A Flow Aggregation Method Based on End-to-End Delay in SDN

Takuya Kosugiyama†, Kazuki Tanabe†, Hiroki Nakayama‡, Tsunemasa Hayashi‡ and Katsunori Yamaoka†

† Tokyo Institute of Technology, Japan
‡ Bosco Technologies Inc., Japan

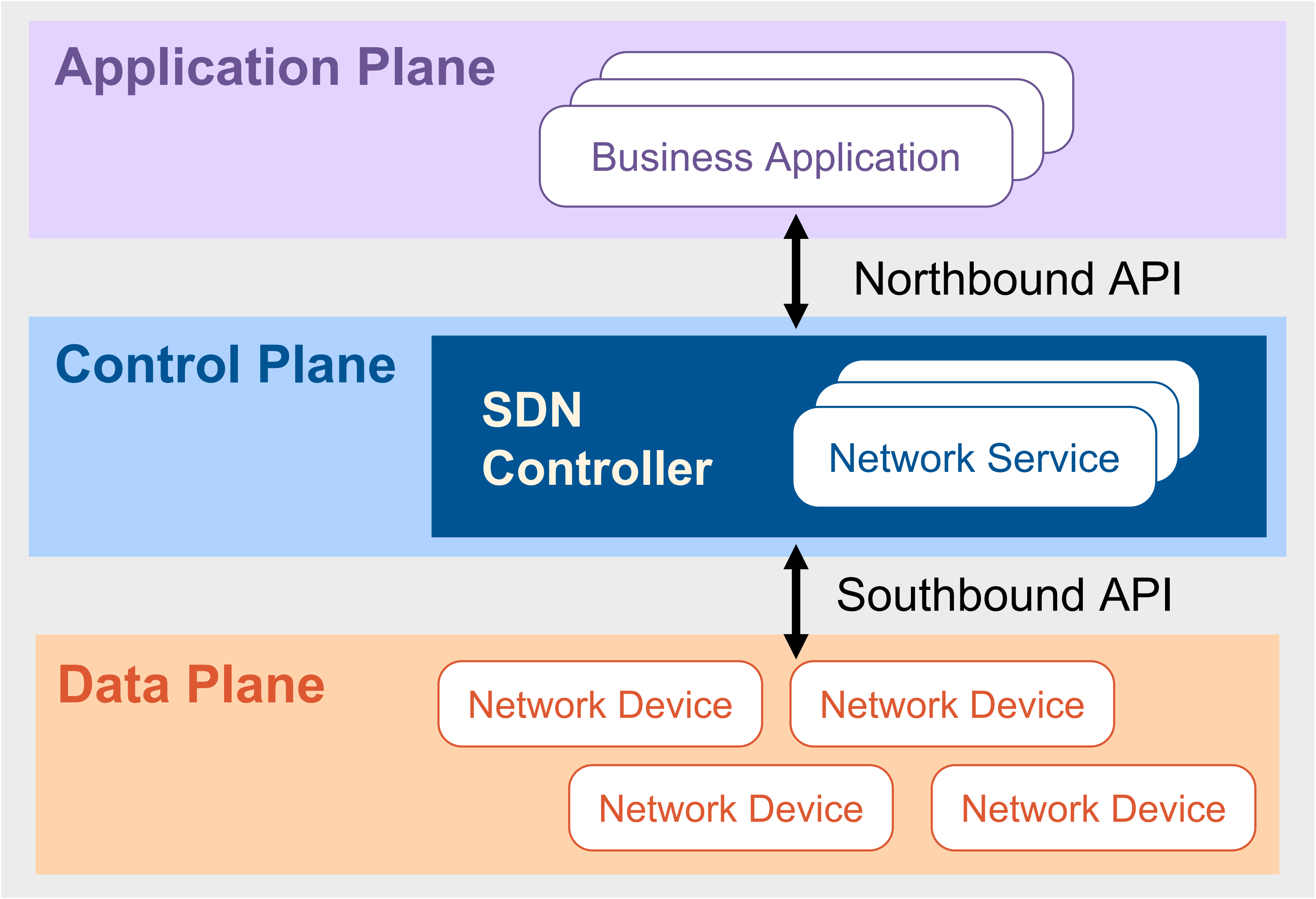
Overview

- Background
- Goal
- Modeling
- Algorithm
- Result
- Summary

What is SDN?

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

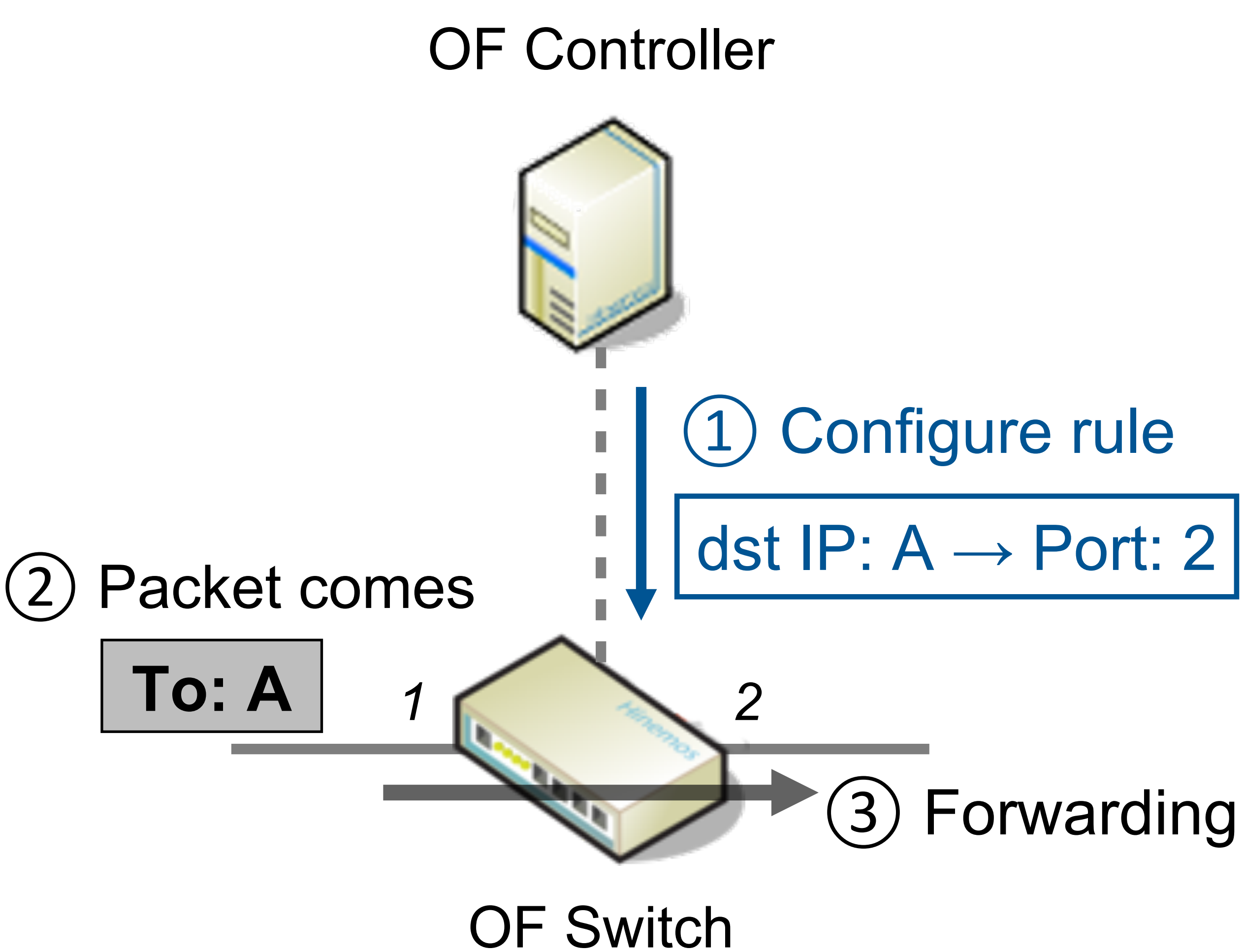
- Software-Defined Networking (SDN)
  - Controllable via API



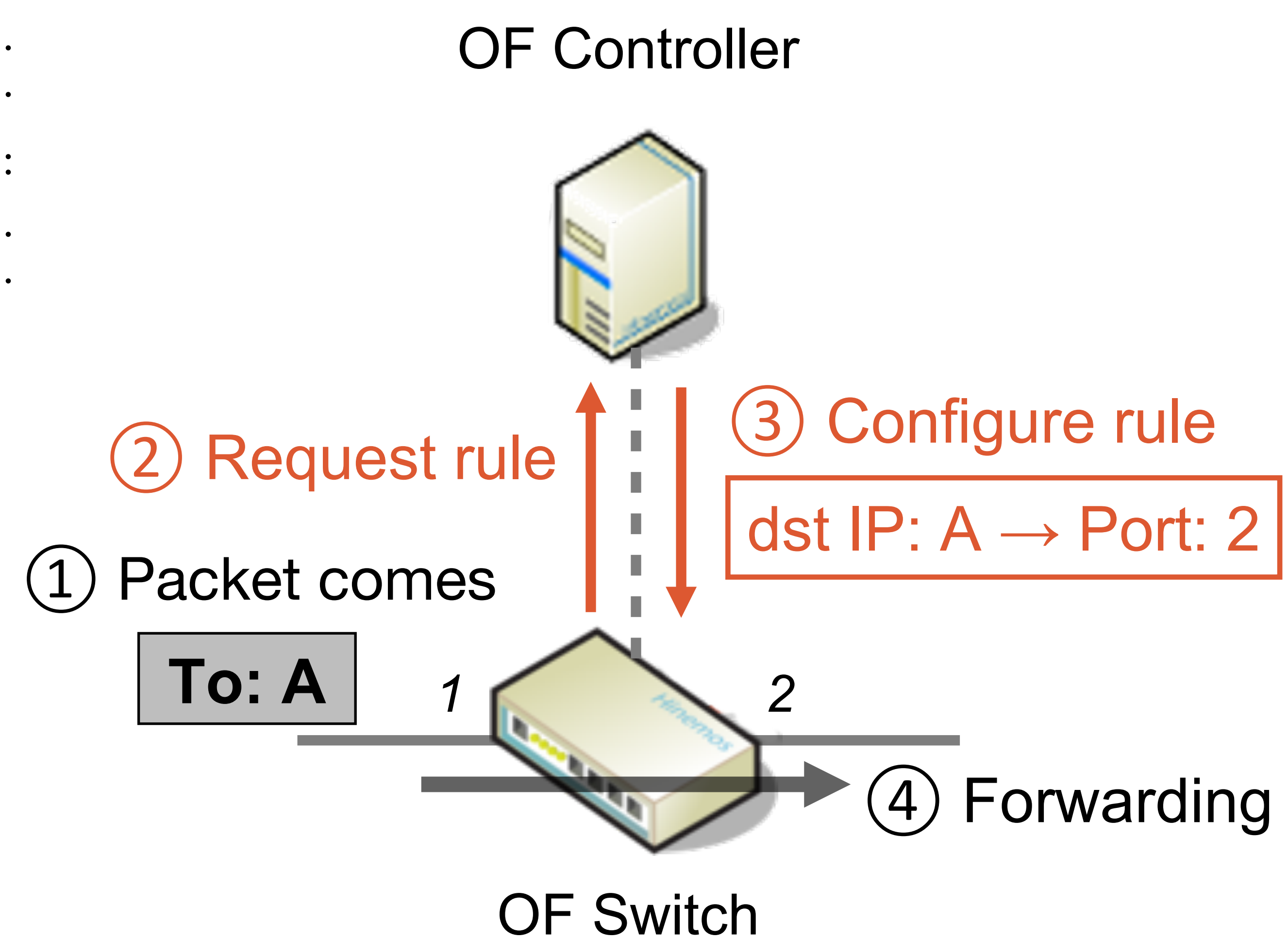
Problem

- Routing Mechanism in OpenFlow (OF)

1. Proactive



2. Reactive



[1] S.H. Yeganeh, A. Tootoonchian, and Y. Ganjali, “On scalability of software-defined networking,” IEEE Communications Magazine, vol.51, no.2, pp.136–141, Feb. 2013.

