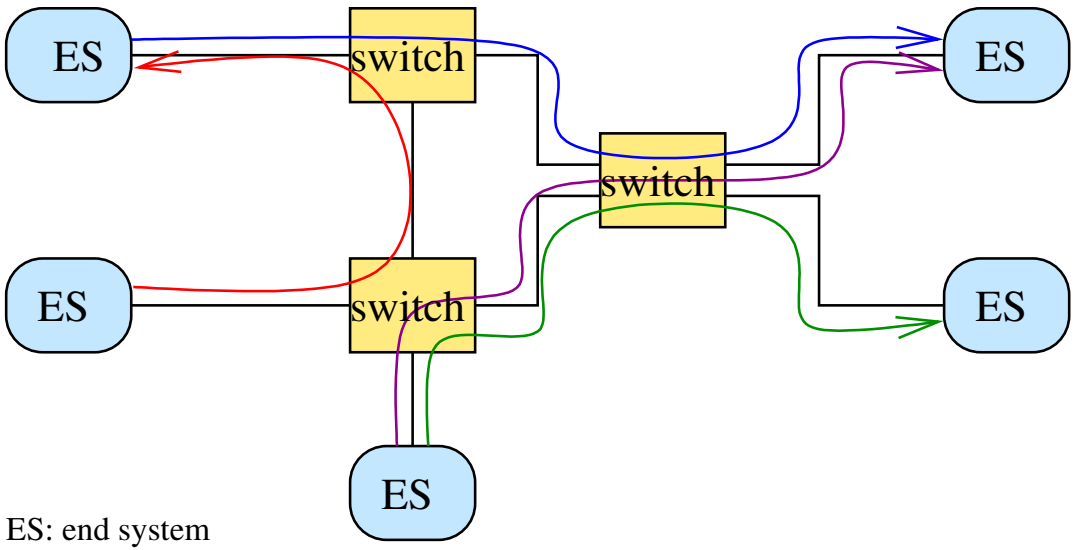
Habilitation à diriger des recherches : Efficiency and algorithms of Network Calculus

Anne Bouillard

École Normale Supérieure

8 avril 2014

AFDX network



AFDX (Avionics Full Duplex):

- 10 switches
- 100 network elements
- 10^4 flows

- Flow crossing the system
  - packet size
  - inter-arrival time (with jitter)
- Network element (server)
  - service rate
  - internal modeling
  - service policy

Objective: deterministic performance guarantees

Compute the maximum time it takes for a packet to cross the system (Worst-case delay)

Network calculus

Real data		**(min,plus) functions**
Real input traffic	$\xrightarrow{\text{abstraction}}$	arrival curve
network element	$\xrightarrow{\text{abstraction}}$	service curve
$\Downarrow$		$\Downarrow$ (min,plus)-operators
Delay	$\xrightarrow{\text{pessimism}}$	Upper bound on the delay

Two kinds of pessimism

1. The abstraction
2. The (min,plus) operations

Network calculus

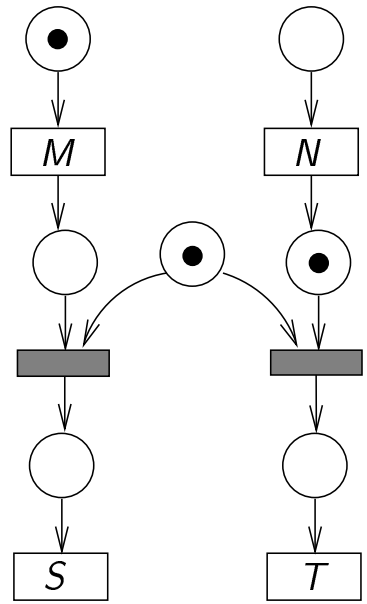
- Theory developed in the 1990's by R.L. Cruz, then developed and popularized by C.S. Chang and J.-Y. Le Boudec.
- Filtering theory in the (min,plus) algebra.
- Applications:
  - Internet: video transmission (VoD),
  - Load-balacing in switches [Birkoff-von Neumann switches, C.S. Chang]
  - Embedded systems: AFDX (Avionics Full Duplex) [Rockwell-Collins software used to certify A380], Networks-on-chip
- Extentions / variations:
  - Real-Time Calculus [L. Thiele, S. Chakraborty]
  - Stochastic network calculus [C.S. Chang, Y.M. Jiang, F. Ciucu, J. Schmitt]

Contributions to network calculus

1. Abstracted model
   - Different ways to abstract the network elements. How do they compare? Is there a unified model? (E. Thierry, L. Jouhet)
   - More accurate description of the input traffic: packet curve (N. Farhi, B. Gaujal)
2. Algorithmic of the (min,plus) operators. Very elegant mathematical modeling. What about their implementation and complexity?
   - Algorithmic toolbox (E. Thierry), implementations: COINC Scilab Toolbox (S. Lagrange) and PEGASE Software (RT@W: N. Navet, J. Migge, L. Fejoz)
   - Fast (min,plus) convolution (E. Faou, M. Zavidovique)
3. Computing exact worst-case delays: get rid of the pessimism introduced by (min,plus) computations
   - Linear programming method to get rid of this pessimism (E. Thierry, L. Jouhet, A. Junier, G. Stea)
   - Complexity class of these problems? (E. Thierry, L. Jouhet)
4. Application of the NC concepts to other fields?
   - Supervision of a flow (A. Junier, B. Ronot)
   - Integrators for the Hamilton-Jacobi equation (E. Faou, M. Zavidovique)

Quality of service in Web-services

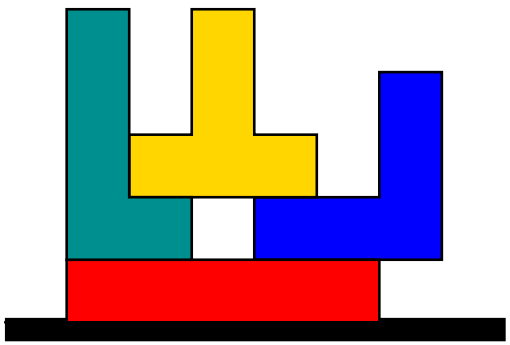
(joint work with S. Rosario, A. Benveniste, S. Haar)



- The monotone systems are those without choice ($\sim$ (max,plus) systems)
- Method to compute the bottleneck by unfolding a Petri net.

