Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visualizing Host Traffic through Graphs

Eduard Glatz
Computer Engineering and Networks Laboratory
ETH Zurich (Switzerland)
eglatz@tik.ee.ethz.ch

VizSec 2010 Sept. 14, 2010

Motivation

- Research in behavioural host profiling and cyber security
  - How can we build easily interpretable host profiles and evolve them?
  - Is this IP address a server or a client?
  - What services is this IP address providing?
  - Why does a host experience one-way flows?

- Teaching/Training
  - How do Berkeley sockets work?
  - What activity does a complex communication pattern represent?

How to represent Host Traffic?

Idea: *use graphs*

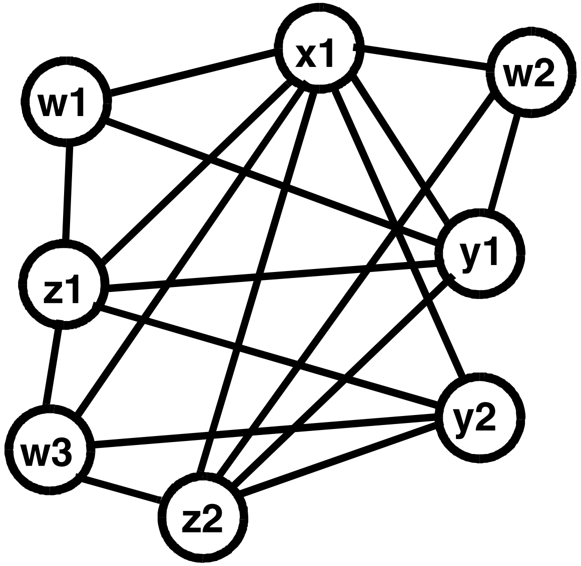
- Nodes correspond to flow attributes
- Links show flow attributes that appear together
- Result: *very dense/noisy graph*

Problem:

- Which relationships are most interesting to illustrate?

Transactions:

w1, x1, y1, z1
w2, x1, y1, z2
w3, x1, y2, z1
w3, x1, y2, z2



Example mapping to flow records:

w: source IP

x: destination IP

y: source port

z: destination port

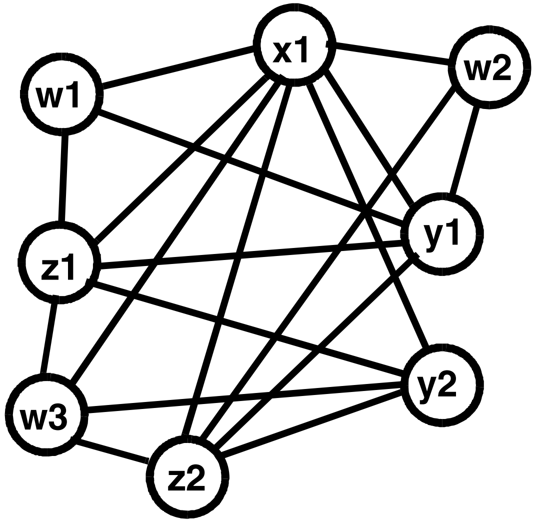
Transaction Visualization by k-Partite Graphs

Approach:

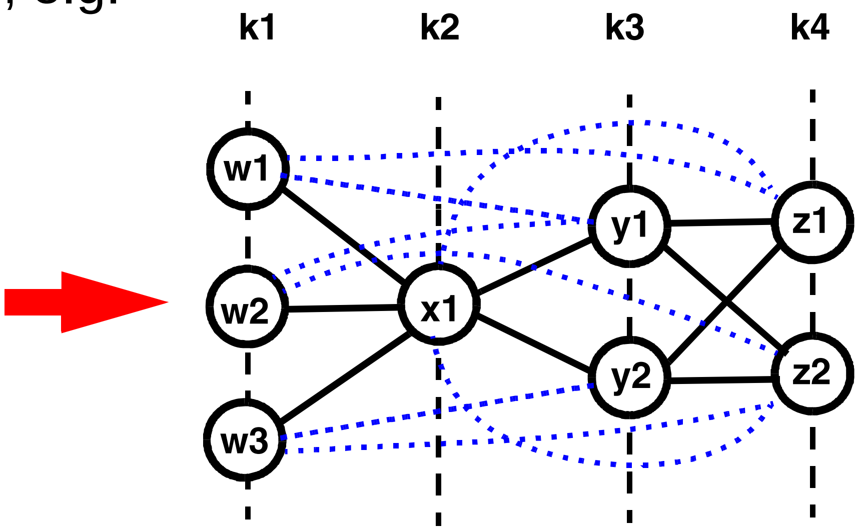
- K-partite graphs plus abstraction, e.g.