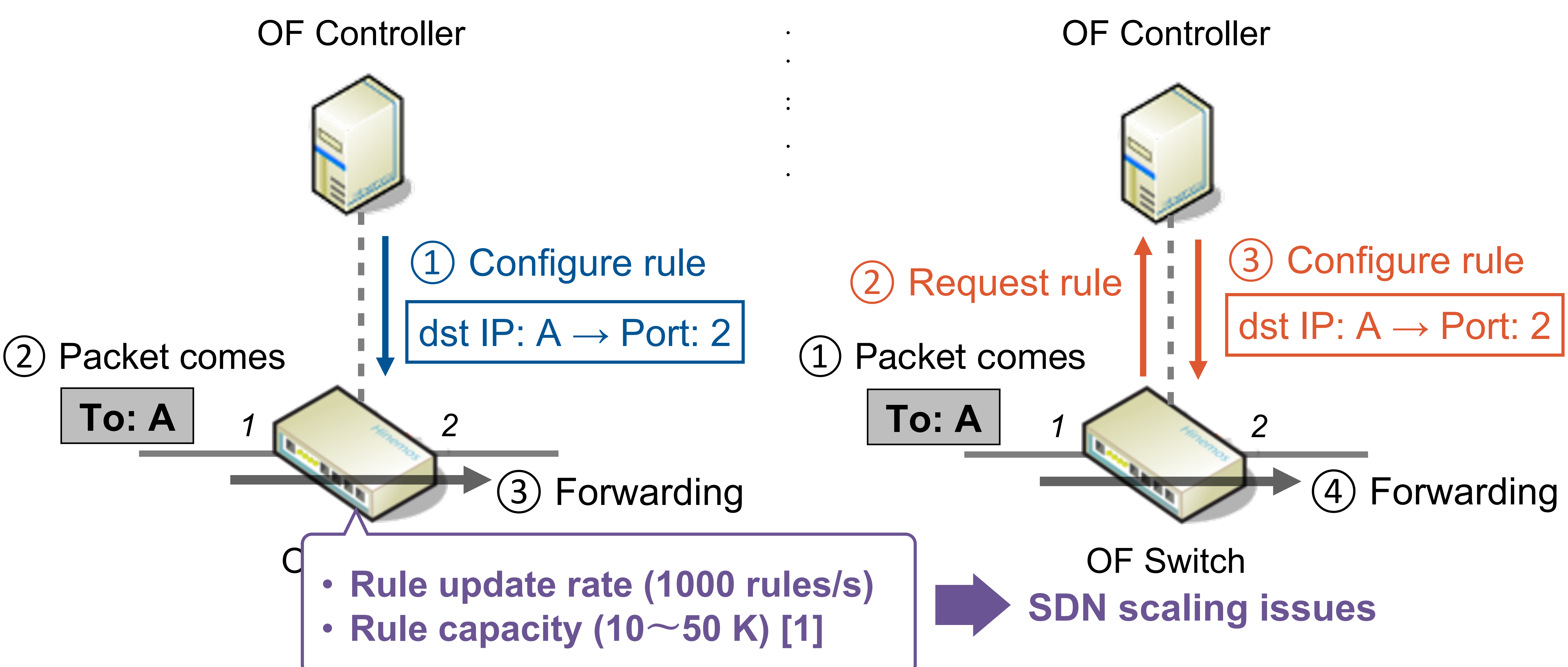
IEEE ICC 2017 CQRM
“A Flow Aggregation Method Based on End-to-End Delay in SDN”
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

Problem

- Routing Mechanism in OpenFlow (OF)

1. Proactive

2. Reactive



[1] S.H. Yeganeh, A. Tootoonchian, and Y. Ganjali, “On scalability of software-defined networking,” IEEE Communications Magazine, vol.51, no.2, pp.136–141, Feb. 2013.

Goal

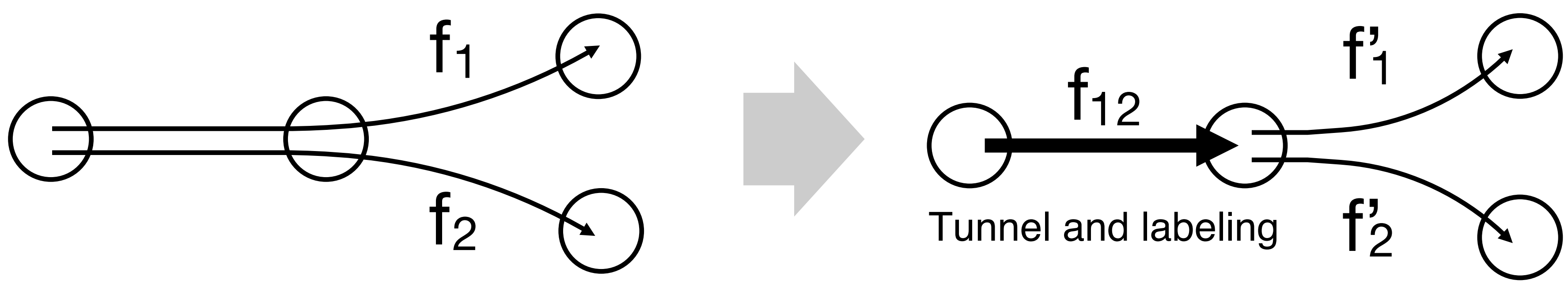
- How to manage massive flows in SDN
  - Aggregate flows and reduce number of flows
    - Rule aggregation method based on bandwidth [1][2]
    - Rule division method to satisfy capacity [3][4]
- There are *no works* to aggregate flows based on End-to-End **allowable delay**
  - Target: IoT / M2M flow

A flow aggregation method to minimize number of flows satisfying allowable delay

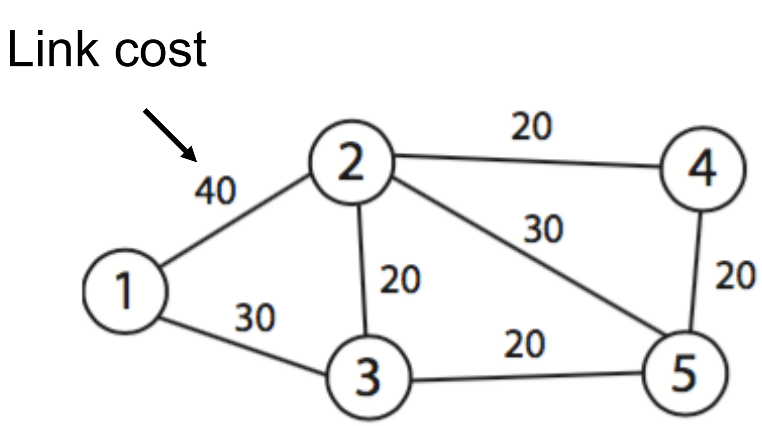
[1] F. Giroire, J. Moulierac, and T. K. Phan. Optimizing rule placement in software-defined networks for energy-aware routing. In 2014 IEEE Global Communications Conference, pp. 2523–2529, Dec 2014.
[2] Xuan Nam Nguyen, Damien Saucez, Chadi Barakat, and Thierry Turletti. OFFICER: A general optimization framework for OpenFlow rule allocation and endpoint policy enforcement. Proceedings - IEEE INFOCOM, Vol. 26, pp. 478–486, 2015.
[3] Yossi Kanizo, David Hay, and Isaac Keslassy. Palette: Distributing tables in software-defined networks. Proceedings - IEEE INFOCOM, pp. 545–549, 2013.
[4] Nanxi Kang, Zhenming Liu, Jennifer Rexford, and David Walker. Optimizing the ”One Big Switch” Abstraction in Software-Defined Networks. Conext’13, p. 17, 2013.

Approach

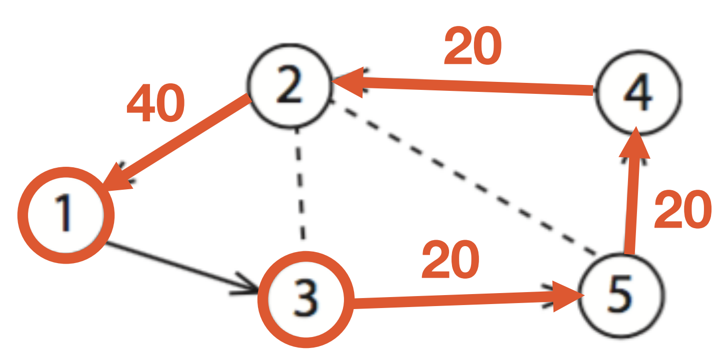
- Basis of flow aggregation
  - Consider flows on same section as one flow



  - Route by aggregated flow
    - ex) Full mesh flows (allowable delay = 80)



(a) Network



(b) Maximum aggregation
not satisfying allowable delay

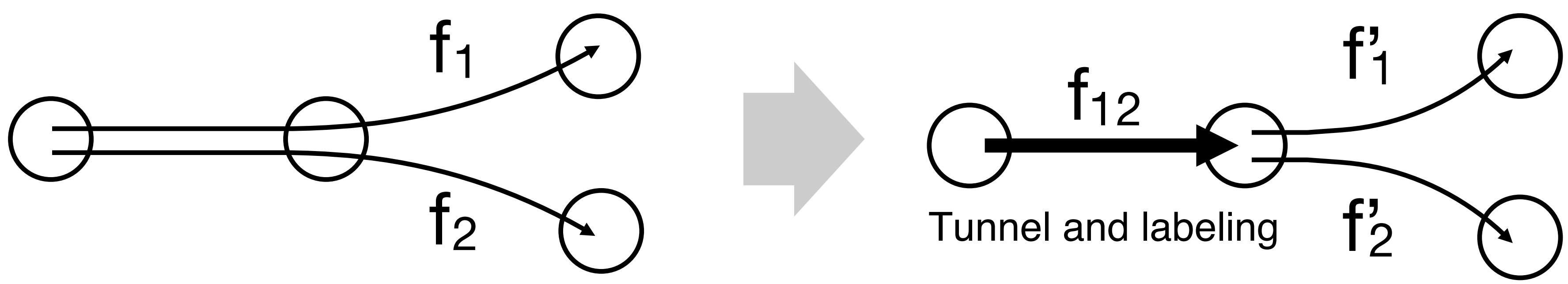
- flow: 3 → 1

20+20+20+40
= 100 > 80

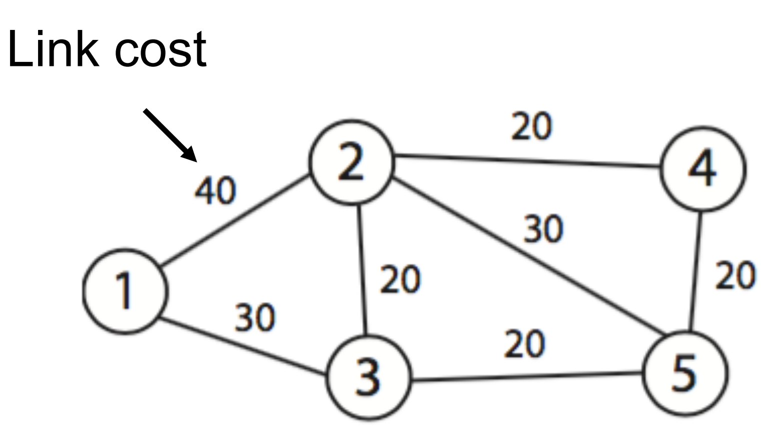
Approach

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

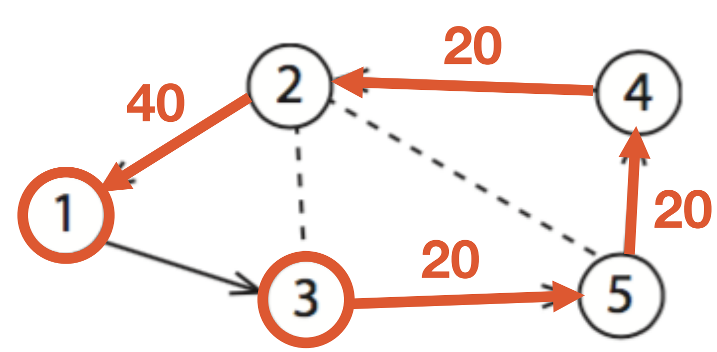
- Basis of flow aggregation
  - Consider flows on same section as one flow



