Universität Rostock

Traditio et Innovatio

Institute of Computer Science

INSTITUT FÜR INFORMATIK

Dynamic Migration of Real-Time Traffic Flows in SDN-Enabled Networks

Peter Danielis[*], **György Dán**[†], **James Gross**[†], **and André Berger**[‡]

[*]University of Rostock, Germany
[†]KTH Royal Institute of Technology, Sweden
[‡]Maastricht University, The Netherlands

Outline

Introduction

System Model and Problem Formulation

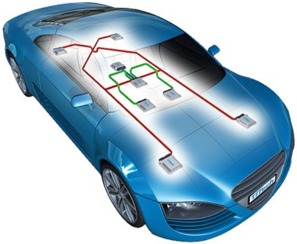
Flow Migration Algorithms

Performance Evaluation

Conclusion & Future Work

Real-Time Communication

- Application requirements
  - Hard real-time capability of network flows
  - Integrate new flows at runtime



© TTech

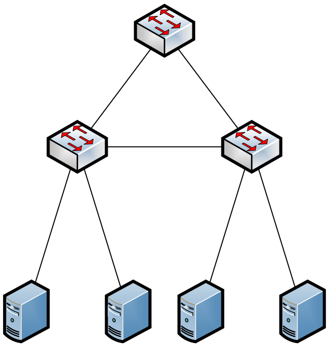
© BMW Group

© NASA

Real-Time Communication

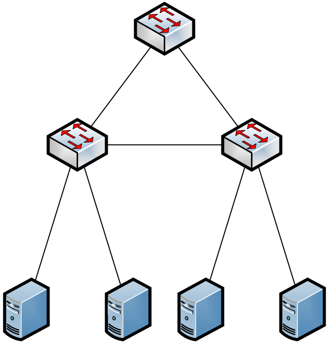
- Application requirements
  - Hard real-time capability of network flows
  - Integrate new flows at runtime

- Introduction of Ethernet
  - Ethernet is emerging as competitor to fieldbusses



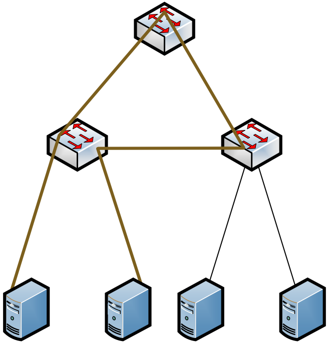
Real-Time Communication and Ethernet

- Issue in case of using Ethernet
  - Medium access impedes real-time capability



Real-Time Communication and Ethernet

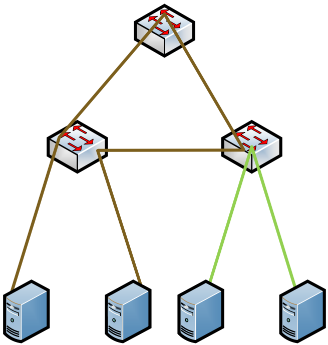
- Issue in case of using Ethernet
  - Medium access impedes real-time capability
- Possible solution
  - Bandwidth reservation for flows to avoid overlapping links



Real-Time Communication and Ethernet

- Issue in case of using Ethernet
  - Medium access impedes real-time capability
- Possible solution
  - Bandwidth reservation for flows to avoid overlapping links

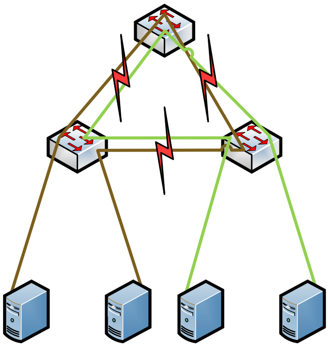
- Integration of new flows **at runtime**
  - If no overlapping links with existing flows: integrate new flows without flow migration



Real-Time Communication and Ethernet

- Issue in case of using Ethernet
  - Medium access impedes real-time capability
- Possible solution
  - Bandwidth reservation for flows to avoid overlapping links

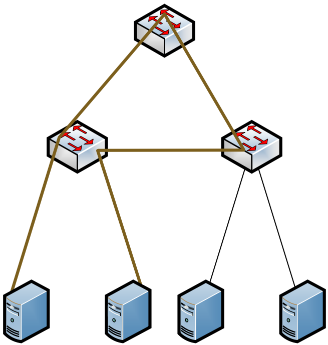
- Integration of new flows **at runtime**
  - If no overlapping links with existing flows: integrate them without flow migration
  - If overlapping links with existing flows: migrate existing flows and integrate new flows



