

Visualizing Graph Dynamics and Similarity for Enterprise Network Security and Management

Qi Liao, Aaron Striegel and Nitesh Chawla

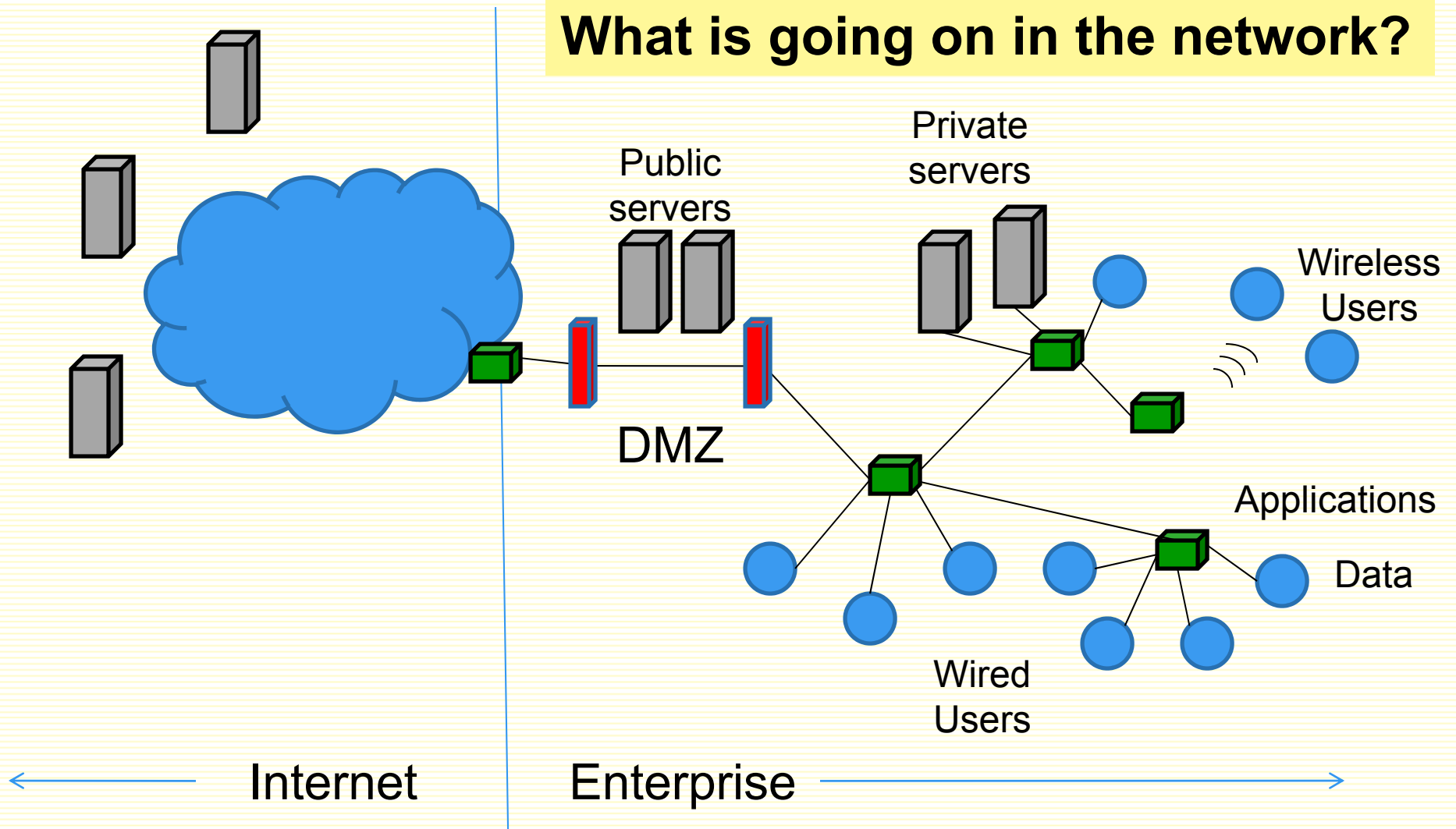
Computer Science and Engineering

University of Notre Dame, USA.



Enterprise Network Management

What is going on in the network?



Traditional Logging

- Network connectivity *logging* usually in form of *IP addresses* and *port numbers*.
- Traditional Cisco **NetFlow** definition:

Ingress Interface	IP Protocol	IP TOS	Src IP	Dst IP	Src port	Dst port
-------------------	-------------	--------	--------	--------	----------	----------

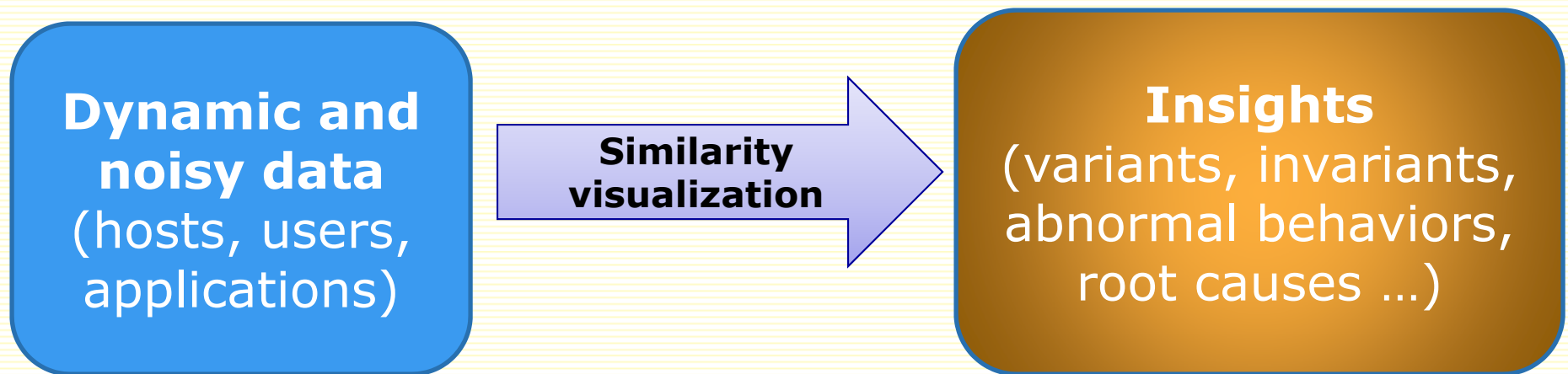
- **Where**, but not **Who** and **What**
- Visual analysis of **hosts**, **users** and **applications** is more important and harder than traditional IP/ports visualization [Hertzog06, Lalanne07, Liao08]

Introduction

- Security management of enterprise networks is hard
 - Users and applications
 - Complex interrelationships
 - So dynamic, constantly changing
 - No clean signal. Traditional data mining for anomaly detection falls short
- Understanding the **dynamics** / similarities is non-trivial
 - Important step for anomaly detection

Similarity visualization

- First step to understand abnormality
- Key questions:
 - What are the changes?
 - What changes are (ab)normal?
 - How different (or similar) from day-to-day activities?
 - How to effectively visualize them?



Hierarchical Similarity Visualization

Graphs

Top-down manner
(overview + context)

- HUA connectivity graphs
- Bipartite graphs
- Similarity graphs

Inter-graphs

- Among network graphs across multiple timelines.

