

# Status from the IPTV probe project

Bifrost Workshop 2010  
d.27/1-2010

by

Jesper Dangaard Brouer <jdb@comx.dk>

Master of Computer Science

ComX Networks A/S

The logo for ComX Networks A/S, featuring the word "comx" in a lowercase, sans-serif font. The letters "com" are in a dark grey color, and the letter "x" is in a bright orange color.

ComX Networks A/S

# Background

- Last Bifrost Workshop (18/3-2009)
  - Presentation about
    - issues with multicast drops
    - developed tools to detect drops
- What happened since last time?
  - Well...
    - ... I have played a lot with 10GbE ;-)
    - ... have completed milestone 1

# Milestones

## Milestone 1: Measure the drops

- Implemented in Wirehark
- Implemented in Kernel as iptables module

## • Milestone 2: Measure the bursts

- Plan to impl. RFC4445 definition of “Delay Factor”

## • Milestone 3: Smooth out the bursts

- Challenging due to variable bit-rate streams

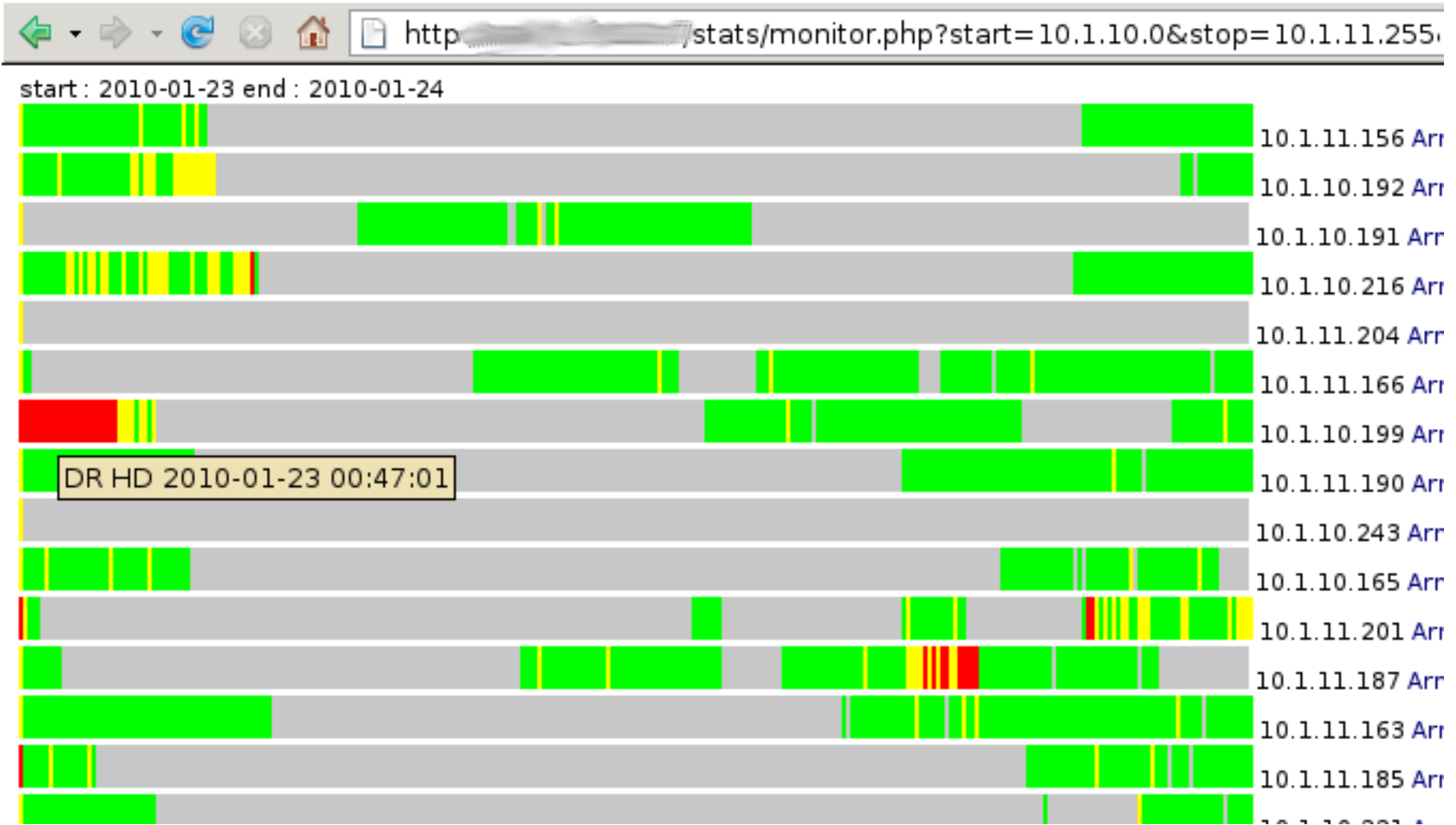
# Milestone 1: Detect Drops

- Kernel module “mp2t” finished
  - in production a long time
  - but, last week started, collecting data
    - and storing centrally
- Wireshark
  - Improved drops detection, since last time

