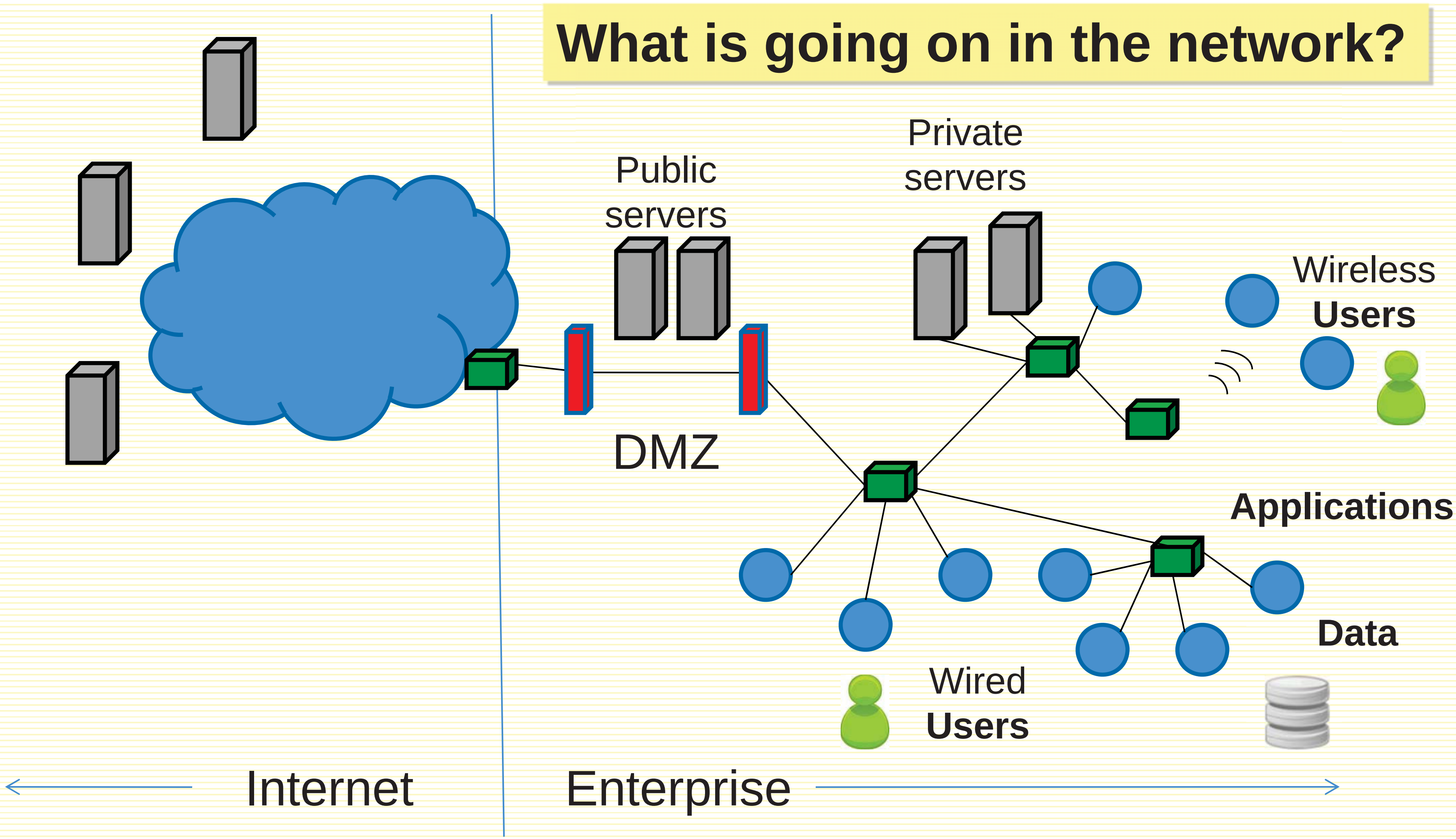
Intelligent Network Management Using Graph Differential Anomaly Visualization

Qi Liao

Department of Computer Science
Central Michigan University

Network Management

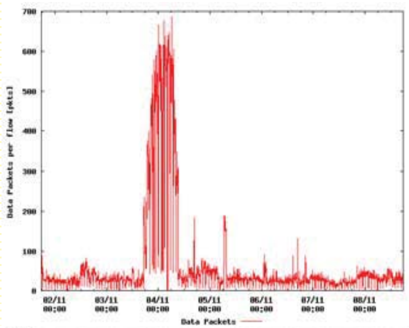


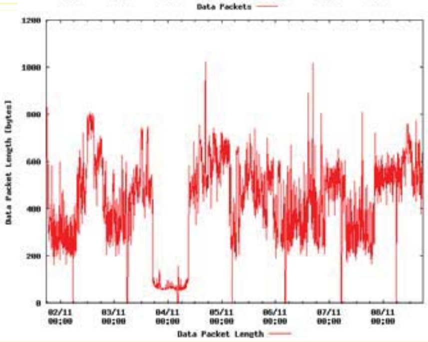
Security Management

- Needs of Network Manager
  - Health check
  - Situation awareness
  - Accountability / Forensics
  - Troubleshoot
- Challenges
  - Huge amount of data
  - Complexity
  - Dynamics
  - Gap: daily monitoring ←→ operational interpretation

Network Anomaly

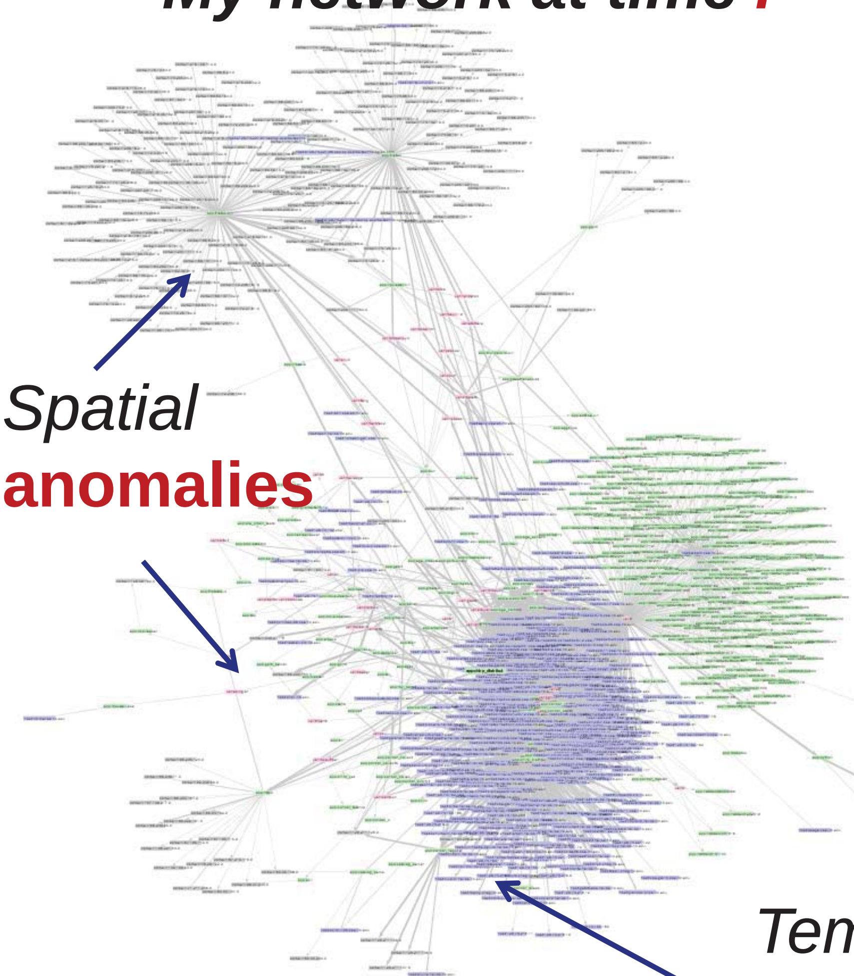
- Network anomaly is useful in many areas of network management.
- Some examples of "easy" anomalies
  - Readings from sensor network
  - DoS attack
  - Port scanning
  - Packet headers match a pattern
- More *general* (harder) anomalies
  - Stealthy
  - Less traffic
  - Given only a time-series of network graphs, can we detect abnormal changes and find the underlying causes?



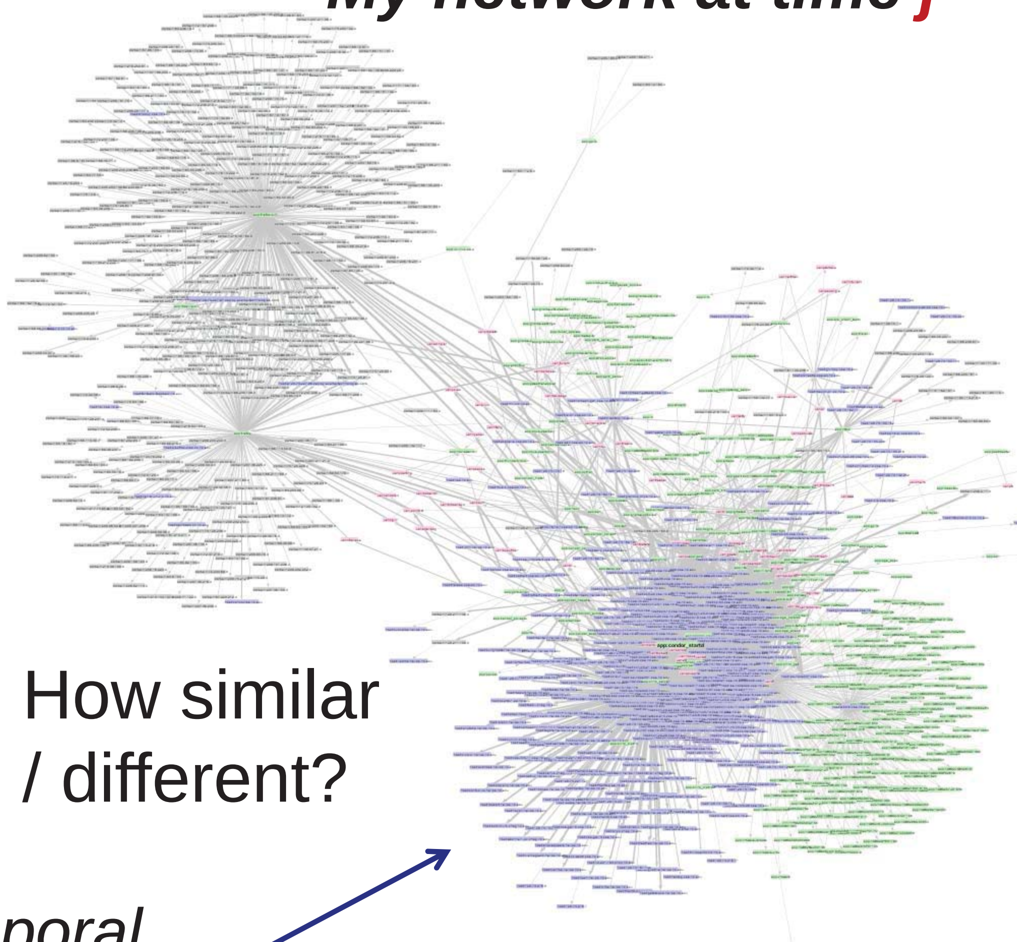


Graph Diff. Anomaly Visualization

My network at time i



My network at time j



Spatial **anomalies**

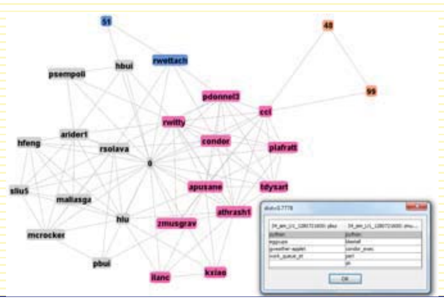
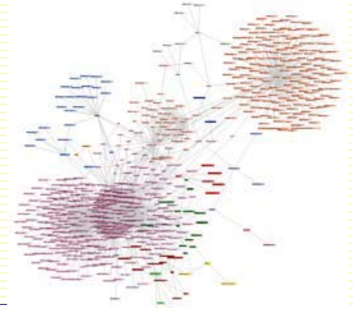
How similar / different?

Temporal **anomalies**

Differential Anomaly Visualization

- Graph *differential anomaly visualization* (*DAV*) framework
  - Whole graphs
  - Nodes and edges
  - Communities (subgraphs)
    - More tolerant to the *dynamics* of network.
- Effectively visualizes the *dynamics* and *abnormal changes* among the heterogeneous, time-series network graphs.