Transactions:

w1, x1, y1, z1
w2, x1, y1, z2
w3, x1, y2, z1
w3, x1, y2, z2



a)



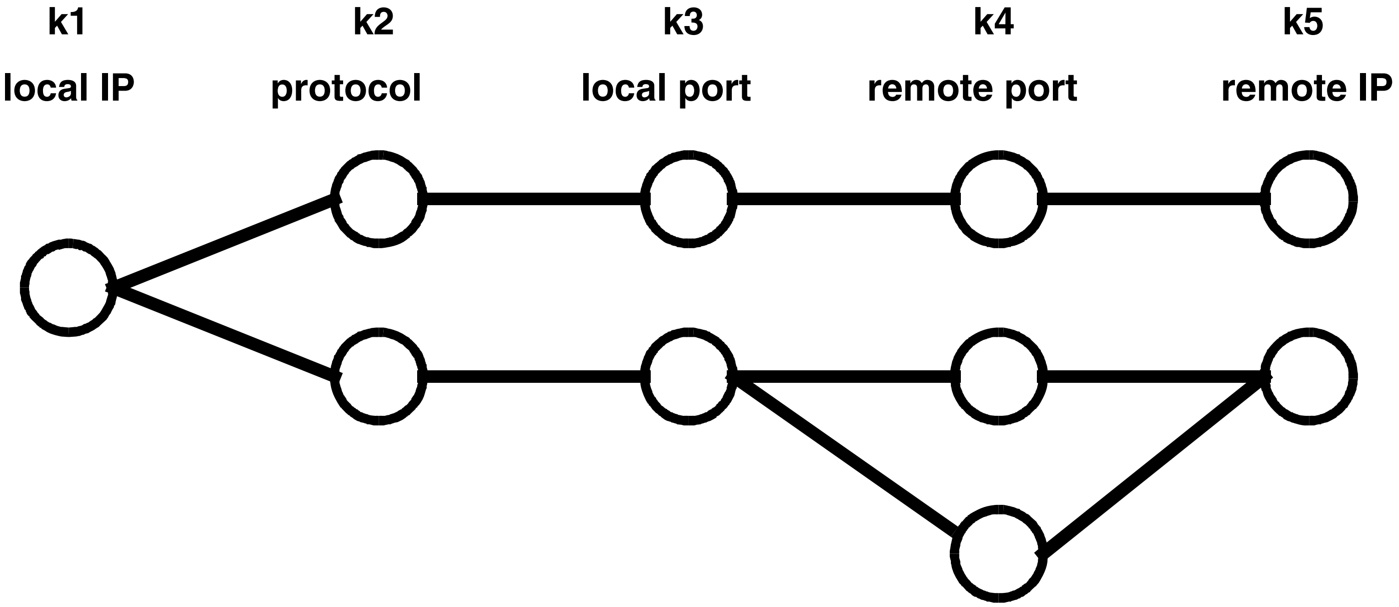
b)

Abstraction:

- Purge blue lines and re-arrange partitions as needed to keep links which are important to identify offered services and host roles