Perfect sampling of (max,plus) systems

(joint work with Bruno Gaujal)



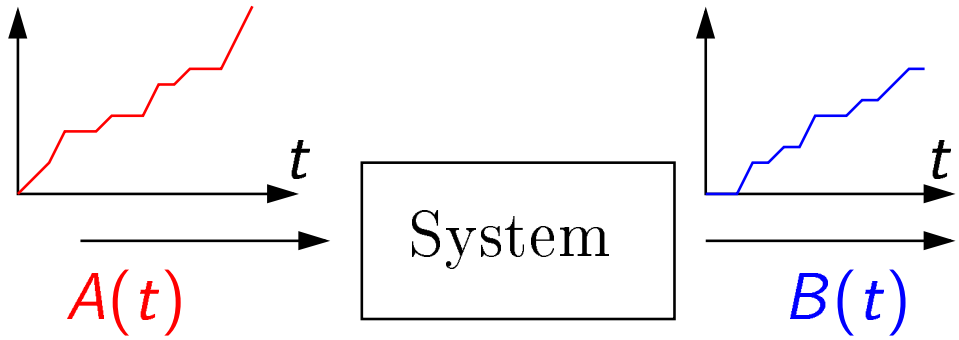
What is the average growth speed of the heap?

Difficult problem when

- The height of the pieces have a probability distribution
- The possible sequences of the pilings form a rational language

Coupling from the past algorithm with (max,plus) matrices for an efficient simulation.

Cumulative processes

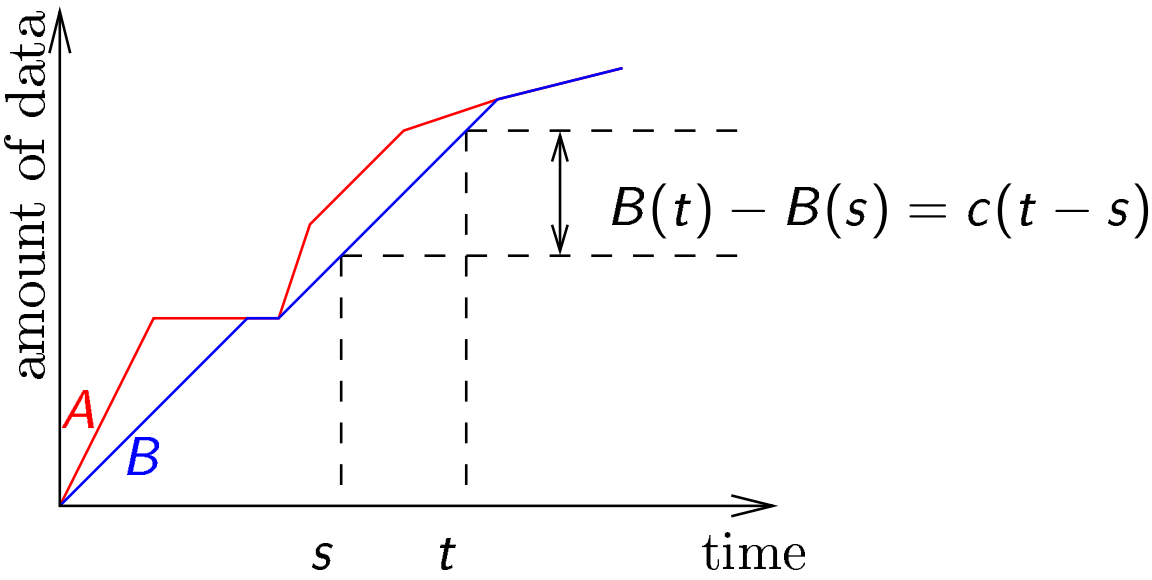


cumulative arrivals at t cumulative departures at t

- $A : \mathbb{R}_+ \to \mathbb{R}_{\min +}$: process of the cumulative arrivals, non-decreasing function
- $B : \mathbb{R}_+ \to \mathbb{R}_{\min +}$: process of the cumulative departures, non-decreasing function
- Causality constraint: $A \geq B$

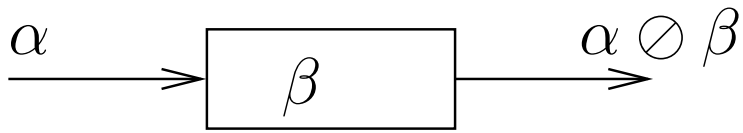
Relation between A and B

Suppose that the system serves c bits of data per unit of time.



$$B(t) = \inf_{0 \leq s \leq t} A(s) + c(t-s) = A * c(t).$$

Network calculus in a slide



Arrival curve $A(t) - A(s) \leq \alpha(t-s)$.

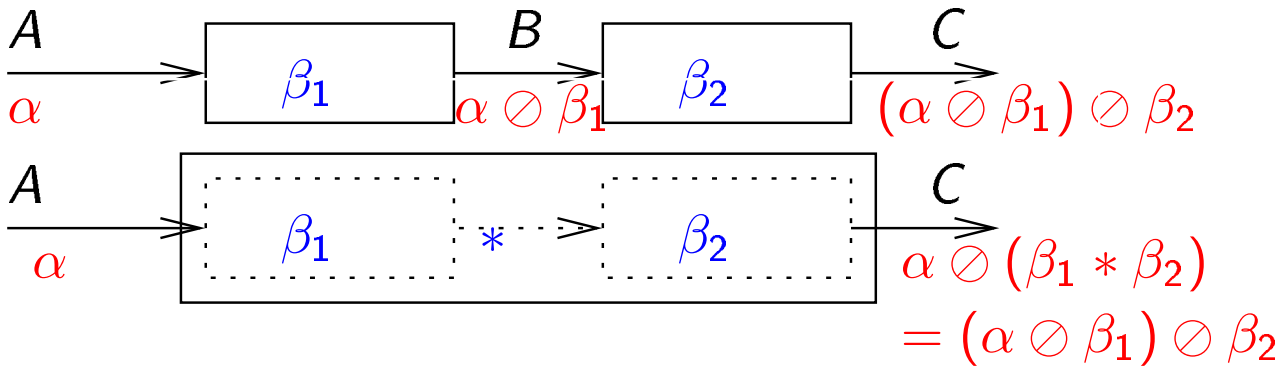
Propagation of constraints

If A is α-constrained and β is a service curve for A, then B is $\alpha \oslash \beta$-constrained.

Service curve $B \geq A * \beta$.