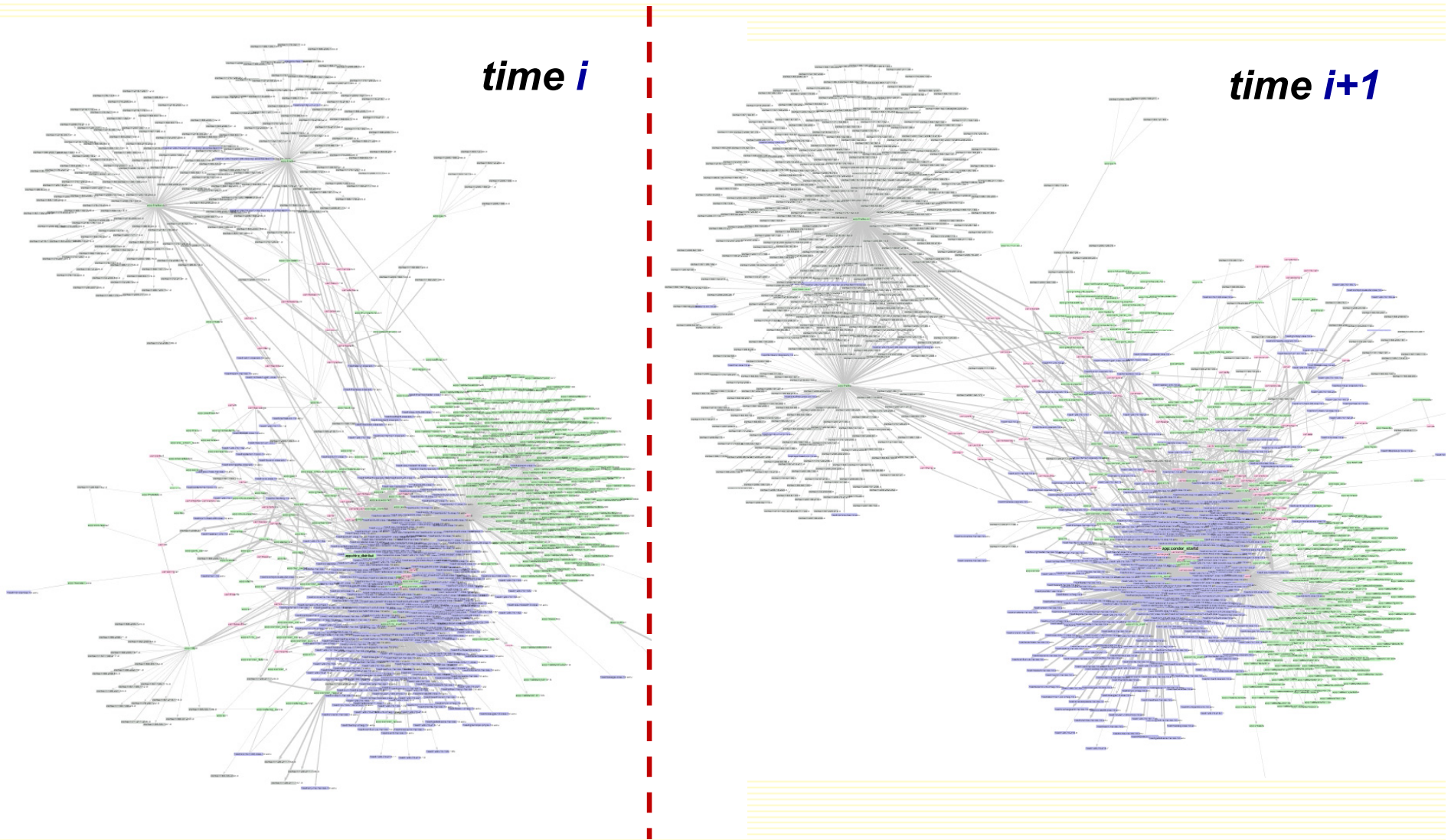
Intra-graph

- Identify similar structure within each individual

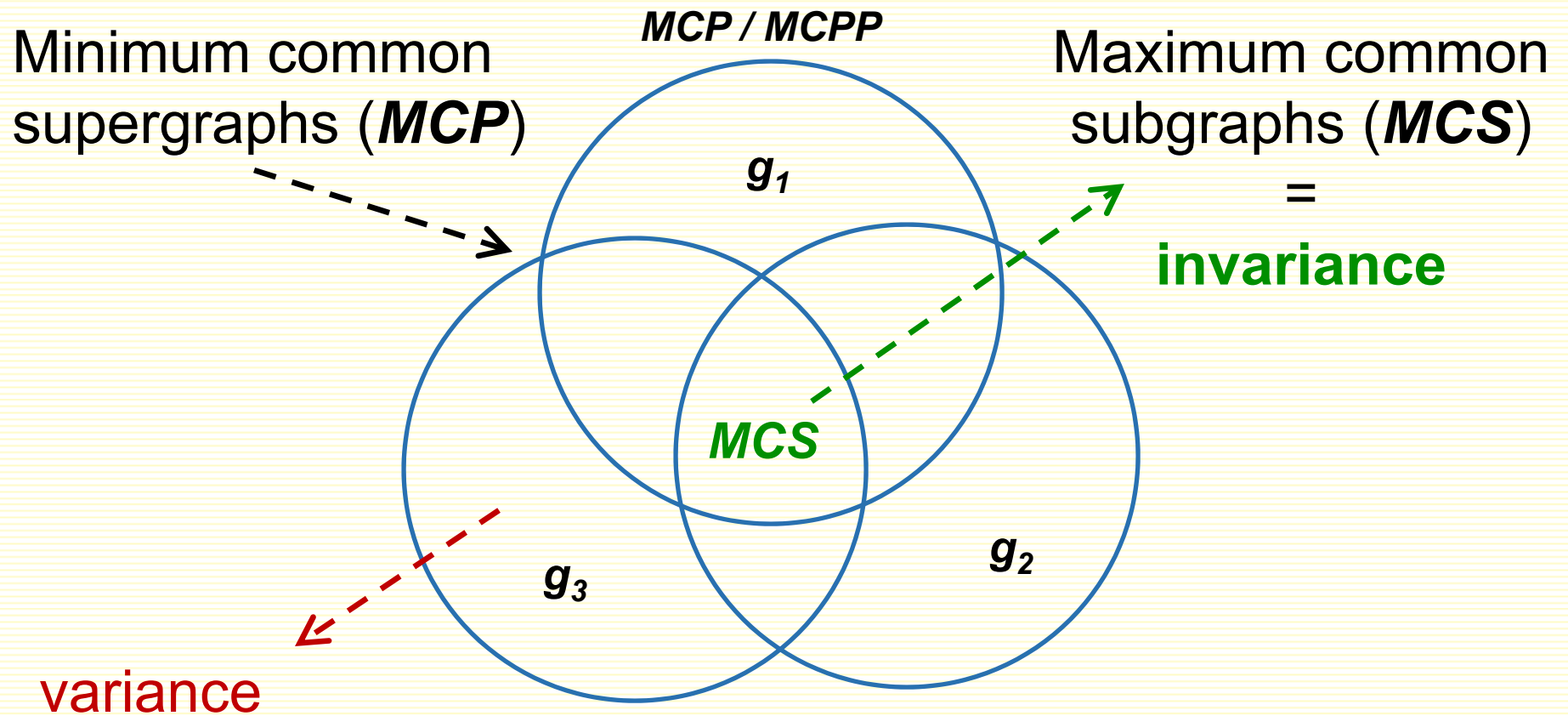
nodes

- Dynamic of neighborhoods changes at each individual node level.