Host Application Profile (HAP) Graphlet

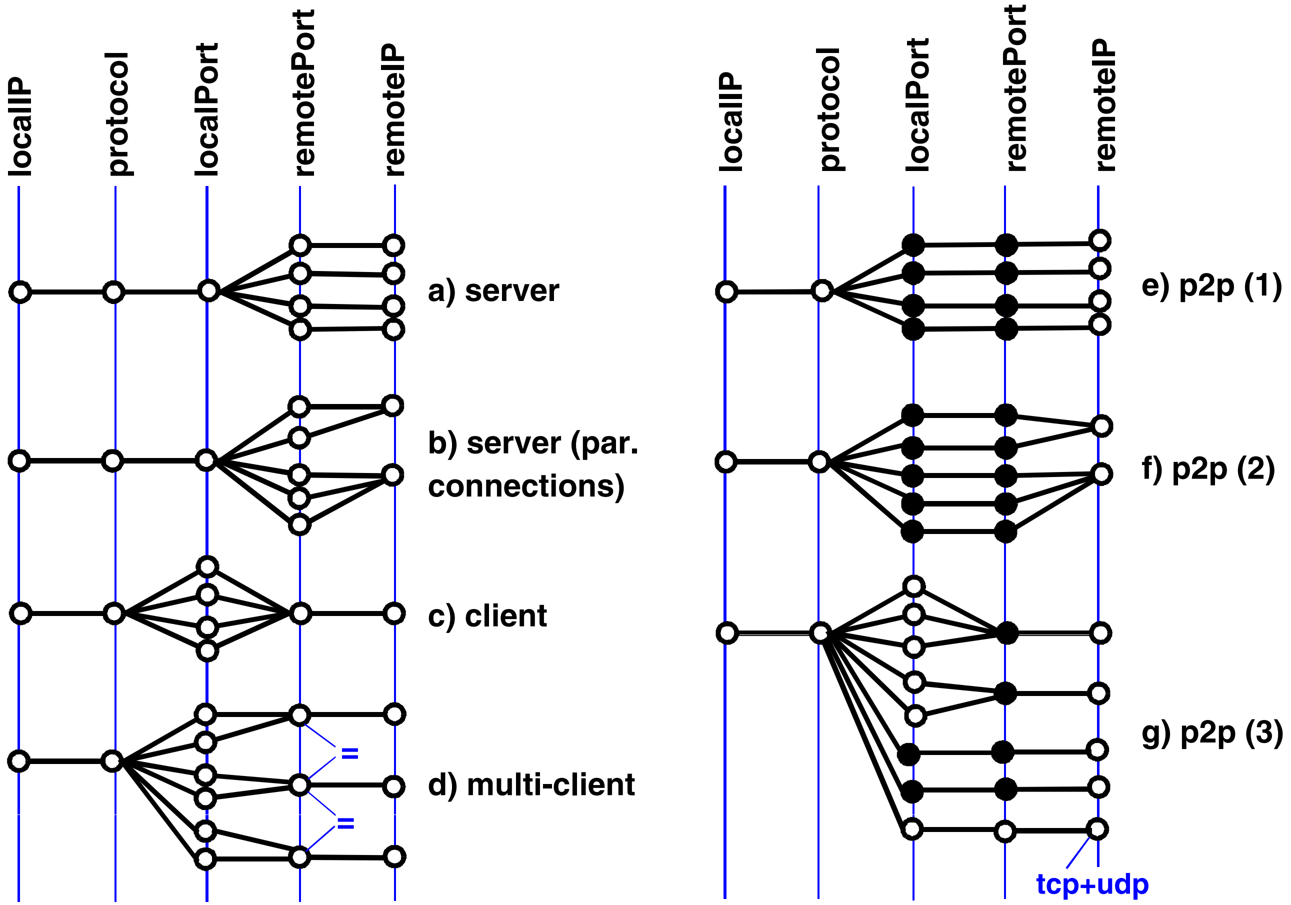
We propose: Host traffic visualization through a 5-partite graph



- Terminology: local/remote instead of source/destination
- Annotations on nodes and links (not shown in example)