Real-Time Communication and Ethernet

- Issue in case of using Ethernet
  - Medium access impedes real-time capability
- Possible solution
  - Bandwidth reservation for flows to avoid overlapping links

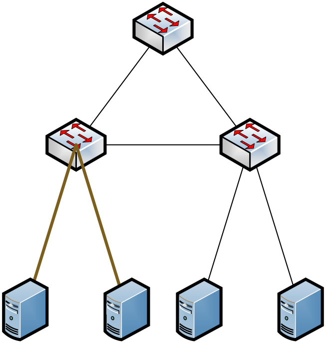
- Integration of new flows **at runtime**
  - If no overlapping links with existing flows: integrate them without flow migration
  - If overlapping links with existing flows: migrate existing flows and integrate new flows



Real-Time Communication and Ethernet

- Issue in case of using Ethernet
  - Medium access impedes real-time capability
- Possible solution
  - Bandwidth reservation for flows to avoid overlapping links

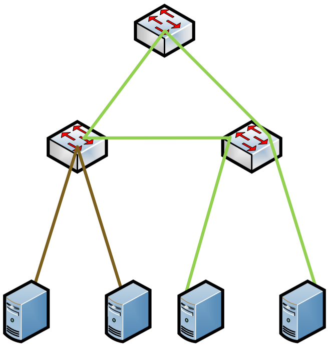
- Integration of new flows **at runtime**
  - If no overlapping links with existing flows: integrate them without flow migration
  - If overlapping links with existing flows: migrate existing flows and integrate new flows



Real-Time Communication and Ethernet

- Issue in case of using Ethernet
  - Medium access impedes real-time capability
- Possible solution
  - Bandwidth reservation for flows to avoid overlapping links

- Integration of new flows **at runtime**
  - If no overlapping links with existing flows: integrate them without flow migration
  - If overlapping links with existing flows: migrate existing flows and integrate new flows



Problem Statement

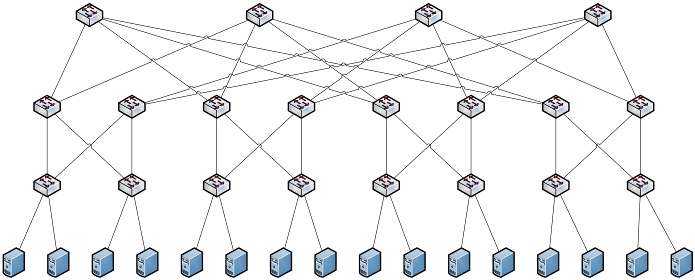
- Flow migration problem (FMP)
  - How to migrate existing flows to place new flows w/o interrupting running traffic?

- Questions related to the FMP
  - When is flow migration necessary?
  - If necessary, how many migration steps are necessary?
  - How computationally expensive is flow migration?

Solution Overview

- Algorithms for direct and indirect flow migration (FM)
  - Direct FM: Migrate any path at most once
  - Indirect FM: Migrate any path more than once

- Numerical analysis of the algorithm for indirect FM on a FatTree topology



Outline

Introduction

System Model and Problem Formulation

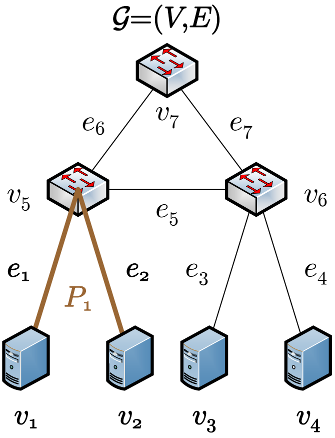
Flow Migration Algorithms

Performance Evaluation

Conclusion & Future Work

System Model

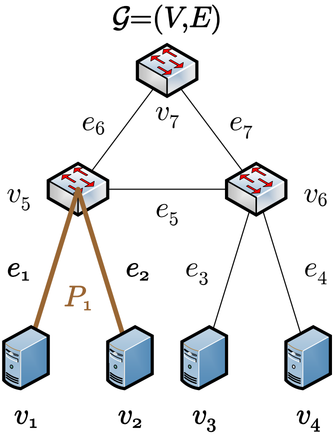
Directed graph $\mathcal{G} = (V, E)$
with vertices $V = \{v_1, \ldots, v_7\}$
and edges $E = \{e_1, \ldots, e_7\}$



