Optimal Congestion Control and Request Routing in Information-Centric Networking

Luca Muscariello
Orange Labs Networks
Network Modeling and Planning Department

Joint work with
Giovanna Carofiglio, Massimo Gallo (Alcatel Lucent, Bell Labs)
Michele Papalini (University of Lugano, Switzerland)
Sen Wang (Tsinghua University, P.R. China)

motivation

- CCN transport model
  - multi-point to point
  - unique end-point at the receiver
  - network nodes perform smart packet processing
- traffic management by request flow control and forwarding
- design lightweight protocols and run large scale experimentation
  - almost no tuning
  - repeatable experimentation (daily base)

outline

- Short introduction of CCN forwarding
- Network model description
- Network protocols
- Experimentation

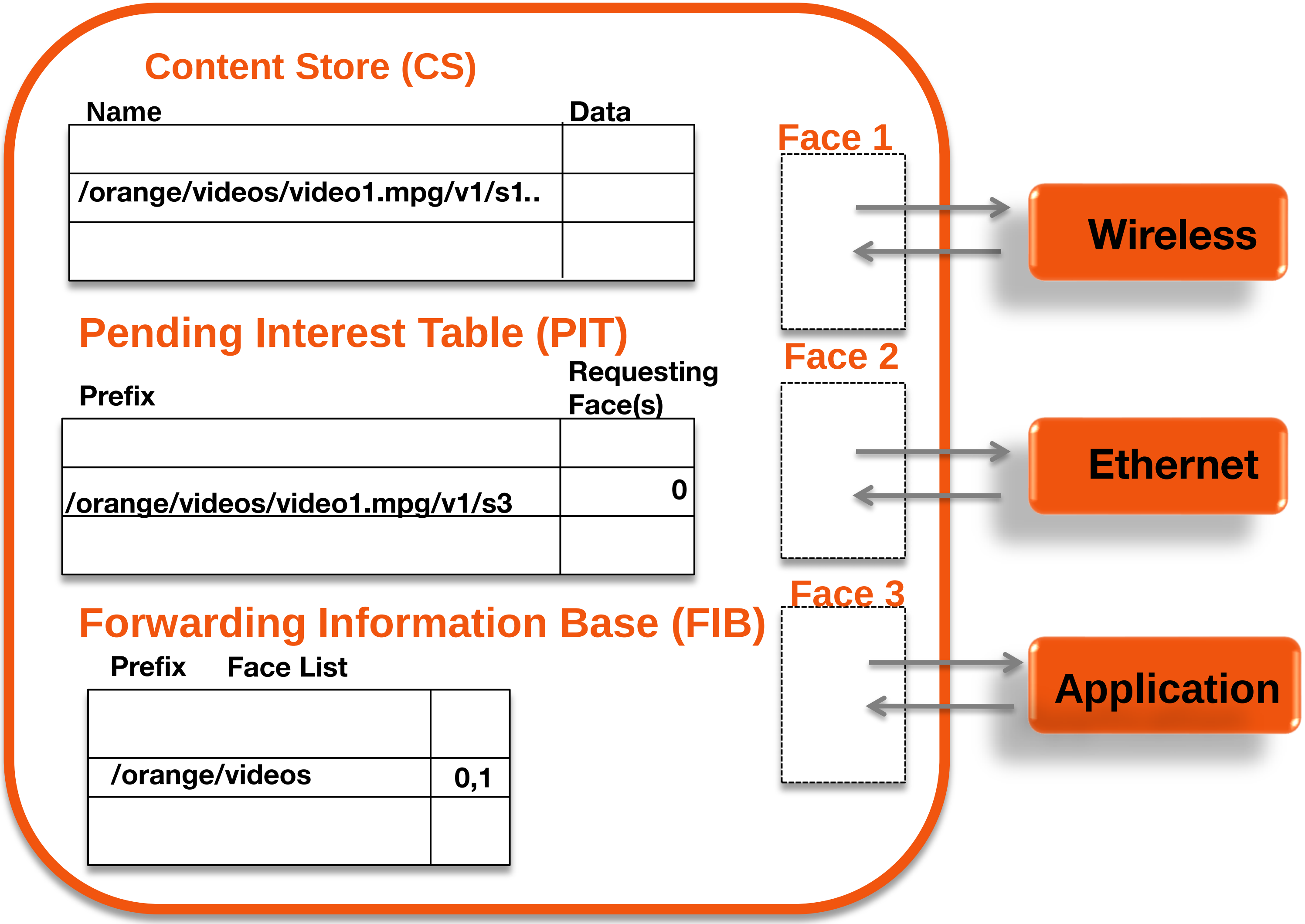
outline

- Short introduction of CCN forwarding
- Network model description
- Network protocols
- Experimentation

