Guaranteeing Stability and Delay in Dynamic Networks based on Infinite Games

Christof Löding — RWTH Aachen University, Germany

joint work with (all formerly RWTH Aachen University):
James Gross — KTH Royal Institute of Technology, Sweden
Frank Radmacher — SAP AG, Germany
Simon Tenbusch — INFORM GmbH, Germany

Third Cassting Meeting, Brussels, May 21–22, 2014

Outline

Problem Setting

- Communication network in which each node has a capacity for forwarding packets
- Primary users can occupy (block) some capacity of the nodes within their reach
- Secondary network defined by a list of source/destination pairs

Question: Under which conditions can we guarantee stable traffic in the secondary network?

Network Model

$$\mathcal{N} = (G, c, \mathcal{D}, \mathcal{B})$$

- Connectivity graph $G = (V, E)$. Edges model communication links between the sites.
- Capacity function $c : V \to \mathbb{N}$, that is, node v has $c(v)$ independent channels for sending packets.
- $\mathcal{D} = [(s_1, d_1), \ldots (s_m, d_m)]$ demand list of source/destination pairs.
- $\mathcal{B}$ is a set of blocking functions $B : V \to \{0, \ldots, c\}$ modelling the activity of the primary network.

Routing Game

Game between Routing Agent and Blocking Agent played in rounds (discrete time steps) as follows:

- one packet is generated at each source s_i
- Blocking Agent chooses a blocking function $B \in \mathcal{B}$
- Routing Agent can forward at each node v up to $c - B(v)$ many packets along the edges
- packets that were generated at s_i and that arrive at d_i are removed from the network

Routing Game

Game between Routing Agent and Blocking Agent played in rounds (discrete time steps) as follows:

- one packet is generated at each source s_i
- Blocking Agent chooses a blocking function $B \in \mathcal{B}$
- Routing Agent can forward at each node v up to $c - B(v)$ many packets along the edges
- packets that were generated at s_i and that arrive at d_i are removed from the network

Properties of interest:

- Stability – Does the number of packets in the network remain below some bound?
- Packet delay – Does there exist a delay bound for packets?

Outline

Stability Game

Routing game with stability condition for Routing Agent:

Routing Agent wins if there is a bound B such that the number of packets in the network never exceeds B.

Question: Does Routing Agent have a winning strategy?

Stability Game

Routing game with stability condition for Routing Agent:

Routing Agent wins if there is a bound B such that the number of packets in the network never exceeds B.

Question: Does Routing Agent have a winning strategy?

Static networks: We start by analyzing the one-player game in which Blocking Agent does not have any choice: there is only one fixed blocking function B

Multi Commodity Flows

Consider a network $\mathcal{N} = (G, c, \mathcal{D}, \mathcal{B})$ with $\mathcal{B} = \{B\}$.

A multicommodity flow (MCF) f for $\mathcal{N}$ is a tuple $f = (f_1, \ldots, f_m)$ of functions $f_j : E \to \mathbb{Q}_{\geq 0}$ with:

- for all $u \in V \setminus \{s_j, d_j\}$: $\underbrace{\sum_{(v,u) \in E} f_j(v,u)}_{f_j\text{-input at } u} = \underbrace{\sum_{(u,w) \in E} f_j(u,w)}_{f_j\text{-output at } u}$

Multi Commodity Flows

Consider a network $\mathcal{N} = (G, c, \mathcal{D}, \mathcal{B})$ with $\mathcal{B} = \{B\}$.

A multicommodity flow (MCF) f for $\mathcal{N}$ is a tuple $f = (f_1, \ldots, f_m)$ of functions $f_j : E \to \mathbb{Q}_{\geq 0}$ with:

- for all $u \in V \setminus \{s_j, d_j\}$: $\underbrace{\sum_{(v,u) \in E} f_j(v,u)}_{f_j\text{-input at } u} = \underbrace{\sum_{(u,w) \in E} f_j(u,w)}_{f_j\text{-output at } u}$
- $\sum_{(u,d_j) \in E} f_j(u, d_j) = 1$ (input at destination is 1)

Multi Commodity Flows

Consider a network $\mathcal{N} = (G, c, \mathcal{D}, \mathcal{B})$ with $\mathcal{B} = \{B\}$.

A multicommodity flow (MCF) f for $\mathcal{N}$ is a tuple $f = (f_1, \dots, f_m)$ of functions $f_j : E \to \mathbb{Q}_{\geq 0}$ with:

- for all $u \in V \setminus \{s_j, d_j\}$: $$\underbrace{\sum_{(v,u) \in E} f_j(v,u)}_{f_j\text{-input at } u} = \underbrace{\sum_{(u,w) \in E} f_j(u,w)}_{f_j\text{-output at } u}$$
- $\sum_{(u,d_j) \in E} f_j(u, d_j) = 1$ (input at destination is 1)
- $\sum_{(d_j,u) \in E} f_j(d_j, u) = 0$ (output at destination is 0)

Multi Commodity Flows

Consider a network $\mathcal{N} = (G, c, \mathcal{D}, \mathcal{B})$ with $\mathcal{B} = \{B\}$.

