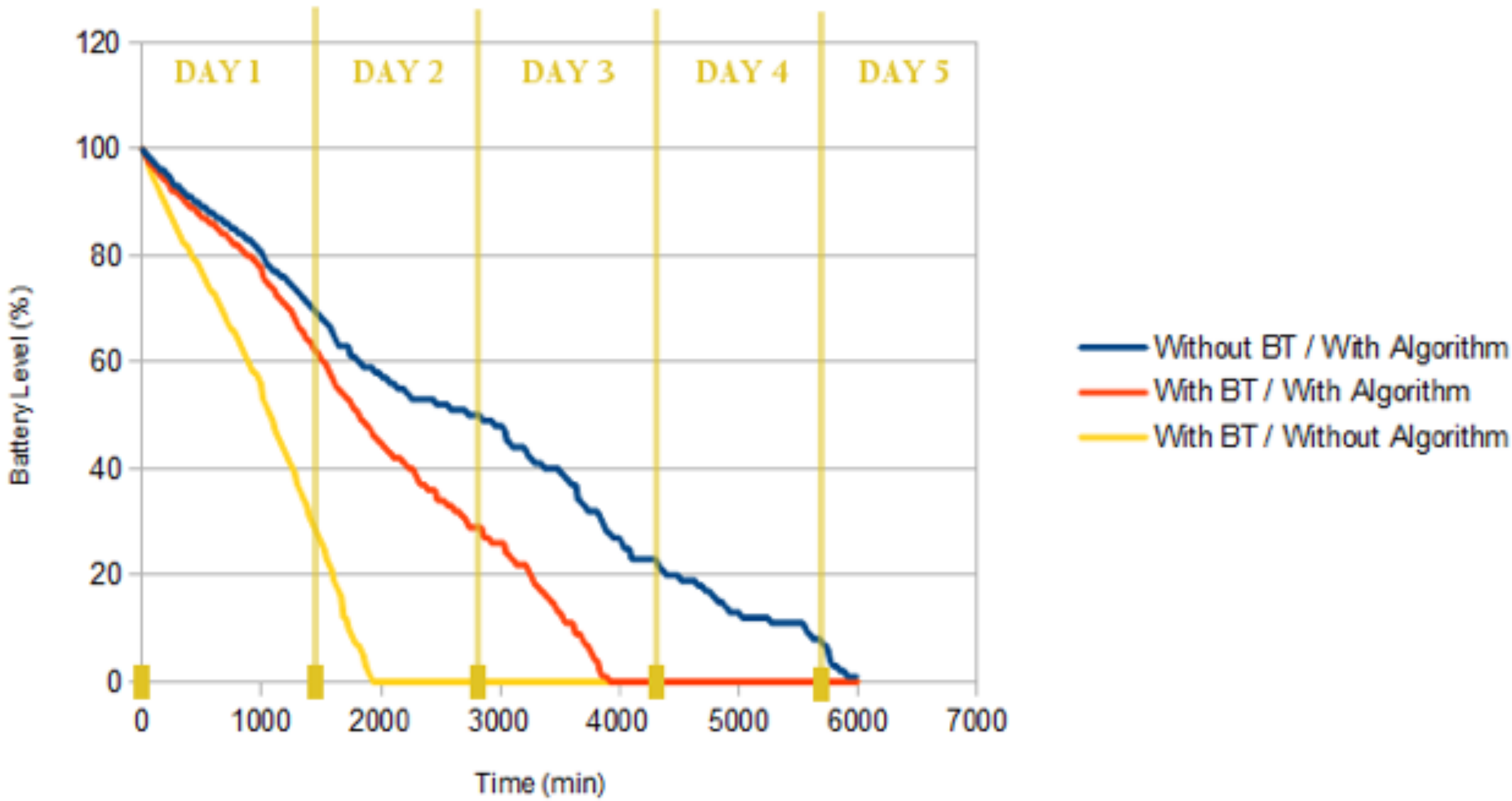
- **Energy efficient app design**
  - Keep the volunteers using the app

- Provide **a motivation** for participating
  - Added value of the app (e.g. visualize its own data, game, …)
  - Financial retribution (voucher)
  - Lottery
  - For the greater good :-) …