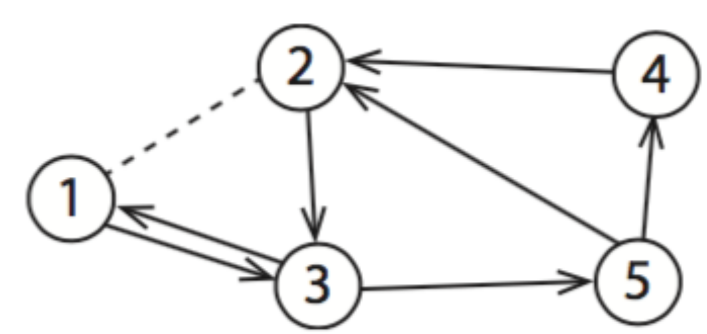
  - Route by aggregated flow
    - ex) Full mesh flows (allowable delay = 80)



(a) Network



(b) Maximum aggregation *not satisfying* allowable delay



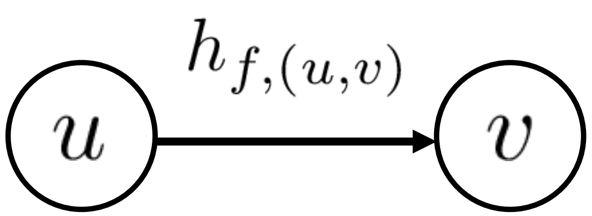
(c) Maximum aggregation satisfying allowable delay

Model setting (1)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- SDN as a directed graph $G = (V, E)$
  - Node: $u \in V$
  - Link: $l = (u, v) \in E \ (u, v \in V, u \neq v)$
    - Link cost: $C_{(u,v)} = C_{(v,u)} \in \mathbb{R}^+$
  - Flow: $f = (f_s, f_d) \in F$
    - Allowable cost: $C_f \in \mathbb{R}^+$
  - Relation between flow and link

$$h_{f,(u,v)} = \begin{cases} 1 & ((u,v) \text{ is over } f \in F) \\ 0 & (\text{otherwise}) \end{cases} \quad (1)$$



Model setting (2)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

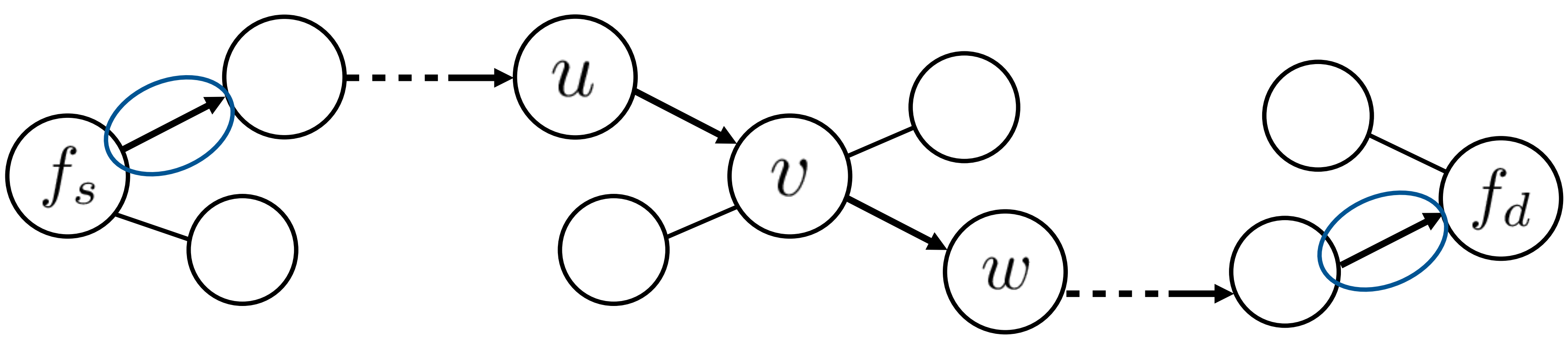
- Constraints to route flow

$$\forall f \in F: \sum_{(f_s,v)\in E} h_{f,(f_s,d)} = \sum_{(u,f_d)\in E} h_{f,(u,f_d)} = 1 \qquad (2)$$

$$\forall f \in F: \sum_{(v,f_s)\in E} h_{f,(v,f_s)} = \sum_{(f_d,u)\in E} h_{f,(f_d,u)} = 0 \qquad (3)$$

$$\forall v \in V \setminus \{f_s, f_d\}, \forall f \in F: \sum_{(u,v)\in E} h_{f,(u,v)} - \sum_{(v,w)\in E} h_{f,(v,w)} = 0 \qquad (4)$$

$$\forall f \in F, \forall v \in V: \sum_{(u,v)\in E} h_{f,(u,v)} = 1 \qquad (5)$$



There is only one flow from a source node and to a destination node

Model setting (2)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

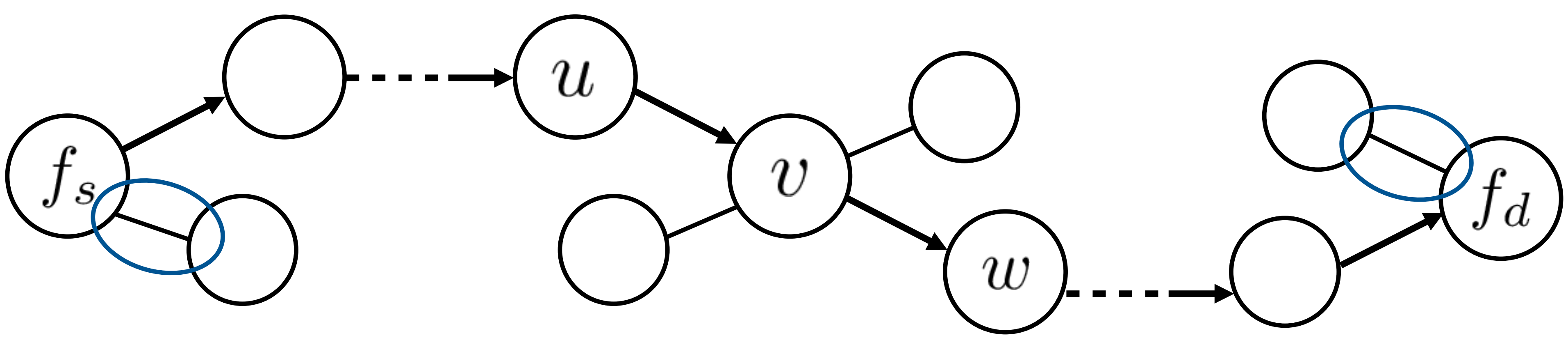
- Constraints to route flow

$$\forall f \in F: \sum_{(f_s,v)\in E} h_{f,(f_s,d)} = \sum_{(u,f_d)\in E} h_{f,(u,f_d)} = 1 \qquad (2)$$

$$\forall f \in F: \sum_{(v,f_s)\in E} h_{f,(v,f_s)} = \sum_{(f_d,u)\in E} h_{f,(f_d,u)} = 0 \qquad (3)$$

$$\forall v \in V \setminus \{f_s, f_d\}, \forall f \in F: \sum_{(u,v)\in E} h_{f,(u,v)} - \sum_{(v,w)\in E} h_{f,(v,w)} = 0 \qquad (4)$$

$$\forall f \in F, \forall v \in V: \sum_{(u,v)\in E} h_{f,(u,v)} = 1 \qquad (5)$$



There is no flow to a source node and from a destination node

Model setting (2)

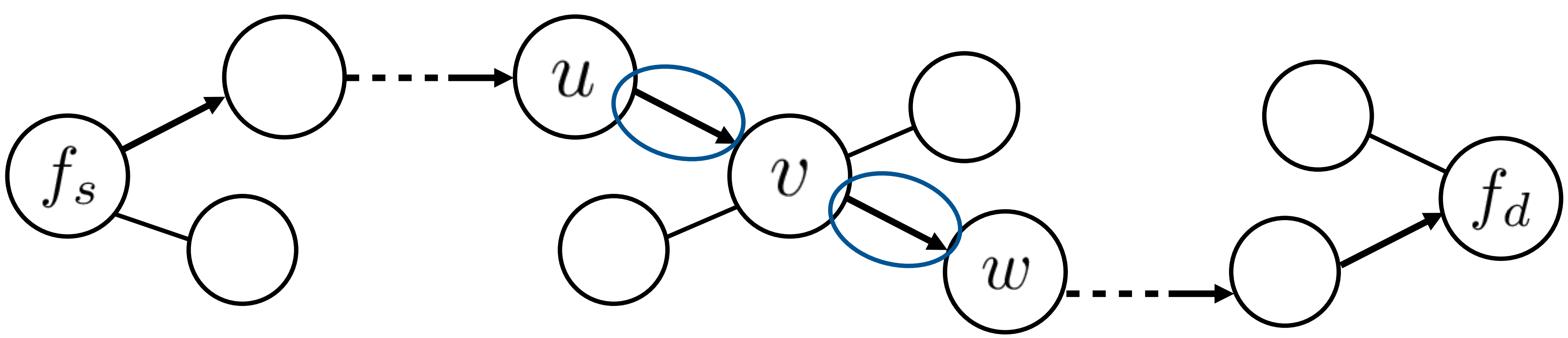
- Constraints to route flow

$$\forall f \in F: \sum_{(f_s,v)\in E} h_{f,(f_s,d)} = \sum_{(u,f_d)\in E} h_{f,(u,f_d)} = 1 \qquad (2)$$

$$\forall f \in F: \sum_{(v,f_s)\in E} h_{f,(v,f_s)} = \sum_{(f_d,u)\in E} h_{f,(f_d,u)} = 0 \qquad (3)$$

$$\forall v \in V \setminus \{f_s, f_d\}, \forall f \in F: \sum_{(u,v)\in E} h_{f,(u,v)} - \sum_{(v,w)\in E} h_{f,(v,w)} = 0 \qquad (4)$$

$$\forall f \in F, \forall v \in V: \sum_{(u,v)\in E} h_{f,(u,v)} = 1 \qquad (5)$$



