

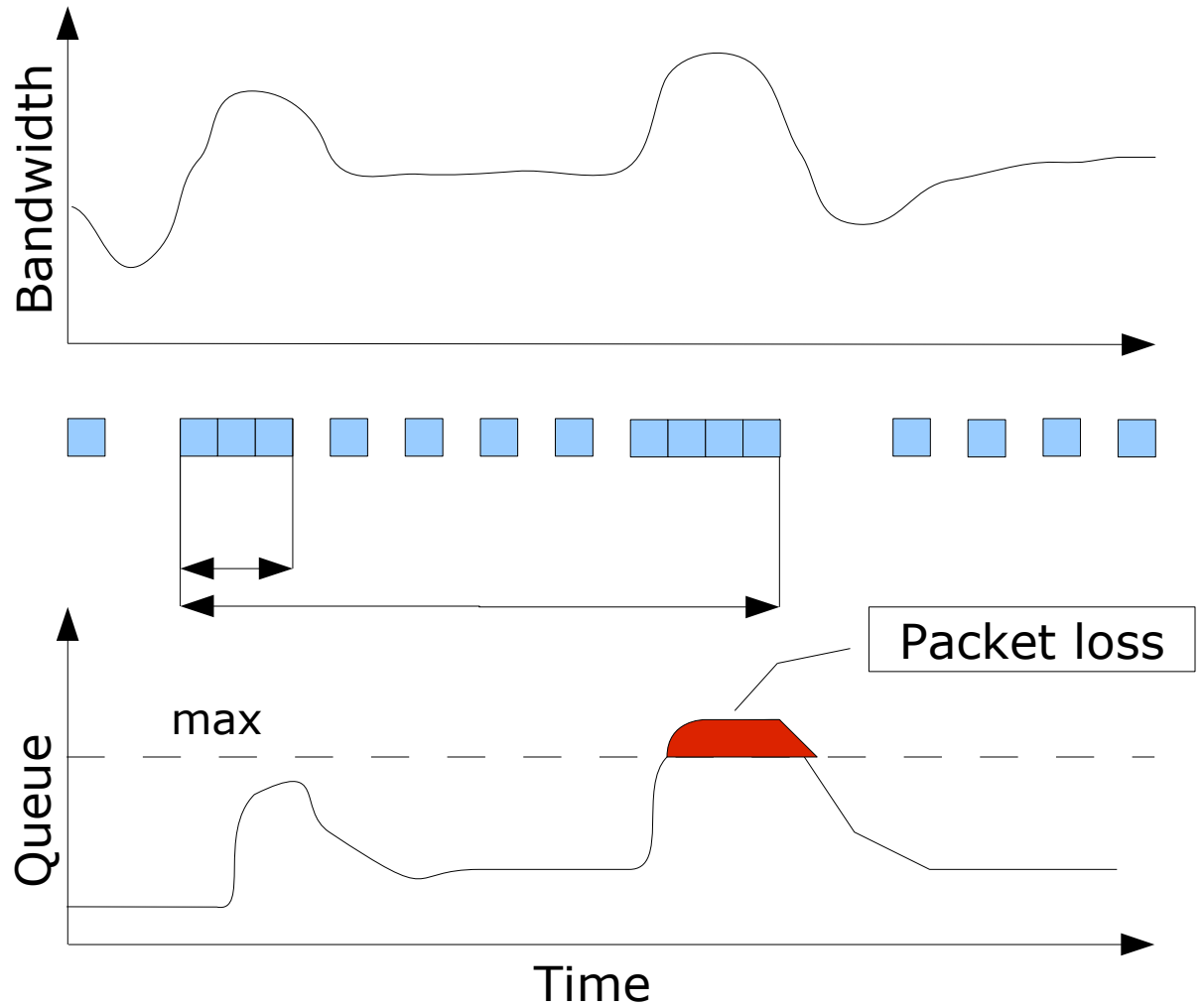
# ***Quantification of traffic burstiness with MAPI middleware***

