Distributed Computer Systems Lab
http://disco.informatik.uni-kl.de

Technische Universität Kaiserslautern

Cross-Traffic Arrival Bounds

WoNeCa 2014

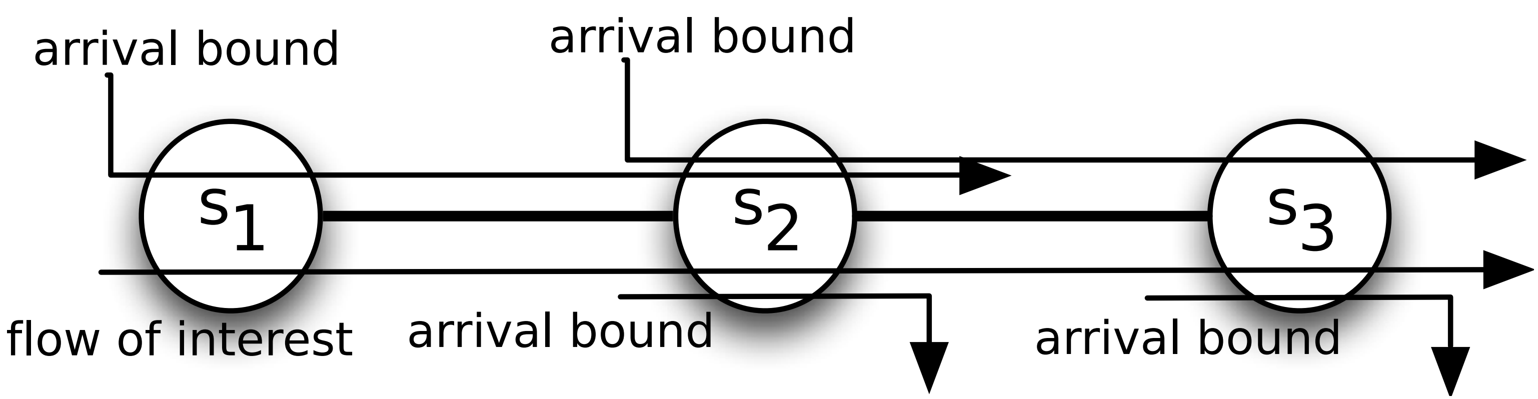
Steffen Bondorf
bondorf@cs.uni-kl.de

Contents

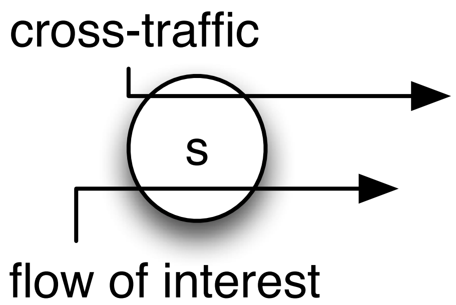
Introduction

- DISCO Deterministic Network Calculator (DiscoDNC)
  - http://disco.cs.uni-kl.de/index.php/projects/disco-dnc
  - Release candidate for version 2.0 available

- Topics of interest
  - Tightness of bounds
  - Large networks
  - Computational effort
  - Cross-traffic arrival bounds
    - i.e., arrival bounds of certain flows at specific locations



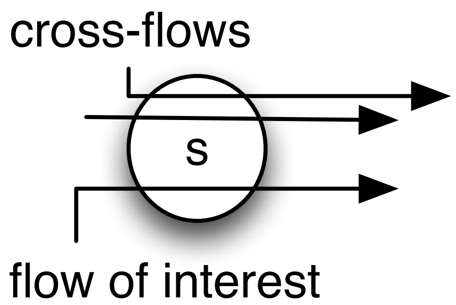
Motivation



$$\beta^{l.o.} = \left[\beta - \alpha^{xtx}\right]^{+}$$

- Separate the flow of interest from its cross-traffic for tight bounds
  - Quantify the cross-traffic arrivals
  - Derive the left-over service curve