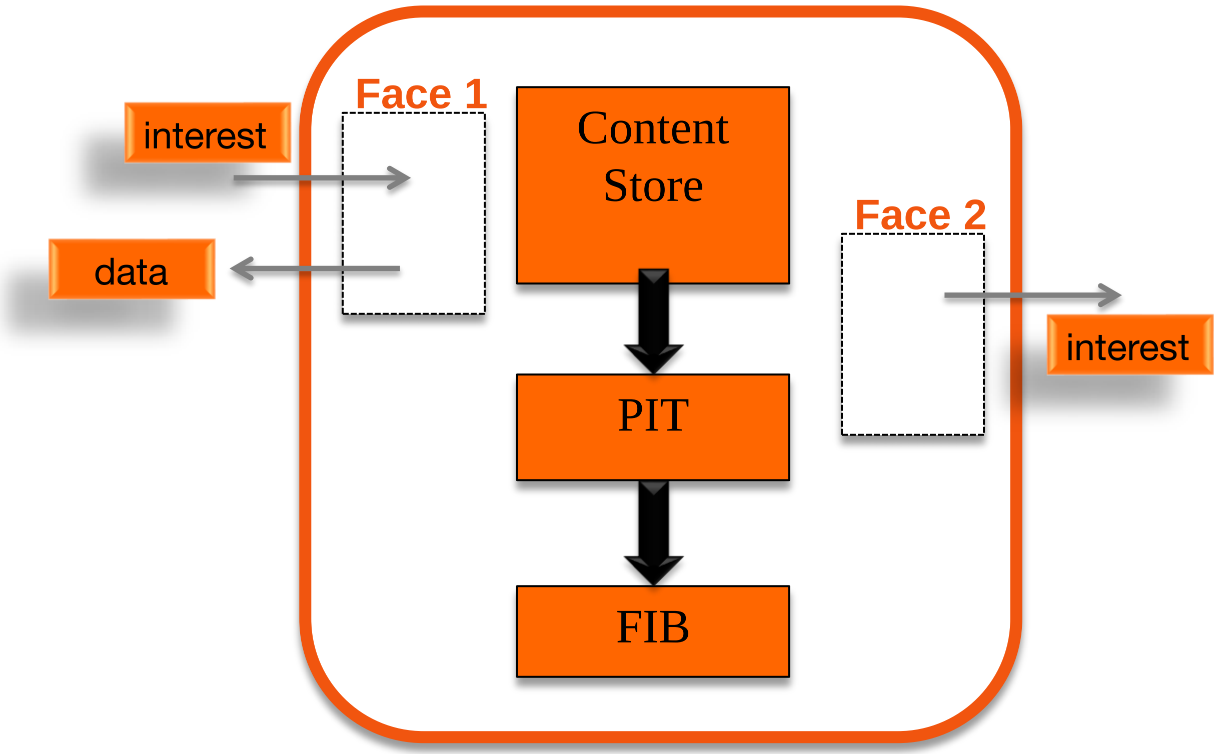
CCN node forwarding (1/3)

- an application:
  - requests content by expressions of INTEREST
  - only INTERESTs packets are routed
- a node receiving an INTEREST from an input face
  - sends DATA back to the input face if in the local content store
  - else checks a local matching pending interest
  - else checks a match in the local FIB
    - forward the INTEREST through the matching output interface

CCN node forwarding (2/3)



CCN forwarding (3/3)



outline

- Short introduction of CCN forwarding
- Network model description
- Network protocols
- Experimentation

network model notation

- The network is modeled as a graph with bi-directional arcs $\mathcal{G} = (\mathcal{V}, \mathcal{A})$
- If INTEREST flows on link (i, j) DATA flows on the reverse link (j, i)
- Links have finite capacity $C_{ij} > 0, \forall (i, j) \in \mathcal{A}$
- A content retrieval $n \in \mathcal{N}$ is uniquely associated to a user node in $\mathcal{U} \subset \mathcal{V}$
- The variables x_{ij}^{n} denotes the DATA rate of flow $n \in \mathcal{N}$ over link (i, j)
- The variables $\tilde{x}_{ji}^{n}$ denotes the INTEREST rate of flow $n \in \mathcal{N}$ over link (j, i)
- The set of ingress nodes $\Gamma^{+}(i)$ for node $i \in \mathcal{V}$
- The set of egress nodes $\Gamma^{-}(i)$ for node $i \in \mathcal{V}$

- We define the function $h^{n}(i) \in \{0, 1\}$ to identify the location of content in the network

$$\mathcal{S}(n) = \{i : h^{n}(i) = 1\}$$

global objectives

$$\begin{cases} \boxed{\max_{\boldsymbol{y}} \sum_n \boldsymbol{U}_n(y_i^n) - \sum_{i,j \in \mathcal{A}} \boldsymbol{C}_{ij}(\rho_{ij})} \\ \sum_{n \in \mathcal{N}} x_{ij}^n = \rho_{ij} \qquad \forall (i,j) \in \mathcal{A} \\ \sum_{\ell \in \Gamma^-(i)} x_{\ell i}^n = y_i^n, \quad \forall i, n \\ \sum_{j \in \Gamma^+(i)} x_{ij}^n (1 - h^n(i)) = y_i^n \qquad \forall i \notin \mathcal{U}, n \\ \rho_{ij} \leq C_{ij} \qquad \forall (i,j) \in \mathcal{A} \\ x_{ij}^n \geq 0 \qquad \forall i, j, n \end{cases}$$

User utility maximization and network cost minimization

link load

node ingress traffic

flow conservation constraint

link capacity constraint

primal decomposition

$$\begin{cases} \min\limits_{\boldsymbol{x}} \sum\limits_{i,j \in \mathcal{A}} \boldsymbol{C}_{ij} \left(\sum\limits_{n \in \mathcal{N}} x_{ij}^n \right) \\ \sum\limits_{\ell \in \Gamma^-(i)} x_{\ell i}^n = y_i^n, \quad \forall i, n \\ \sum\limits_{j \in \Gamma^+(i)} x_{ij}^n (1 - h^n(i)) = y_i^n \qquad \forall i \notin \mathcal{U}, n \\ x_{ij}^n \geq 0 \qquad \forall (i,j) \in \mathcal{A} \qquad \forall n \in \mathcal{N} \end{cases}$$

master problem
network cost minimization

$$\begin{cases} \max\limits_{\boldsymbol{y}} \sum\limits_{n} \boldsymbol{U}_n(y^n) \\ \sum\limits_{n:(i,j) \in \mathcal{L}(n)} y^n \leq C_{ij} \qquad \forall (i,j) \in \mathcal{A} \\ y^n \geq 0 \qquad \forall n \in \mathcal{N} \end{cases}$$

sub-problems
at the receivers

network cost minimization P1

$$
\begin{cases}
\min\limits_{\boldsymbol{x}} \sum\limits_{i,j \in \mathcal{A}} \boldsymbol{C}_{ij} \left(\sum\limits_{n \in \mathcal{N}} x_{ij}^n \right) \\
\sum\limits_{\ell \in \Gamma^-(i)} x_{\ell i}^n = y_i^n, \quad \forall i, n \\
\sum\limits_{j \in \Gamma^+(i)} x_{ij}^n (1 - h^n(i)) = y_i^n \qquad \forall i \notin \mathcal{U}, n \\
x_{ij}^n \geq 0 \qquad \forall (i,j) \in \mathcal{A} \qquad \forall n \in \mathcal{N}
\end{cases}
$$

