

Preserving the Independence of Flows in General Topologies Using Turn-Prohibition

Markus Fidler

Centre for Quantifiable Quality of Service in Communication Systems (Q2S)
Norwegian University of Science and Technology (NTNU)

Oliver Heckmann and Ralf Steinmetz

Multimedia Communications Lab (KOM)
Technical University Darmstadt

Outline

- Motivation: Probabilistic network calculus
 - Importance of independence
 - Statistical multiplexing
- Characterization of dependencies
- Analysis using dependency graphs
- Strategies to avoid dependencies
 - Link-prohibition
 - Turn-prohibition
- Example: German Research Network (DFN G-WiN)

Probabilistic network calculus

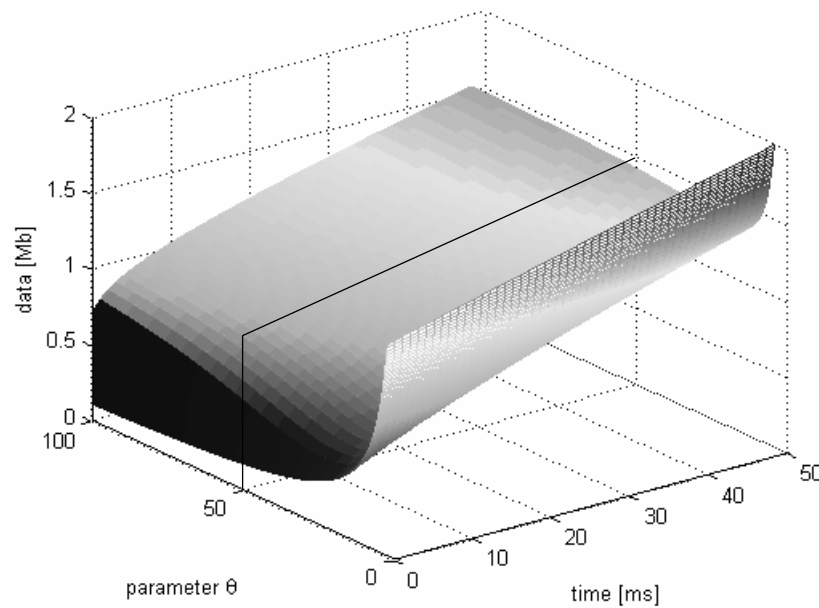
- Network calculus
 - Deterministic delay bounds, i.e. **all** packets arrive within 100 ms
 - Often too conservative since the worst-case is considered
- Probabilistic network calculus
 - Statistical delay bounds, i.e. 99.9 % of the packets arrive within 10 ms
 - Loosing a small percentage usually provides a huge gain
 - Non-exhaustive and exemplary list of relevant approaches
 - Stochastic (σ, ρ) -calculus [Chang, 1994]
 - Effective Envelopes [Boorstyn, Burchard, Liebeherr, Ottomakorn, 2000]
 - Exponentially bounded burstiness [Yaron, Sidi, 1993]
 - Stochastically bounded burstiness [Starobinski, Sidi, 2000]
 - Generalized stochastically bounded burstiness [Yin, Jiang, Jiang, Kong, 2002]

