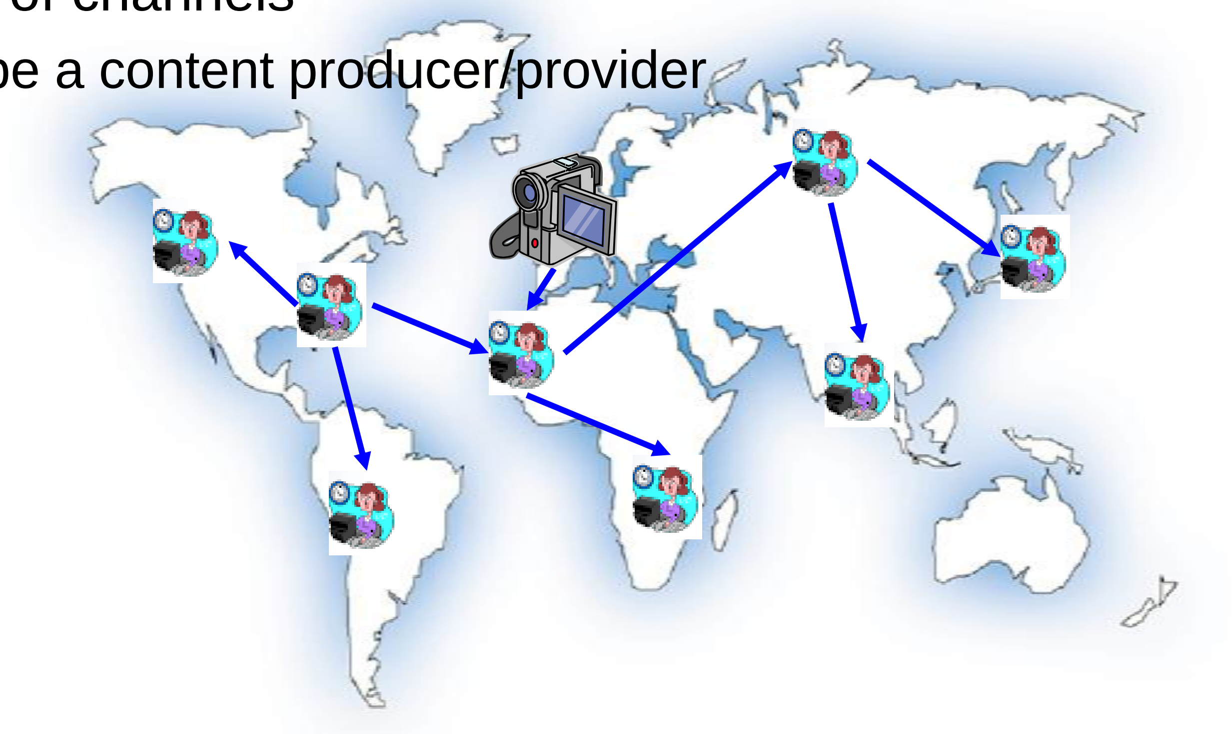
Architecture of a Network-Aware P2P-TV Application: the NAPA-WINE Approach

Slough, 11th November 2011

Emilio Leonardi
Politecnico di Torino

Internet Video Streaming

- Enable video distribution from **anywhere**, to **any number** of people **anywhere** in the world
- Unlimited number of channels
  - Everyone can be a content producer/provider



CDN vs P2P

- Content Delivery Networks
  - \- resources (costs) demanded to servers scale linearly with the number of users
  - \+ fully controllable by the content provider and Internet provider
- P2P systems (Peer Assisted)
  - \+ resources (costs) demanded to servers, potentially independent from the system scale
  - \- requires high bandwidth access
  - \- much more difficult to control

Open issues in peer assisted systems

- peer authentication (access control pricing)
- incentives to cooperation
- robustness against attacks (s.a pollution)
- localization of traffic

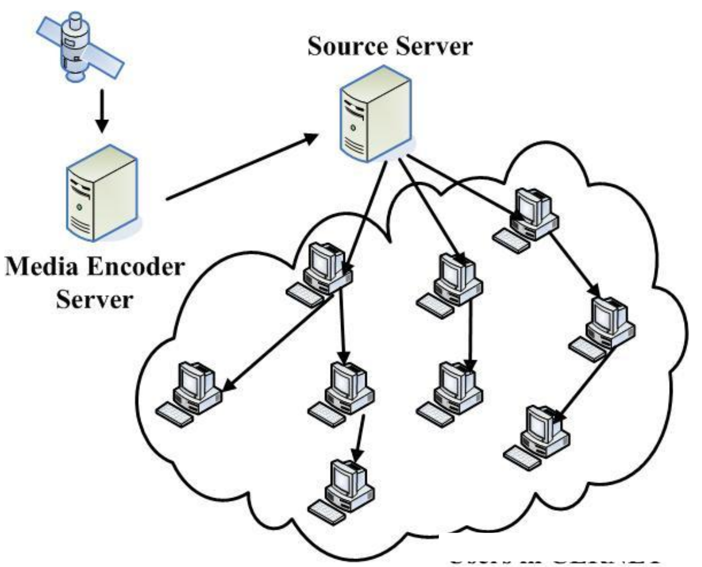
In a nutshell how to make peer assisted systems, secure, fully controllable, and network gently?

NAPA-WINE Project Objectives

- Definition and implementation of a network aware P2P-TV application that is able to minimize the impact on the transport network
  - Design of a distributed monitoring tool to be integrated within the application.
  - Design of algorithms for the control of cooperative P2P-TV systems
  - Characterization of P2P-TV traffic

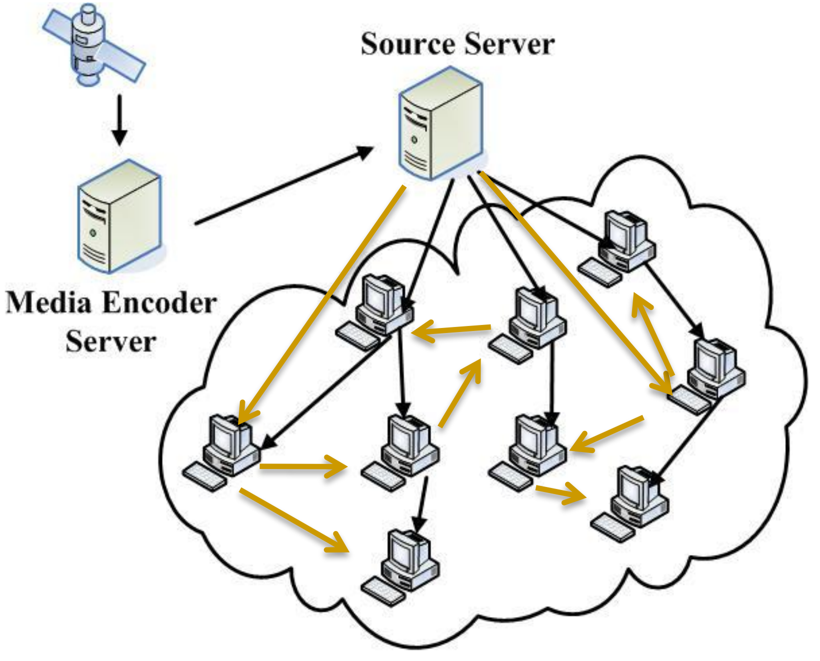
Tree based P2P-TV systems

- Different peers are organized in a tree structure routed at the source
- The content is distributed along the tree



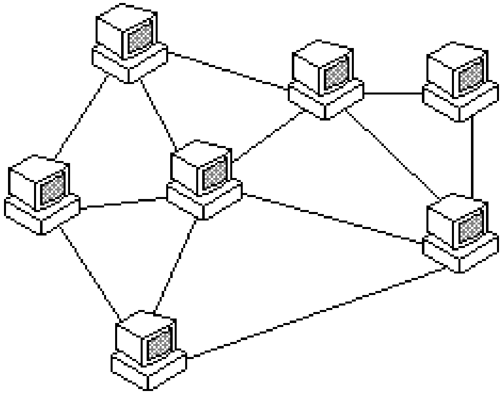
Multi tree P2P-TV systems

- The source adopts a multi-description encoder
- Each description is distributed on a different tree



Unstructured Systems

- peers are arranged according to a generic highly connected network
- the stream is subdivided in portions called *chunks*
- each chunk is distributed along a possibly different spanning tree (SP)
- SP are selected using simple random fully distributed algorithms



Scheduling algorithm at nodes

- Chunks are distributed through the network using a swarm like (epidemic) approach
  - as soon as, a peer obtains a new chunk c, it will offer c to its neighbors
  - Chunks are not propagated perfectly in order; however chunk timing is critical (due to the application requirements)
  - Each chunk has a deadline after which it is not useful (this deadline is related to the play-out buffer)