System Model

Directed graph $\mathcal{G} = (V, E)$
with vertices $V = \{v_1, ..., v_7\}$
and edges $E = \{e_1, ..., e_7\}$

Source-destination pair
$\mathcal{C} = \{(s_1, t_1)\} = \{(v_1, v_2)\}$

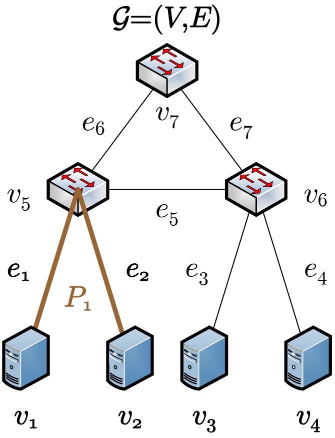


System Model

Directed graph $\mathcal{G} = (V, E)$
with vertices $V = \{v_1, ..., v_7\}$
and edges $E = \{e_1, ..., e_7\}$

Source-destination pair
$\mathcal{C} = \{(s_1, t_1)\} = \{(v_1, v_2)\}$

Length-constrained paths
$P_1 = (e_1, e_2)$ with $l(P_1) = 2$



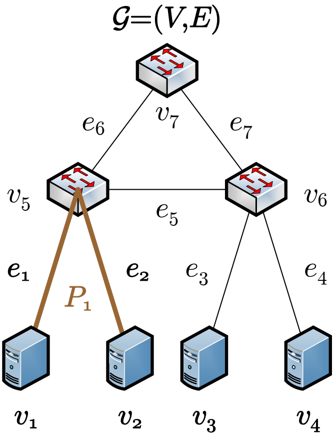
System Model

Directed graph $\mathcal{G} = (V, E)$
with vertices $V = \{v_1, ..., v_7\}$
and edges $E = \{e_1, ..., e_7\}$

Source-destination pair
$\mathcal{C} = \{(s_1, t_1)\} = \{(v_1, v_2)\}$

Length-constrained paths
$P_1 = (e_1, e_2)$ with $l(P_1) = 2$

Valid collection of paths
$R = \{P_1\}$



System Model

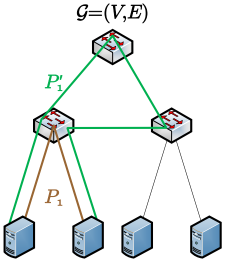
Directed graph $\mathcal{G} = (V, E)$
with vertices $V = \{v_1, ..., v_7\}$
and edges $E = \{e_1, ..., e_7\}$

Source-destination pair
$\mathcal{C} = \{(s_1, t_1)\} = \{(v_1, v_2)\}$

Length-constrained paths
$P_1 = (e_1, e_2)$ with $l(P_1) = 2$

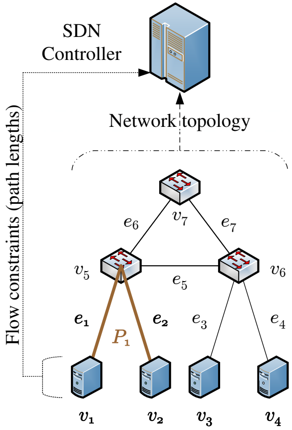
Valid collection of paths
$R = \{P_1\}$

All possible collections for the host pair in $\mathcal{C}$
$\mathcal{R}_\mathcal{C} = \{\{P_1\}, \{P_1'\}\}$



