Tractability/accuracy trade-offs in FIFO networks

Anne Bouillard
October 8, 2020

HUAWEI

Accuracy/scalability trade-off

Modular analysis

- Total flow analysis
- Separated flow analysis

Properties

- Low complexity (linear/quadratic)
- Potentially very pessimistic

Global analysis

- Linear programming
- Tandem matching analysis
- Deborah for FIFO

Properties

- High complexity ((super-)exponential)
- Quasi-tight bounds

Accuracy/scalability trade-off

Modular analysis

- Total flow analysis
- Separated flow analysis

Properties

- Low complexity (linear/quadratic)
- Potentially very pessimistic

Global analysis

- Linear programming
- Tandem matching analysis
- Deborah for FIFO

Properties

- High complexity ((super-)exponential)
- Quasi-tight bounds

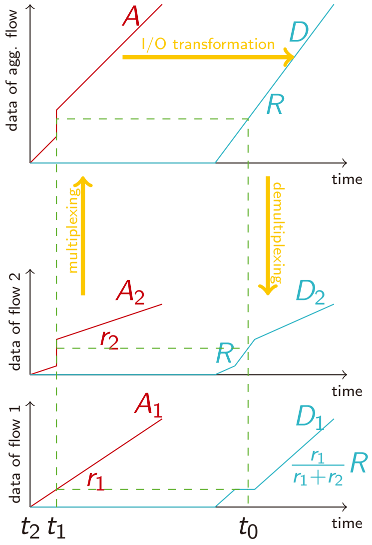
Question

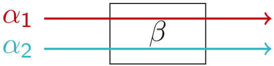
Can intermediate methods be defined? and used for small/medium scale networks?

HUAWEI

Contents

FIFO policy for one server





Theorem

Consider a FIFO server with service curve β, crossed by two flows with respective arrival curves α_1 and α_2. For all $\theta \geq 0$, β_θ is a residual service curve for the first flow, with

$$\beta_\theta = [\beta - \alpha_2 * \delta_\theta]_+ \wedge \delta_\theta.$$

Network model and notations

Servers

- Minimum service curve $\beta_j : t \mapsto R_j(t - T_j)_+$
- (Greedy) shaping curve is $\sigma_j : t \mapsto L_j + C_j t$ ($C_j = \eta_j R_j$ with $\eta_j \geq 1$);

Flows

- Arrival curve: $\alpha_i : t \mapsto b_i + r_i t$
- Path: $\pi_i = \langle \pi_i(1), \ldots, \pi_i(\ell_i) \rangle$
- Arrival curve at server j: $\alpha_i^{(j)} : t \mapsto b_i^{(j)} + r_i t$
- Successor of server j for flow i: $\mathrm{succ}_i(j)$

HUAWEI

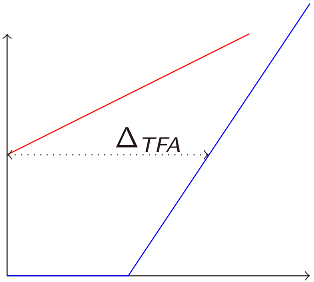
Total flow analysis (TFA)

[Grieux04, Mifdaoui, Leydier17]

Ideas

TFA The worst-case delay in a FIFO server is the same for all flows crossing it

begin
  foreach server j in the topological order **do**
    $b \leftarrow \sum_{i \in \mathrm{Fl}(j)} b_i^{(j)}$;
    $d_j \leftarrow T_j + \frac{b}{R_j}$;
    $b_i^{(\mathrm{succ}_i(j))} \leftarrow b_i^{(j)} + r_i d_j$
  return $\sum_{j \in \pi_i} d_j$



HUAWEI

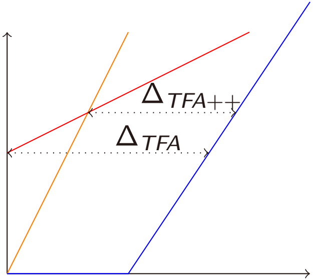
Total flow analysis (TFA)

[Grieux04, Mifdaoui, Leydier17]