Pros and Cons of Unstructured Architectures

+ fully resilient to churning

+ no need of centralized control

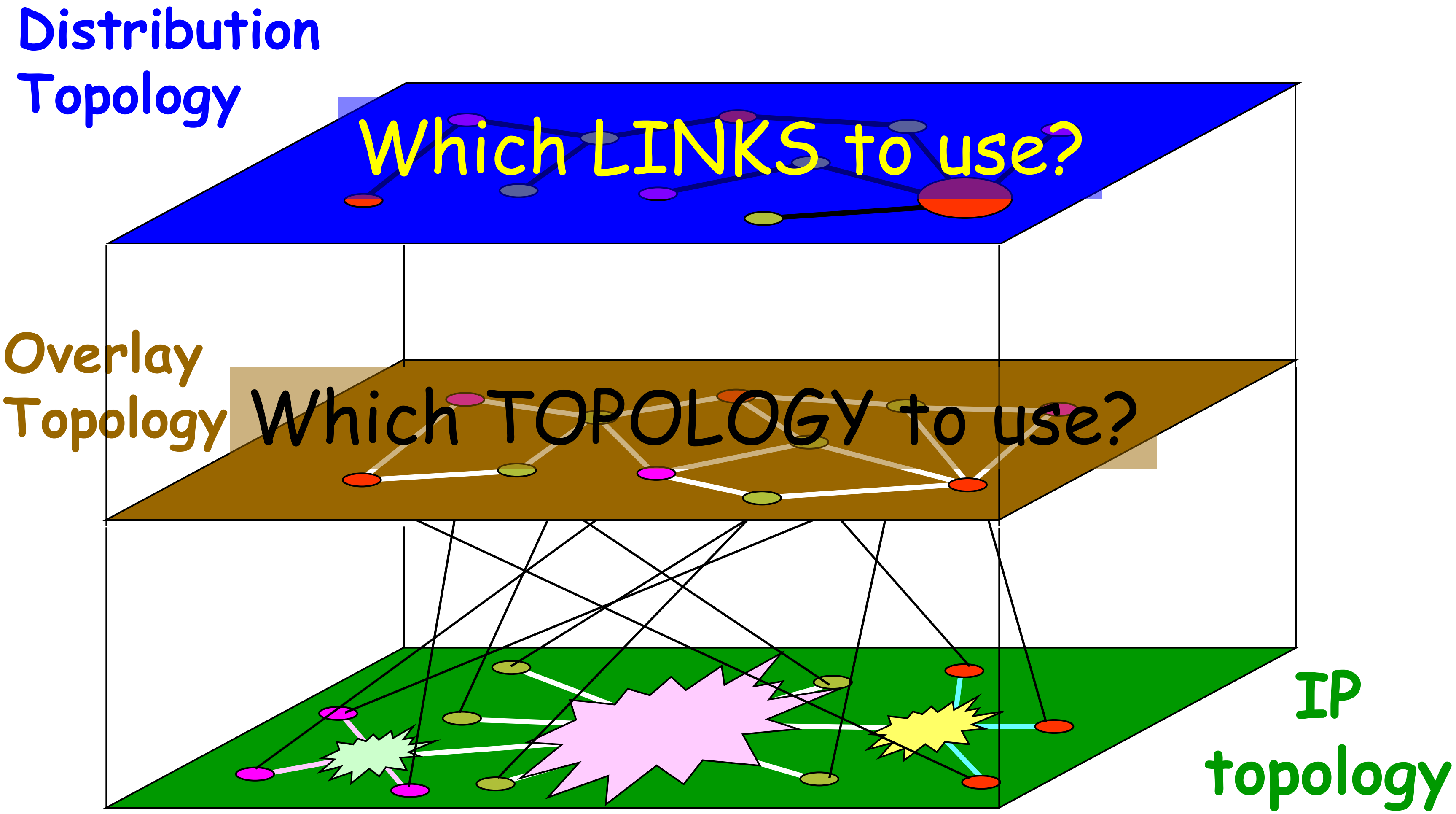
+ efficient to exploit the bandwidth

- larger delays in delivering information

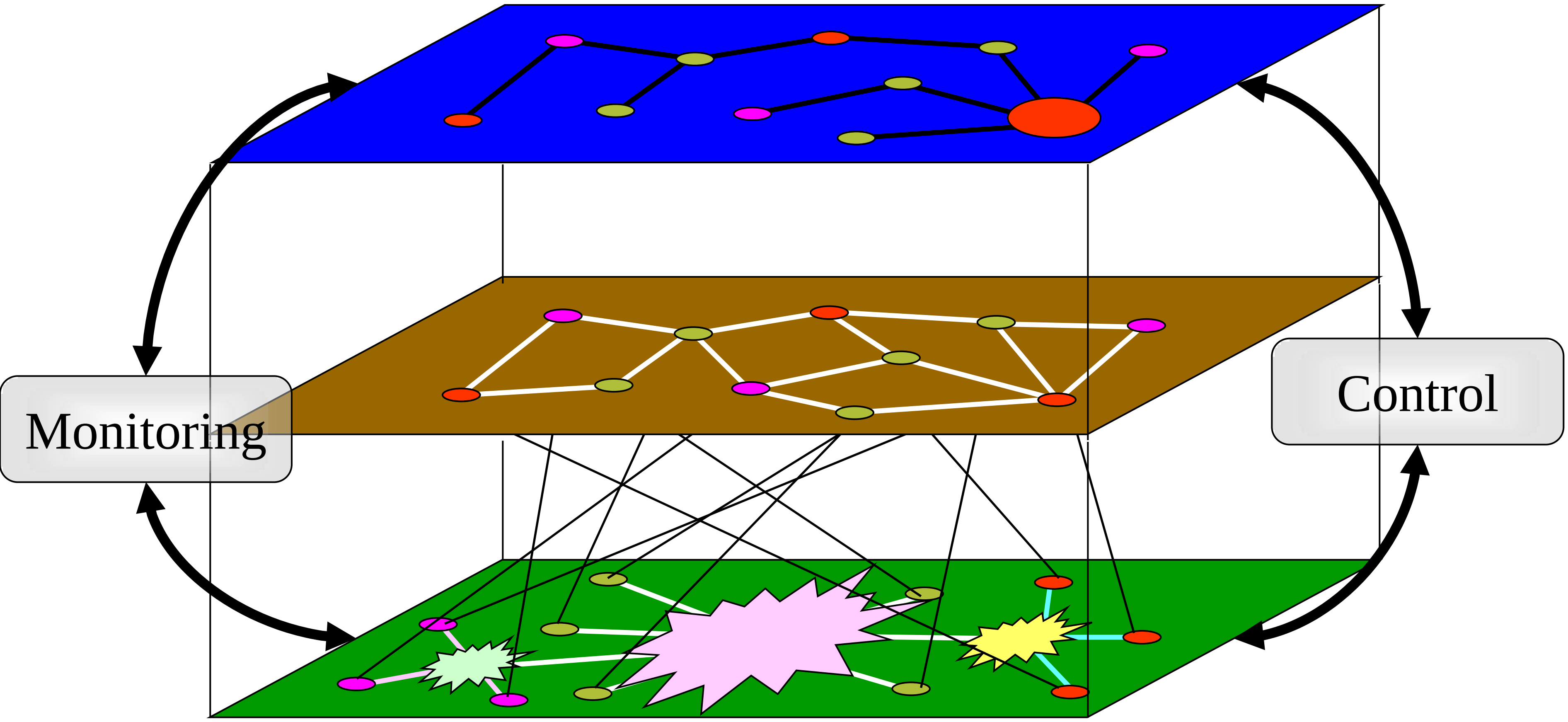
- very difficult to control and predict the performance