***Sven Ubik, Aleš Friedl, Stanislav Hotmar***  
**CESNET**

# Traffic Burstiness

## Burstiness impact

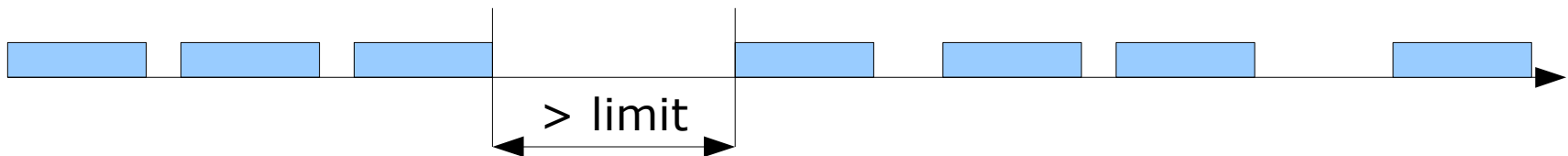
- Queue overflow
- Slow connection startup
- Delay
- Packet loss
- Low reliability
- Degraded QoS



- **Definitions**
  - Burst, traffic burstiness
- **Classification methods**
- **Implementation**
  - Architecture
  - Usage
  - User interface
- **Conclusion**
  - Future directions

## Burst

- Sequence of consecutive packets with inter-packet gaps less than specified limit



## Inter-packet gap (Interarrival time)

- From the last byte of the first packet (CRC)
- To the first byte of the subsequent packet (the Ethernet header)
- Minimal: 20 bytes ~ 16 ns on 10 Gb/s Ethernet

## **Burstiness monitoring** (*we focus on*)

- Passive, non-intrusive
- Continuous
- Focused on short scale
- Several types of classification

## **Classification**

- Histogram of bursts, bursts are classified by their size in bytes
- For each class
  - Number of bursts
  - Number of packets
  - Number of bytes
- Histogram of inter-burst gaps
- 256 classes (hardware implementation)
- Optional number of classes (software implementation)

## Histogram

- Minimal and maximal size
- Resolution (step)

Category Burst or gap size in bytes

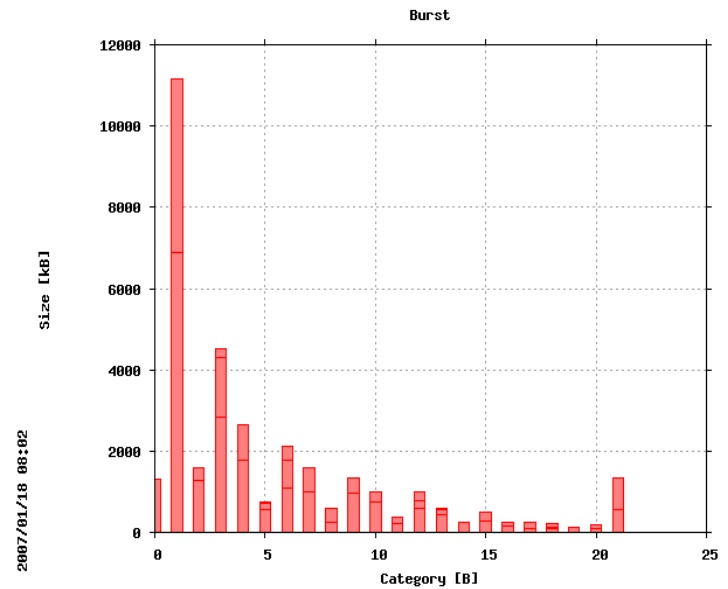
0 <0, min)

1 <min, min + step)