Energy depletion with movement detection

% remaining battery if the phone stands still

- w./w.o. movement detection
- w./w.o. bluetooth measurements



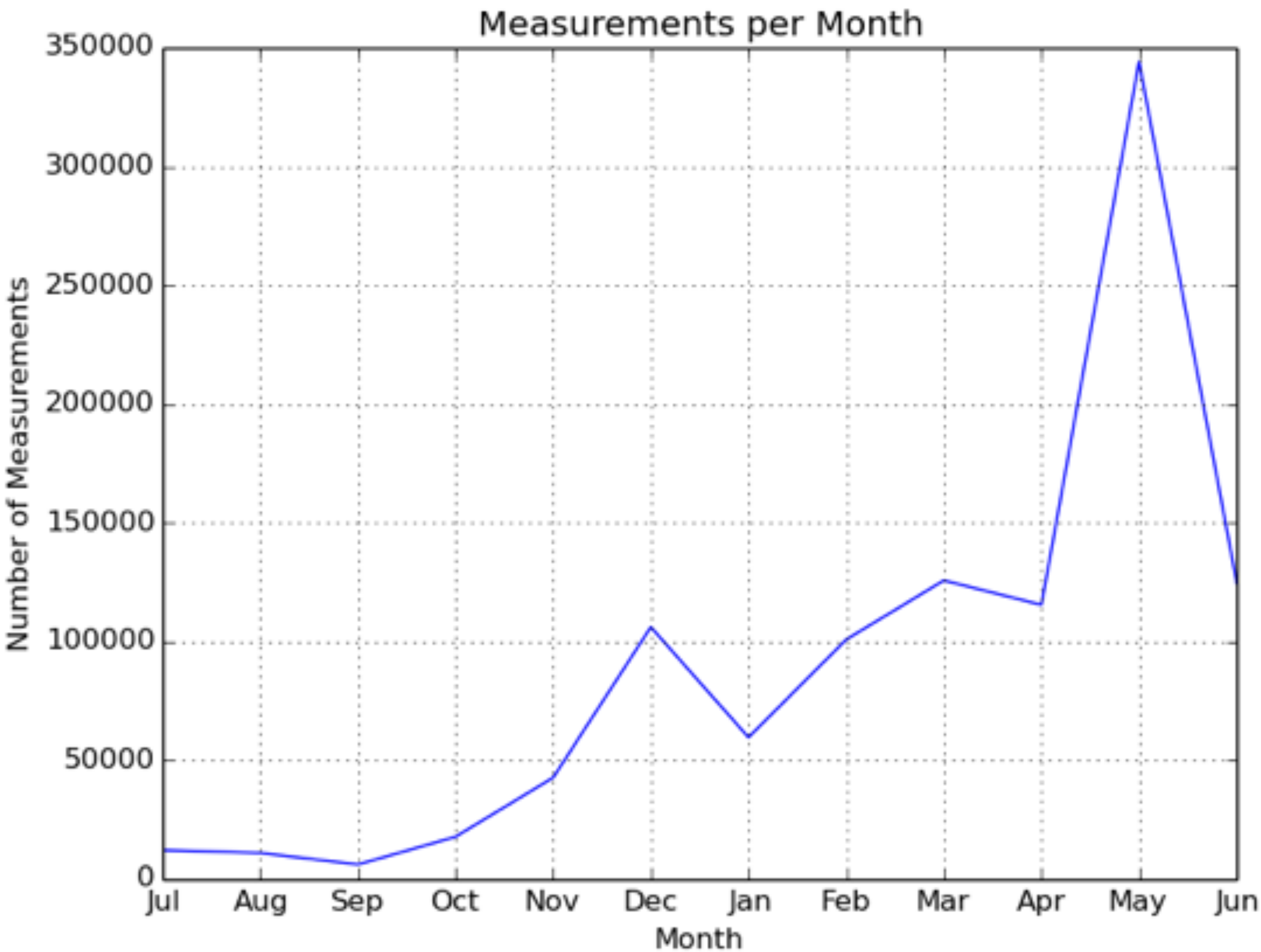
First Macaco data statistics

- Collected with MacacoApp
  - Up to now, for one year (2014 July – 2015 June)
  - 57 devices over one year
  - 1,069,083 Measurements

- Top contributors:

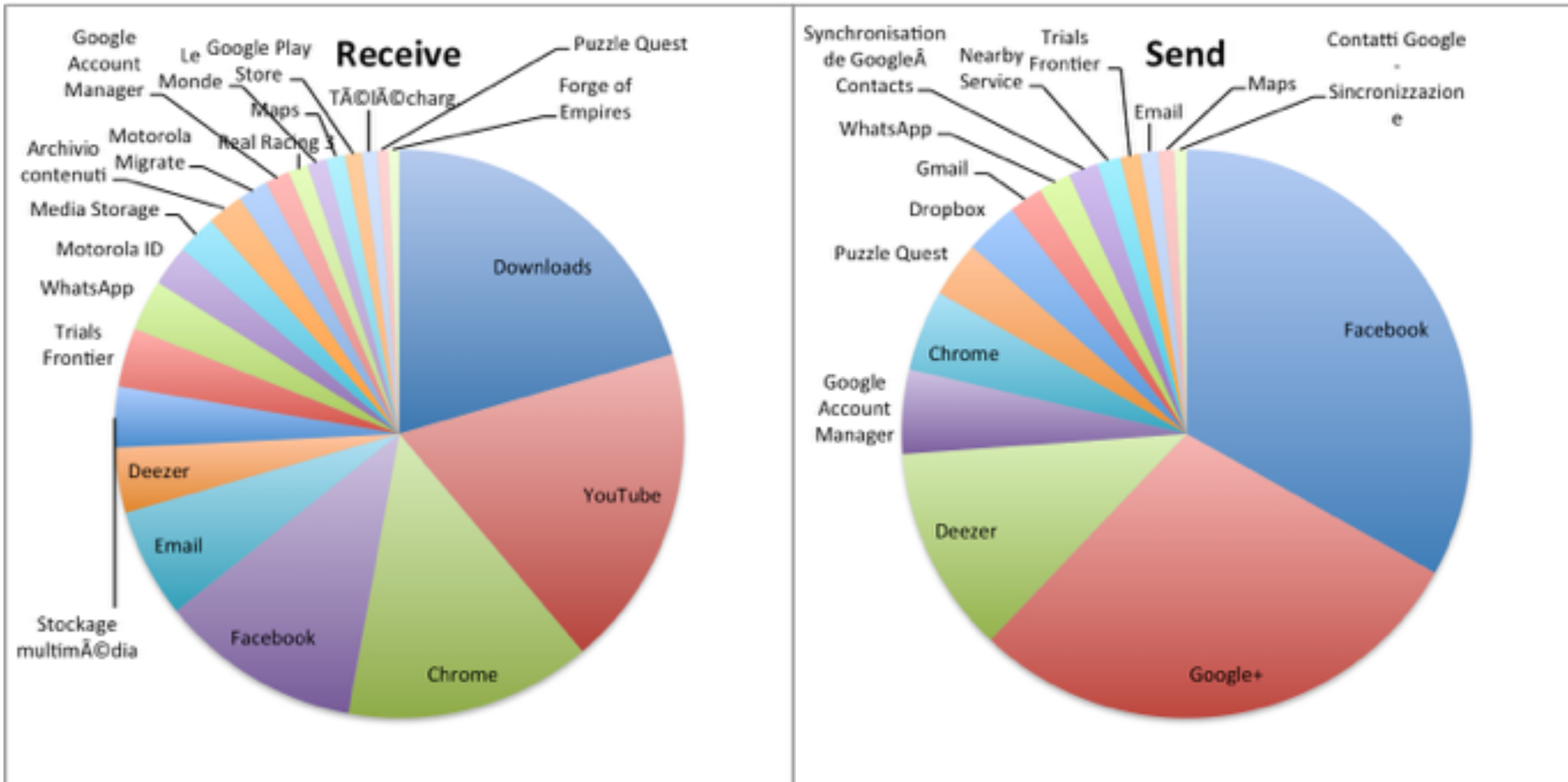
Hash(IMEI)	Period	# measurements
203a...	2014-11-04 - 2015-06-22	187879
bacd...	2014-08-27 - 2015-06-22	145619
f1d9...	2014-08-06 - 2015-06-20	126215
46bd...	2014-08-19 - 2015-06-13	119634
4517...	2012-01-01 - 2015-06-22	65812
e6d2...	2015-05-05 - 2015-06-22	59997
008f...	2015-05-07 - 2015-06-22	55059

First Macaco data statistics



First Macaco data statistics

- Total traffic download: 55534 MB
- Total traffic upload: 10679 MB



Now, what can we do with this data?

