Decentralized Clustering for Botnet Detection

Alessio Guerrieri

18 Novembre 2010

Outline

1. Introduction
   - Botnets
   - Clustering
2. BotMiner: centralized botnet detection framework
   - Overview
   - Framework
   - Results
3. Decentralized botnet detection framework
   - Intro
   - Framework
   - Results
   - Future work

Botnets

Definition

A group of nodes forms a botnet if they

- communicate using a similar pattern
- commit similar malicious activities

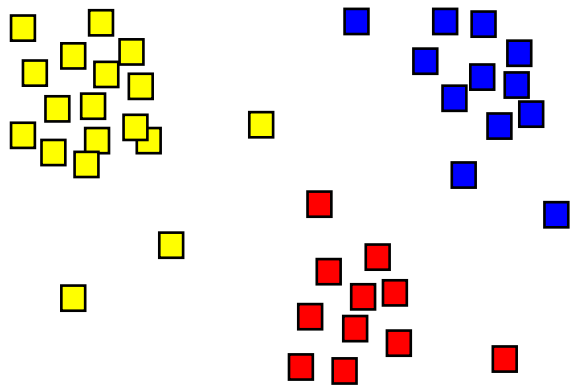
- No specialization
- We can try to clusters nodes with similar behaviour

CLUSTERING ALGORITHMS

CLUSTERING PROBLEM

Given a set of items, divide them in subsets such that items in the same subsets are “similar”

- Items are usually represented as points in R^n
- Similarity measured with distance function



K-Means clustering

- Needs K (number of clusters) as input
- Partitions the points to minimize the within-cluster sum of squares
- Problem is NP-HARD

Standard algorithm (Lloyd 1982)

- Choose K centroids at random
- While changes:
  - Assign each point to closest centroids
  - Move each centroid to the average of the points assigned to it

Notes on K-Means clustering

- Results depends on starting centroids
- Can terminate at local optima
- Time complexity is $2^{\Omega(n)}$
- Smoothed analysis shows polynomial time

XMeans (Pelleg 2000)

- Does not need the exact number of clusters
- Uses Bayesian Information Criterion to compare models
- Receives MAX= maximum number of clusters

Algorithm

- Run 2-means on the data to get 2 clusters
- While number of cluster less than MAX:
  - For each cluster:
    - Run 2-means locally
    - If improvement, replace cluster with result of 2-means
  - If no replacement, replace worst clusters
- Return best clustering found at any iteration

- Can be speeded up with kd-trees and caching

BotMiner

- From paper "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection" (Gu 2008)

Framework overview

- Each node is mapped to two points in R^n, one created from communication pattern (C-Plane), one from malicious activites (A-Plane)
- The C-Plane and A-Plane are clustered independently
- We look for subset of nodes that are in the same cluster in both planes.

Mapping

C-Plane

- number of flows per hour
- the number of packets per flow
- the average size of the packets
- the number of bytes sent per second.

A-Plane

- Scan activity
- Spam activity
- Binary downloading
- Exploit activity

For each activity, keep additional features (ex: ports)

Clustering

C-Plane

Two-step clustering using X-Means:

- Global clustering on 8 features
- Local clustering using all features.

A-Plane

First cluster according to activities, then using the features.

Cross-Clustering Correlation

"BotNet Score" is computed using two components:

- amount of malicious activities of the node
- how many nodes share the same characteristics (Cross-clustering correlation)

Cross-clustering correlation

If node n is in clusters A_i and C_j:

$$ccc(n) = \frac{A_i \bigcap C_j}{A_i \bigcup C_j}$$

- Nodes with score bigger than threshold are reported

RESULTS

- Used ten botnet traces embedded in anormal traffic trace

Botnet	Bots	Detected?	Clustered	Det. Rate	False Pos.	FP Rate
IRC-rbot	4	YES	4	100%	1/2	0.003
IRC-sdbot	4	YES	4	100%	1/2	0.003
IRC-spybot	4	YES	3	75%	1/2	0.003
IRC-N	259	YES	258	99.6%	0	0
HTTP-1	4	YES	4	100%	1/2	0.003
HTTP-2	4	YES	4	100%	1/2	0.003
P2P-Storm	13	YES	13	100%	0	0
P2P-Nugache	82	YES	82	100%	0	0