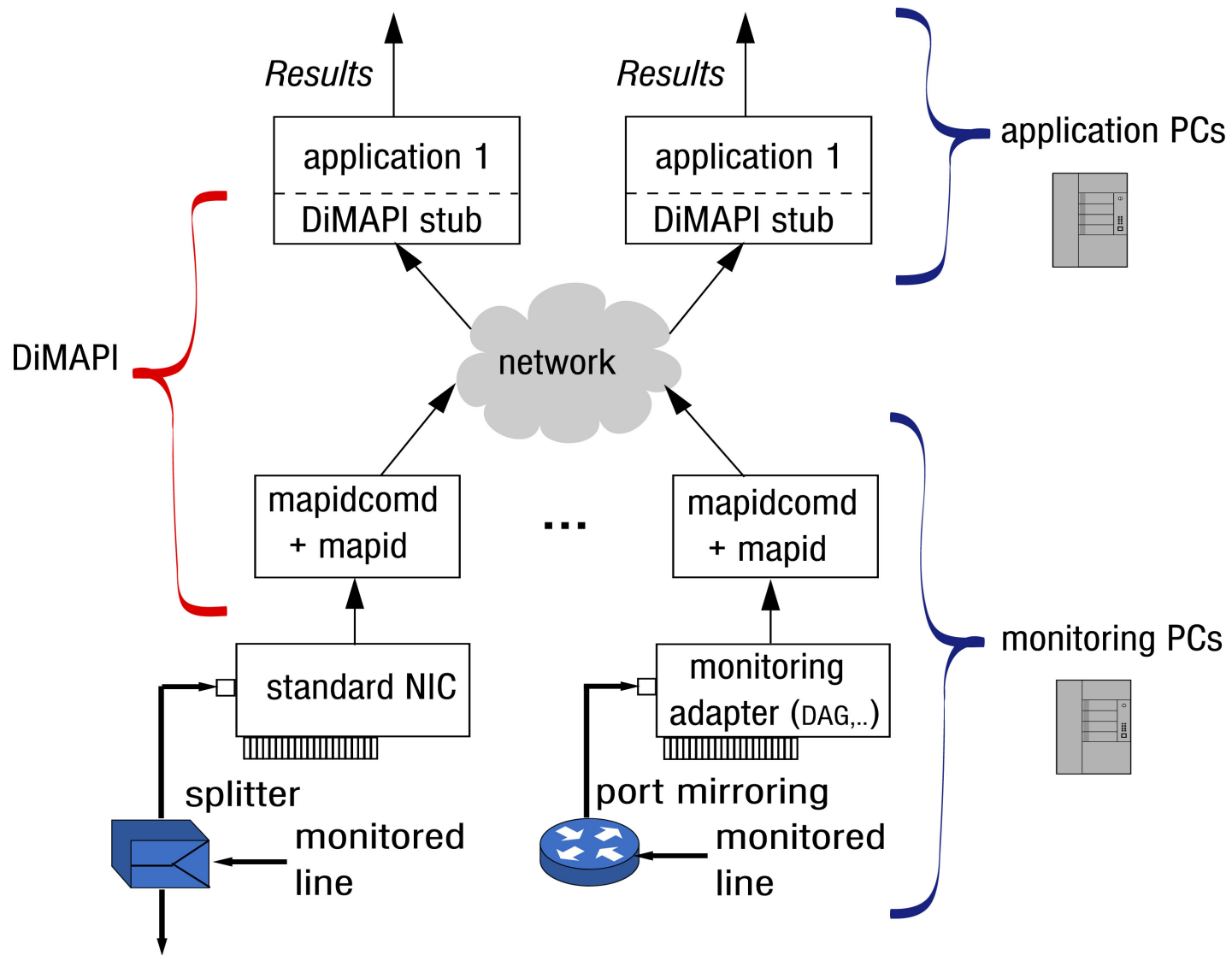
...