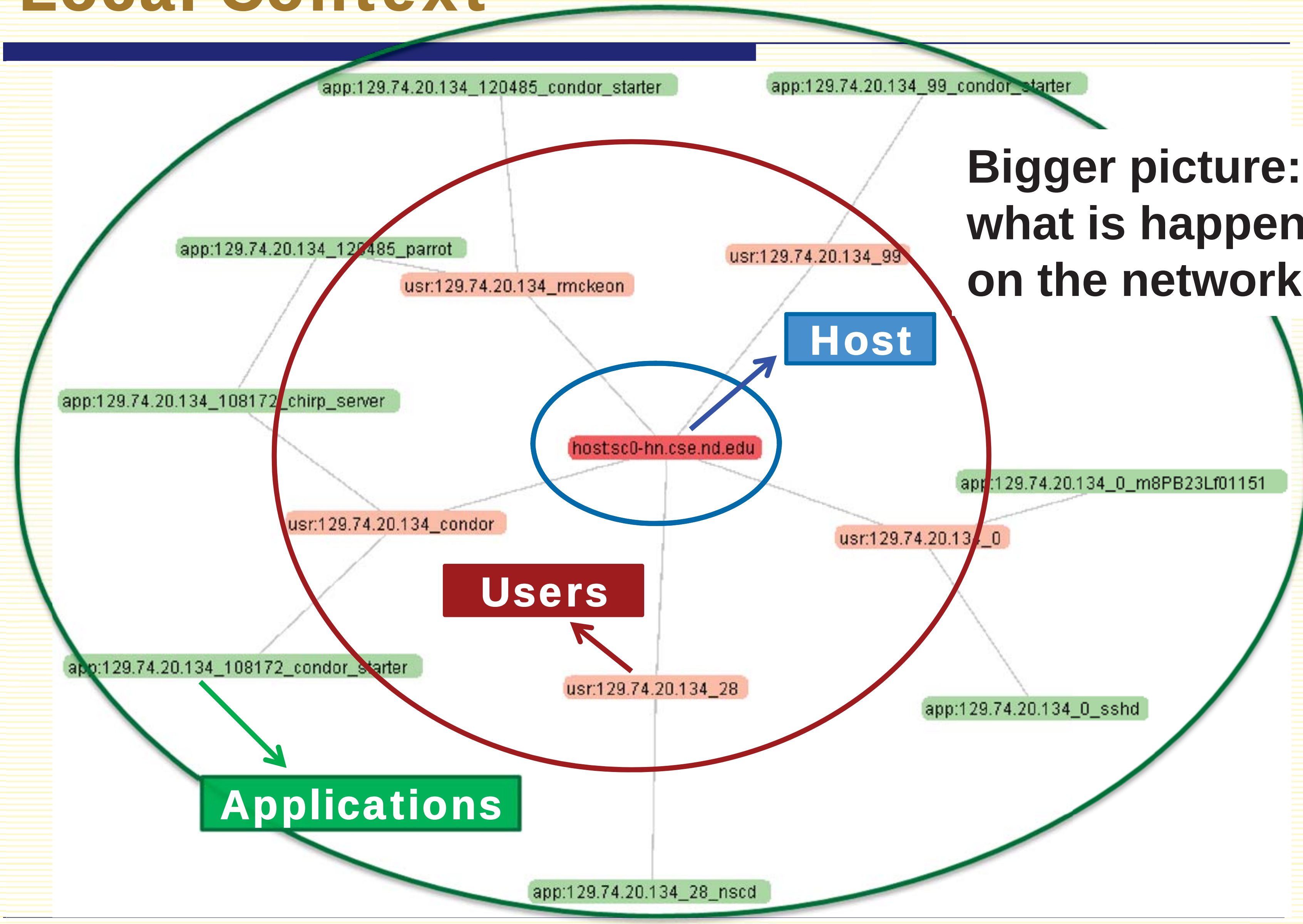


Monitoring Where, Who, and What

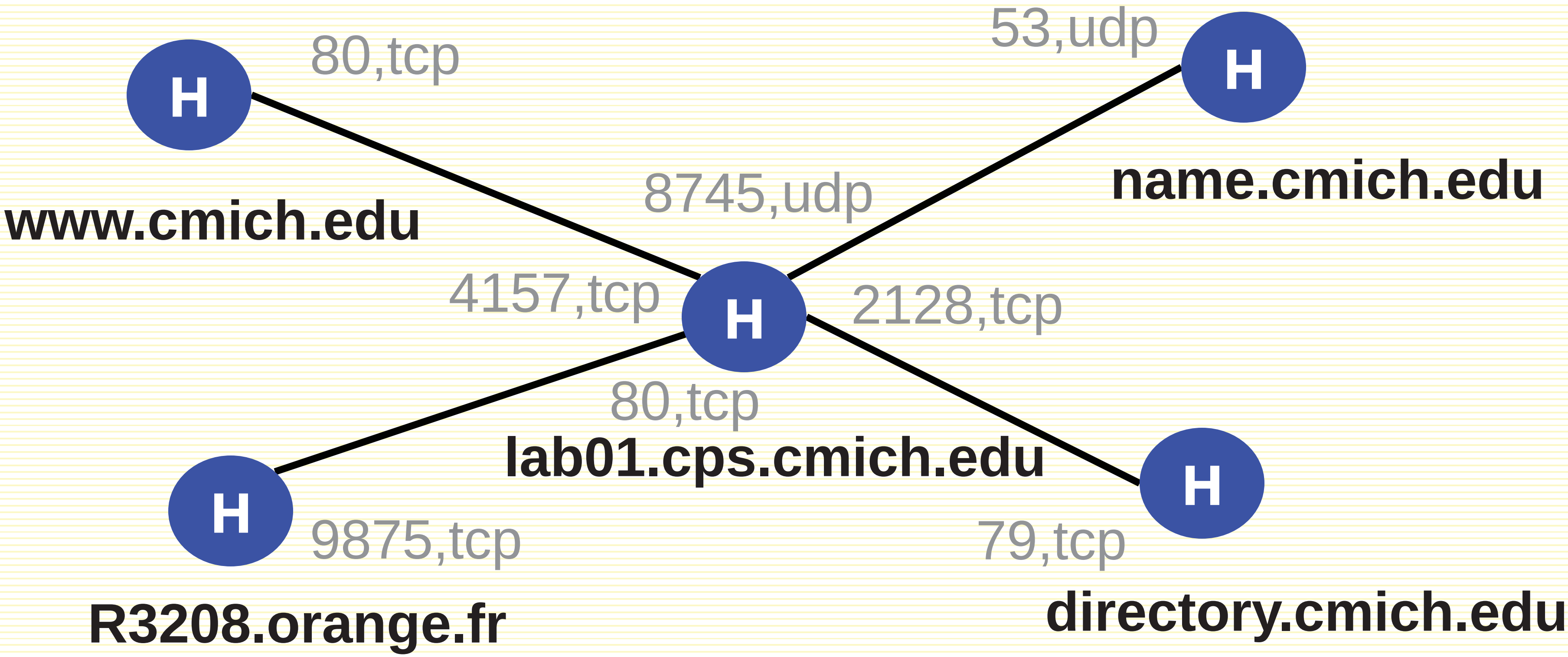
- Need finer granularity than **raw network connectivity**
- Two important enterprise network components
  - ***Who*** (users) are responsible
  - ***What*** (applications) are running on the network.
- ***CONTENT*** vs. ***CONTEXT***
  - Associated with each network connection
  - *Users*, *applications*, *parameters*, *file accesses, etc.*

Local Context



Bigger picture: what is happening on the network

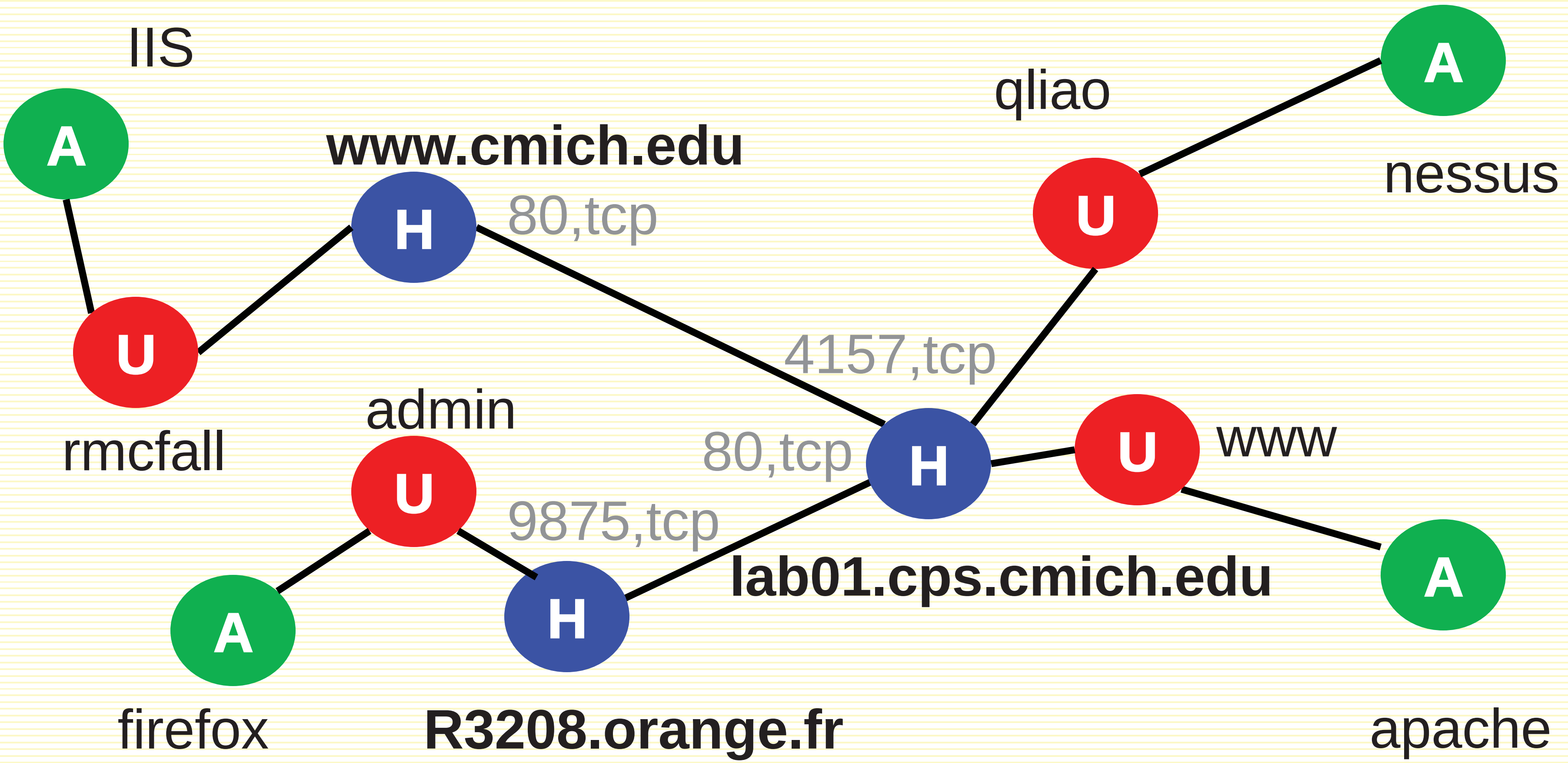
Traditional view



Most existing tools show this view

Web traffic in, web traffic out, DNS, Active Directory

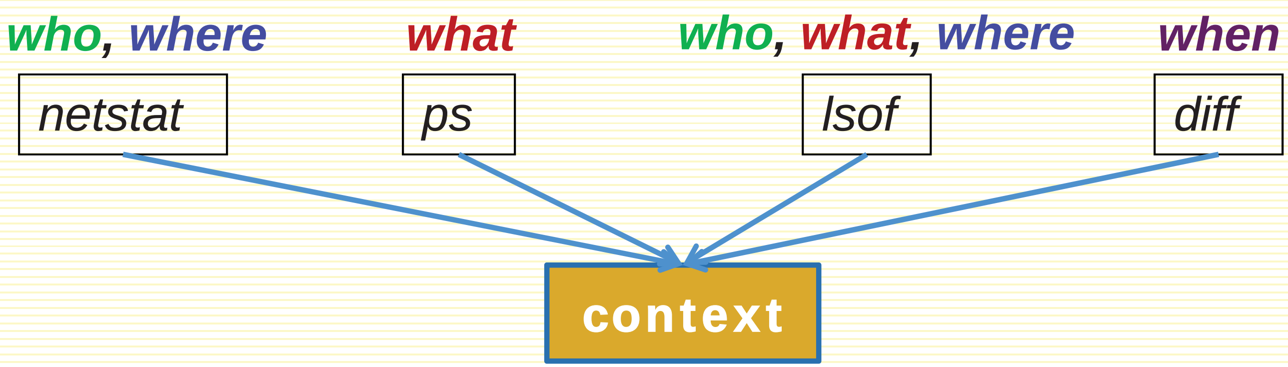
Network flows – Who and what?



Network Context Graphs

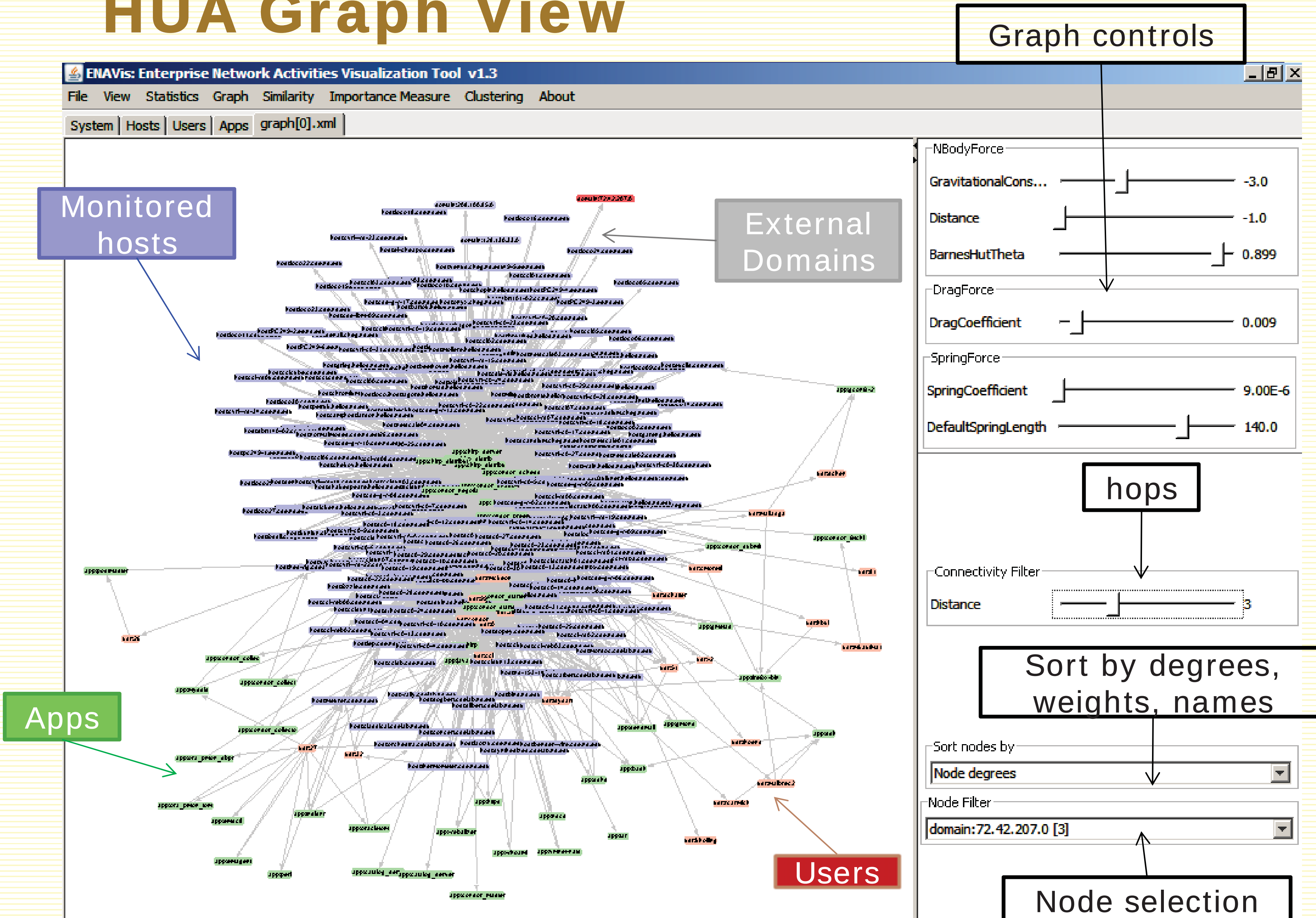
Data Collection Agent

- Gathers context from local hosts
  - ***who*** (***users***), ***what*** (***applications***), ***when*** (***time***), ***where*** (***hosts***)
- Built-in system tools (free and robust)



- Easy to deploy (no change to existing systems)
- Lightweight
  - CPU< 2%
  - Bandwidth (1000 hosts: 240 Kbps = 0.2% of 100Mbps)
  - Disk (1GB /host/year)

