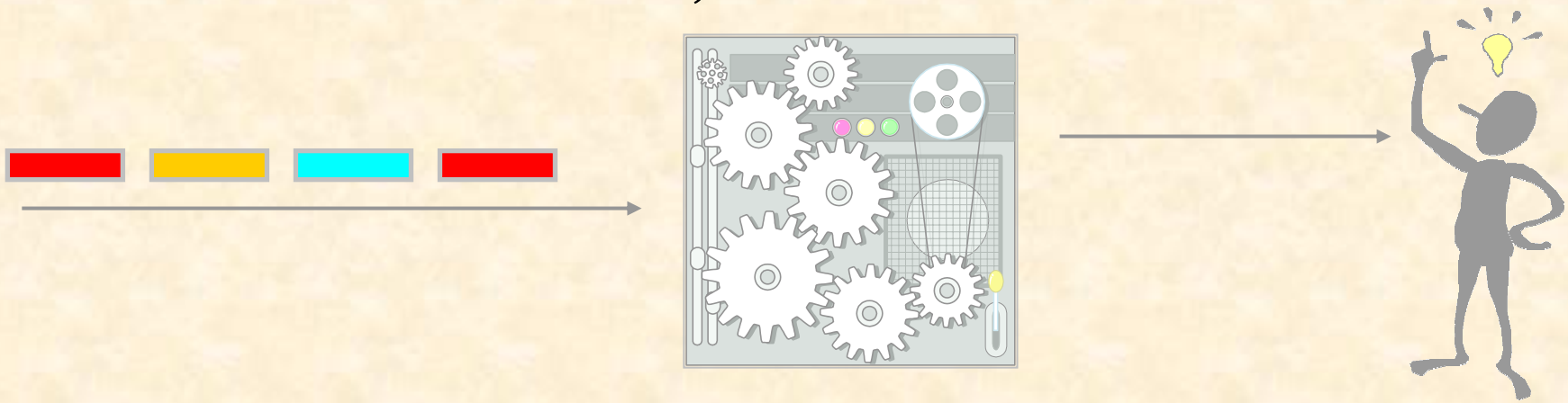
Interactive traffic analysis and visualization with Wisconsin Netpy