NAPA-WINE application is unstructured

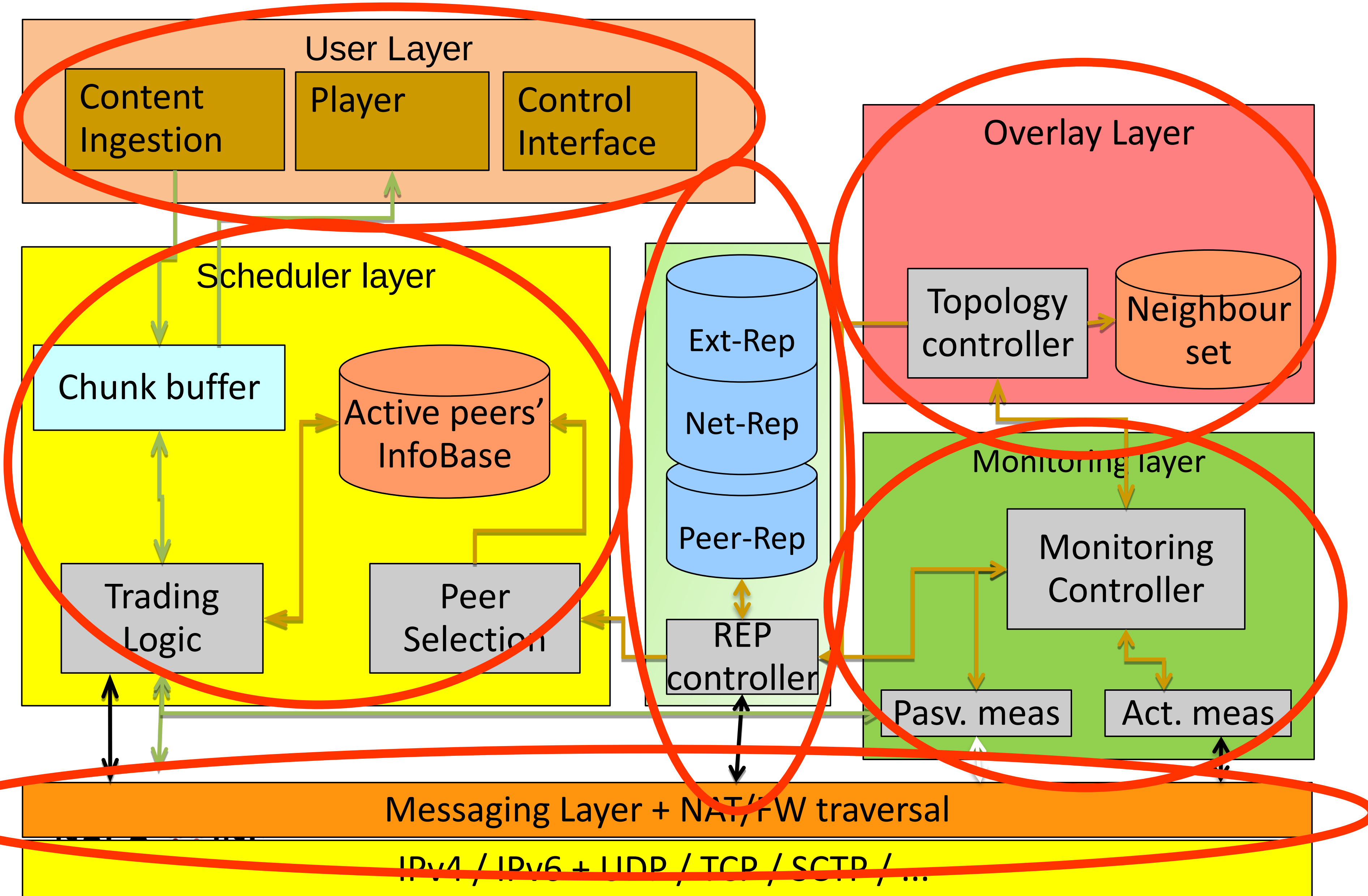
P2P-TV Simple View



P2P-TV: NAPA-WINE Approach



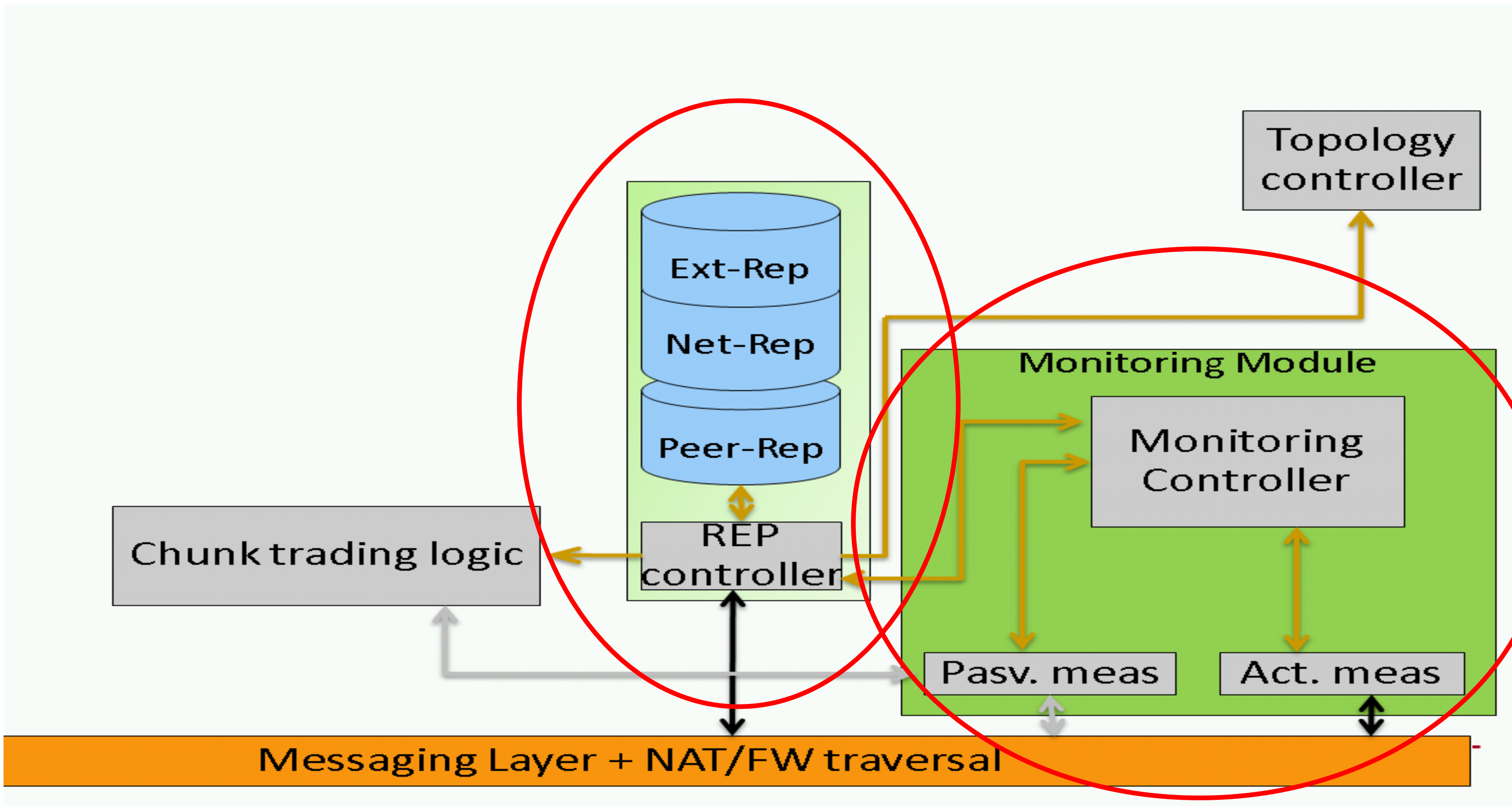
Overview of the architecture



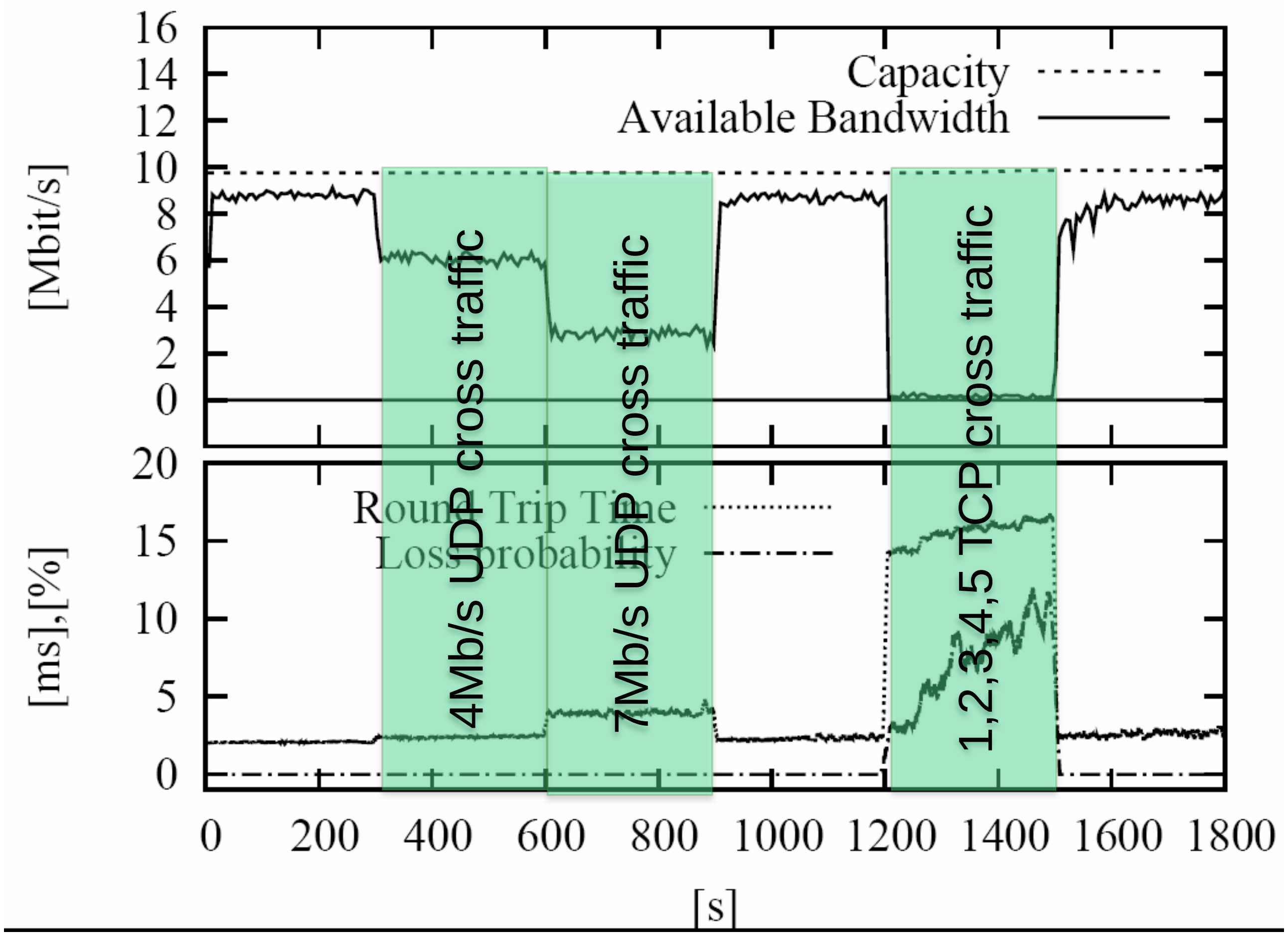
Network Monitoring Module

- A number of measurement functions are available
  - RTT, Hop count, capacity, loss rate,
  - Capacity and available bandwidth
  - pluggable measurement functions can be added