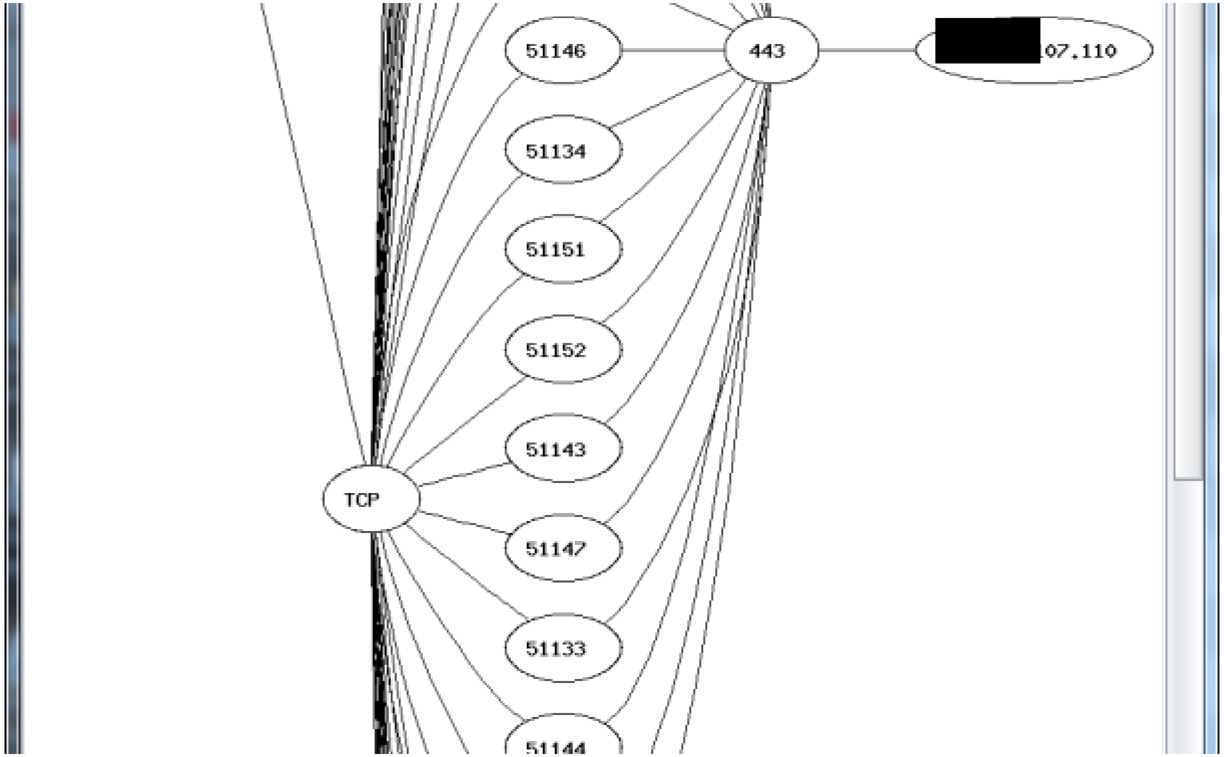
What Graph Structures can we expect?

Most prevalent host roles:



Need for Summarization

- Ideally, a HAP graphlet fits into available screen area
- But …



Host Role Summarization

Idea:

- Compress *per-role subgraphs*

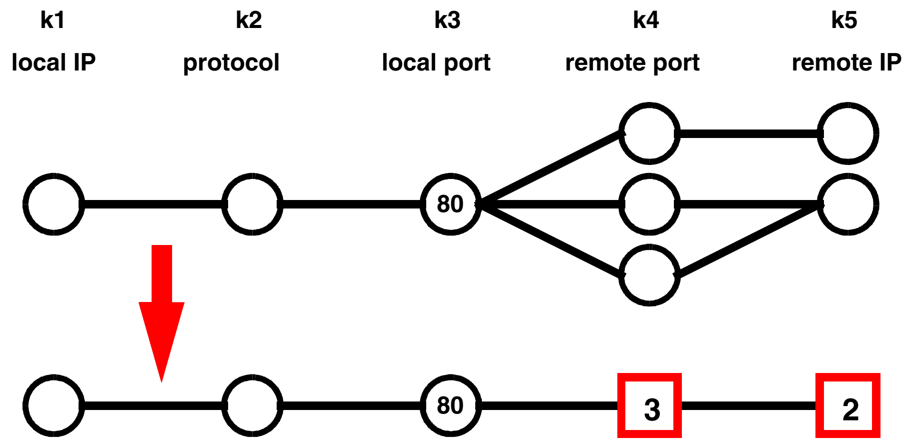
Prequisite:

- Roles can be associated with sub-graphs

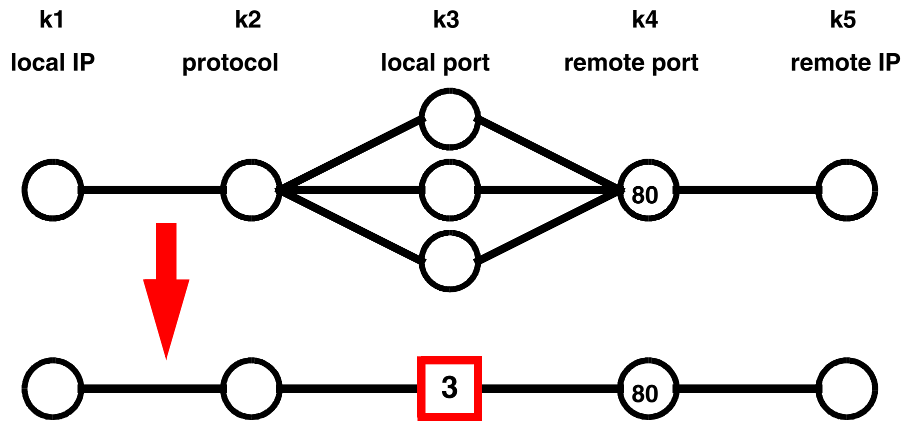
Methodology:

- Decompose graphlet into role-related subgraphs
- Replace role-related sub-graphs by summary sub-graphs
- ➢ Decomposition and replacement algorithm depends on role types (server/client/p2p roles)

Examples of Role Summarization



Server role



Client role

Flow Classification and Filtering

- Regular applications use bidirectional communication to acknowledge received data

- Done on transport layer (TCP) or application layer (UDP)