- Mobility analysis:
  - Plot the **trajectory** of one user.
  - Extract **points of interest** for users: Home, Office, School …
  - Find regular patterns in mobility (working / non working days)
- Understand App usage of the phone:
  - **Which** apps are the favorite ones?
  - **When** are these apps used? How often? When in the week?
  - **Where** do the users start these favorite apps?
- Understand the networks around:
  - With WiFi or Bluetooth: map location of APs, BT networks
  - Understand **when** is WiFi/3G/Bluetooth available / actually used

Macaco Mobility trace (for a given device)

id	timestamp	provider	accuracy	Latitude	Longitude
750190	1431003983411	network	37	43602061	1455570
750244	1431007275360	network	20	43601951	1455536
750248	1431007463084	network	36	43603029	1454255
750249	1431007576600	network	30	43604800	1456844
750251	1431007639234	network	37	43606018	1456409
750252	1431007767307	gps	32	43609879	1455394
750254	1431007824257	network	40	43611817	1457947
750254	1431007877331	network	30	43612946	1456268
750256	1431007939888	network	32	43617212	1455507
750256	1431007998466	network	30	43619232	1456513
750257	1431008056744	network	27	43620054	1457180

Information:

- Id of measurement in whole database
- Timestamp (Unix epoch - since Jan 1st. 1970)
- Origin of measurement (gps / network)
- Measurement accuracy
- Latitude and Longitude

Macaco Apps trace (for a given device)

id	timestamp	UID	Label	Packages	rx_traffic	tx_traffic

685383 1430990451320 10115 "Reconnaissance vocale" com.vlingo.client.samsung
0 1224

599898 685385 1430990571708 10062 "Google Play Store" com.android.vending
4991 2750

599899 685385 1430990571708 10015 "Services Google Play"
com.google.android.gsf.login com.google.android.location com.google.android.gms
com.google.android.syncadapters.bookmarks com.google.android.syncadapters.contacts
com.google.android.gsf 4815 1039

Information:

- **Id** of measurement
- **Timestamp** (Unix epoch - since Jan 1st. 1970)
- **UID** : app identifier on Android
- **Label** in Android
- Name of installation **package** in Android
- **Number of bytes downloaded** since last measurement
- **Number of bytes uploaded** since last measurement

Macaco Bluetooth trace (for a given device)

_	id	timestamp	NetworkName	MAC@	bond_state	RSSI (dBm)
	9193333	1431154755555	scala rider Q3	00:0A:9B:2A:81:91	0	-77
	9194850	1431242655321		00:03:19:1D:F4:85	0	-91
	9196198	1431320962110		6C:A7:80:87:F3:DB	0	-83
	9196202	1431321198797	99249	00:07:80:76:E1:6C	0	-76
	9196203	1431321259218		00:0B:CE:09:A1:98	0	-94
	9196432	1431334999027	EVERTEK E29	0C:2E:B7:E0:62:50	0	-93
	9196434	1431335116777	EVERTEK E29	0C:2E:B7:E0:62:50	0	-95

Information:

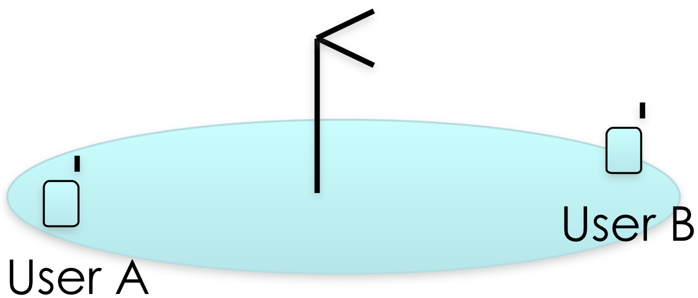
- Id of measurement in whole database
- Timestamp (Unix epoch - since Jan 1st. 1970)
- Bluetooth network name
- Measurement accuracy
- MAC address (BT device physical address)
- Bond_state : connected to ? 0/1
- RSSI received signal strength (dBm)

Network Traffic
Device1 WiFi

Network Traffic mapping
Device1 - 3G traffic

How to exploit such datasets?

- Other open datasets exist (cf. Crawdad http://crawdad.cs.dartmouth.edu/)
- Different types of temporal contact measurements
  - Measure a direct link between User A and B (e.g. Bluetooth, WiFi Direct connectivity)
  - Assume a link exists between User A and User B if they are connected to the same WiFi access point



  - Measure location of users (GPS): if users are close enough, assume they are connected

Example open data sets

Data collection to build *contact traces*

- Log the contact time and duration of a node to an access point
- Log the GPS coordinates of mobile nodes regularly

Derive a time-varying contact graph

Dataset	Local	# entities	Duration	Type	Avg. # encounters/ node/day
Dartmouth [1]	campus	1156	2 months	Individuals	145.6
USC [2]	campus	4558	2 months	Individuals	23.8
San Francisco [3]	City	551	1 month	Cabs	834.7

- Dartmouth and USC collect connection dates/durations to WiFi APs,
- San Francisco collects GPS locations of taxi cabs.

[1] T. Henderson et al. "The changing usage of a mature campus-wide wireless network," in Proc. of ACM MobiCom 2004

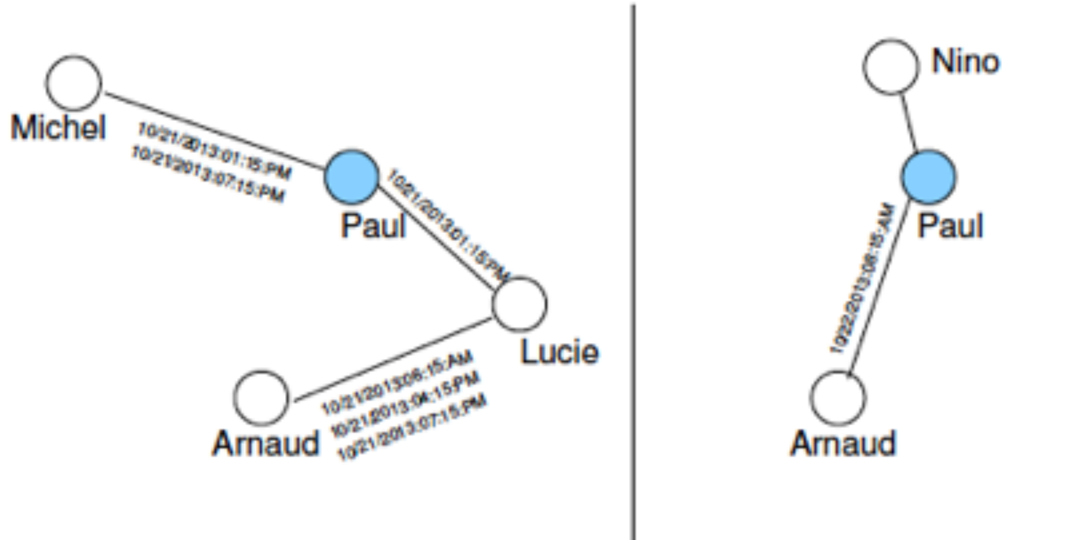
[2] W. jen Hsu et al. "Impact: Investigation of mobile-user patterns across university campuses using wlan trace analysis," CoRR, vol. abs/cs/0508009, 2005

[3] A. Rojas et al. "Experimental validation of the random waypoint mobility model through a real world mobility trace for large geographical areas," in Proc. of the 8th ACM MSWiM 2005

Graphs extracted from contact traces

Two possible representations

1. δ event graph: $\mathcal{G}_k(\mathcal{V}_k, \mathcal{E}_k)$
   There is an edge in $\mathcal{E}_k$ if contact within $\delta = 1$ day for instance.