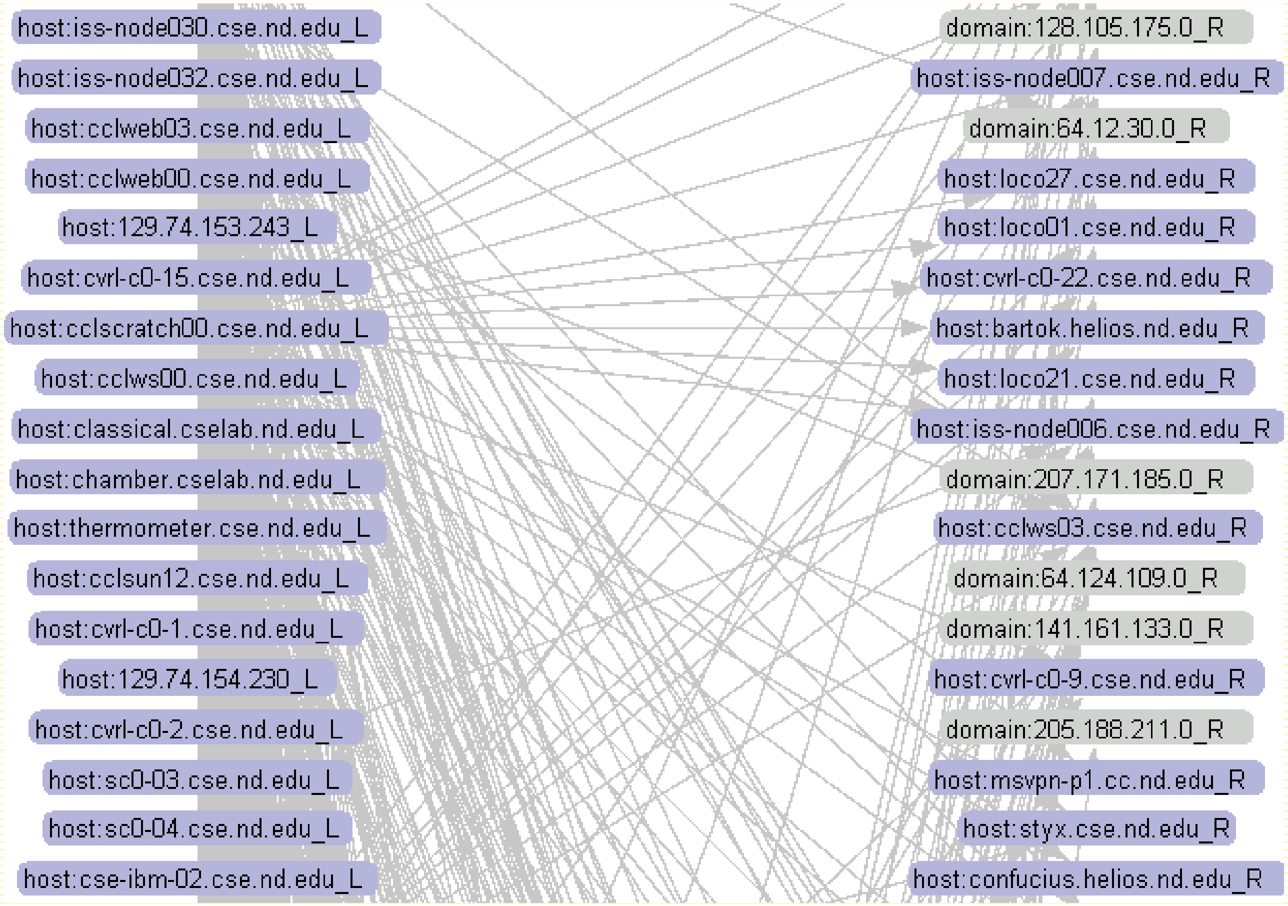
HUA Graph View



Bipartite graphs

- The general ***HUA connectivity graphs*** can be separated into ***(multi-)bipartite graphs***.

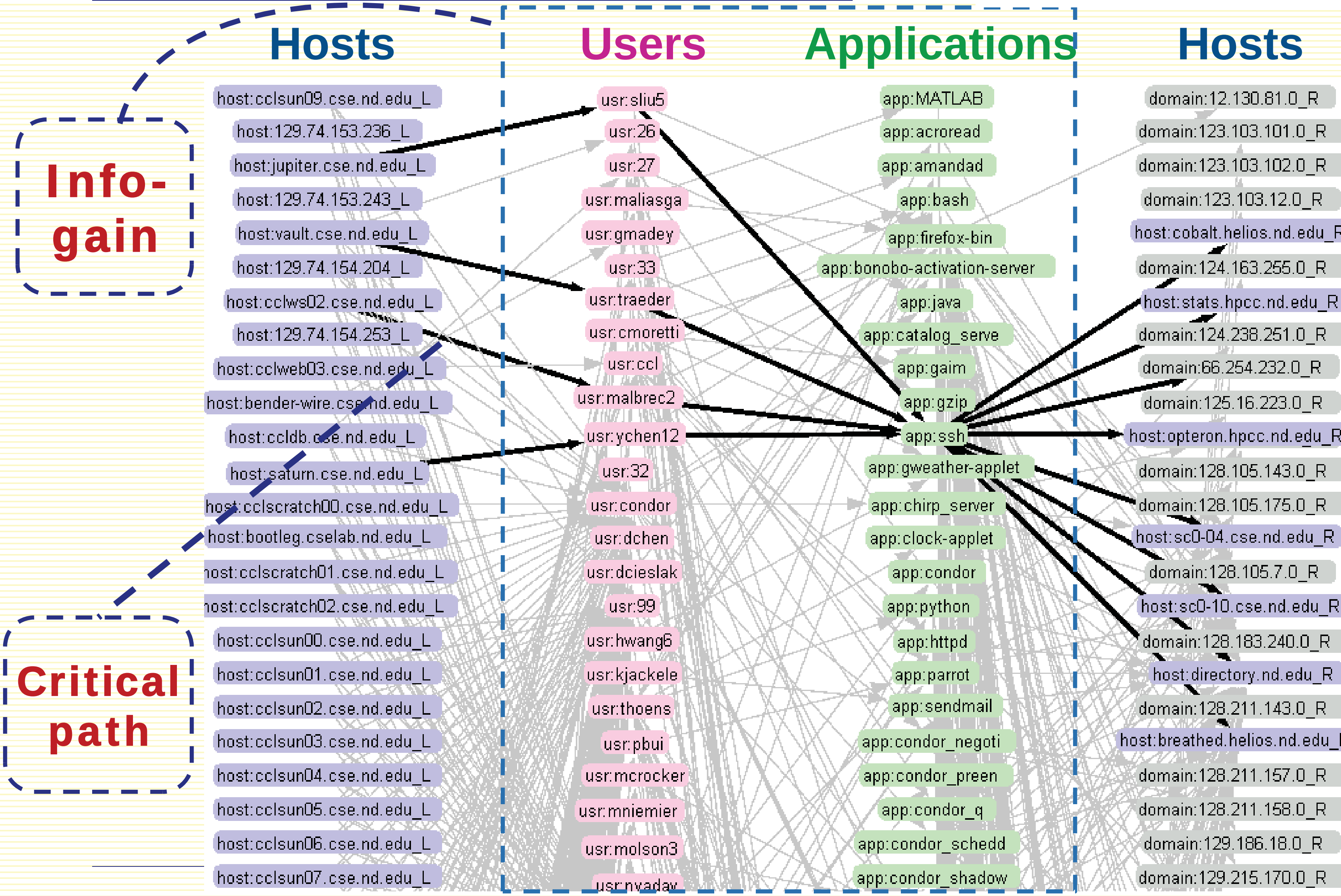
src host



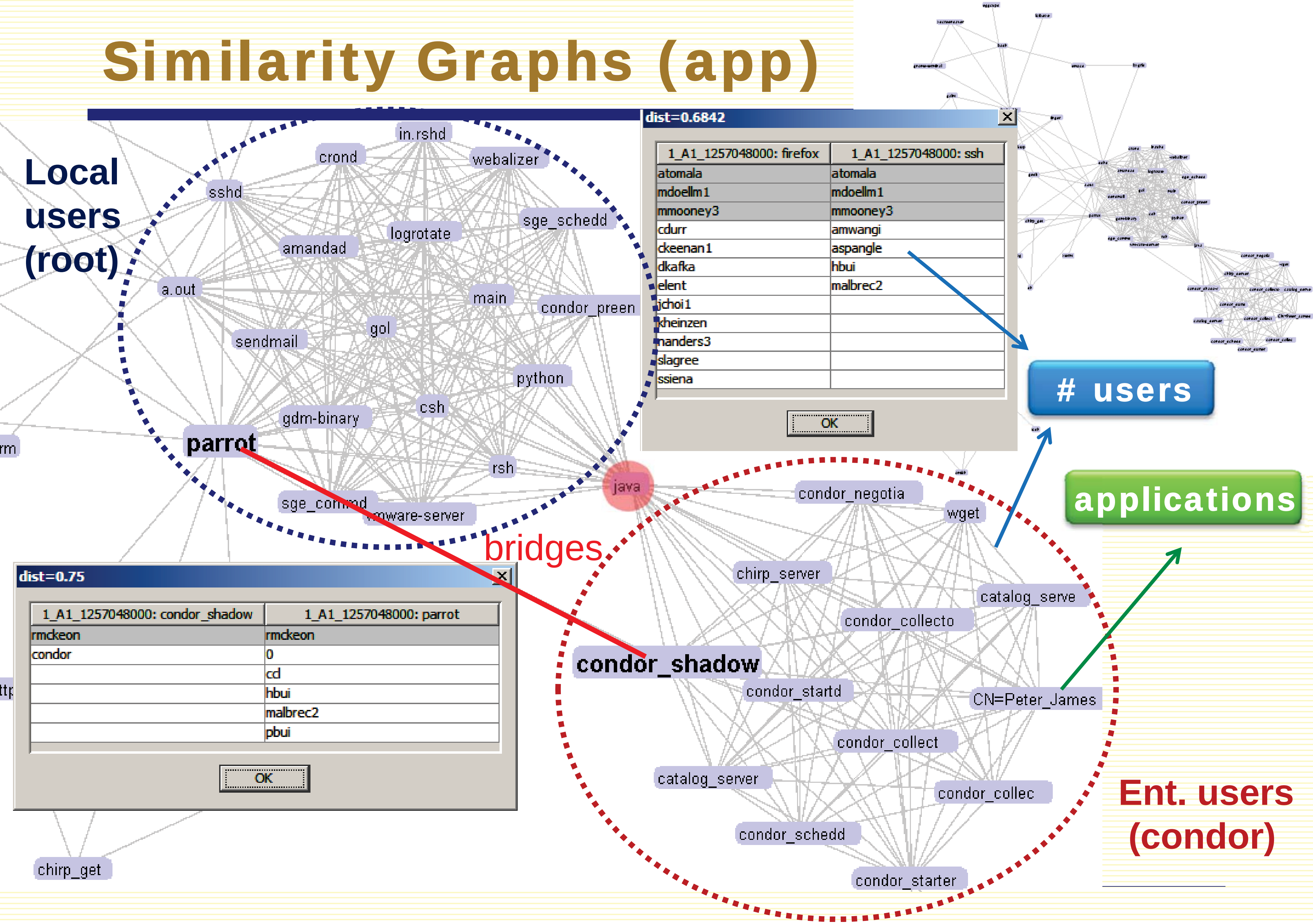
dst host

K-partite graphs

Quadripartite graph



Similarity Graphs (app)



1_A1_1257048000: firefox	1_A1_1257048000: ssh
atomala	atomala
mdoellm1	mdoellm1
mmooney3	mmooney3
cdurr	amwangi
ckeenan1	aspangle
dkafka	hbui
elent	malbrec2
jchoi1	
kheinzen	
manders3	
slagree	
ssiena	

1_A1_1257048000: condor_shadow	1_A1_1257048000: parrot
rmckeon	rmckeon
condor	0
	cd
	hbui
	malbrec2
	pbui

Visual Analysis for Network Management

Data mining / machine learning

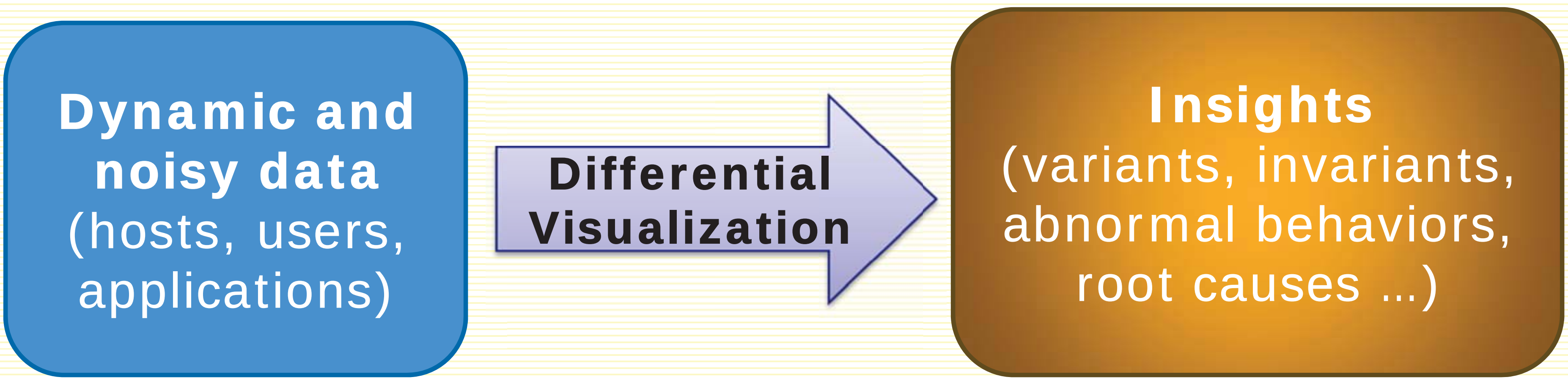
- Automatic
- Algorithmic, analytic methods

Visualization

- Manual
- interactive visual exploration
- Bring in domain knowledge from experienced managers.

