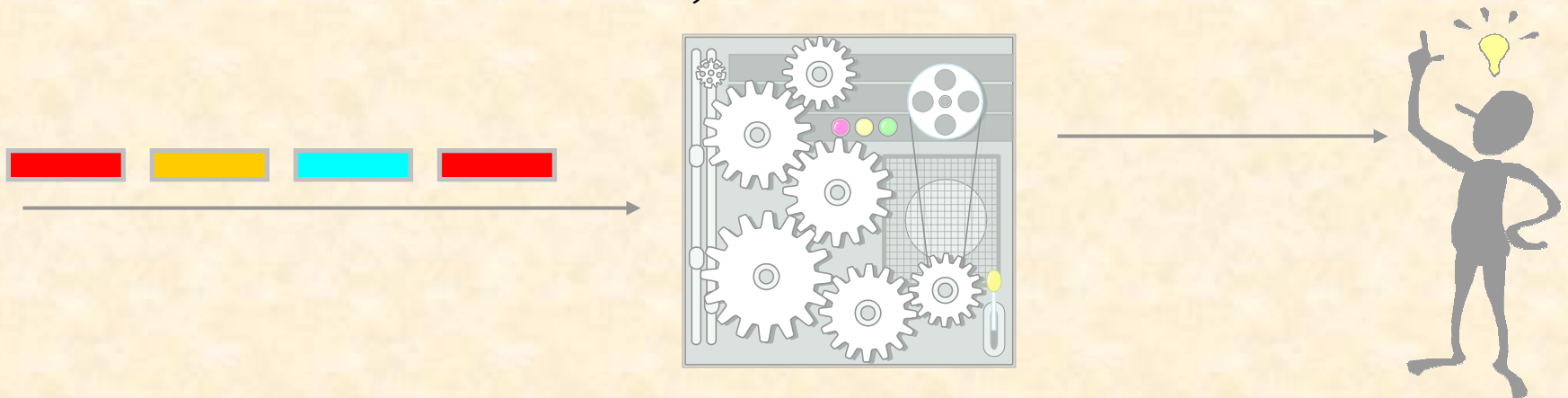


Interactive traffic analysis and visualization with Wisconsin Netpy

Cristian Estan, Garret Magin

University of Wisconsin-Madison

USENIX LISA, 19 December 2005

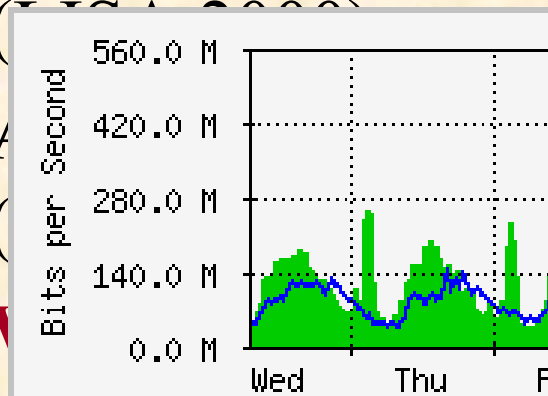


Traffic monitoring – the big picture

Tool

- MRTG
(LISA 1998)