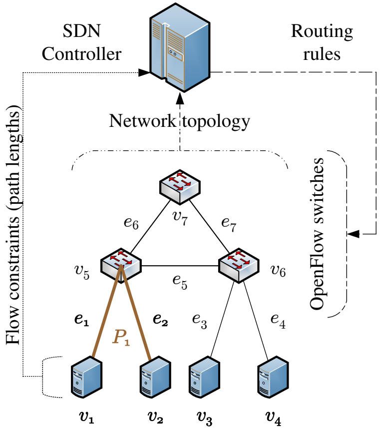
Problem Formulation

- SDN controller has network overview



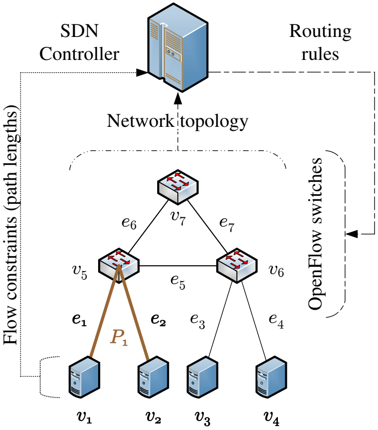
Problem Formulation

- SDN controller has network overview
- It installs rules via atomic forwarding table configurations (AFTCs)



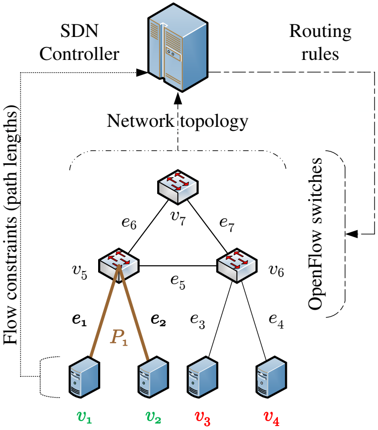
Problem Formulation

- SDN controller has network overview
- It installs rules via atomic forwarding table configurations (AFTCs)
- Definition of Flow Migration (FM)
  - FM from R to R' is a sequence $\mathcal{S} = (R^{(0)}, \ldots, R^{(K)})$
  - $R^{(0)} = R$ and $R^{(K)} = R'$
  - $R^{(k)}$ can be obtained from $R^{(k-1)}$ by a sequence of AFTCs



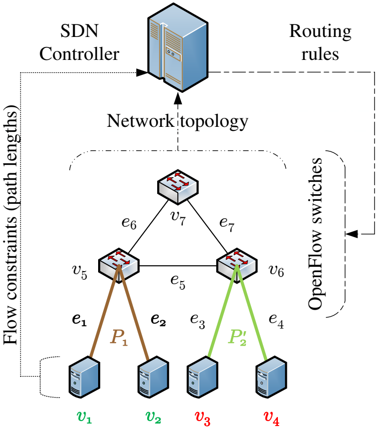
Problem Formulation

- SDN controller has network overview
- It installs rules via atomic forwarding table configurations (AFTCs)
- Definition of Flow Migration (FM)
  - FM from R to R' is a sequence $\mathcal{S} = (R^{(0)}, \ldots, R^{(K)})$
  - $R^{(0)} = R$ and $R^{(K)} = R'$
  - $R^{(k)}$ can be obtained from $R^{(k-1)}$ by a sequence of AFTCs
- Flow Migration Problem (FMP)
  - Given: $\mathcal{C}$, R, (s_{N+1}, t_{N+1})



Problem Formulation

- SDN controller has network overview
- It installs rules via atomic forwarding table configurations (AFTCs)
- Definition of Flow Migration (FM)
  - FM from R to R' is a sequence $\mathcal{S} = (R^{(0)}, \ldots, R^{(K)})$
  - $R^{(0)} = R$ and $R^{(K)} = R'$
  - $R^{(k)}$ can be obtained from $R^{(k-1)}$ by a sequence of AFTCs
- Flow Migration Problem (FMP)
  - Given: $\mathcal{C}$, R, (s_{N+1}, t_{N+1})
  - Unknown: FM $\mathcal{S}$ from R to R'
  - Constraints: $\exists P'_{N+1}$ for which $\{P'_1, \ldots, P'_{N+1}\} \in \mathcal{R}_{\mathcal{C}'}$ is a path collection for the set $\mathcal{C}' = \mathcal{C} \cup \{(s_{N+1}, t_{N+1})\}$



$N = 1, R = P_1 = P'_1 = R'$

Outline

Introduction

System Model and Problem Formulation

Flow Migration Algorithms

Performance Evaluation

Conclusion & Future Work

Considerations for Flow Migration Algorithms

1. FMP is infeasible
2. FMP is feasible
   2.1 No FM is needed: there is P'_{N+1} such that $\{P_1, \ldots, P_N, P'_{N+1}\}$ is a path collection
   2.2 FM is needed: there is P'_{N+1} such that $\{P'_1, \ldots, P'_N, P'_{N+1}\} \in \mathcal{R}_{\mathcal{C}'}$ is a path collection
      – Migrate R to $R' \in \mathcal{R}^* = \{(P'_1, \ldots, P'_N) | \exists P'_{N+1} s.t. (P'_1, \ldots, P'_N, P'_{N+1}) \in \mathcal{R}_{\mathcal{C}'}\}$