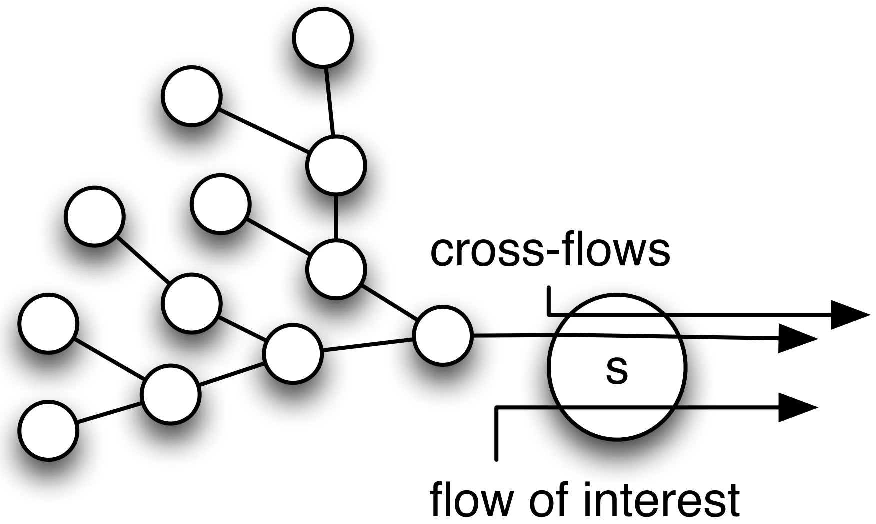
Motivation



$$\beta^{l.o.} = \left[\beta - \sum_{xf \in xtx} \alpha^{xf}\right]^{+}$$

- Cross-traffic usually consists of different cross-flows
  - May be joining from different sub-topologies

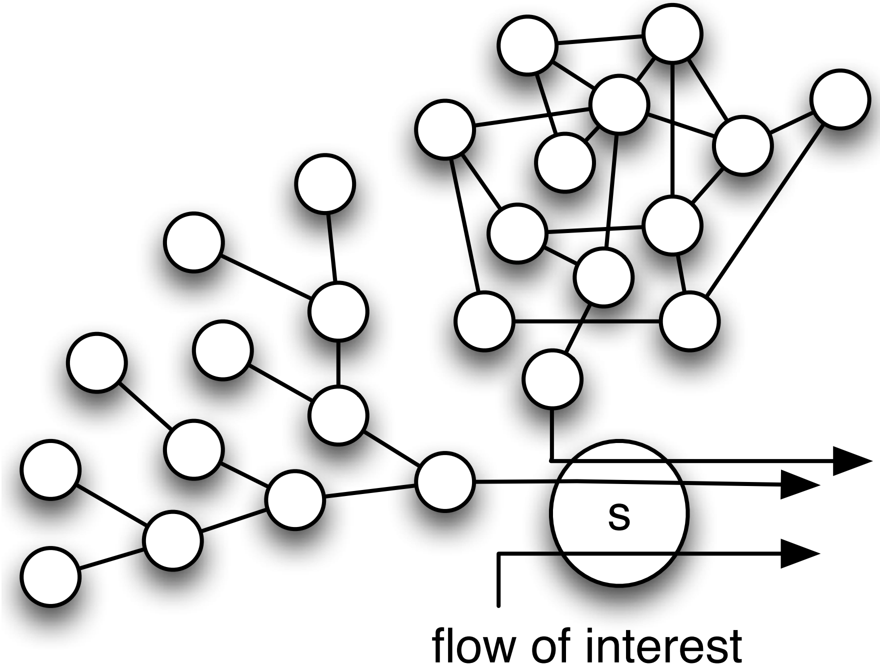
Motivation



$$\beta^{l.o.} = \left[\beta - \sum_{xf \in xtx} \alpha^{xf}\right]^{+}$$

- Cross-traffic usually consists of different cross-flows
  - May be joining from different sub-topologies
    - Trees