254 <max - step, max)

255 >= max



# MAPI Architecture



## **MAPI**

- A multi-user programming interface designed to simplify the development of network monitoring software
- Main abstraction which provides is network flow
- Applications can specify what flow statistics they are interested in by applying functions to provided flows
- Functions can be hardware or software implemented
- We added support for traffic burstiness classification

## **DiMAPI**

- Transparent access to remote monitoring station

## **Mapi function BURST**

- Hardware implementation
  - Combo6 cards
- Software implementation
  - MAPI stdlib

## Mapi function BURST

- MAPI provides transparent access to BURST function, once this function is applied to *flow*, either hardware (if available) or software implementation is used

## Software implementation

- Implemented as a function in standard MAPI library
- Works with any monitoring card supported by MAPI (DAG, NIC, ...)
- 149 – 172 instructions per packet (measured with PAPI\*)

## Hardware implementation

- STU-BURST module
- 256 categories in histogram
- 64 bit counters

*\*) Performance Application Programming Interface*

## Burst function usage

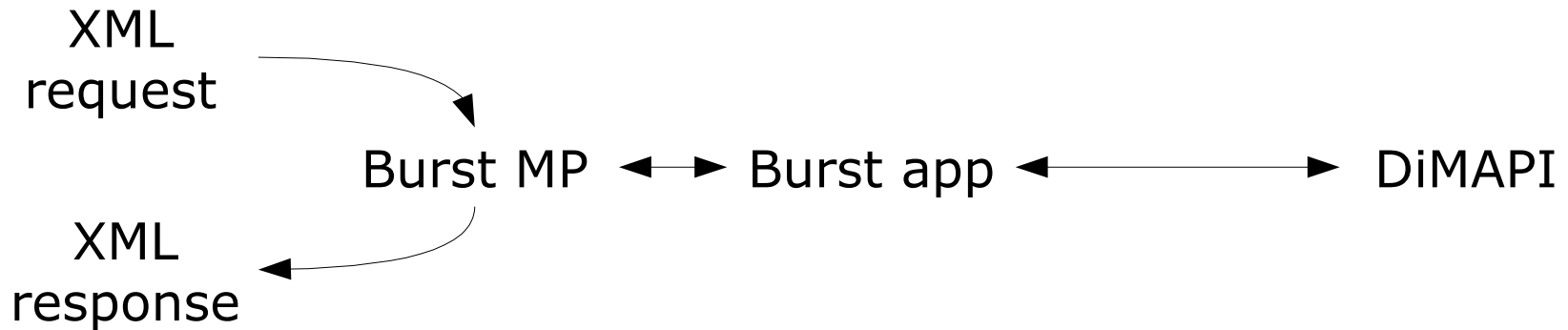
```
// create a flow
fd = mapi_create_flow("hostname:/dev/dag0");

// apply a burst function
fid = mapi_apply_function(fd, "BURST", min, max, step, iatime,
late, early, linkspeed);

while(1) {
    sleep(10);
    // read results (returns an array -- histogram)
    res = mapi_read_results(fd, fid);
}
```