Number of incoming flows equals to number of outgoing flows on a node

Model setting (2)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

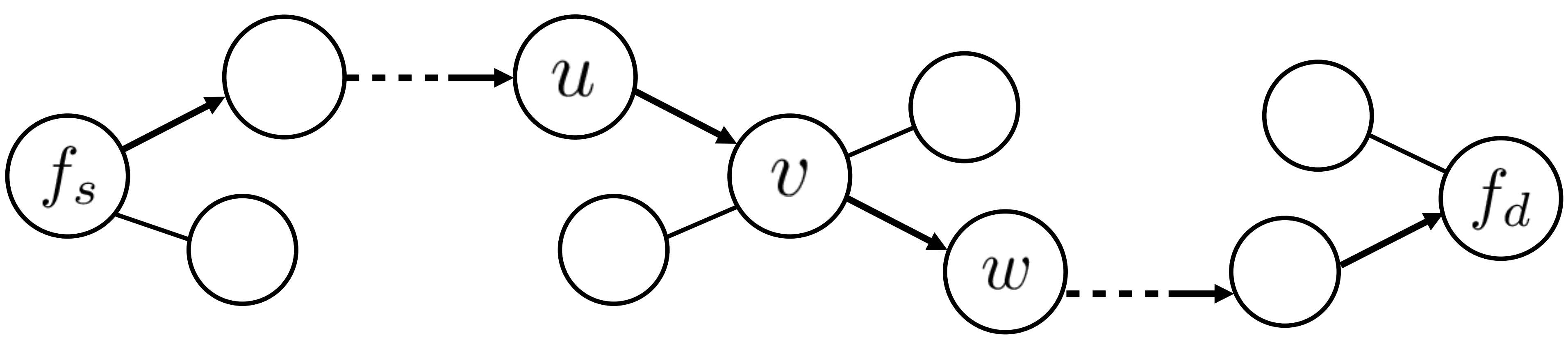
- Constraints to route flow

$$\forall f \in F : \sum_{(f_s,v)\in E} h_{f,(f_s,d)} = \sum_{(u,f_d)\in E} h_{f,(u,f_d)} = 1 \qquad (2)$$

$$\forall f \in F : \sum_{(v,f_s)\in E} h_{f,(v,f_s)} = \sum_{(f_d,u)\in E} h_{f,(f_d,u)} = 0 \qquad (3)$$

$$\forall v \in V \setminus \{f_s, f_d\}, \forall f \in F : \sum_{(u,v)\in E} h_{f,(u,v)} - \sum_{(v,w)\in E} h_{f,(v,w)} = 0 \qquad (4)$$

$$\forall f \in F, \forall v \in V : \sum_{(u,v)\in E} h_{f,(u,v)} = 1 \qquad (5)$$



A node is visited only one time

Model setting (3)

- Constraint of allowable cost

$$\forall f \in F : \sum_{(u,v)\in E} h_{f,(u,v)} C_{(u,v)} \leq C_f \tag{6}$$

- Aggregate flows
  - Flows on same link and same direction

$$g_{(u,v)} = \begin{cases} 1 & ((u,v) \text{ is over } \exists f \in F) \\ 0 & (\text{otherwise}) \end{cases} \tag{7}$$

$$\forall f \in F, \forall (u,v) \in E : g_{(u,v)} - h_{f,(u,v)} \geq 0 \tag{8}$$

  - Objective: minimize number of flows

$$minimize \sum_{(u,v)\in E} g_{(u,v)} \quad \text{s.t.(2)-(6)(8)} \tag{9}$$

Model setting (4)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Deformation

- $\boldsymbol{G} = (g_{(u,v)}, g_{(v,u)}, \cdots)\ (\forall(u,v) \in E)$
- $\boldsymbol{H} = (H_{f_1}, H_{f_2}, \cdots, H_{f_m})\ (f_1, f_2, \cdots, f_m \in F)$
  - $\boldsymbol{H_f} = (h_{f,(u,v)}, h_{f,(v,u)}, \cdots)$
- $\boldsymbol{x} = (\boldsymbol{G}, \boldsymbol{H})$
  - **y** and **z** are vectors (0,…1,…,0) which correspond to the index of h and g

Apply to $(6)(9)$

$$minimize \sum_{(u,v)\in E} \boldsymbol{x}\boldsymbol{z}^T_{(u,v)}$$

$$\text{s.t.} \forall f \in F : \boldsymbol{x}\boldsymbol{y}^T_{f,(u,v)} C_{(u,v)} \leq C_f$$

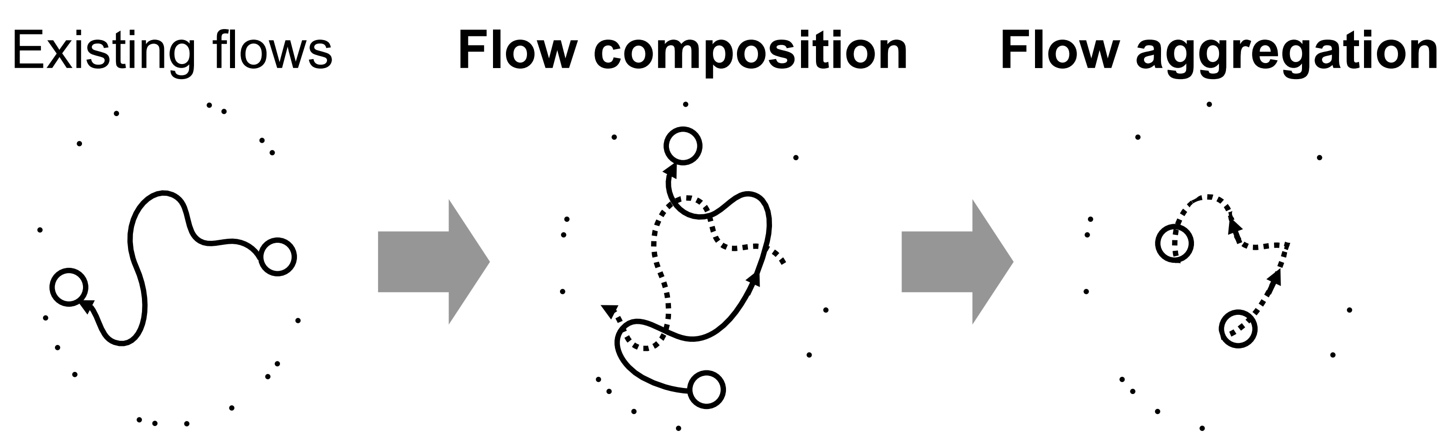
0,1-Knapsack Problem $\Rightarrow$ **NP-hard**

Heuristics

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Overview

1. Ascending sort by (minimum cost - allowable cost) ⇒ *Fs*
   = flexibility of changing route
2. Select the first element of *Fs* ⇒ *f*
3. [**Flow compotision**]
   - Compose route of *f* using existing flows
4. [**Flow aggregation**]
   - Aggregate flows in *Fs* into existing flows
5. Return to 2. unless *Fs* is empty



Flow composition

- Path composition by Best-First Search (BFS)
  - Evaluation value of node y : $b_y = \{t_y, c_y\}$
    - t_y : number of times existing rules are used
    - c_y : minimum cost to reach from f_s to y
  - Prioritize two nodes

$$comp(u,v) = \begin{cases} 1 & (t_u < t_v \bigvee (t_u = t_v \bigwedge c_u > c_v)) \\ 0 & (\text{otherwise}) \end{cases}$$

  - Time complexity: $\mathcal{O}(|V|(|E| + |V|))$

Flow composition: Ex.1

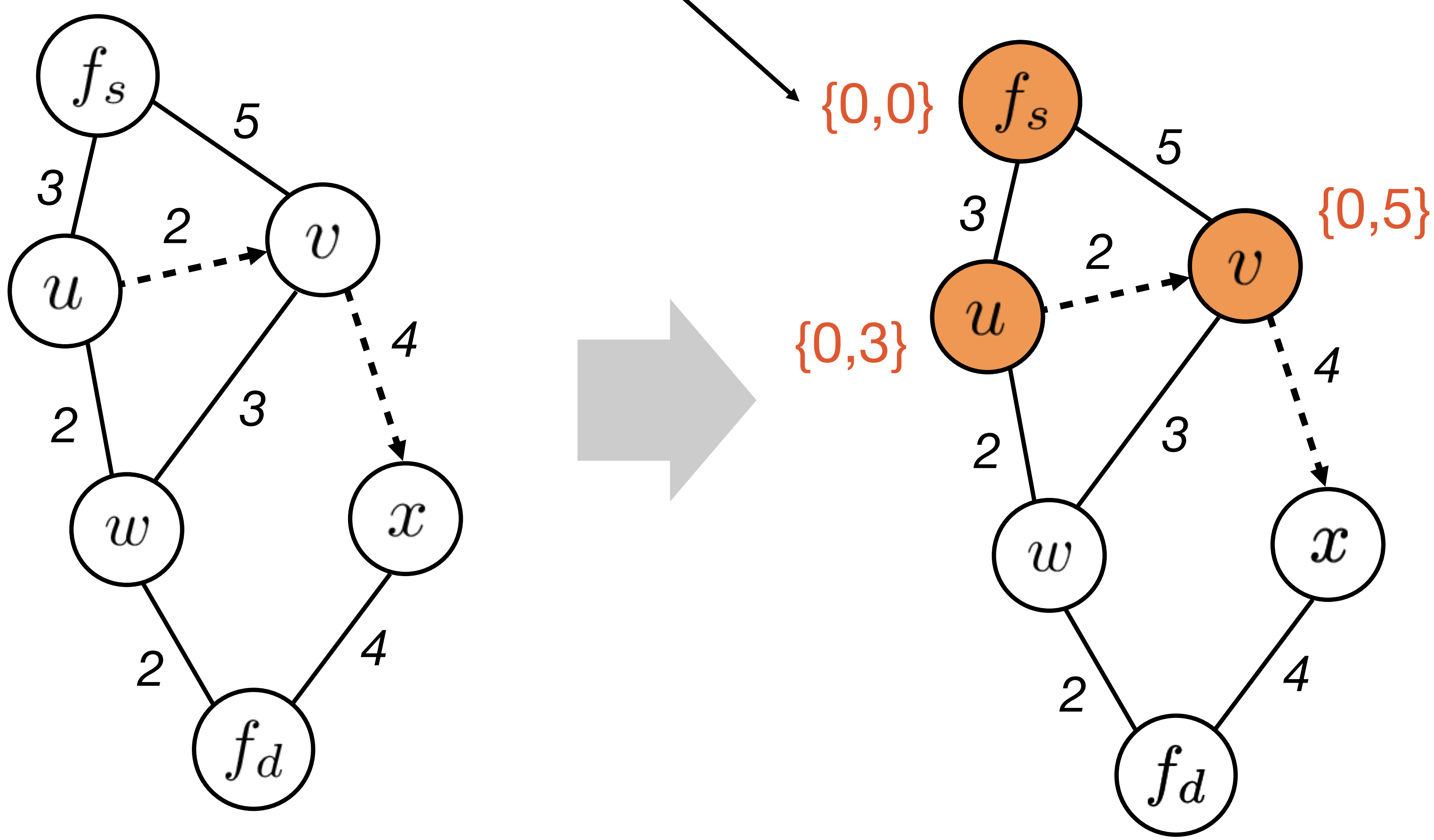
IEEE ICC 2017 CQRM
“A Flow Aggregation Method Based on End-to-End Delay in SDN”
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Flow composition by Best-First Search
  - $C_f = 13$