# Trick: Use settop box as probes

- See what the customer sees
  - As many probes as customers
- Trick: The settop box runs Linux
  - local tool support asking for “sync errors”
  - install small bash script
    - periodically poll, and submit result back central
- It works!

# Settox boxes as probes



# Still need probes

- Still need probes, in the network
  - need to identify the network segment
  - introducing the packet drops
- Central logging
  - probes drop detection
    - status:
      - collector daemon is working
      - user interface still missing (in progress)

# Milestone 2: Wireshark

- Wireshark
  - Implemented decoding PCR clock
    - PCR tell us which speed
      - the streamer intended for this signal
      - thus, know the optimal inter-packet arrival time
    - patches not accepted :-(
      - No easy burstiness detect impl. yet
        - Advanced use of “IO graph” can show it



# Milestone 2: Kernel module

- Next stage is burstiness
  - First decode PCR clock
    - optimal intergap for the stream
    - comparing, with network packet intergaps

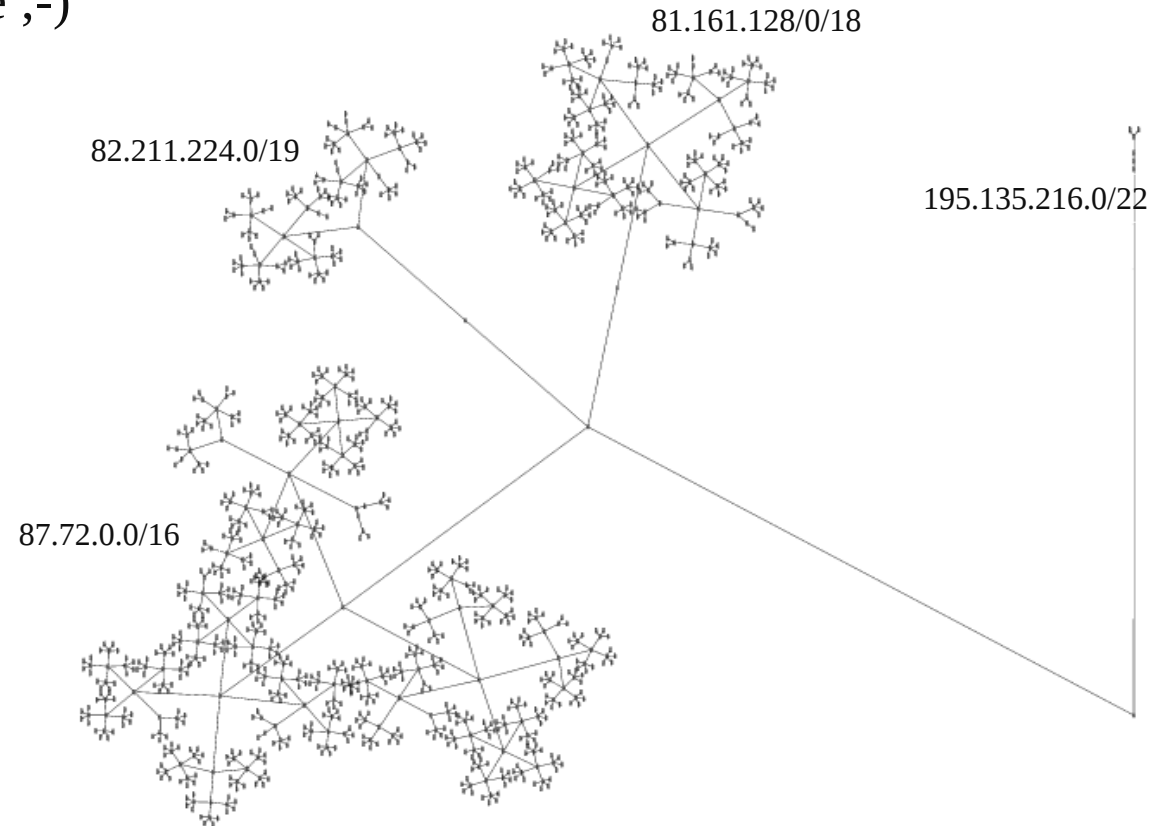
# Milestone 3: The shaper

- To early to say if it ever happens?!
- Need experience from
  - real-life precision of the PCR signal quality,
  - hoping, use PCR clock
  - reclock signal, by delaying one PCR clock