Considerations for Flow Migration Algorithms

1. FMP is infeasible
2. FMP is feasible
   2.1 No FM is needed: there is P'_{N+1} such that $\{P_1, \ldots, P_N, P'_{N+1}\}$ is a path collection
   2.2 FM is needed: there is P'_{N+1} such that $\{P'_1, \ldots, P'_N, P'_{N+1}\} \in \mathcal{R}_{\mathcal{C}'}$ is a path collection
   - Migrate R to $R' \in \mathcal{R}^* = \{(P'_1, \ldots, P'_N) | \exists P'_{N+1} s.t. (P'_1, \ldots, P'_N, P'_{N+1}) \in \mathcal{R}_{\mathcal{C}'}\}$

- Assumptions for algorithms
  - Set $\mathcal{R}^*$ of path collections can be computed
  - Problem in general is NP-hard
  - Each FM bases on changing a single path at a time = *elementary FM*

Direct Flow Migration (DFM)

- **Data:** $R = \{P_1, \ldots, P_N\}$, $R' = \{P'_1, \ldots, P'_N\}$
- **Result:** DFM sequence $(i_1, \ldots, i_N)$
- **Constraints:** For every $1 \leq r \leq N$, $\{P'_{i_1}, \ldots, P'_{i_r}, P_{i_{r+1}}, ..., P_{i_N}\}$ is a path collection

Direct Flow Migration (DFM)

- **Data:** $R = \{P_1, \ldots, P_N\}$, $R' = \{P'_1, \ldots, P'_N\}$
- **Result:** DFM sequence $(i_1, \ldots, i_N)$
- **Constraints:** For every $1 \leq r \leq N$, $\{P'_{i_1}, \ldots, P'_{i_r}, P_{i_{r+1}}, ..., P_{i_N}\}$ is a path collection
- **Definitions**
  - Conflict graph $\mathcal{G}^* = (V^*, A^*)$
  - $V^* = [N] = \{1, ..., N\}$
  - $\exists arc(i, j) \in A^*$ if and only if $P'_i \cap P_j \neq \emptyset$

Direct Flow Migration (DFM)

- **Data:** $R = \{P_1, \ldots, P_N\}$, $R' = \{P'_1, \ldots, P'_N\}$
- **Result:** DFM sequence $(i_1, \ldots, i_N)$
- **Constraints:** For every $1 \leq r \leq N$, $\{P'_{i_1}, \ldots, P'_{i_r}, P_{i_{r+1}}, \ldots, P_{i_N}\}$ is a path collection
- **Definitions**
  - Conflict graph $\mathcal{G}^* = (V^*, A^*)$
  - $V^* = [N] = \{1, \ldots, N\}$
  - $\exists arc(i, j) \in A^*$ if and only if $P'_i \cap P_j \neq \emptyset$

Lemma

The direct FM problem has a feasible solution if and only if the graph $\mathcal{G}^*$ is acyclic. If $\mathcal{G}^*$ is acyclic, a feasible solution can be found in polynomial time (for proof see paper).

Direct Flow Migration (DFM)

- **Data:** $R = \{P_1, \dots, P_N\}$, $R' = \{P'_1, \dots, P'_N\}$
- **Result:** DFM sequence $(i_1, \dots, i_N)$
- **Constraints:** For every $1 \leq r \leq N$, $\{P'_{i_1}, \dots, P'_{i_r}, P_{i_{r+1}}, ..., P_{i_N}\}$ is a path collection
- **Definitions**
  - Conflict graph $\mathcal{G}^* = (V^*, A^*)$
  - $V^* = [N] = \{1, ..., N\}$
  - $\exists arc(i, j) \in A^*$ if and only if $P'_i \cap P_j \neq \emptyset$

Lemma

The direct FM problem has a feasible solution if and only if the graph $\mathcal{G}^*$ is acyclic.
If $\mathcal{G}^*$ is acyclic, a feasible solution can be found in polynomial time (for proof see paper).