Model

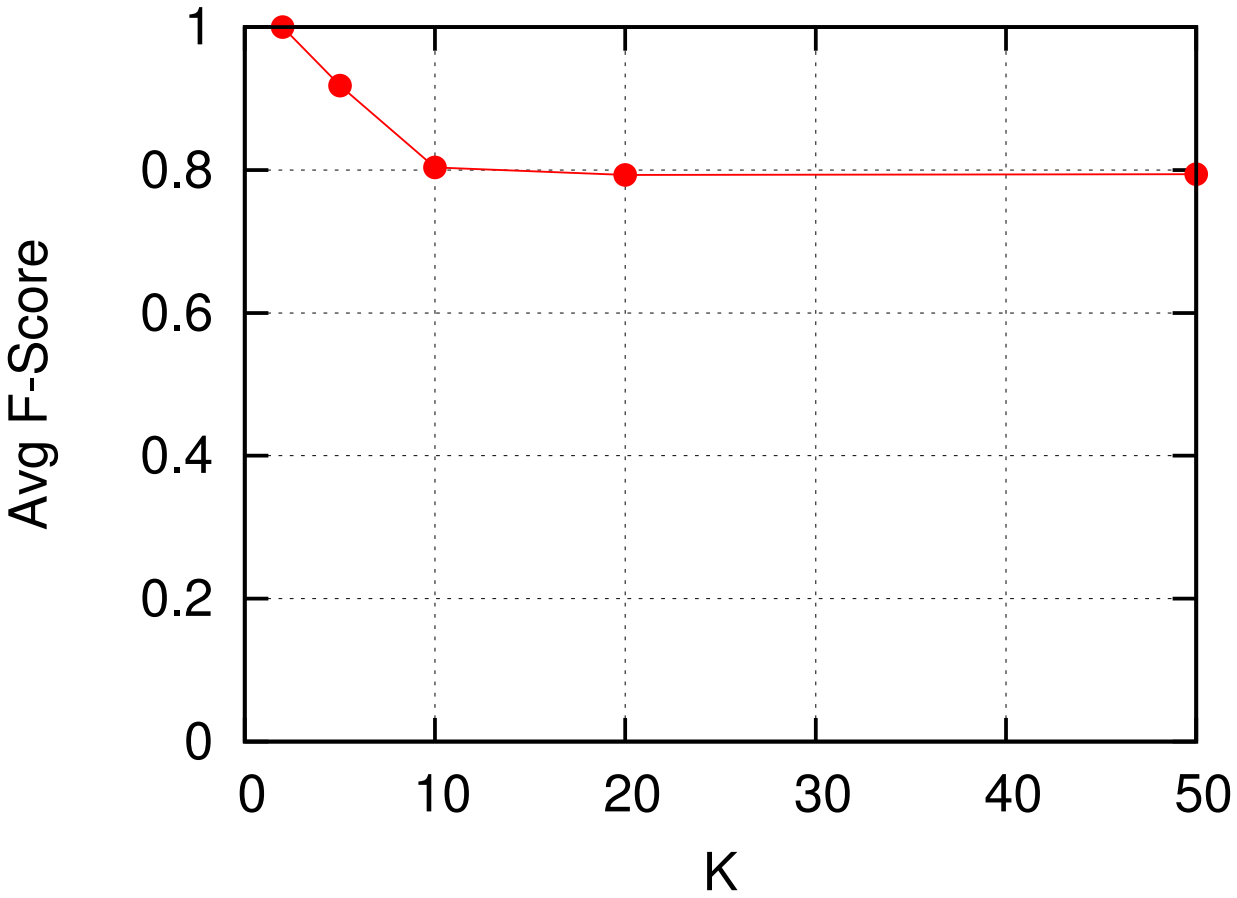
- Network divided in many subnetworks
- Each subnetwork contains one computing node and many monitor nodes
- Monitor nodes observe the subnetwork and send data to computing node.
- Subnetworks do not want to share sensitive data