Ideas

- TFA The worst-case delay in a FIFO server is the same for all flows crossing it
- ++ The maximum service rate of the previous server shapes the arrival process

begin
  foreach server j in the topological order **do**
    $\alpha \leftarrow \sum_{h \in \text{prec}(j)} \min(\sigma_h, \sum_{i \in \text{Fl}(h,j)} \alpha_i) + \sum_{i \mid \pi_i(1)=j} \alpha_i$;
    $d_j \leftarrow d_{\max}(\alpha, \beta_j)$;
    $b_i^{(\text{succ}_i(j))} \leftarrow b_i^{(j)} + r_i d_j$
  return $\sum_{j \in \pi_i} d_j$



Separated Flow Analysis (SFA)

begin

  foreach server j in the topological order **do**

    foreach flow $i \in \mathrm{Fl}(j)$ **do**

      $b \leftarrow \sum_{k \in \mathrm{Fl}(j)-i} b_k^{(j)}$;

      $b_i^{(\mathrm{succ}_i(j))} \leftarrow b_i^{(j)} + (T_j + b/R_j) r_i$;

      $T_i^{(j)} \leftarrow (T_j + b/R_j) r_i$;

      $R_i^j \leftarrow R_j - \sum_{k \in \mathrm{Fl}(j)-i} r_k$

  return $\sum_{j \in \pi_{i_0}} T_{i_0}^{(j)} + b_{i_0} / (\min_{j \in \pi_{i_0}} R_{i_0}^j)$

A locally optimal choice for the value of θ, minimizing the output burst.

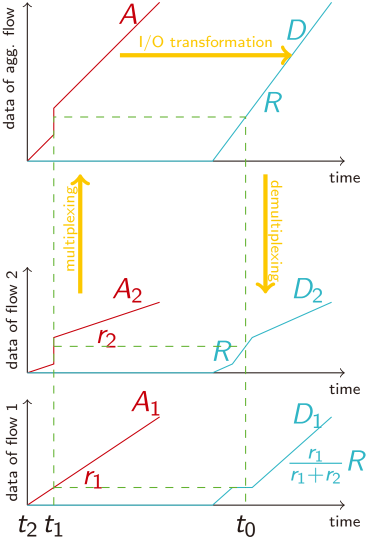
$$\theta = T + b/R$$

if $\beta(t) = R(t - T)_+$
and $\alpha_c(t) = b + rt$

HUAWEI

Linear programming

[B., Stea 12]



Maximize $t_1 - t_2$
Under the constraints

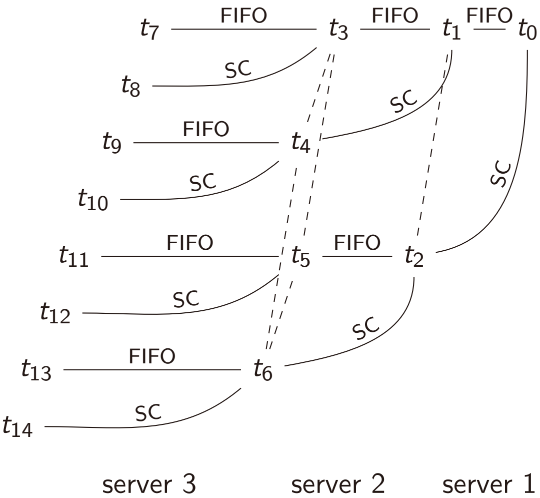
- Dates: $t_0 \geq t_1 \geq t_2$
- Monotonicity: $A_i t_0 \geq A_i t_1, \; i \in \{1, 2\}$
- Arrival: $A_i t_1 - A_i t_2 \leq \alpha_i(t_1 - t_2), \; i \in \{1, 2\}$
- Service: $D_1 t_0 + D_2 t_0 \geq A_1 t_2 + A_2 t_2 + \beta(t_0 - t_2)$
- FIFO: $D_i t_0 = A_i t_1, \; i \in \{1, 2\}$