- **Algorithm**
  - There must be a vertex $i_1 \in V^*$ with outdegree=0, i.e., $P'_{i_1} \cap P_j = \emptyset$ for all $j \neq i_1$
  - Remove i_1 from $\mathcal{G}^*$ and continue in the same manner to obtain the feasible sequence

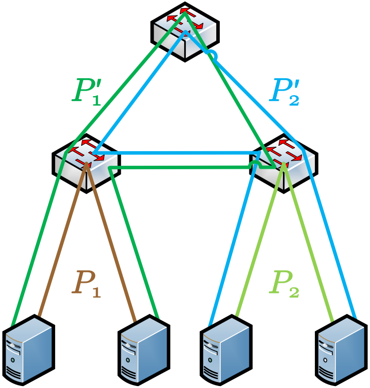
Generic Flow Migration (GFM) Algorithm

Data: Graph $\mathcal{G}$,Path collections $\mathcal{R}_{\mathcal{C}}$,Set $\mathcal{C}'$
Result: GFM sequence $\mathcal{S}$

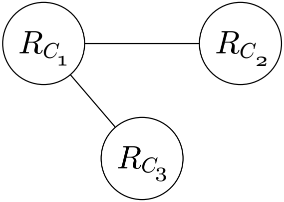
$\forall i$ compute $\mathcal{P}_i$;
Construct graph $\mathcal{T} = (\mathcal{R}_{\mathcal{C}}, \mathcal{F})$, where $(R, R') \in \mathcal{F}$ if R' is elementary FM of R ;
Let $\mathcal{R}^* = \{(P'_1, \ldots, P'_N) | \exists P'_{N+1} s.t. (P'_1, \ldots, P'_N, P'_{N+1}) \in \mathcal{R}_{\mathcal{C}'}\}$;
Find shortest path from R to any $R^* \in \mathcal{R}^*$ in graph $\mathcal{T}$;
if *dist*$(R, R^*) < \infty$ **then**
    $\mathcal{S} = path(R, R^*)$;
else
    $\mathcal{S} = \emptyset$;
end

Generic Flow Migration (GFM) Example

$\mathcal{G}=(V,E)$



$\mathcal{T}=(\mathcal{R_C},\mathcal{F})$

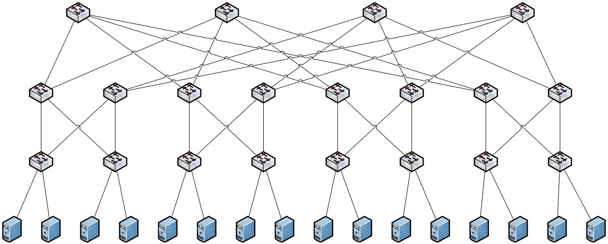


$$\mathcal{R_C}=\{R_{C_1},R_{C_2},R_{C_3}\}=$$
$$\{\{P_1,P_2\},\{P'_1,P_2\},\{P_1,P'_2\}$$