  - extensible framework

Monitoring Platform



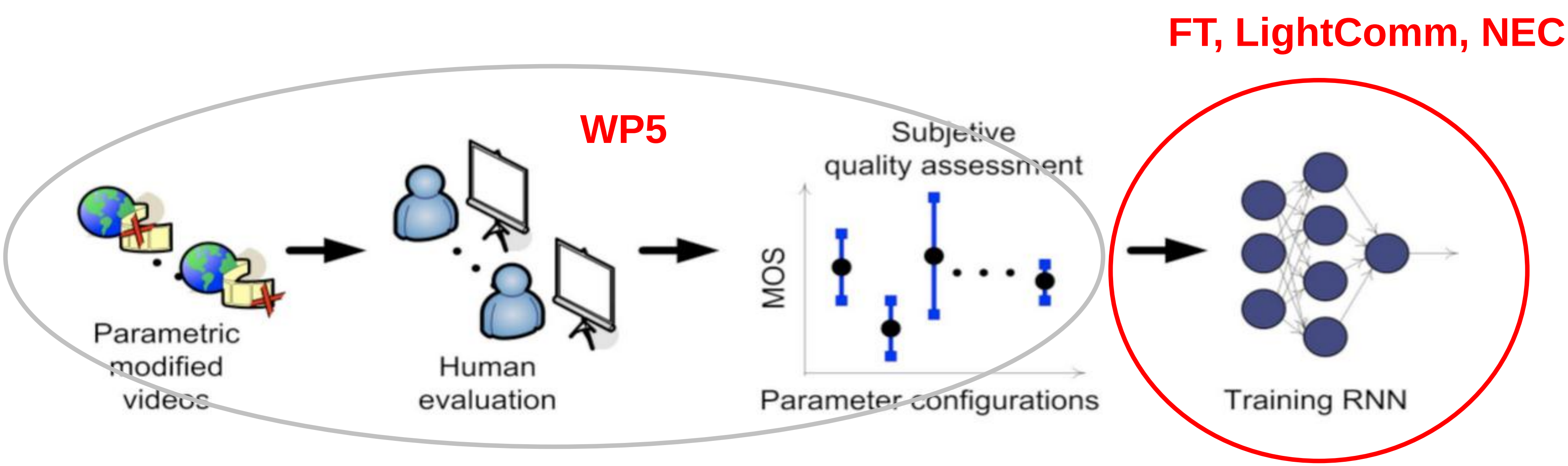
Simple example of Capacity and available bandwidth measurement



NAPA WINE

On-line QoE estimation

- A QoE estimator has been developed:
  - It estimates online the quality of the audio/video on the base of losss patterns and potentially other parameters (trained database)
  - based on random neural network (Gelenbe, Rubino)