Cristian Estan, Garret Magin

University of Wisconsin-Madison

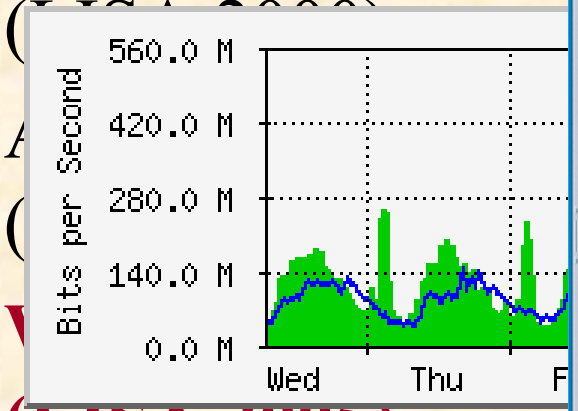
USENIX LISA, 19 December 2005



Traffic monitoring – the big picture

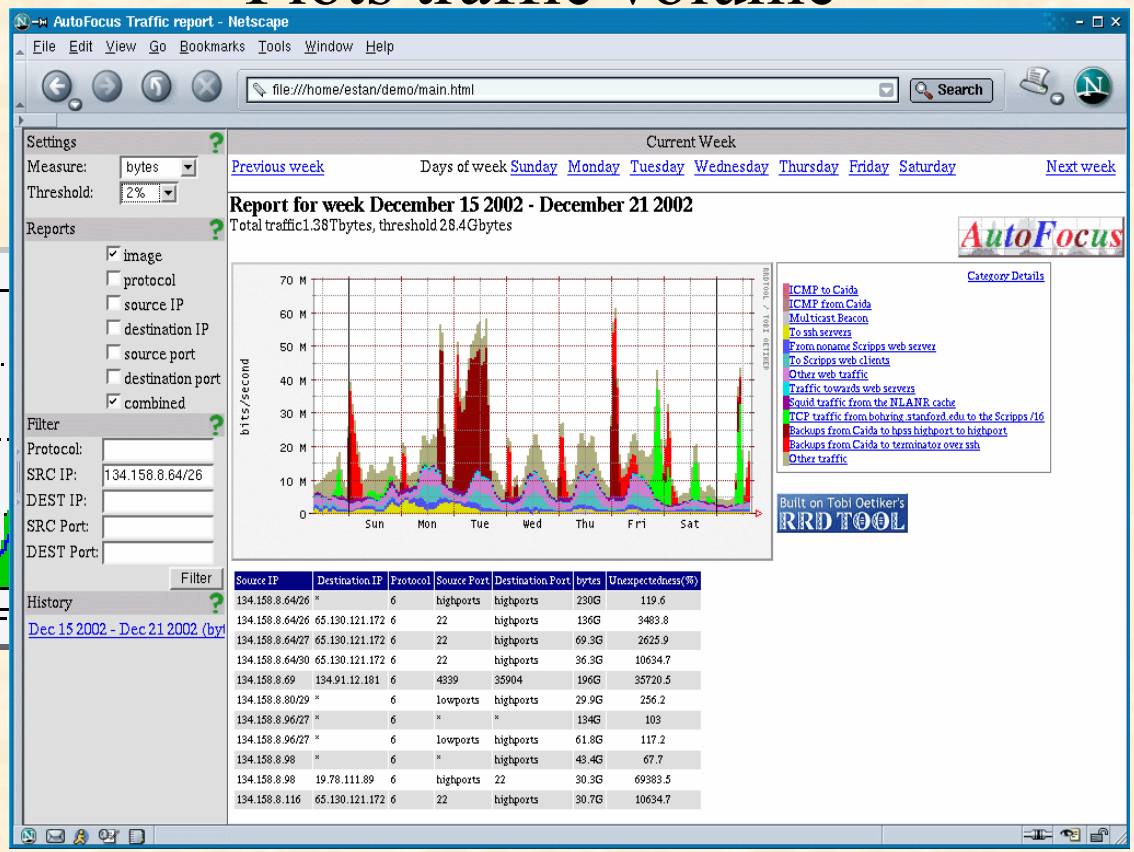
Tool

- MRTG (LISA 1998)
- FlowScan (
- A
- (
- **W**
- **(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