Day 1 event graph $\mathcal{G}_1(V_1, E_1)$

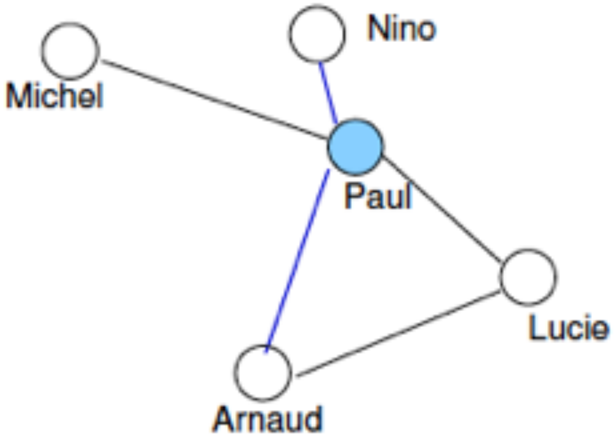
Day 2 event graph $\mathcal{G}_2(V_2, E_2)$

2. Accumulative graph $G_t(V_t, E_t)$

Graphs extracted from contact traces

Two possible representations

1. δ event graph: $\mathcal{G}_k(\mathcal{V}_k, \mathcal{E}_k)$
   There is an edge in $\mathcal{E}_k$ if contact within $\delta = 1$ day for instance.
2. Accumulative graph $G_t(V_t, E_t)$: $G_t = \{\mathcal{G}_1 \cup \mathcal{G}_2 \cup \ldots \cup \mathcal{G}_t\}$



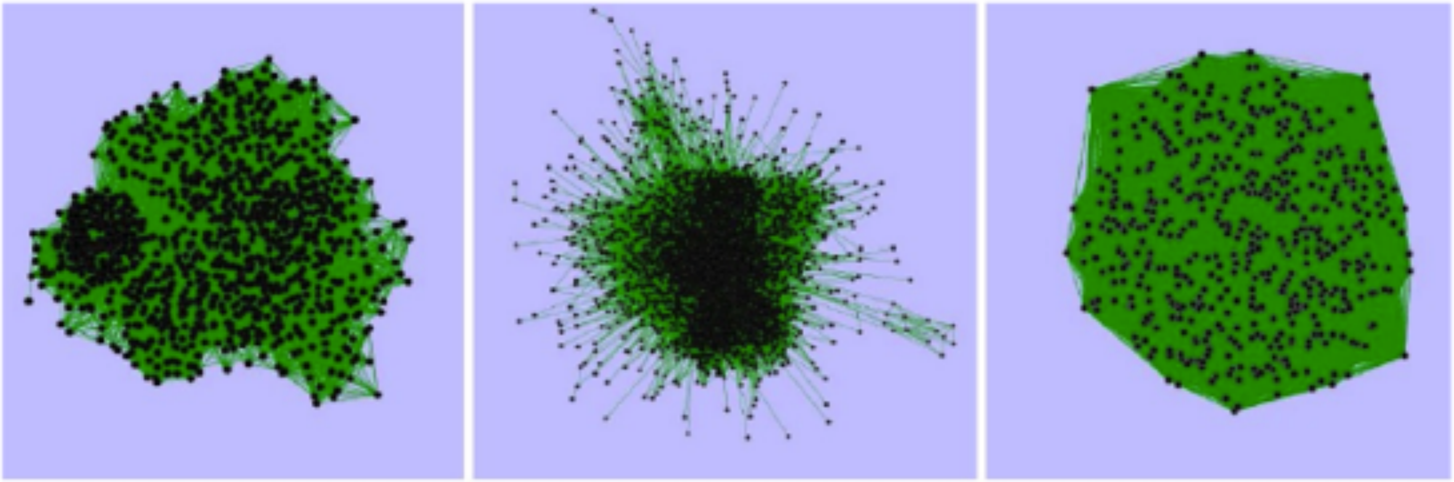
$G_2(V_2, E_2)$ Accumulative graph up to Day 2

Accumulates all event graphs up to time step t.

Graphs extracted from contact traces

Example accumulative graph G_t for $t = 2$ weeks

For $\delta = 1$ day and using force-direct layout algorithm for plotting



(a) Dartmouth (b) USC (c) San Francisco

Seems difficult to extract any knowledge from these social graphs:
$\rightarrow$ gathers all social AND random interaction!

RECAST classifier [1]

- Characterizes the interactions of nodes based on their probability to originate from a random or social behavior
- Identify different kinds of social interactions (friends, acquaintances, bridges or random)
- No geographical dependency, i.e., is of general validity

Together with

Pedro O. Vaz de Melo, Antonio Loureiro – UMFG Brazil

Aline Viana - Inria, Marco Fiore - IIT-CNR Italy

Frédéric Le Mouël – INSA Lyon

[1] RECAST: Telling Apart Social and Random Relationships in Dynamic Networks,
P. Olmo Vaz de Melo, A. Viana, M. Fiore, K. Jaffrès-Runser, F. Le Moüel and A. A. F. Loureiro, 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWim 2013), Barcelona, Spain, 3-8 November 2013.

Social network features: Regularity and Similarity

Social nodes' behavior tend to

- repeat on a regular basis (because of daily activities for instance) $\rightarrow$ Regularity
- build persistent communities and generate common acquaintances $\rightarrow$ Similarity