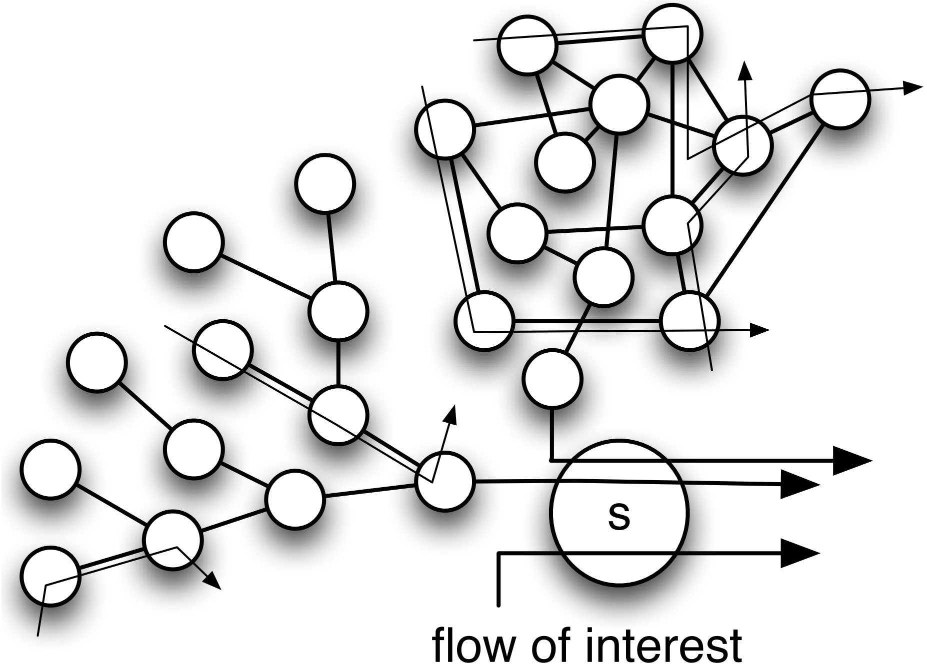
Motivation



$$\beta^{l.o.} = \left[\beta - \sum_{xf \in xtx} \alpha^{xf} \right]^{+}$$

- Cross-traffic usually consists of different cross-flows
  - May be joining from different sub-topologies
    - Trees
    - Arbitrary topologies, flows without cyclic dependencies
  - Concatenation theorem cannot be applied

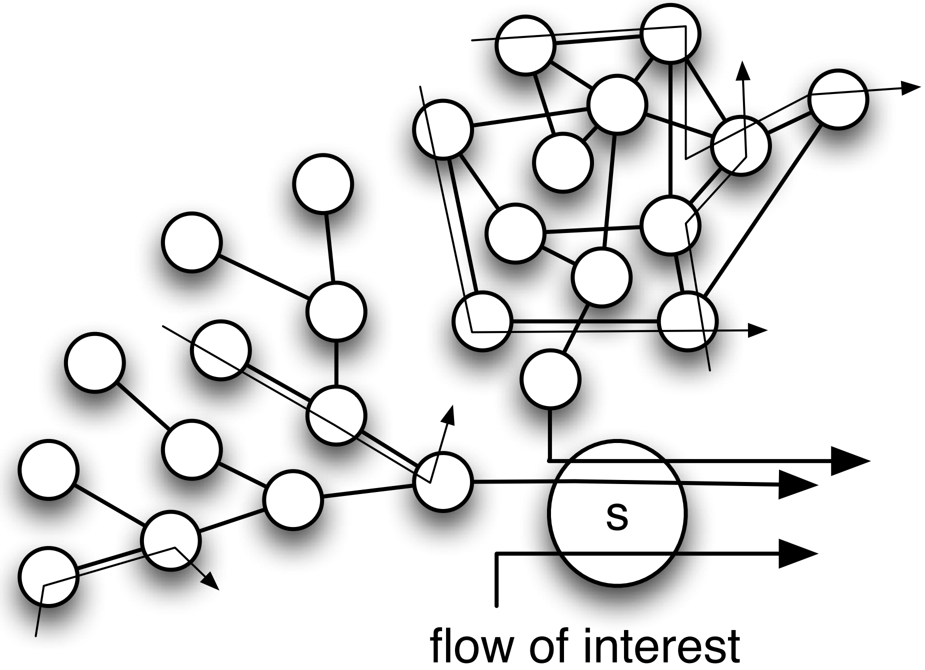
Motivation



$$\beta^{l.o.} = \left[\beta - \sum_{xf \in xtx} \alpha^{xf}\right]^+$$

- Cross-traffic usually consists of different cross-flows
  - May be joining from different sub-topologies
    - Trees
    - Arbitrary topologies, flows without cyclic dependencies
  - Concatenation theorem cannot be applied
  - Cross-traffic of cross-traffic may merge and separate