Multiplexing models

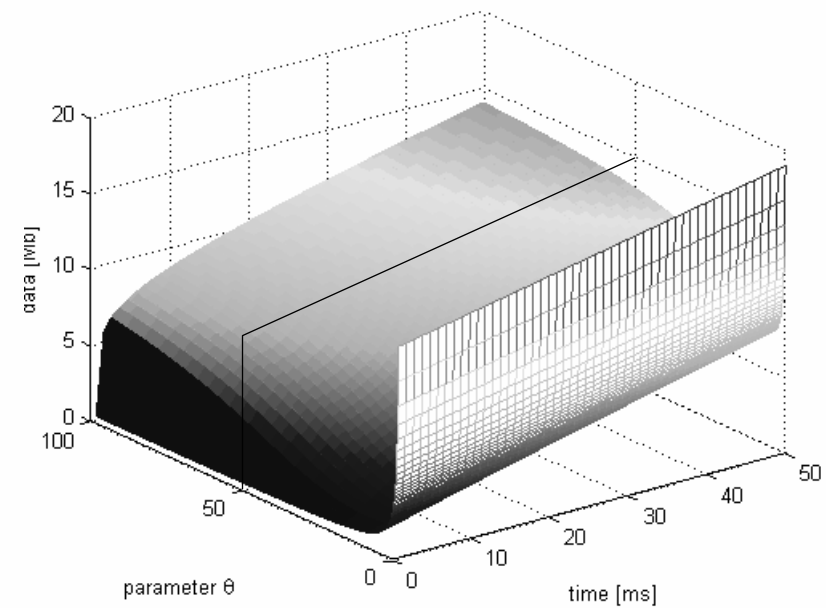
- Deterministic/general multiplexing
 - Joint worst-case analysis
 - Accumulation of bursts for the traffic aggregate
 - Statistical multiplexing
 - Requires independence of flows
 - Smoothing of bursts for the traffic aggregate
 - Fundamental in network performance analysis
 - The main reason for probabilistic network calculus
 - Today: Widely used simplifying assumptions
 - Optimistic: Independence everywhere → Bounds are not rigorous
 - Pessimistic: Dependence everywhere → Bounds are not tight
- Required: Realistic model with exact knowledge about dependencies

Effective envelopes

- Multiplexing of n leaky bucket sources
 - Burst size $b = 0.1$ Mb
 - Sustainable rate $r = 1$ Mb/s
 - Violation probability $\varepsilon = 10^{-6}$



$n = 10$ sources

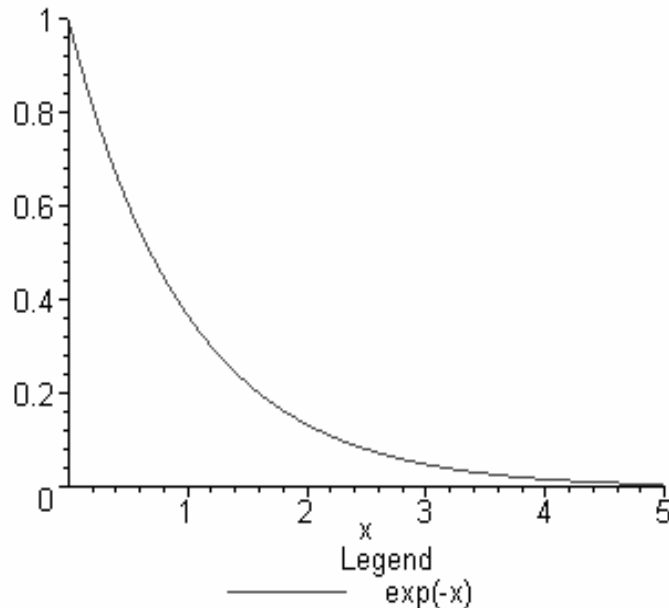


$n = 100$ sources

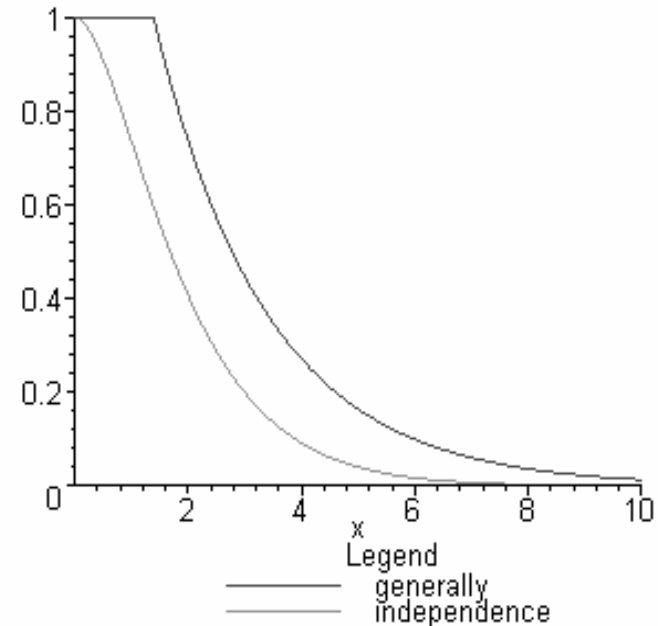
Stochastically bounded burstiness

- Multiplexing of two sources with
 - Exponential bound on the burstiness
 - Given sustainable rate