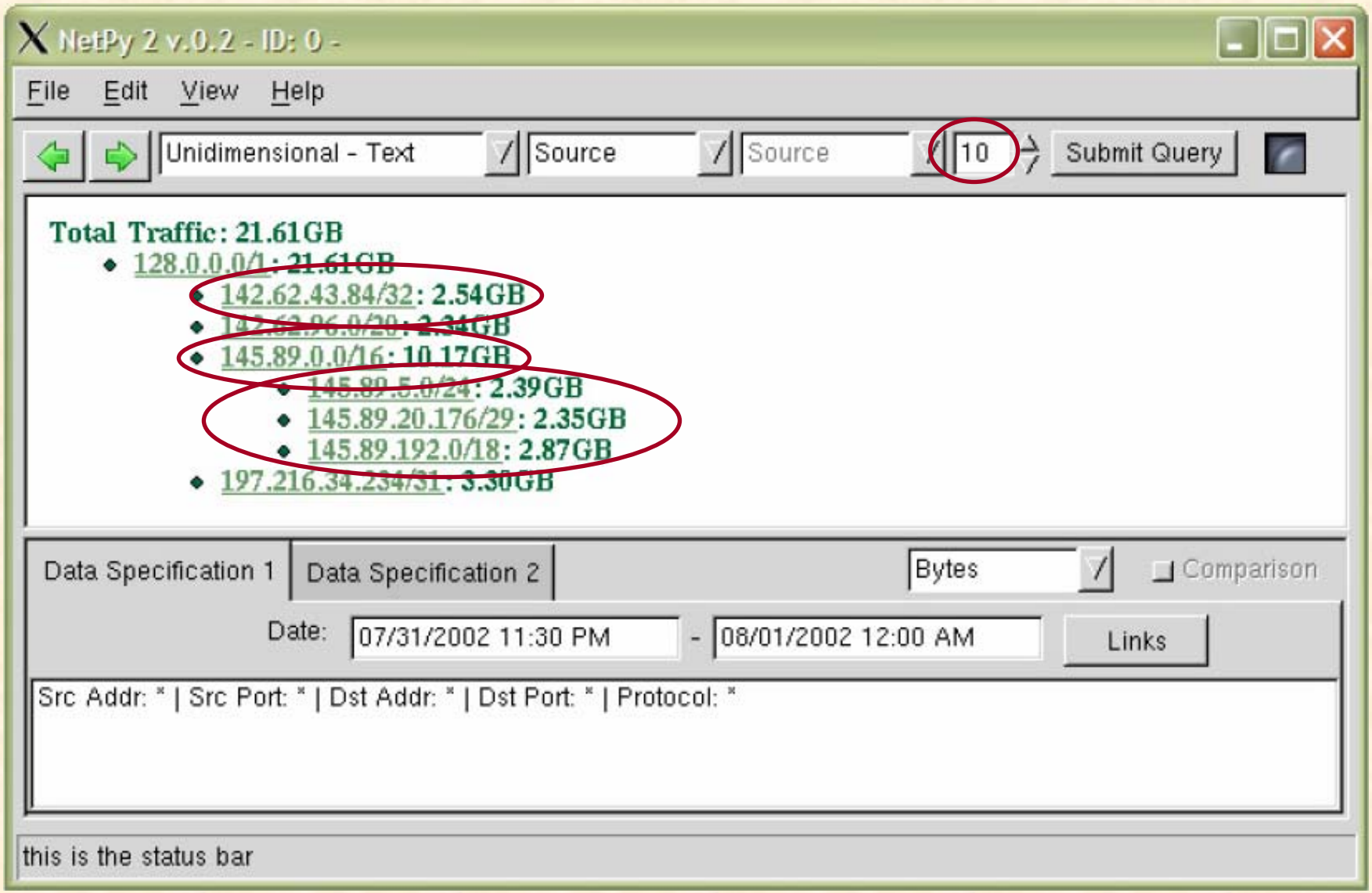
Example: who sends much traffic?

Aproach	**Which sources' traffic to report**
Pre-configured	Pre-configured servers x,y, and z
Heavy hitters (top k)	Whichever IP addresses send ≥ 1% of total traffic
Hierarchical heavy hitters	**IP addresses and prefixes that send ≥ 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.