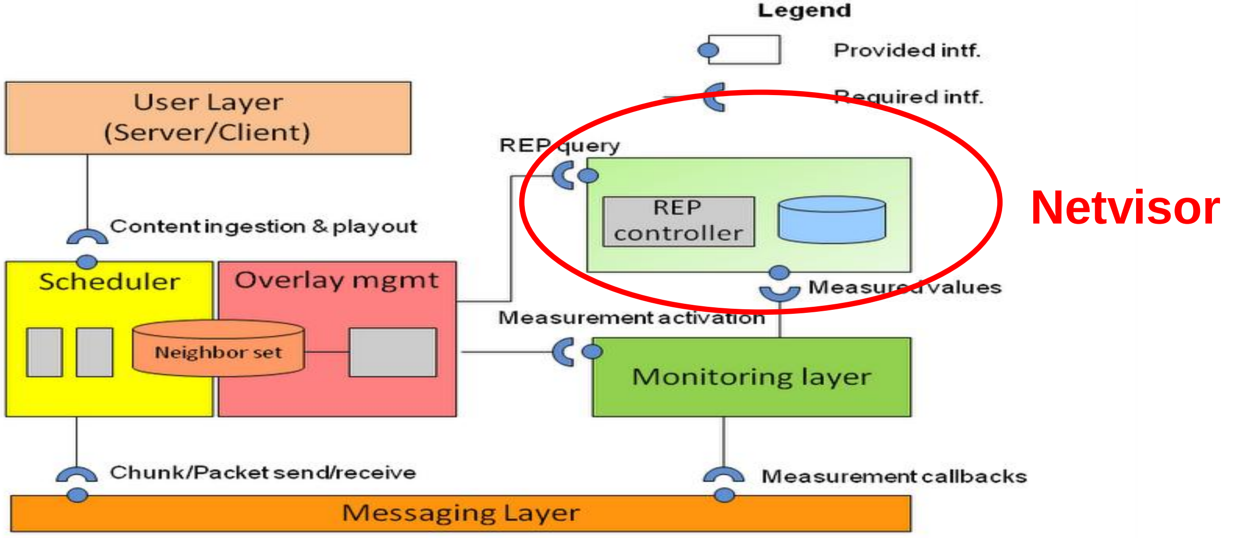


Conclusive experience

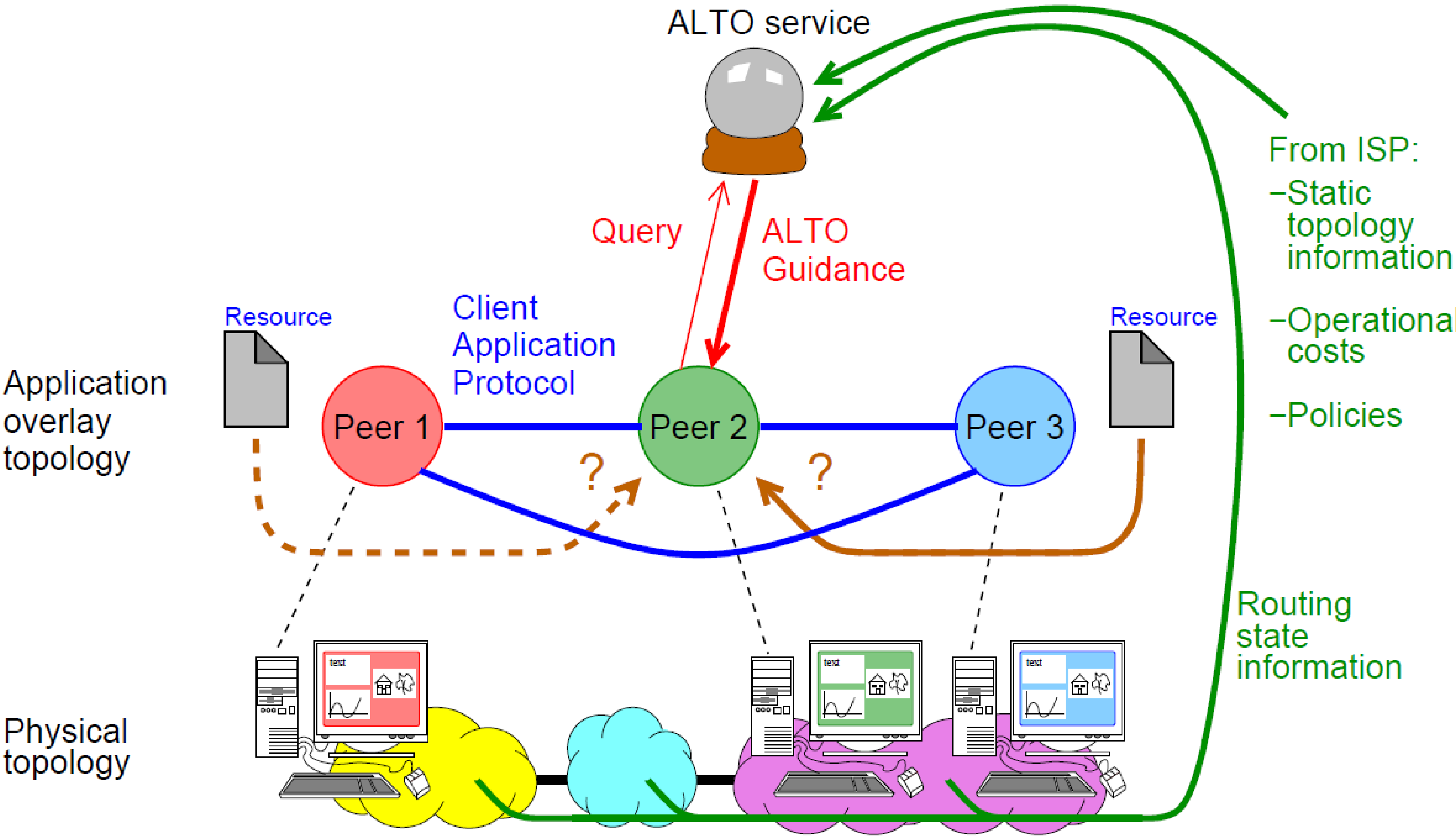
- useful parameters can easily be measured
  - RTT, hop counts
  - useful for topology management and peer scheduling
- some parameters are difficult to obtain
  - available bandwidth technique are error prone
  - the capacity of the bottleneck may be intrusive
- some parameters must be carefully measured
  - Input parameters for the Neural Network: losses, loss burst size, delays
  - misguided measurement in the RNN input will mistake the QoE estimation process
  - measurement accuracy is crucial for good QoE estimation

Repository

- A repository has been developed and released:
  - It stores information published by peers
    - SQL information base
    - HTTP communication interface
    - Currently implement the peer repository
      - E-REP (ALTO server) is alto under development

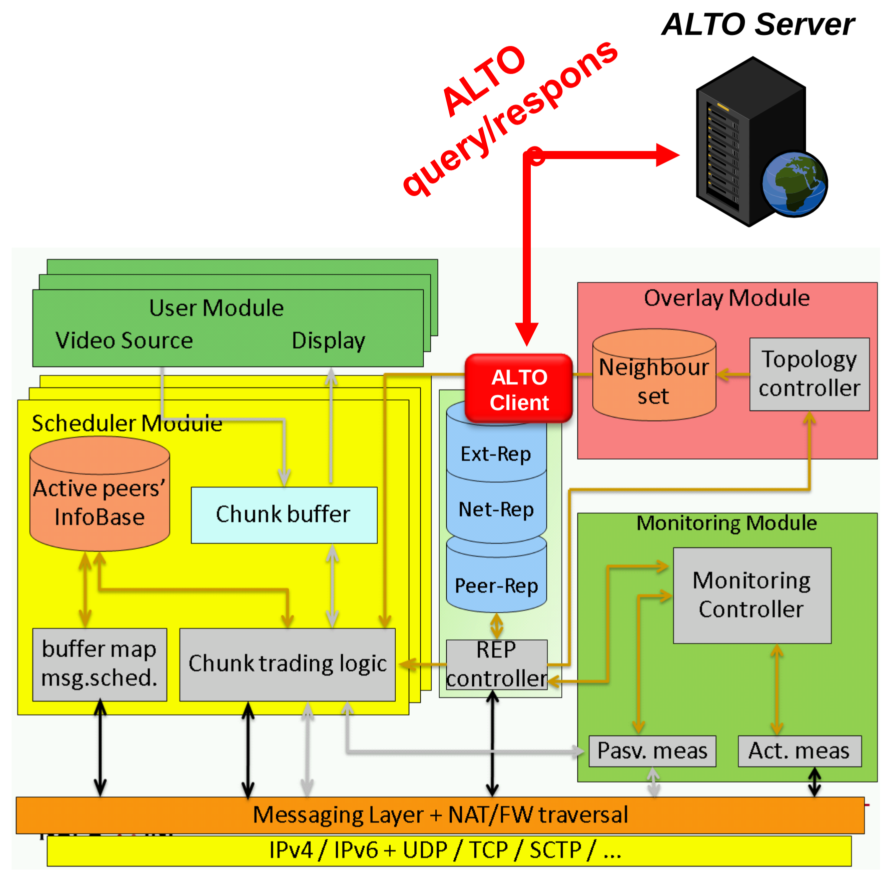


Application Layer Traffic Optimization (ALTO)

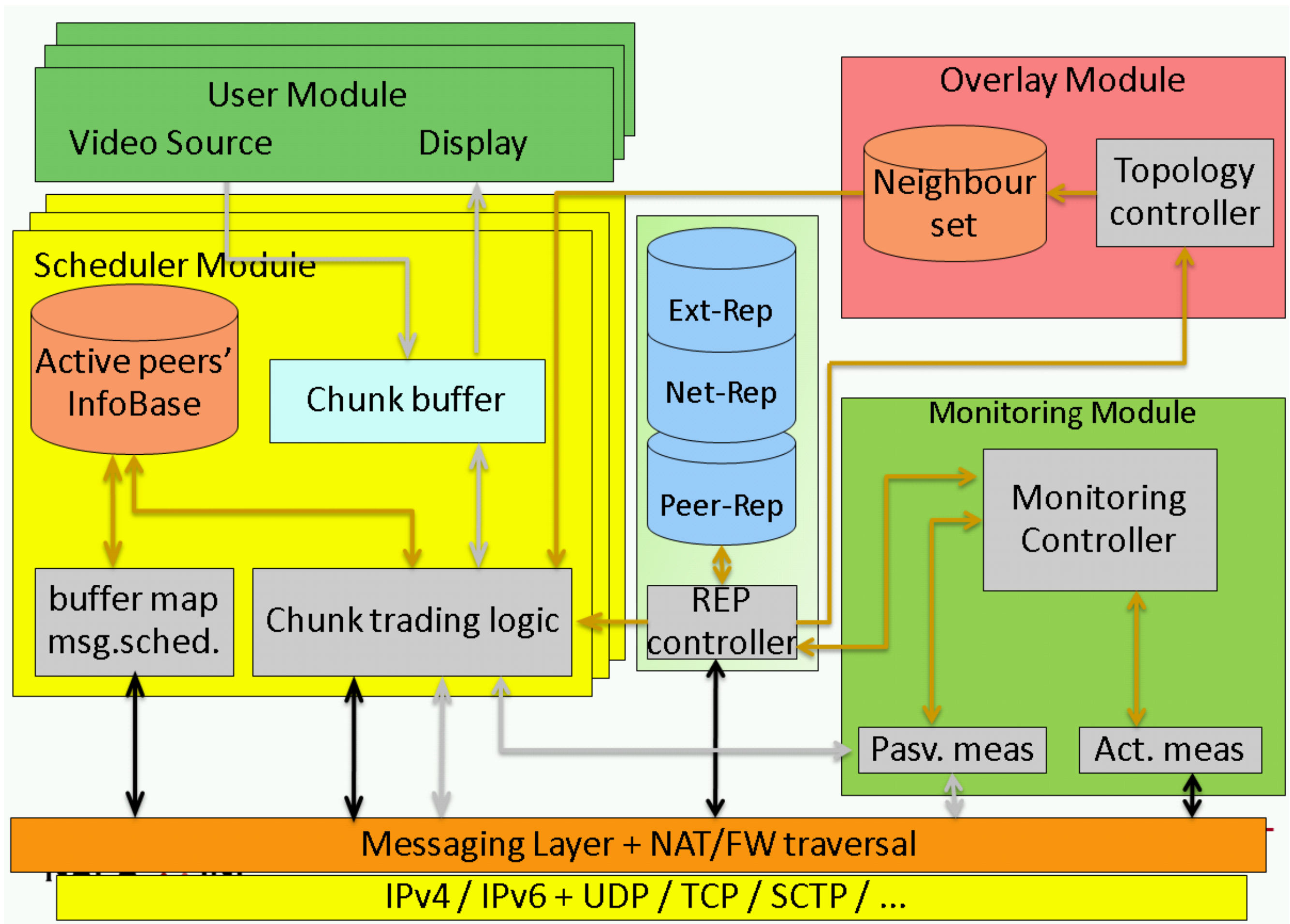


ALTO in Napa-Wine Architecture

- Integration of ALTO Server + Client into Napa-Wine Architecture
  - ALTO Client is part of the External Repository (E-Rep)
  - E-Rep can contact ALTO-server to gain network-layer information the peers cannot measure themselves