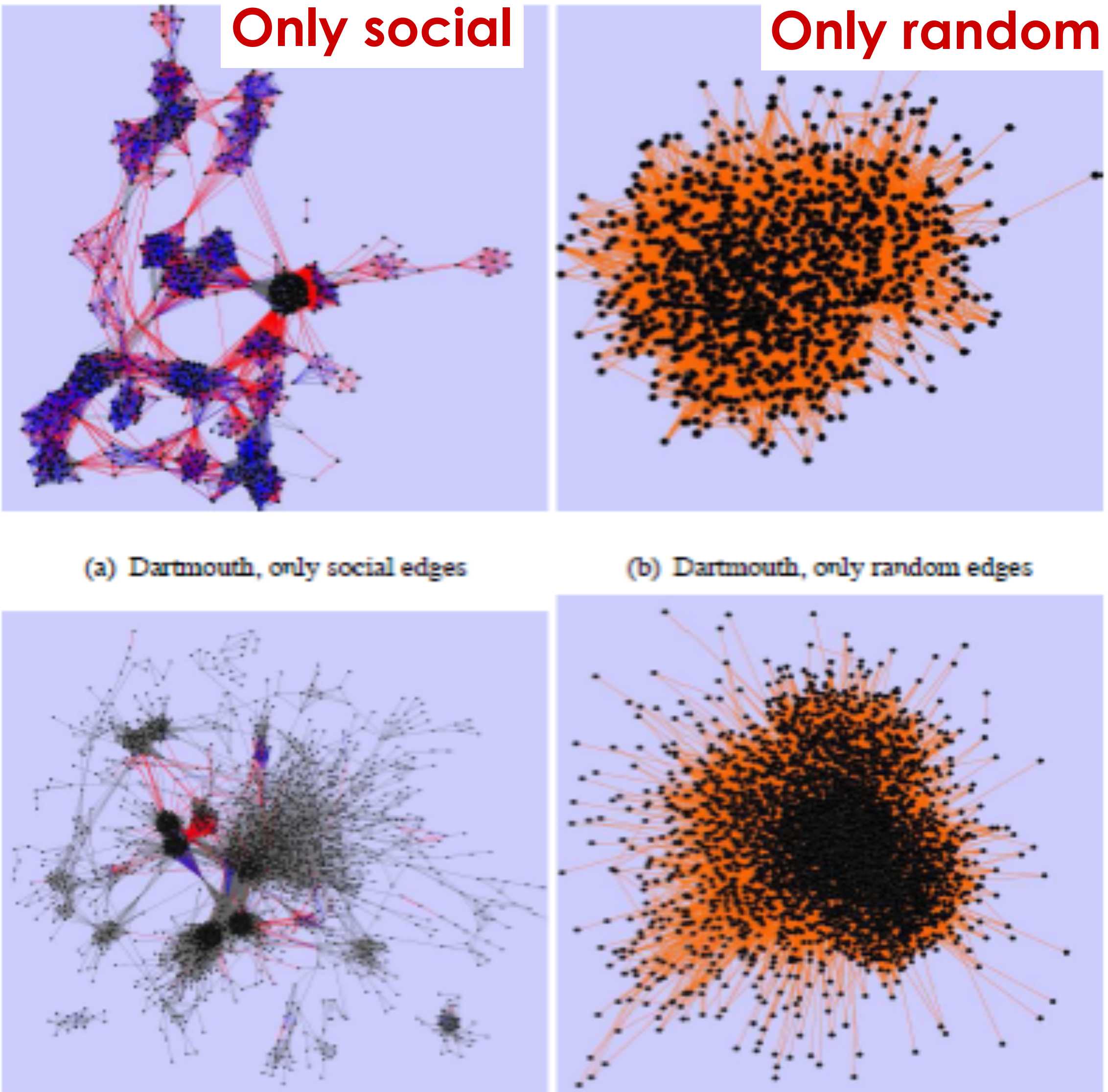
Mathematical metrics

- Edge persistence $per(i,j)$ [7]:
  Percentage of time steps an edge exists over the past discrete time steps in the event graphs $\{\mathcal{G}_i\}_{i \in [1,..,t]}$
- Topological overlap $to(i,j)$ [8]:
  Ratio of neighbors shared by two nodes calculated for the accumulative graph G_t.

[7] N. Eagle et al., "From the Cover: Inferring friendship network structure by using mobile phone data," Proceedings of the National Academy of Sciences, Sept. 2009

[8] J. P. Onnela et al., "Structure and tie strengths in mobile communication networks", Proc. of the National Academy of Sciences, May 2007

Classification after 2 weeks



(a) Dartmouth, only social edges

(b) Dartmouth, only random edges

(c) USC, only social edges

(d) USC, only random edges

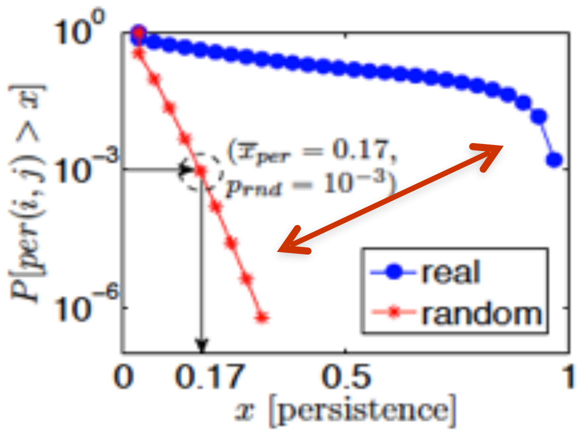
Friends edges are in blue
Bridges edges are in red
Acquaintance edges are in gray
Random edges are in orange

- Social-edges network Complex structure of *Friendship* communities, linked to each other by *Bridges* and *Acquaintanceship*

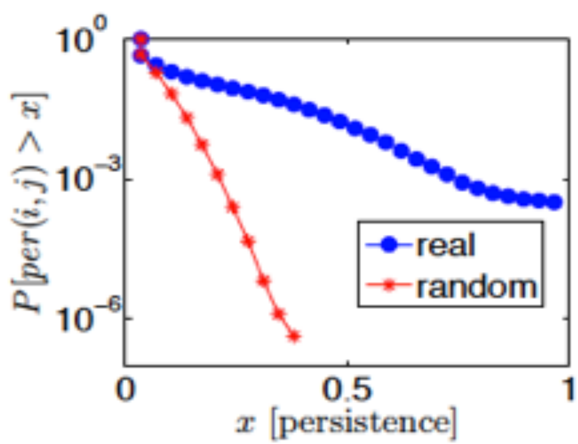
- Random-edges network No structure appears, looking like random graphs

CCDF of edge persistence after 4 weeks

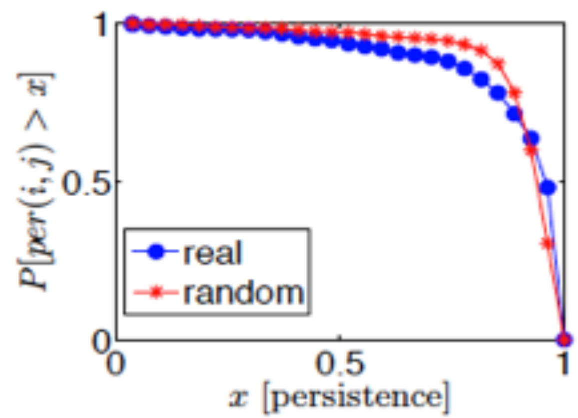
Individuals tend to see each other regularly



(d) Dartmouth



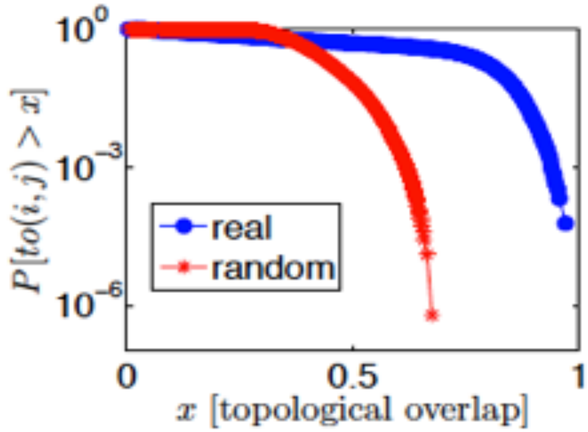
Encounters occur almost in a random fashion