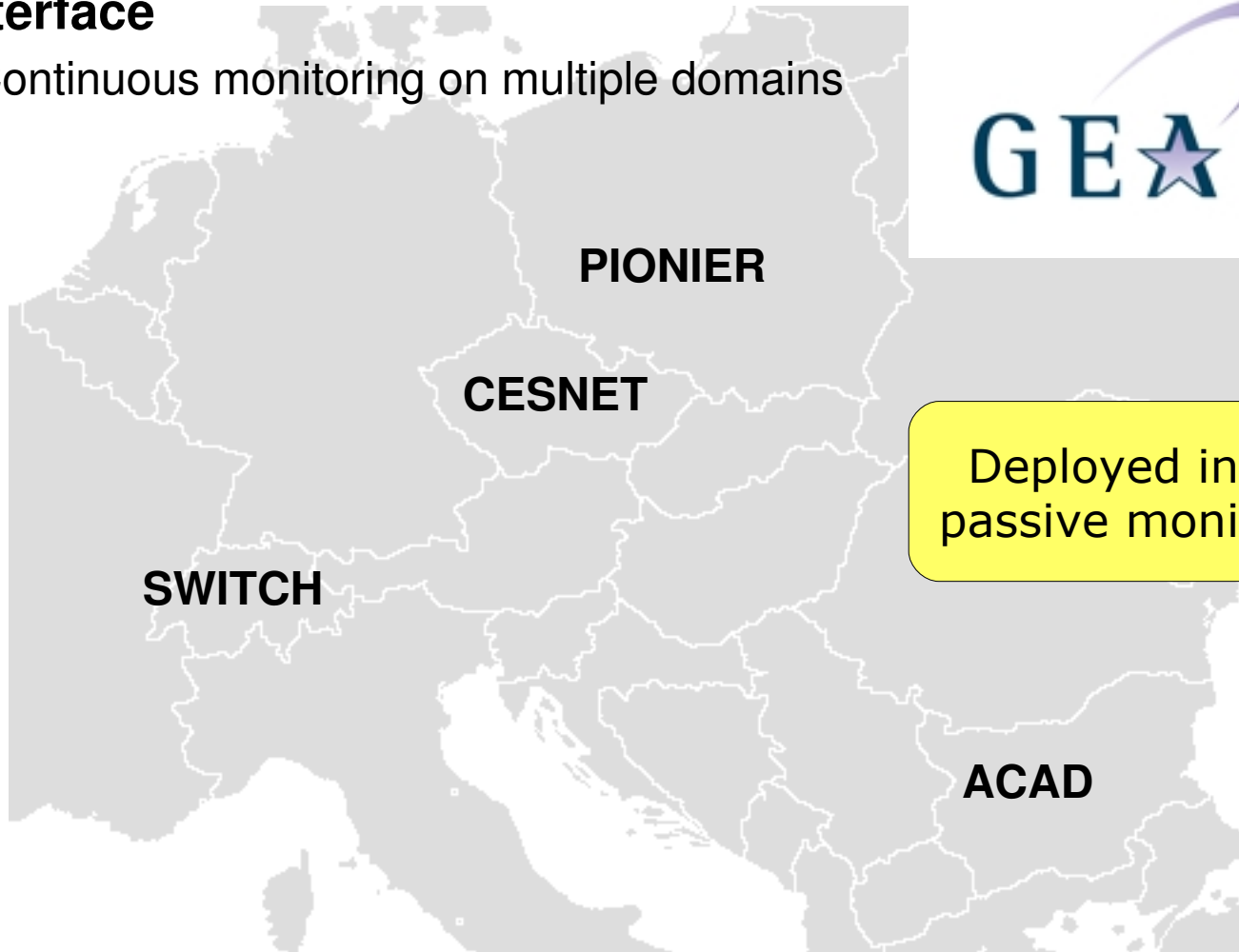
## Burst MP (Measure Point)

- Daemon running as PerfSONAR service
- Provides an interface to BURST function
- XML request/response based



## Web interface

- Continuous monitoring on multiple domains



Deployed in GN2 SA3  
passive monitoring pilot

## BURST - Monitoring of traffic dynamics



Step 1: Select monitored links:

PIONIER <- GN2 (pm1)	<input type="checkbox"/>
PIONIER -> GN2 (pm2)	<input type="checkbox"/>
SWITCH -> GN2 (pm1)	<input type="checkbox"/>
SWITCH <- GN2 (pm2)	<input type="checkbox"/>
ACAD <-> GN2 (pm1)	<input type="checkbox"/>
CESNET <-> GN2	<input type="checkbox"/>
all links	<input checked="" type="checkbox"/>

Step 2: Select time period and interval:

Time type:

Predefined  User defined start time  User defined end time

Predefined:

last 1 hour (interval 1 min)	<input checked="" type="checkbox"/>
last 10 hours (interval 10 min)	<input type="checkbox"/>
last 1 day (interval 1 hour)	<input type="checkbox"/>
last 1 week (interval 1 day)	<input type="checkbox"/>
last 1 month (interval 1 day)	<input type="checkbox"/>
last 1 year (interval 1 week)	<input type="checkbox"/>