DISTRIBUTED K-MEANS

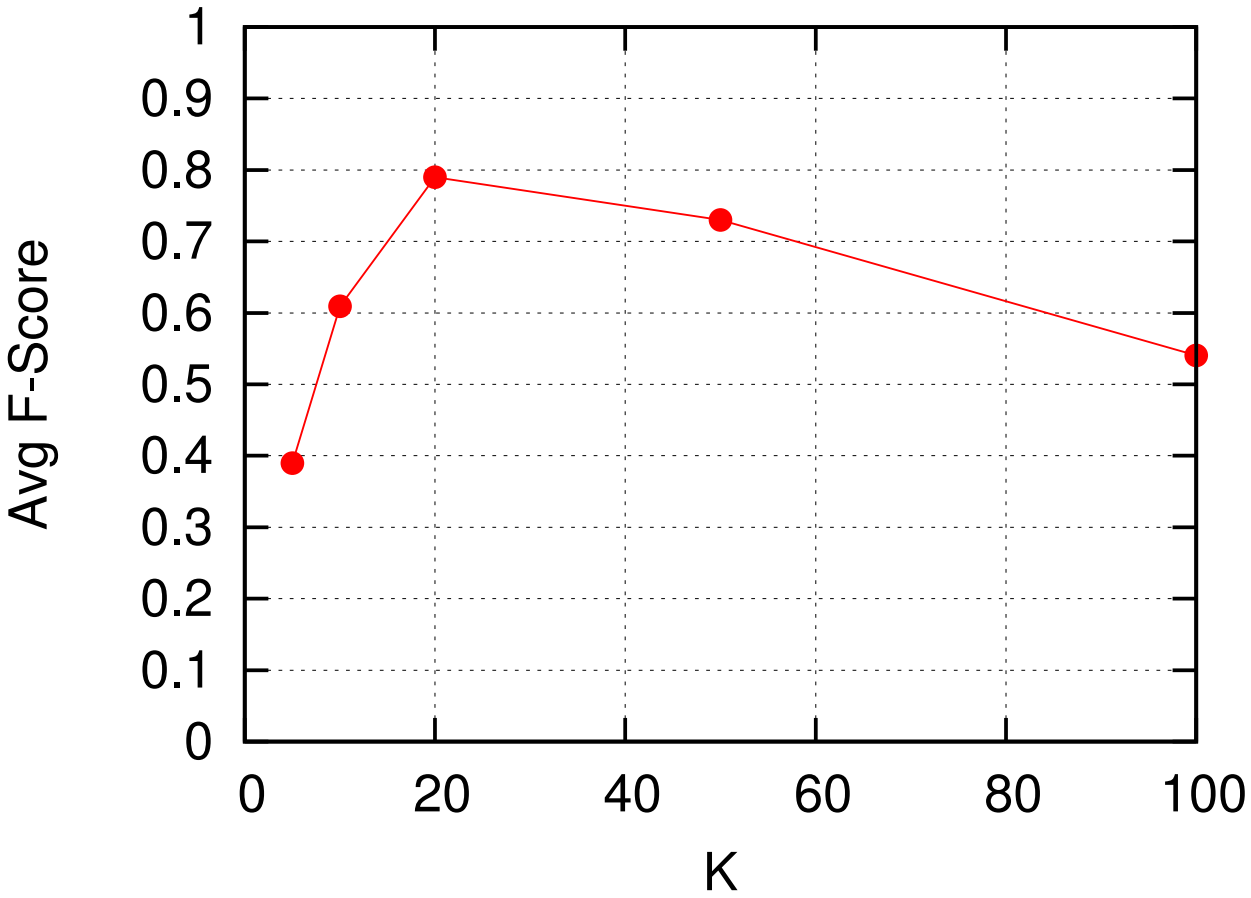
From “Clustering distributed data streams in peer-to-peer environments” (Bandyopadhyay 2006)