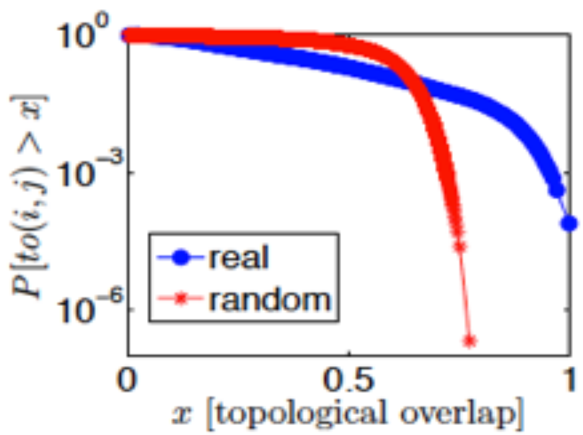
(f) San Francisco

CCFD of topological overlap after 4 weeks

Individuals of G_t have common neighbors

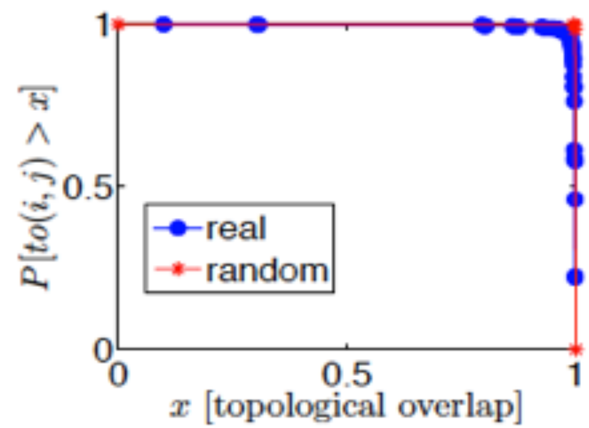


(g) Dartmouth



(h) USC

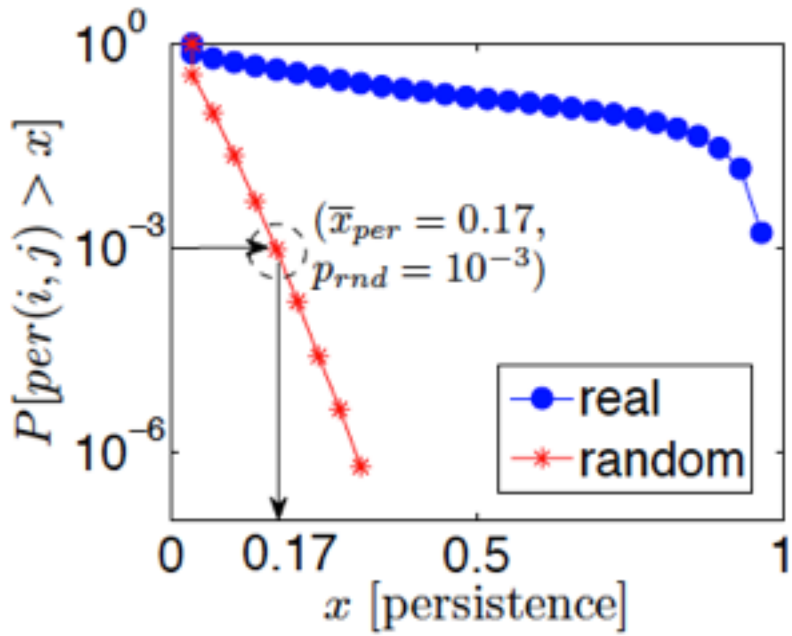
Common neighbors occur in a random fashion



(i) San Francisco

Social vs. Random Edges

In the random network, we only have a probability of 10^{-3} to have edges with a persistence of more than $\bar{x}_{per} = 0.17$.



$\rightarrow$ Thus, in the social graph G_t:

- edges with $per(i,j) > \bar{x}_{per}$ can be classified as *social edges*
- edges with $per(i,j) < \bar{x}_{per}$ can be classified as *random edges*

Note that there is a p_{rnd} chance that a social edge is actually random (mis-classification)

Social graph and its random counterpart

Random graph equivalent of G

Calculate a random graph G^R from a graph $G(V, E)$:

- Keep same number of vertices and edges,
- Randomly assign edges to keep the same node degree distribution using RND algorithm[6]:

An edge is set between nodes of degree d_i and d_j with probability $p_{ij} = (d_i \times d_j) / \sum_{k=1}^{|V|} d_k$

Random accumulative graph G_t^R

Random accumulative graph derived from event graphs $\{\mathcal{G}_i\}_{i \in [1,..,t]}$

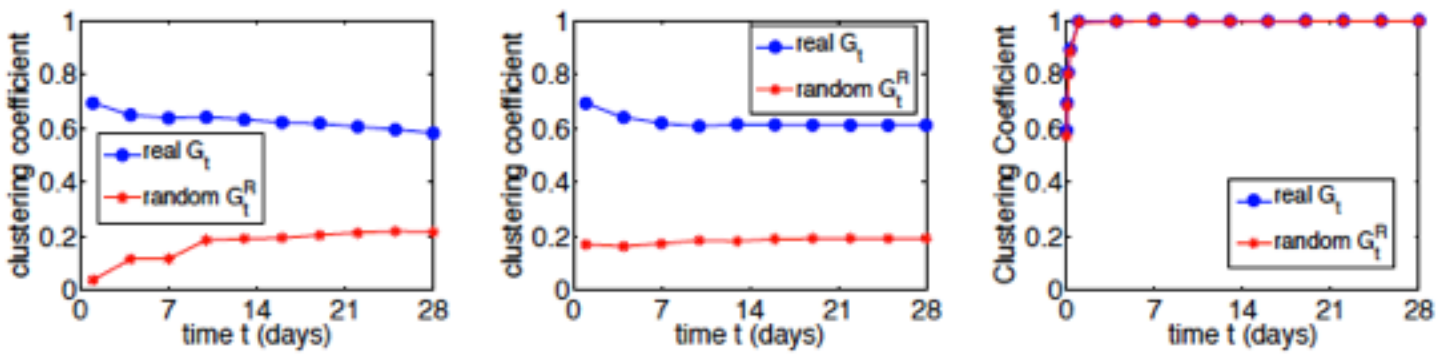
$$G_t^R = \{RND(\mathcal{G}_1) \cup RND(\mathcal{G}_2) \cup \ldots \cup RND(\mathcal{G}_t)\}$$

[6] F. Chung and L. Lu, "Connected Components in Random Graphs with Given Expected Degree Sequences," Annals of Combinatorics. Nov. 2002

Comparison social vs. random graphs

Network clustering coefficient can identify a network with an elevated number of clusters (i.e. communities).