- Tight performances in feed-forward network
- Super-exponential MILP

Contents

Simplification of the linear program

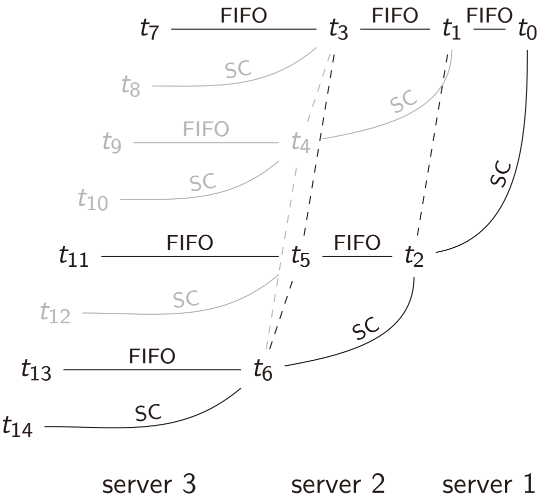
- Number of dates exponential in the number of servers
  - Have to be ordered (with Boolean variables)
  - First approximation: relax monotony constraints and remove Boolean variables
- For each server: monotony, arrival (of new flows)
- Remove service constraints: the number of dates is quadratic



HUAWEI

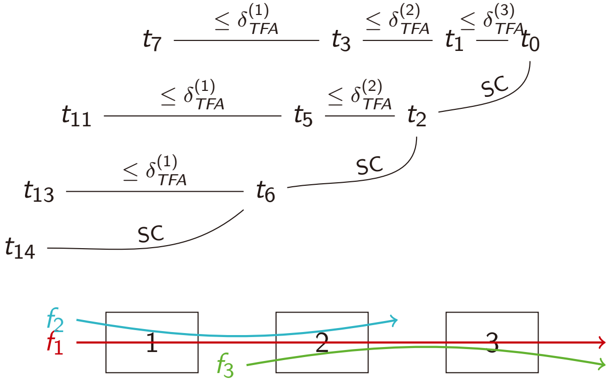
Simplification of the linear program

- Number of dates exponential in the number of servers
  - Have to be ordered (with Boolean variables)
  - First approximation: relax monotony constraints and remove Boolean variables
- For each server: monotony, arrival (of new flows)
- Remove service constraints: the number of dates is quadratic

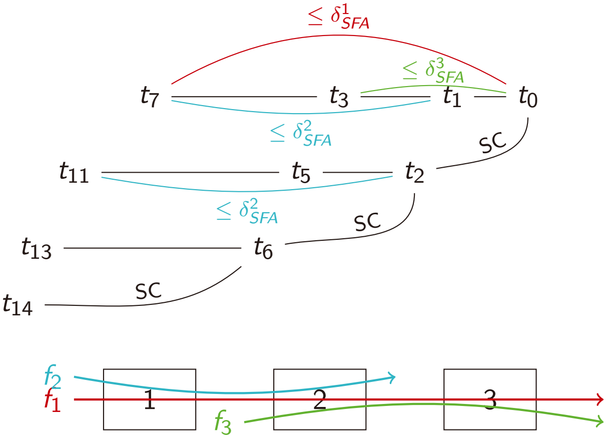


 Huawei

Adding TFA++ constraints

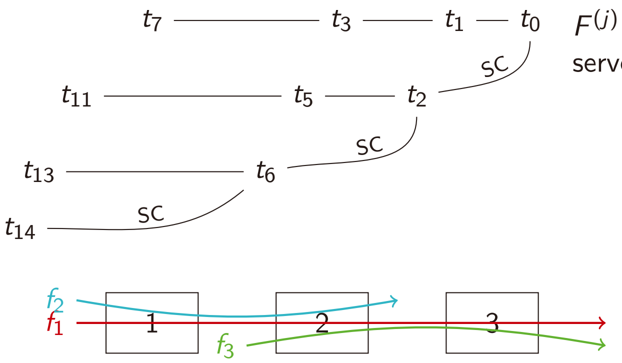


Adding SFA constraints



Add shaping constraints

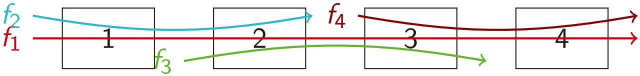
σ_j is a shaper after server j (modeling the maximum service rate of the link for example)