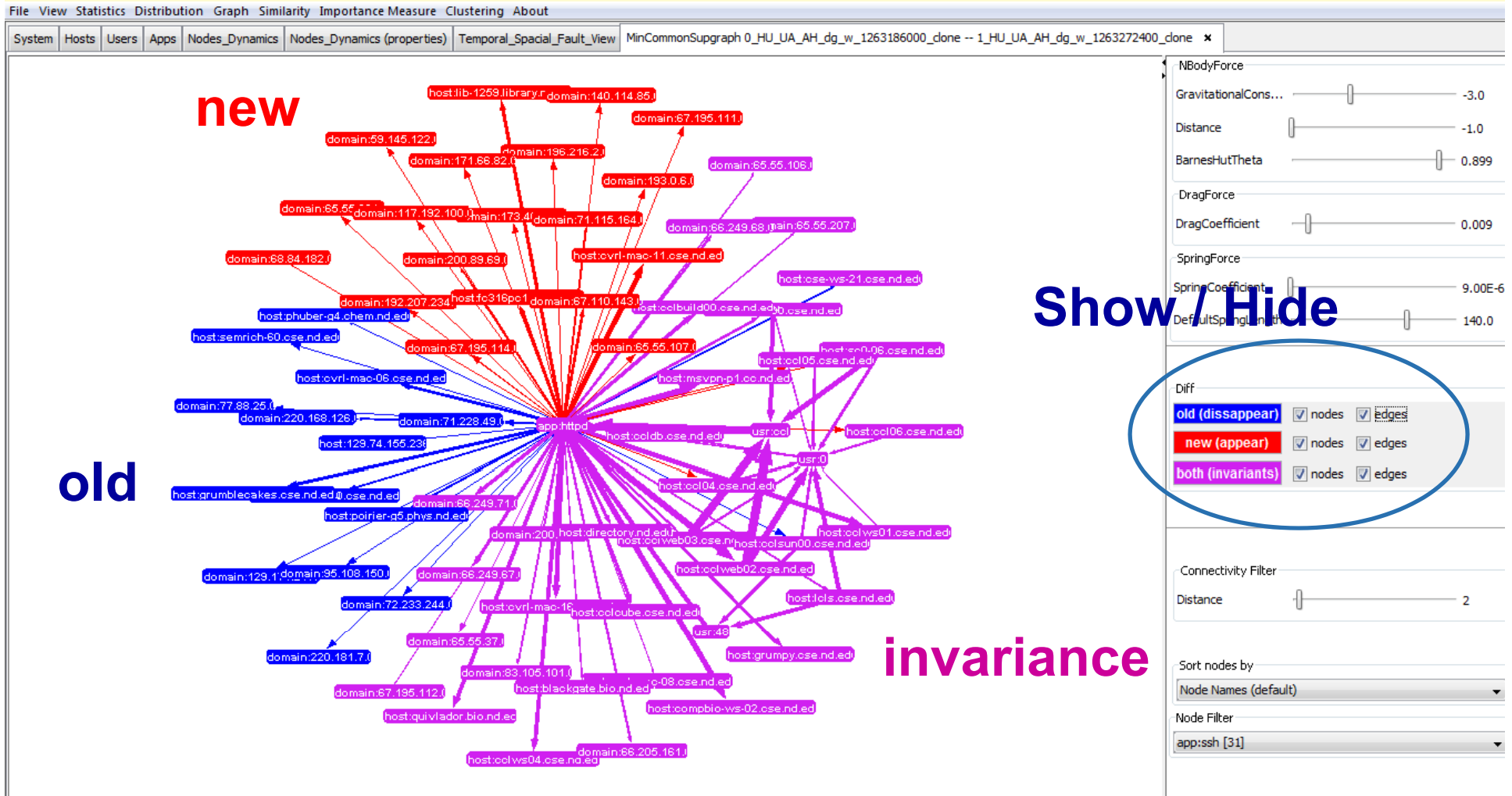
Similarity / Difference Visualization



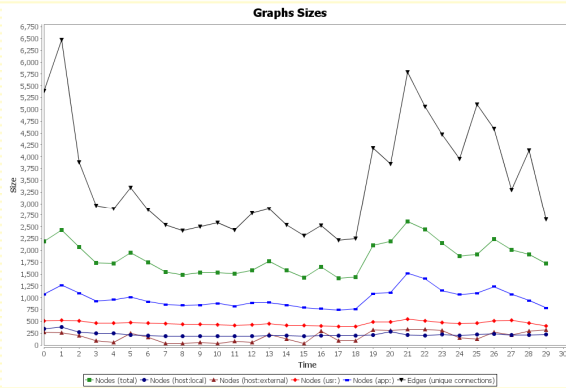
Variance vs. invariance



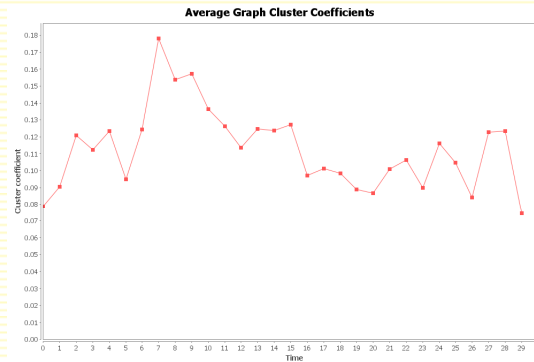
Differential view



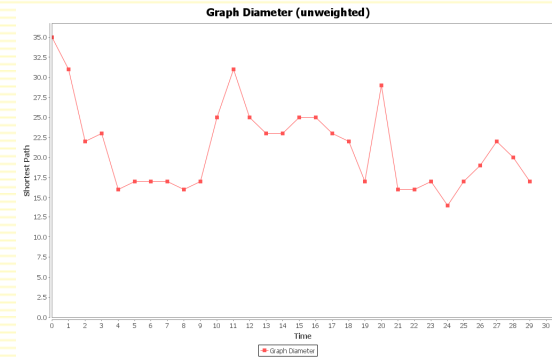
Visualizing graph property changes



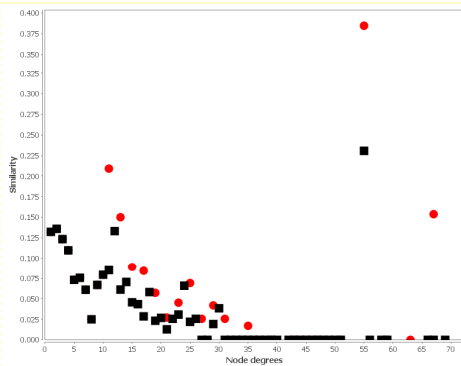
Graphs sizes



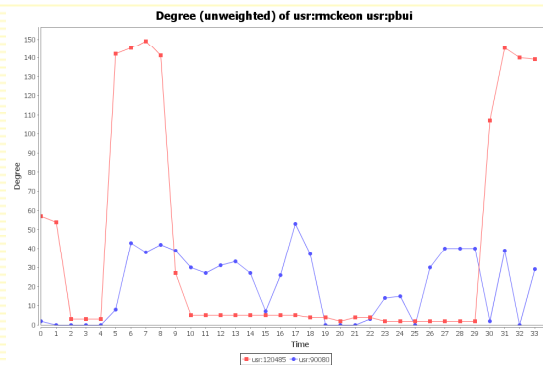
Cluster coefficients



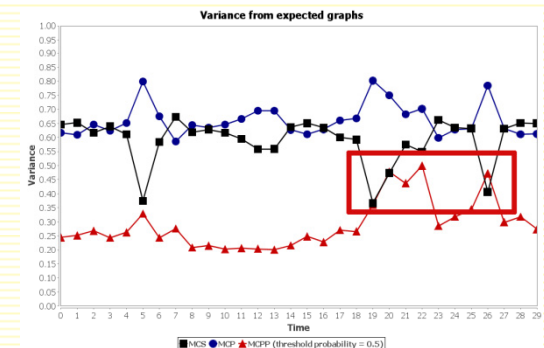
Graphs diameters



Degree distributions



Graphs distances



Graphs variance scores

Graph distance

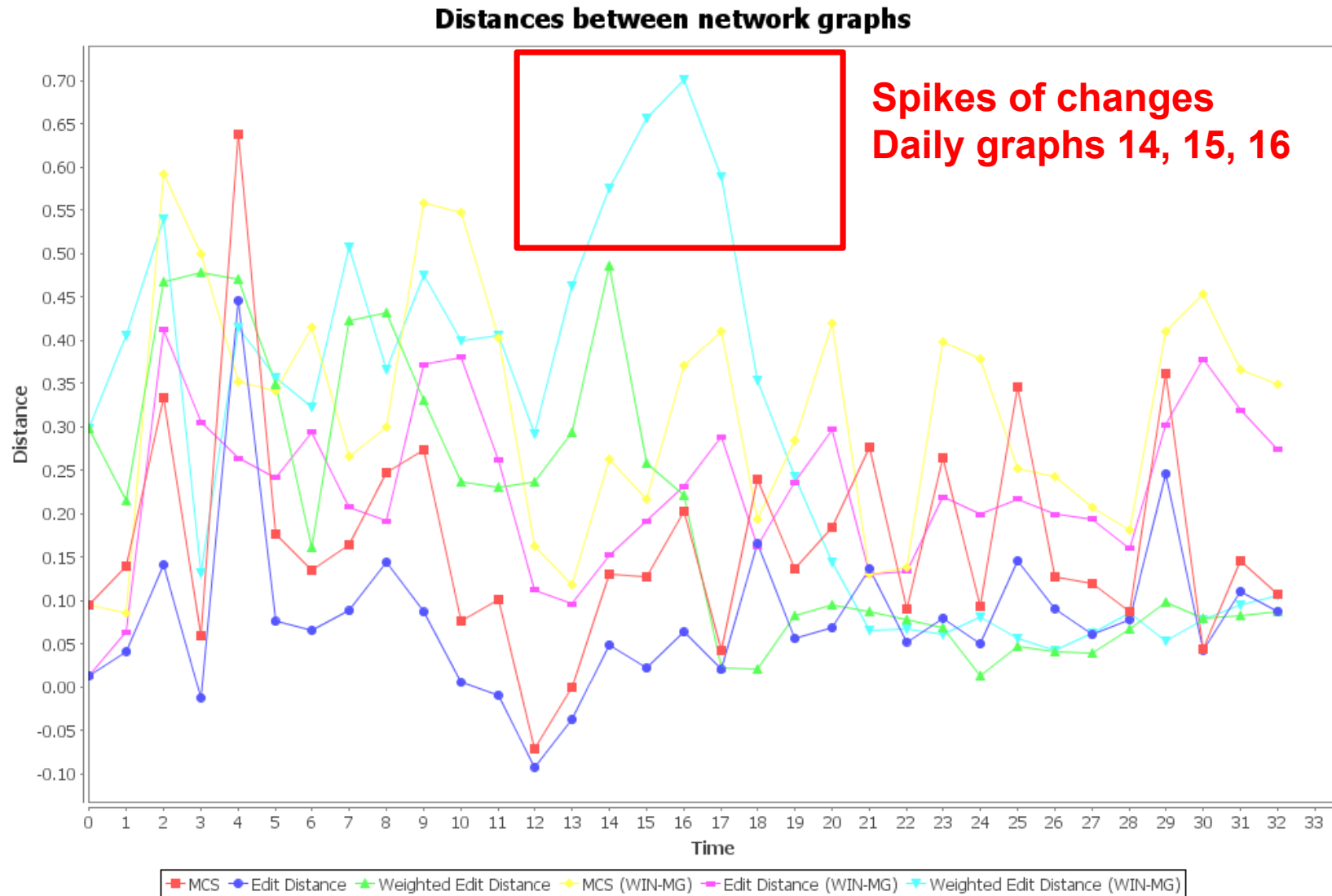
- Quantification through normalized distance functions:
- Schenker:2003, Bunke 1998, 2007.
- MCS based:

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

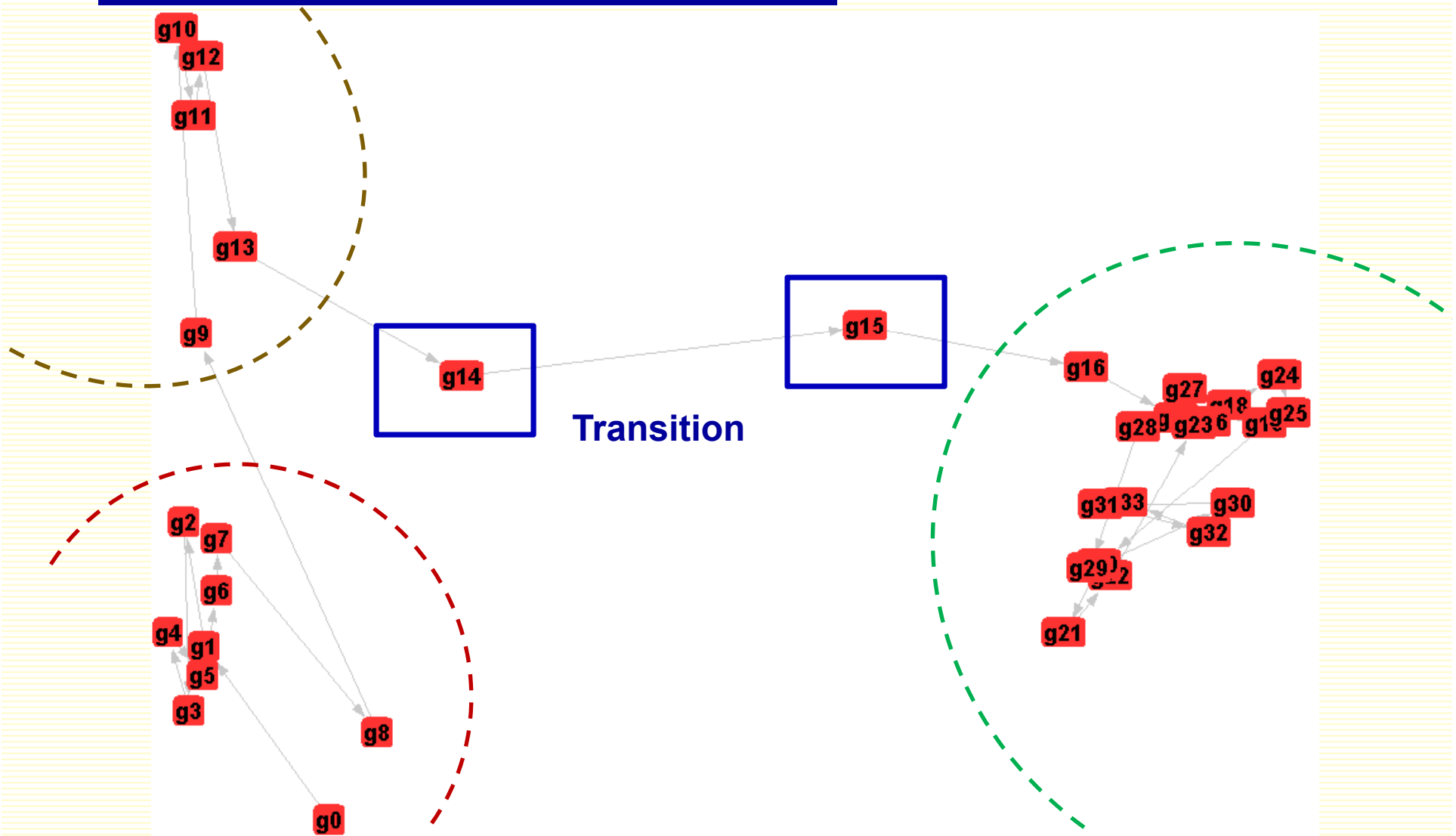
- Graph edit distance (GED) based:

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

Graph distance



MDS view (cluster evolution)



Top network components responsible

(12) graph component	distance	(%)
usr:108172--app:java	32,283	12.3%
usr:90080--app:parrot	11,985	4.57%
host:129.74.20.134--usr:108172	11,565	4.41%
app:parrot--host:129.74.152.44	10,788	4.11%
usr:0--app:python	9,414	3.59%