- If $\bar{cc}(G) >> \bar{cc}(G^R)$, parts of the decisions of the nodes of G are NOT random



(a) Dartmouth (b) USC (c) San Francisco

- Dartmouth / USC traces have an order of magnitude higher $\bar{cc}$ than G^R $\rightarrow$ social decisions
- San Francisco: each individual taxi in the trace encounters most of the other taxis $\rightarrow$ closer to a random behavior

RECAST classification algorithm

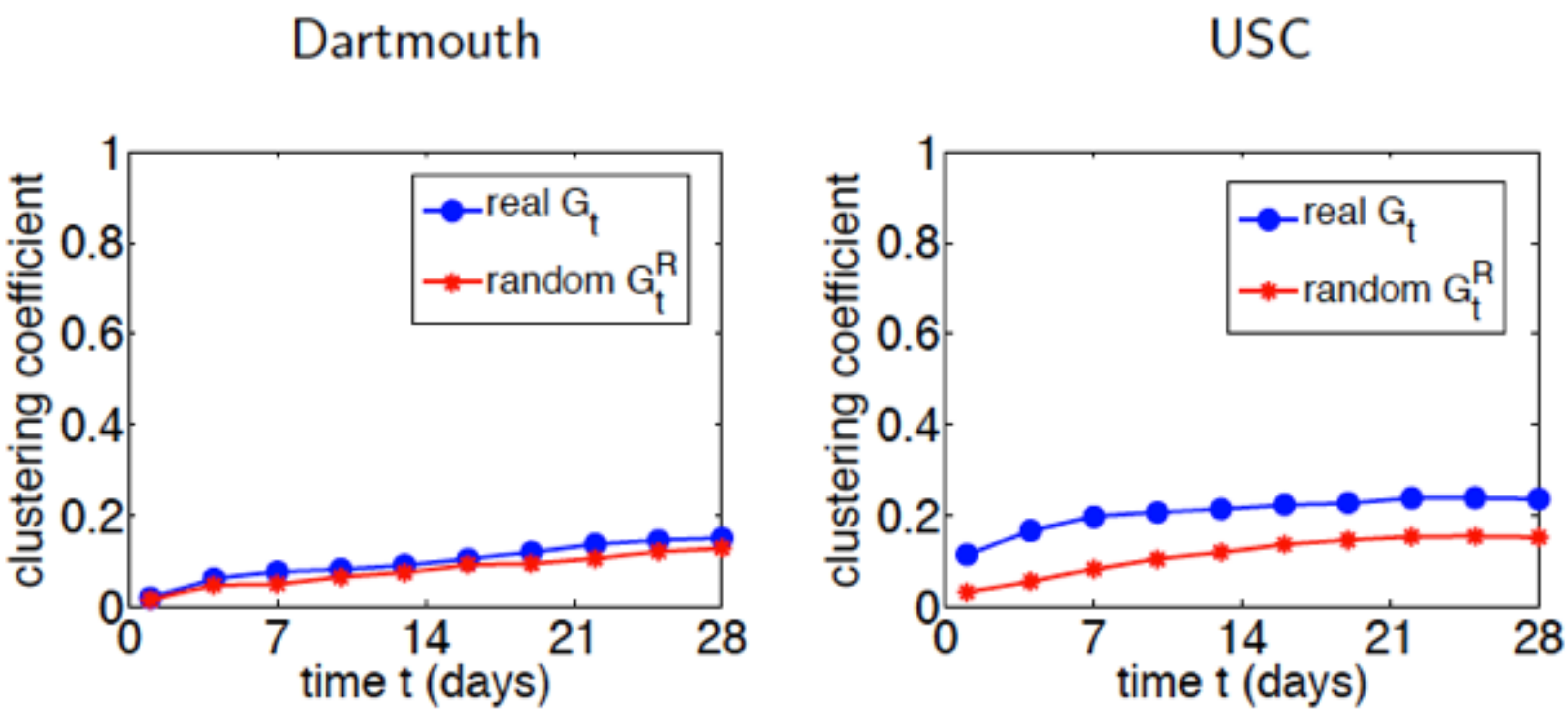
Only parameter of RECAST: p_{rnd}, the mis-classification error bound.

Main steps

- Calculate the $per(i,j)$ and $to(i,j)$ for each edge
- Knowing p_{rnd}, calculate $\bar{x}_{per}$ and $\bar{x}_{to}$ from CCDF's
- For each edge,
  - if $per(i,j) > \bar{x}_{per} \rightarrow (i,j)$ is social for edge persistence
    else (i,j) is random for edge persistence
  - if $to(i,j) > \bar{x}_{to} \rightarrow (i,j)$ is social for topological overlap
    else (i,j) is random for topological overlap
- Classify edges into classes of relationships according to:

Class	Edge persistence	Topological overlap
Friends	social	social
Acquaintances	random	social
Bridges	social	random
Random	random	random

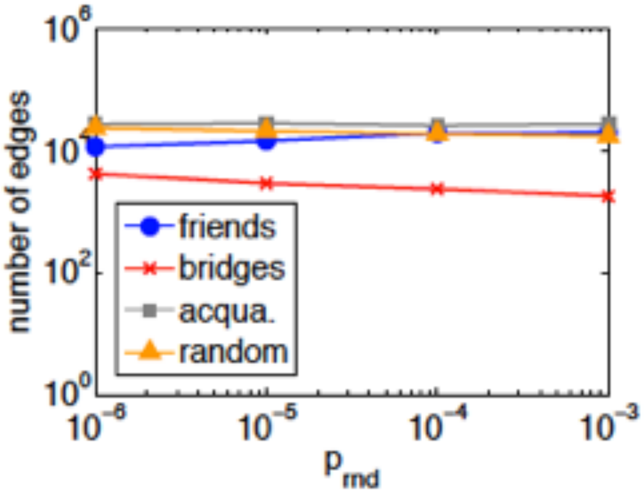
Cluster coefficient analysis for **random edges only**



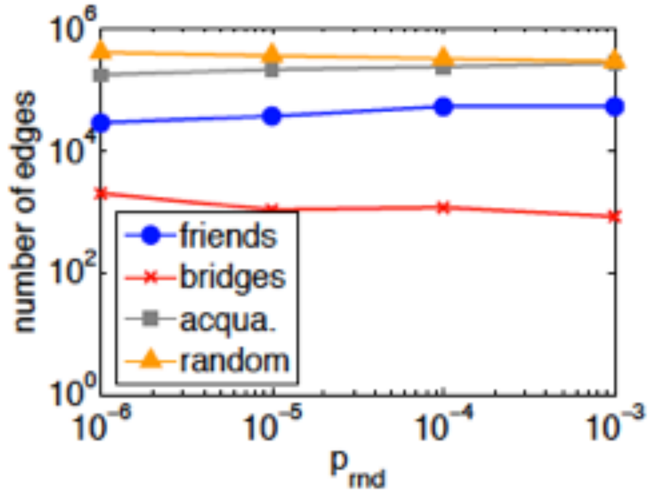
Validates the efficiency of RECAST to identify random edges for Dartmouth and USC