gSBB bounding function for one flow



gSBB bounding function for two multiplexed flows

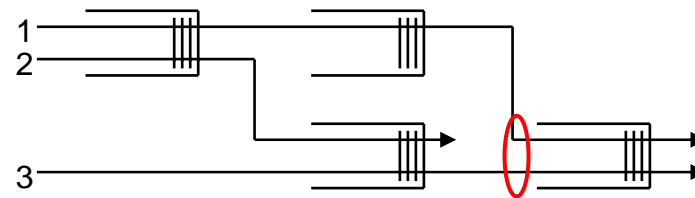
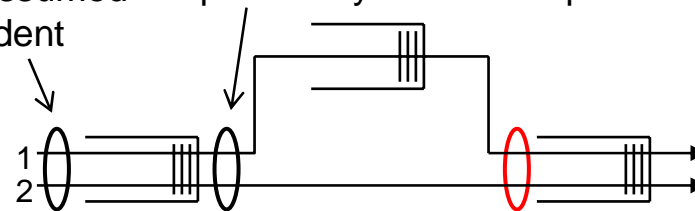


Types of dependencies

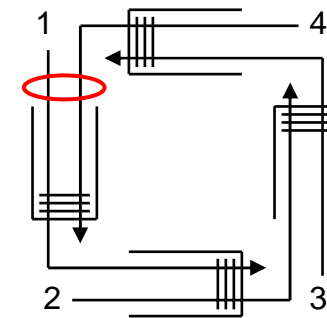
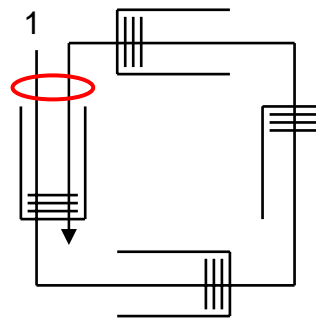
- Plain
 - Direct
 - Indirect

Initially flows are assumed to be independent

Whenever flows pass a common queue they become dependent



- Cyclic
 - Direct
 - Indirect

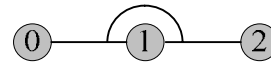


Cyclic dependencies

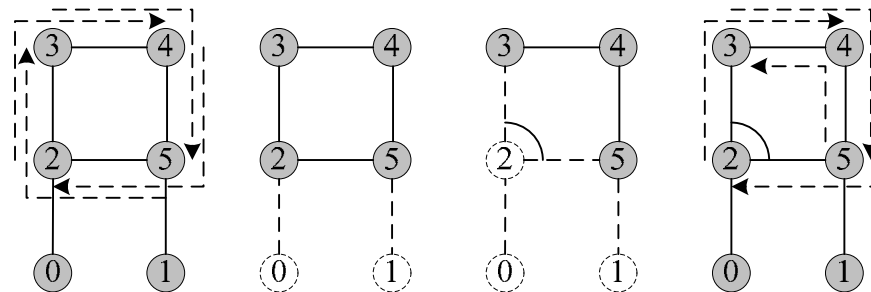
- Direct
 - Not an issue in case of loop-free routing
- Indirect
 - Quite challenging
 - No inductive application of network calculus/queuing theory
 - Cases of instability even if the load $\rho < 1$
 - Families of networks with arbitrary large delays if a certain, usually small utilization threshold is exceeded [Charny, Le Boudec, 2000]
 - In practice synchronization and bad performance possible

Turn-prohibition

- Turn-prohibition [Starobinski et al., 2002]
 - Generally breaks all cyclic dependencies while preserving connectivity
 - Turn: Concatenation of two successive links



- Algorithm in brief
 - Visit nodes in the order of their degree
 - Prohibit all turns around the current node
 - Remove the current node and all connected links