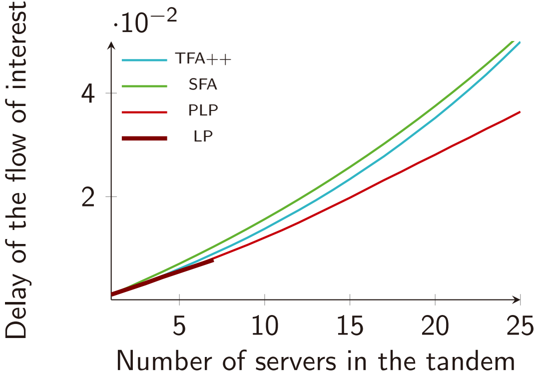


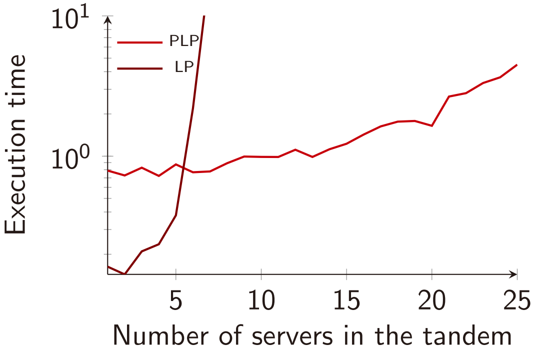
$F^{(j)}$ is the aggregate process entering server j:

$$F^{(2)}(t_3) - F^{(2)}(t_5) \leq \sigma_1(t_3 - t_5)$$
$$F^{(2)}(t_3) - F^{(2)}(t_6) \leq \sigma_1(t_3 - t_6)$$
$$F^{(2)}(t_5) - F^{(2)}(t_6) \leq \sigma_1(t_5 - t_6)$$
$$F^{(3)}(t_1) - F^{(3)}(t_2) \leq \sigma_2(t_1 - t_2)$$

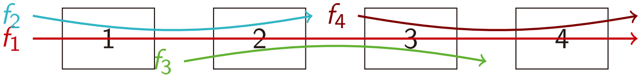
Numerical experiments: two-hop cross-traffic

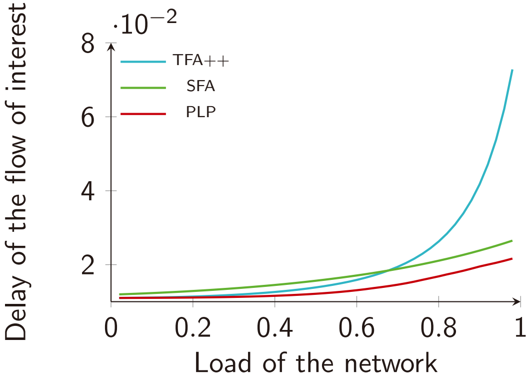


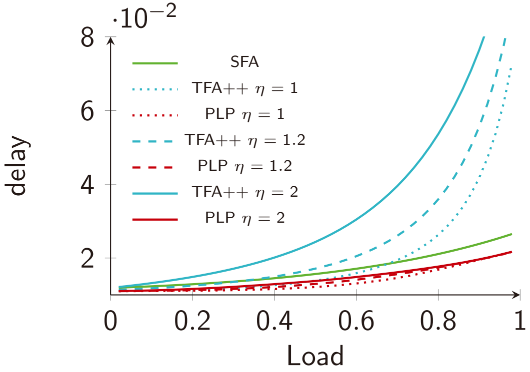




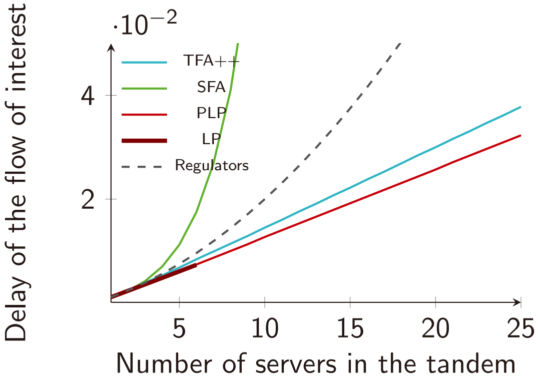
Numerical experiments: two-hop cross-traffic