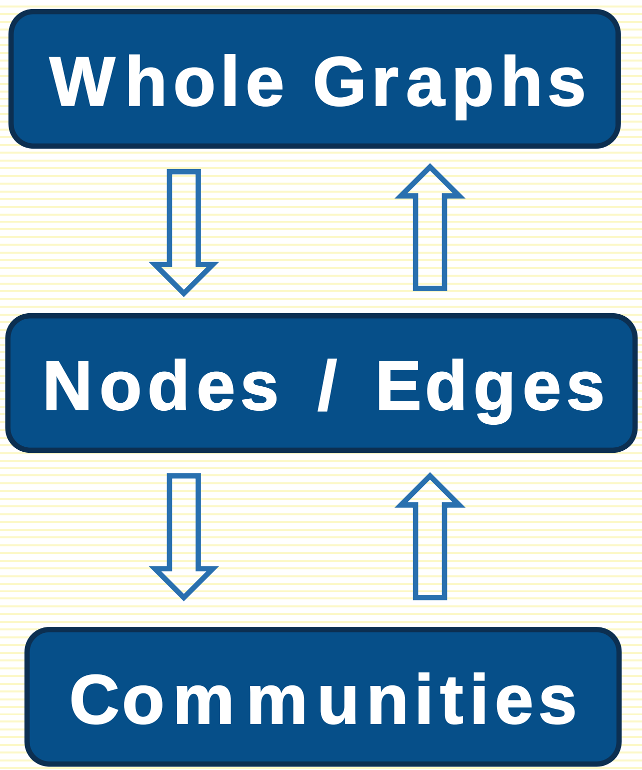
Differential Anomaly Visualization

- What are the changes?
- What are the *variance* and *invariance*?
- How *similar* (*different*) from day-to-day network activities?
- What changes are *normal* / *abnormal*?
- How to quantify and visualize the *evolution* of changes?



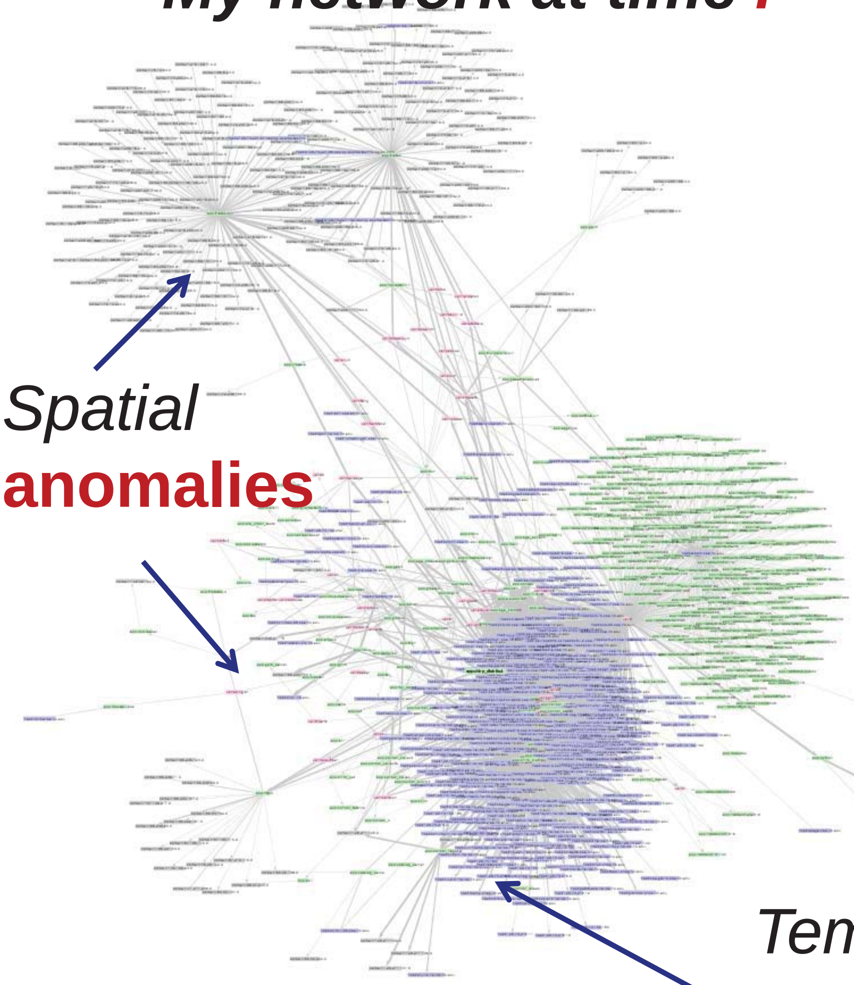
Hierarchical DAV

(overview + context)

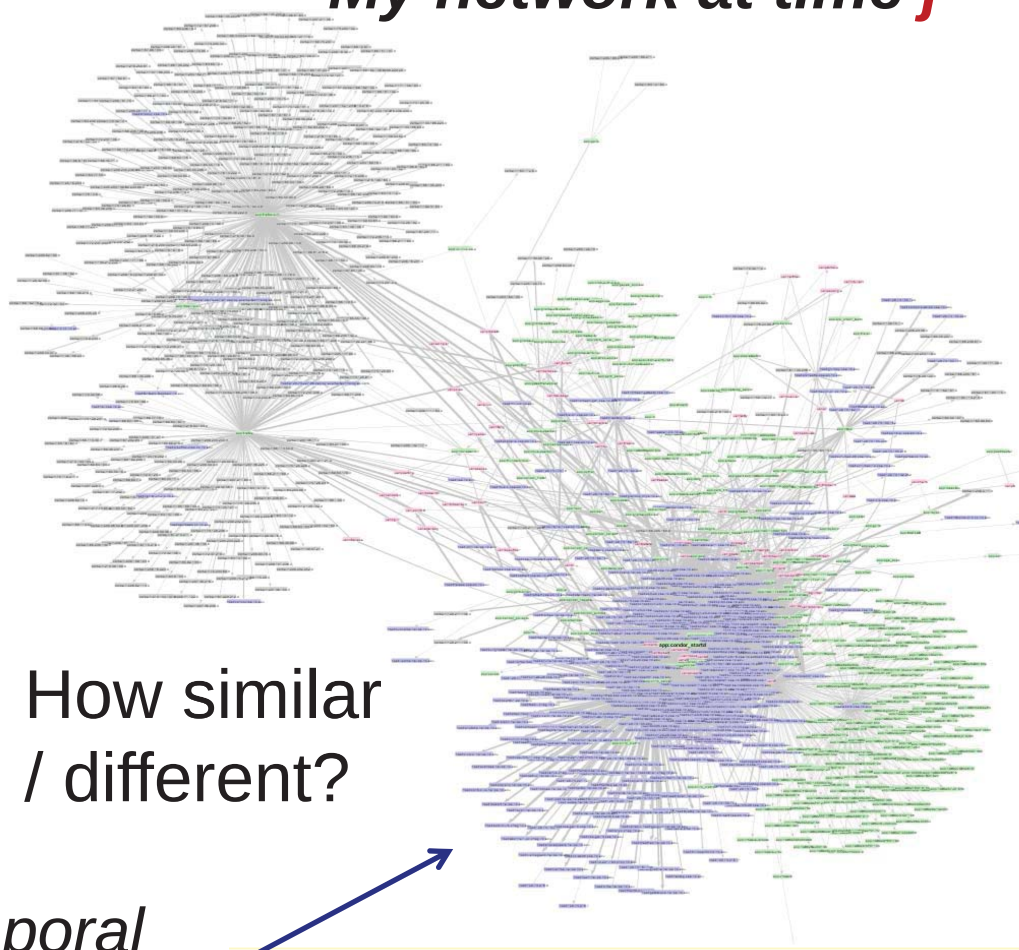


Graph Diff. Anomaly Visualization

My network at time i



My network at time j

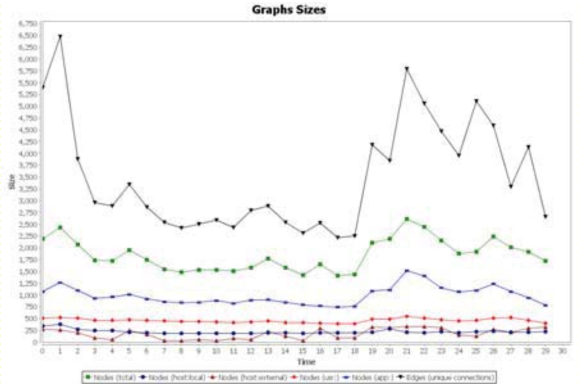


Spatial
anomalies

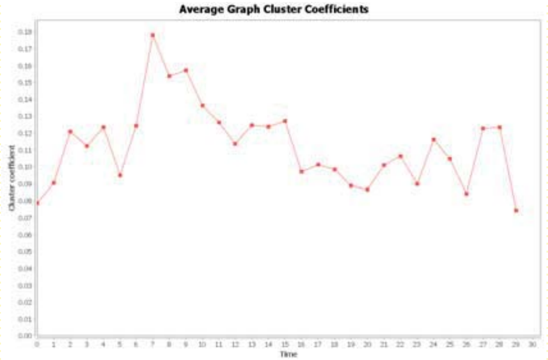
How similar / different?

Temporal
anomalies

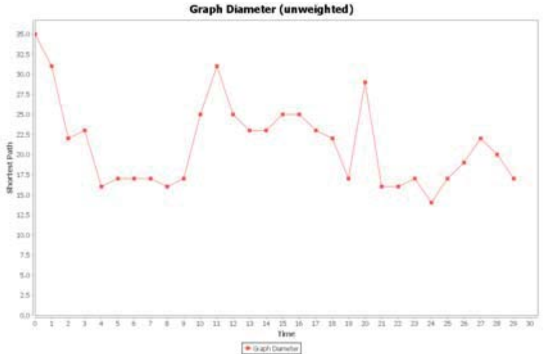
Graph Properties



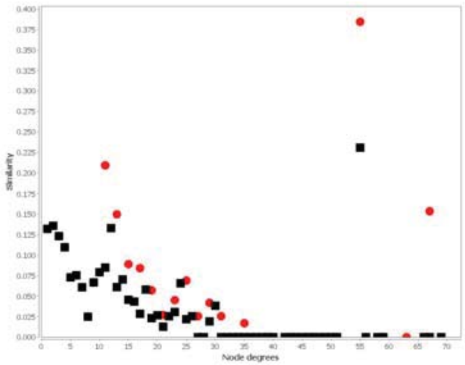
Graph sizes



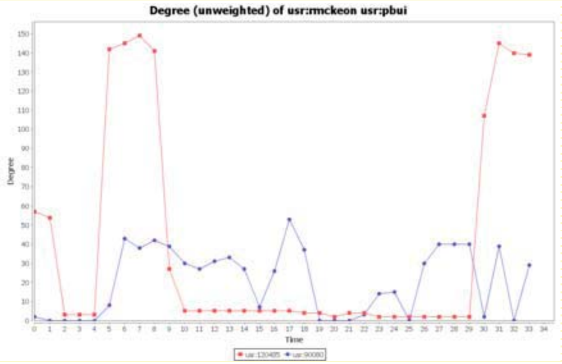
Cluster coefficients



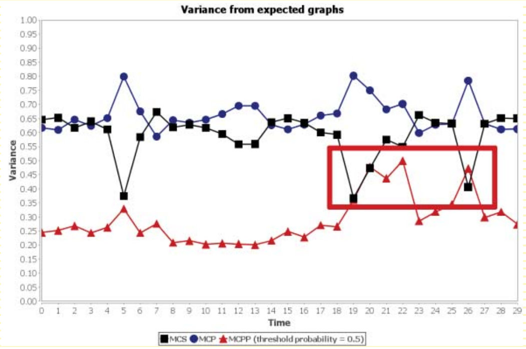
Graph diameters



Degree distributions



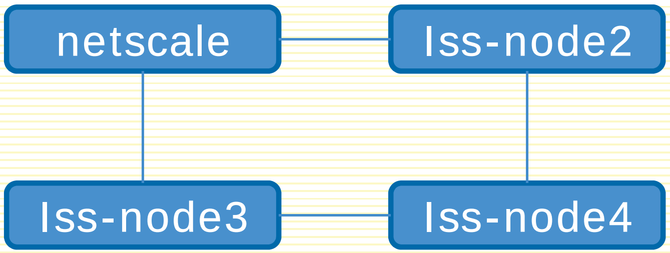
Graph distances

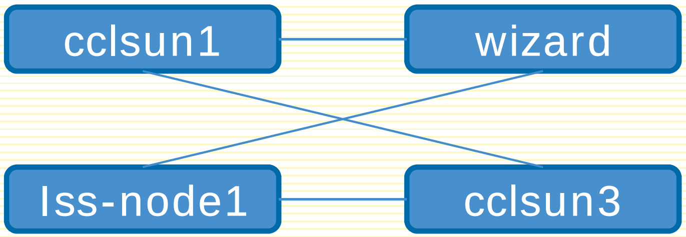


Graph variance scores

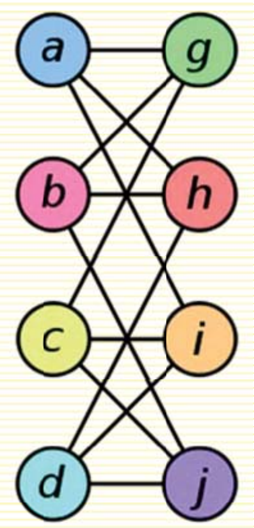
Graph Similarity

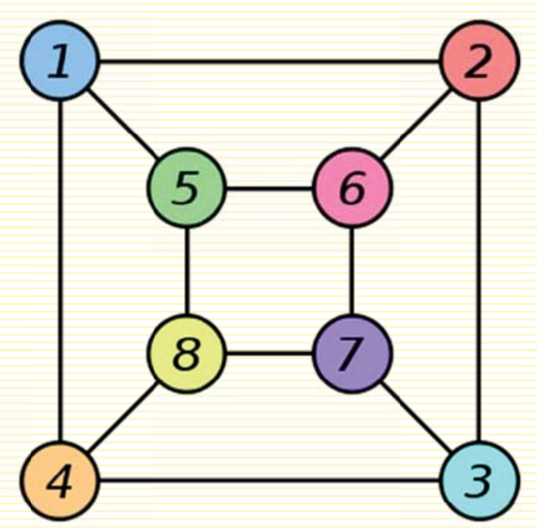
- General graph *isomorphism*





A more complex example





Graph distance

- ***Edit distance***: number of operations required to transform one into the other.
- ***Graph Edit Distance*** (GED) [Bunke07] to measure the graphs' similarities.
- Maximum common subgraphs (MCS) based:

$$d(g_1, g_2) = 1 - \frac{|mcs(g_1, g_2)|}{\max(|g_1|, |g_2|)}$$

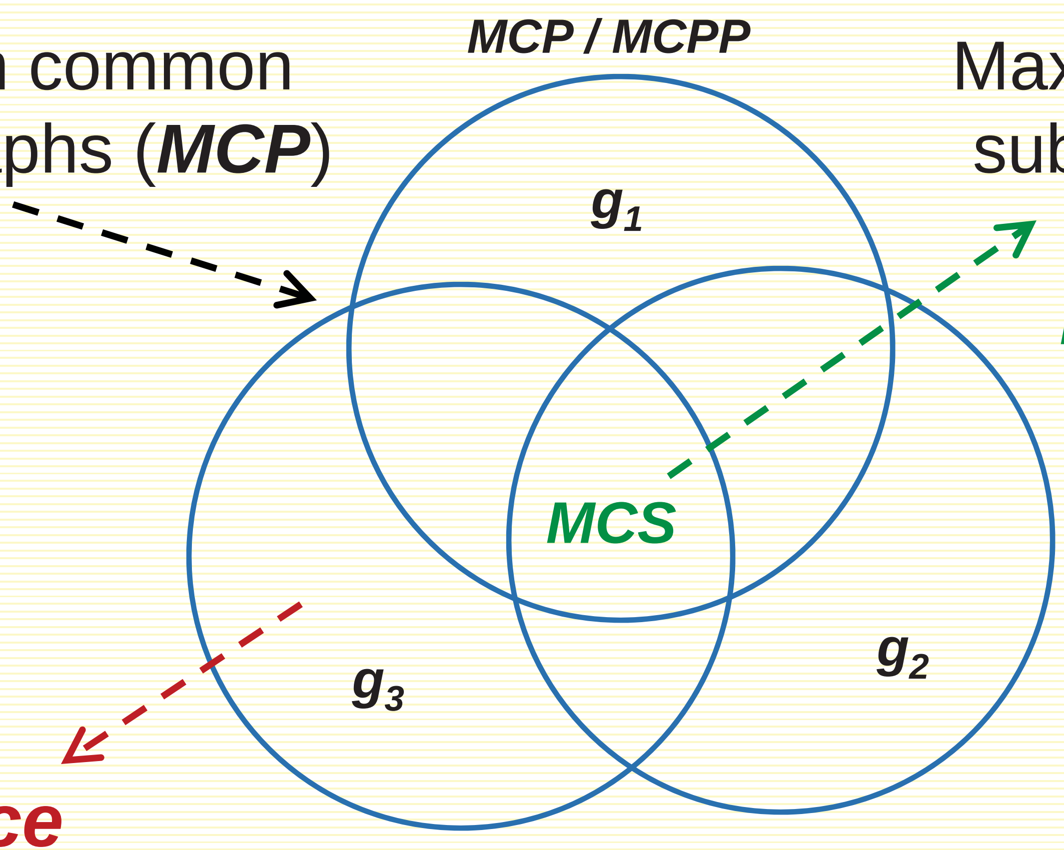
- Graph edit distance (GED) based:

$$d(g_1, g_2) = \frac{|g_1| + |g_2| - 2|mcs(g_1, g_2)|}{|g_1| + |g_2|}$$

Expected Graphs (EG)