Evaluation value: {t, c}

t : number of times existing rules are used

c : minimum cost to reach from source to current



Dashed arrows: existing flows

Flow composition: Ex.1

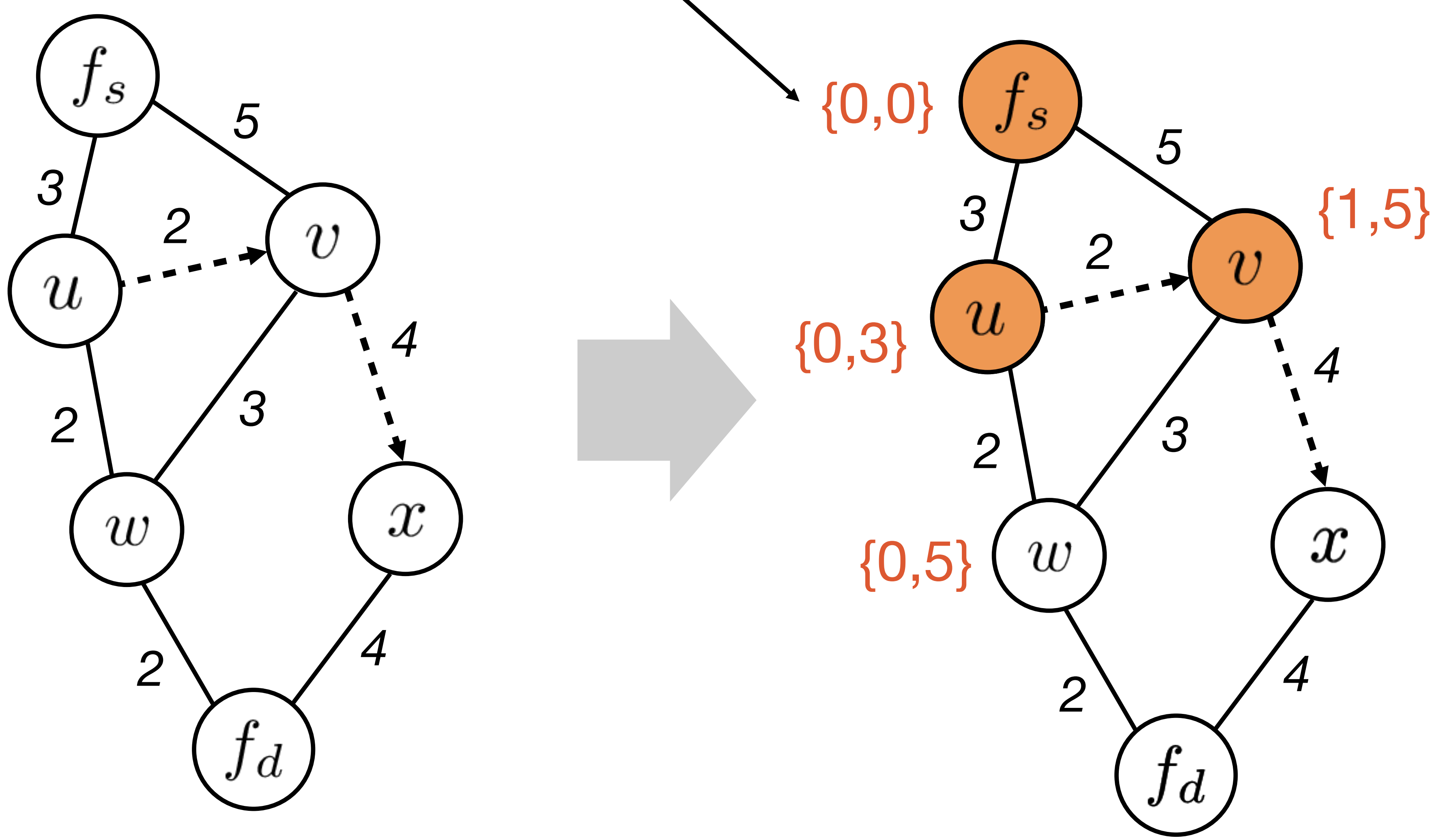
IEEE ICC 2017 CQRM
“A Flow Aggregation Method Based on End-to-End Delay in SDN”
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Flow composition by Best-First Search
  - $C_f = 13$

Evaluation value: {t, c}

t : number of times existing rules are used

c : minimum cost to reach from source to current



Dashed arrows: existing flows

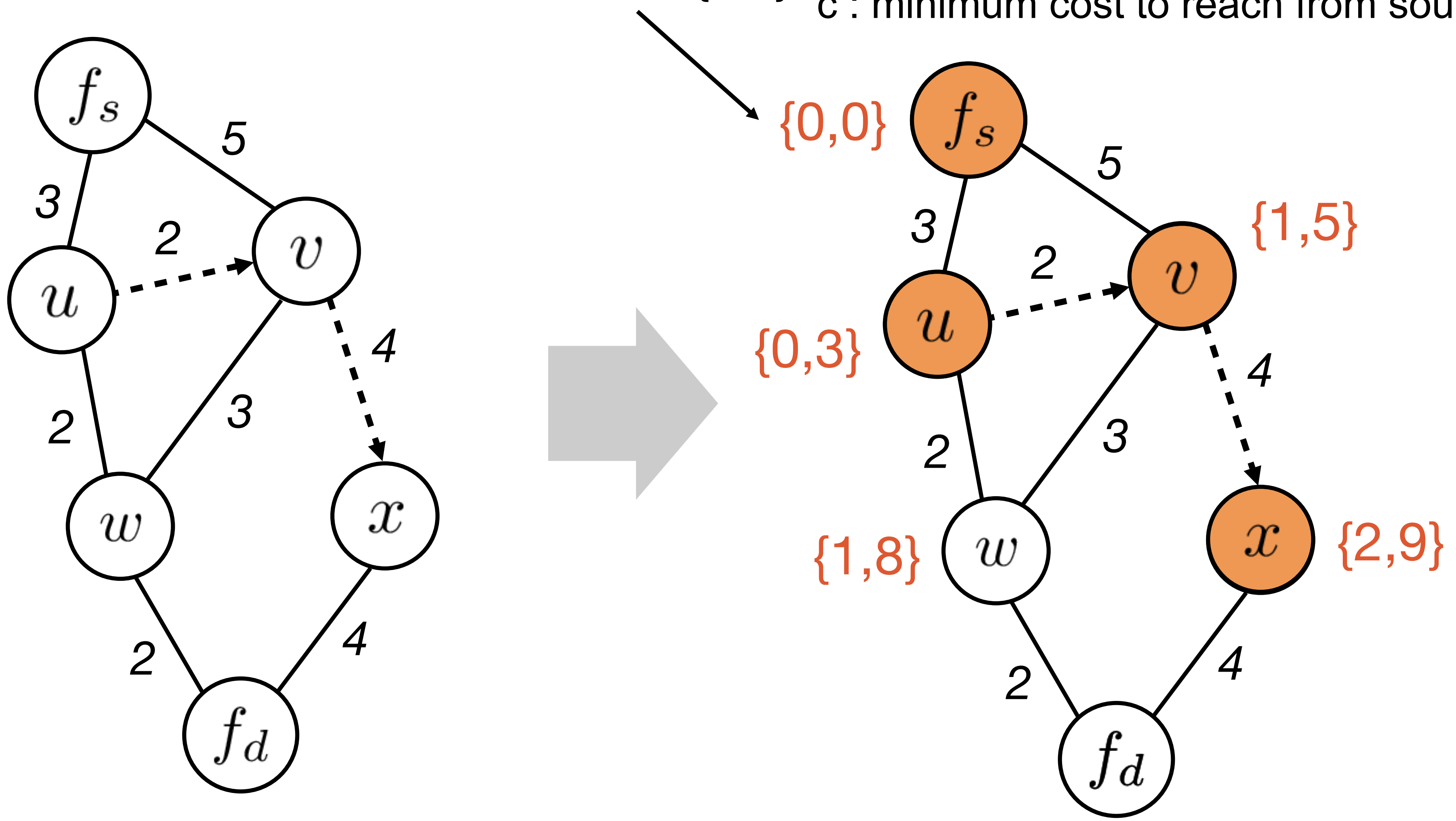
Flow composition: Ex.1

- Flow composition by Best-First Search
  - $C_f = 13$

Evaluation value: {t, c}

t : number of times existing rules are used

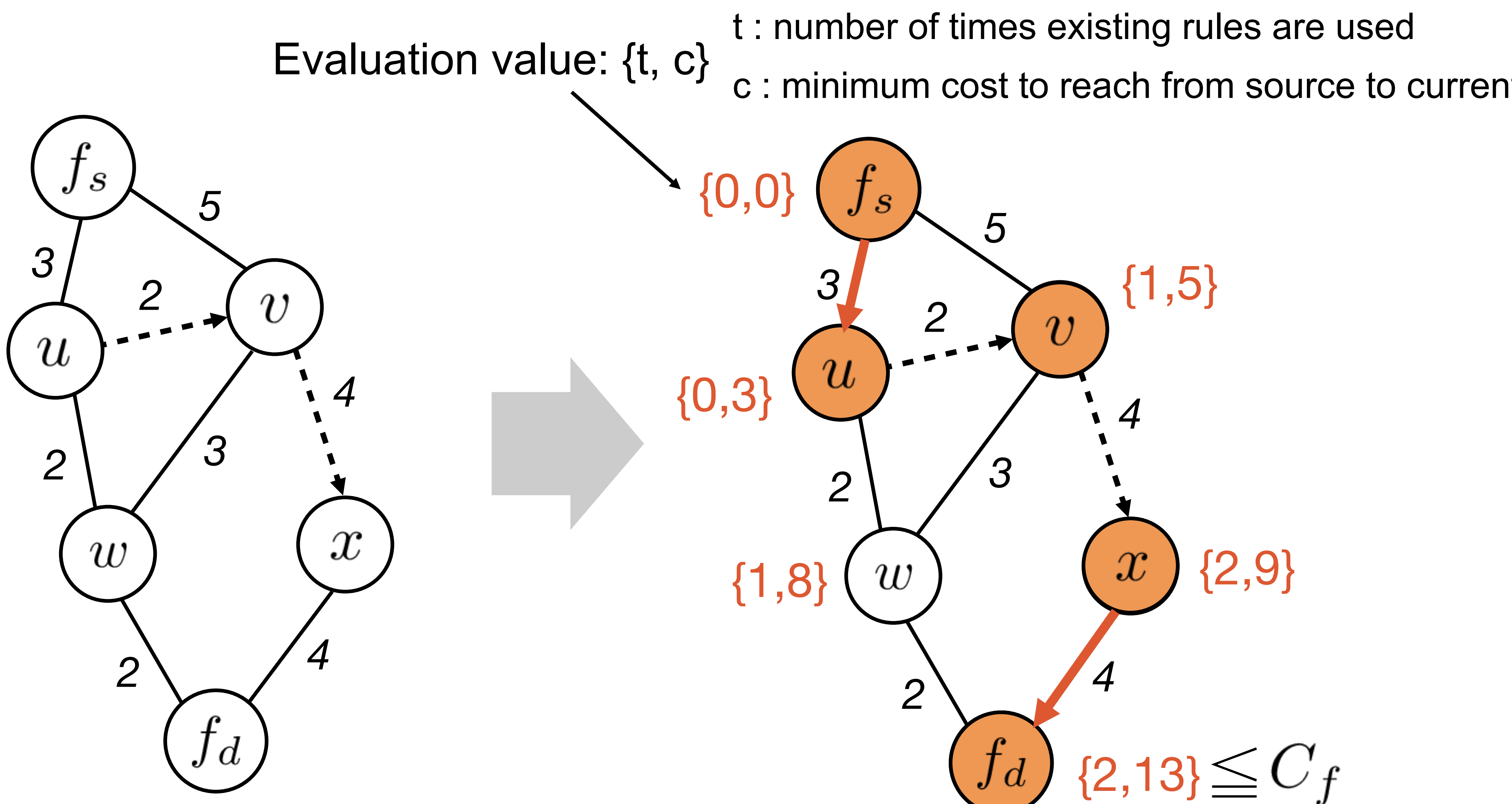
c : minimum cost to reach from source to current