Impact of p_{rnd}

Number of edges of a each class that appear in the first 4 weeks vs. p_{rnd}



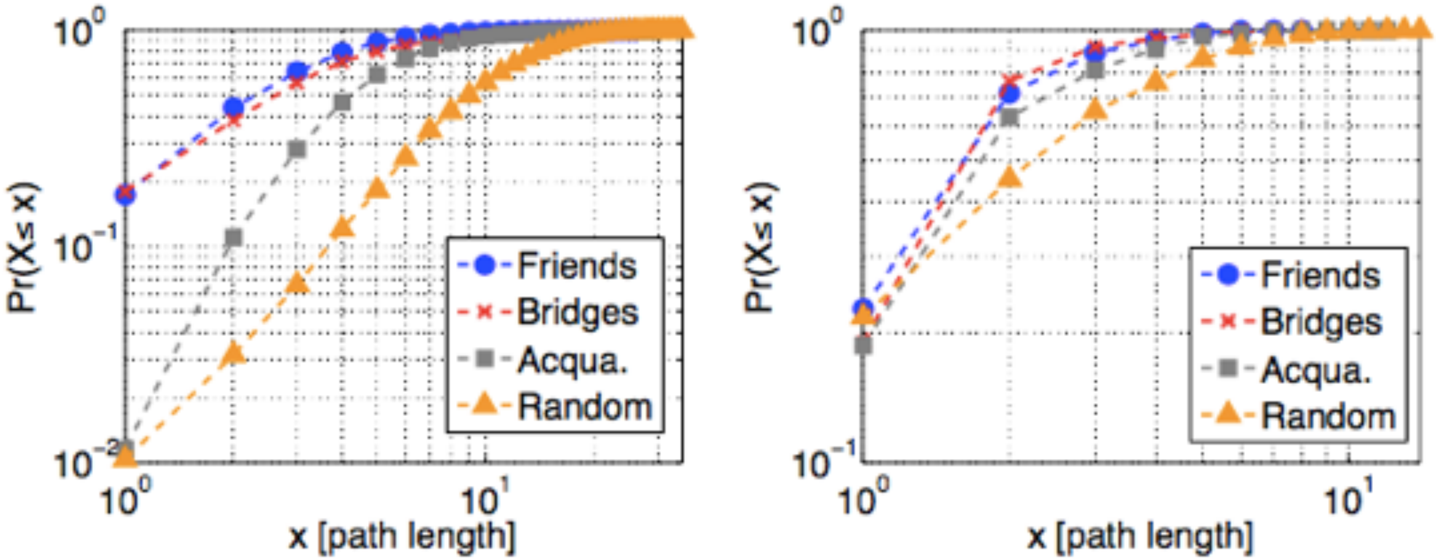
Dartmouth



USC

RECAST is not sensitive to p_{rnd} !

Forwarding using relationship information



(a) Dartmouth (b) USC

Figure 11: The histogram of the path lengths of messages between users i and j who share a determined class of relationship.

Forwarding with recast or FB data

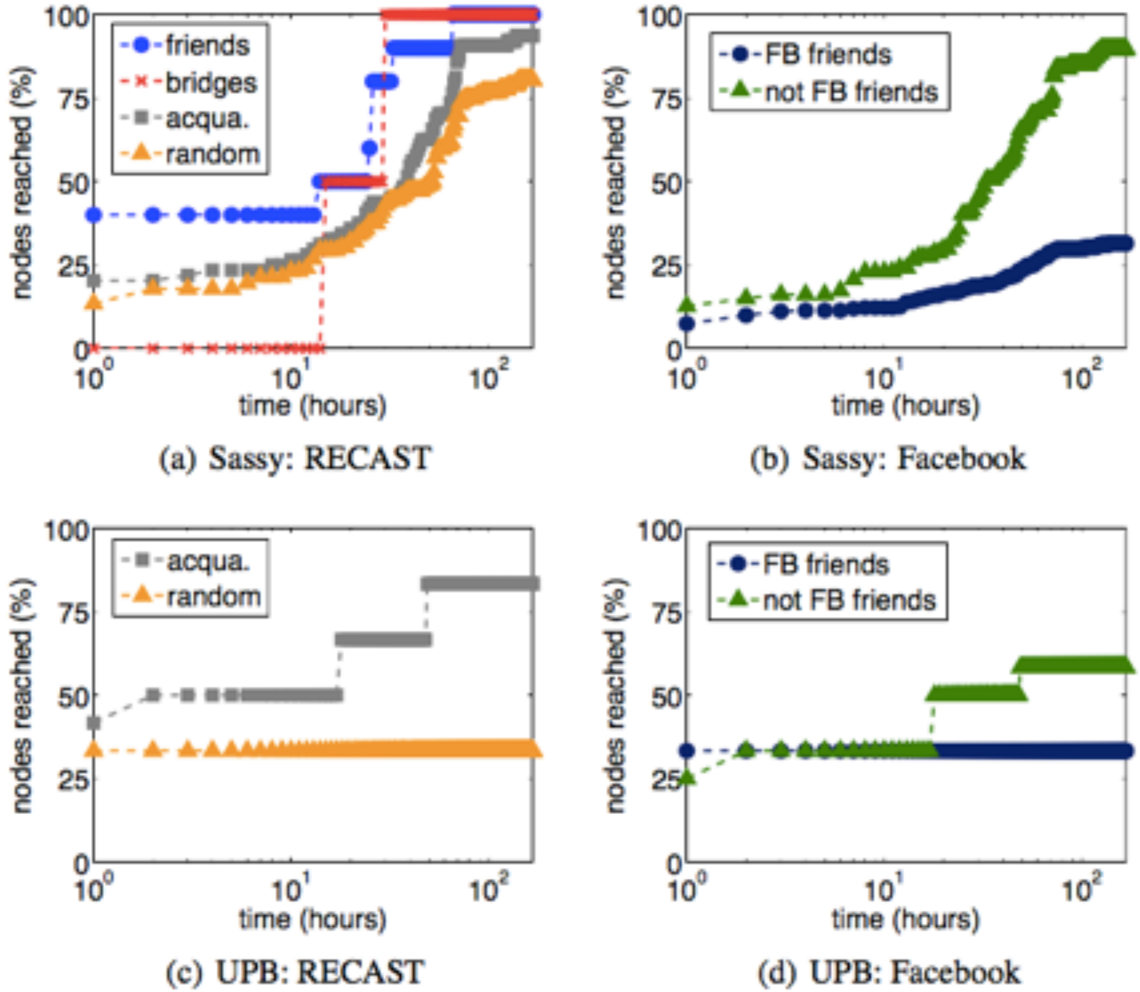


Figure 12: The % of users who were reached over time grouped by RECAST classes and Facebook (FB) friendship.

Next…

- Having this data, exhibit the correlations between content and context
  - Do users have regular habits in data usage?
  - If yes, is it possible to model these networks with the content plane in mind?
- Using network models, deriving data pre-fetching strategies to adjust the load off available networks

….