- Each node use same starting centroid
- Until nodes or neighbours do not change
  - Map local points to centroid
  - Compute new position of centroids
  - Ask neighbours for their centroids
  - Answer requests if any
  - Replace local centroids with average of centroids received

DISTRIBUTED K-MEANS WITH CORRECT K

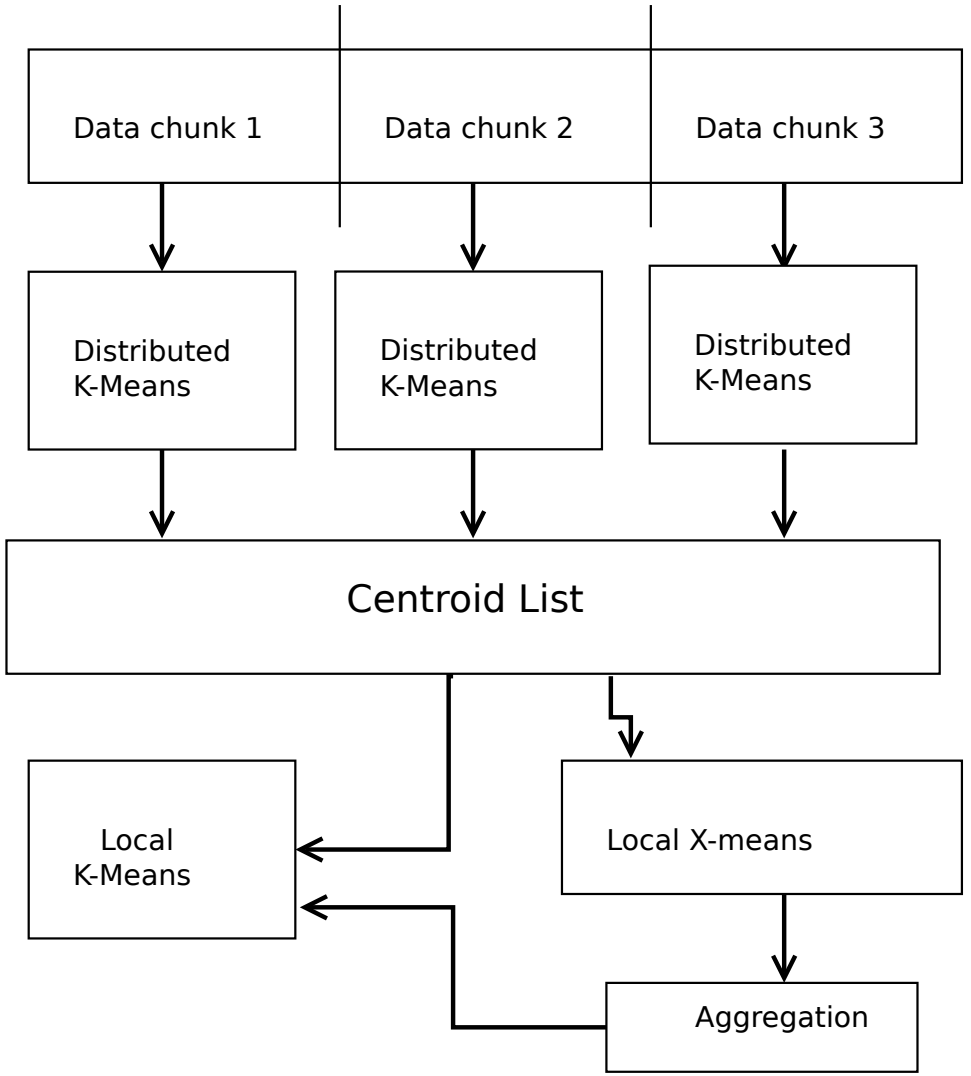


DISTRIBUTED K-MEANS WITH REAL K=20

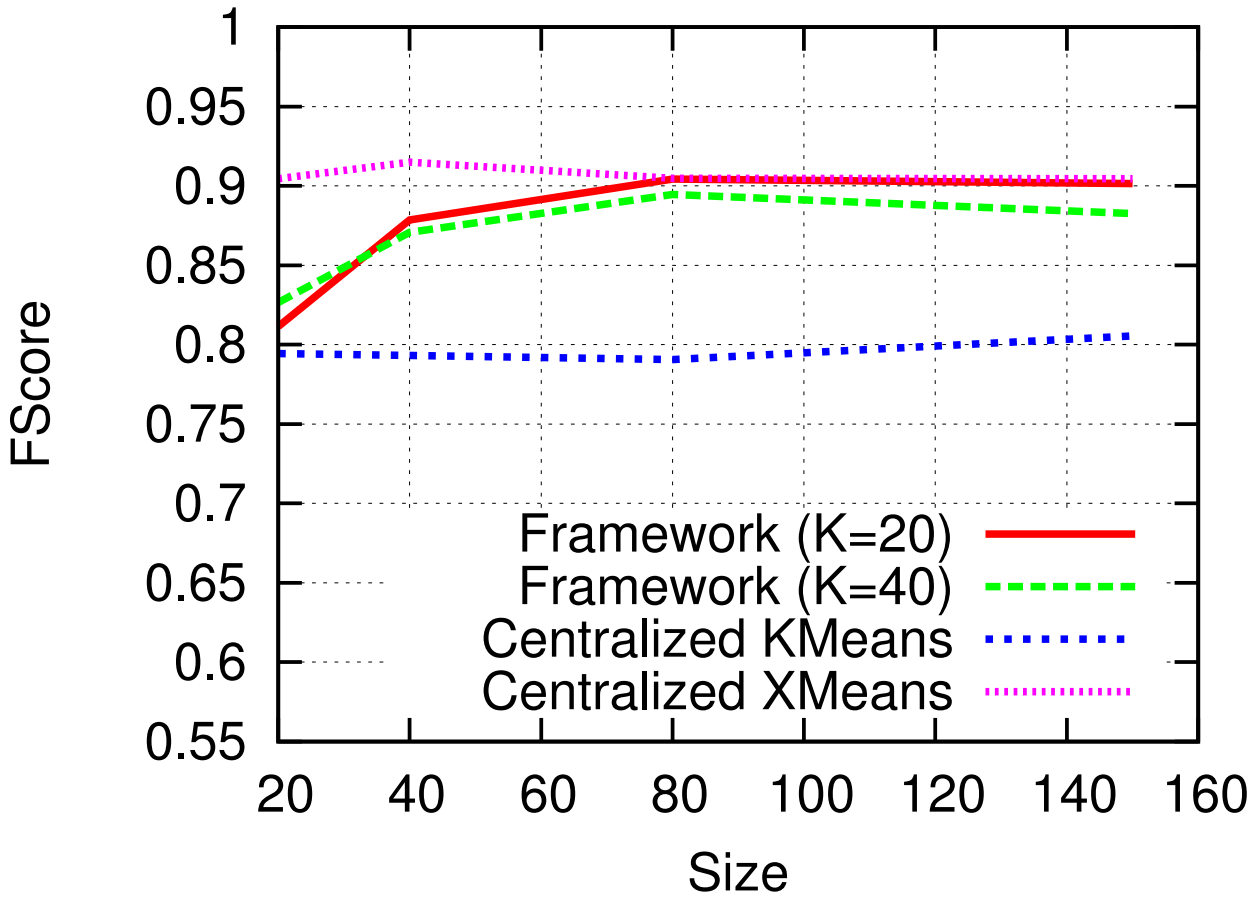