Dashed arrows: existing flows

Flow composition: Ex.1

- Flow composition by Best-First Search
  - $C_f = 13$



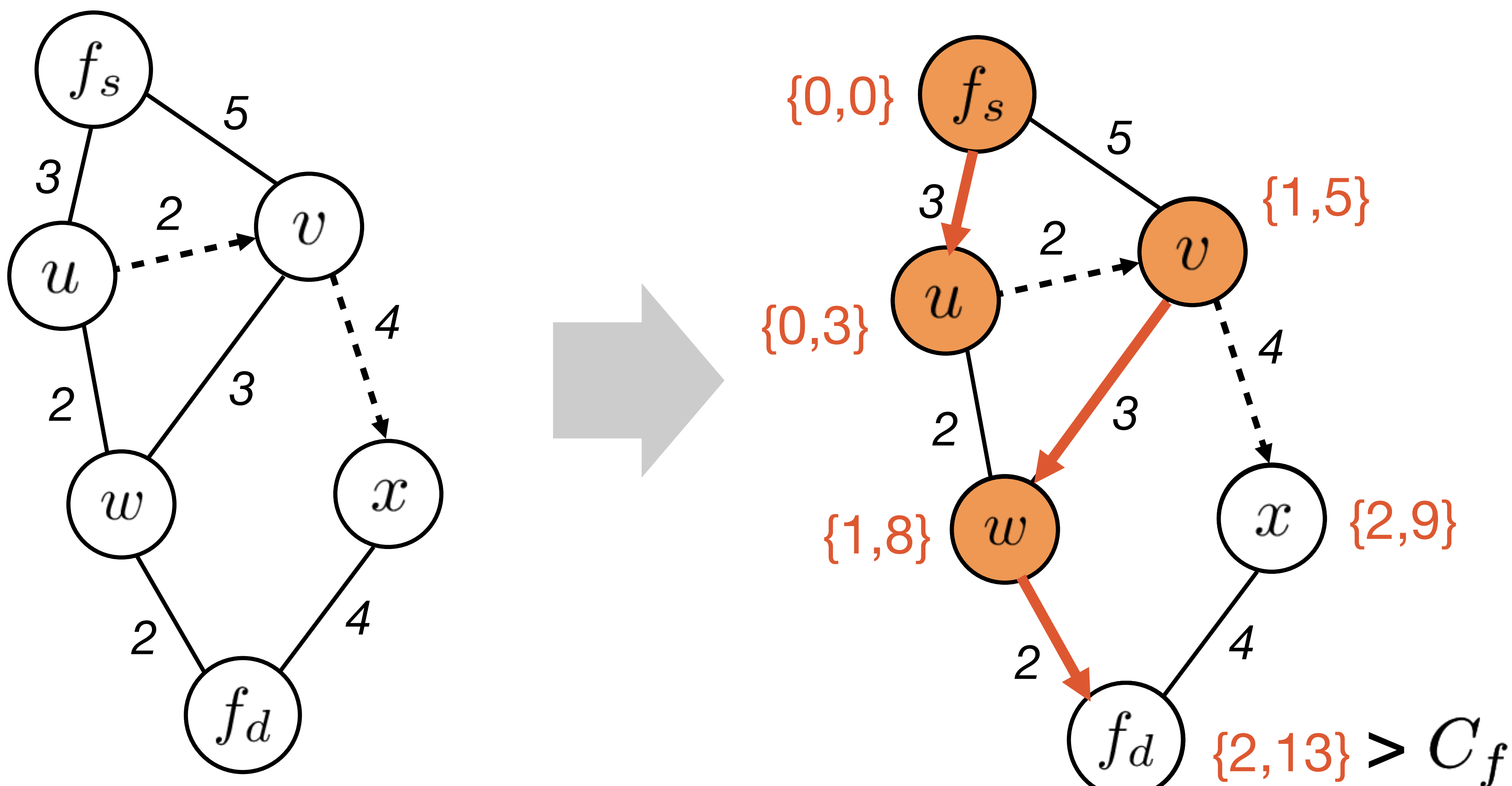
Dashed arrows: existing flows

Number of adding flows: 2

Flow composition: Ex.2

IEEE ICC 2017 CQRM
“A Flow Aggregation Method Based on End-to-End Delay in SDN”
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

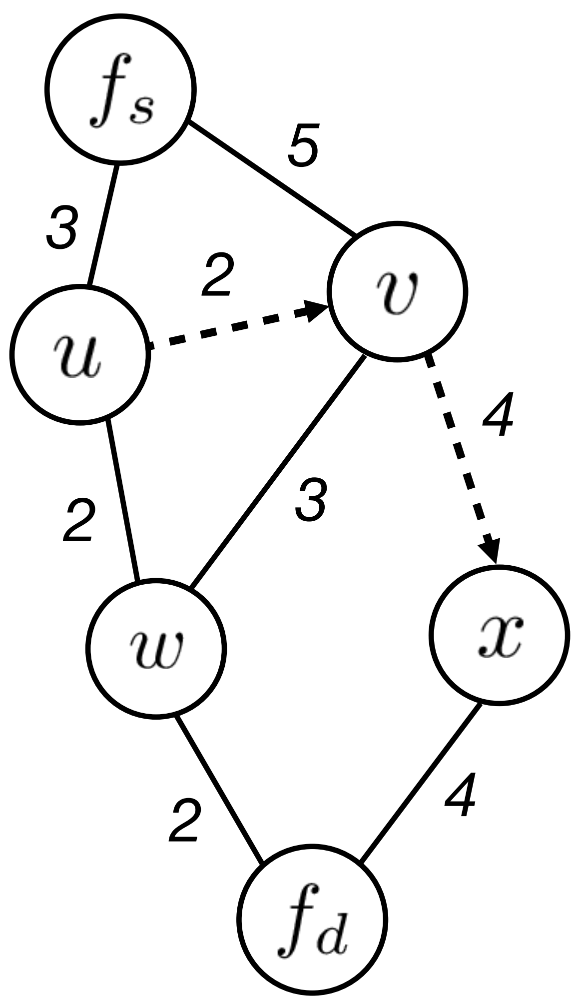
- Flow composition by Best-First Search
  - $C_f = 12$

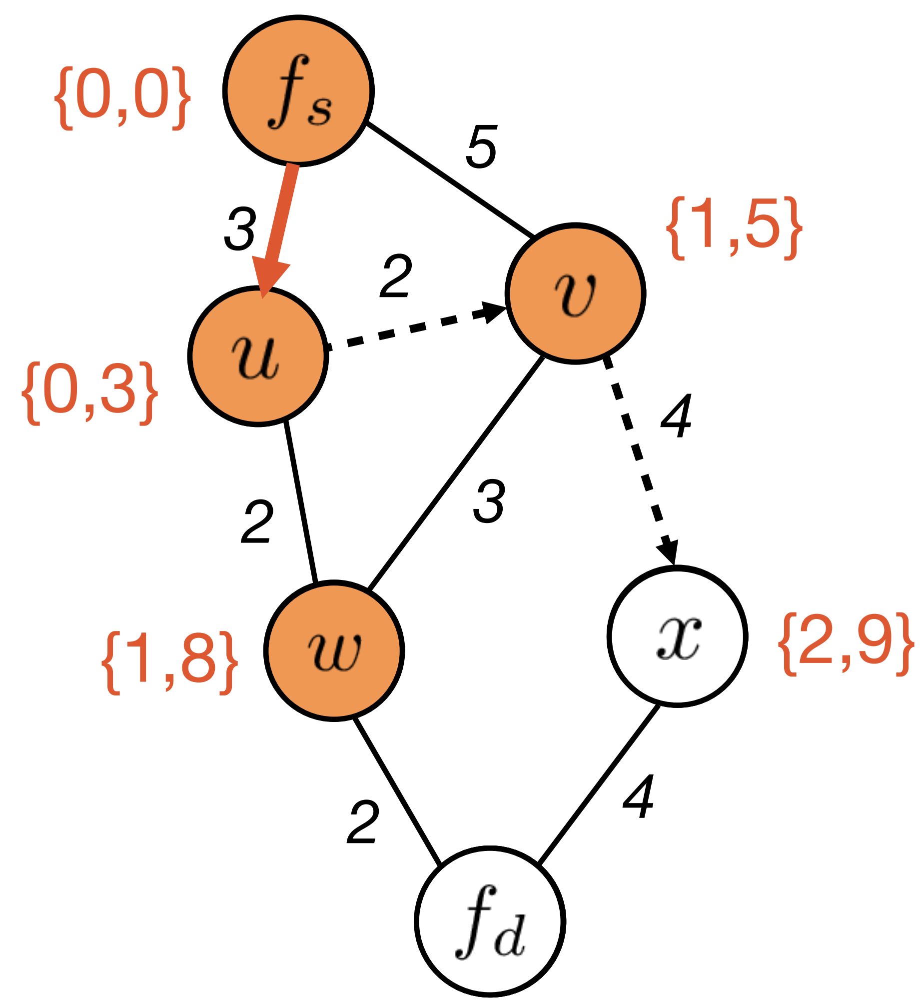


Dashed arrows: existing flows

Flow composition: Ex.2

- Flow composition by Best-First Search
  - $C_f = 12$



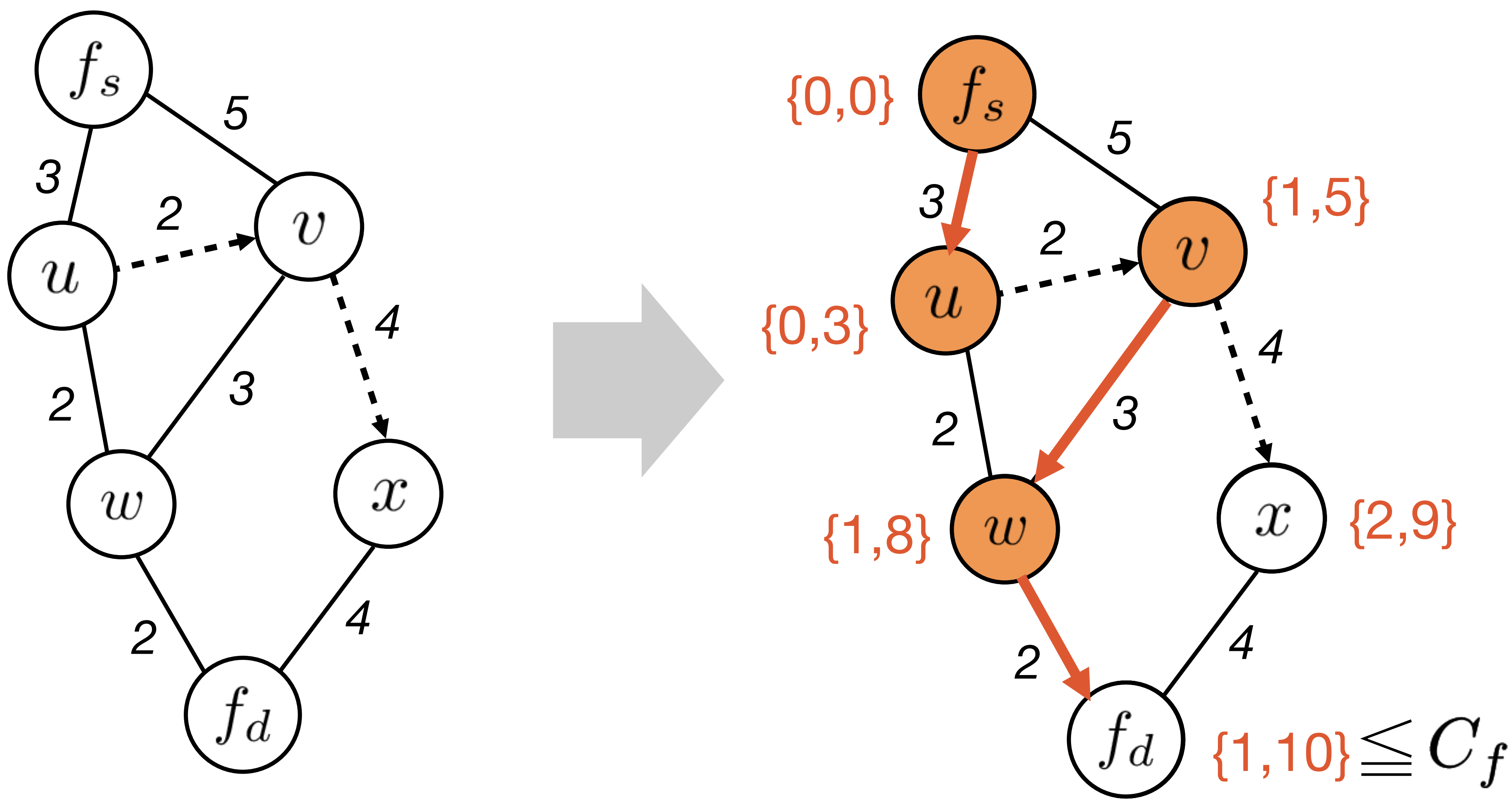


Dashed arrows: existing flows

Flow composition: Ex.2

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Flow composition by Best-First Search
  - $C_f = 12$