Motivation



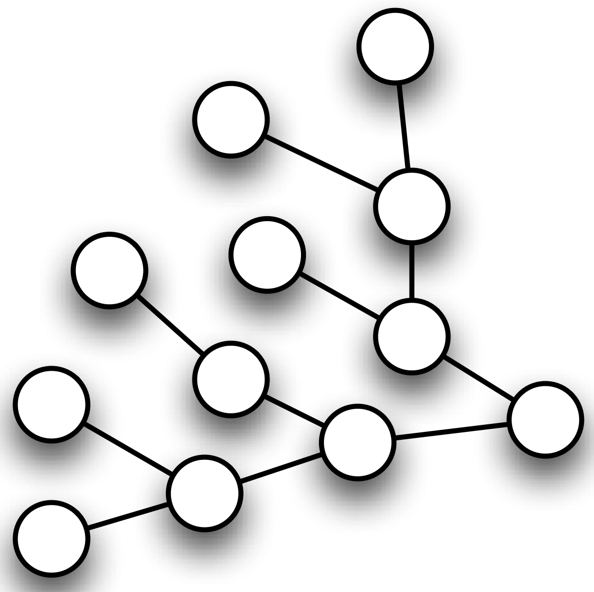
$$\beta^{l.o.} = \left[\beta - \sum_{xf \in xtx} \alpha^{xf} \right]^+$$

- Complex derivation, needs complete topological information
  - Recursive processing of dependencies at servers flows merge, done hop-by-hop
  - Account for aggregation as much as possible in order to get tight bounds
- Limited reusability of results if flow of interest or arrival curve changes
- Huge computational effort to fully analyze a network
  - Usually completed by a central instance in the design phase

Contents

- Introduction and Motivation
- **Recent Work**
- Future Work

Recent Work: Sensor Network Calculus



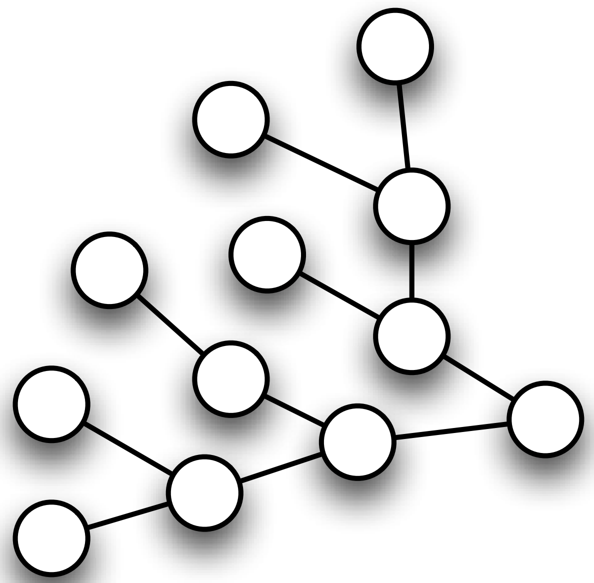
Context

- Sink-tree topology
  - Single sink
  - Flow aggregation
  - No demultiplexing
- Network calculus restrictions
  - Rate-latency service curves
  - Token bucket arrival curves

Objective

- Reduce computational effort
  - Every sensor should calculate bounds
  - Distributed task fulfillment, e.g., monitoring
    - Limited resources!
- Do not compromise tightness