FRAMEWORK IN BRIEF

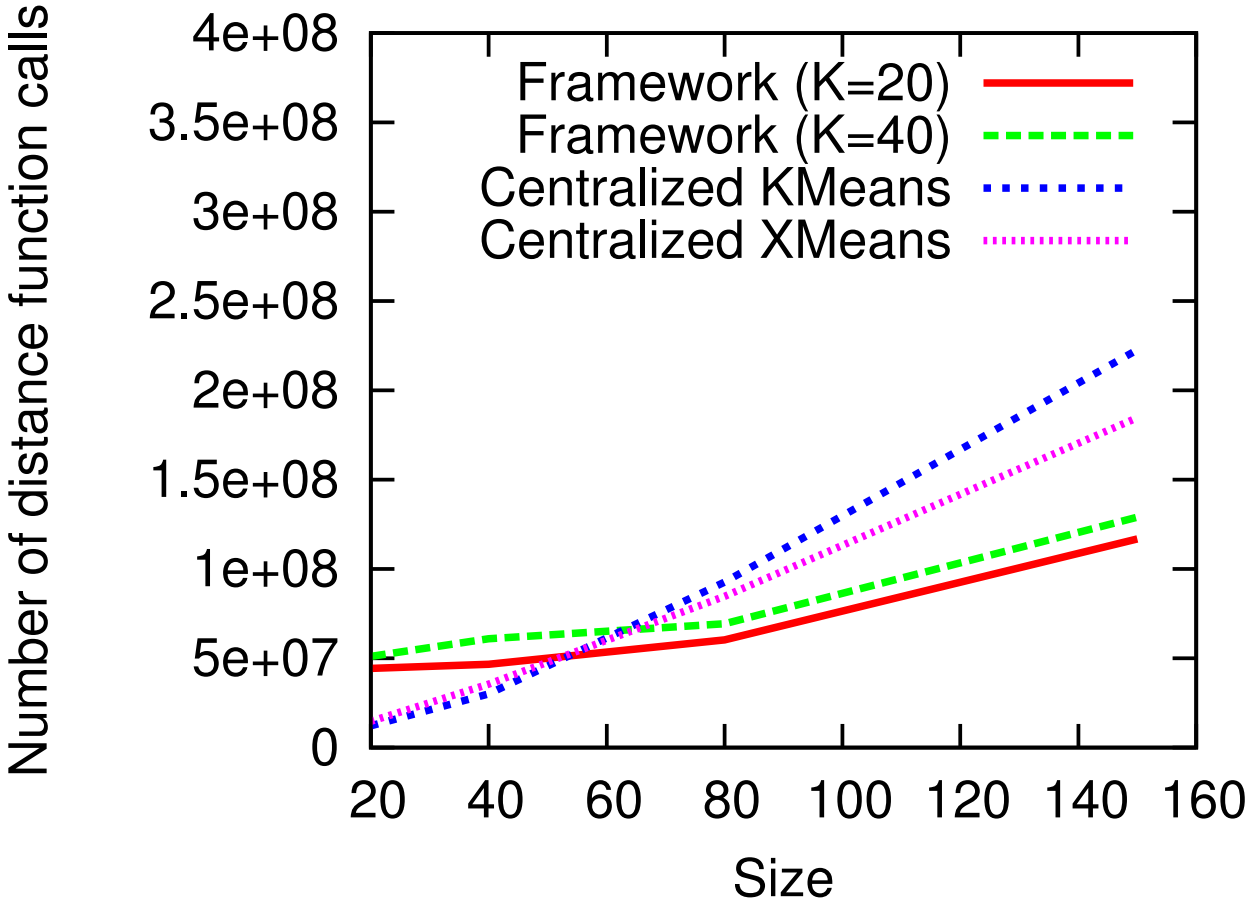
- Each node computes points in A-Plane and C-Plane
- Distributed clustering according to scheme
- Cross clustering Correlation TBD