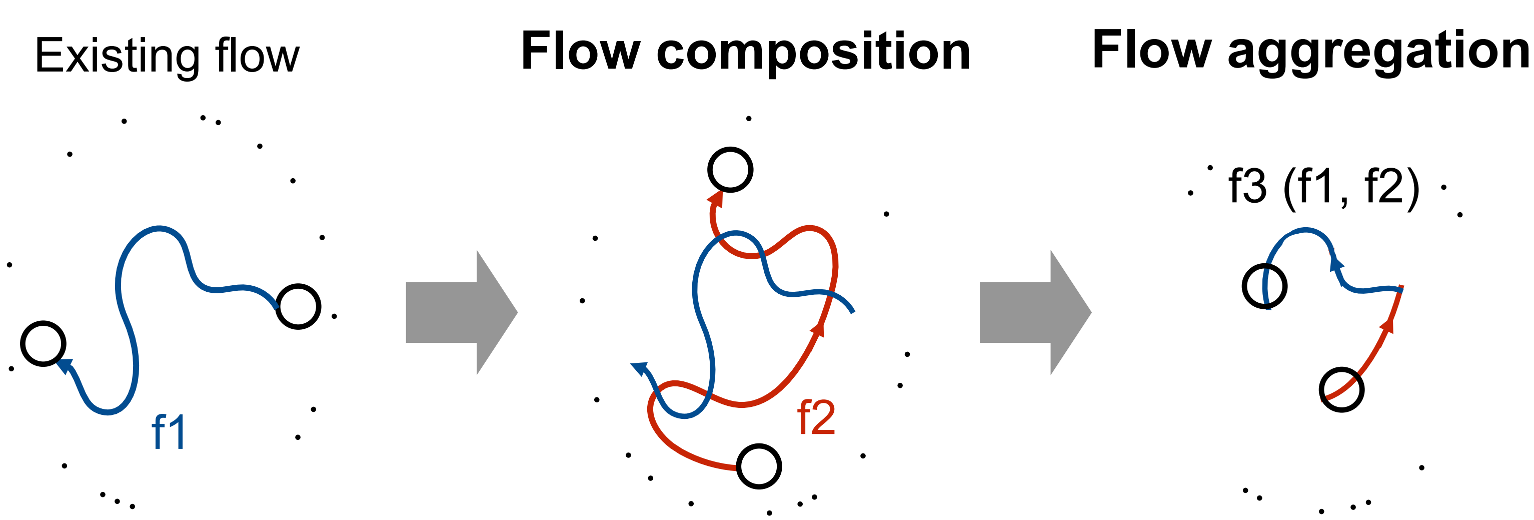
Dashed arrows: existing flows

Number of adding flows: 3

Heuristics

IEEE ICC 2017 CQRM
“A Flow Aggregation Method Based on End-to-End Delay in SDN”
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

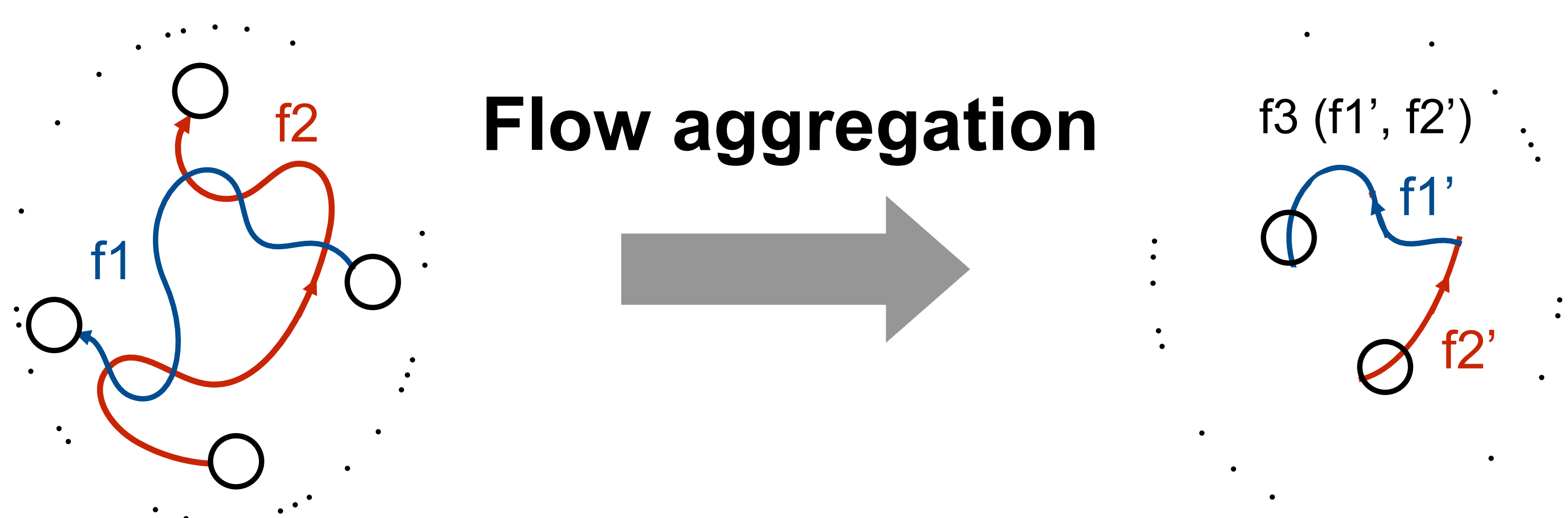
- Overview
  1. Ascending sort by (minimum cost - allowable cost) ⇒ *Fs*
  2. Select the first element of *Fs* ⇒ *f*
  3. [**Flow compotision**]
     - Compose route of *f* using existing flows
  4. [**Flow aggregation**]
     - Aggregate flows in *Fs* into existing flows
  5. Return to 2. unless *Fs* is empty



Flow aggregation

IEEE ICC 2017 CQRM
“A Flow Aggregation Method Based on End-to-End Delay in SDN”
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Delete flows in Fs over existing flows
  - Calculate flows can be reached within allowable delay by Dijkstra’s algorithm
  - Time complexity: $\mathcal{O}(|F|(|E| \log |V|))$



f3 can be routed using part of f1 and f2

Simulation settings

Table: topologies

Topology name	$\lvert V \rvert$	$\lvert E \rvert$	$max(\lvert F \rvert)$
Random-small	50	101	$\mathcal{O}(10^3)$
Random-large	500	7472	$\mathcal{O}(10^5)$
Scalefree-small	50	96	$\mathcal{O}(10^3)$
Scalefree-large	500	995	$\mathcal{O}(10^5)$

- Simulation
  - Topology
    - Model: ER / BA
    - Size: |V| = 50, 500
  - Parameters
    - Link cost: [5, 15], discrete uniform distribution
    - alpha: ratio of allowable cost of flows to the maximum value of the minimum delay of flows
      - Ex) min cost = 10, alpha = 2 ⇒ Cf = 20
  - Comparison
    - Minimum cost / Minimum hop routing with aggreagtion

Performance evaluation (1)

Topology name	$\|V\|$	$\|E\|$	$max(\|F\|)$
Random-small	50	101	$\mathcal{O}(10^3)$
Random-large	500	7472	$\mathcal{O}(10^5)$
Scalefree-small	50	96	$\mathcal{O}(10^3)$
Scalefree-large	500	995	$\mathcal{O}(10^5)$

- Effect of allowable cost
  - |F| = 1000

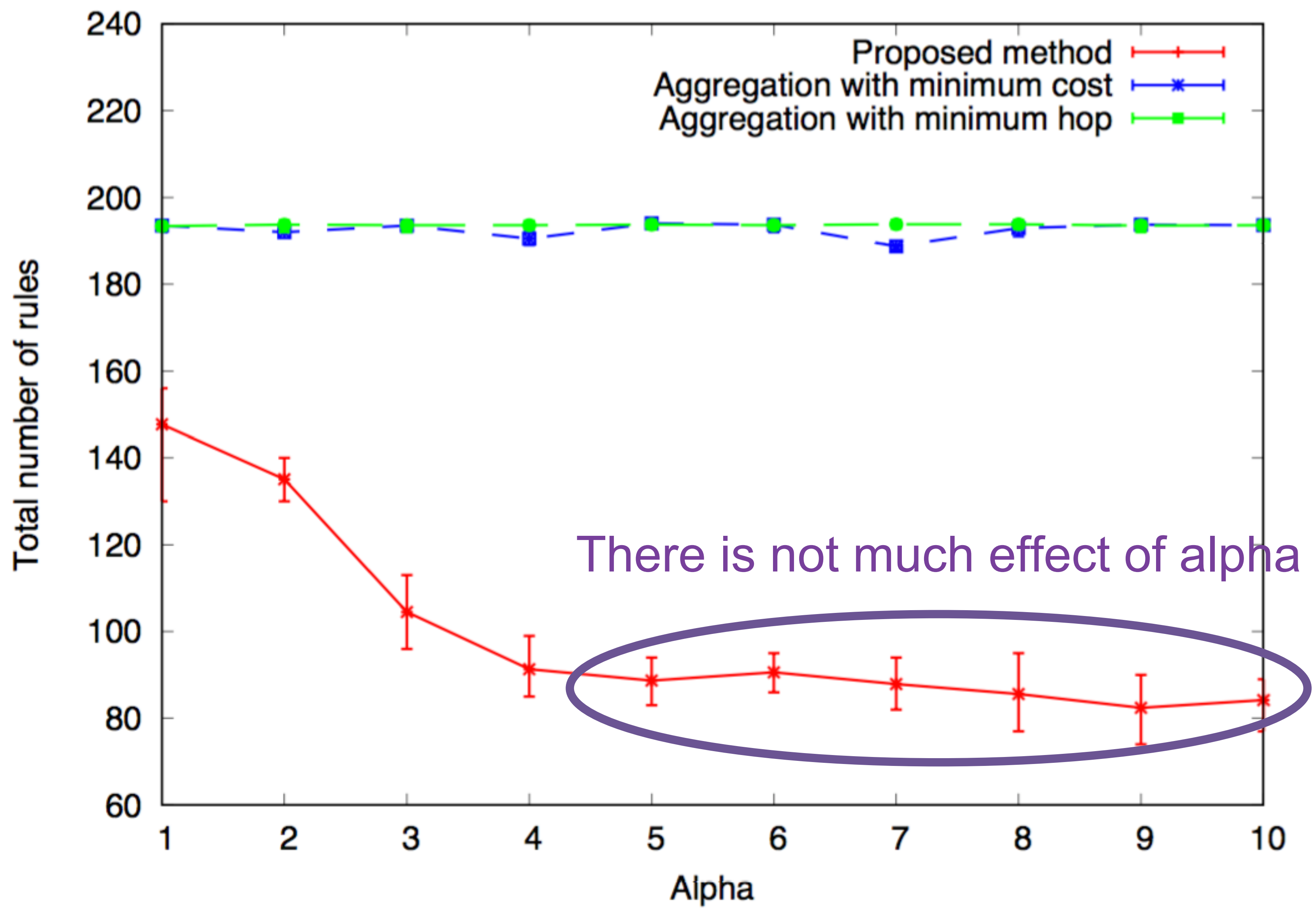


Fig. 1 Effect of allowable cost of flows (Scalefree-small)

Performance evaluation (1)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

Topology name	$\|V\|$	$\|E\|$	$max(\|F\|)$
Random-small	50	101	$\mathcal{O}(10^3)$
Random-large	500	7472	$\mathcal{O}(10^5)$
Scalefree-small	50	96	$\mathcal{O}(10^3)$
Scalefree-large	500	995	$\mathcal{O}(10^5)$