Recent Work: Sensor Network Calculus



Achievement

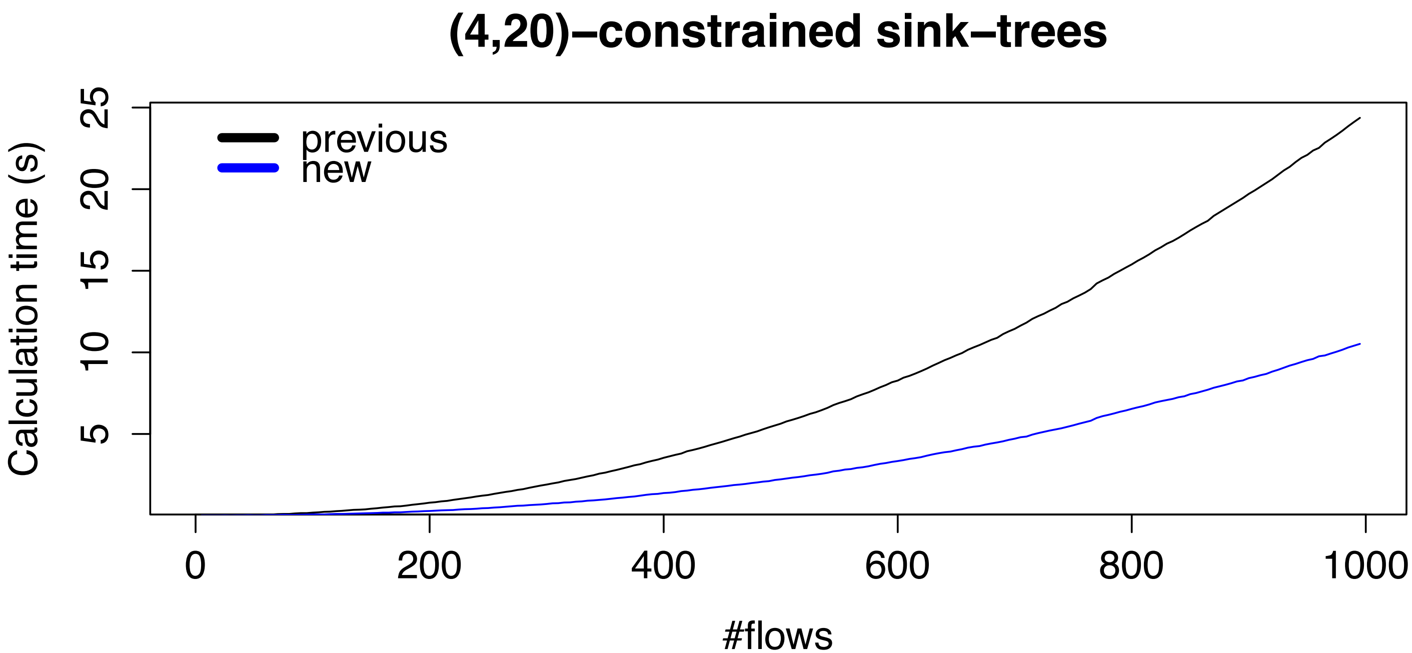
- Derive impact of single cross-flow
  - Virtually separate cross-flows from each other
  - No recursive consideration of cross-flowsx
  - Only iteration over flows' paths
  - Combine per-flow results to cross-traffic result
  - Robustness against parameter change

- Reduced resource demand
- In-network deployment possible
  - Allows for distributed execution
  - Flows carry concatenated service curve instead of letting sensors iterate over their path

Recent Work: Sensor Network Calculus

Exemplary results

- Random (o,d)-constrained sink-trees
  - max o child-nodes per server
  - sink-tree with max depth d
  - every node generates one flow
- Compute all end-to-end delay bounds and backlog bounds

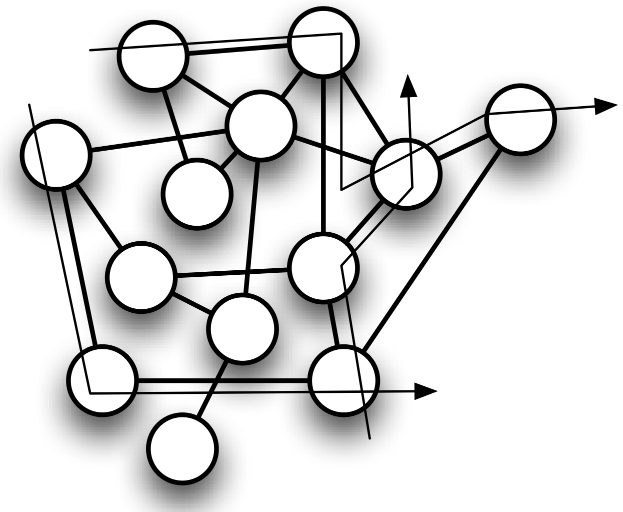


- Best result without trading off tightness

Contents

- Introduction and Motivation
- Recent Work
- **Future Work**

Future Work: Arbitrary Topologies



Problem Setting

- No cyclic dependencies between flows
  - Feed forward property
- Multiple sinks
- Demultiplexing
- Generic solution available in the DiscoDNC, however, …