(13) graph component	distance	(%)
usr:108172--app:condor_starter	25,730	5.33%
usr:90080--app:parrot	21,734	4.5%
app:parrot--host:129.74.152.44	15,592	3.23%
host:129.74.152.44--usr:27	15,589	3.23%
usr:27--app:mysql	15,589	3.23%

(14) graph component	distance	(%)
usr:0--app:sge_commd	67,004	8.34%
usr:1025--app:sge_execd	63,774	7.94%
usr:108172--app:condor_starter	35,159	4.38%
usr:108172--app:java	23,773	2.96%
usr:90080--app:parrot	20,813	2.59%

(15) graph component	distance	(%)
usr:1025--app:sge_execd	82,218	12.31%
usr:0--app:sge_commd	81,906	12.26%
usr:108172--app:condor_starter	15,777	2.36%
usr:108041--app:chirp	10,462	1.57%
usr:108172--app:java	9,883	1.48%

OK

Day 14



Day 15



Dynamic interactive and exploration

The screenshot displays the ENAVis software interface. The main window shows a network graph with nodes representing users, hosts, and applications. A blue callout box highlights a "2-hop view from an investigated user node". The right side of the interface features several control panels for adjusting simulation parameters:

- NBodyForce**: GravitationalConstant (-3.0), Distance (-1.0), BarnesHutTheta (0.899)
- DragForce**: DragCoefficient (0.009)
- SpringForce**: SpringCoefficient (9.00E-6), DefaultSpringLength (140.0)
- Connectivity Filter**: Distance (2)

A list of connections is shown below the filter, with one entry circled in green:

- app:condor_starter -- host:cclweb03.cse.nd.edu [10739.0]
- usr:1025 -- app:sge_execd [10759.0]
- usr:0 -- app:sge_commd [12812.0]
- usr:condor -- app:condor_startd [13880.0]
- app:java -- host:sc0-hn.cse.nd.edu [15062.0]
- usr:ccl -- app:chirp [15259.0]
- host:sc0-hn.cse.nd.edu -- usr:condor [16873.0]
- usr:pbui -- app:parrot [20816.0]
- usr:ccl -- app:chirp [15259.0]

Queries for degree trend

app:sge_qmaster

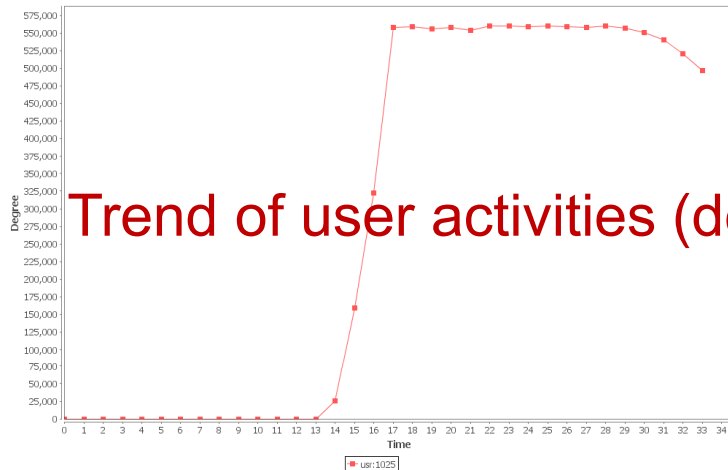
one-hop view from an investigated user node

host:iss-node023.cse.nd.edu

usr:1025

host:head.cse.nd.edu

Degree (weighted) of usr:1025



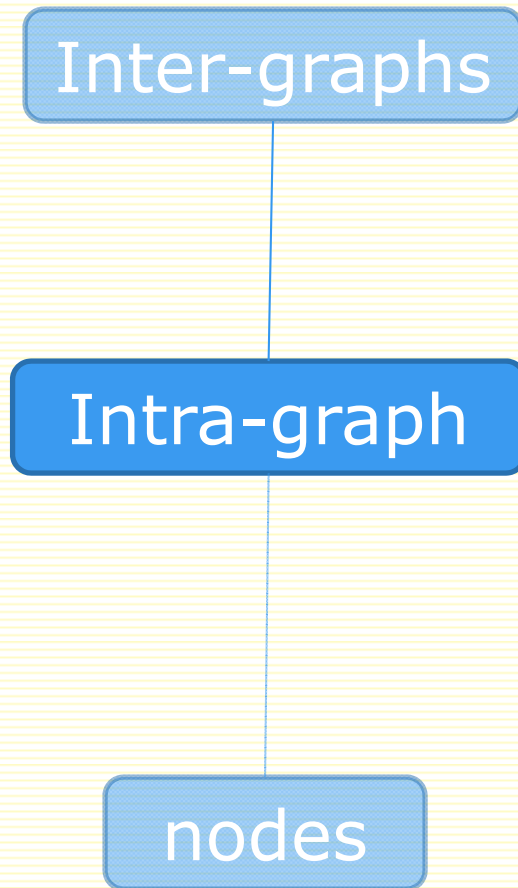
Trend of user activities (degrees)