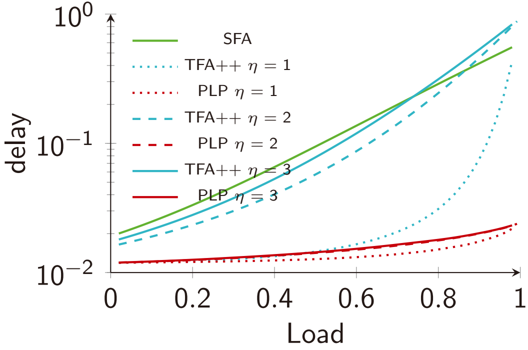






Numerical experiments: source/sink cross-traffic





Contents

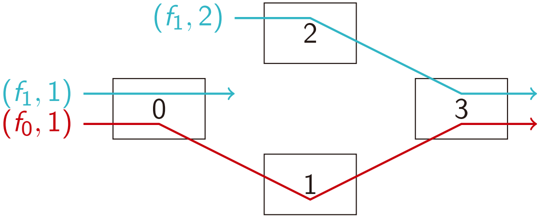
Feed-forward networks: unfolding

Unfolding construction

- One server per path to the sink-node.
- Duplicate the flows along those servers.

HUAWEI

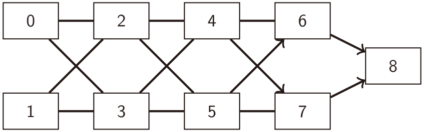
Feed-forward networks: splitting

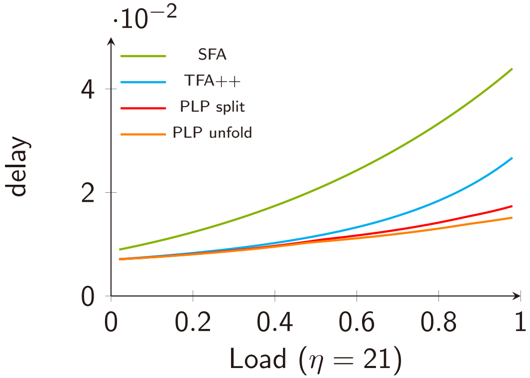


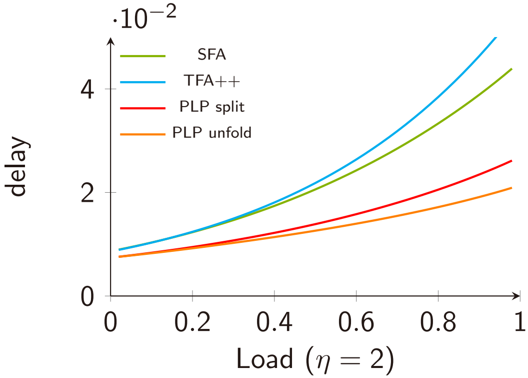
Splitting construction

- Cut the flows to obtain a tree topology
- Compute the arrival curves where the flows are split
- Use maximum service curve as shaping for the "new" arrival flows (but not for the flows of interest)

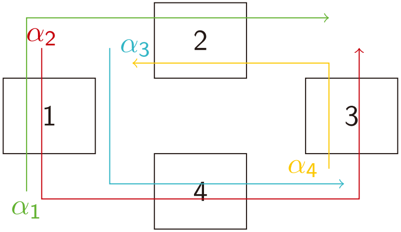
Numerical experiments: mesh network







Network with cyclic dependencies



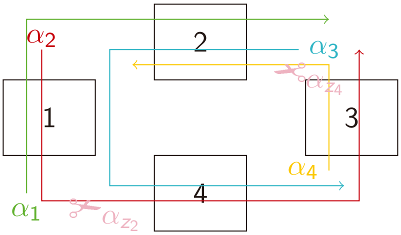
Fix-point equation

$x = (x_z)$ is the vector of the bursts.
For each flow z:

$$\mathcal{L}_z(x) = \max\{A_z(x,y)^t \mid B_z(x,y)^t \leq C_z,\ (x,y) \geq 0\},$$

Solve: $x = \mathcal{L}_z(x)$.

Network with cyclic dependencies



Fix-point equation

$x = (x_z)$ is the vector of the bursts.
For each flow z:

$$\mathcal{L}_z(x) = \max\{A_z(x,y)^t \mid B_z(x,y)^t \leq C_z,\ (x,y) \geq 0\},$$

Solve: $x = \mathcal{L}_z(x)$.