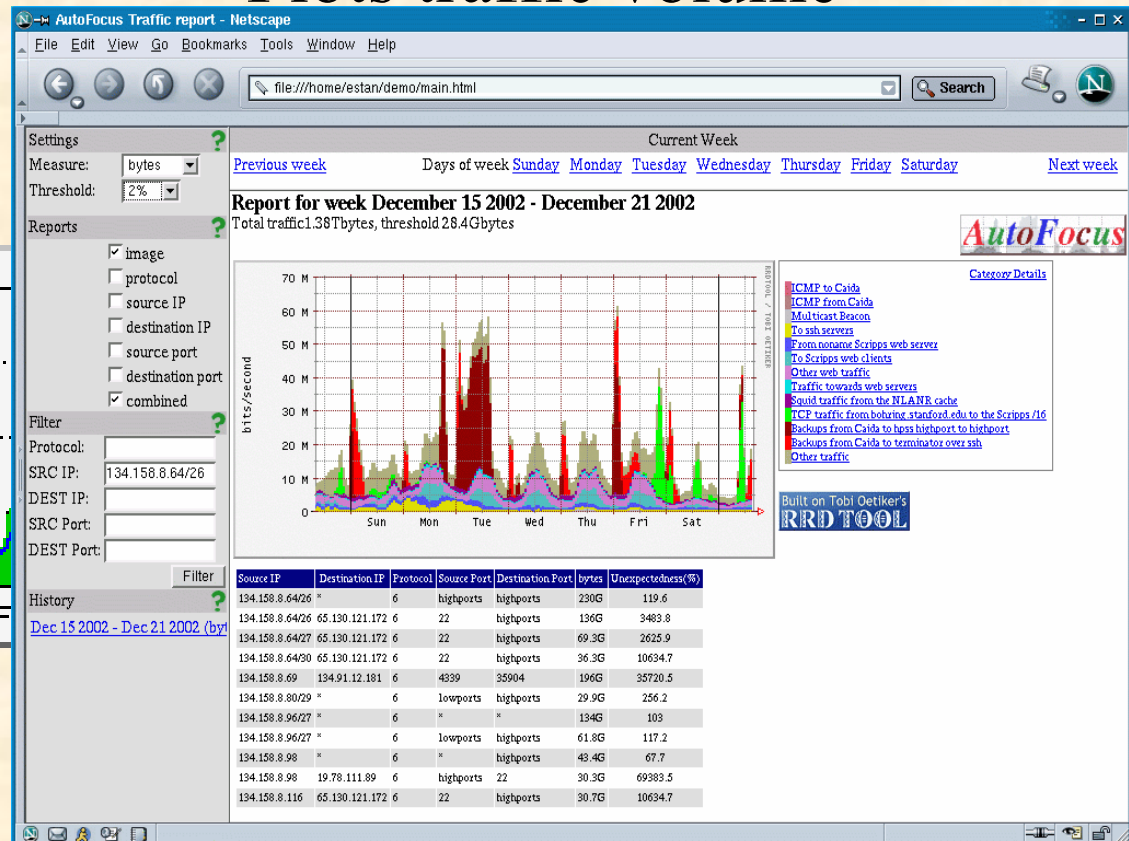
- FlowScan



(LISA 2005)

Major new feature

- Plots traffic volume



Talk overview

- Hierarchical heavy hitter analysis
- Traffic analysis with Netpy's GUI
- Netpy's database of flow data
- Future directions

Example: who sends much traffic?

| Approach | Which sources' traffic to report |
|-----------------------------------|--|
| Pre-configured | Pre-configured servers x,y, and z |
| Heavy hitters (top k) | Whichever IP addresses send $\geq 1\%$ of total traffic |
| Hierarchical heavy hitters | IP addresses and prefixes that send $\geq 1\%$ |

Refining hierarchical heavy hitters

- **Problem:** might generate large, redundant reports
- **Example:** heavy hitter IP address X is part of 32 more general prefixes and all will be reported even if they contain no traffic other than the traffic of X
- **Solution:** Report prefixes only if their traffic is significantly beyond that of more specific prefixes reported (difference \geq threshold)
- **Generalization:** can use other hierarchies that focus on ports, AS numbers, routing table prefixes, etc.

HHH report example

The screenshot shows the NetPy 2 v.0.2 application window. The title bar reads "NetPy 2 v.0.2 - ID: 0". The menu bar includes "File", "Edit", "View", and "Help". The toolbar contains navigation arrows, a dropdown menu set to "Unidimensional - Text", two "Source" dropdowns, a text input field containing "10" (circled in red), and a "Submit Query" button. The main display area shows traffic data:

- Total Traffic: 21.61GB
- ◆ 128.0.0.0/1: 21.61GB
- ◆ 142.62.43.84/32: 2.54GB
- ◆ 142.62.96.0/20: 2.34GB
- ◆ 145.89.0.0/16: 10.17GB
- ◆ 145.89.5.0/24: 2.39GB
- ◆ 145.89.20.176/29: 2.35GB
- ◆ 145.89.192.0/18: 2.87GB
- ◆ 197.216.34.234/31: 3.30GB

Below the traffic data, there are fields for "Data Specification 1", "Data Specification 2", a unit dropdown set to "Bytes", and a "Comparison" checkbox. The "Date" field shows a range from "07/31/2002 11:30 PM" to "08/01/2002 12:00 AM", with a "Links" button to the right. A search filter bar at the bottom contains the text "Src Addr: * | Src Port: * | Dst Addr: * | Dst Port: * | Protocol: *". The status bar at the very bottom contains the text "this is the status bar".

