Traffic Characteristics and Measurement: Internet Measurements

Arno Wagner

wagner@tik.ee.ethz.ch

Communication Systems Laboratory

Swiss Federal Institute of Technology Zurich (ETH Zurich)

Lecture Outline Outline

1. Why Capture Network Traffic?
2. Packet Capturing
3. Flow Capturing
4. Case-Study: The DDoSVax-Project
5. Related Student Thesis offers @ CSG

1. Why Capture Network Traffic?

- Accounting
- Network monitoring
- Forensics
- Research
- Intelligence
- ...

Tasks Related to Data Capturing

- Obtaining/creating suitable sensors
- Sensor placement and operation
- Short-term storage at or close to sensor
- Transfer off-site
- (Long-term) storage
- Processing (libraries, infrastructure)
- ...

2. Packet Capturing

Capturing alternatives

- Complete packets, i.e. complete Layer 3 payload
- Layer 3+4 packet headers
  (Typically first 48 bytes of Layer 3 packet)
- First 60 bytes of each Layer 3 packet
- ...

Reasons to Capture Packets

- ”Complete” network traffic, most accurate
- No aggregation or preprocessing needed
- Single packet timing and sizes
- Application layer data
- ...

Reasons Not to Capture Packets

- Difficult/impossible for fast links
- Massive amount of data on fast links
  - difficult to store (short and long term)
  - difficult to transfer
  - difficult to process
- Data payloads often not needed
- May be illegal to do
- ...

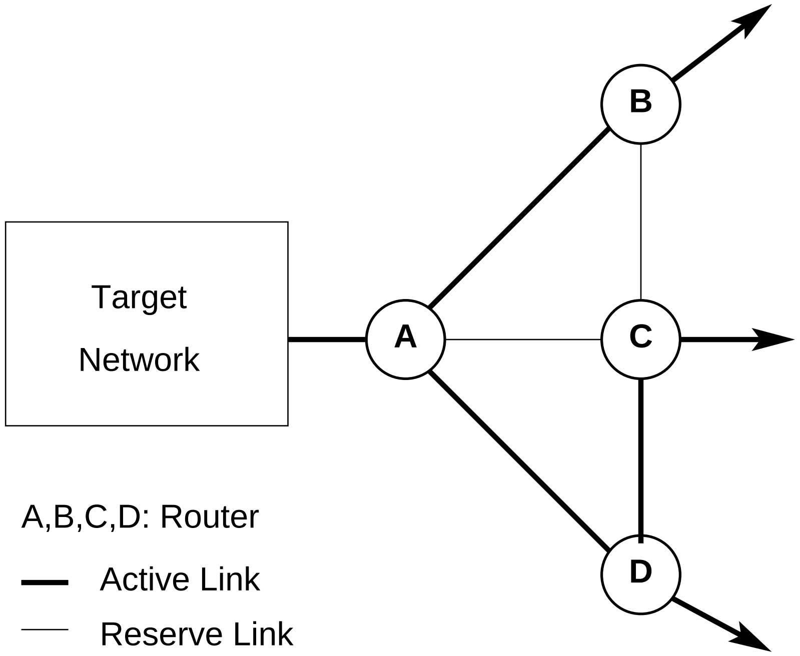
Packet Sensors

- Fast Ethernet (FE), 100Mb/s:
  Standard PCs with Linux, FreeBSD, ...
- Gigabit Ethernet (GbE):
  (Fast PC), network processor, special hardware
  Special software, tailored to the hardware
- > GbE: special hardware (custom built?)
- Alternative for specific traffic:
  (More or less) transparent proxies

Sensor Placement I

- Needs to see all traffic to/from site of interest
- Needs to have data transfer resources
- May need to be invisible (intelligence operations)
- ...

Sensor Placement Example



Valid Placements: A or (B **and** C **and** D) !

Sensor Placement II

- Capturing a link with bandwidth n gives **two** packet streams with bandwidth n!
- Capturing at a router with m links that have bandwidth n gives

$$2 * m * n$$

  bandwidth to be captured!

Storage and Transfer

Example: Gigabit Ethernet (GbE), single bidirectional link

- Full traffic, worst case:
  200MB/s = 720GB/h = 17TB/day = 6.3PB/year
  Needs >= 2.5GbE dedicated link for reliable transfer
- Headers only: ?

Headers: Same worst-case (SYN-flooding, ICMP,...).

- PCI bus: 135 MB/s
- Harddisk/tape-drive: 50MB/s

About Time-Stamps

Timestamps are critical to correlate data from multiple sensors.

- May be needed to determine packet sequence
  $\Rightarrow$ Nanosecond accuracy may be needed!
- May have absolute time
  $\Rightarrow$ >= 32 bit for seconds

May be up to 64 bit per timestamp.

Processing

- Real-time: Same problems as capturing and storage
- Connections need to be reconstructed
  $\Rightarrow$ May need partial network stack
- Payload processing has arbitrary complexity

Legal and Ethical Aspects

Disclaimer: I am no expert!

- Payloads may fall under privacy laws
  $\Rightarrow$ Capturing and/or storage may be illegal
- Respect the privacy of individuals !/?
- Payloads may contain passwords, credit-card numbers, etc.
  $\Rightarrow$ Liability if misused (e.g. identity-theft) ?
- Criminal activity in payloads: Obligation to report?

Headers are far less problematic.

Summary for Packets

- Very accurate
- Difficult/expensive to capture, transfer, store, process
- Long-term continuous monitoring often infeasible
- Real-time monitoring difficult
- May cause legal problems
- ...

3. Flow Traces

A *network flow* is an aggregated stream of packets from one source (IP, port) to one destination (IP, port).