Minimum common supergraphs (***MCP***)

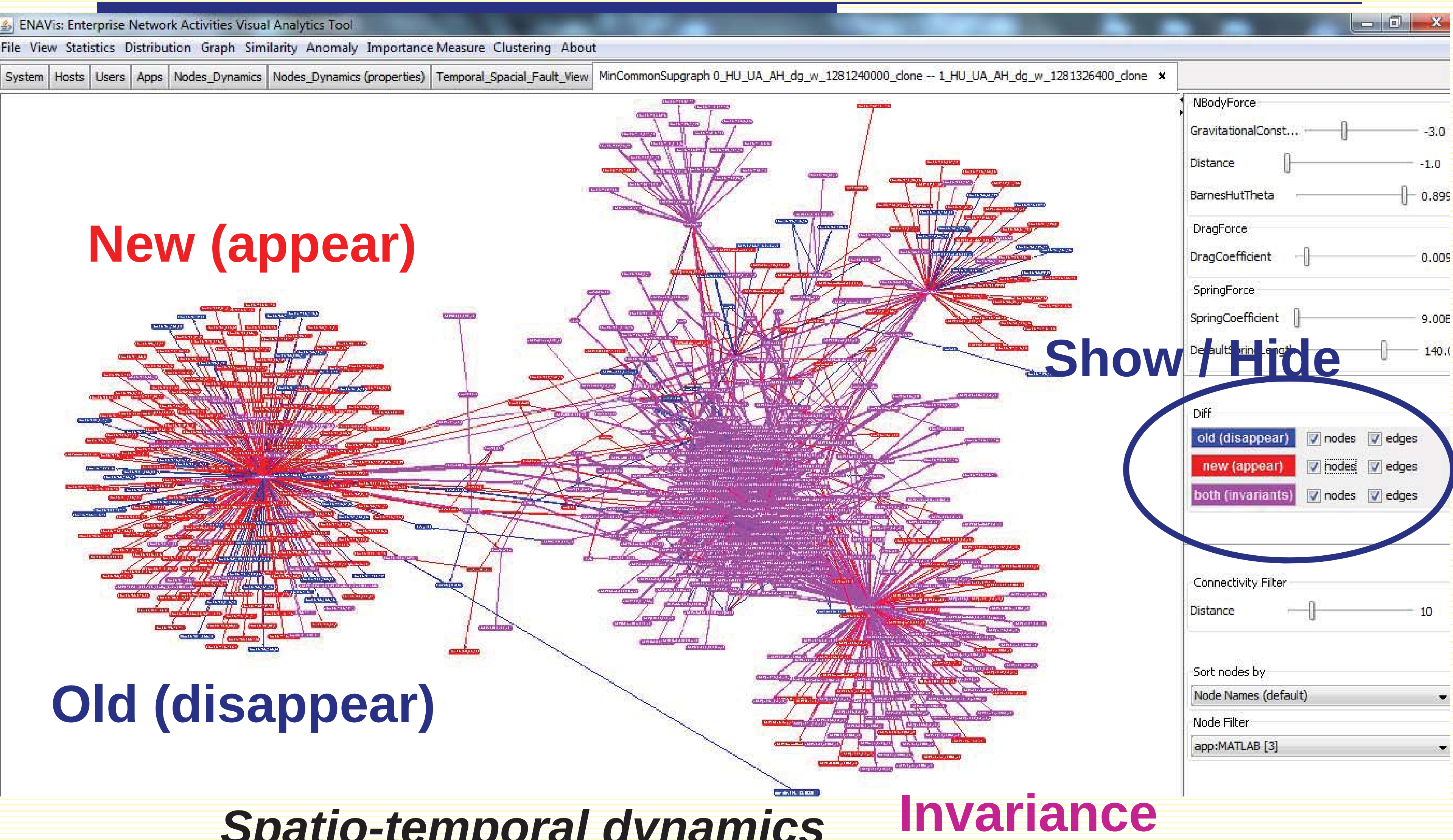
Maximum common subgraphs (***MCS***)
=
invariance



variance

Median Graph (***MG***)

Differential visualization



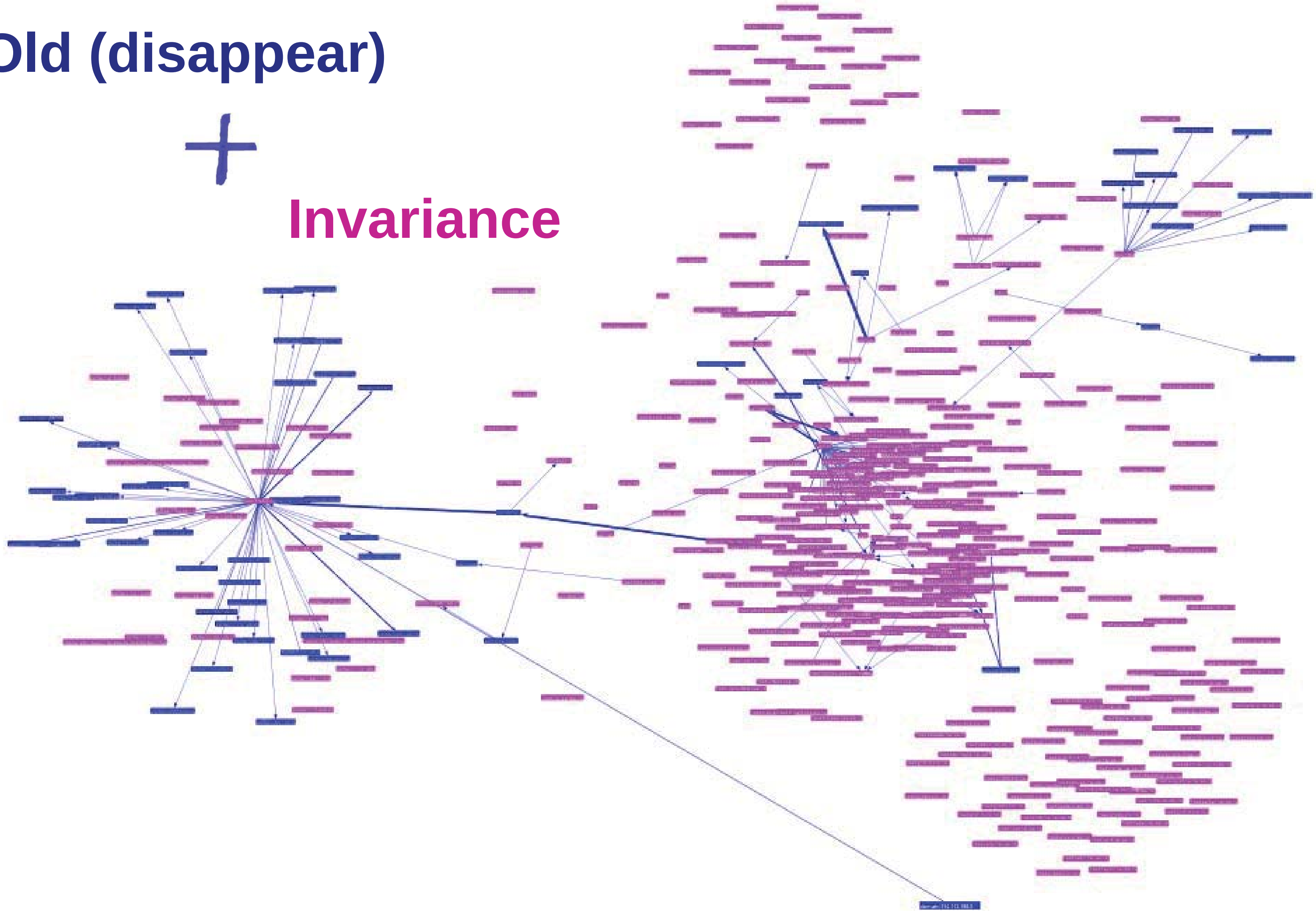
Spatio-temporal dynamics

Differential visualization

Old (disappear)

+

Invariance



Link Anomalies

- Not exactly ***link prediction*** problem.
  - Common neighbors assumption
  - Known nodes only assumption
  - Non-dynamic assumption
- Proof-of-concept
  - Non-linear weighting frequency function

$$P(L_i) = \frac{\sum_{t=1}^{N} w(t) \cdot d}{\sum_{t=1}^{N} w(t)}, \quad d_{t,i} \in \{0,1\} \qquad w(t) = e^{-\lambda(1-\frac{t}{N})}$$