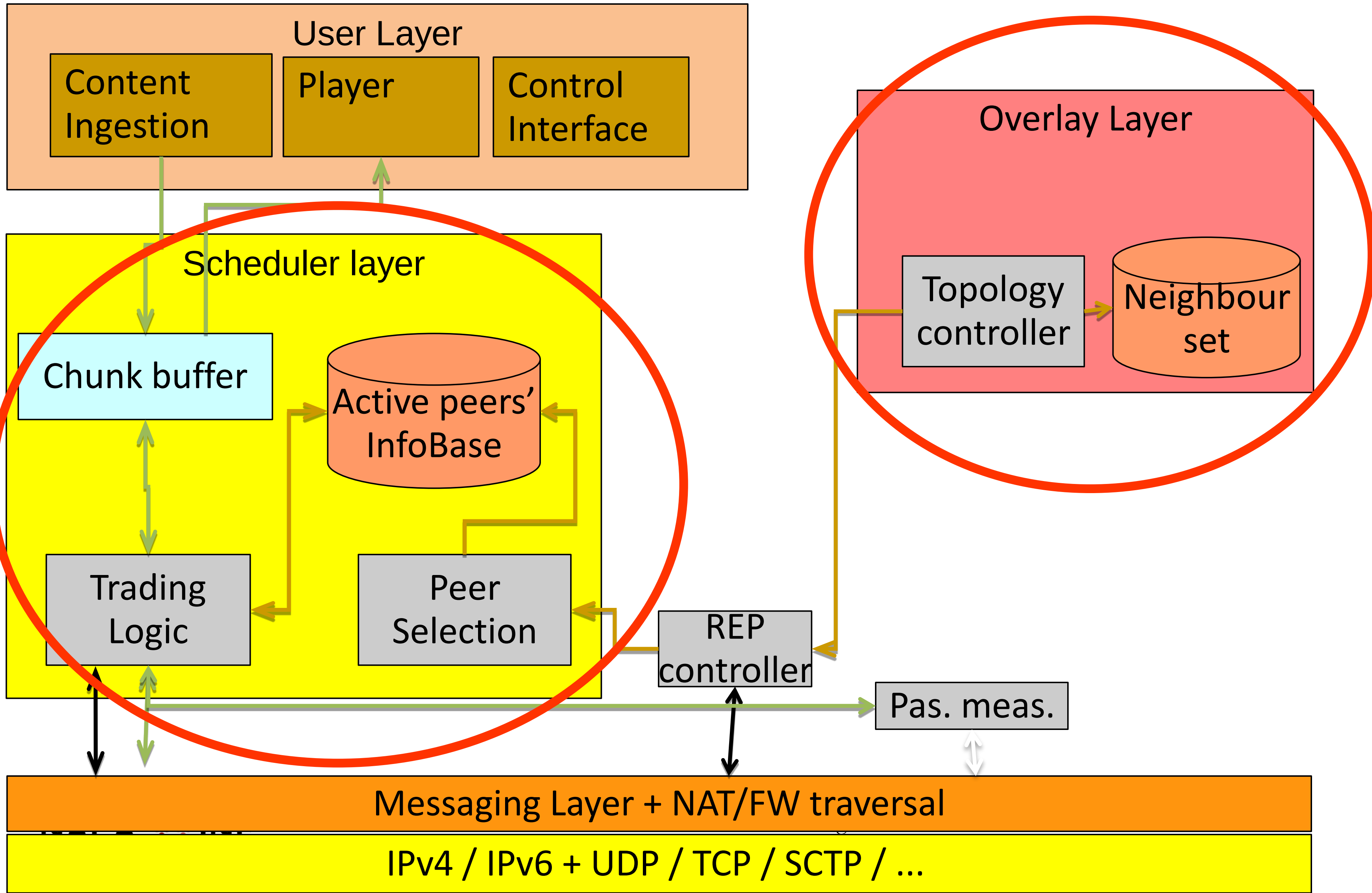


Scheduling and overlay



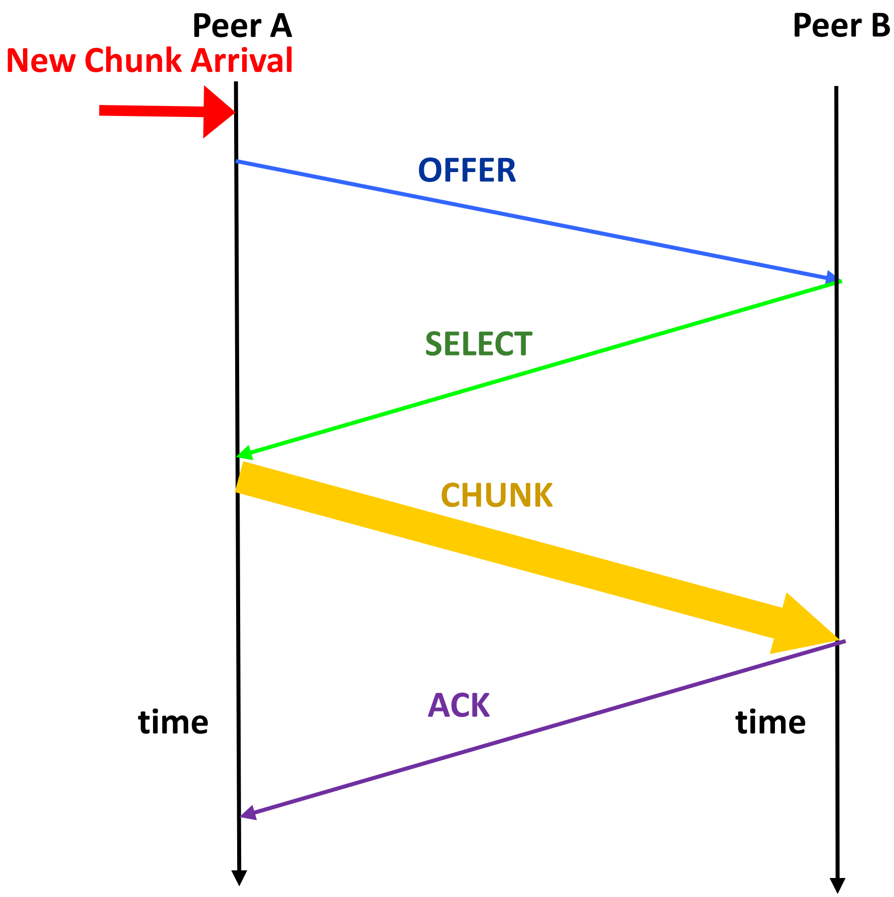
NAPA WINE

Scheluning and Overlay modules

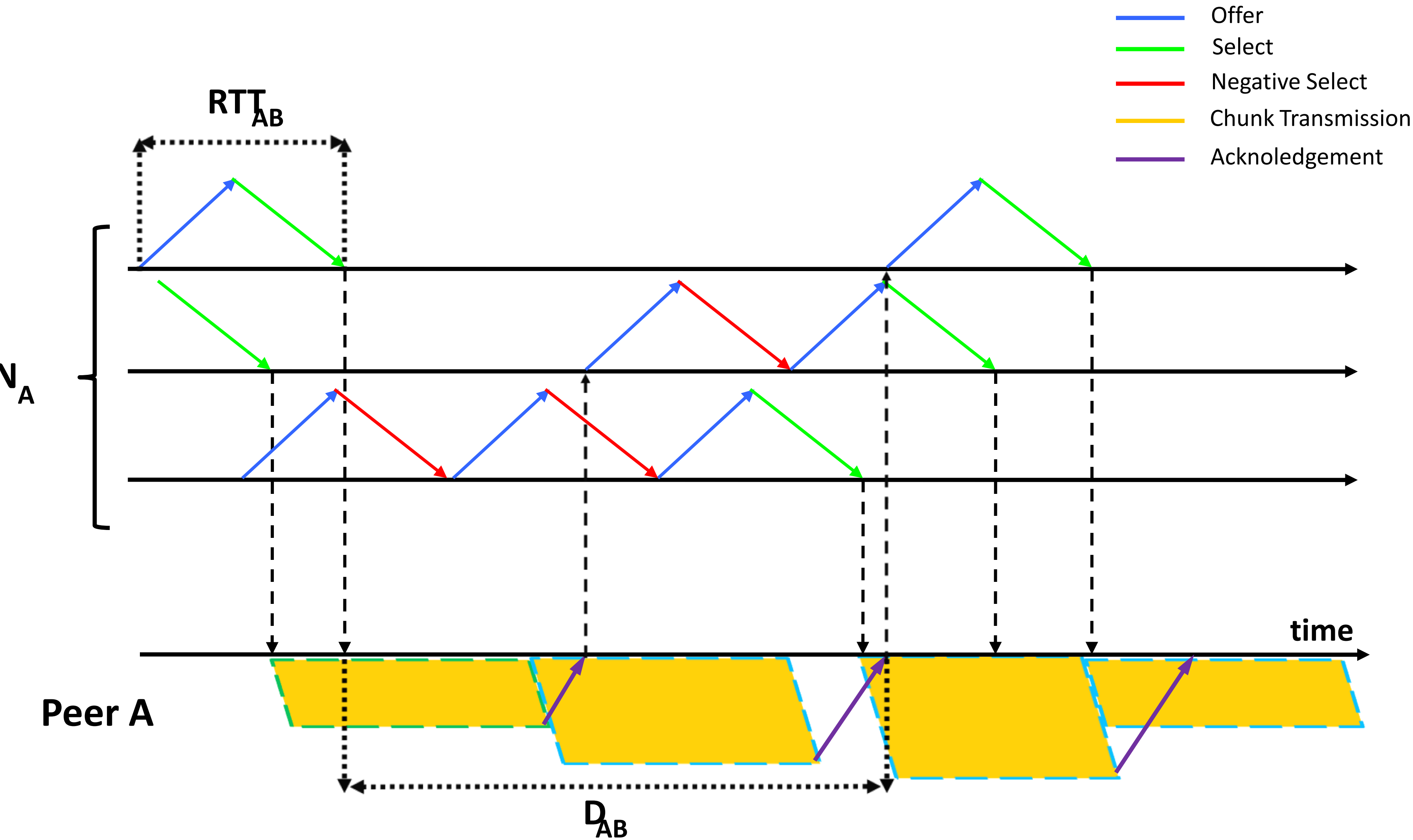


Signalling Thread

- A peer publishes the set of chunks it possesses through an ***offer*** message.
- Peers specify the chunk they are interested in with a ***select*** message.
- Once the select message is received, chosen chunk is transmitted (over UDP).
- An ***ack*** is sent back once chunk is received



System Dynamics

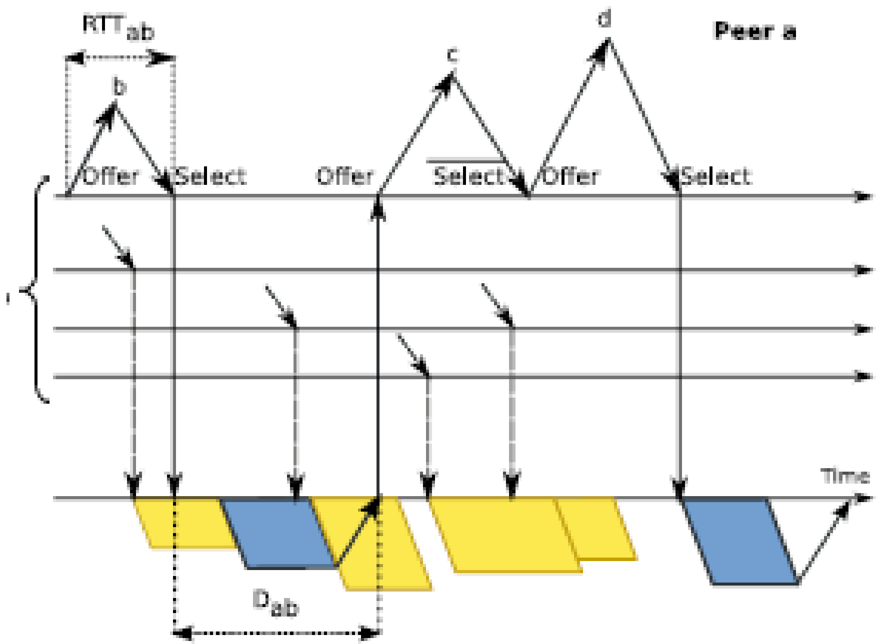


Congestion Control is needed

- ***The number of parallel threads N_A*** must match peers' upload capacity.
  - If N_A is too small,
    - Peers' upload bandwidth is not exploited at best.