A multicommodity flow (MCF) f for $\mathcal{N}$ is a tuple $f = (f_1, \ldots, f_m)$ of functions $f_j : E \to \mathbb{Q}_{\geq 0}$ with:

- for all $u \in V \setminus \{s_j, d_j\}$: $\underbrace{\sum_{(v,u)\in E} f_j(v,u)}_{f_j\text{-input at } u} = \underbrace{\sum_{(u,w)\in E} f_j(u,w)}_{f_j\text{-output at } u}$
- $\sum_{(u,d_j)\in E} f_j(u,d_j) = 1$ (input at destination is 1)
- $\sum_{(d_j,u)\in E} f_j(d_j,u) = 0$ (output at destination is 0)
- for all $u \in V$: $\underbrace{\sum_{j=1}^{m} \left(\sum_{(u,v)\in E} f_j(u,v) \right)}_{\text{overall output at } u} \leq \underbrace{c(u) - B(u)}_{\text{capacity at } u}$

Stability Theorem

Theorem. In the (one player) stability game for $\mathcal{N} = (G, \mathcal{D}, \mathcal{B})$ with $\mathcal{B} = \{B\}$, Routing Agent has a winning strategy if and only if there exists an MCF $f = (f_1, \ldots, f_m)$ for $\mathcal{N}$.

MCF to Routing Strategy

Assume the is an MCF $f = (f_1, \ldots, f_m)$ for $\mathcal{N}$.

- Let p be period that is big enough (bigger than the least common multiple of the denominators of the values for f)
- The strategy works in rounds of p time steps
- At each node u, forward (up to) $p \cdot f_j(u, v)$ many packets along (u, v) in each round
- The flow properties ensure that this is a stable routing strategy

Stable Routing Strategy to MCF

Assume that Routing Agent has a stable routing strategy.

- Safety condition (keep the number of packets bounded)
- Exists stable routing strategy whose decisions only depend on the current network configuration
- If the number of packets is bounded, then the number of network configurations is bounded
- Behaviour is periodic, with some period p

Stable Routing Strategy to MCF

Assume that Routing Agent has a stable routing strategy.

- Safety condition (keep the number of packets bounded)
- Exists stable routing strategy whose decisions only depend on the current network configuration
- If the number of packets is bounded, then the number of network configurations is bounded
- Behaviour is periodic, with some period p
- For each j, define $f_j(u, v)$ to be $\frac{n}{p}$, where n is the number of (s_j, d_j)-packets traversing (u, v) in one cycle of the period.
- This defines an MCF because
  - in one cycle of the period the number of packets entering and leaving a node must be the same
  - the capacity of the nodes is respected by the routing strategy

Extension to Dynamic Networks

Theorem. In the game for $\mathcal{G} = (G, \mathcal{D}, \mathcal{B})$ routing agent has a winning strategy if and only if there exists an MCF $f^B = (f_1^B, \ldots, f_m^B)$ for $(G, \mathcal{D}, \{B\})$ for each $B \in \mathcal{B}$.

Idea:

- For each B choose a stable routing strategy σ_B.
- Whenever B is active, play according to σ_B, and only forward packets that were generated when B was active.

Outline

Stability vs. Delay

A stable routing strategy does not guarantee that each packet is delivered:

- Some nodes might be used for some blocking functions but are isolated for other blocking functions.
- Even if no node can be isolated by a blocking function, we did not succeed in defining a routing strategy that guarantees bounded delay.
- We need an MCF with slightly higher throughput to obtain a bounded delay strategy.

Delay Bounds

Theorem. Let $\varepsilon > 0$, and let $\mathcal{N} = (G, \mathcal{D}, \mathcal{B})$ be a network with $c(u) - B(u) \geq 1$ for all $u \in V$ and $B \in \mathcal{B}$.

If for each $B \in \mathcal{B}$ there is an MCF $f^B = (f_1^B, \ldots, f_m^B)$ for $(G, \mathcal{D}, \{B\})$ with throughput at least $1 + \varepsilon$, then routing agent has a strategy such that all packets are delivered within a bounded number of steps.

Idea:

- In the strategy for stability, under blocking function B, only packets are forwarded that were generated under B. The other packets are put on hold.
- The strategy for bounded delay uses the additional ε for forwarding such packets.