- Addresses: IP-addresses, ports (TCP, UDP)
- Timestamp: First packet, last packet
- Counters: Bytes, packets
- Flags (TCP): SYN, FIN, RST, ...
- ...

Limitations of Flow Capturing

Examples:

- Inaccurate/incomplete header information
- No payload information
- No packet sizes
- Maximum flow duration (e.g. 15 minutes)
- Maximum idle timeout (e.g. 30 seconds)
- Maximum data length (e.g. 4GiB)

Flow Data Format Alternatives

- NetFlow v5 (v7)
- NetFlow v9 (not yet implemented widely)
- IPFIX (still in definition)

Bi-directional unusual, Internet has asymmetric routing!

Example: NetFlow v5

- Available in current Cisco Routers
- Exports UDP packets from the routers
- 24 Byte packet header
- 48 Bytes per flow
- Grown historically

NetFlow v5 UDP Packet Header

```
struct netflow_v5_header {
  uint16_t  version;
  uint16_t  count;
  uint32_t  SysUptime;
  uint32_t  unix_secs;
  uint32_t  unix_nsecs;
  uint32_t  flow_sequence;
  uint8_t   engine_type;
  uint8_t   engine_id;
  uint16_t  reserved;
};
```

NetFlow v5 UDP packet Flow Record

```
struct netflow_v5_record {
  uint32_t  addr;          uint32_t  dstaddr;
  uint32_t  nexthop;       uint16_t  input;
  uint16_t  output;        uint32_t  dPkts;
  uint32_t  dOctets;       uint32_t  First;
  uint32_t  Last;          uint16_t  port;
  uint16_t  dstport;       uint8_t   pad1;
  uint8_t   tcp_flags;     uint8_t   prot;
  uint8_t   tos;           uint16_t  _as;
  uint16_t  dst_as;        uint8_t   _mask;
  uint8_t   dst_mask;      uint16_t  pad2;
};
```

Flow Sensors

- Typically router-integrated (”free”)
- Export e.g. via UDP
- Export can be in dedicated link or within normal traffic
- Data-rate limited by sensor (limited buffer)
  $\Rightarrow$ Data loss with too many short flows
- In fast networks sampling may be used

Transfer, Storage

- Typically feasible with commodity hardware
- Long-term storage needs tape/disk library
- Compression unproblematic

See case study for more information

4: Case-Study: DDoSVax Project

`http://www.tik.ee.ethz.ch/~ddosvax/`

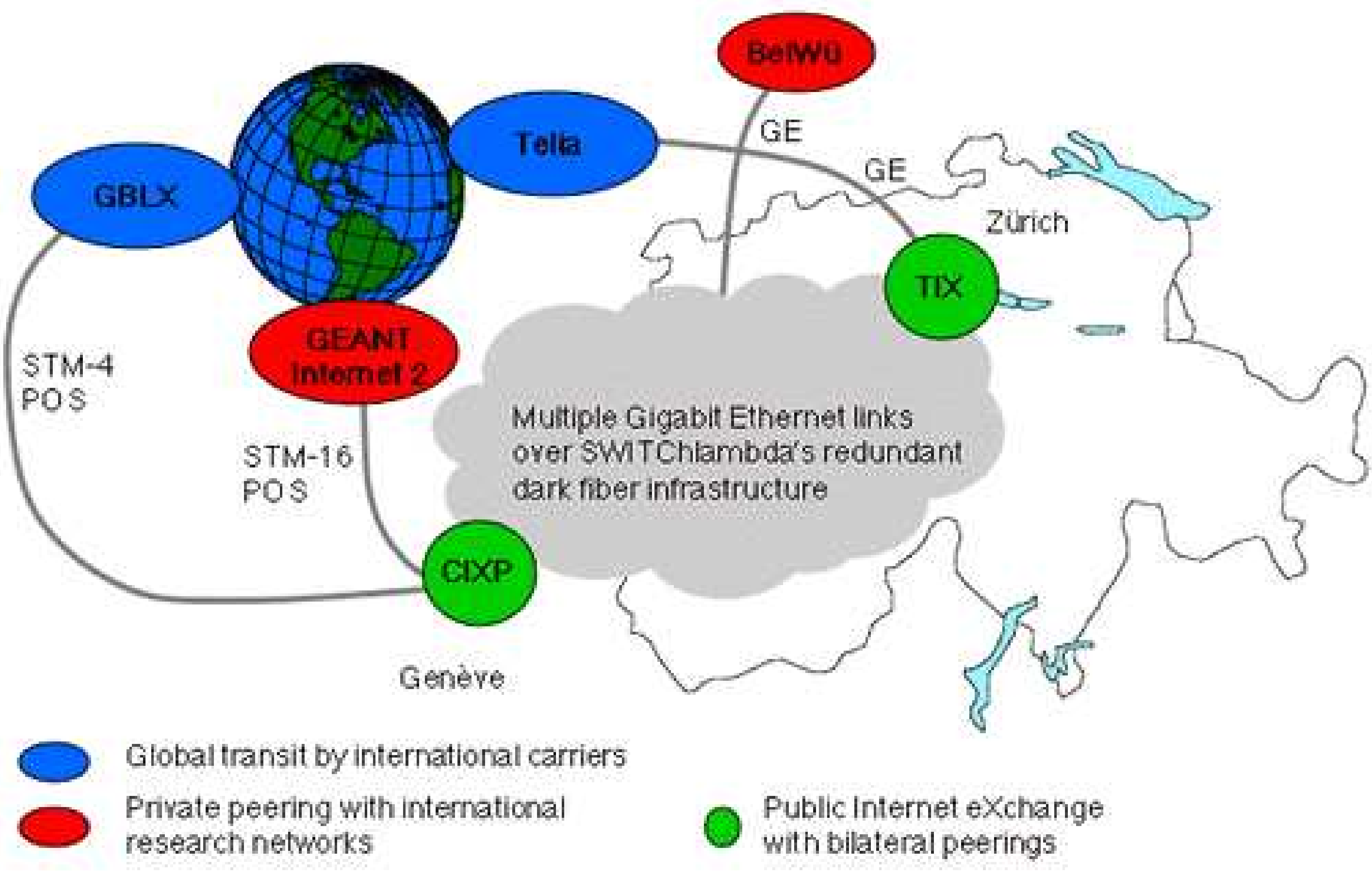
- Collaboration between SWITCH (www.switch.ch) and ETH Zurich (www.ethz.ch)
- Aim (long-term): Analysis and countermeasures for DDoS-Attacks and Internet Worms
- Start: Begin of 2003
- Funded by SWITCH and the Swiss National Science Foundation