Composition of service curves

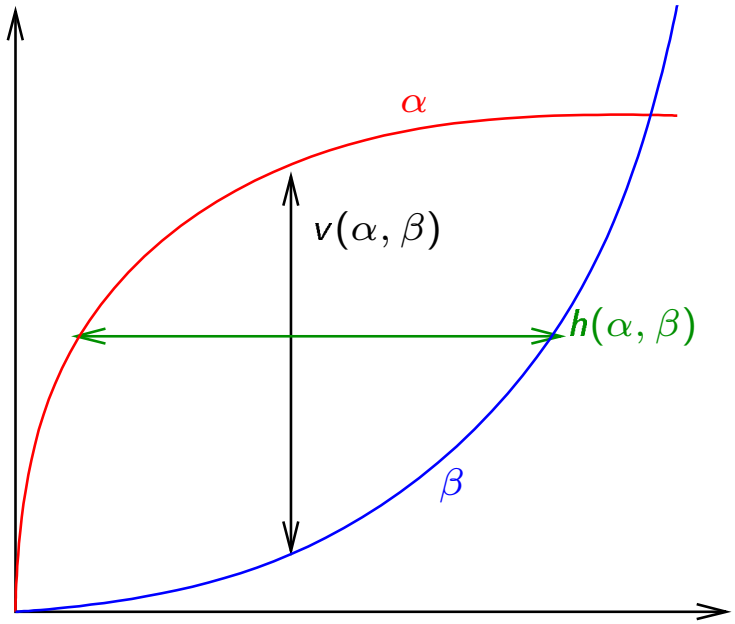
The concatenation of two servers offering respective service curves β_1 and β_2 guarantees a service curve $\beta_1 * \beta_2$ to the flow.



From constraints to performance bounds

Maximum backlog:
$B_{\max} = \sup_{t \geq 0} A(t) - B(t)$

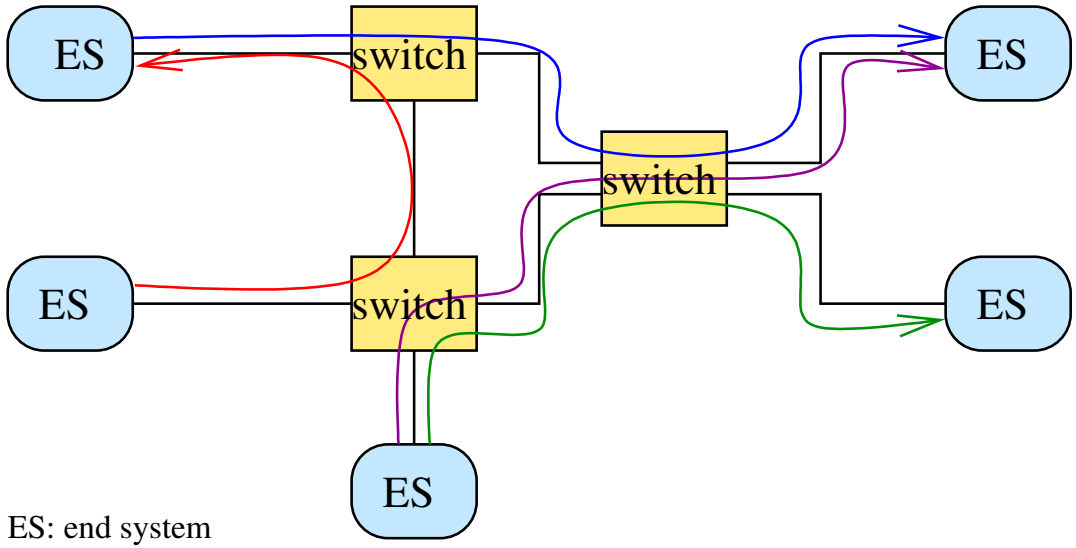
Maximum delay:
$D_{\max} = \inf\{d \mid \forall t \in \mathbb{R}_+,\ B(t+d) \geq A(t)\}$



Performance bounds

- $B_{\max} \leq \alpha \oslash \beta(0) = v(\alpha, \beta) = \sup\{\alpha(t) - \beta(t) \mid t \geq 0\}$
- $D_{\max} \leq h(\alpha, \beta) = \inf\{\forall t \geq 0,\ d \geq 0 \mid \alpha(t) \leq \beta(t+d)\}$ (for FIFO per flow).

Computing performance bounds in a network

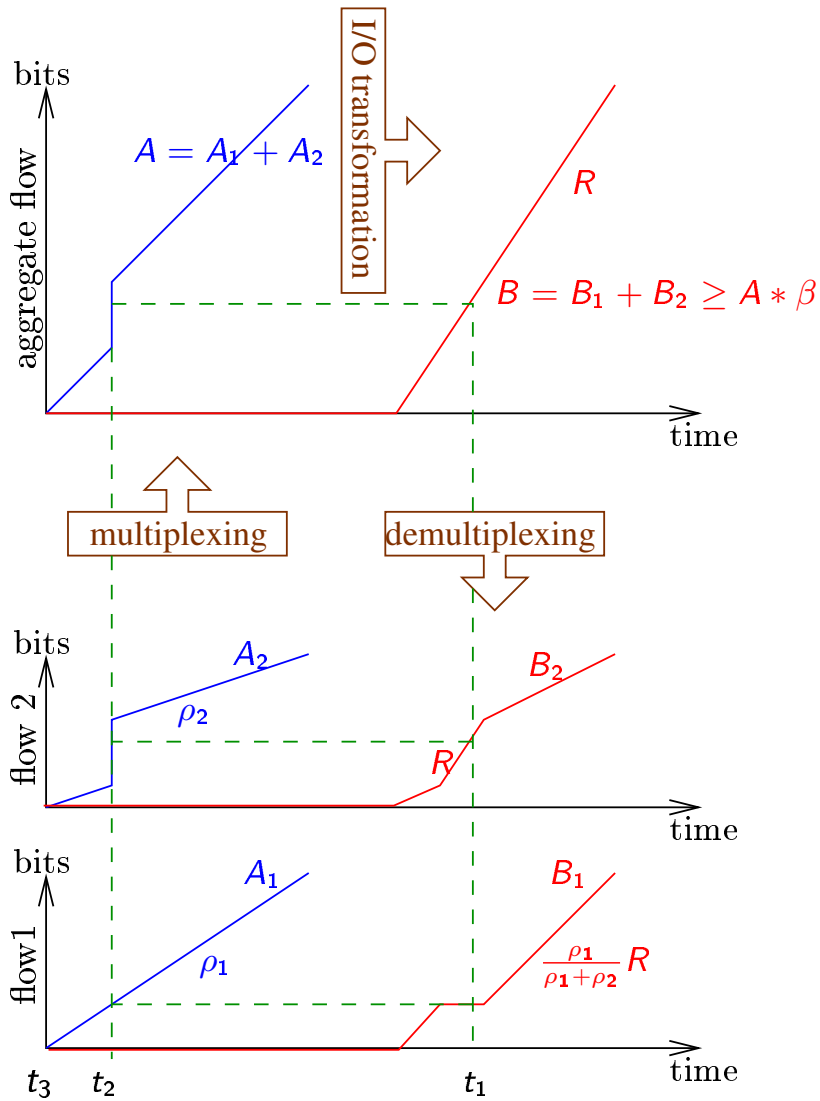


- Each flow i is described by its arrival curve α_i
- Each server h is described by its service curve β_h and a service policy

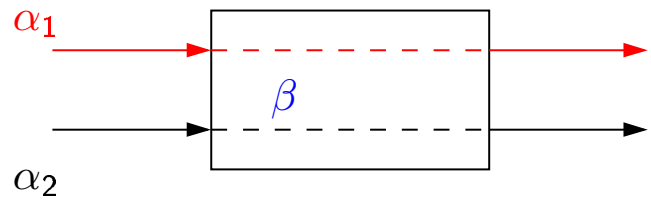
Goal

Compute the maximum time that it takes for a bit of data of a flow to cross the system, when the cumulative processes that respect the constraints given by the curves.