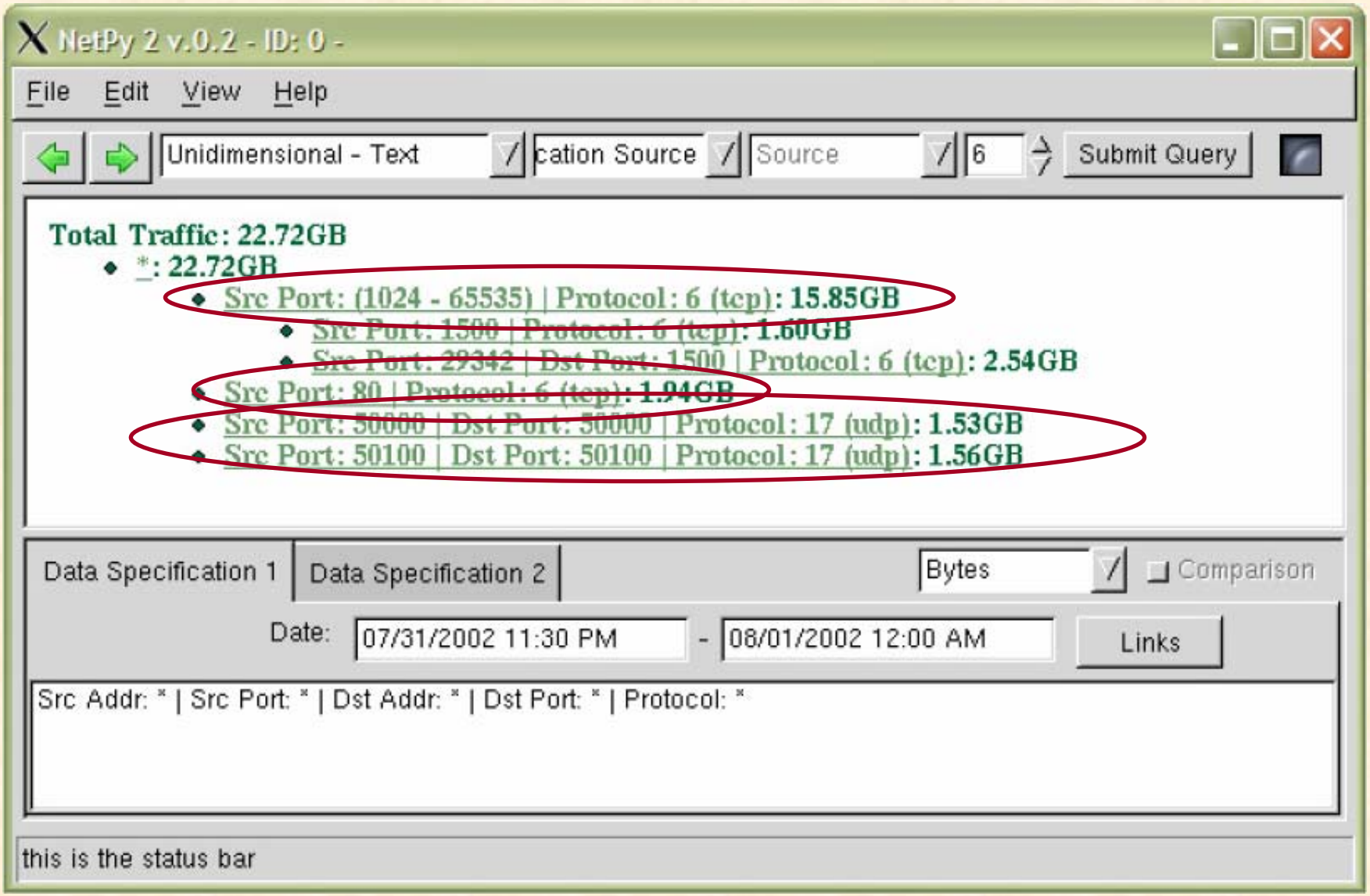
HHH report example



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

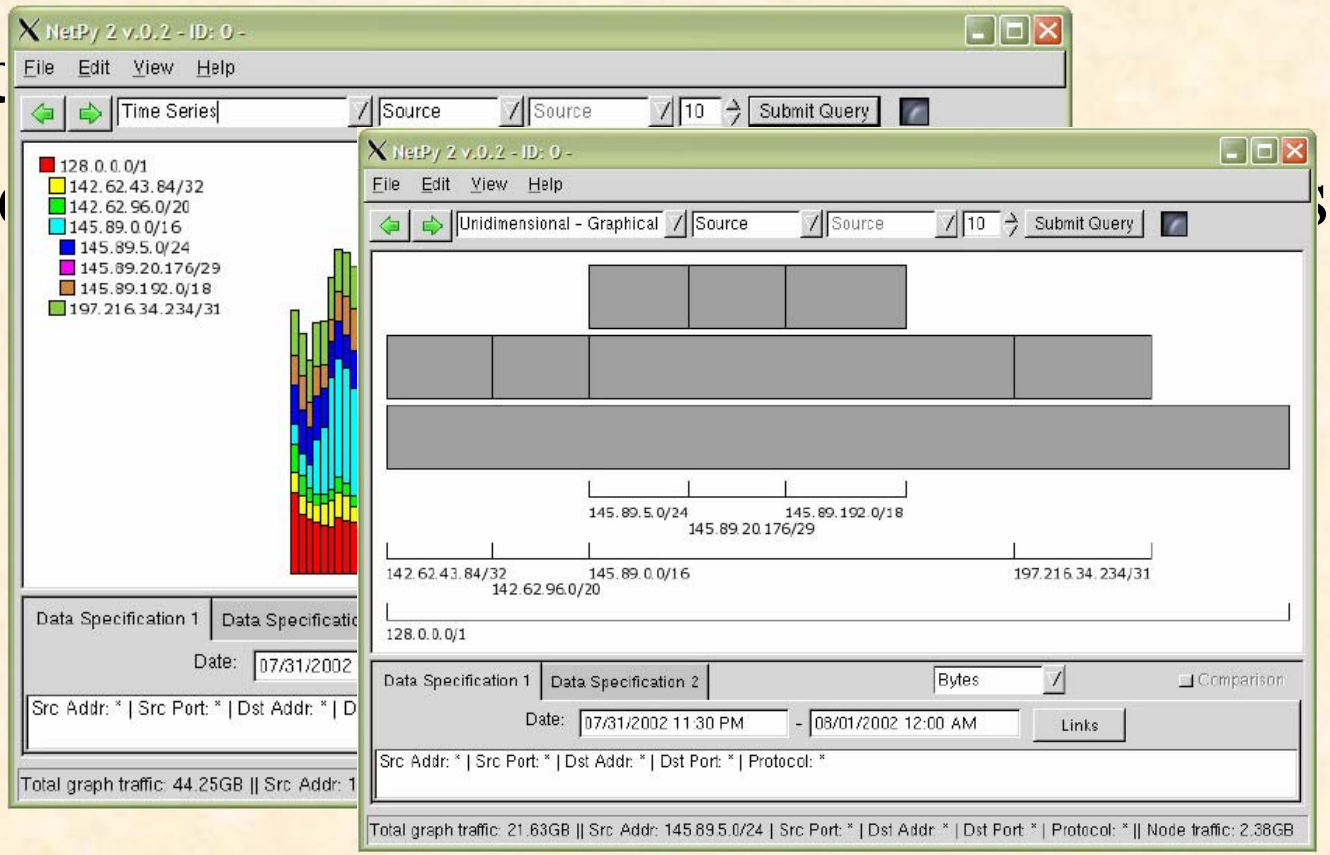


Overview

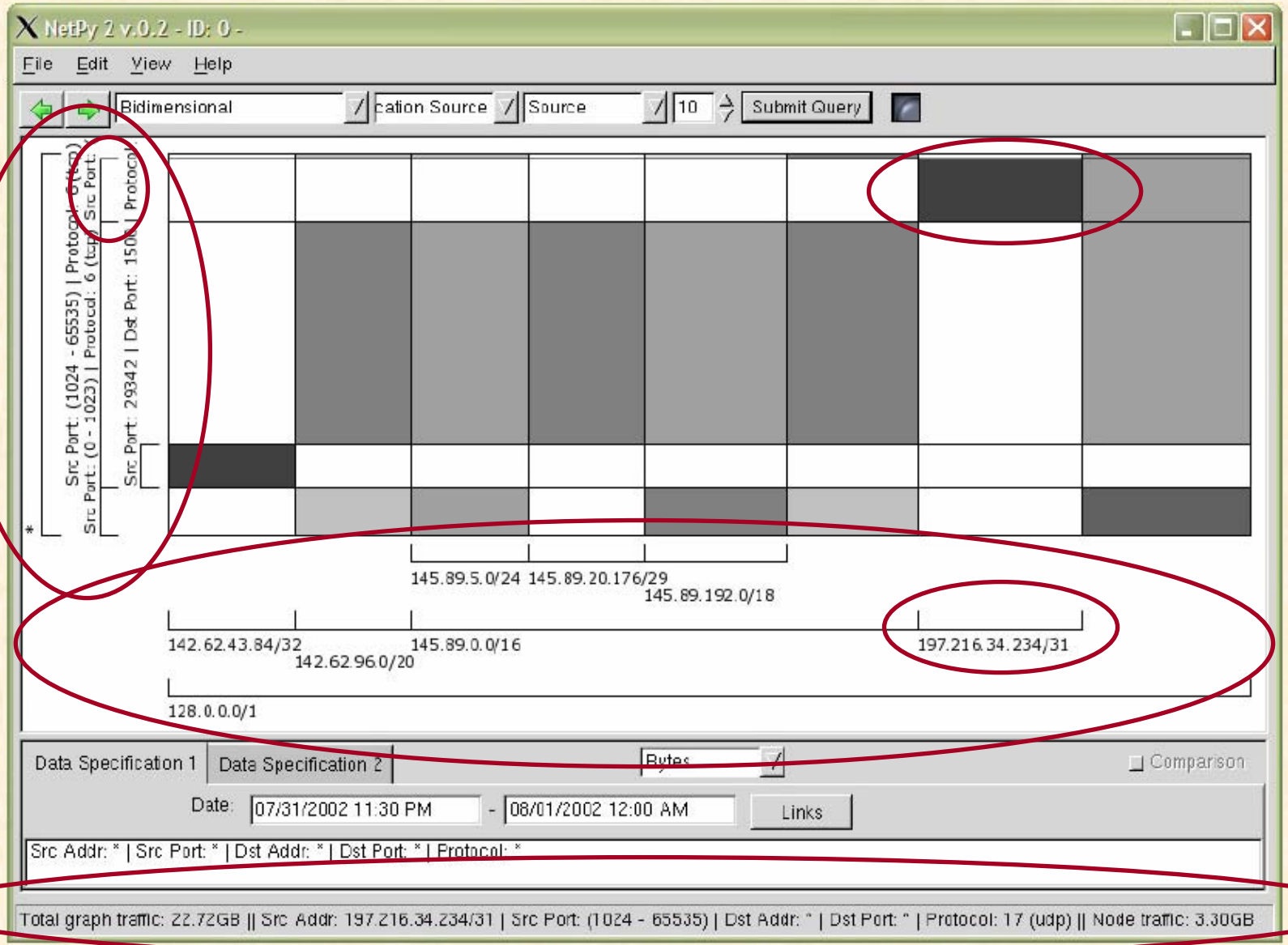
- Hierarchical heavy hitter analysis
- Traffic analysis with Netpy's GUI
  - Types of analyses supported
  - Selecting data to analyze (interactive drill-down)
- Netpy's database of flow data
- Future directions