SWITCH

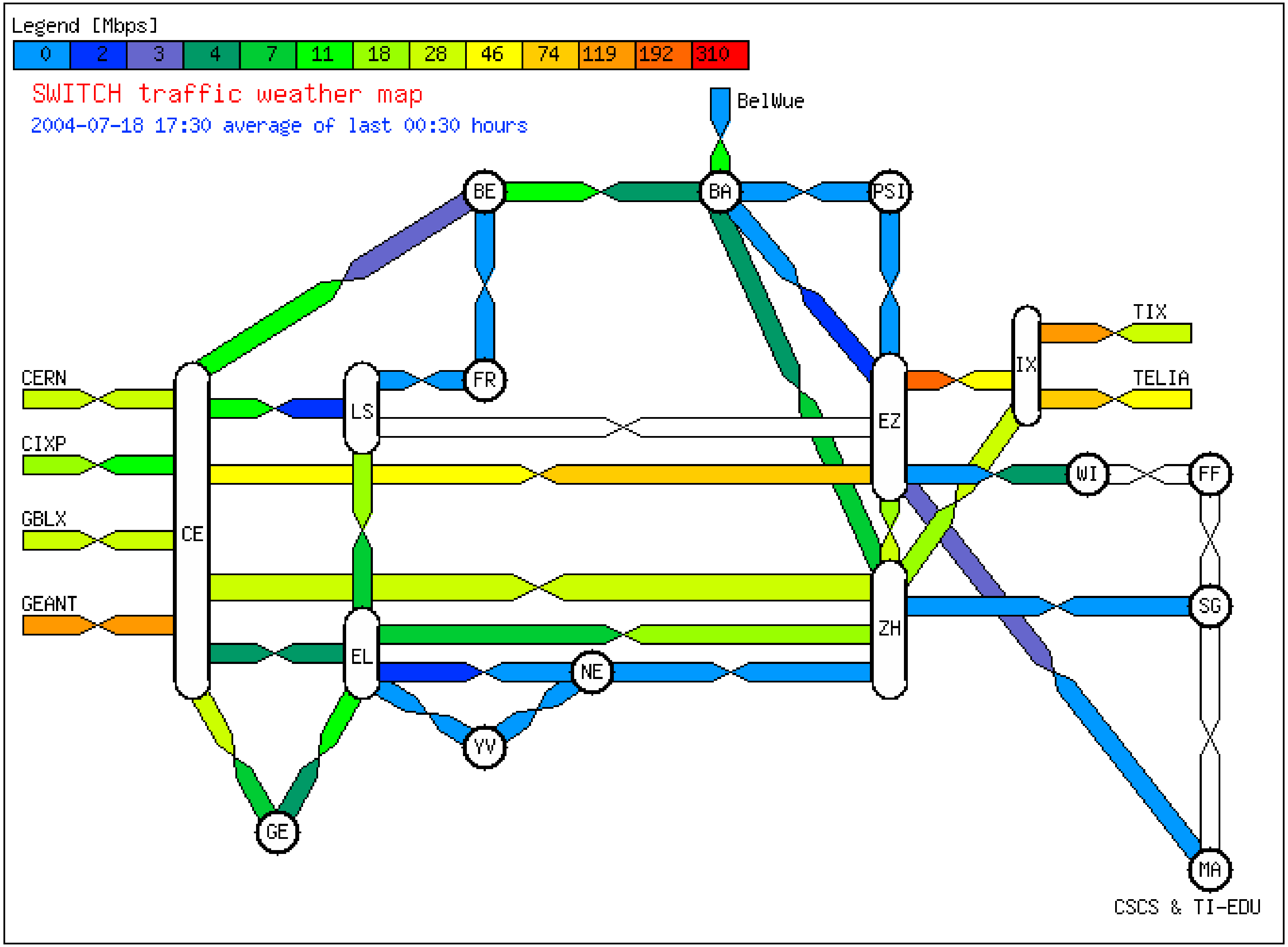
The Swiss Academic And Research Network

- .ch Registrar
- Links most (all?) Swiss Universities
- Connected to CERN
- Carried around 5% of all Swiss Internet traffic in 2003
- Around 60.000.000 flows/hour
- Around 300GB traffic/hour