Other hierarchies used by Netpy

- **Application hierarchy** (source port centric)
 - First group by protocol
 - Within TCP and UDP separate traffic coming from low ports (<1024) and high ports (≥ 1024)
 - Separate by individual source port
 - Separate by (source port, destination port) pair
- Destination port centric application hierarchy
- **User defined categories**
 - Group traffic into categories using ACL-like rules
 - Report all categories above the threshold
 - Can modify mappings at run time

Example: application HHH report

NetPy 2 v.0.2 - ID: 0 -

File Edit View Help

Unidimensional - Text / Location Source / Source / 6 Submit Query

Total Traffic: 22.72GB

- *: 22.72GB
 - Src Port: (1024 - 65535) | Protocol: 6 (tcp): 15.85GB
 - Src Port: 1500 | Protocol: 6 (tcp): 1.60GB
 - Src Port: 29342 | Dst Port: 1500 | Protocol: 6 (tcp): 2.54GB
 - Src Port: 80 | Protocol: 6 (tcp): 1.94GB
 - Src Port: 50000 | Dst Port: 50000 | Protocol: 17 (udp): 1.53GB
 - Src Port: 50100 | Dst Port: 50100 | Protocol: 17 (udp): 1.56GB

Data Specification 1 | Data Specification 2 | Bytes / Comparison

Date: 07/31/2002 11:30 PM - 08/01/2002 12:00 AM Links

Src Addr: * | Src Port: * | Dst Addr: * | Dst Port: * | Protocol: *

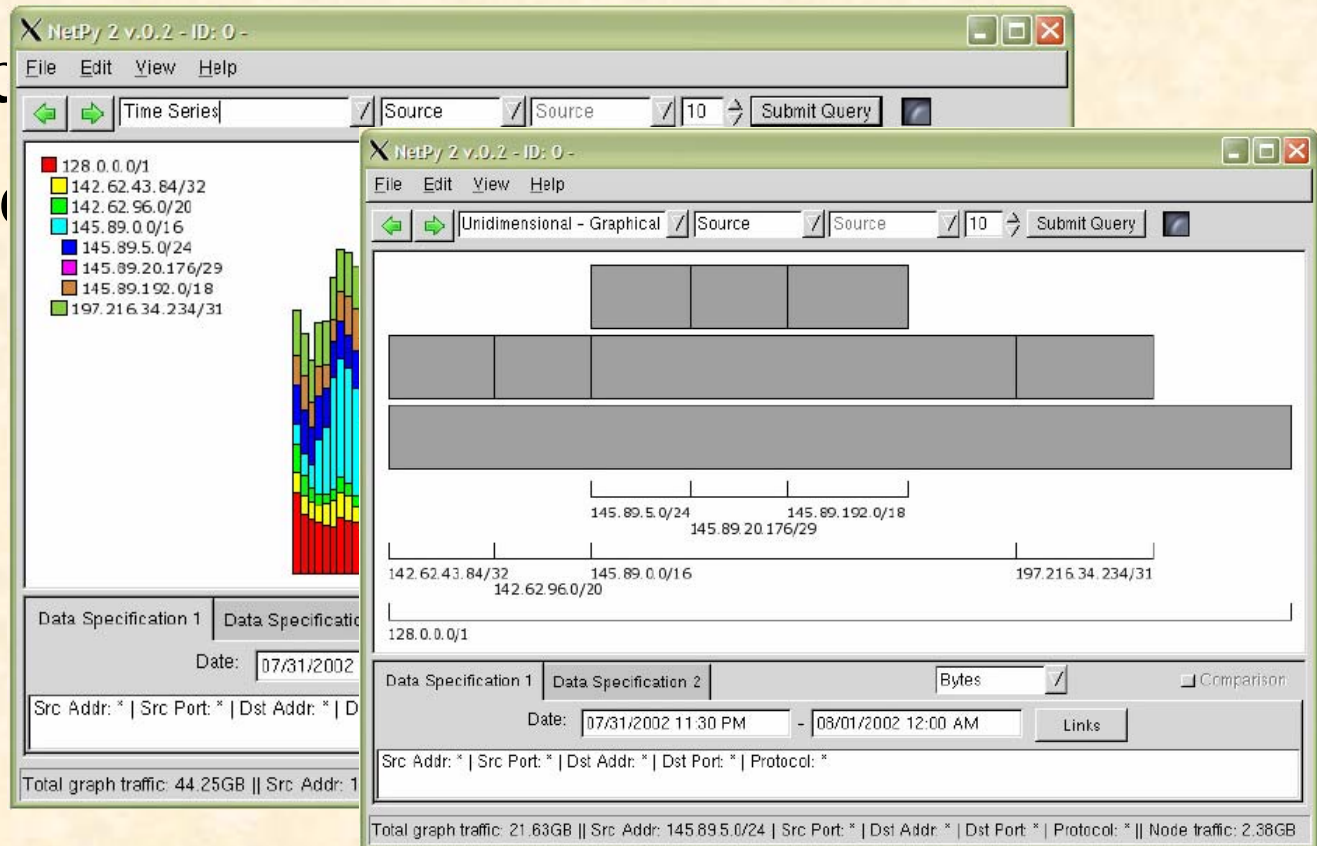
this is the status bar

Overview

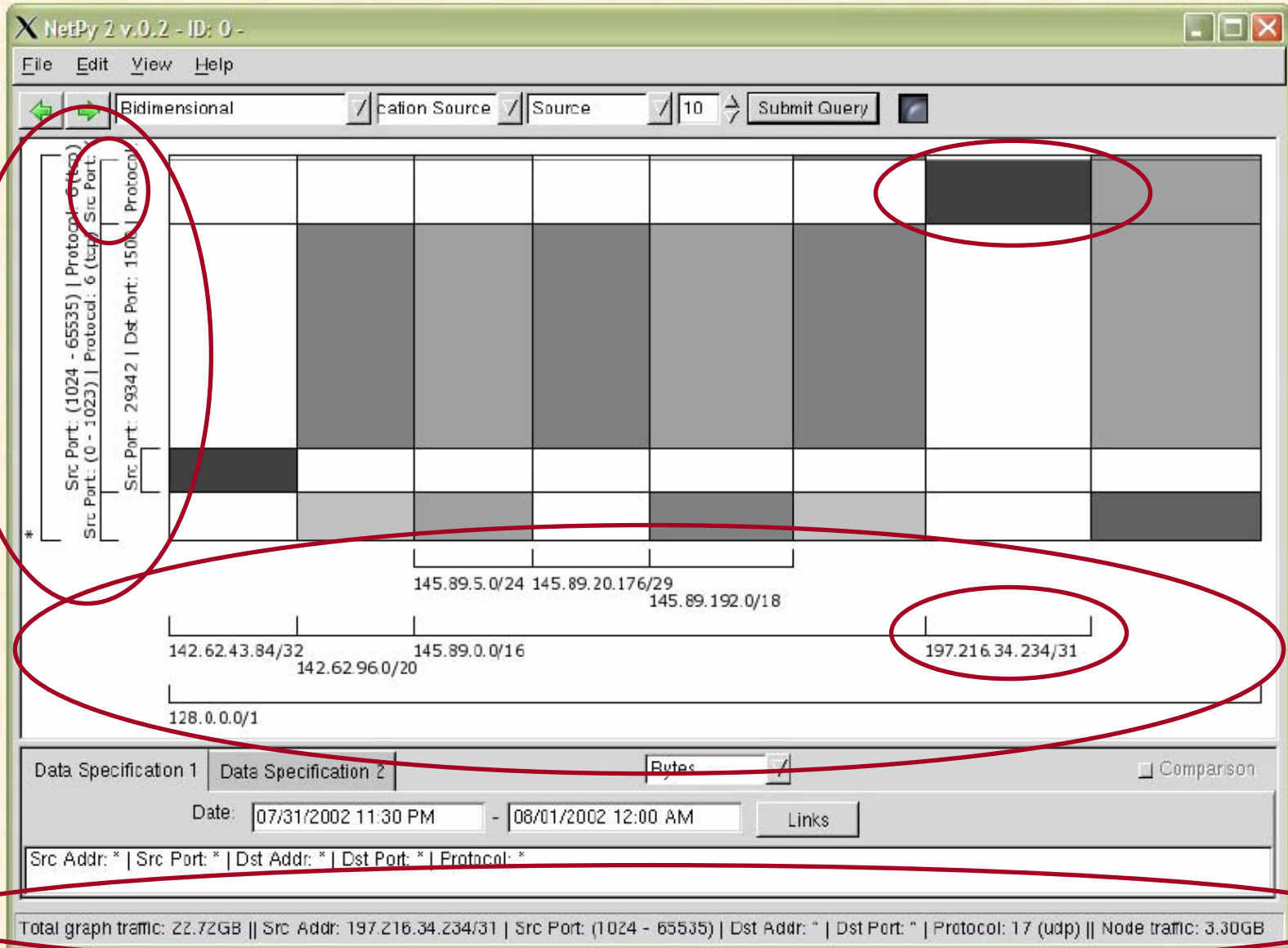
- Hierarchical heavy hitter analysis
- Traffic analysis with Netpy's GUI
 - Types of analyses supported
 - Selecting data to analyze (interactive drill-down)
- Netpy's database of flow data
- Future directions