- host
- User
- app
- domain
- Scores
- trend
- degree

app:sge_execd

host:iss-node024.cse.nd.edu

Hierarchical Similarity Visualization



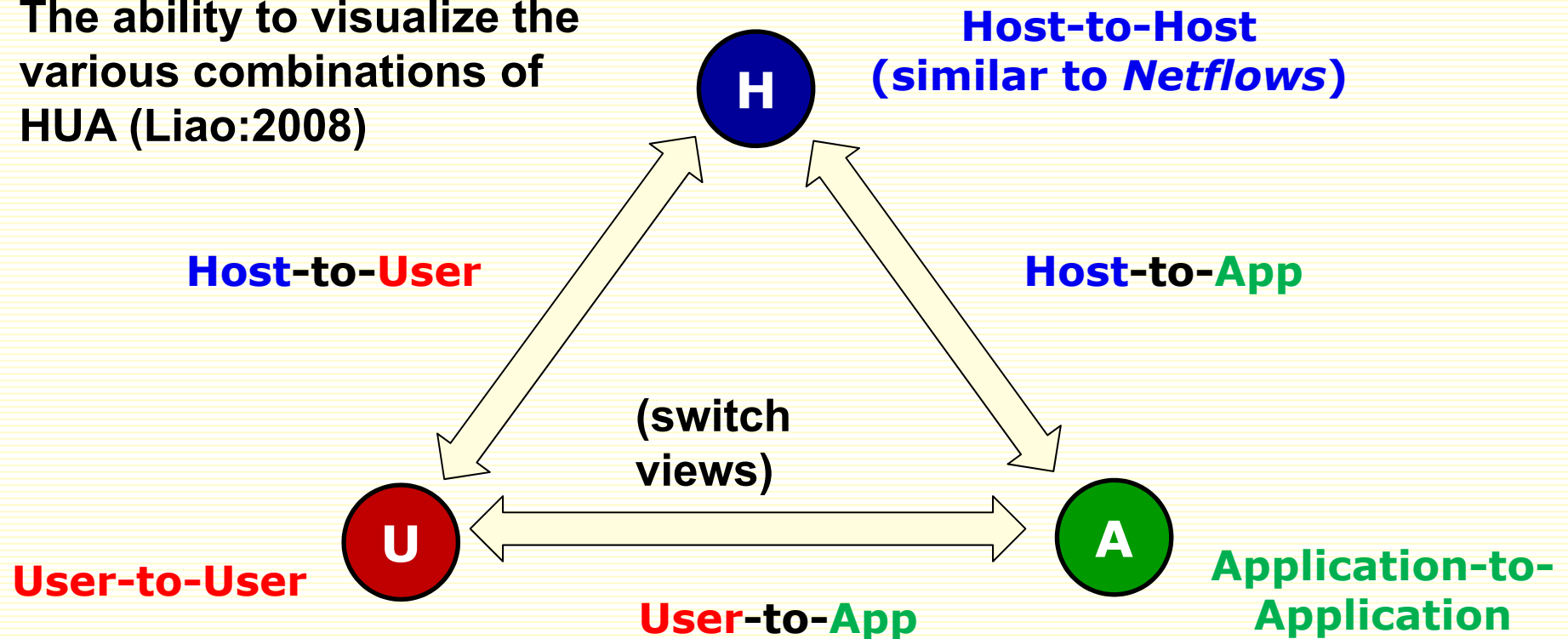
Intra-graph clustering visualization

- HUA connectivity graphs
- (Multi-)bipartite graphs
- Similarity graphs

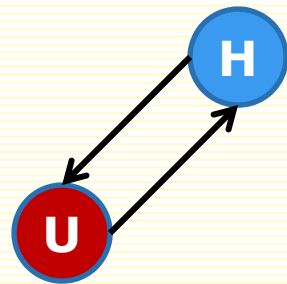
HUA Graph Model

- *Heterogeneous* graph
- 4D space
 - Hosts, Users, Applications (**HUA**), Time

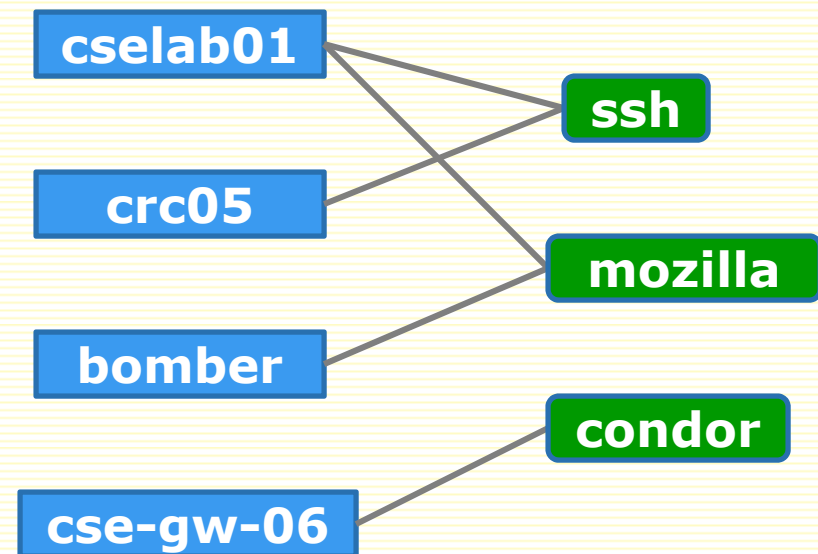
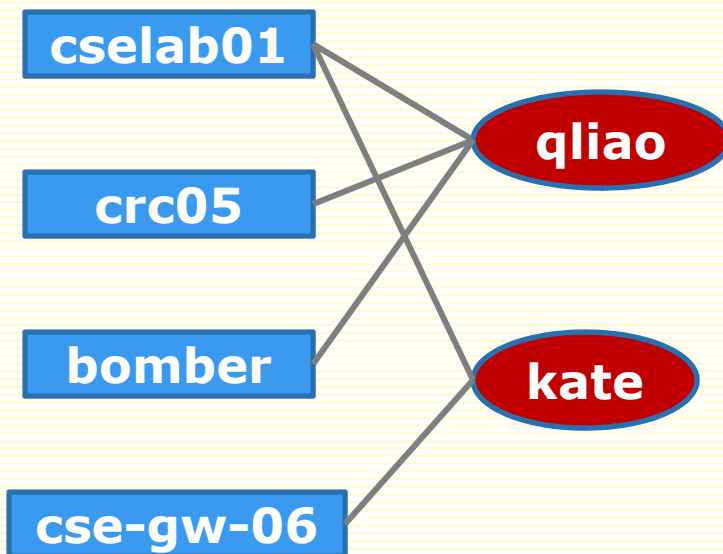
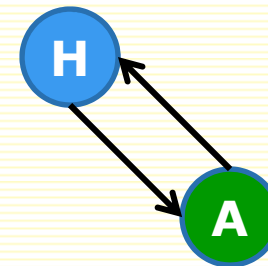
The ability to visualize the various combinations of HUA (Liao:2008)



Examples (HU, HA)

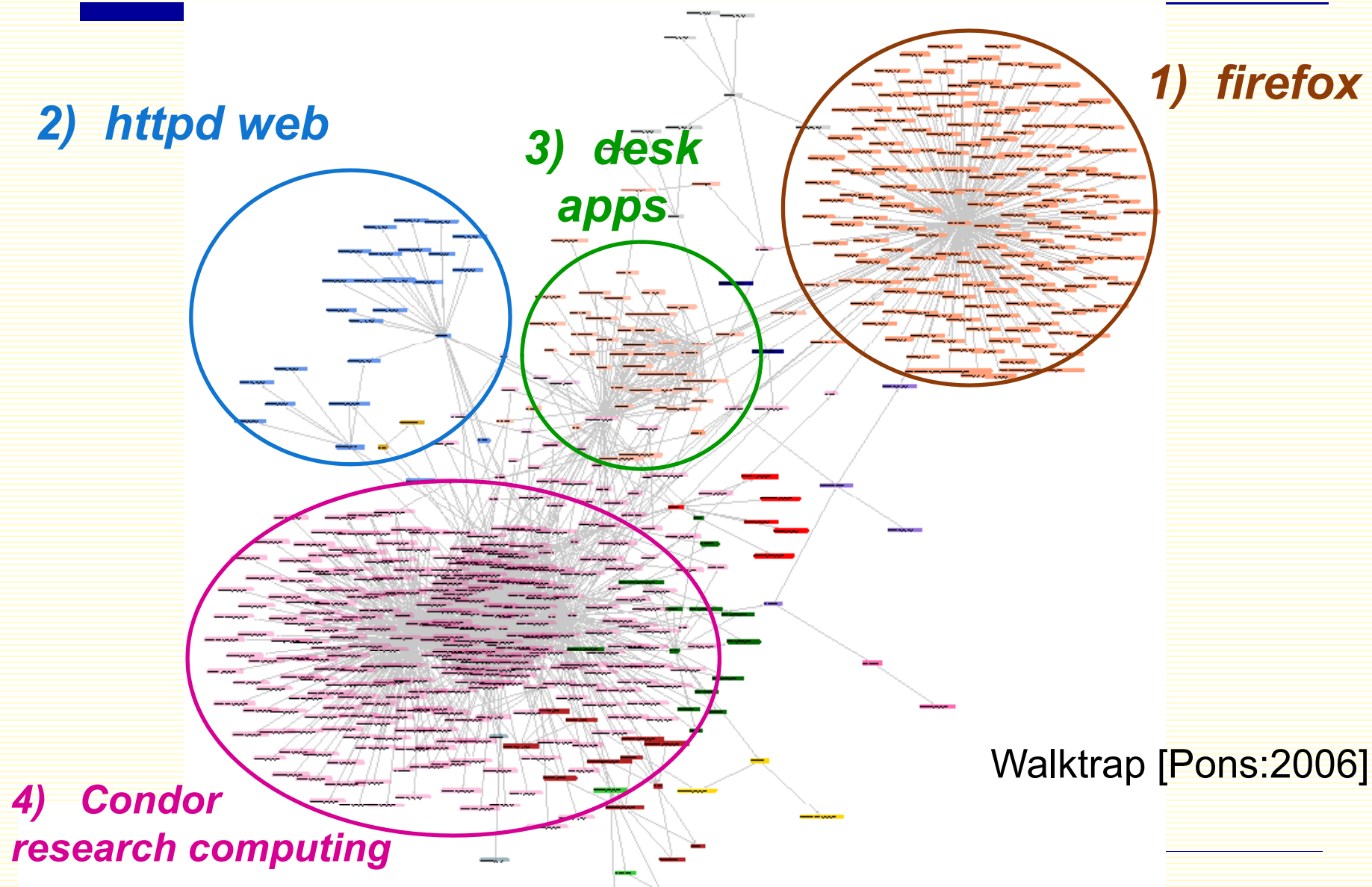


(Bipartite graphs)



Contribution: the ability to visualize the various combinations of HUA

Intra-graph clusters visualization



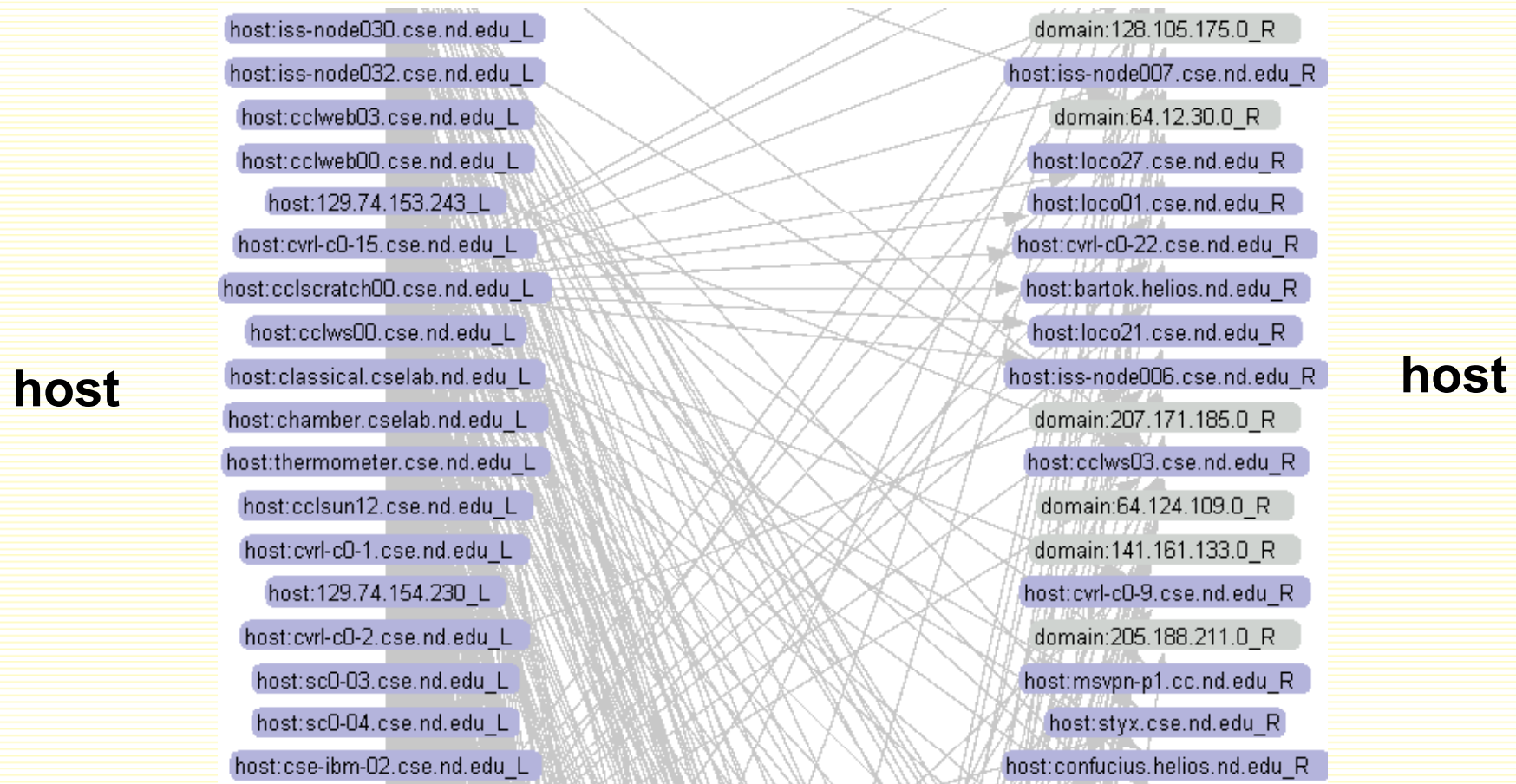
Cluster similarity visualization

- Visualizing the intra-graph clusters can provide:
 - Understanding of network usage **pattern**
 - Closely related **community** formed by similar *hosts*, *users*, *apps*.
 - **Insight** and potential **anomaly** analysis
- Quantify graphs changes through **cluster distance**
 - similar to Rand Index [Rand 1971]