SWITCH Peerings



SWITCH Traffic Map

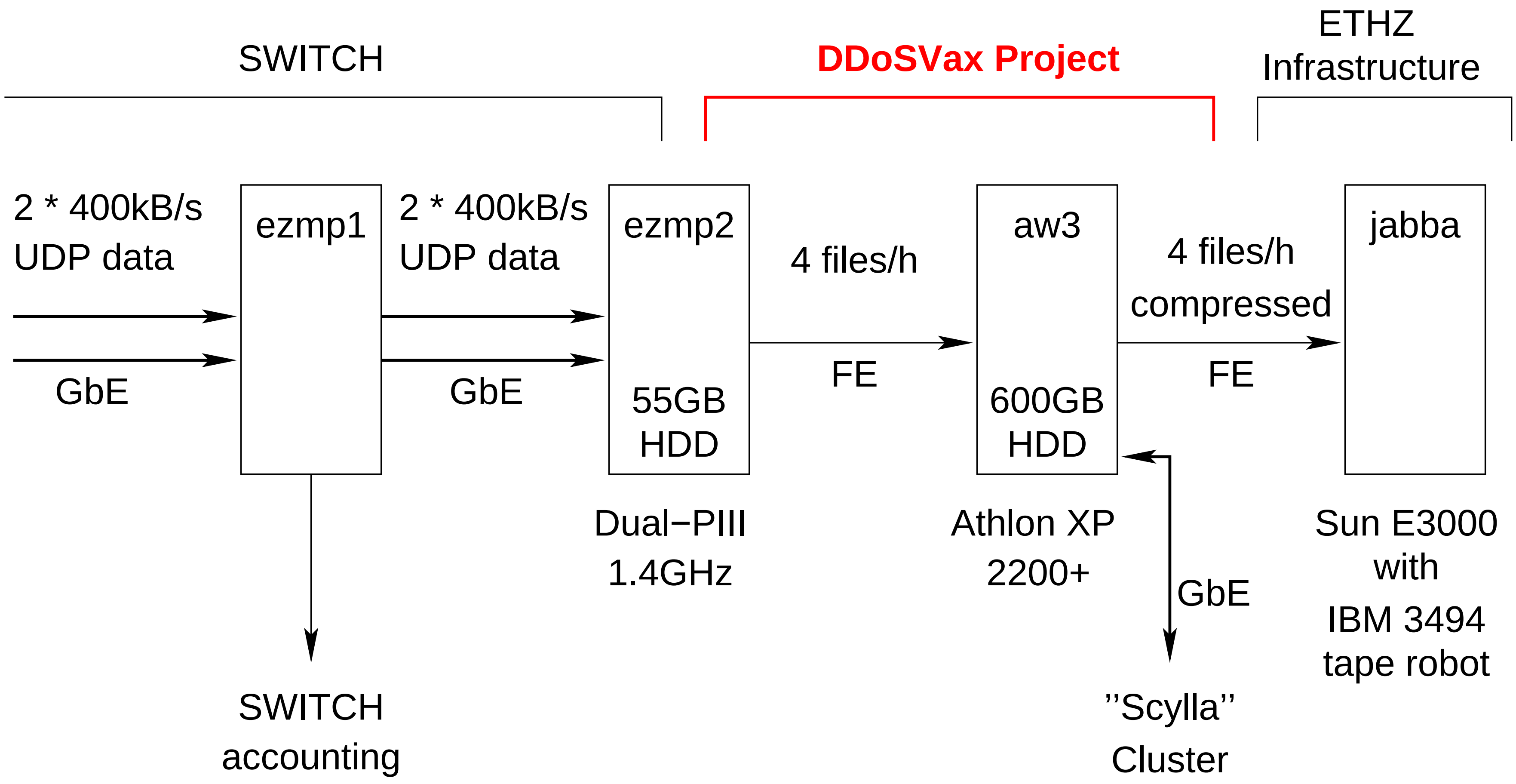


NetFlow Data Usage at SWITCH

- Accounting
- Network load monitoring
- SWITCH-CERT, forensics
- DDoSVax (with ETH Zurich)

Transport: Over the normal network

NetFlow Data Flow



NetFlow Capturing

- One Perl-script per stream
- Data in one hour files
- Timestamps and src-IP in ”stat” file

Critical: Linux socket buffers:

- Default: 64kB/128kB max.
- Maximal possible: 16MB
- We use 2MB (app-configured)
- 32 bit Linux: May scale up to 5MB/s per stream

Capturing Redundancy

- Worker / Supervisor (both demons)
- Super-Supervisor (cron job)
  For restart on reboot or supervisor crash
- Space for 10-15 hours of data

No hardware redundancy

Data Transfer to ETHZ

- Cron job, every 2 hours
- Single Perl script
- Transfer: scp (no compression, RC4)
- Remote deletion: ssh

No compression on ezmp2. (Some other Software running there)
Bzip2 compression on ezmp2 would be possible!

Long-Term Storage Format

Full data since March 2003
Bzip2 compressed raw NetFlow V5 in one-hour files

- We need most data and precise timestamps
- We don't know what to throw away
- We have the space
- Preprocessing for specific work still possible

Latency: 5-10 minutes / hour of data

Computing Infrastructure

The ”Scylla” Cluster
`http://www.tik.ee.ethz.ch/~ddosvax/cluster/`
Servers:

- aw3: Athlon XP 2200+, 600GB RAID5, GbE
- aw4: Dual Athlon MP 2800+, 800GB RAID5, GbE
- aw5: Athlon XP 2800+, 800GB RAID5, GbE

Nodes:

- 22 * Athlon XP 2800+, 120GB, GbE

Information somewhat outdated.

Infrastructure Cost (2004)

Hardware and full installation:

- aw3 (capturing): 1600 USD + 2 MD
- aw4 (dual CPU server): 2500 USD + 3 MD
- Cluster: 24.000 USD + 1MM
- Maintenance: 1-2 MD/month

Hidden cost: Computer room, network infrastructure, software development
Scalability: Add 2*200GB HDD to each node
$\Rightarrow$ 8TB additional at 6000 USD

Lessons learned

Most important: KISS!

- Use scripting wherever possible
- Worker and Supervisor pairs are simpler
  $\Rightarrow$ ”crash” as error recovery model
- Cron as basic reliable execution service
- Email for notification: Do rate-limiting
- File-copy: Interlock and age check
- ssh, scp password-less (user key)
- Nothing needs to run as ”root”!