Types of analyses supported

- Textual HHH analyses on all 5 hierarchies
- Time series plots on all 5 hierarchies
- Graphic
- "Bidim



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

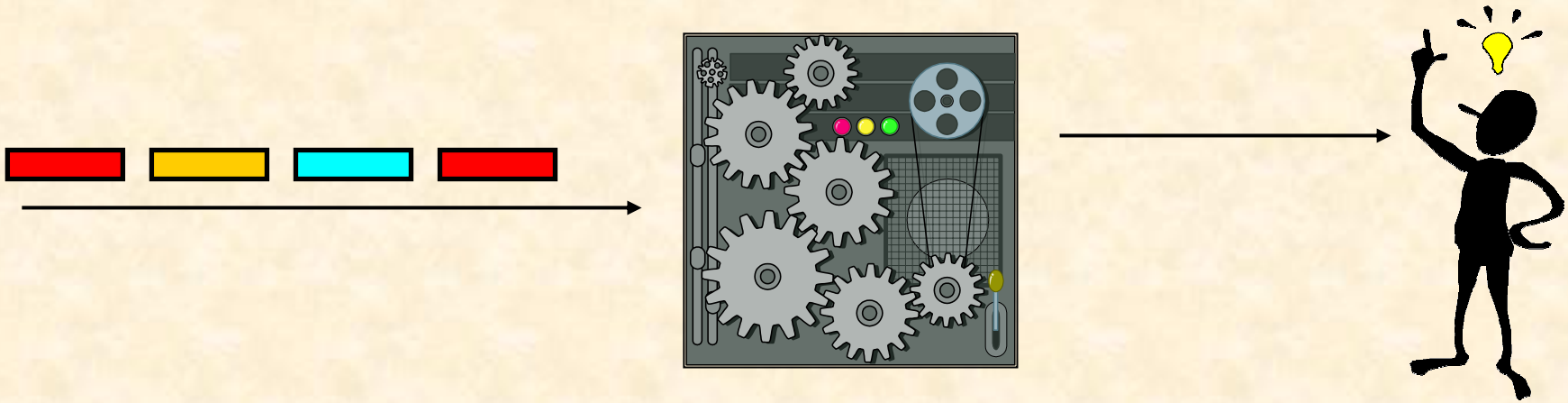
Scalability through sampling

- **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