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

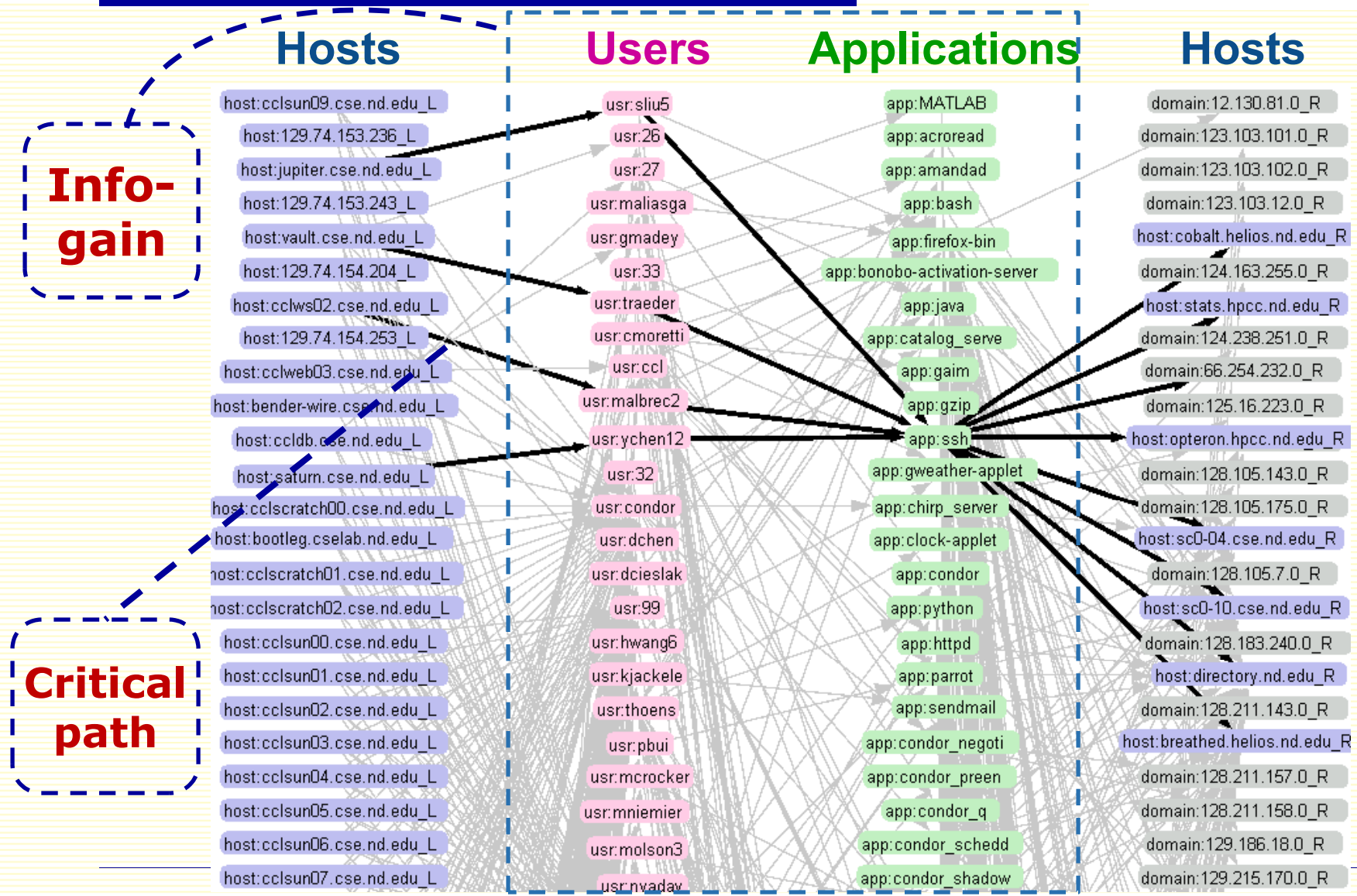
Bipartite graphs

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



Multi-bipartite graphs

Quadripartite
graph



Biclique communities

- Membership changes

□ *JDoe*

ONE



ONE &
TWO

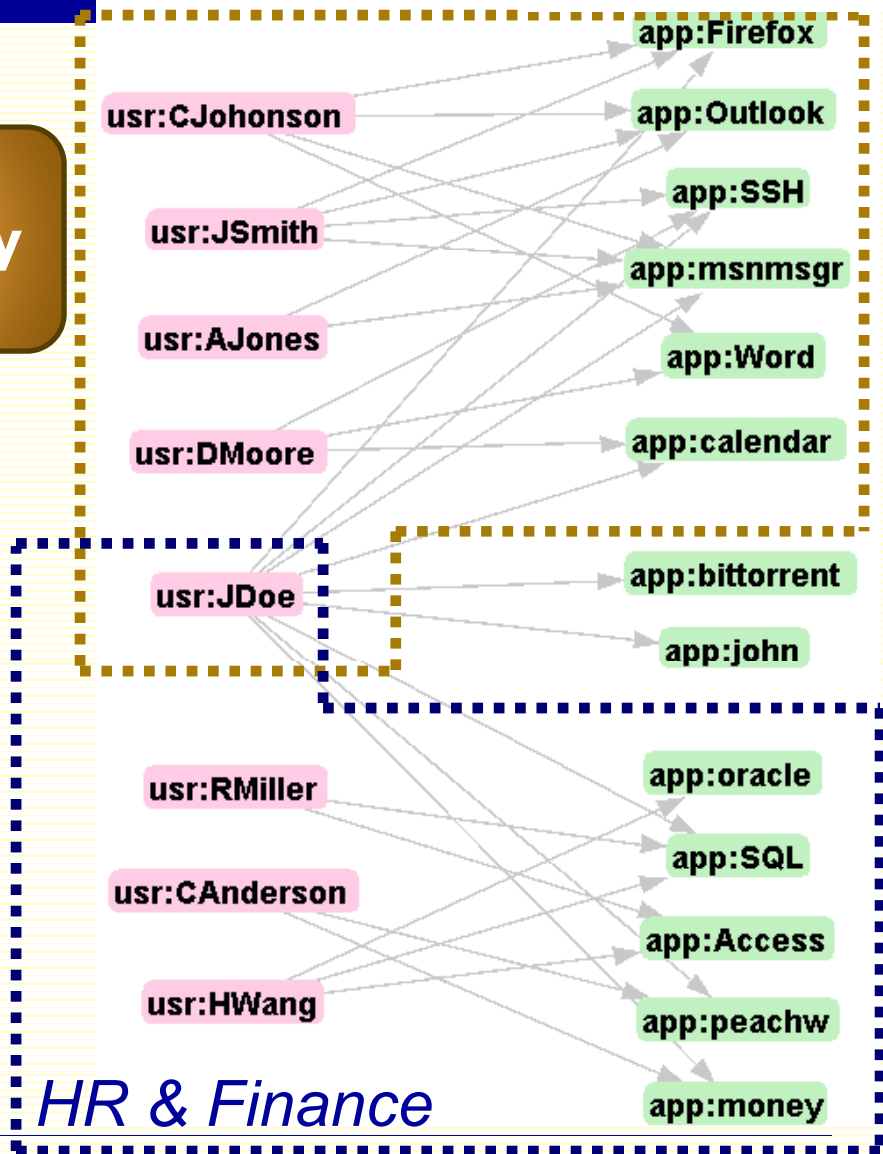
overlapping
communities



- Biclique communities [Lehmann:2008]