Remarks on Software

- Linux is stable enough
- Linux is fast enough
- Linux Software RAID1/5 works well
- XFS has issues with Software RAID
- Perl is suitable for demons
- Python is suitable for demons

Remarks on Hardware

PC hardware works well, but:

- Get good quality components (PSUs!)
- Get good cooling (HDDs/CPUs)
- Do SMART monitoring
- Do regular complete surface scans
- Have cold spares handy
- ...

Remarks on Linux Clusters

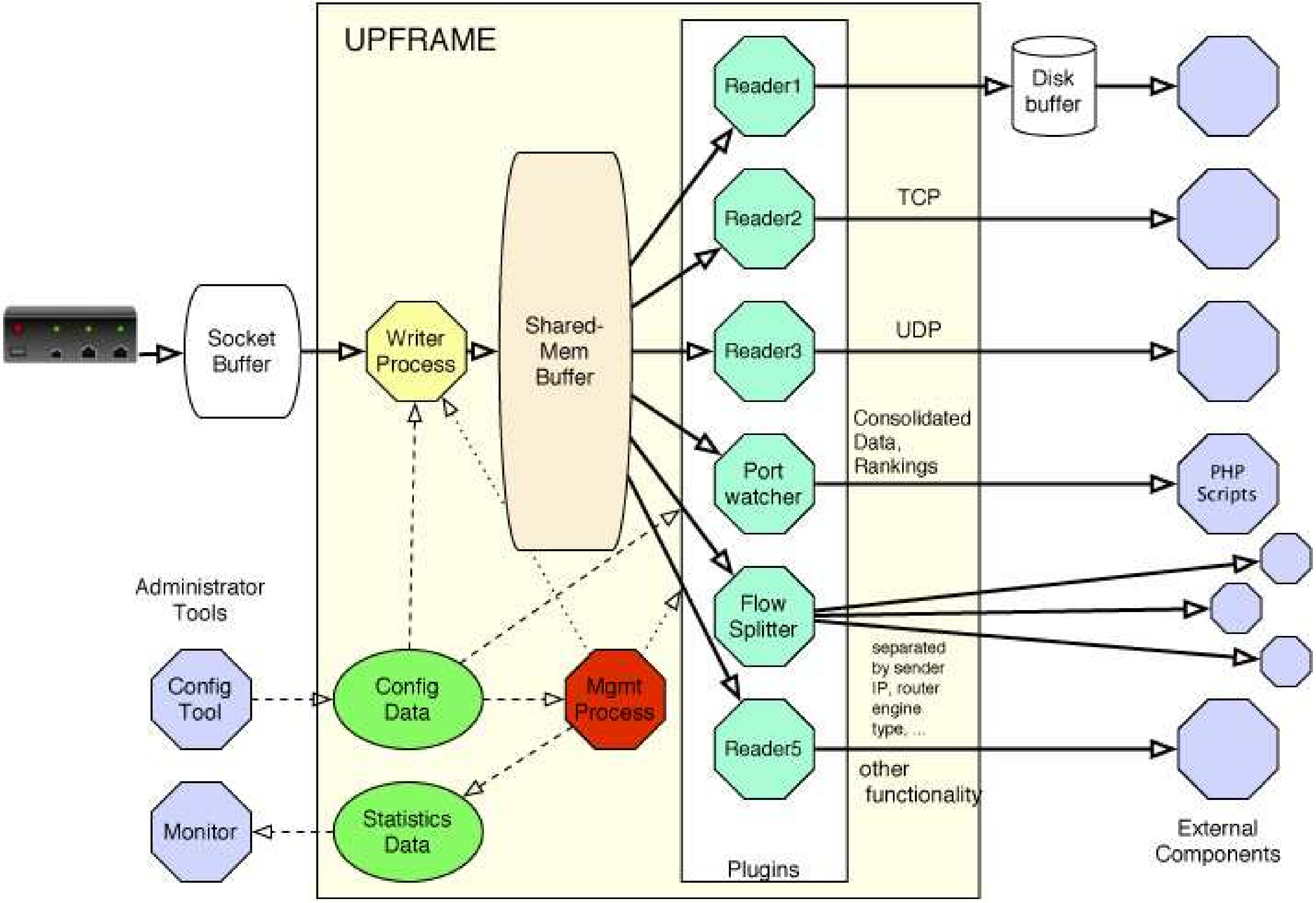
- Rackmount vs. ”normal”
- Cooling / Power needs planning
- Gigabit Ethernet "star" topology is nice
- KVM not for all nodes needed
- FAI (Fully Automatic Installation) for installation
- Local Debian mirror
  $\Rightarrow$ 10 Min for complete reinstallation
- No global connectivity for the nodes
- Private addresses for the nodes

UPFrame

`http://www.tik.ee.ethz.ch/~ddosvax/upframe/`

- UDP plugin framework
- E.g. for online analysis of NetFlow data
- Can be used as traffic-shaper
- Robust: For experimental plugins

UPFrame Structure



Summary for Data and Infrastructure

- SWITCH is large enough and small enough
- No special hardware / software needed for capturing
- Long-term storage is unproblematic
- Linux can be used in the whole infrastructure
- Online processing is more difficult
- Simplicity and Reliability are the main issues
- ...