FIFO policy: residual service curves



Multiplexing: $B_1 + B_2 \geq (A_1 + A_2) * \beta$



Theorem (Le Boudec, Thiran, 2001)

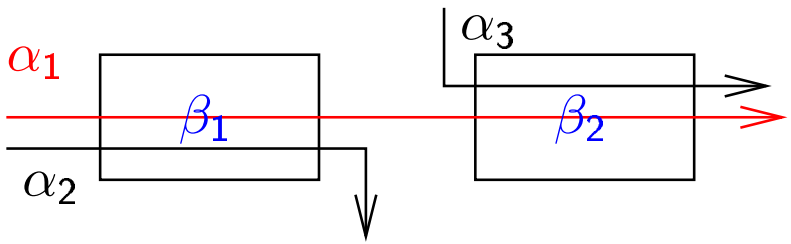
$\forall \theta \in \mathbb{R}_+$, *flow 1 is guaranteed service curves*

$$\beta_\theta^1(t) = [\beta(t) - \alpha_2(t-\theta)]_+ 1_{t>\theta}.$$

Infinite family of service curves that cannot be compared

Composition of FIFO servers

$$\beta^1_\theta(t) = [\beta(t) - \alpha_2(t-\theta)]_+ 1_{t>\theta}.$$



$$\beta^1_{1,\theta_1} * \beta^1_{2,\theta_2}$$

- Concatenation of servers $1, \ldots, N$: how to find the best $(\theta_1, \ldots, \theta_n)$, that minimizes the end-to-end delay of a flow? [Lenzini, Stea *et al.*]
- Is the delay the exact worst-case delay bound? NO.
- Software Deborah based on this approach.

Linear programming and network calculus

Joint work with G. Stea

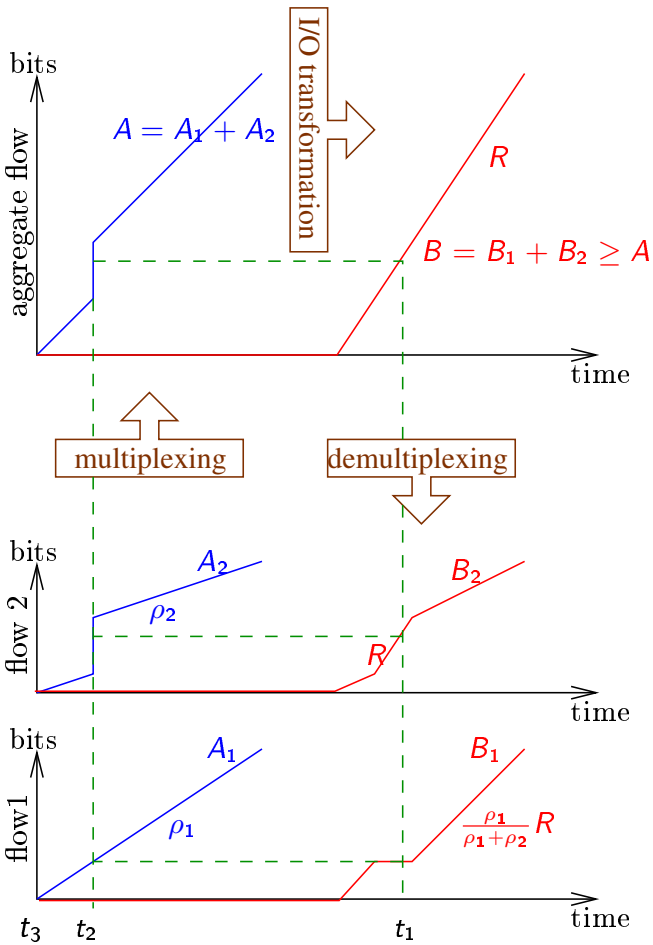
Idea

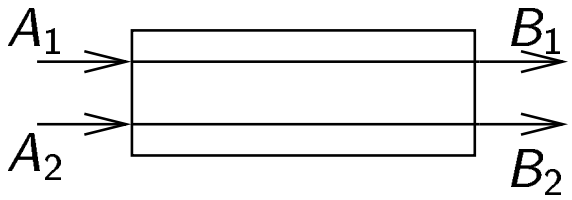
1. Encoding the Network calculus constraints by linear constraints.
2. No computation of residual service curves.

Assumptions

1. Feed-forward network.
2. Concave piecewise affine arrival curves.
3. Convex piecewise affine service curves.
4. Left-continuous cumulative processes.

One server and two flows

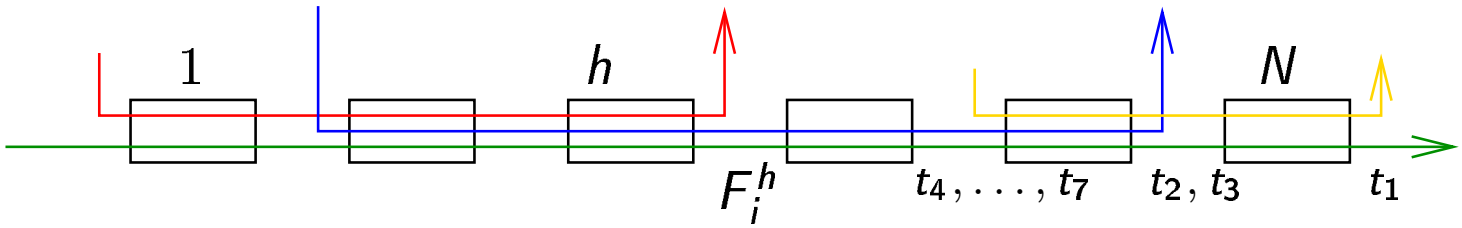


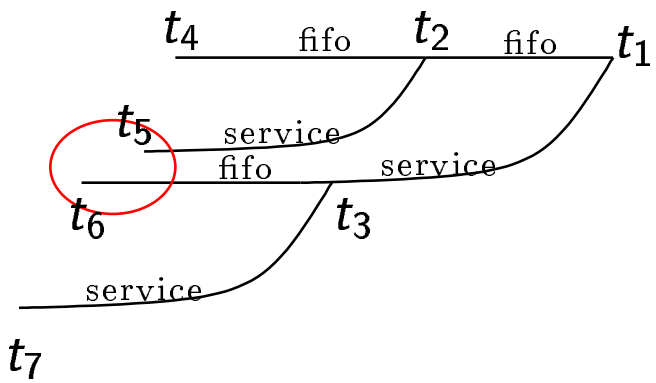


Maximize $t_1 - t_2$ with

- $t_3 \leq t_2 \leq t_1$
- $A_1(t_2) = B_1(t_1)$
- $A_2(t_2) = B_2(t_1)$
- $(B_1 + B_2)(t_1) \geq (A_1 + A_2)(t_3) + \beta(t_1 - t_3)$
- $A_i(t_2) - A_i(t_3) \leq \alpha_i(t_2 - t_3)$
- $A_i(t_3) \leq A_i(t_2)$
- $A_i(x) \geq B_i(x)$