Users

Applications



Similarity graphs

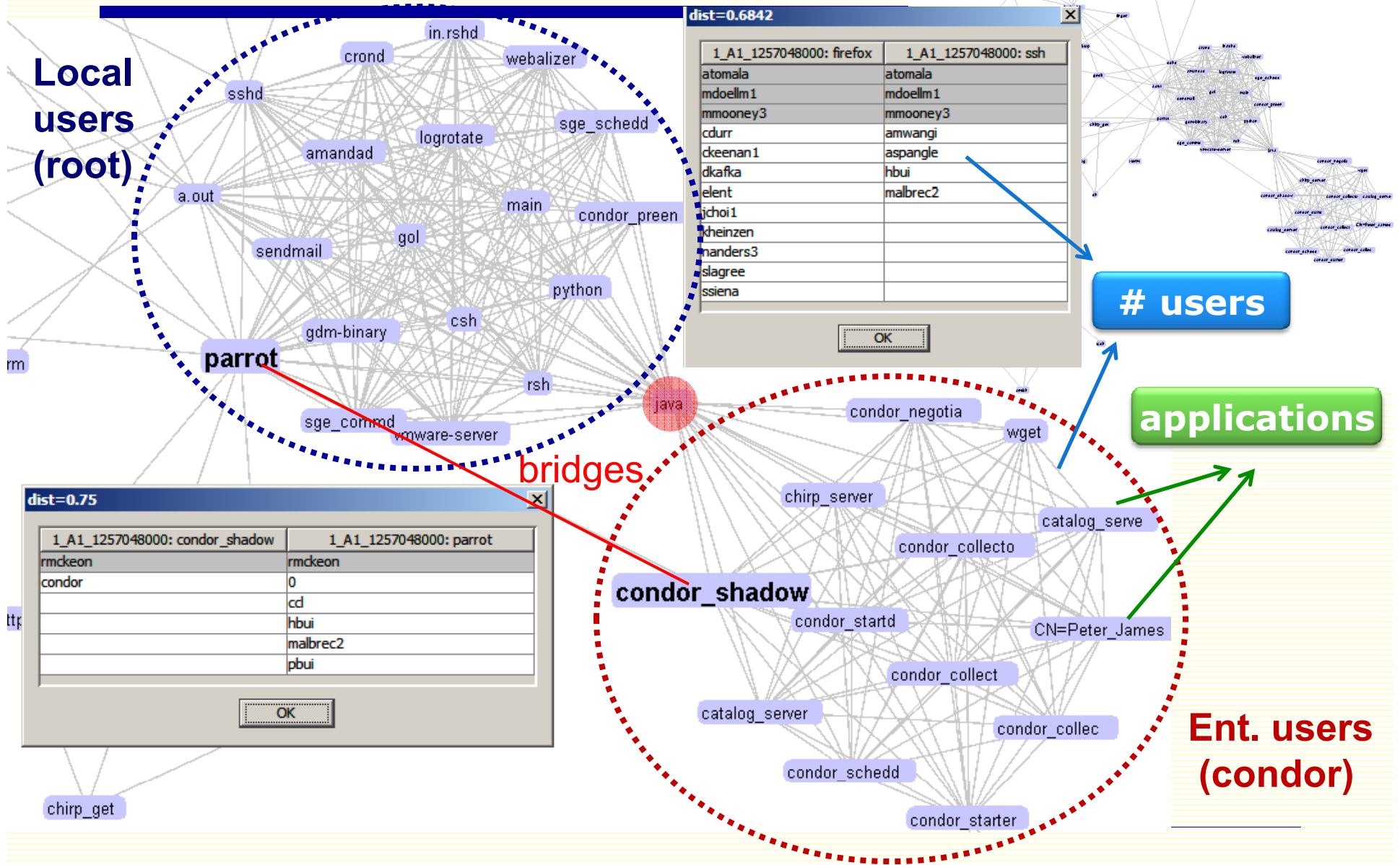
- Previous

- HUA heterogeneous graphs
- (multi-)bipartite graphs

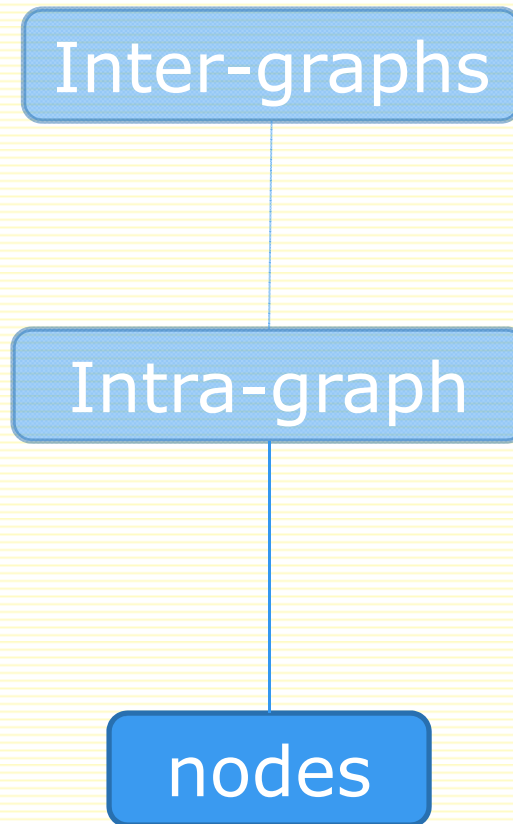
- ***Similarity graphs***

- Heterogeneity → Homogeneity
- Push ***similarity*** into the ***edge weights***
 - (Example)
 - nodes = users
 - edge weights = number of applications they share.

Similarity Graphs

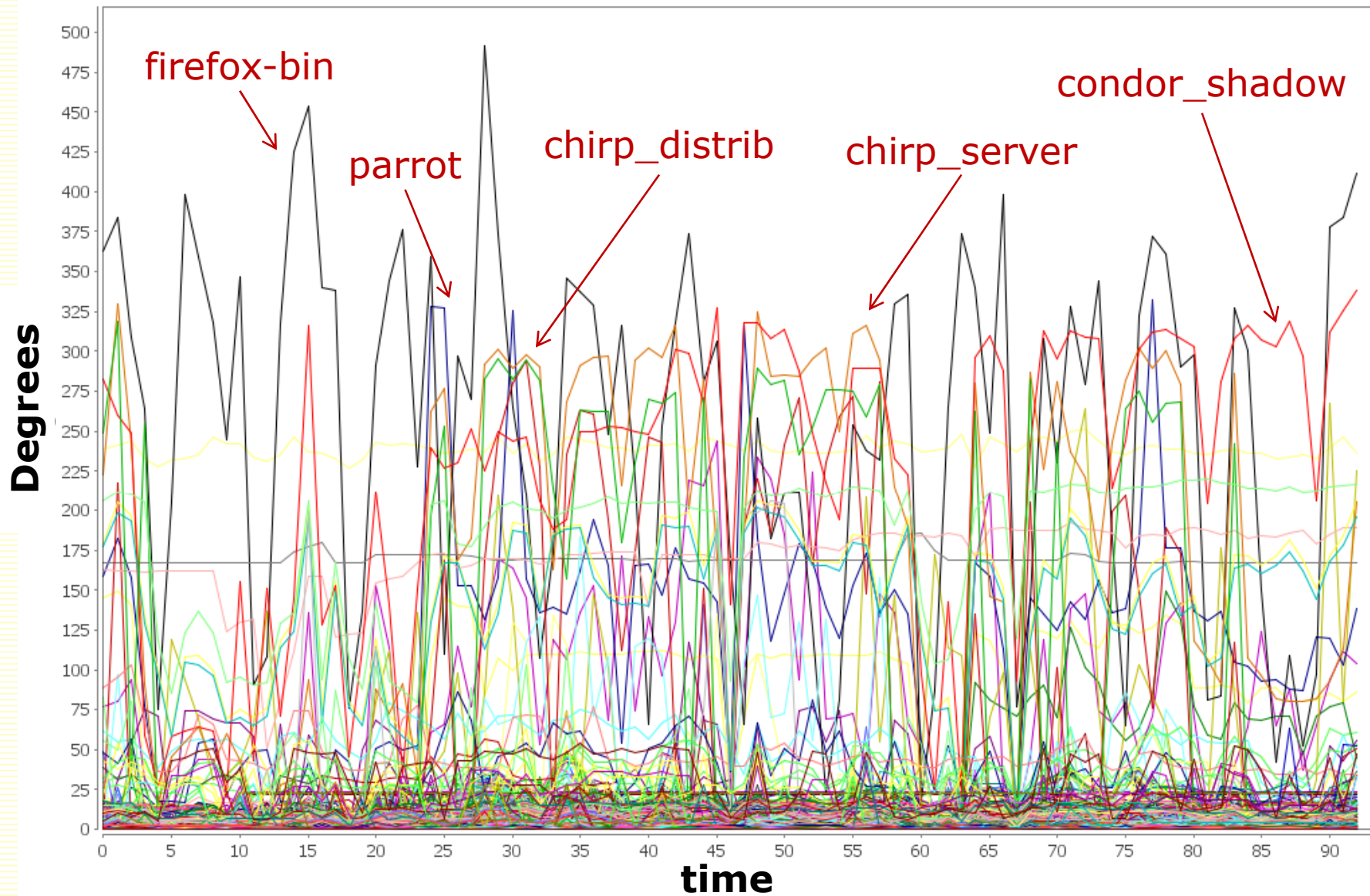


Hierarchical Similarity Visualization



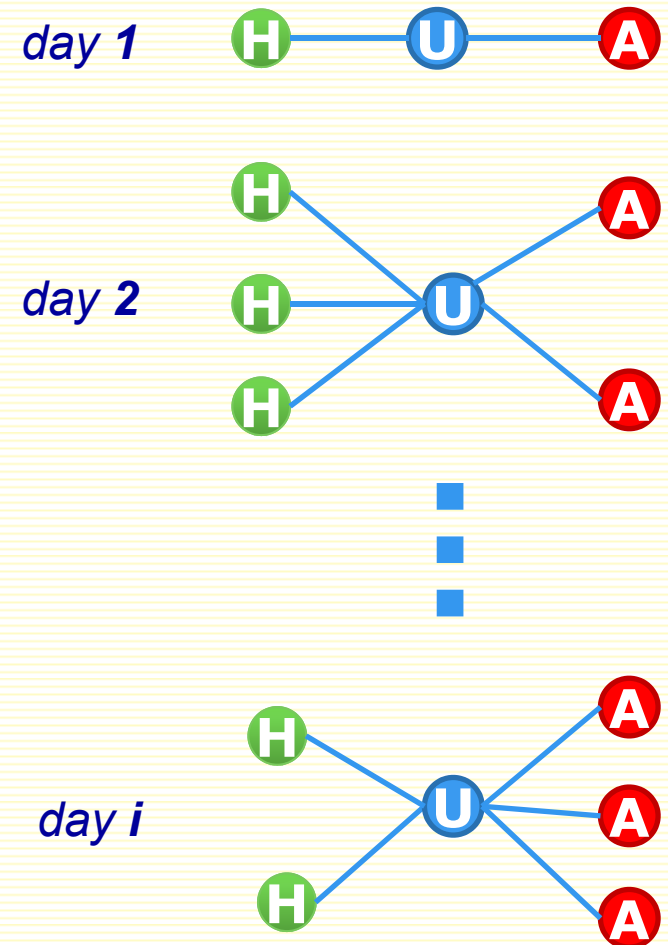
- Dynamics / similarity of each individual nodes
- Visualization of neighborhood changes over time
 - Quick visual analysis on node's anomalous behaviors