- Effect of allowable cost
  - |F| = 1000

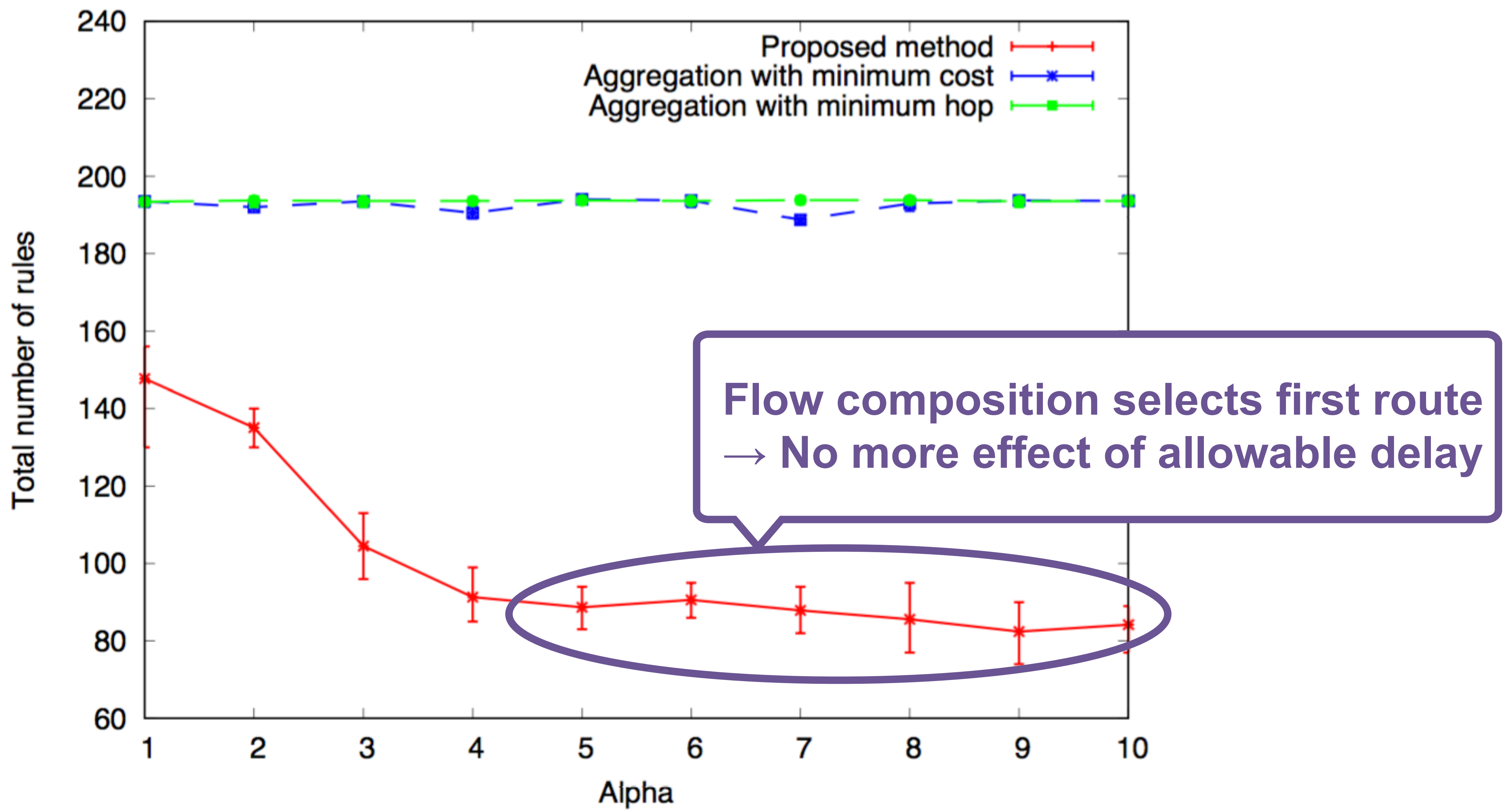


Fig. 1 Effect of allowable cost of flows (Scalefree-small)

Performance evaluation (2)

IEEE ICC 2017 CQRM
"A Flow Aggregation Method Based on End-to-End Delay in SDN"
T. Kosugiyama, K. Tanabe, H. Nakayama, T. Hayashi, K. Yamaoka

- Effect of <u>number of flows</u>
  - α = 5

Topology name	$\lvert V \rvert$	$\lvert E \rvert$	$max(\lvert F \rvert)$
Random-small	50	101	$\mathcal{O}(10^3)$
Random-large	500	7472	$\mathcal{O}(10^5)$
Scalefree-small	50	96	$\mathcal{O}(10^3)$
Scalefree-large	500	995	$\mathcal{O}(10^5)$

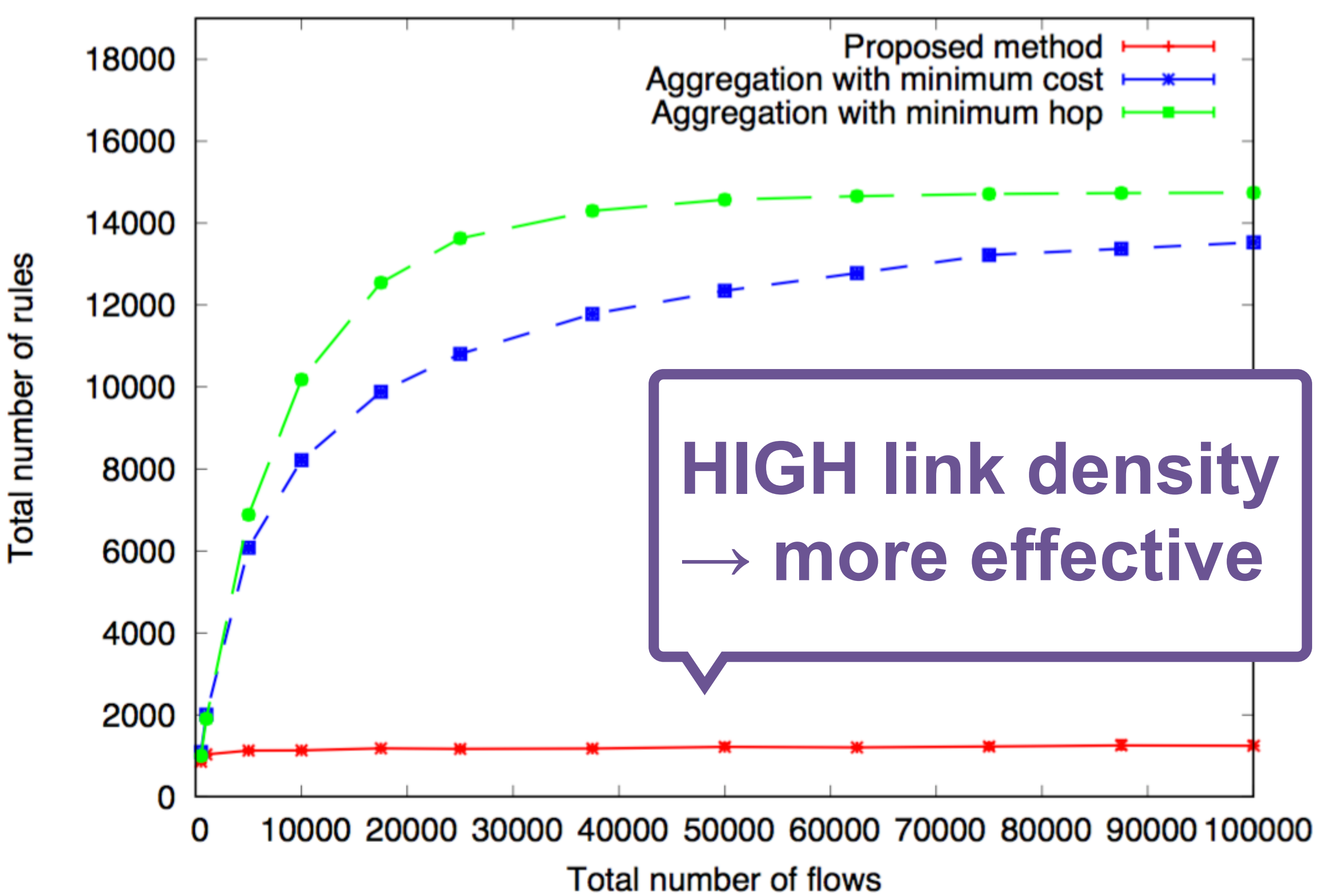


Fig. 2 Random-large

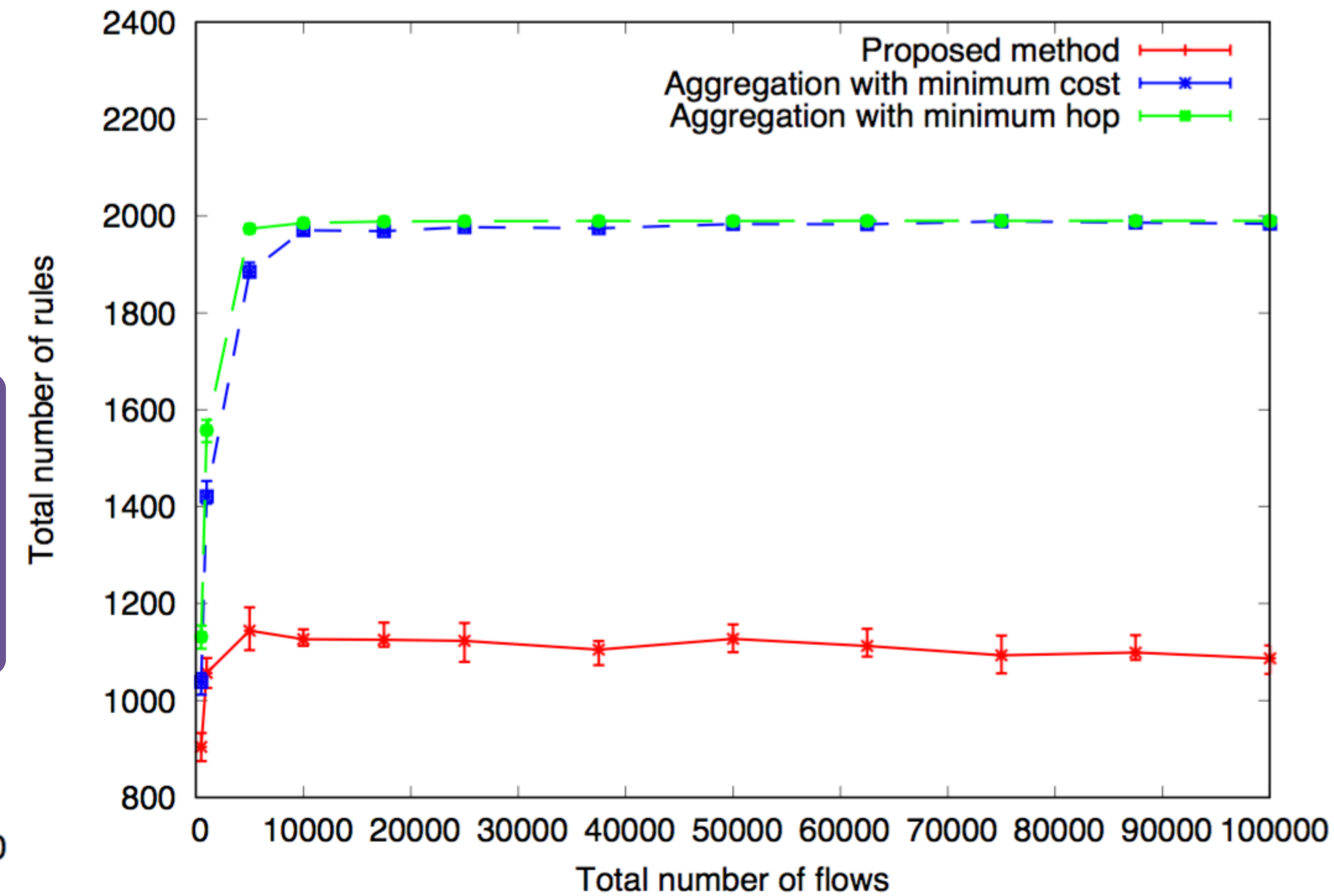


Fig. 3 Scalefree-large

Summary & Future work

- We propose a flow aggregation method to minimize number of flows **satisfying End-to-End delay**
- Simulation in several topologies
  - Higher performance than simple comparison method
  - **Flexibility of changing route** is important
- Future work
  - Expand our model more realistic
  - Bypassing on weak topology