Tandem networks





Variables

dates: $t_1, \ldots, t_{2^{N+1}-1}$, where

- t_{2k} FIFO hypothesis with regards to t_k
- t_{2k+1} service curve constraints with regards to t_k

function values: $F_i^h(t_k)$ *cumulative process of flow i after server h at time t_k*

One Mixed Integer Linear Program (MILP)

We can add Boolean variables ($q \in \{0,1\}$) so that one linear program totally orders the dates and maintains the monotony properties:

- Take M large enough
- for each couple of dates t_i, t_j, add the variable $q_{i,j} \in \{0,1\}$ and the constraints:

$$
\begin{aligned}
t_i + q_{i,j} M &\geq t_j \\
t_j + (1 - q_{i,j}) M &\geq t_i \\
f(t_i) + q_{i,j} M &\geq f(t_j) \\
f(t_j) + (1 - q_{i,j}) M &\geq f(t_i)
\end{aligned}
$$

  for each function f concerning t_i and t_j.

Worst-case delay for a tandem network

$\Delta = \max t_1 - t_{2N}$ with

- Date ordering, monotony
- FIFO hypothesis $(A_1(t_{2k}) = B_1(t_k))$;
- service curve constraint
  $((B_1 + B_2)(t_k) \geq (A_1 + A_2)(t_{2k+1}) + \beta(t_k - t_{2k+1}))$;
- arrival curve $(A_i(t_h) - A_i(t_k) \leq \alpha_i(t_h - t_k))$

Theorem

The worst-case delay for the flow crossing the system is Δ.

Exponential number of variables and constraints, Boolean variables: in practice, this delay is computable for networks with up to 6 servers.

An upper and lower bounds

Upper bound

Relax the monotony and arrival constraints: keep a partial order on the dates.

- Exponential number of dates
- No Boolean variable

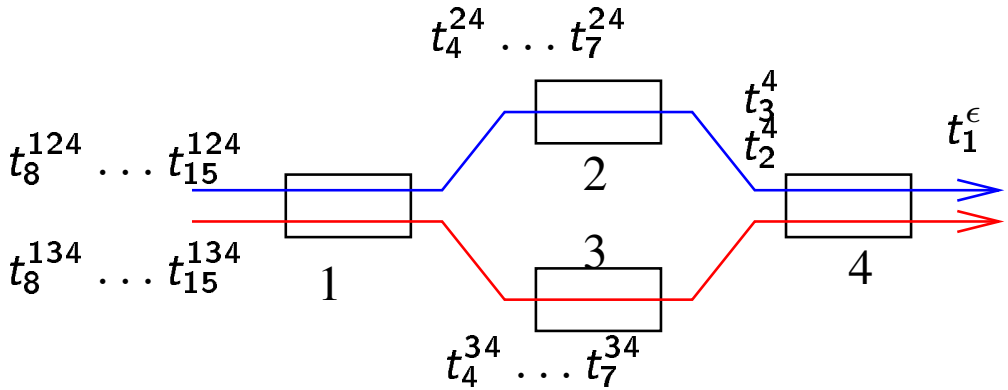
Lower bound

Impose some additional constraints on the dates:

$$t_{2k+1} = t_{2k'+1} \quad \text{for} \quad 2^h \leq k, k' < 2^{h+1} - 1$$

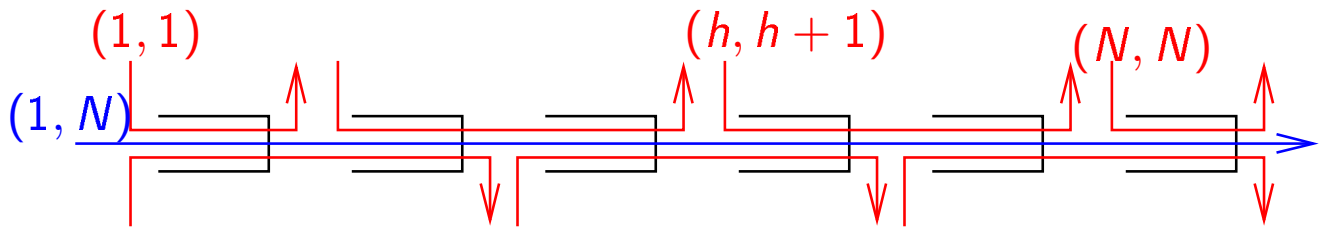
- Quadratic number of dates
- complete order on the dates