Lagrangians for P1

$$L_C(\boldsymbol{x}, \boldsymbol{\mu}) =$$

$$\sum_{i,j} \boldsymbol{C}_{ij} \left(\sum_{n \in \mathcal{N}} x_{ij}^n \right) - \sum_n \sum_i \mu_i^n (\sum_{l \in \Gamma^-(i)} x_{li}^n - \sum_{j \in \Gamma^+(i)} x_{ij}^n)$$

$$= \sum_{i,j} \boldsymbol{C}_{ij}(\rho_{ij}) - \sum_n \sum_{i,l} \mu_i^n x_{li}^n + \sum_n \sum_{i,j} \mu_i^n x_{ij}^n$$

$$= \sum_i \boxed{\sum_{l \in \Gamma^-(i)} [\boldsymbol{C}_{li}(\rho_{li}) - \sum_n (\mu_i^n - \mu_l^n) x_{li}^n]}$$

user utility maximization P2

$$
\begin{cases}
\max\limits_{\boldsymbol{y}} \sum\limits_{n} \boldsymbol{U}_n(y^n) \\
\sum\limits_{n:(i,j)\in\mathcal{L}(n)} y^n \leq C_{ij} \qquad \forall (i,j) \in \mathcal{A} \\
y^n \geq 0 \qquad \forall n \in \mathcal{N}
\end{cases}
$$

Lagrangian for P2

$$L_{\boldsymbol{U}}(\boldsymbol{y}, \boldsymbol{\lambda}) = \sum_n \boldsymbol{U}_n(y^n) - \sum_{ij} \lambda_{ij}\Big(\sum_{n:(i,j)\in\mathcal{L}(n)} y^n - C_{ij}\Big)$$

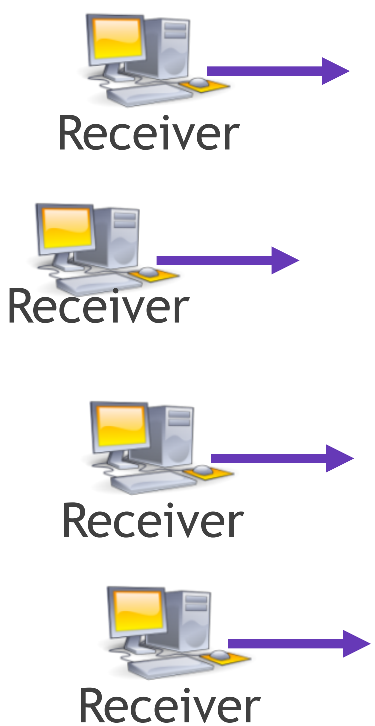
$$= \sum_n \boldsymbol{U}_n(y^n) - \sum_n \sum_{(i,j)\in\mathcal{L}(n)} \lambda_{ij} y^n + \sum_{ij} \lambda_{ij} C_{ij}$$

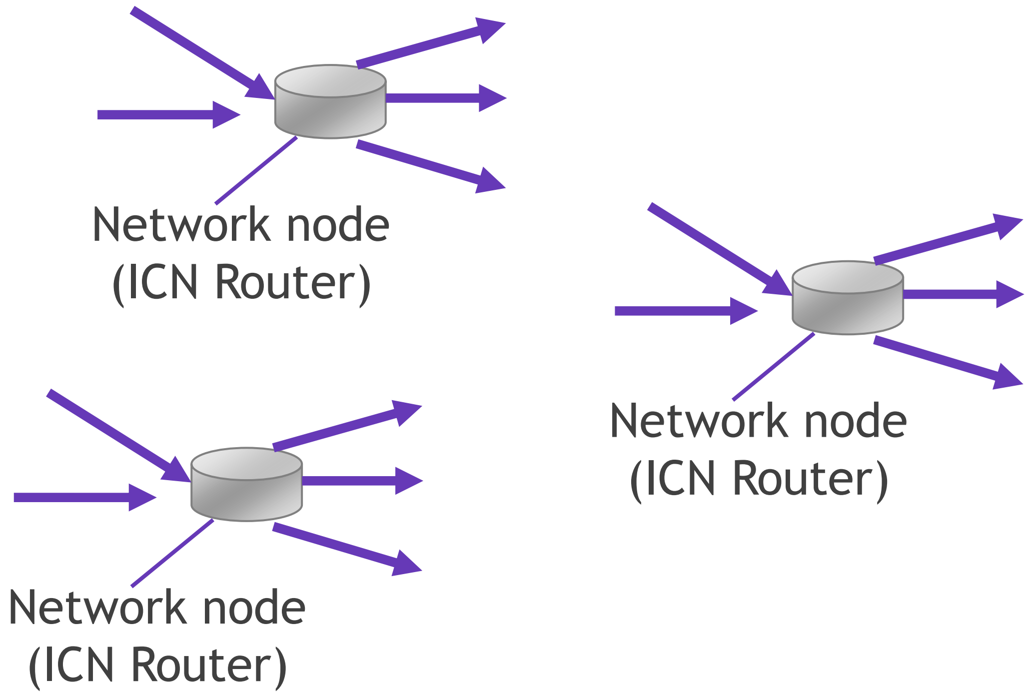
$$= \sum_n \boxed{(\boldsymbol{U}_n(y^n) - \lambda^n y^n)} + \sum_{(i,j)} \lambda_{ij} C_{ij}$$