# The End

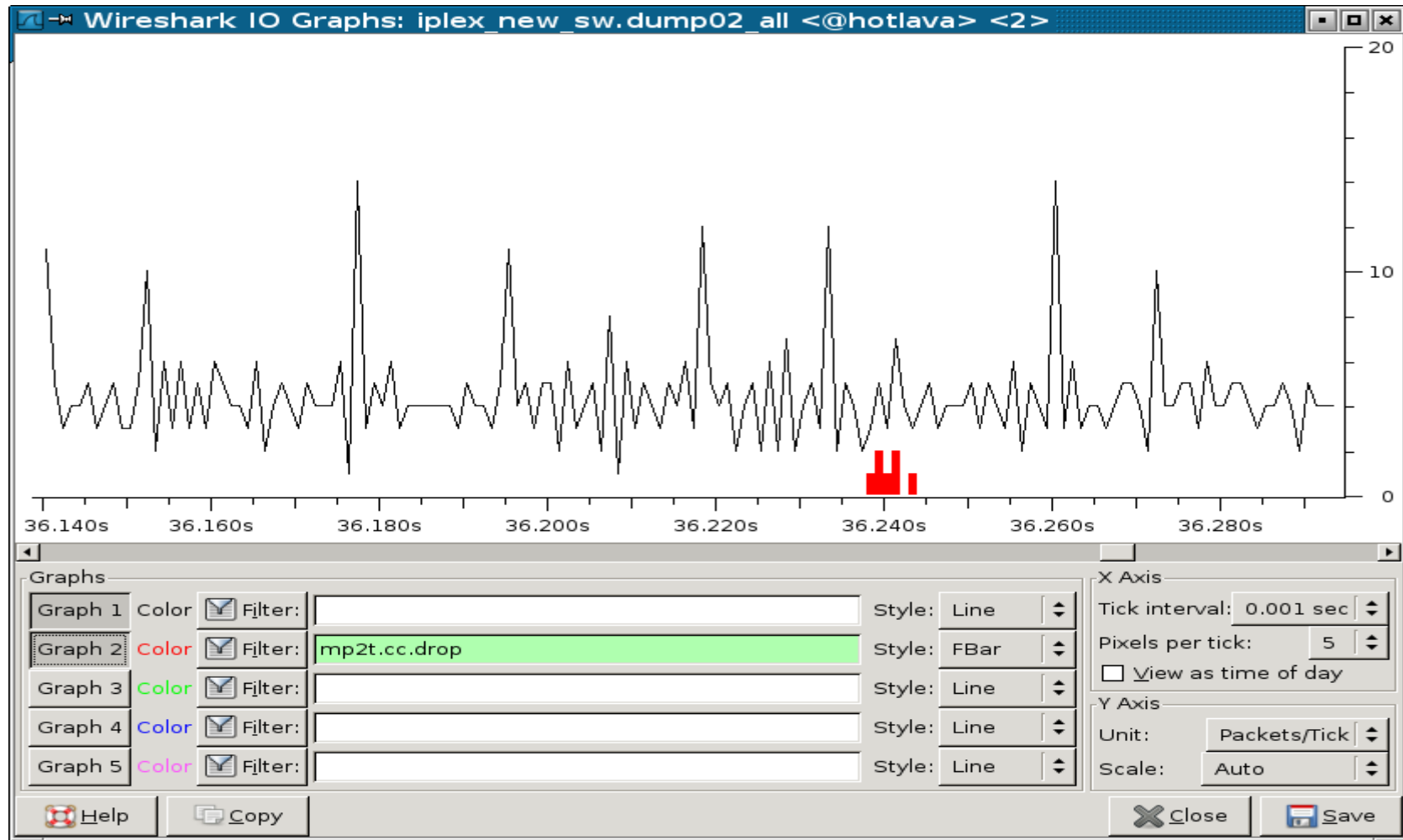
This was the last talk...

... thank you for staying a wake ;-)



# Wireshark IO-graph

- Menu: Statistics → IO Graph → Tick Interval 0.001



# Who am I

- Name: Jesper Dangaard Brouer
  - Edu: Computer Science for Uni. Copenhagen
    - Focus on Network, Dist. sys and OS
  - Linux user since 1996, professional since 1998
    - Sysadm, Developer, Embedded
  - OpenSource projects
    - Author of
      - ADSL-optimizer
      - CPAN IPTables::libiptc
    - Patches accepted into
      - Linux kernel, iproute2, iptables and Wireshark