User defined:

Start or end time:

07 35

Start or end date:

23 September 2008

Time length:

Minutes

# Web Interface (cont.)

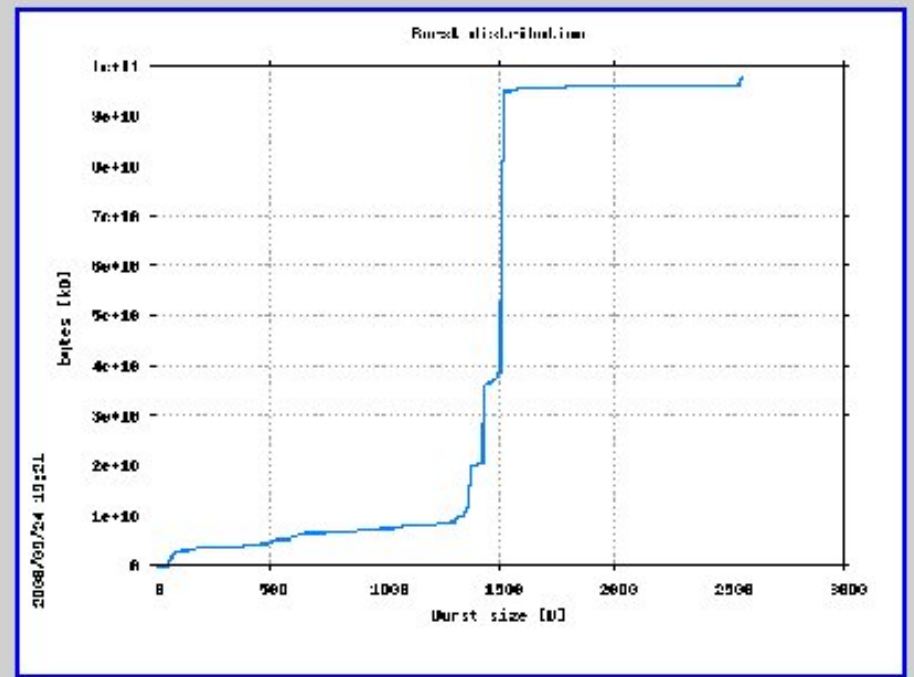
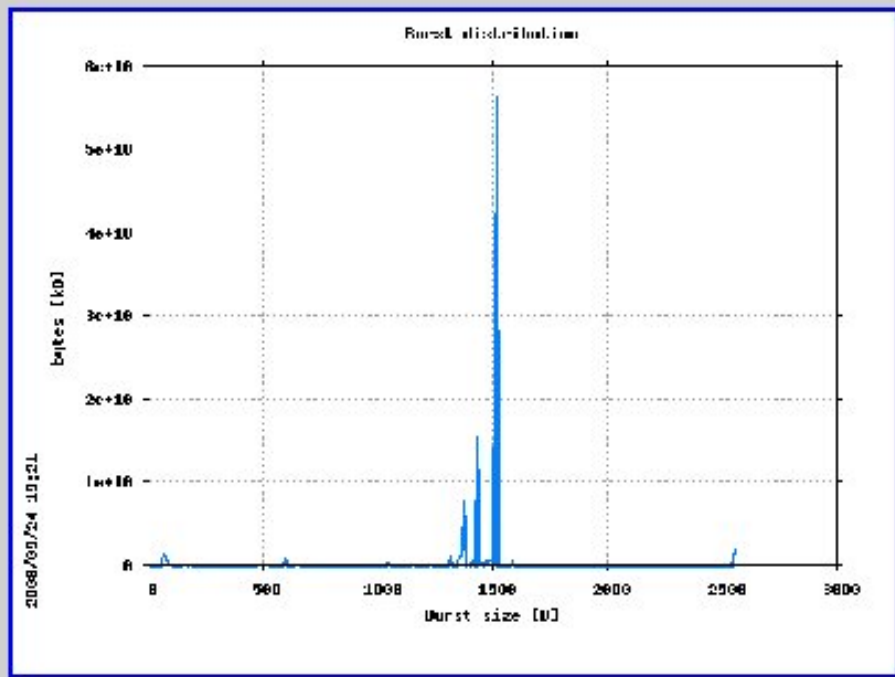
Step 3: Select characteristics:

Bytes	<input checked="" type="checkbox"/>
Packets	<input type="checkbox"/>
Bursts	<input type="checkbox"/>

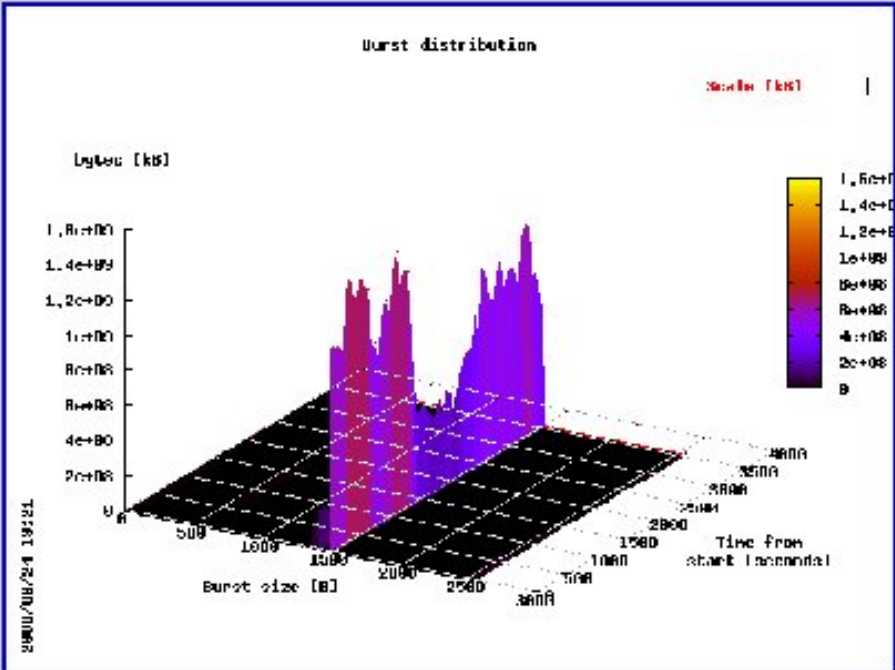
Step 4: Select type of graphs:

2D	<input checked="" type="checkbox"/>
2D cummulative	<input checked="" type="checkbox"/>
3D	<input checked="" type="checkbox"/>
3D cummulative	<input checked="" type="checkbox"/>
3D map	<input checked="" type="checkbox"/>
3D map cummulative	<input checked="" type="checkbox"/>

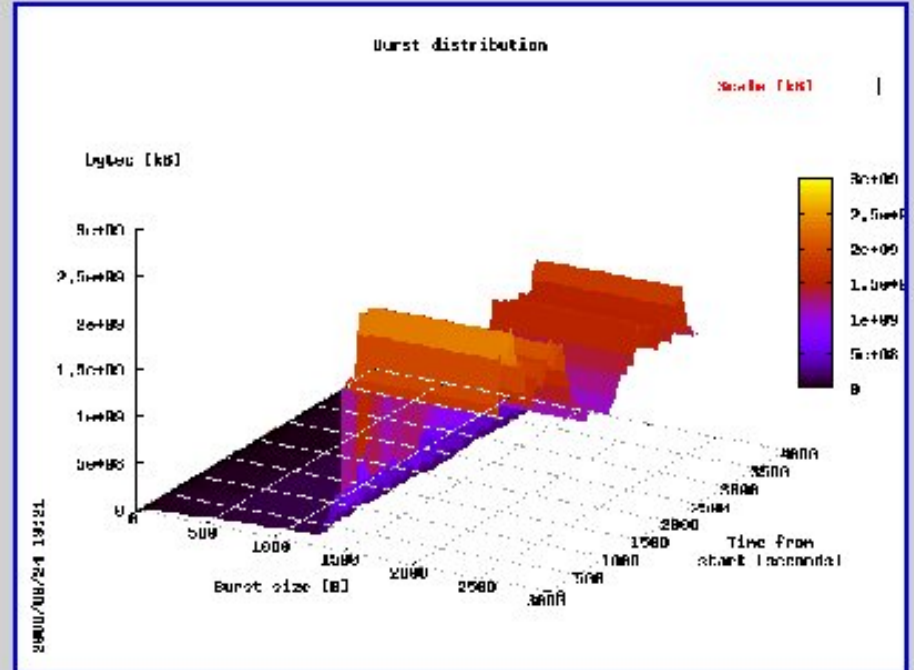
Generate graphs (at least one option must be selected in each step 1 - 4 to produce graphs)