$$\lambda^n \equiv \sum\nolimits_{(i,j)\in\mathcal{L}(n)} \lambda_{ij}$$

optimal distributed solution

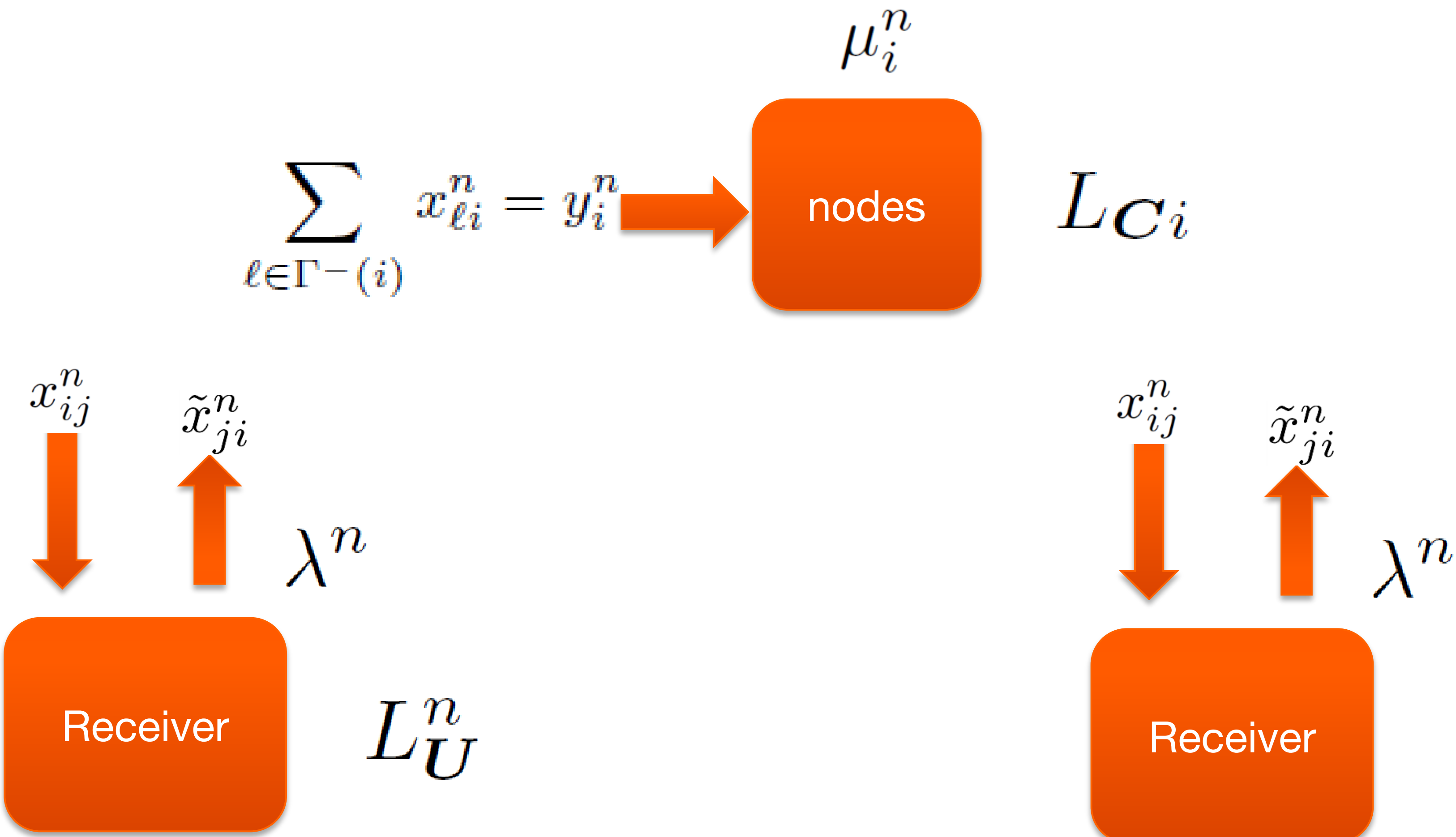
$$L_{\boldsymbol{U}}^n = \boldsymbol{U}_n(y^n) - \lambda^n y^n$$





$$L_{\boldsymbol{C}i} = \sum_{l \in \Gamma^-(i)} \left[\boldsymbol{C}_{li}(\rho_{li}) - \sum_n (\mu_i^n - \mu_l^n) x_{li}^n\right]$$

algorithm main functioning



local Lagrangians optimization

- standard techniques based on gradients
  - minimization of $L_{\boldsymbol{C}i}$
  - maximization of $L_{\boldsymbol{U}}^{n}$
- Gradient based solutions

$$\nabla f(x_1, \ldots, x_N, \ldots, \mu_l, \ldots) \cdot e_i = \frac{\partial x_i}{\partial t}$$

outline

- Short introduction of CCN forwarding
- Network model description
- Network protocols
- Experimentation

receiver driven congestion control (1/2)

- computation of link Lagrange multipliers
- compute path aggregate $\lambda^n \equiv \sum_{(i,j)\in\mathcal{L}(n)} \lambda_{ij}$
- maximize local Lagrangian

$$\frac{d}{dt}\lambda_{ij}(t) = \kappa_{ij}(t)\Big(\sum_{n:(i,j)\in\mathcal{L}(n)} y^n(t) - C_{ij}\Big)$$

$$\frac{d}{dt}y^n(t) = \gamma^n(t)(\boldsymbol{U}'_n(y^n(t)) - \lambda^n(t))$$

receiver driven congestion control (2/2)