PRECISION OF FRAMEWORK



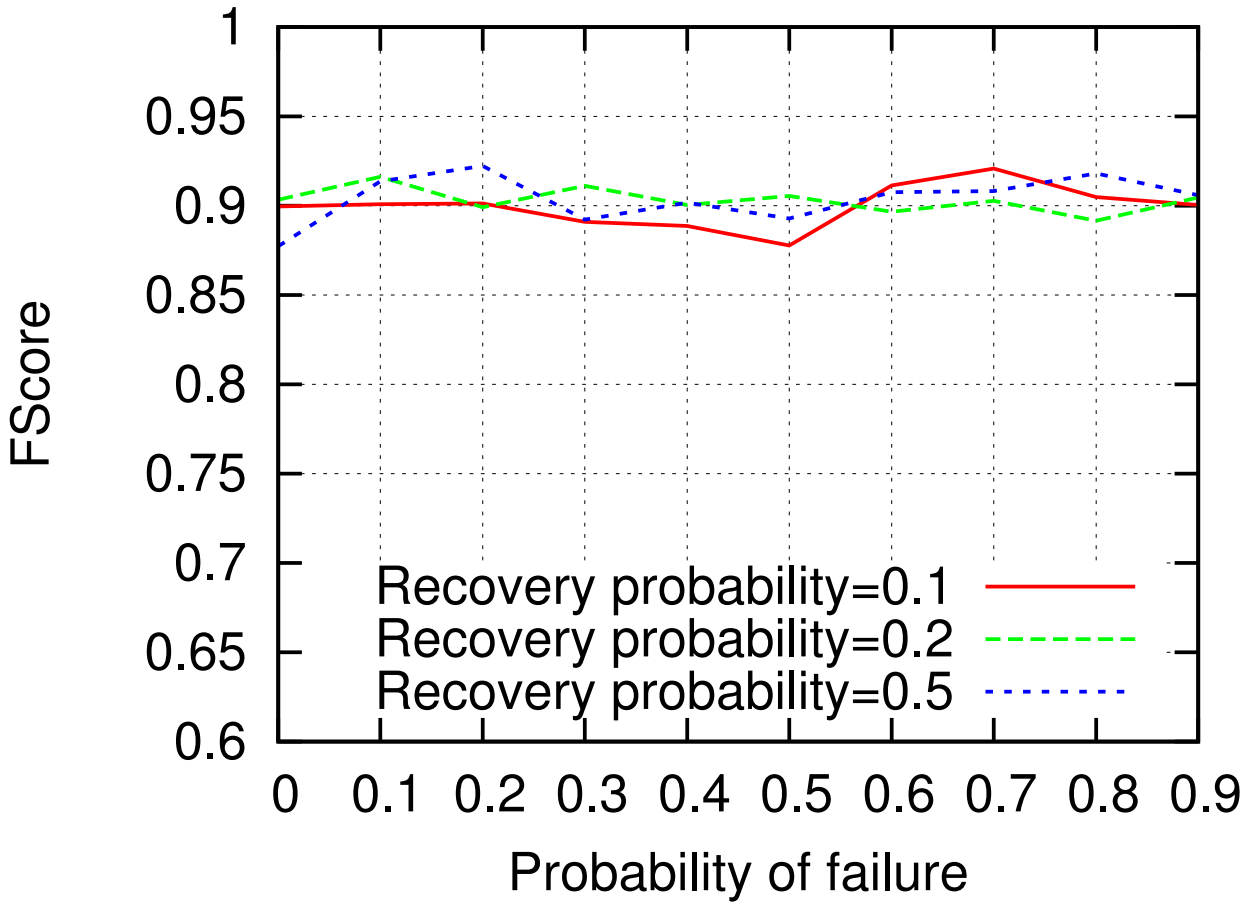
Number of Distance function calls



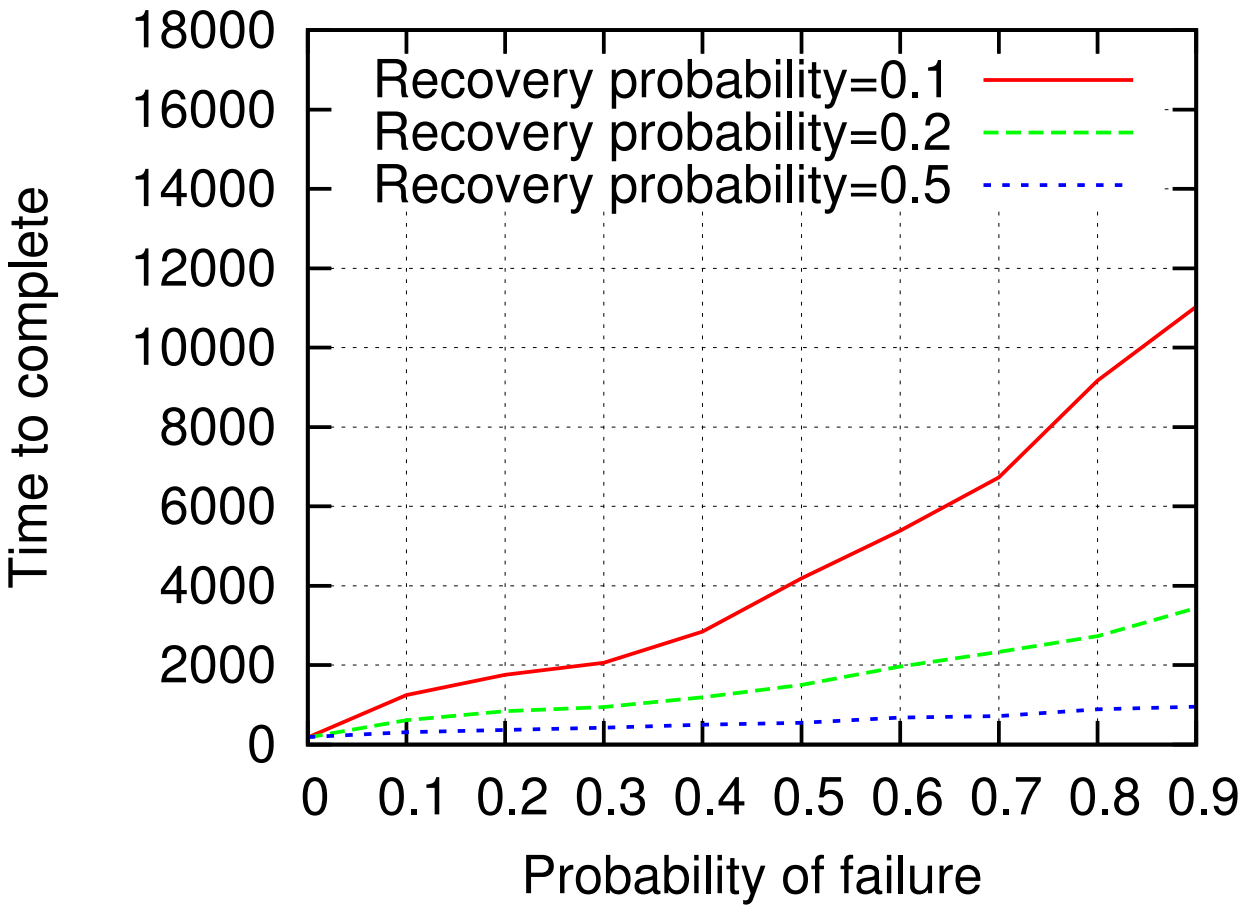
Robustness setting

- We use two parameters, P_u, P_d
- Every 10 units of time
  - Each up node goes down with prob. P_d
  - Each down node goes up with prob. P_u
- Down nodes do not analize data or answer messages

ROBUSTNESS


FScore
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Probability of failure
Recovery probability=0.1
Recovery probability=0.2
Recovery probability=0.5

ROBUSTNESS


Recovery probability=0.1
Recovery probability=0.2
Recovery probability=0.5
Time to complete
Probability of failure
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

Next steps

- Development of Distributed Cross Clustering Correlation
- Testing of framework with real data
- Work on online features