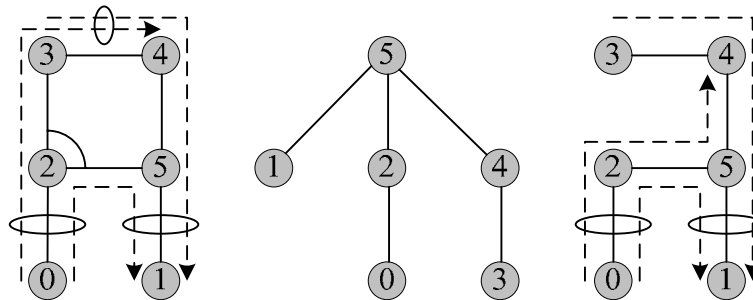
Plain dependencies

- Direct
 - Only in case of multi-path routing

- Indirect
 - Impact varies significantly for different topologies
 - Not always meaningful to avoid all dependencies of this type
 - Required
 - Method for analysis
 - Tools for avoidance
 - Individual choice to either
 - Apply more general deterministic multiplexing where necessary or
 - Modify routing to ensure independence and apply statistical multiplexing

Link-prohibition

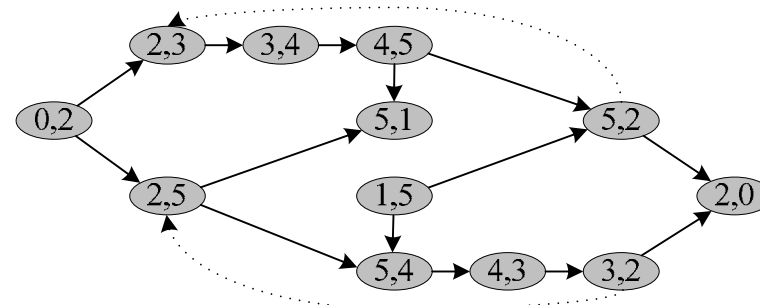
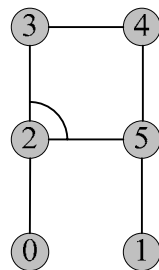
- Trivial solution: Spanning tree



- Huge performance impact due to disabled links
- Better: Extending turn-prohibition

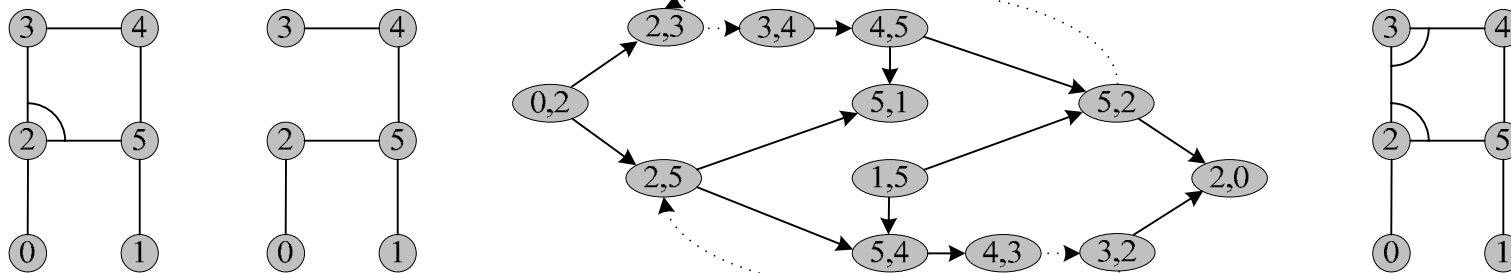
Dependency graphs

- Known from deadlock analysis [Duato 2003]
- Network transformation
 - Links become nodes
 - Turns become links
- Resulting graphs describe the dependencies in queuing networks
 - Each node represents a link respective queuing and scheduling unit
 - Traffic that flows from one queue to another creates a dependence of the downstream queue on the respective upstream queue
 - Need to avoid that more than one path from any queue to any other exists