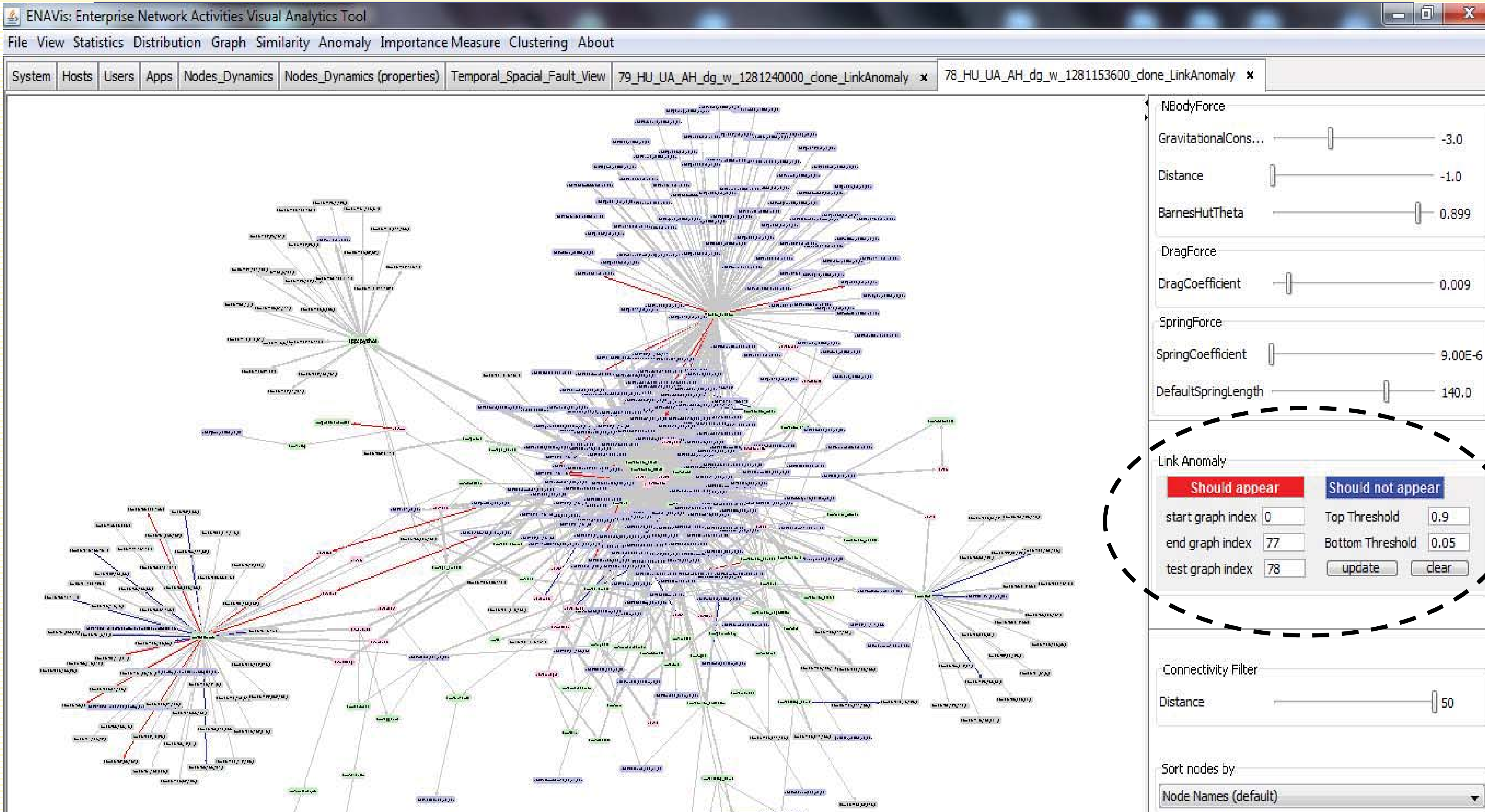
probability of *i*-th link to appear

whether *i*-th link appears at time *t*

non-linear time weighting function

- Can take inputs from future link anomaly algorithms

Link Anomalies Visualization



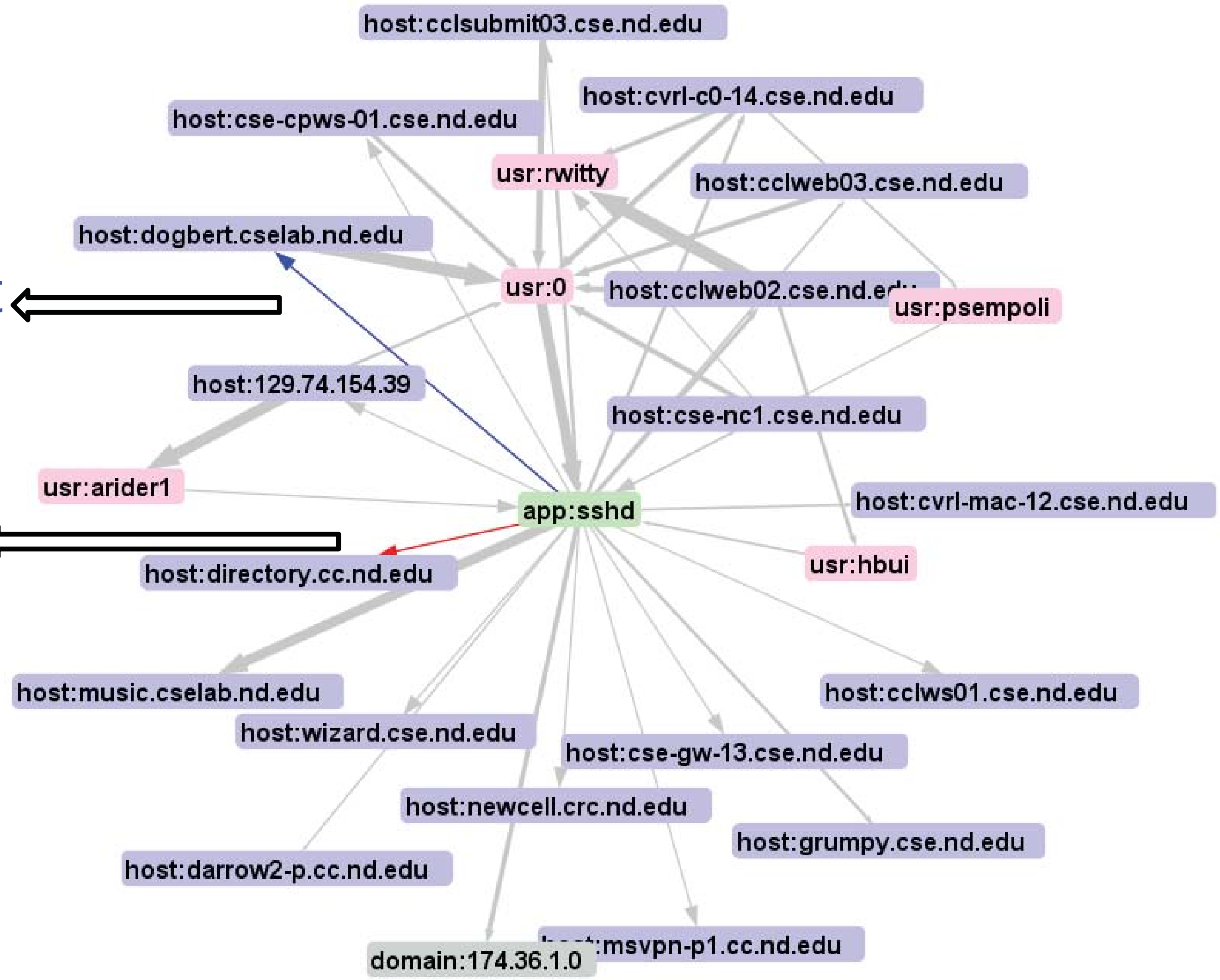
RED: Type-I anomaly: should appear but did not appear