An equivalent formulation for the fix-point

Fix-point equation (1)

$$\sup\{x \mid x \leq \mathcal{L}(x)\} = \sup\{x \mid x_z \leq \max\{A_z(x,y)^t \mid B_z(x,y)^t \leq C_z, (x,y) \geq 0\}\}.$$

Linear program (2)

$$\max\{\textstyle\sum_z x_z \mid x_z \leq A_z(x,y_z)^t, B_z(x,y_z)^t \leq C_z,\ (x,y) \geq 0,\ \text{for all } z \in Z\}.$$

HUAWEI

An equivalent formulation for the fix-point

Fix-point equation (1)

$$\sup\{x \mid x \leq \mathcal{L}(x)\} = \sup\{x \mid x_z \leq \max\{A_z(x,y)^t \mid B_z(x,y)^t \leq C_z, (x,y) \geq 0\}\}.$$

Linear program (2)

$$\max\{\textstyle\sum_z x_z \mid x_z \leq A_z(x,y_z)^t, B_z(x,y_z)^t \leq C_z,\ (x,y) \geq 0,\ \text{for all } z \in Z\}.$$

Theorem

The two following statements are equivalent.

1. x *is the maximal solution of (1).*
2. x *is the vector of variables extracted from the optimal solution of (2).*

An equivalent formulation for the fix-point

Fix-point equation (1)

$$\sup\{x \mid x \leq \mathcal{L}(x)\} = \sup\{x \mid x_z \leq \max\{A_z(x,y)^t \mid B_z(x,y)^t \leq C_z, (x,y) \geq 0\}\}.$$

Linear program (2)

$$\max\{\sum_z x_z \mid x_z \leq A_z(x,y_z)^t, B_z(x,y_z)^t \leq C_z,\ (x,y) \geq 0,\ \text{for all } z \in Z\}.$$

Theorem

The two following statements are equivalent.

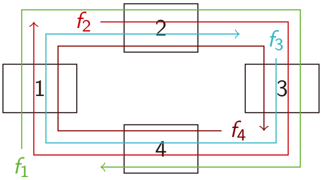
1. *x is the maximal solution of (1).*
2. *x is the vector of variables extracted from the optimal solution of (2).*

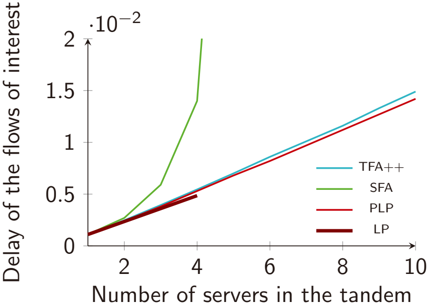
Theorem

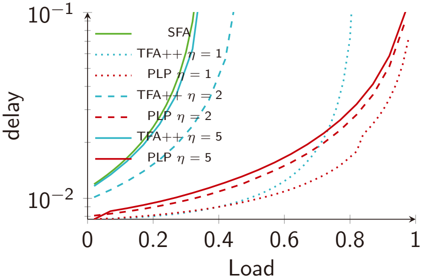
The fix-point of $x = \mathcal{L}(x)$ is unique.

HUAWEI

Numerical experiments







Conclusion

Contributions

- Novel LP approach for FIFO networks offering a trade-off between scalability and accuracy
  1. Polynomial number of constraints
  2. Takes into account the shaping of the link capacity
  3. Generalized to cyclic dependencies
- Comparison with the state of the art
  1. TFA++: accurate for small/medium load and strong shaping
  2. naive SFA: not efficient

Conclusion

Contributions

- Novel LP approach for FIFO networks offering a trade-off between scalability and accuracy
- Comparison with the state of the art

Future work

1. Improve again scalability? can we drop more constraints (arrival/shaping constraints?)
2. Use these methods in large networks: network decomposition
   - What is a good decomposition? (Heuristics, Deep learning?)
   - What is the acceptable size of each component?

Thank you.

Bring digital to every person, home and organization for a fully, connected, intelligent world.

HUAWEI