The DDoSVax Dataset

- NetFlow v5 (converted from V7 by SWITCH)
- About 60.000.000 flows/hour
- Weekday: About 200k internal and 800k external IPs
- Unsampled
- Stored in full since March 2003

Flow Data Analysis by SWITCH

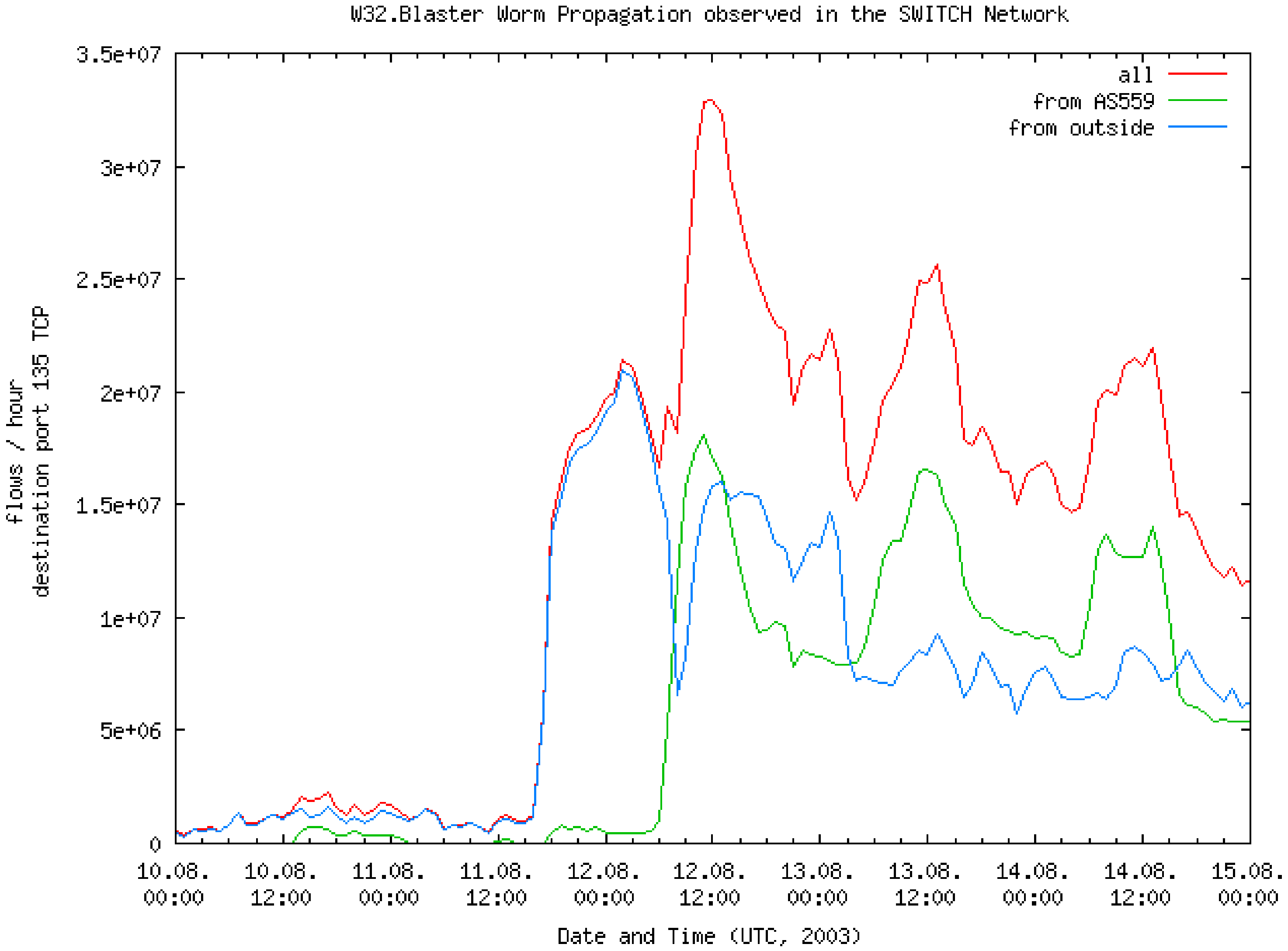
SWITCH-CERT: Short-term forensics (reduced)

- Single fast computer with hardware RAID-5
- No compression
- Sorted into minute (?) intervals
- Fast search with regular expressions
- Several weeks online
- No (?) long term storage