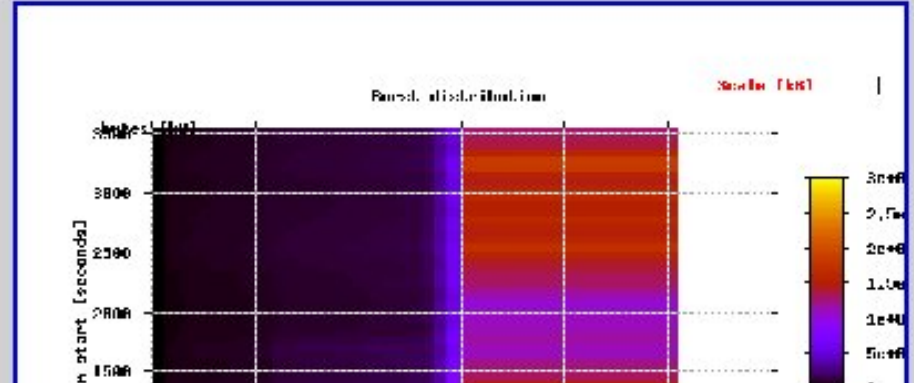
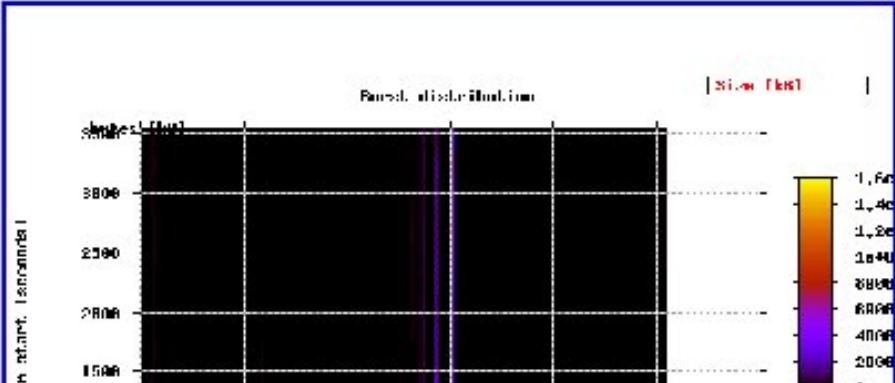
# Web Interface (cont.)



Monitored link: CESNET <-> GN2  
last 1 hour (interval 1 min)  
bytes, 3D



Monitored link: CESNET <-> GN2  
last 1 hour (interval 1 min)  
bytes, 3D cumulative



## Burst distribution histogram

- Explore correlation of inter-burst gaps distribution and burst distribution
- Explore effect of gap size limit on burst distribution
- Explore changes in time in burst/gap distribution
- Can we estimate some link limitation out of specific burst distribution?
  - For example, is it safe to add some additional flow (elastic / nonelastic)?
- Explore possibilities of detecting some specific traffic
  - network attack detection
  - anomaly detection

# Questions?