    - The transmission queue empties quickly.
    - Long periods of inactivity.
  - If N_A is too large,
    - Transmission queue becomes too long.
    - Large delivery delays and, possibly, losses.
- Exploit **upload bandwidth** and mantain **short queues** to limit the delivery delay!
- Optimal setup depends on the network scenario which is **unpredictable**

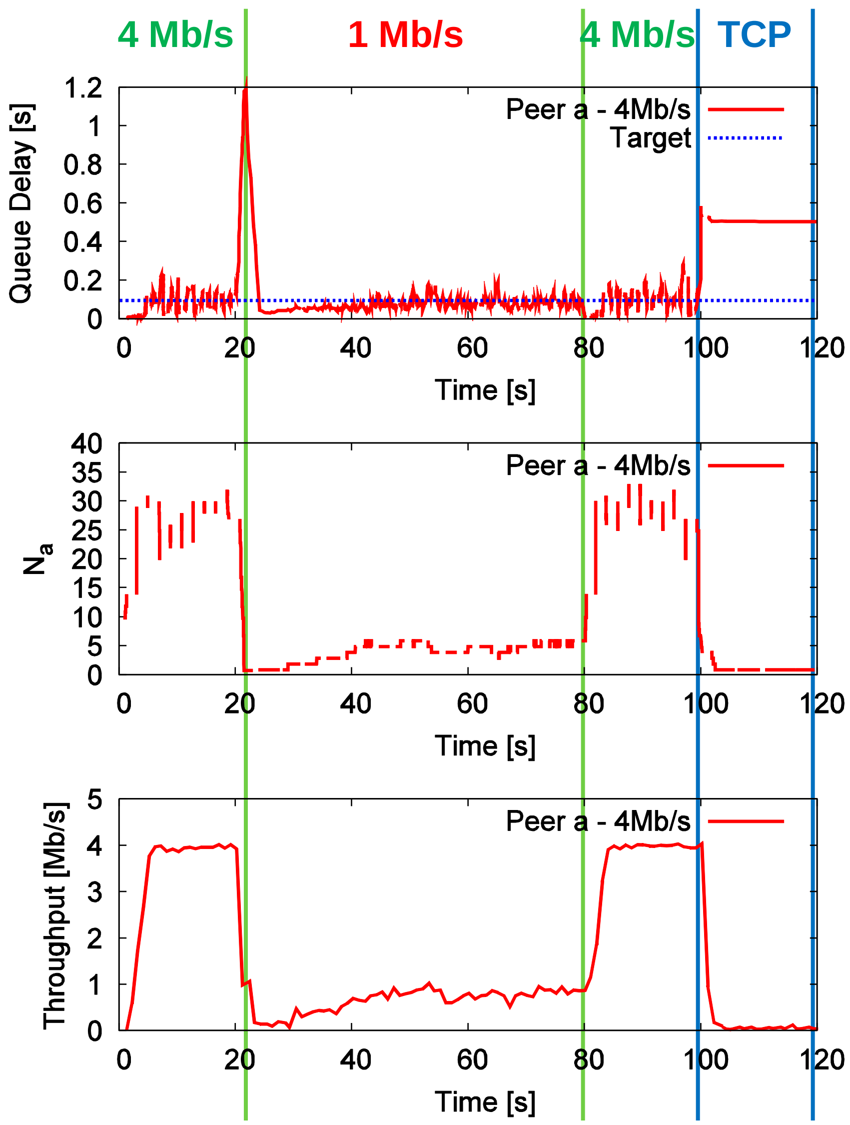
Hose Rate Control

- The algorithm runs everytime an **ACK** is received:
  1. $D = t_{rx,ack} - t_{rx,sel} - RTT_{AB}$
  2. $W_A(n) = W_A(n-1) - \alpha(D-D_0)$
  3. $\Delta N_A = floor(W_A(n)) - floor(W_A(n-1))$



HRC Performance

- **Queue delay** (*D*), **number of active signalling threads** (N_A) and **throughput** evolution during time adopting HRC (ρ = 0.9, D_0=150ms).