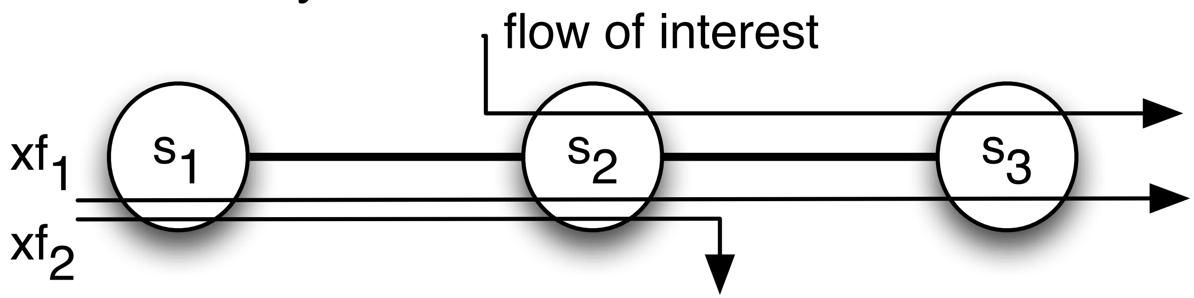
Future Work: Arbitrary Topologies

… there's a problem

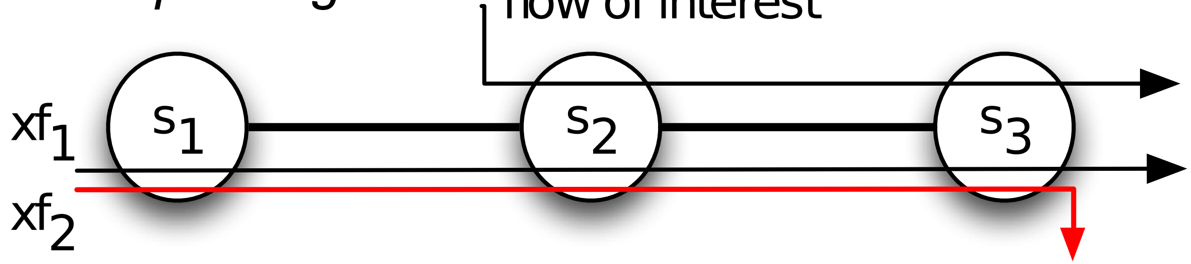
- Usually assume that shifting subtraction as far back as possible results in tighter bounds (referred to as PMOO analysis)
  - Convolve into a single system first
- PMOO demands grouping of cross-flows according to the server they
  a) merge with the flow of interest and
  b) demultiplex from the flow of interest
- Only flows in same group are considered cross-traffic that can be bound as an aggregate
- The analysis chosen for the flow of interest can negatively influence cross-traffic arrival bounds and thus loosen its bounds!
- SFA can outperform PMOO
  - SFA is a per-hop analysis subtracting cross-traffic arrivals first

Future Work: Arbitrary Topologies

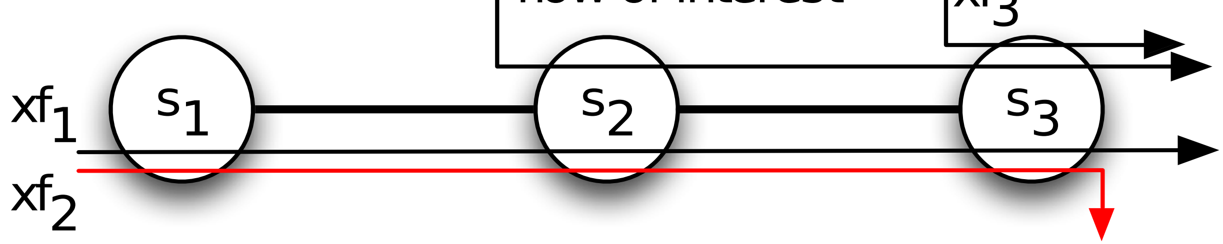
- xf_1 and xf_2 belong to different groups
- At server s_1 they are mutually considered as cross-traffic



- Possible solution: *Flow prolongation*



- But it's no general solution:



Thank you for your attention