- by taking $\kappa_{ij}(t) = \frac{1}{C_{ij}}$

$$\frac{d}{dt}\lambda_{ij}(t) = \kappa_{ij}(t)\Big(\sum_{n:(i,j)\in\mathcal{L}(n)} y^n(t) - C_{ij}\Big)$$

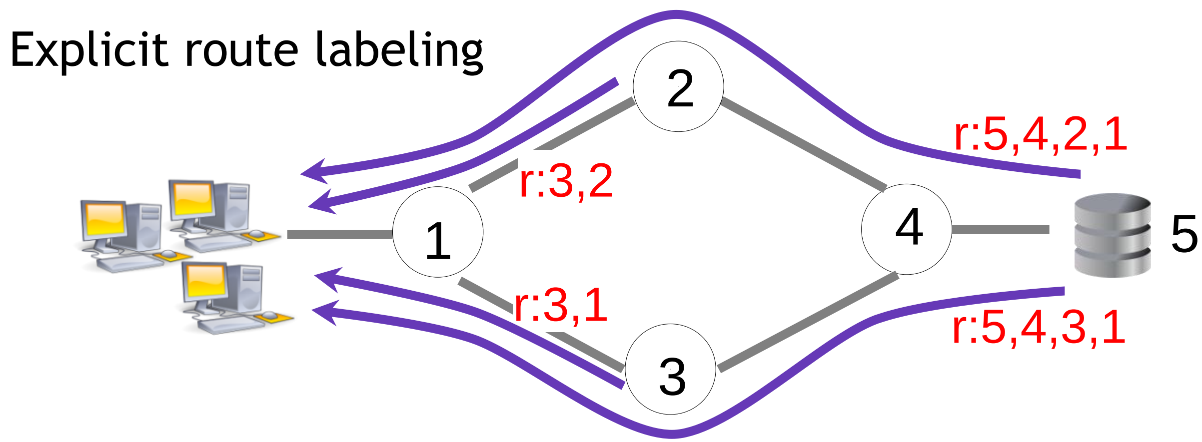
- Fluid queue delay evolution at a link
- So that $\lambda^n(t)$ is the total path delay associated to a given flow
- The INTEREST/DATA protocol can be used to measure such delay at the receiver

per-path delay measurements

- give a label to a path so that the receiver can keep per path statistics

$$\lambda^n(r,t) = \sum_{(i,j)\in r : r\in\mathcal{R}^n} \lambda_{ij}(t)$$

$$\frac{d}{dt} y^n(t) = \gamma^n(t)(\boldsymbol{U}'_n(\tilde{y}^n(t)) - \sum_{r\in\mathcal{R}^n(s)} \lambda^n(r,t)\phi(r,t))$$



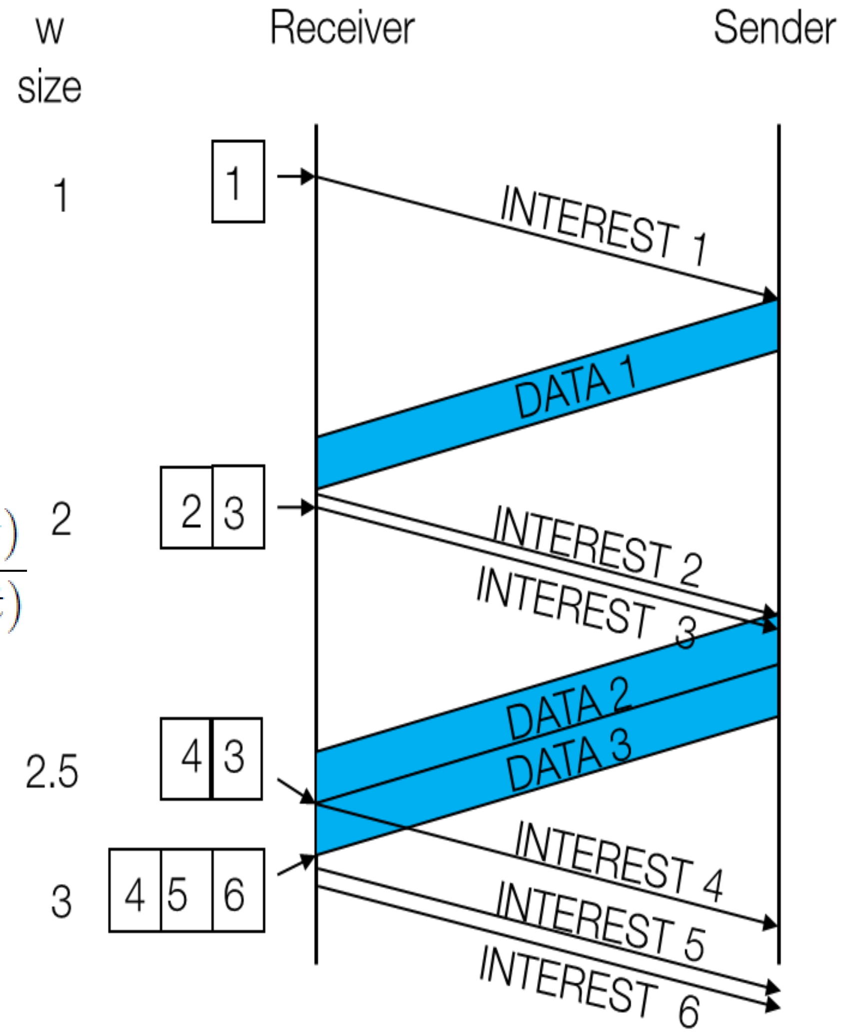
the network protocol:
window based flow control @RX

$$y^n \longrightarrow \widetilde{y^n}$$

$$p_r(t) = p_{\min} + \Delta p_{\max} \frac{R_r(t) - R_{\min,r}(t)}{R_{\max,r}(t) - R_{\min,r}(t)}$$

$$\frac{d}{dt} W(t) = \frac{\eta}{R(t)} - \beta W(t) \sum_{r \in \mathcal{R}} p_r(t) \phi_r(t) \frac{W(t)}{R_r(t)}$$