BLUE: Type-II anomaly: should not appear but appeared

Link Anomalies Visualization

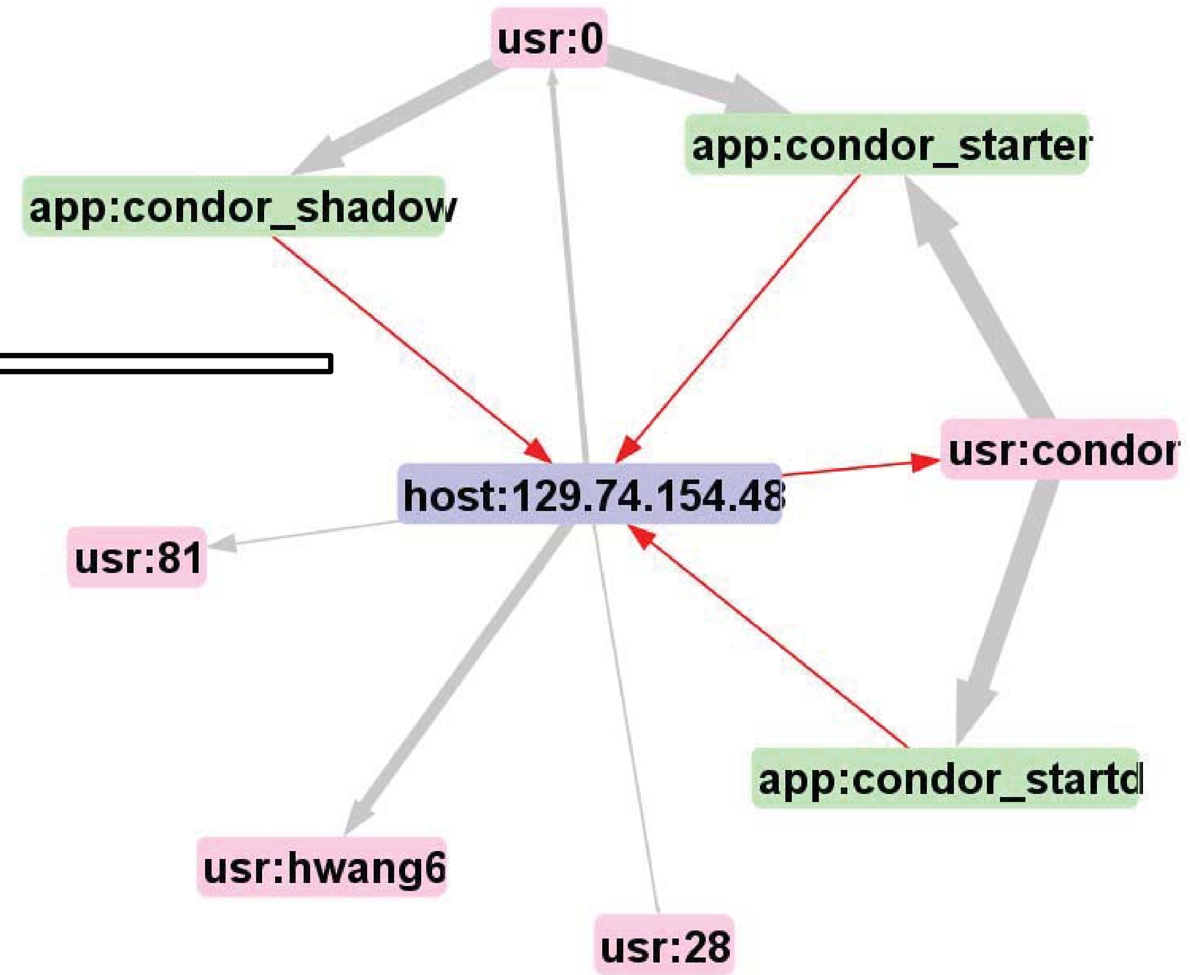
Should not appear

Should appear



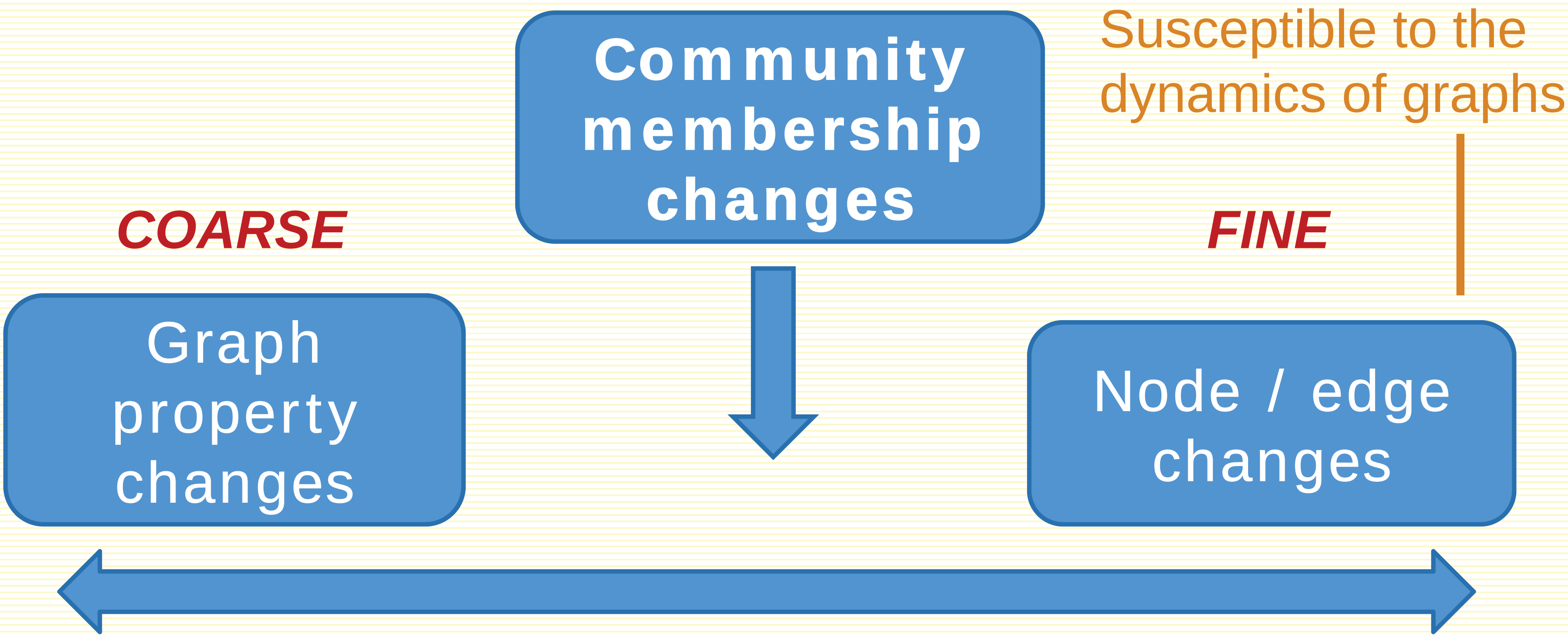
Link Anomalies Visualization

Should appear



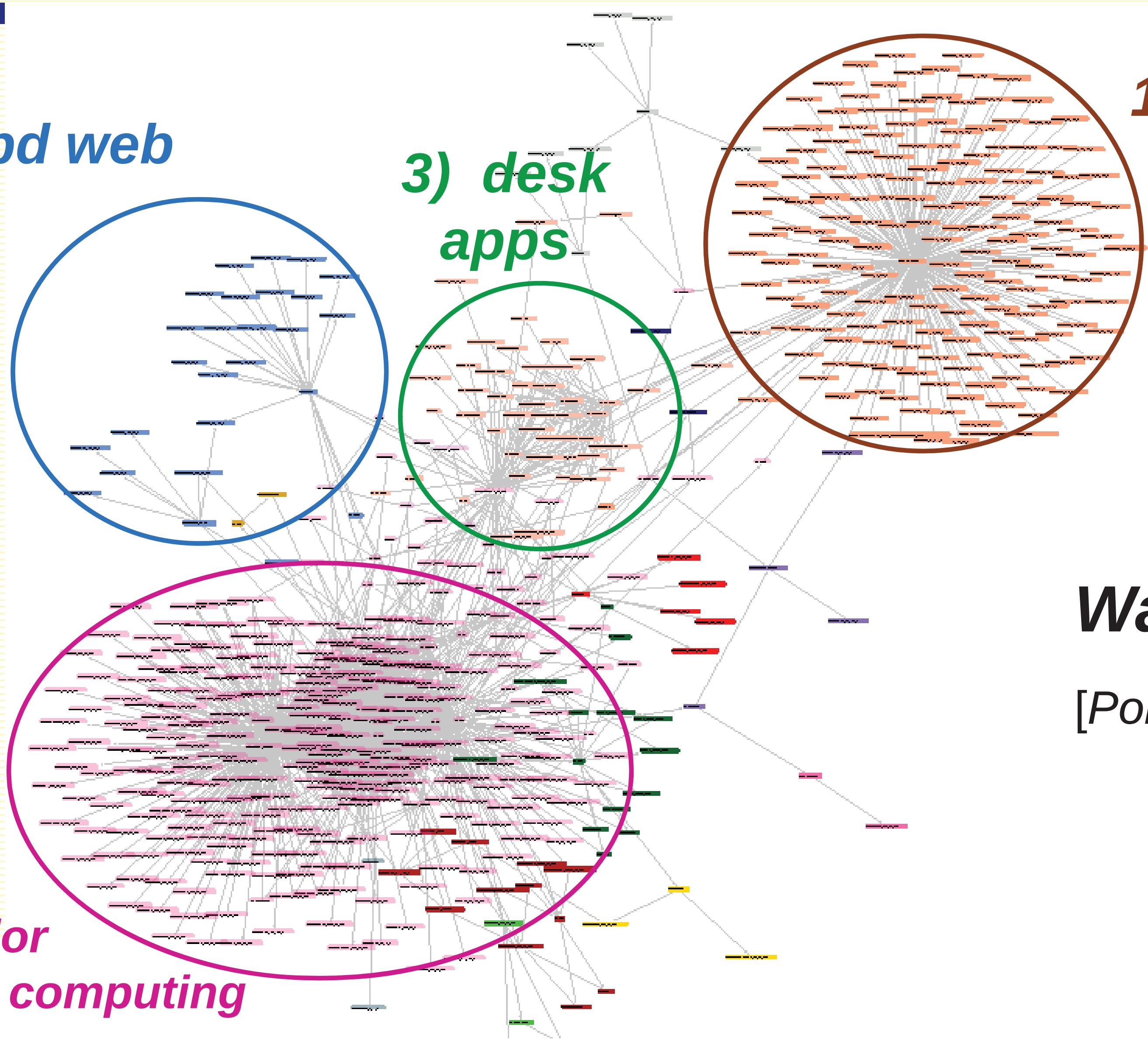
Community-based DAV

- *Intermediate* similarity metric



Balance of granularity and complexity

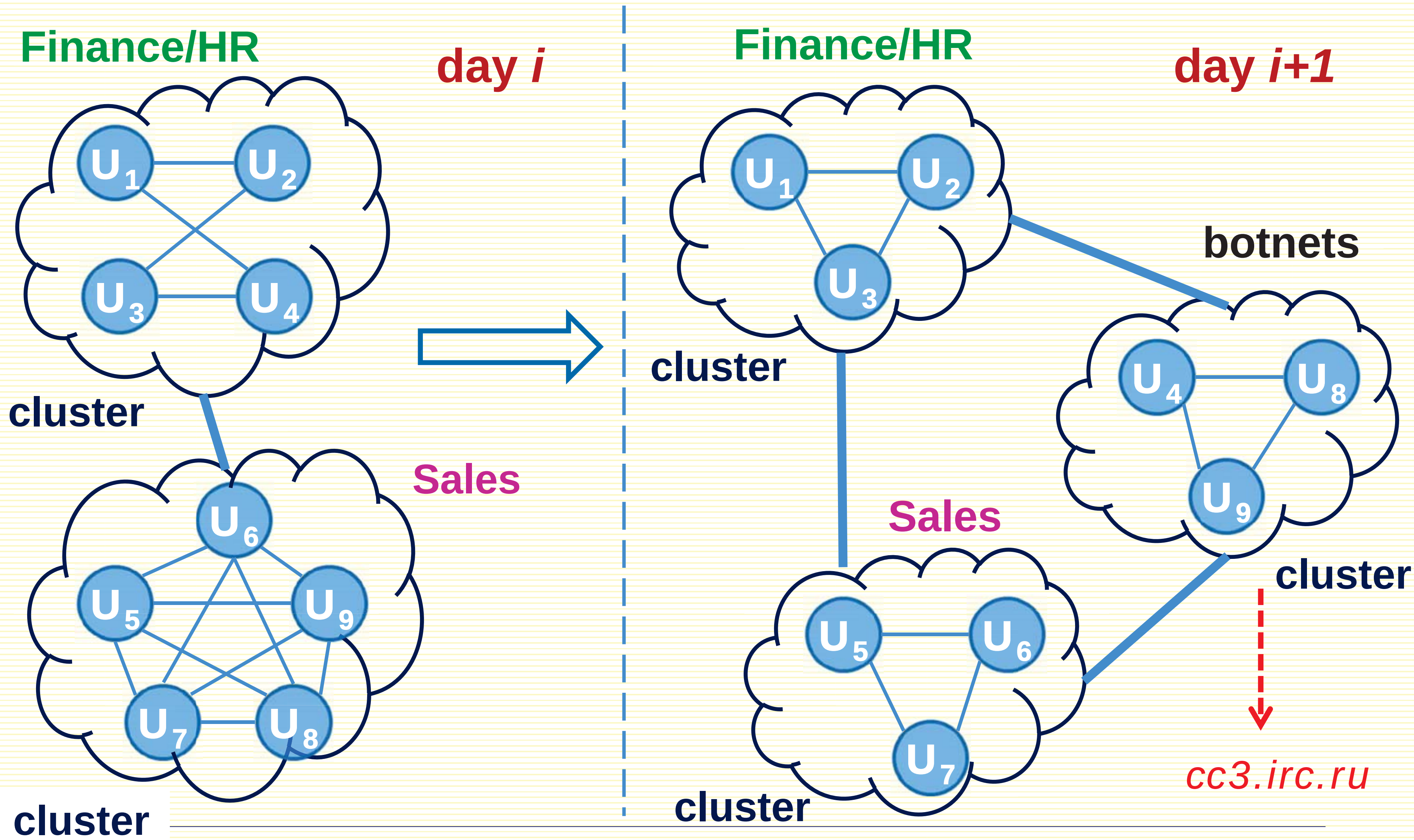
Intra-graph clusters visualization



Walktrap

[*Pons:2006*]

Temporal Community Evolution



Community-based DAV

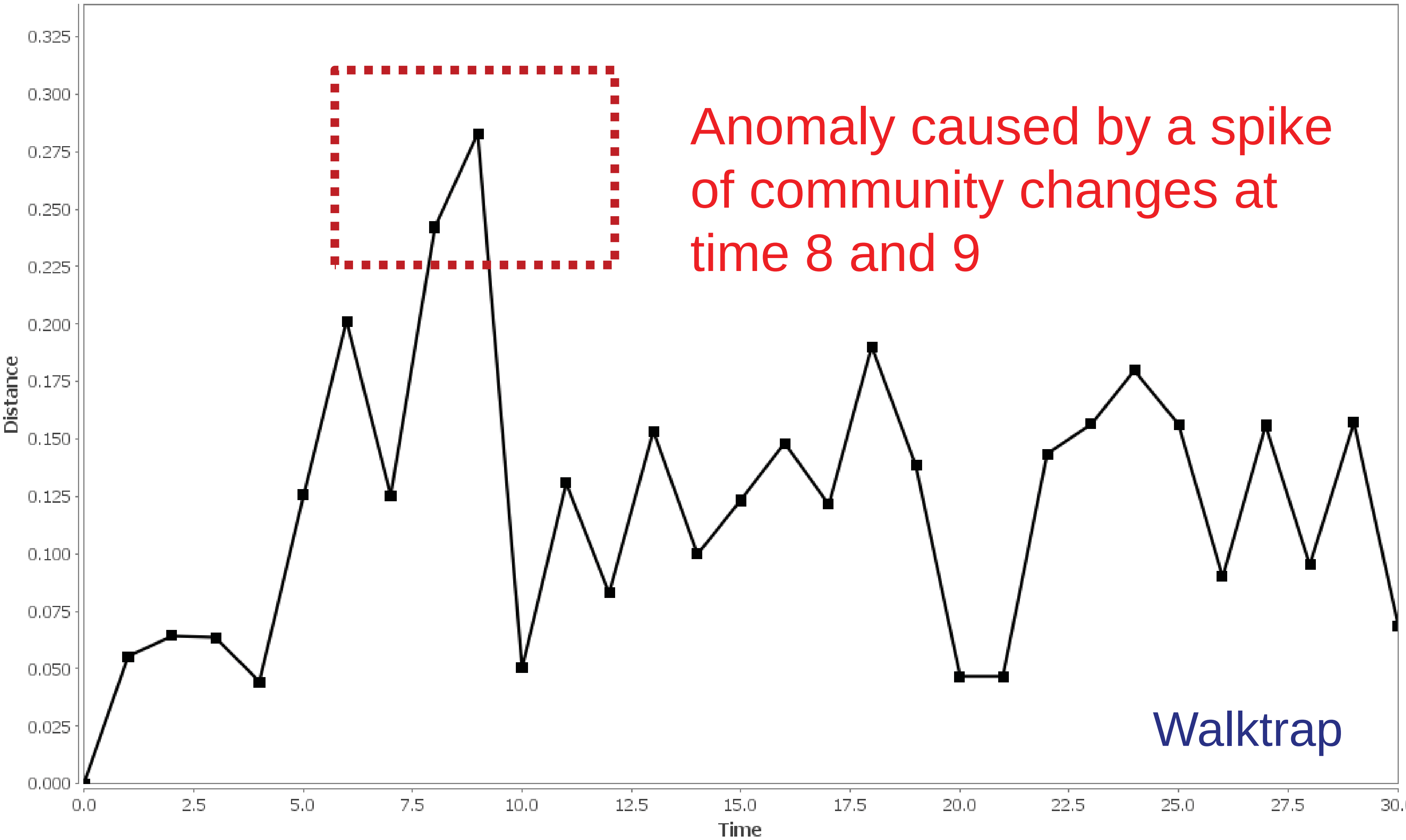
- Graphs changes via ***community similarity***
  - Similar to Rand Index [*Rand71*]

$$dist(C_1, C_2) = 1 - \frac{SS + DD}{SS + SD + DD + DS}$$

- ***Flexibility***
- Suitability for highly ***dynamic*** networks

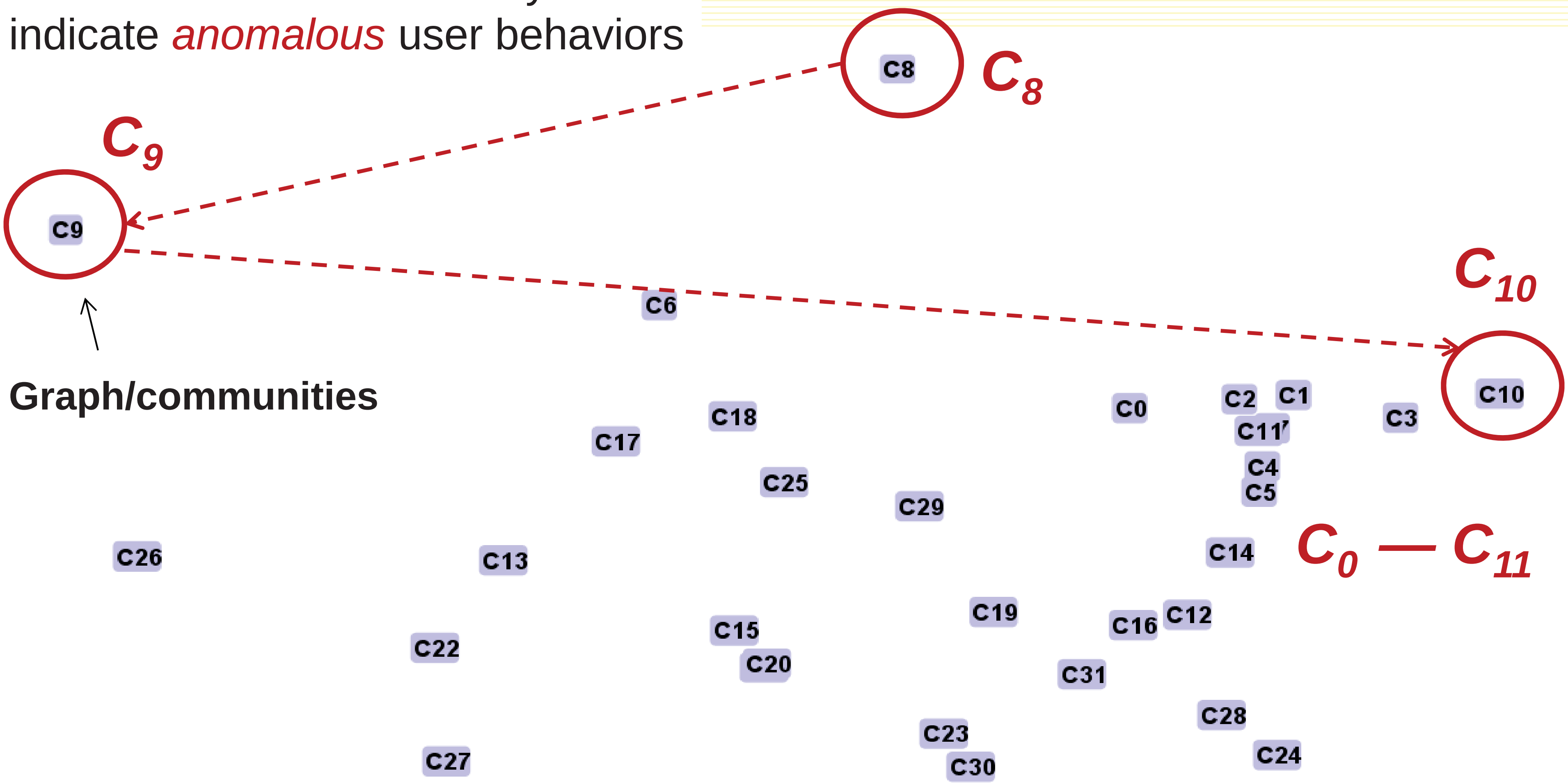
Nodes *consistently* belong to the *same* (or *different*) communities ⟷ **changes are *normal***

Community-based DAV (example)

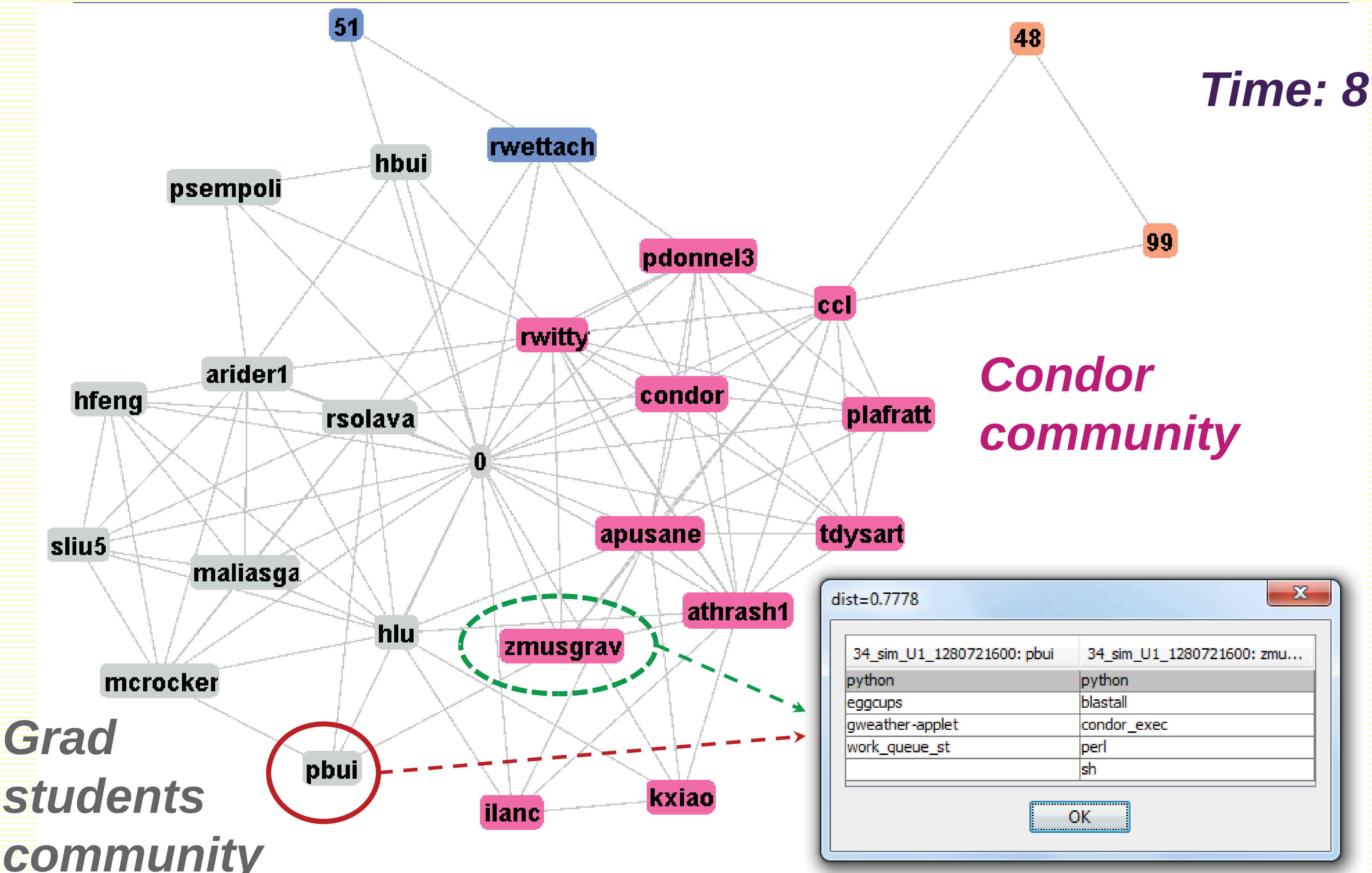


Community-based DAV (MDS view)