$$\mathcal{F}=\{(R_{C_1},R_{C_2}),(R_{C_1},R_{C_3})\}$$

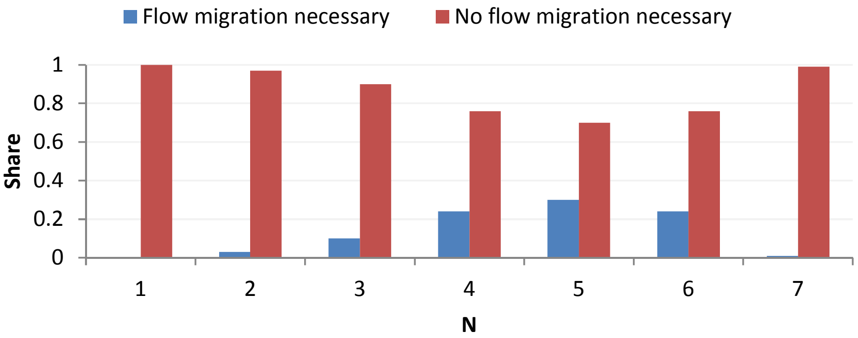
Outline

Methodology



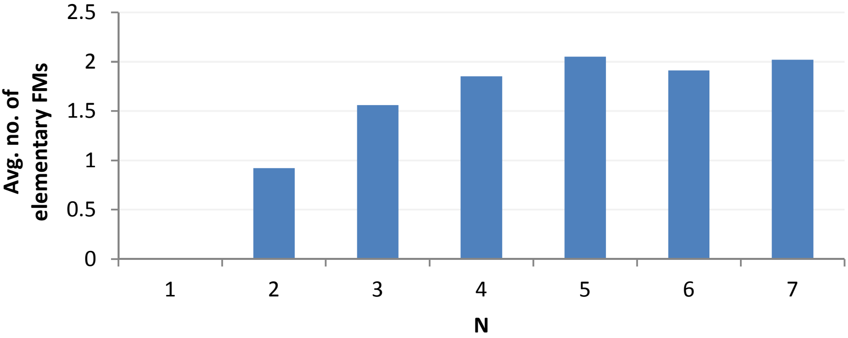
- FatTree topology: 16 hosts, 20 OpenFlow switches
- $N = \{1, ..., 7\}$ source-destination pairs with length-constraint of 6 edges
- 5000 sets $\mathcal{C}$ for each N generated at random without replacement
- $N + 1$-st source-destination host pair chosen at random to create an FMP instance
- Results obtained with Matlab R2015b (64 bit Win8, Intel i7-2600 CPU, 16 GB RAM)

Necessity of FM



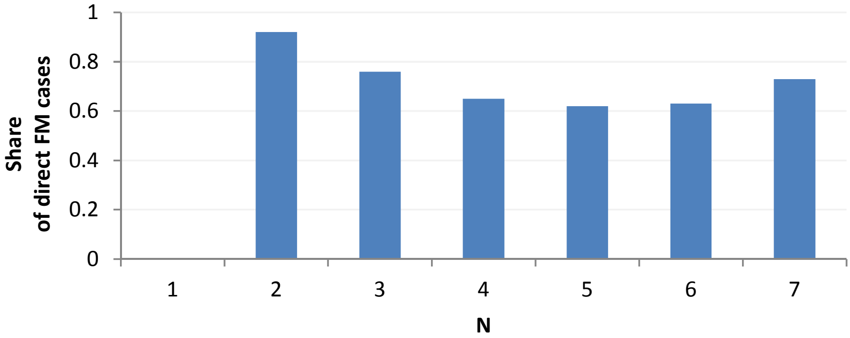
- Low N values: FM often not necessary
- Increasing N: Share of FMP instances with necessary FM increases until $N = 5$

Average number of elementary FMs



- No. of elementary FMs increases until $N = 5$ (5 elementary FMs in some cases)
- No. of elementary FMs remain constant for higher N

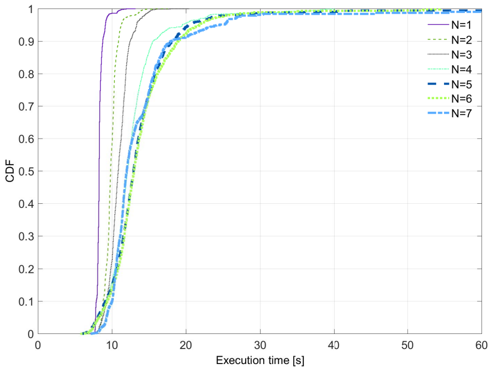
Share of Direct FM Cases



- Direct FM sufficient in 62 - 92% of all feasible FMPs
- $N = 4 \ldots 6$: 24 - 30% of all FMPs require FM (approx. 60% are direct)

CDF of the Execution Times of the GFM Algorithm

- Execution time of GFM algorithm increases with N
- High N: longer execution times (10% over 15s)



Outline

Introduction

System Model and Problem Formulation

Flow Migration Algorithms

Performance Evaluation

Conclusion & Future Work

Conclusion

- Dynamic flow migration for delay constrained traffic in software-defined networks
- Two algorithms for conflict-free flow migration
  - Direct flow migration: polynomial runtime but limited solution set
  - Generic flow migration: increased computational complexity but complete solution set
- Numerical results
  - Flow migration necessary in 24% to 30% of all cases with several migration steps
  - Direct flow migration possible in 62% to 92% of the cases

Future Work

- Evaluation of algorithms in simulation environment
- Comparison to existing solutions
- Consideration of weighted graphs

Thank you for your attention!