- one single window
- splitting is managed by the forwarding engine
- congestion signals decomposed in paths



request forwarding (1/2)

- The local Lagrangian minimization at every network node is equivalent to minimizing the Lagrangian multiplier μ_i^n
- proof by imposing Karush-Kuhn-Tucker condition on the local Lagrangians

$$\frac{\partial \boldsymbol{C}_{ji}}{\partial x_{ji}^n} = \frac{\partial \boldsymbol{C}_{ji}}{\partial \rho_{ji}} \frac{\partial \rho_{ji}}{\partial x_{ji}^n} = \frac{\partial \boldsymbol{C}_{ji}}{\partial \rho_{ji}} = \mu_i^n - \mu_j^n$$

$$\frac{\partial \boldsymbol{C}_{ji}}{\partial x_{ji}^n} \leq \frac{\partial \boldsymbol{C}_{li}}{\partial x_{li}^n}, \qquad \forall l \in \Gamma^-(i)$$

- by iterating up to a hitting cache

$$\mathcal{S}(n) = \{i : h^n(i) = 1\}$$

request forwarding (2/2)

- using a gradient algorithm we can obtain the Lagrangian multipliers μ_i^n
- which have a protocol interpretation: they are the total number of pending interest for a given flow (content item)

$$x_{ij}^n \longrightarrow \widetilde{x_{ji}^n}$$

$$\frac{d\mu_i^n}{dt} = \eta_i^n \left(\sum_{j \in \Gamma^+(i)} x_{ij}^n - \sum_{l \in \Gamma^-(i)} x_{li}^n \right)$$

Pending Interest Table (PIT)

chunk name	interface
ccnx:/data/**4**	1,2
ccnx:/data/**3**	1
ccnx:/data/**6**	1

local optimization of PIT size

- minimize the sum of pending interests on all interfaces

$$\mu = \sum_{j \in \Gamma^-(i)} \mu_j$$

$$\sum_{j \in \Gamma^-(i)} x_j = y$$

- drawbacks
  - the optimal solution of this problem has zero rate on some interfaces
  - additional probing techniques
  - requires to know an analytical expression of the delay

changing the objective

- minimize the most loaded interface

$$\min \max_j \mu_j(x_j)$$
$$\sum_{j\in\Gamma^-(i)} x_j \;=\; y$$

- advantages
  - the solution is very simple $\mu_j = \mu$

$$\mu = \lim_{T\to\infty} \frac{1}{T}\int_0^T \mu_j(t)dt = \lim_{T\to\infty} \frac{\int_0^T x_j(t)\mathrm{VRTT}_j(x_j(t))dt}{\sum_l \Delta T_l}$$
$$= \frac{\Delta \bar{T}_j}{\sum_l \Delta \bar{T}_l} y \mathrm{VRTT}_j(y) = \phi_j \bar{\mu}_j$$

$$\phi_j \;=\; \mu/\bar{\mu}_j \qquad\qquad 1/\mu \;=\; \sum_j \mu_j$$

request forwarding network protocol

- the number of pending interests for a given name prefix is available in the PIT
- filter out high variable components with low pass filters
- load balance future requests using a randomized load balancer $\phi_j = \mu/\bar{\mu}_j$

outline

- Short introduction of CCN forwarding
- Network model description
- Network protocols
- Experimentation