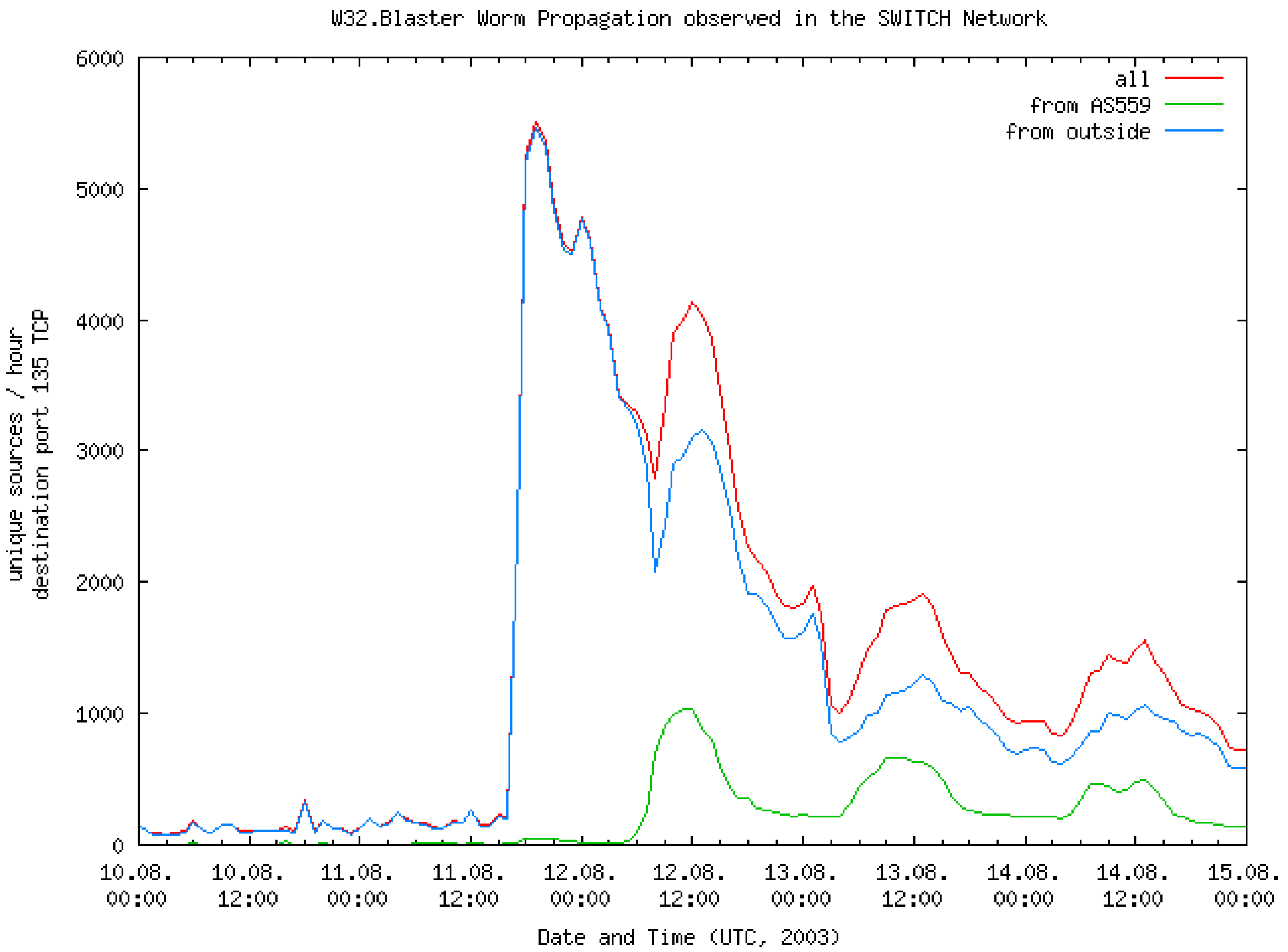
DDoSVax Offline Analysis

- E.g. for network/email worms
- Customised tools for some analyses
  - Single hour / prototyping: netflow_to_text and Perl
  - Days...weeks: From C-template
- Also other things: P2P, IRC, ...