Dynamics of Node Degrees

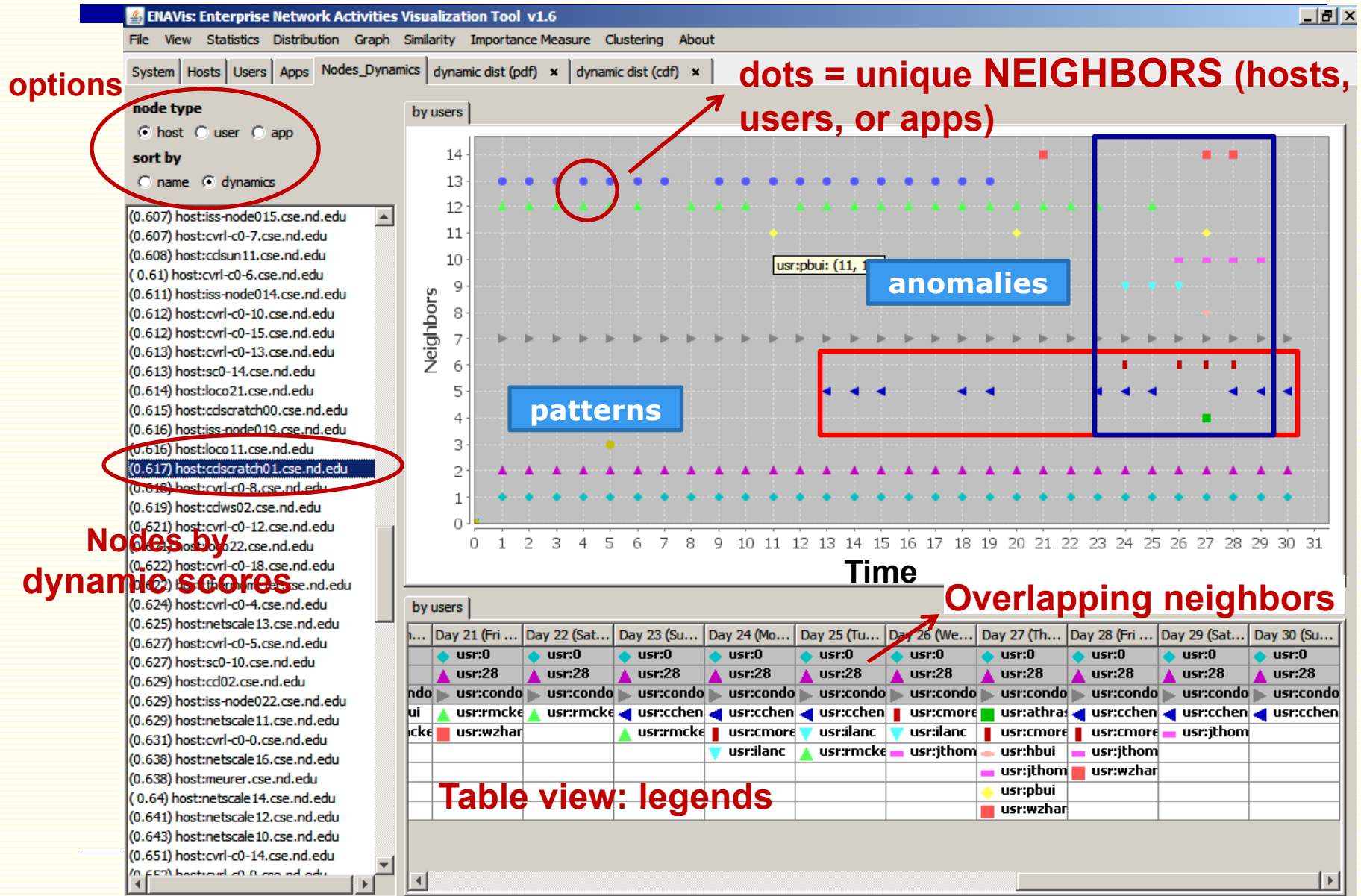


Node Dynamics/Similarity Visualization

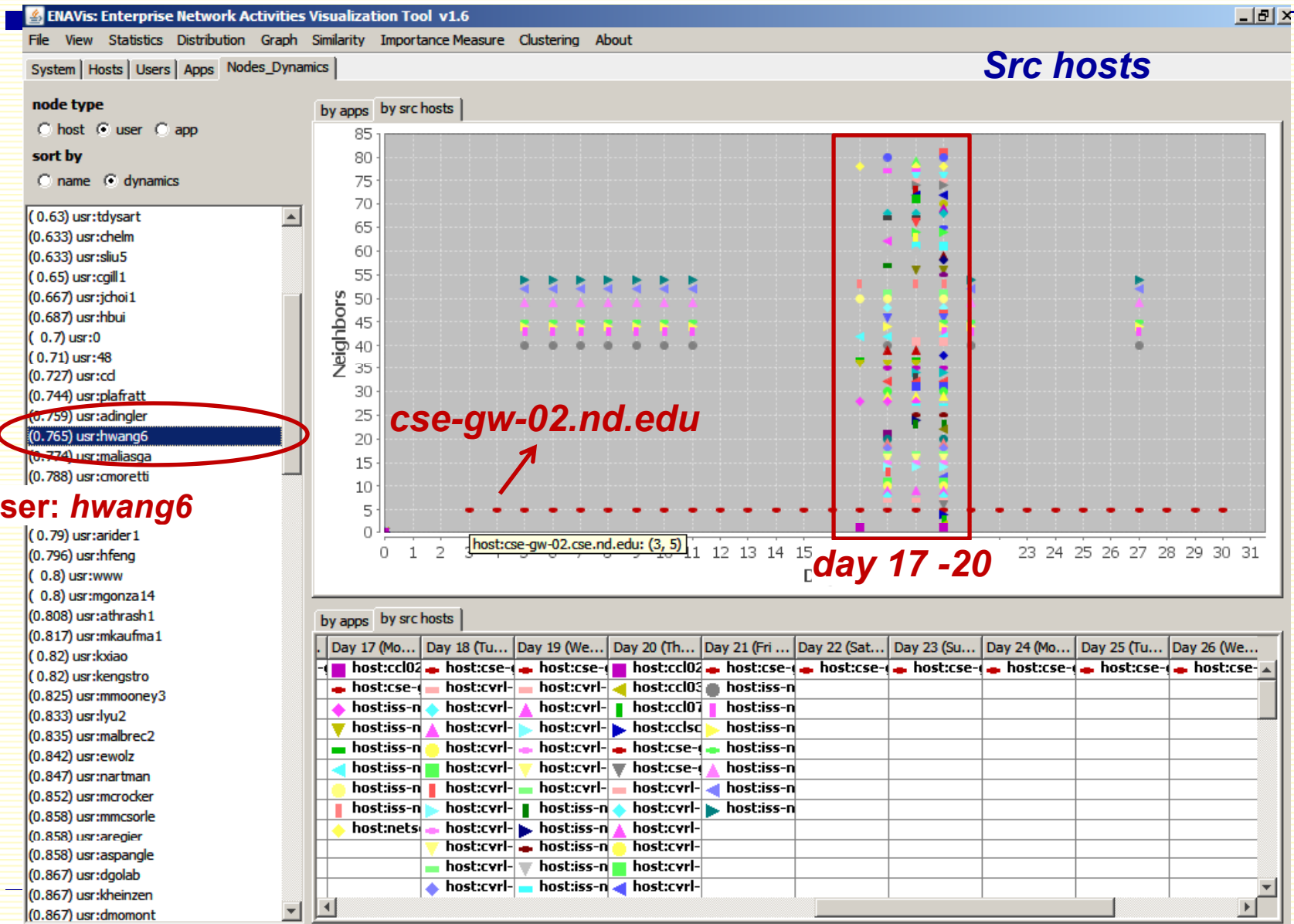
- Neighborhood changes
- Hosts:
 - users
- Users:
 - hosts
 - applications
- Applications:
 - users
 - src/dst hosts
- Quick and easy visualization
 - 2D scatter plots



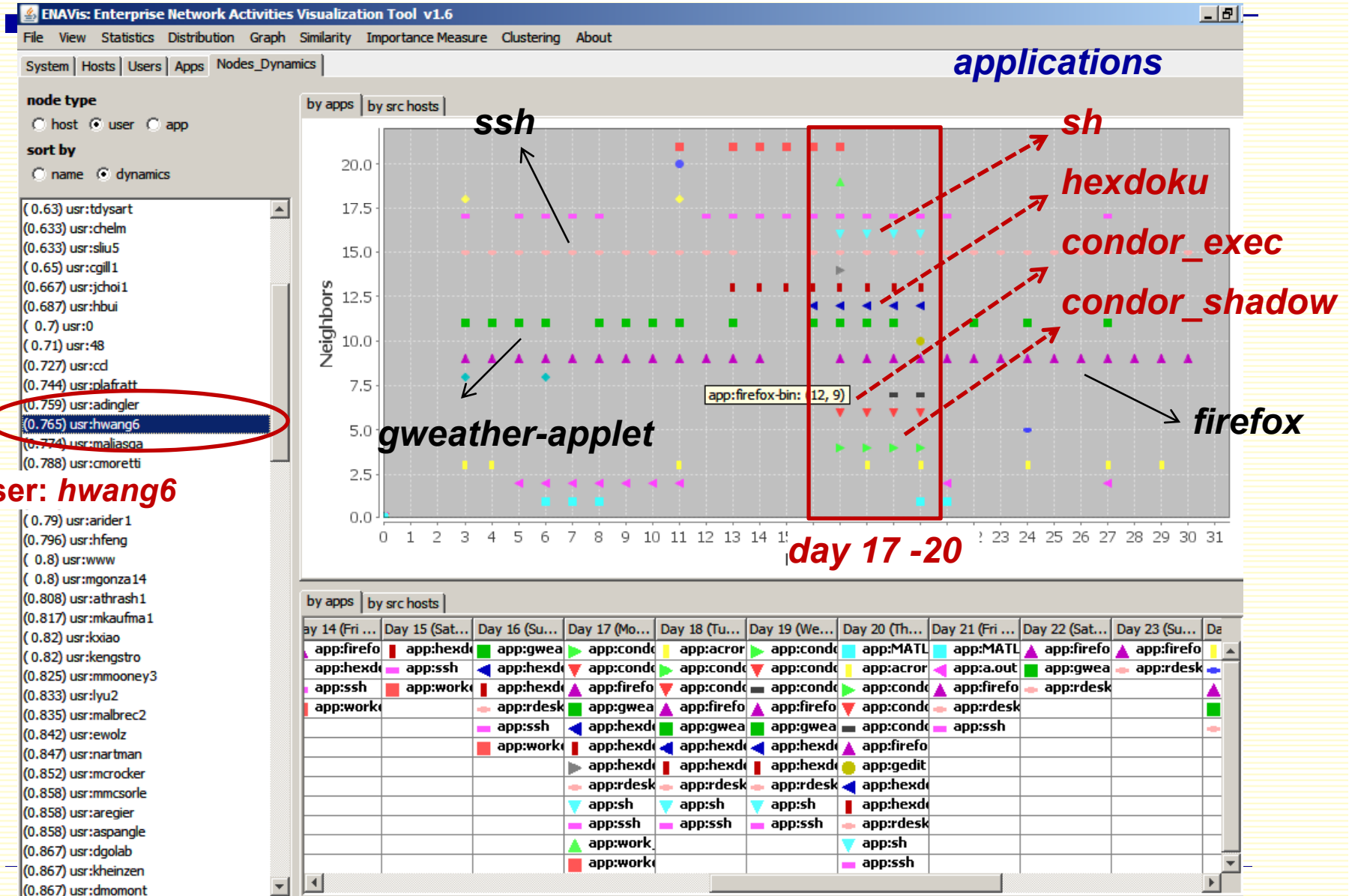
Node Similarity Visualization



Node Similarity Visualization



Node Similarity Visualization



Conclusion

- Visualizing *dynamic relationships* among **hosts**, **users**, and **applications** vs. traditional IP/port-based *Netflow* monitoring.
- Importance and challenges of **similarity / differences** of network graphs
 - Security / forensics / policy audit
 - Network management, troubleshoot
 - Anomaly analysis
- **Similarity visualization** : a promising approach.
- Novel transformation of graphs
 - HUA connectivity graphs, MDS graphs, (multi-)bipartite graphs, similarity graphs
- Hierarchical similarity visualization framework
 - Inter-graphs, intra-graphs, nodes
- More info available at <http://netscale.cse.nd.edu/Lockdown>