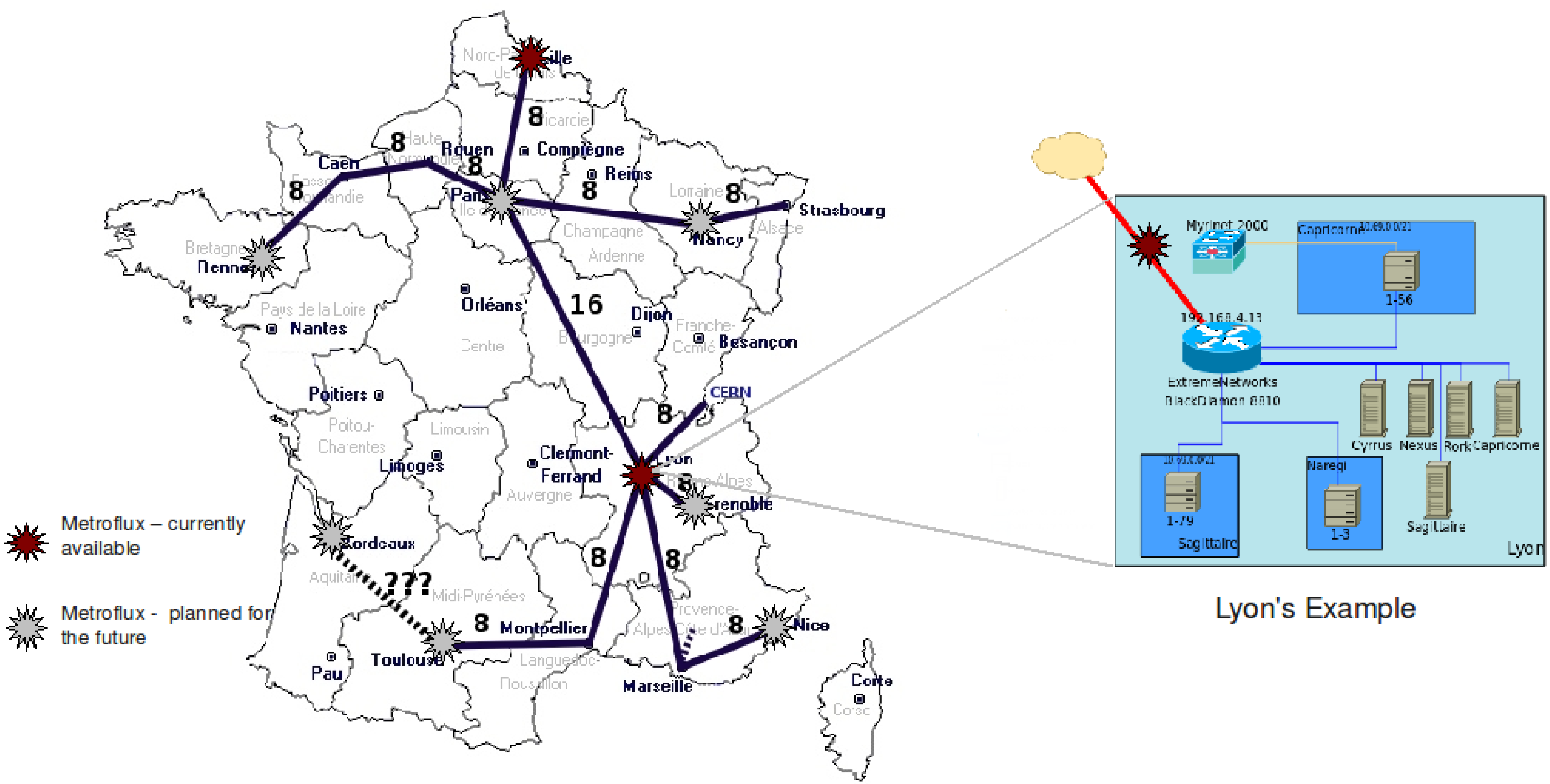
implementation

- we implemented the above described mechanisms in CCNx-0.6.1 (works on latest releases also)
- novel caching framework
  - flexible to implement your preferred caching system
  - LRU for this paper
- new strategy layer
  - exploit an “almost” clean available “API”
- a new data retrieval application (Interest generator)
  - implementation of flow control + path labeling in nodes
- new virtual repo based on WashU (thanks Patrick)
  - allows for large scale virtual content repositories
  - with ccnr you would spend 90% of your test filling repos.

Experimentation

- we performed several experiments in a data center
  - dynamic bootable 3.x linux kernels images
  - CCNx overlay over IP tunnels among servers
  - IP over IP as a layer 2 encapsulation only
  - Linux tc to shape IP overlay link rates
  - pre-configured CCNx routing to fill FIBs
- we analyzed 4 different scenarios:
  - scenario 1: efficiency in multipath flow control
  - scenario 2: impact of in-network caching
  - scenario 3: request forwarding scalability
  - scenario 4: link failure reaction

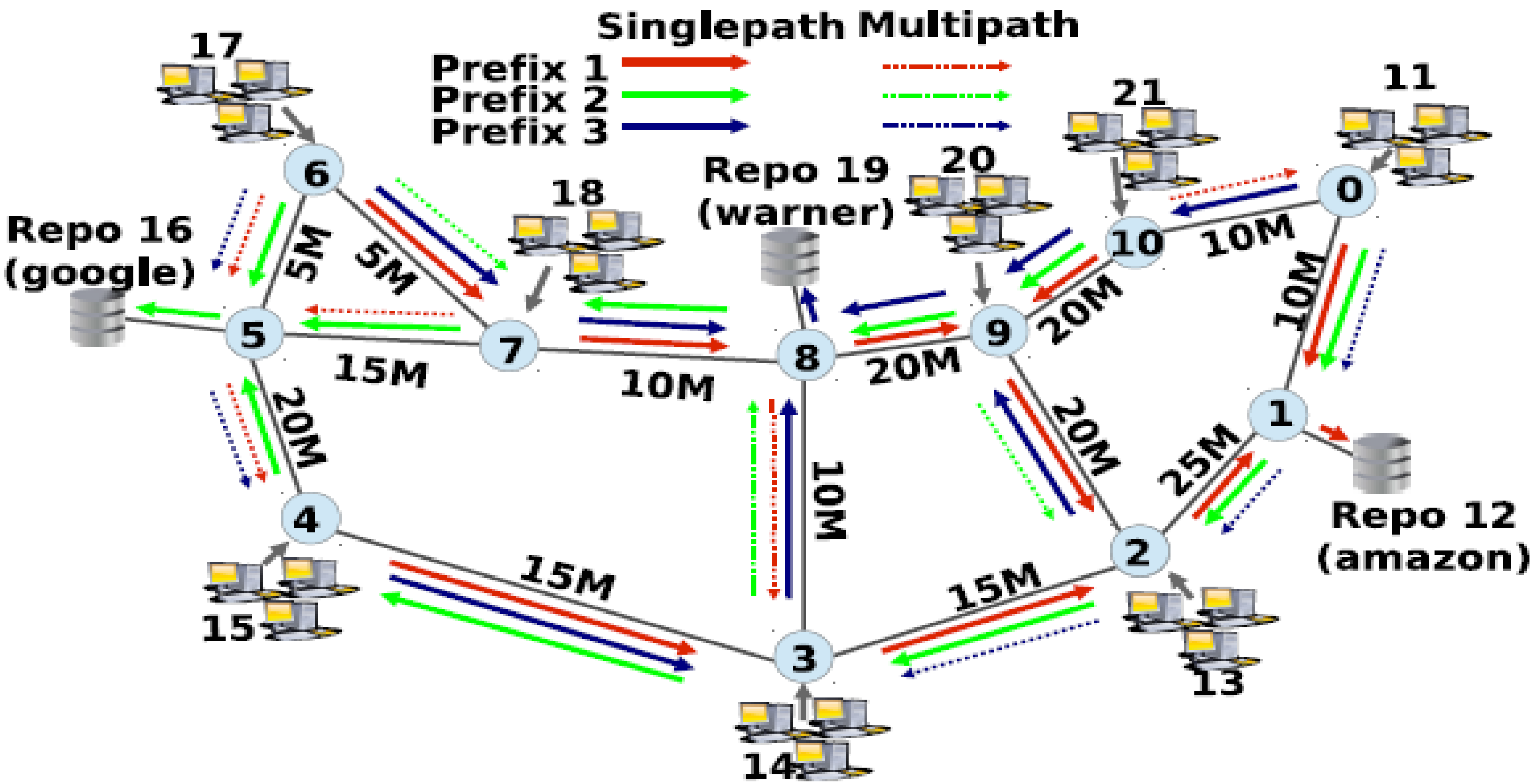
data center for large scale experimentation: the Grid5000