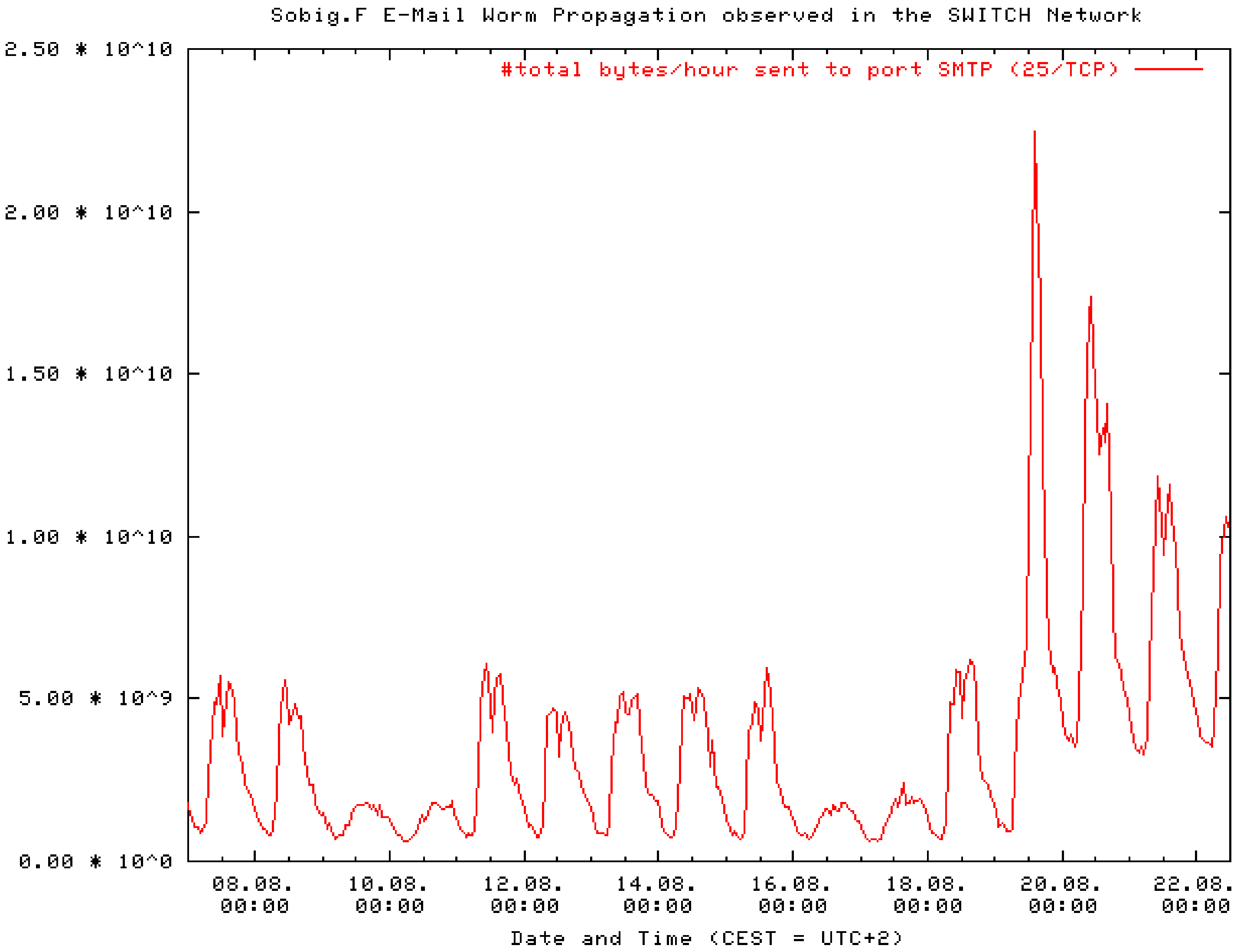
Example: Blaster - Flows



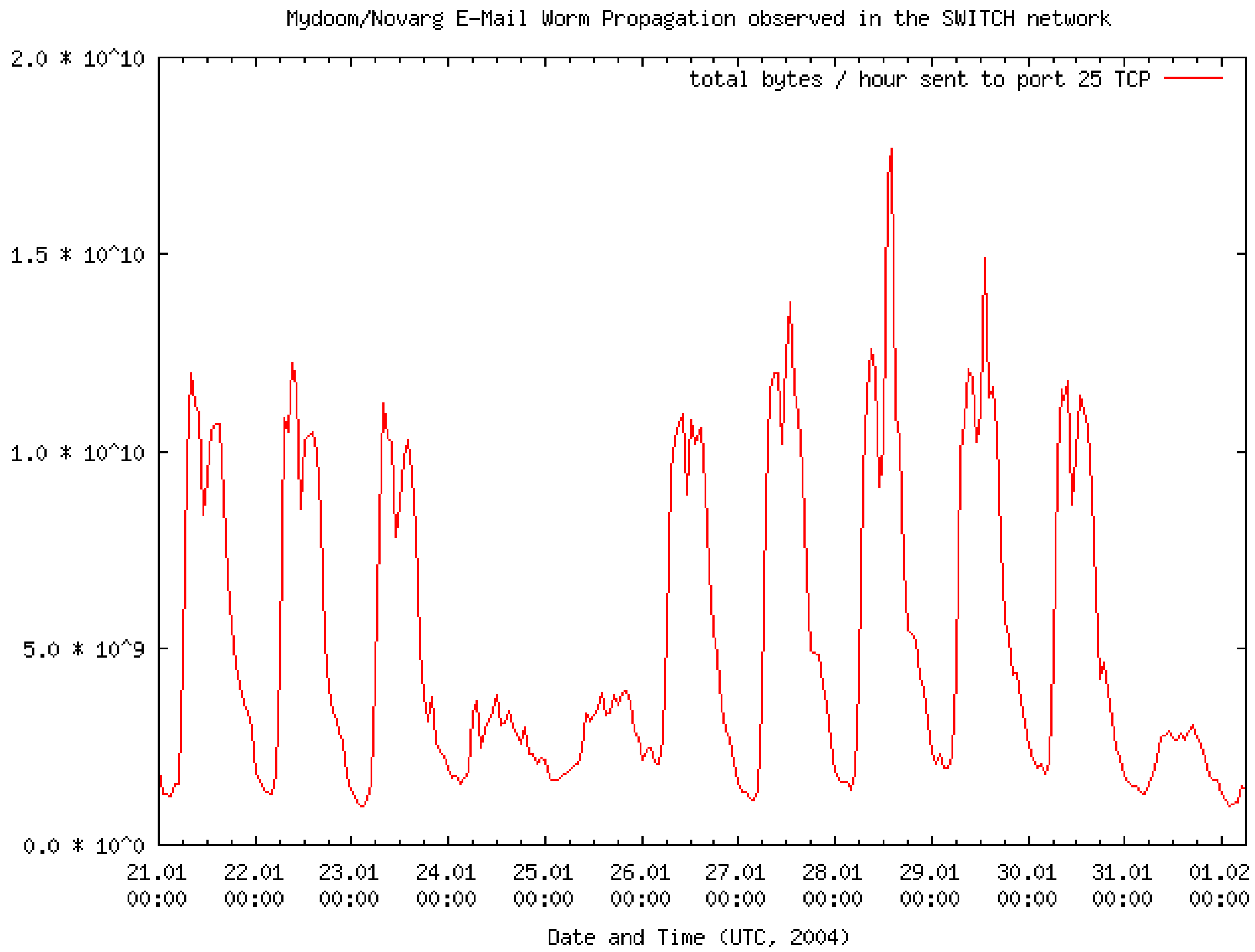
Example: Blaster - Unique Sources



Example: Sobig



Example: MyDoom



Traffic Amount vs. Unique Sources

Traffic Amount:

- Easy to do
- Works reasonably well
- Sensitive to data generation problems
- Sensitive to observed network

Unique Sources:

- More complicated, more robust
- Weakly dependent on observed network
- Allows to get global picture

Analysis-tools: Scripting

”netflow_to_text”

- Takes one data file, outputs one line
- Well suited as ”grep”/Perl input

Example:

```
TCP pr 111.131.210.8 si 1111.136.200.121
di 1264 sp 135 dp 48 le 1 pk
12:59:51.965 st 12:59:51.965 en 0.000 du
```

Analysis-tools: C

”Iterator template”

- Iterates over all records in a set of files
- Preprocesses timestamps, etc.
- Reading of input files encapsulated

Performance Issues

- 5-10 minutes / hour of data bunzip2
- I/O limit at 10 cluster nodes reading from one NFS partition
- Memory limitations

5: Student Theses

`http://www.tik.ee.ethz.ch/~ddosvax/sada/`

Open topics for SA/DA/MA theses:

- Generic: Attack and Worm Outbreak Online Detection
- Visibility of World of Warcraft and Half-Life 2 traffic in NetFlow data
- Generic: Attack Detection and/or Signature Generation for Honeypots (Contact: B. Tellenbach, <betellen@tik.ee.ethz.ch>)
- ...

Topics proposed by students are welcome!