Nodes that are farther away indicate *anomalous* user behaviors



Communities of a User Similarity Graph

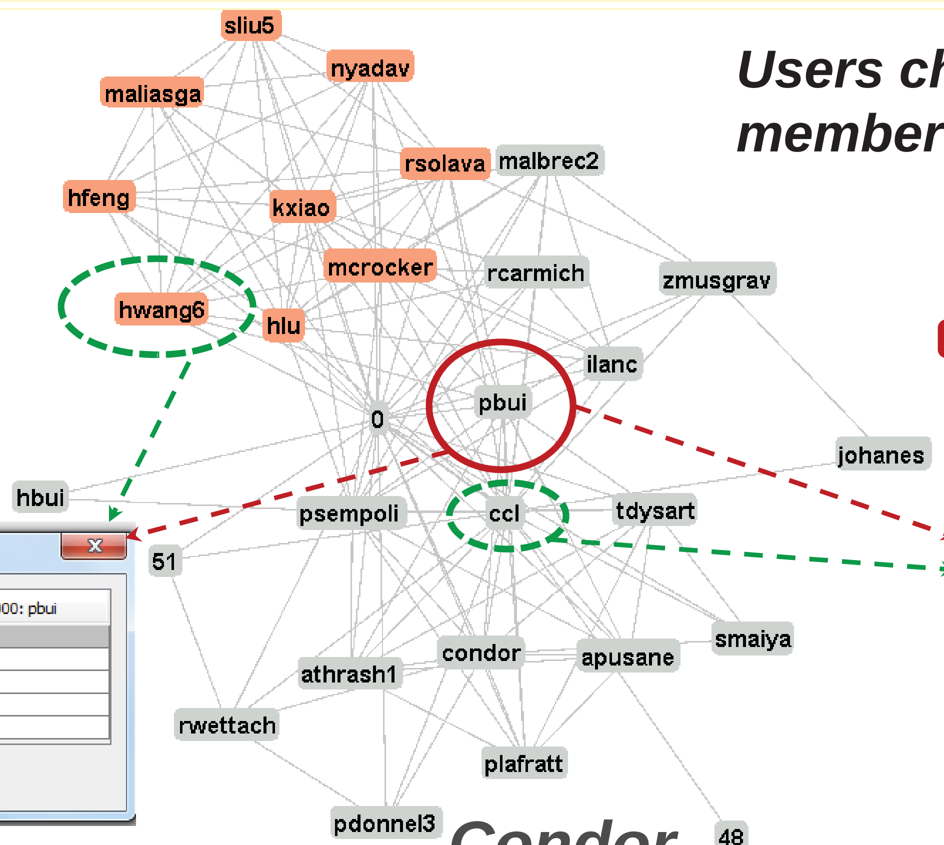


Communities of a User Similarity Graph

Time: 9

Grad students community

Users change community membership



dist=0.7778

35_sim_U1_1280808000: hwang6	35_sim_U1_1280808000: pbui
eggcups	eggcups
acroread	condor_shadow
firefox-bin	gweather-applet
rdesktop	python
	work_queue_st

OK

dist=0.913

35_sim_U1_1280808000: ccl	35_sim_U1_1280808000: pbui
condor_shadow	condor_shadow
chirp	eggcups
condor_exec	gweather-applet
condor_q	python
condor_starter	work_queue_st
condor_status	
condor_submit	
condor_vacate	
cp	
curl	
du	
httpd	
muon_worker	
parrot	
perl	
sendmail	
sh	
ssh	

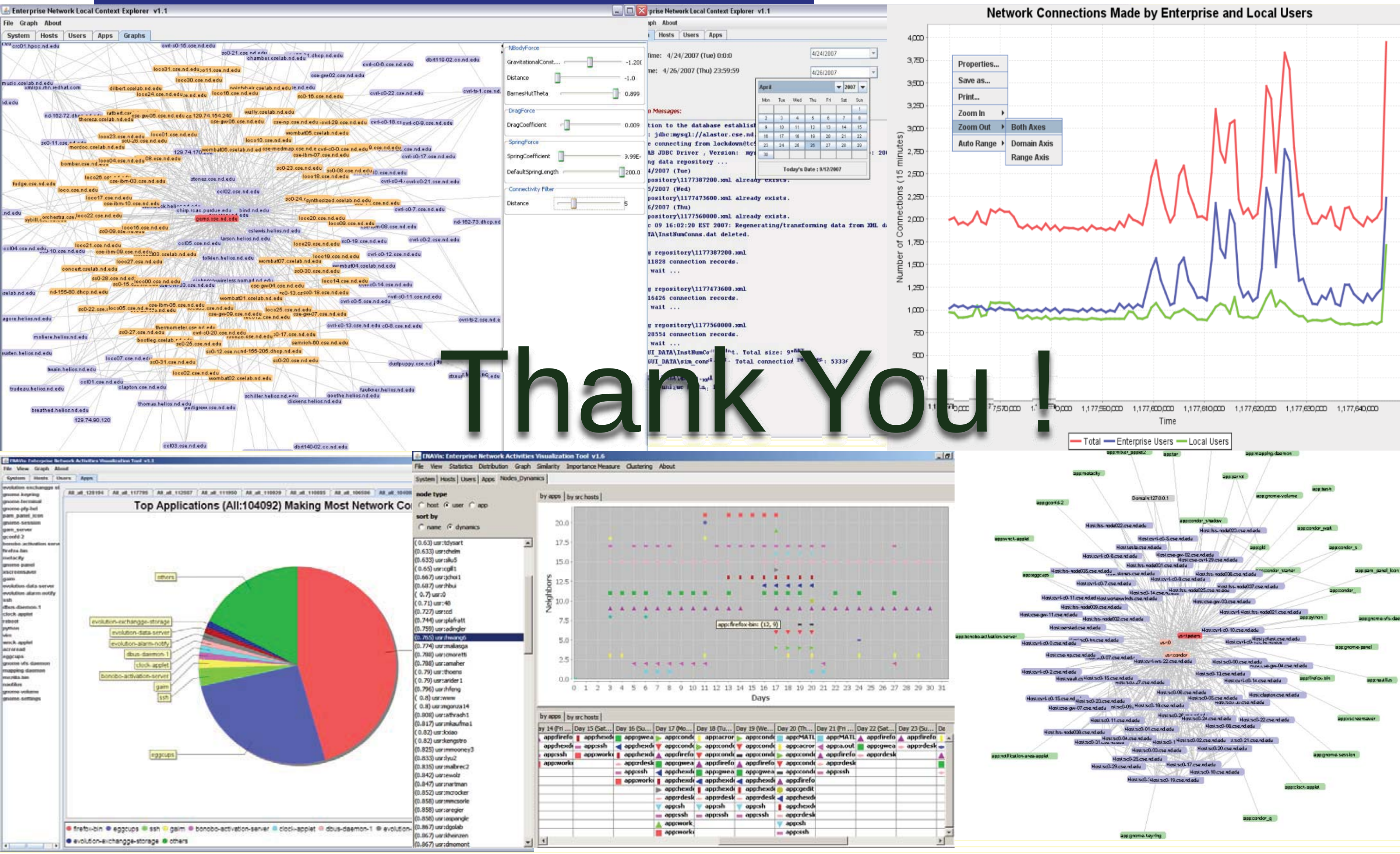
OK

Condor community

Conclusion

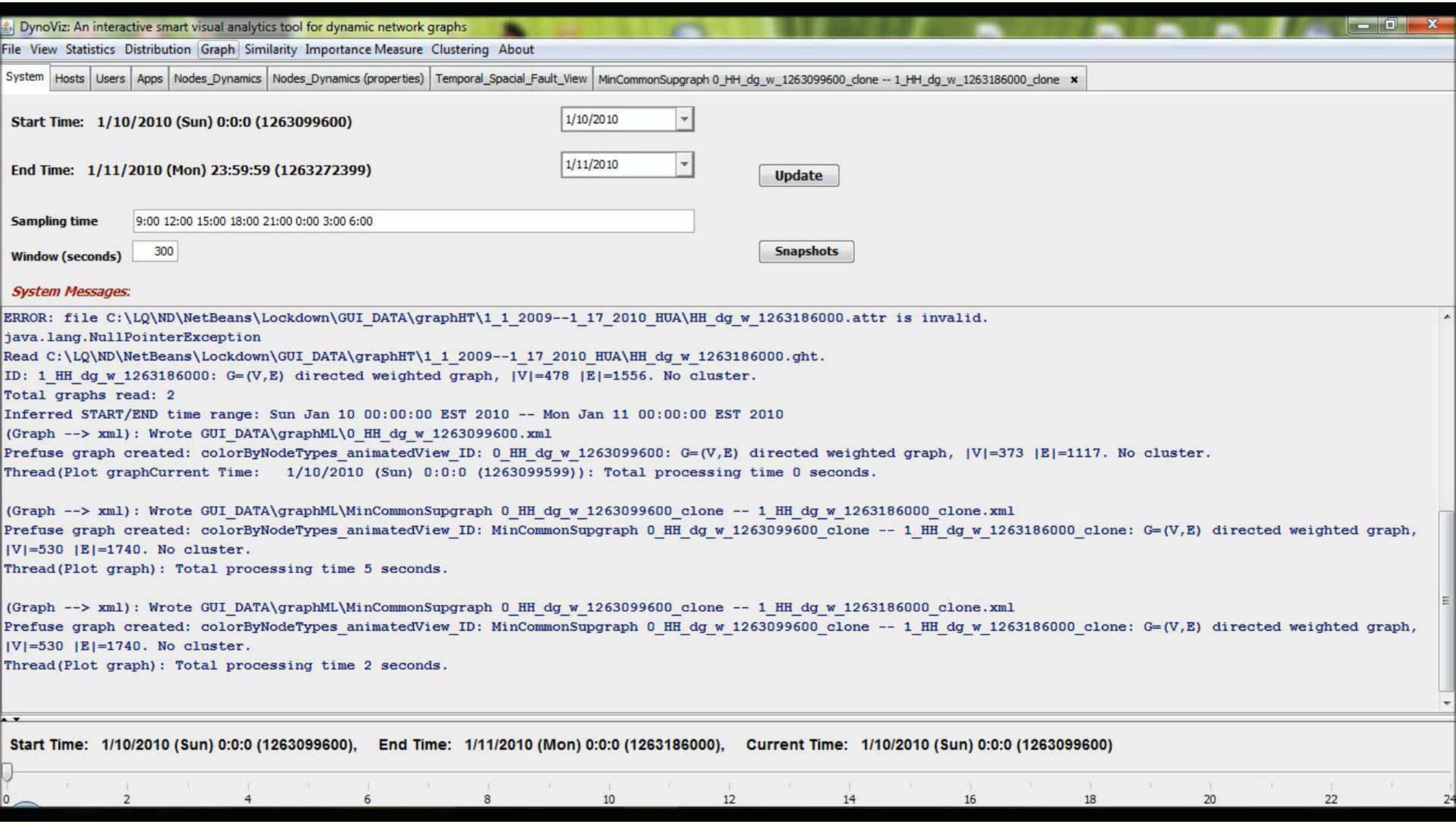
- Network (security) management is hard.
  - Large scale, heterogeneity, dynamics, complexity
- Anomaly detection and analysis is important yet challenging.
- We developed a novel hierarchical graph *differential anomaly visualization* (DAV) framework
  - Combines automated graph data mining and manual exploration.
  - At different levels: Graphs, Nodes/Edges, Communities
- Completeness
  - Overview vs. Details-on-demand
  - Exact changes vs. Dynamic churns
  - Detection vs. root causes
- *DAV:* intelligent, *time-efficient* management alternative.

More info visit http://cps.cmich.edu/liao1q



Thank You !

Questions


DynoViz: An interactive smart visual analytics tool for dynamic network graphs
File View Statistics Distribution Graph Similarity Importance Measure Clustering About
System Hosts Users Apps Nodes_Dynamics Nodes_Dynamics (properties) Temporal_Spacial_Fault_View MinCommonSupgraph 0_HH_dg_w_1263099600_clone -- 1_HH_dg_w_1263186000_clone
Start Time: 1/10/2010 (Sun) 0:0:0 (1263099600)
1/10/2010
End Time: 1/11/2010 (Mon) 23:59:59 (1263272399)
1/11/2010
Update
Sampling time
9:00 12:00 15:00 18:00 21:00 0:00 3:00 6:00
Window (seconds)
300
Snapshots
System Messages:
ERROR: file C:\LQ\ND\NetBeans\Lockdown\GUI_DATA\graphHT\1_1_2009--1_17_2010_HUA\HH_dg_w_1263186000.attr is invalid.
java.lang.NullPointerException
Read C:\LQ\ND\NetBeans\Lockdown\GUI_DATA\graphHT\1_1_2009--1_17_2010_HUA\HH_dg_w_1263186000.ght.
ID: 1_HH_dg_w_1263186000: G=(V,E) directed weighted graph, |V|=478 |E|=1556. No cluster.
Total graphs read: 2
Inferred START/END time range: Sun Jan 10 00:00:00 EST 2010 -- Mon Jan 11 00:00:00 EST 2010
(Graph --> xml): Wrote GUI_DATA\graphML\0_HH_dg_w_1263099600.xml
Prefuse graph created: colorByNodeTypes_animatedView_ID: 0_HH_dg_w_1263099600: G=(V,E) directed weighted graph, |V|=373 |E|=1117. No cluster.
Thread(Plot graphCurrent Time: 1/10/2010 (Sun) 0:0:0 (1263099599)): Total processing time 0 seconds.
(Graph --> xml): Wrote GUI_DATA\graphML\MinCommonSupgraph 0_HH_dg_w_1263099600_clone -- 1_HH_dg_w_1263186000_clone.xml
Prefuse graph created: colorByNodeTypes_animatedView_ID: MinCommonSupgraph 0_HH_dg_w_1263099600_clone -- 1_HH_dg_w_1263186000_clone: G=(V,E) directed weighted graph, |V|=530 |E|=1740. No cluster.
Thread(Plot graph): Total processing time 5 seconds.
(Graph --> xml): Wrote GUI_DATA\graphML\MinCommonSupgraph 0_HH_dg_w_1263099600_clone -- 1_HH_dg_w_1263186000_clone.xml
Prefuse graph created: colorByNodeTypes_animatedView_ID: MinCommonSupgraph 0_HH_dg_w_1263099600_clone -- 1_HH_dg_w_1263186000_clone: G=(V,E) directed weighted graph, |V|=530 |E|=1740. No cluster.
Thread(Plot graph): Total processing time 2 seconds.
Start Time: 1/10/2010 (Sun) 0:0:0 (1263099600), End Time: 1/11/2010 (Mon) 0:0:0 (1263186000), Current Time: 1/10/2010 (Sun) 0:0:0 (1263099600)
0 2 4 6 8 10 12 14 16 18 20 22 24