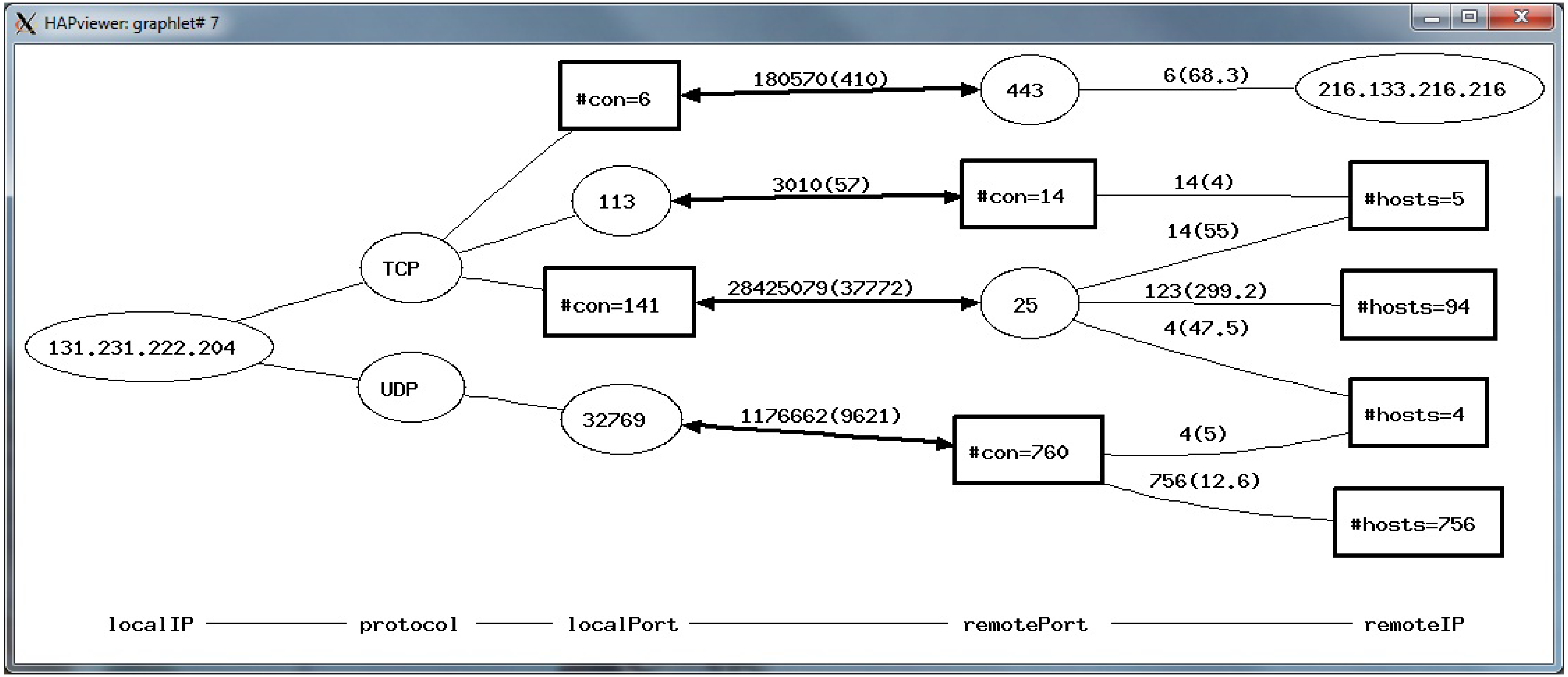
- Idea: differentiate one- and two-way flows

- Methodology:
  - Pair unidirectional flows in opposite direction that use identical endpoints
  - Look „over the fence“ (i. e. observation interval borders) when searching a buddy for a within-interval unidirectional flow

Role Summarization and Filtering

Example:

Real-world data: 1082 flows, 48722 packets, 107 one-way flows filtered



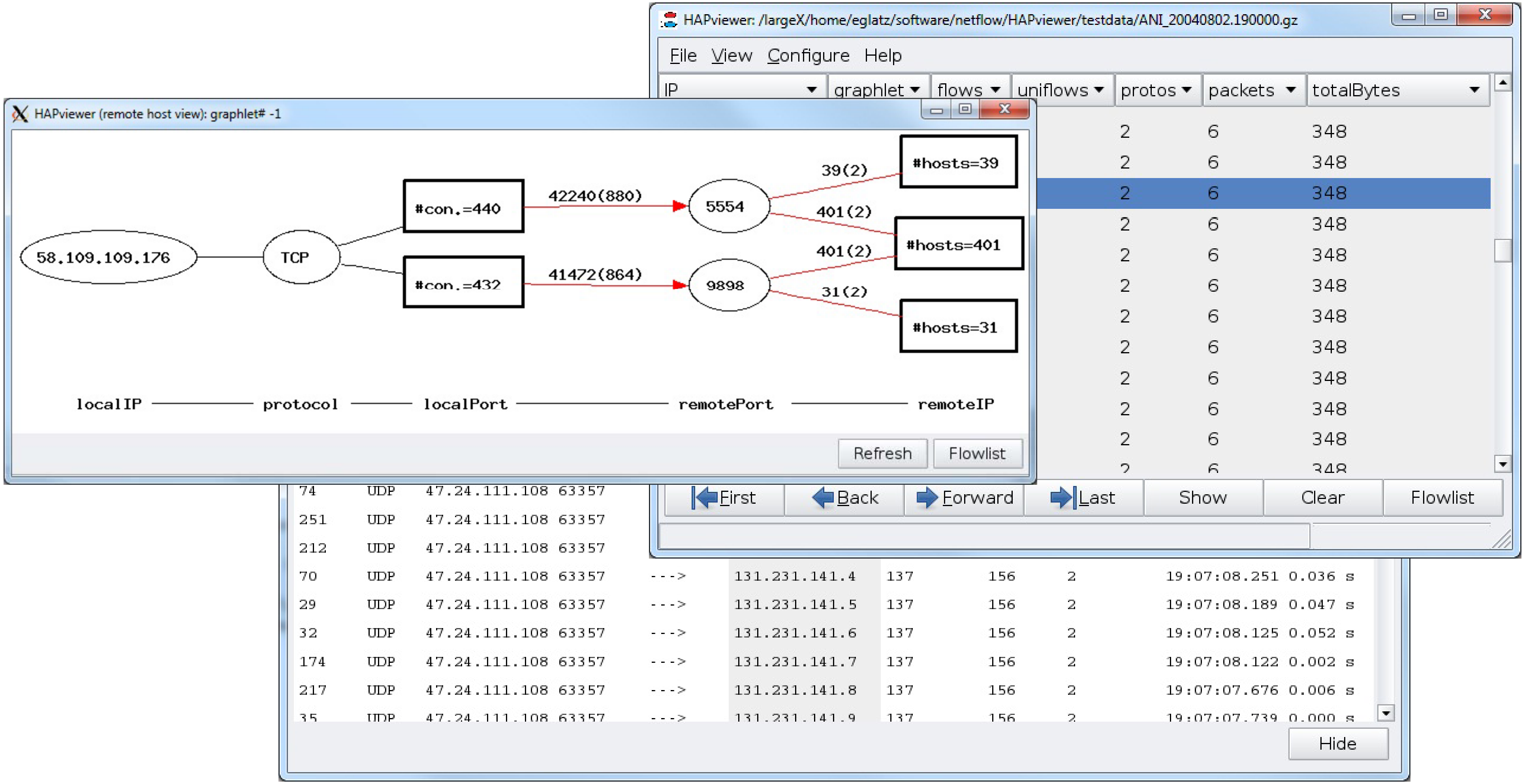
The Tool: HAPviewer

- Stand-alone Unix/Linux application with GUI
- Additionally, program library for integration into powerful network monitoring frameworks
- Typical use cases:
  - Qualitative studies of roles incorporated by hosts
  - Interpretation of complex connection structures
  - Identifying unknown service ports
  - Investigation of a host marked as suspicious by IDS/firewall alerts
  - Teaching of Berkeley socket model

Conclusions

- Graph-based host traffic visualization
  - Provides an analyst a quick and easy interpretable overview of host activities involving hundreds or thousands of flows
- Tool *HAPviewer*
  - Available as open source from http://hapviewer.sourceforge.net
  - Two versions: stand-alone GUI application and program library
- Outlook:
  - Integration into NfSen monitoring framework (project started)
  - Usability studies involving security professionals
  - Correlation of security alerts with host profile changes

Questions?



```
protos  packets  totalBytes
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
2       6        348
```

```
74    UDP   47.24.111.108 63357
251   UDP   47.24.111.108 63357
212   UDP   47.24.111.108 63357
70    UDP   47.24.111.108 63357   --->   131.231.141.4   137   156   2   19:07:08.251 0.036 s
29    UDP   47.24.111.108 63357   --->   131.231.141.5   137   156   2   19:07:08.189 0.047 s
32    UDP   47.24.111.108 63357   --->   131.231.141.6   137   156   2   19:07:08.125 0.052 s
174   UDP   47.24.111.108 63357   --->   131.231.141.7   137   156   2   19:07:08.122 0.002 s
217   UDP   47.24.111.108 63357   --->   131.231.141.8   137   156   2   19:07:07.676 0.006 s
35    UDP   47.24.111.108 63357   --->   131.231.141.9   137   156   2   19:07:07.739 0.000 s
```