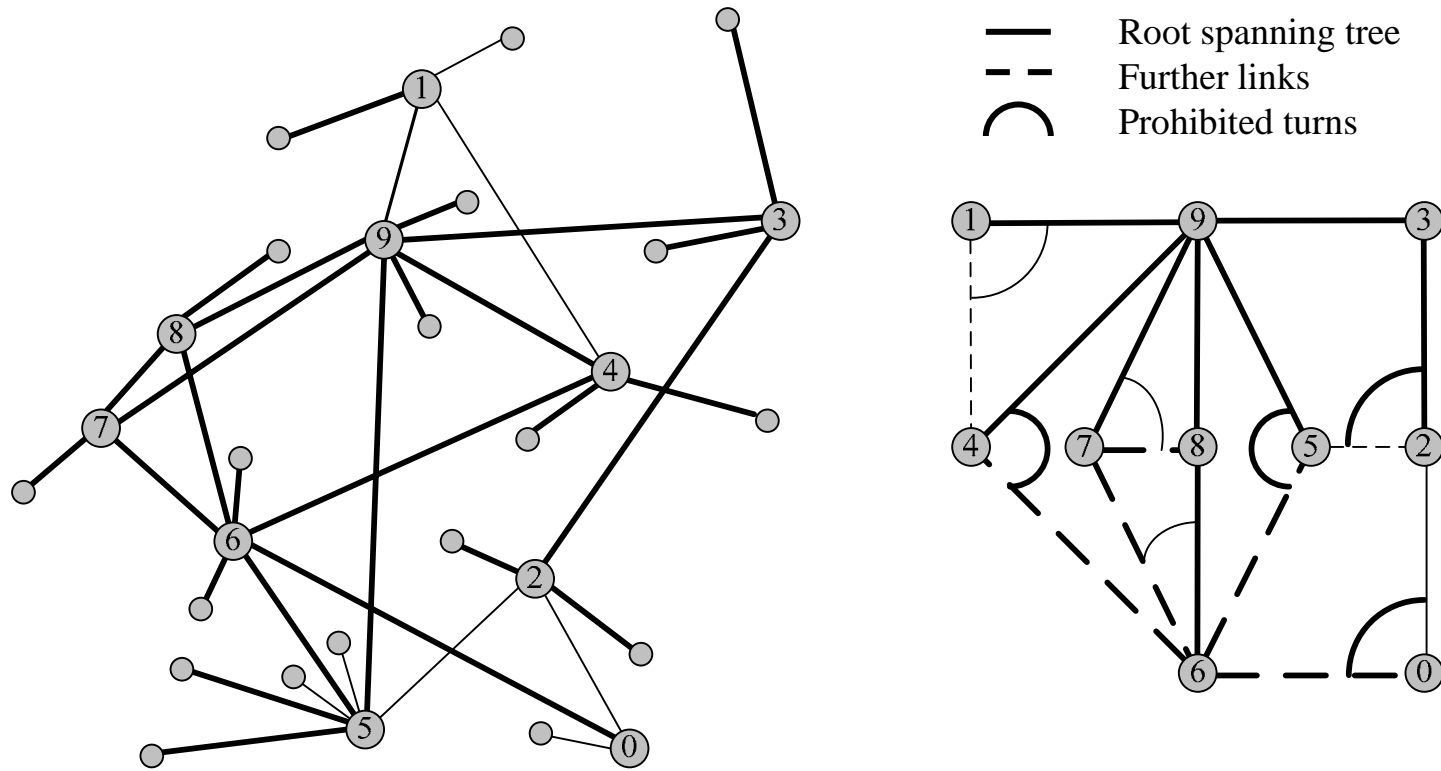


Extended turn-prohibition

- Turn-prohibition prevents from cyclic dependencies
- Root spanning tree from the last node visited by turn-prohibition
 - No cyclic or plain dependencies
 - Provides full connectivity (holds generally only for the last node visited)
 - Belonging dependency graph serves as initial solution
- Turns are substituted by links if possible (i.e. assume a link (3,5) exists)
- All remaining links are added
- Remaining turns are added if no new plain dependencies are created