Types of analyses supported

- Textual HHH analyses on all 5 hierarchies
- Time series plots on all 5 hierarchies
- Graphical
- “Bidimensional”



Example: bidimensional report



Selecting data to analyze

- User selects **time interval** to analyze
- Can select whether to measure data in **bytes, packets, or flows** (helps catch scans)
- Can specify a **filter** (ACL-like rules) to select the portion of the traffic mix to analyze
- **Clicking** on graphical elements in the reports updates the rules in the filter
 - This allows **interactive drill-down**

Overview

- Hierarchical heavy hitter analysis
- Traffic analysis with Netpy's GUI
- Netpy's database of flow data
 - Grouping traffic by links
 - Adding traffic through the console
 - Scalability through sampling
- Future directions

Grouping traffic into links

- Can configure Netpy to group traffic by **“link”**
 - ACL-like syntax, based on NetFlow fields:
 - Exporter IP address (prefix match)
 - Next hop (prefix match)
 - Source/destination address (prefix match)
 - Input/output interface (exact match)
 - Engine type/ID (exact match)
- Flow records grouped into files by start time, separate directory for every link

Adding traffic through the console

- Netpy's **console** has command for adding **NetFlow files** to database
 - Accepts anything flow-tools can parse
 - If using sampled NetFlow, specify sampling rate
 - Can override link mappings from configuration file

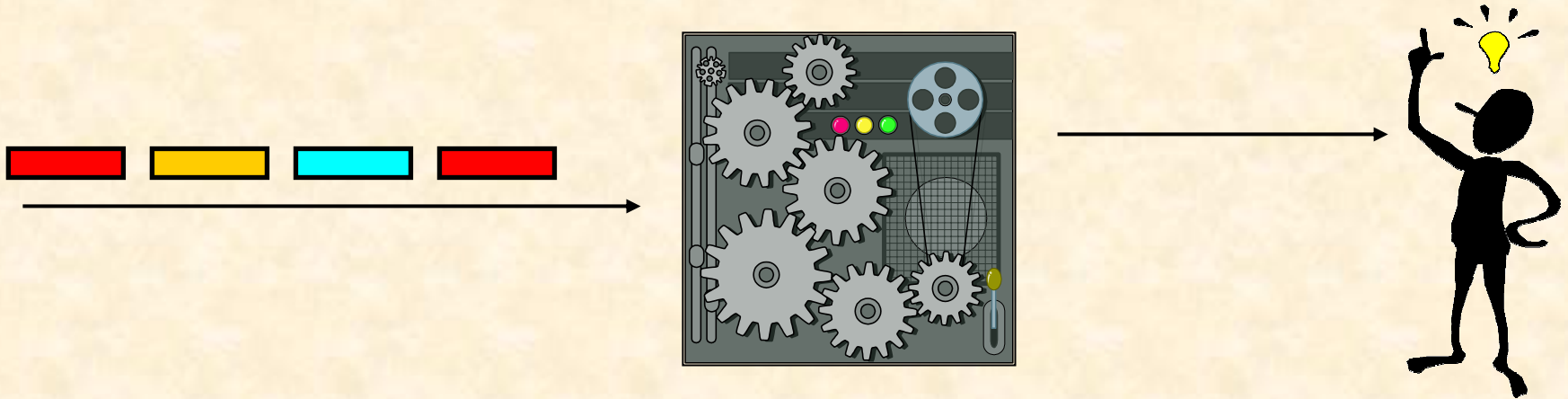
Scalability through sampling

- **When writing** to database Netpy samples flow records to ensure database won't get too large
 - Configuration file gives size limit (MB/hour)
- **When reading** from database, if the number of flow records is too large even after applying the filter, further sampling is performed
 - Helps speed up HHH algorithms

The future of Netpy

- Features on the roadmap
 - Feedback, suggestions, patches – all welcome
 - Client/server operation
 - Better performance (caching, multilevel database)
 - More hierarchies (e.g. based on DNS)
 - Comparative analysis of two data sets
 - Anomaly detection, generating alerts
 - We need your help with getting this one right

Questions?



- Netpy home page: <http://wail.cs.wisc.edu/netpy/>
- Acknowledgements
 - Netpy implementors: Garret Magin, Cristian Estan, Ryan Horrisberger, Dan Wendorf, John Henry, Fred Moore, Jaeyoung Yoon, Brian Hackbarth, Pratap Ramamurthy, Steve Myers, Dhruv Bhoot
 - Other help from: Mike Hunter, Dave Plonka, Glenn Fink, Chris North