Generalization to feed-forward networks

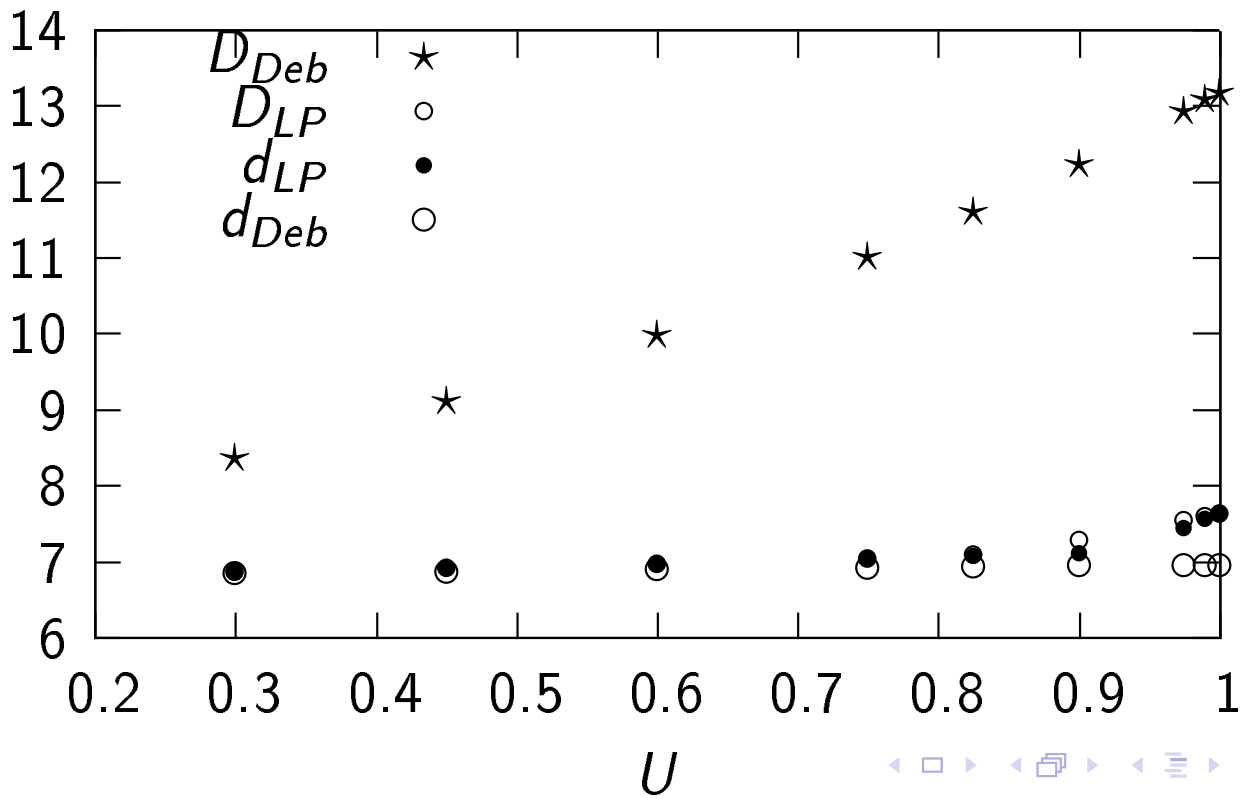


The number of dates increases exponentially again, but the MILP will be similar.

Numerical experiments



$\beta(t) = 10(t-1)_+$, $\sigma(t) = 1 + \rho t$ and $U = 3\rho/10$.



Other results obtained using linear programming

Arbitrary multiplexing [B., Thierry, Jouhet]

- Exact worst-case delay and backlog bounds;
- polynomial algorithm for tandem networks;
- no simplification using (MILP);
- NP-hard problem for general feed-forward networks.

Fixed priorities [B., Junier]

- Improved known bounds;
- exact worst-case only in some limited cases;
- general networks (the priorities break the cyclic dependencies).

Supervision of a flow

Joint work with A. Junier and B. Ronot

Learn the arrival rate of the packets of a flow and detect the changes of the arrival rates over time.

Let (x_n) be the non-decreasing sequence of arrival times of the packets.

What Network calculus usually does:

- Compute the arrival cumulative process: $A(t) = \max\{n \mid x_n \leq t\}$.
- Compute the best arrival curve $\alpha = A \oslash A$

Limitation of this approach:

- Only one curve that describes the flow at all times
- Very loose if there are drastic changes in the arrival rate
- No efficient on-line computation as $\alpha_n(t) = \max(\alpha_{n-1}(t), max\{j \mid x_n - x_{n-j} \leq t\})$.

Idea: use simple curves on finite intervals

Curves on a finite interval

The flow (x_n) is $(\underline{\alpha}, \overline{\alpha})$-*constrained on the interval* $[m, n]$ if $\forall m \leq m' \leq n' \leq n$,

$$\underline{\alpha}(x_{n'} - x_{m'}) \leq n' - m' \leq \overline{\alpha}(x_{n'} - x_{m'}).$$

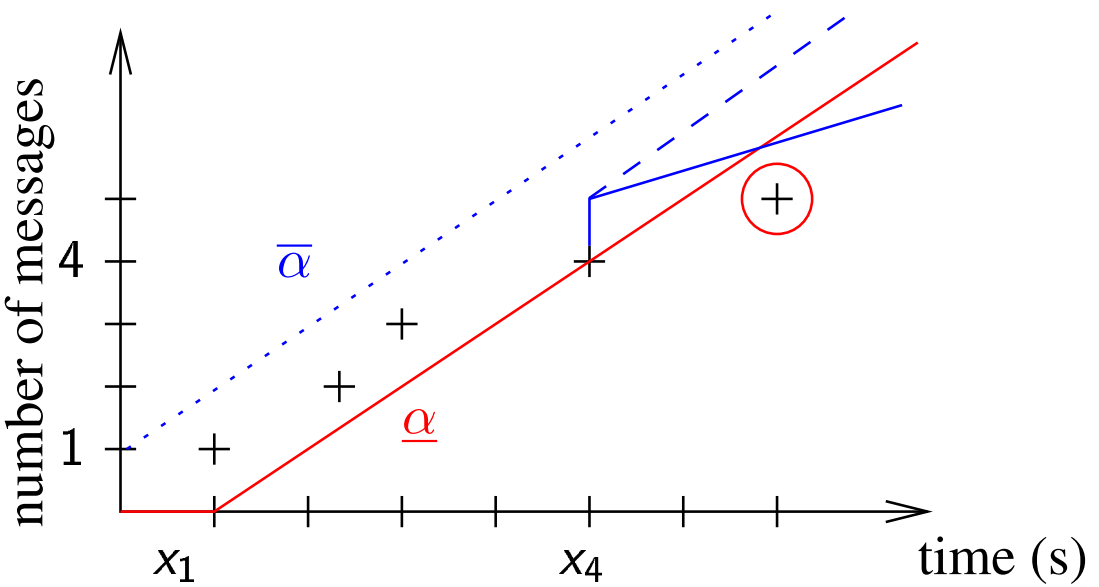
Simple curves

$\overline{\alpha}_{\rho,\sigma} : t \mapsto \sigma + \rho t$

$\underline{\alpha}_{\rho,T} : t \mapsto \rho(t - T)_+$.

constant time complexity

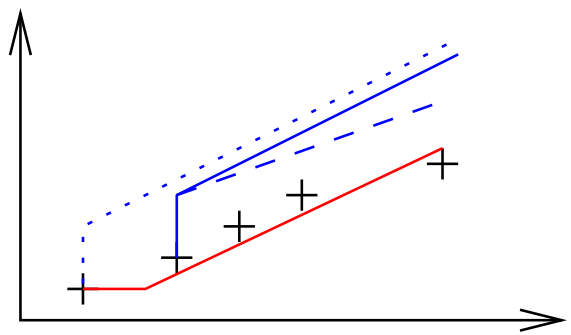
- check the constraints are satisfied
- update the strongest constraint

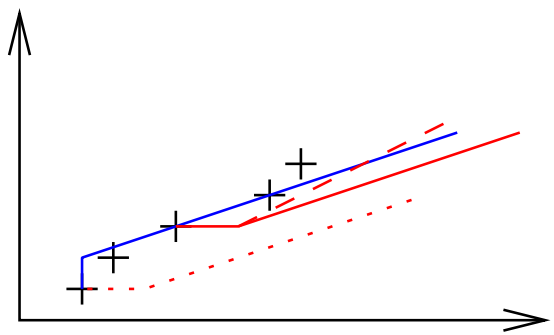


Core algorithm

σ and T are fixed.
At each arrival of a packet

1. Check that its arrival time respects the constraints;
2. If yes, update the strongest constraints
3. Otherwise, compute a new rate.