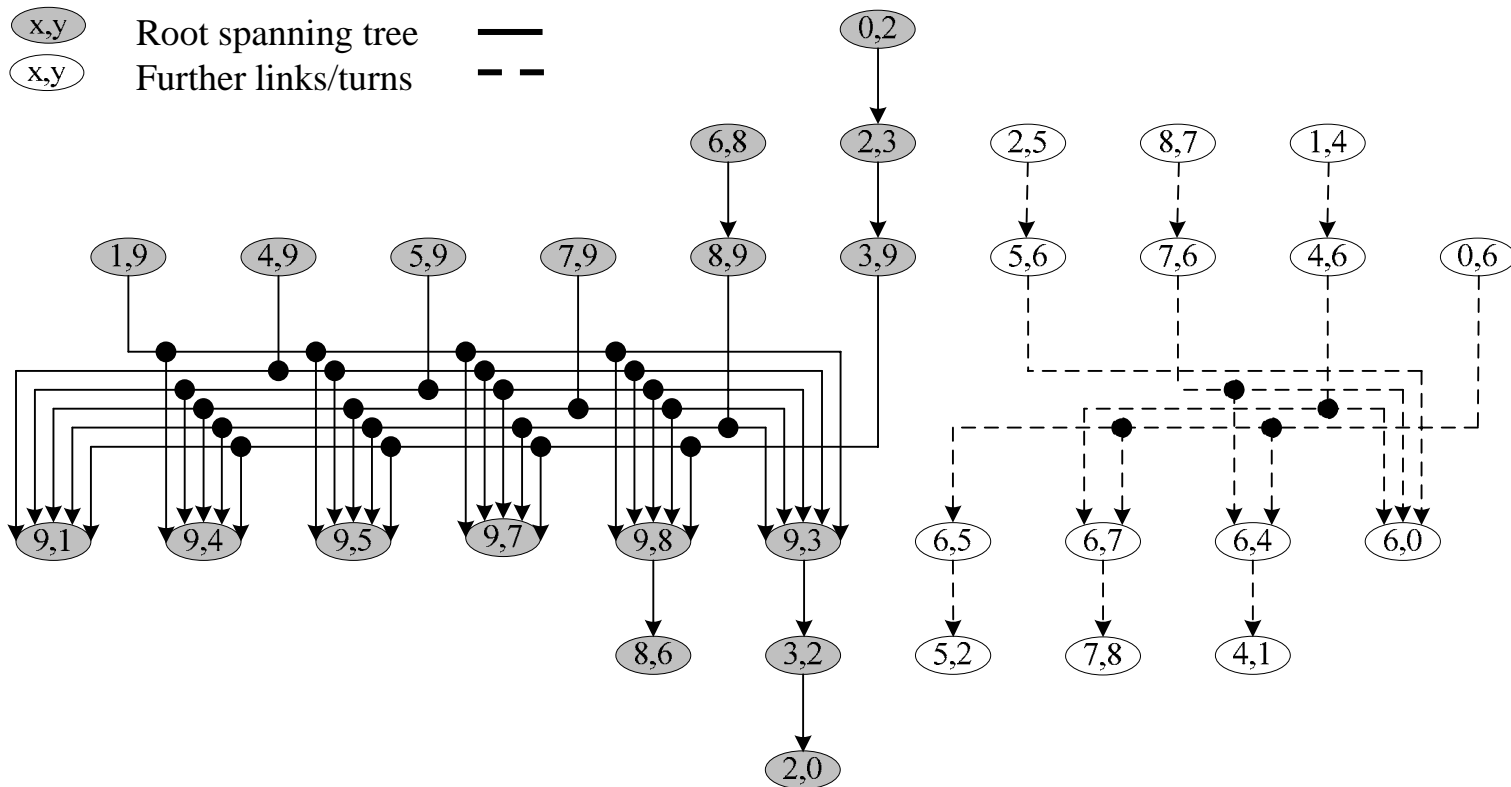


DFN G-WiN turn-prohibition example f_{Q2S}



- 7 out of 43 bi-directional turns prohibited to break all cyclic dependencies
- Shortest paths between all source destination pairs are possible

DFN G-WiN dependency graph



- Prohibition of 11 bi-directional turns avoids plain dependencies in the core
- Only one shortest path (0,6,8) not possible, instead (0,6,7,8)

Conclusions

- Probabilistic network calculus can
 - take significant advantage from independence
 - deal also with dependencehowever, the structure of dependencies has to be known
- Analysis of dependencies
 - Dependency graphs indicate where independence does not hold
 - Allows applying appropriate multiplexing models
- Prevention of dependencies
 - Turn-prohibition to enforce independence where reasonable
 - Explicit routing with prohibited turns for selected services