Outline

Blocking Functions

- In a realistic scenario, the blocking functions are given by constraints (are not listed explicitly)
- We consider the following type of constraints
  - $O_{\max}$: maximal total number of blocked channels
  - $c_{\min}$: the minimal number of available channels at each node
- $\mathcal{B}(O_{\max}, c_{\min})$ is the set of all blocking functions respecting these constraints

Blocking Functions

- In a realistic scenario, the blocking functions are given by constraints (are not listed explicitly)
- We consider the following type of constraints
  - $O_{\max}$: maximal total number of blocked channels
  - $c_{\min}$: the minimal number of available channels at each node
- $\mathcal{B}(O_{\max}, c_{\min})$ is the set of all blocking functions respecting these constraints

Size of $\mathcal{B}(O_{\max}, c_{\min})$:

- network with 50 nodes
- capacity $c = 3$ at each node
- $c_{\min} = 1$

$O_{\max}$	$\lvert\mathcal{B}(O_{\max}, 1)\rvert$
1	51
2	1.326
3	23.376
4	313.701
5	3.412.461
6	31.298.361
7	248.635.761
8	1.744.483.611
9	10.970.926.711
10	62.561.143.641

Backtracking Algorithm

Start with blocking function B_0 that does not block anything.

- Given the current blocking function B, compute MCF f with throughput $1 + T$ with T maximal.
- If $T < 0$, then there is no MCF for B.

Backtracking Algorithm

Start with blocking function B_0 that does not block anything.

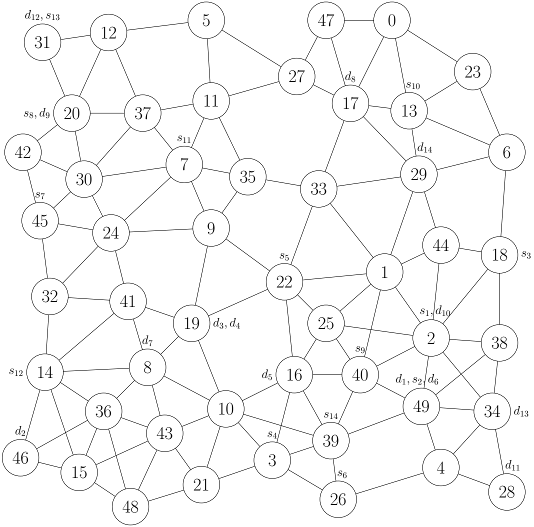
- Given the current blocking function B, compute MCF f with throughput $1 + T$ with T maximal.
- If $T < 0$, then there is no MCF for B.
- If T is bigger than the number of channels that can still be blocked, then no extension of B can “break” f. (Each additional channel blocked at a node can reduce the throughput of f by at most 1.)

Backtracking Algorithm

Start with blocking function B_0 that does not block anything.

- Given the current blocking function B, compute MCF f with throughput $1 + T$ with T maximal.
- If $T < 0$, then there is no MCF for B.
- If T is bigger than the number of channels that can still be blocked, then no extension of B can “break” f. (Each additional channel blocked at a node can reduce the throughput of f by at most 1.)
- Otherwise, normalize f to throughput 1 and increase the number of blocked channels at some node u to break f.
- Continue with this extended blocking function.

Example Network for Evaluation

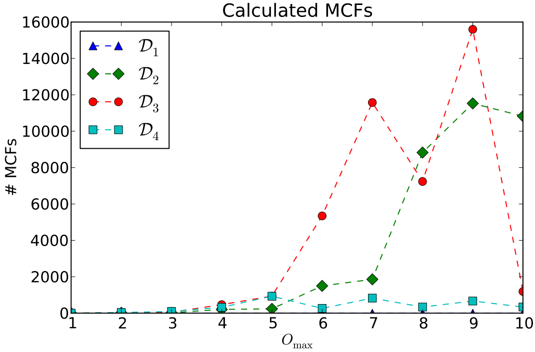


$$\mathcal{D}_2 = \mathcal{D}_1 \cup \{(40,20), (13,2)\}$$
$$\mathcal{D}_3 = \mathcal{D}_2 \cup \{(7,28), (14,31)\}$$
$$\mathcal{D}_4 = \mathcal{D}_3 \cup \{(31,34), (39,29)\}$$

$$\mathcal{D}_1 = \{(2,49), (49,46), (18,19), (3,19), (22,16), (26,49), (45,8), (20,17)\}$$

Results

	$O_{\max}$ 1	2	3	4	5	6	7	8	9	10
$\mathcal{D}_1$	●	●	●	●	●	●	●	●	●	●
$\mathcal{D}_2$	●	●	●	●	●	●	●	●	●	○
$\mathcal{D}_3$	●	●	●	●	●	●	●	○	○	○
$\mathcal{D}_4$	●	●	●	○	○	○	○	○	○	○



Outline

Conclusion

Summary:

- Characterization of stable networks using MCFs
- Sufficient condition for delay bound
- Implementation working for large sets of blocking functions
- Constructed routing strategy is global and thus not useful in practice. But we can use our results to provide a sufficient condition such that backpressure routing is stable.

Ideas for future work:

- Analysis for finer network and routing models
- Dynamic changes of source/destination pairs
- Further investigation of delay bounds