Convergence of the algorithm

Proposition

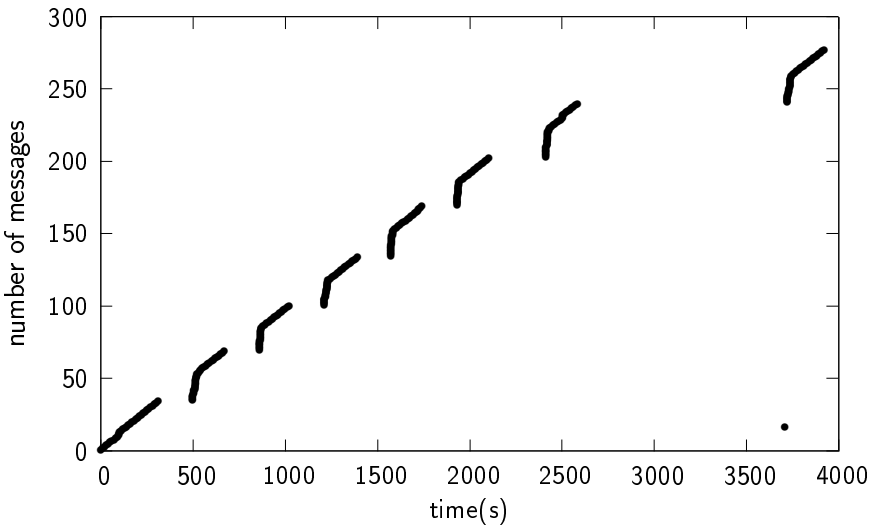
If (x_n) is a periodic and $(\underline{\alpha}_{\rho,T}, \overline{\alpha}_{\rho,\sigma})$-constrained flow, then either the algorithm finds the rate ρ in finite time (and the rate will not be updated anymore) or it ultimately has a periodical behavior.

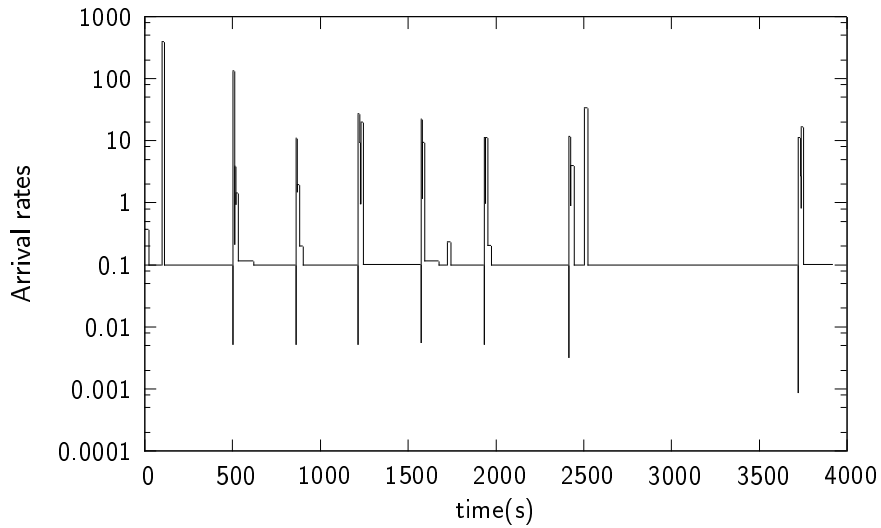
Theorem

If (x_n) is balanced, the algorithm always converges.

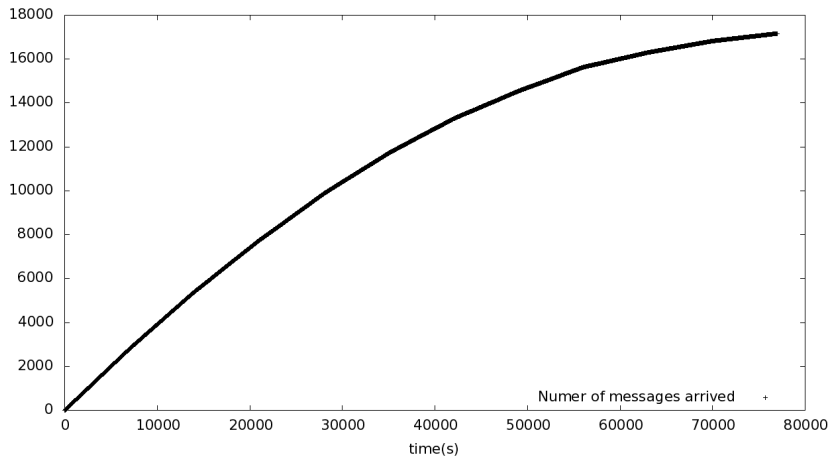
Example of a flow from an OSPF router

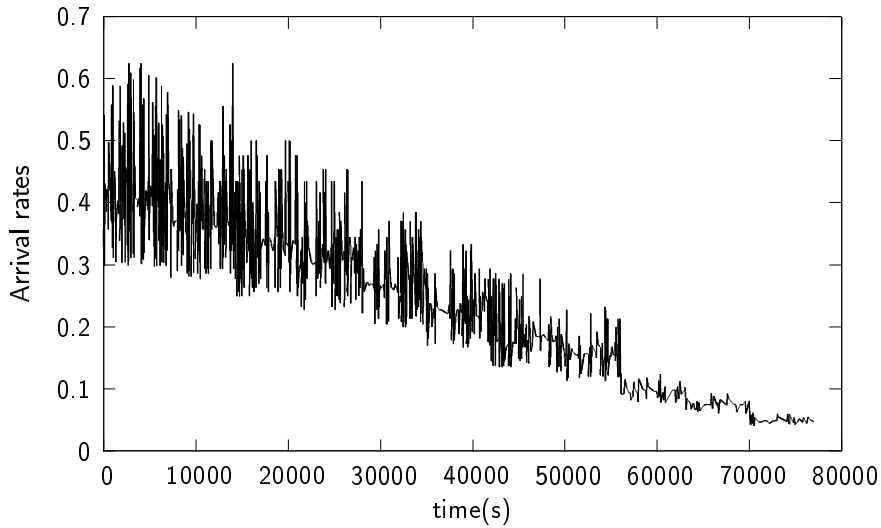
(OSPF: Open Shortest Path First)
Flow of messages received by a router for the OSPF protocol when the router periodically reboots.