Lyon's Example

Example of cluster in Grid5000

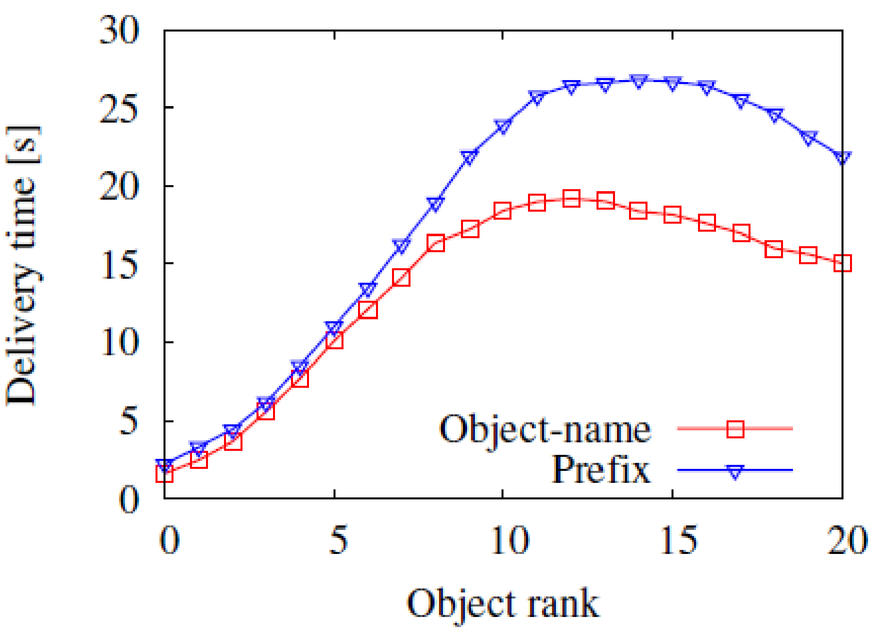
scenario 3

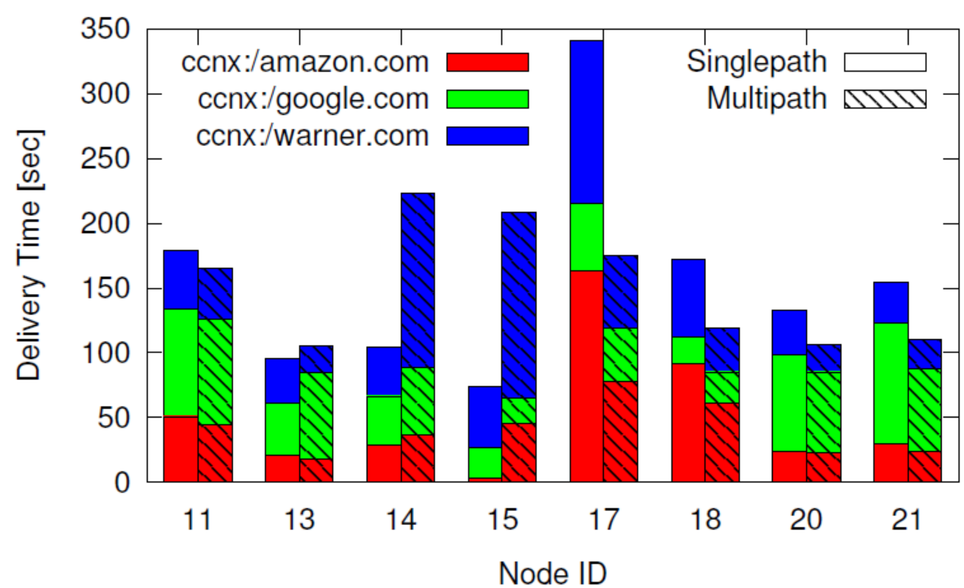


- Dynamic workload: Poisson + Zipf content popularity
- Flow level load: “data object retrieval rate” X “Size” / Capacity < 1
- Three content providers within a name prefix: Amazon, Google, Warner

experimental results – scenario 3

- observations
  - 100% utilization of bandwidth bottlenecks
  - caching benefits downstream bandwidth bottlenecks
  - per-prefix forwarding scalable but suboptimal
- caching benefits for popular content
- multipath benefits for less popular content
- for some content items trade-off between caching and multipath can be reached extracting some name-prefixes from the PIT for the FIB





conclusion and current work

- CCN forwarding plane can be optimized with lightweight protocols
  - the forwarding engine is reach enough to build simple protocols
  - multi-point communication is intrinsically supported by the underlying communication model
- protocol development and large scale experimentation helps improving current prototypes
  - software/hardware prototyping at 40Gbps
  - no high speed with the current software

Questions