NAPA WINE

Logical topology

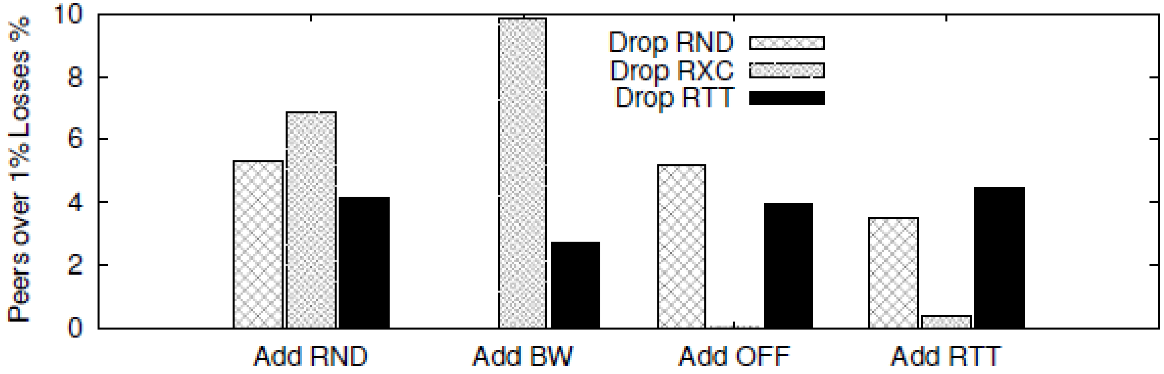
- The logical topology is a directed graph, every node chooses its K in-neighbors (parents).
- It can be built either exploiting repository information gossiping mechanisms (Newscast)
- Every T sec. peer p updates the list of in-neighbors NI(p). At every update, NI (p) is the result of two separate filtering functions:
  - one that selects the peers *to drop,*
  - *another* one selecting parents *to add.*

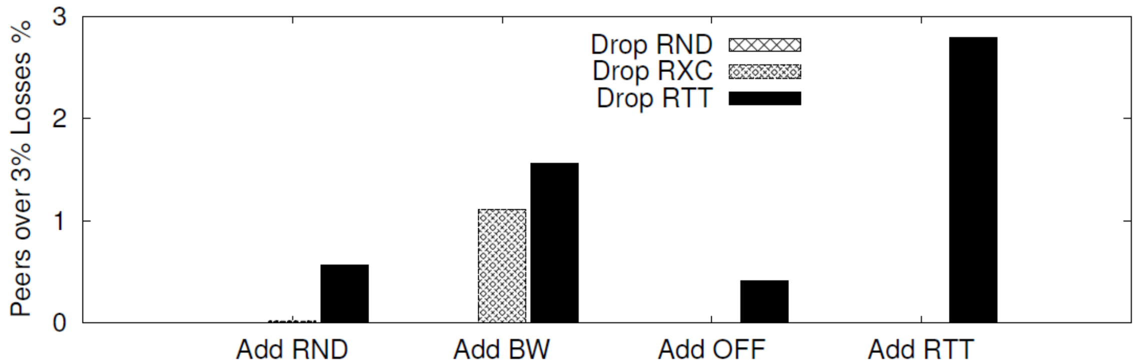
Logical Topology (cnt)

- For these filtering functions we consider:
  - peer upload bandwidth,
  - path RTT or
  - path packet loss rate,
- and some application layer metrics
  - the peer offer rate
  - number of received chunks from a parent.

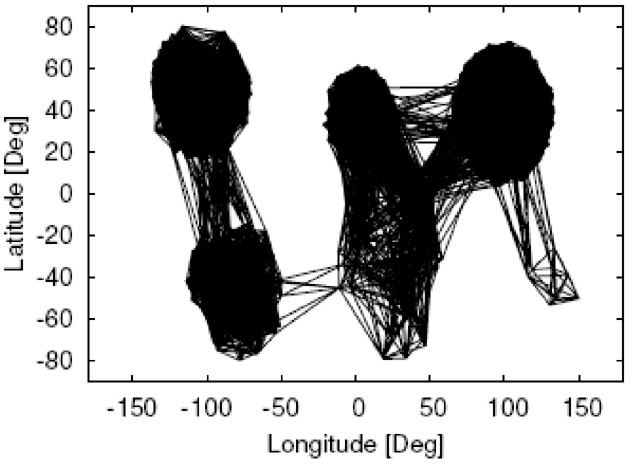
A sufficient degree of randomness must be guaranteed!

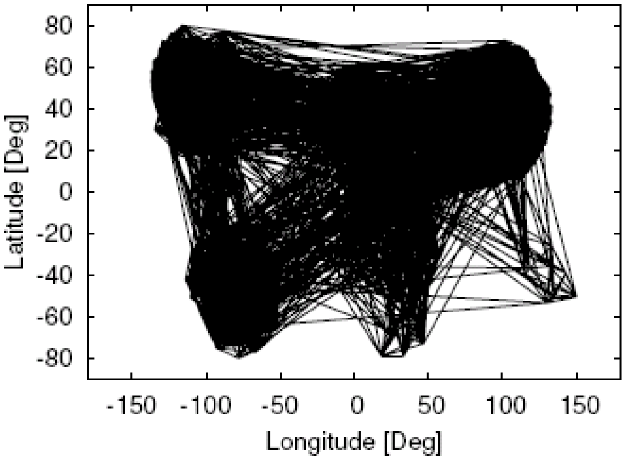
Performance





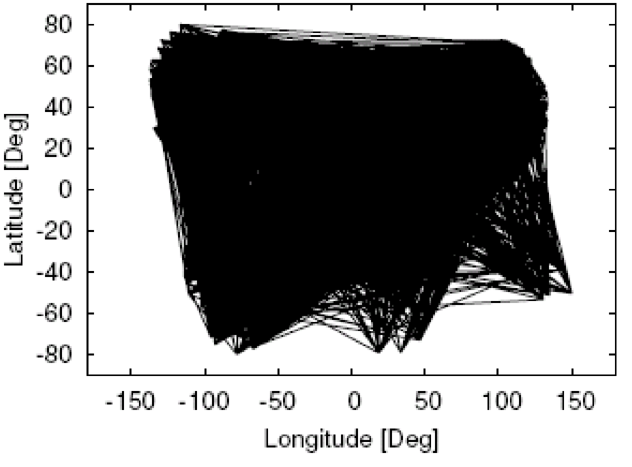
Examples of topologies





RTT-Random

RTT-RTT



Random-Random

NAPA WINE

Scientific Conclusions

- In most of the scenarios it is possible to localize the traffic without endangering the perceived QoE.
  - Being too extreme in localizing traffic may cause degradations of the QoE.
  - Nevertheless there are margins within which traffic can be localized without degrading QoE.
  - In several cases a smart localization strategy can even slightly improve the application performance.

Scientific Conclusions (Cnt)

- Localization is more effective if the application can exploit cost metrics exported by the operators through the ALTO interface that has been standardized within the IETF, and of which NAPA-WINE is one of the principal contributors.

Scientific Conclusions (Cnt)

- Continuous monitoring of the network status can greatly improve the ability of detecting anomalies and the ability to promptly react to them.
- Network monitoring can easily be achieved by embedding a distributed measurement platform within the application (as done in Winestreamer).

Scientific Conclusions (Cnt)

- To achieve good performance it is necessary that the distributed algorithms for the design and the maintenance of the overlay topology guarantee a sufficient degree of randomness.
- Local selections of neighboring peers according to deterministic rules can result in an overall overlay topology with bad graph properties.

Scientific Conclusions (Cnt)

- Information about the upload bandwidth of peers can be effectively exploited to design algorithms for the design and maintenance of the overlay topology and chunk scheduling that optimize the system performance.

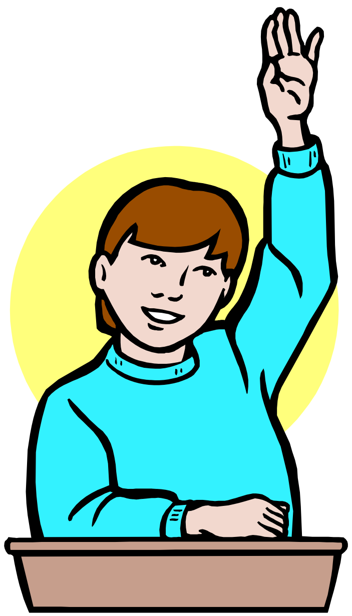
Scientific conclusions (Cnt)

- UDP is preferable to TCP as transport protocol, since it significant reduces the chunk transfer times.
- Peers must be supplied with a simple application level rate control mechanism to avoid bandwidth wastage and congestions

Winestreamer/Peerstreamer

- Available at http://peerstreamer.org

THE END



Thank you!

Questions?
Comments?