Example of a random flow with decreasing arrival rate



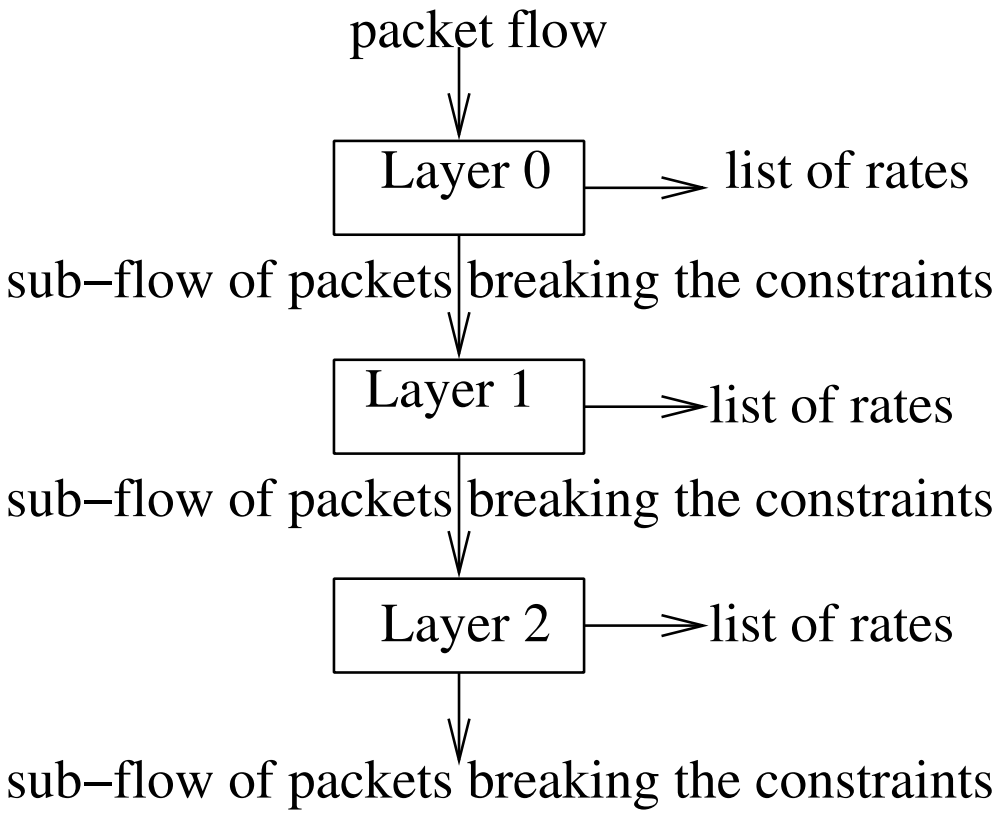


Output: sequence of rates and packets breaking the constraints.

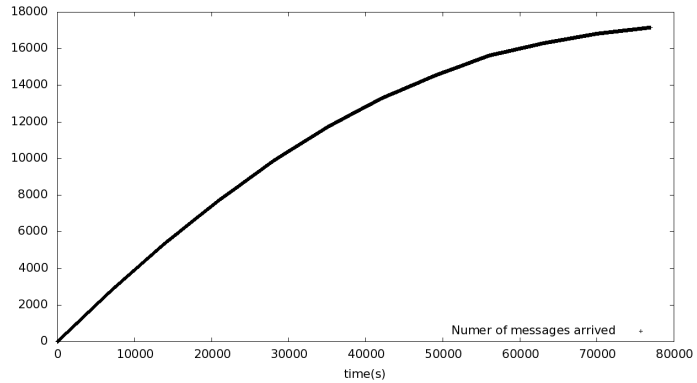
Limitations:

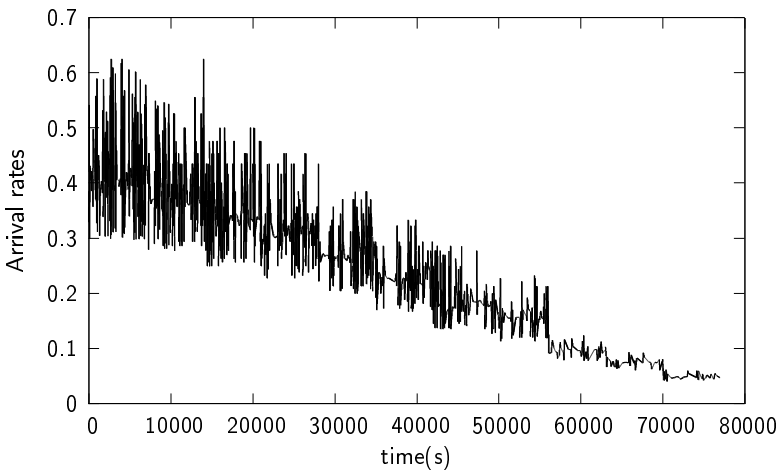
1. σ and T must be fixed in advance
   $\rightarrow$ develop learning/adaptive algorithms
2. Detailed vs. rough view of the flow
   $\rightarrow$ use the algorithm several times.

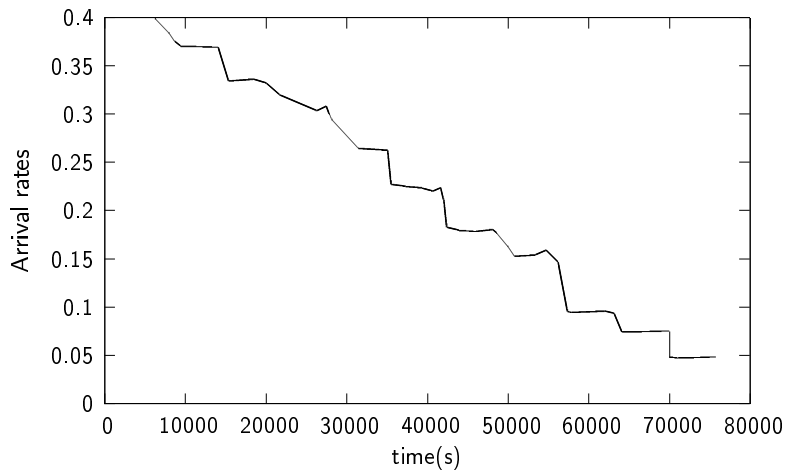
Multilayer algorithm



Numerical results







Conclusion and perspectives

1. Exact worst-case performance in feed-forward networks.
   - Arbitrary multiplexing and FIFO
   - NP-hard problem.
2. Application of the NC concepts to other areas
   - Supervision of a flow
   - Fast (min,plus) convolutions for Hamilton-Jacobi equation intergation
3. Coupling from the past algorithm