The Augmentation-Speed Tradeoff for Consistent Network Updates

Arash Pourdamghani, TU Berlin

Joint work with Monika Henzinger, Ami Paz, Stefan Schmid

SOSR 2022

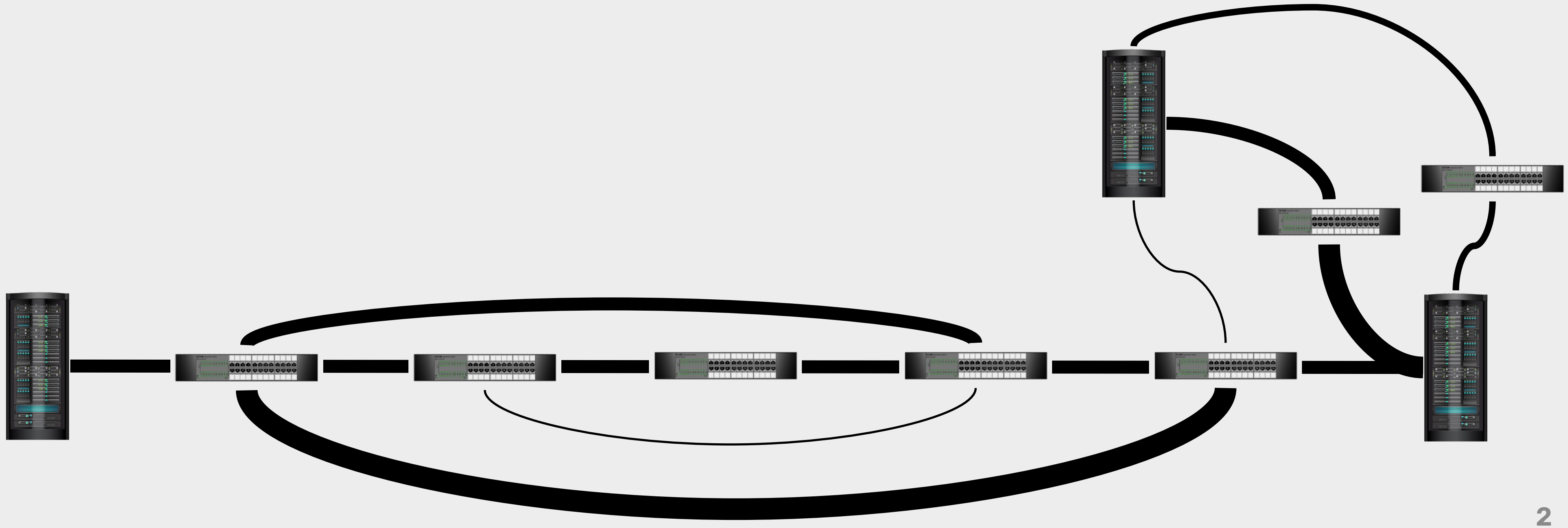
LISN LABORATOIRE INTERDISCIPLINAIRE DES SCIENCES DU NUMÉRIQUE

universität wien

Technische Universität Berlin

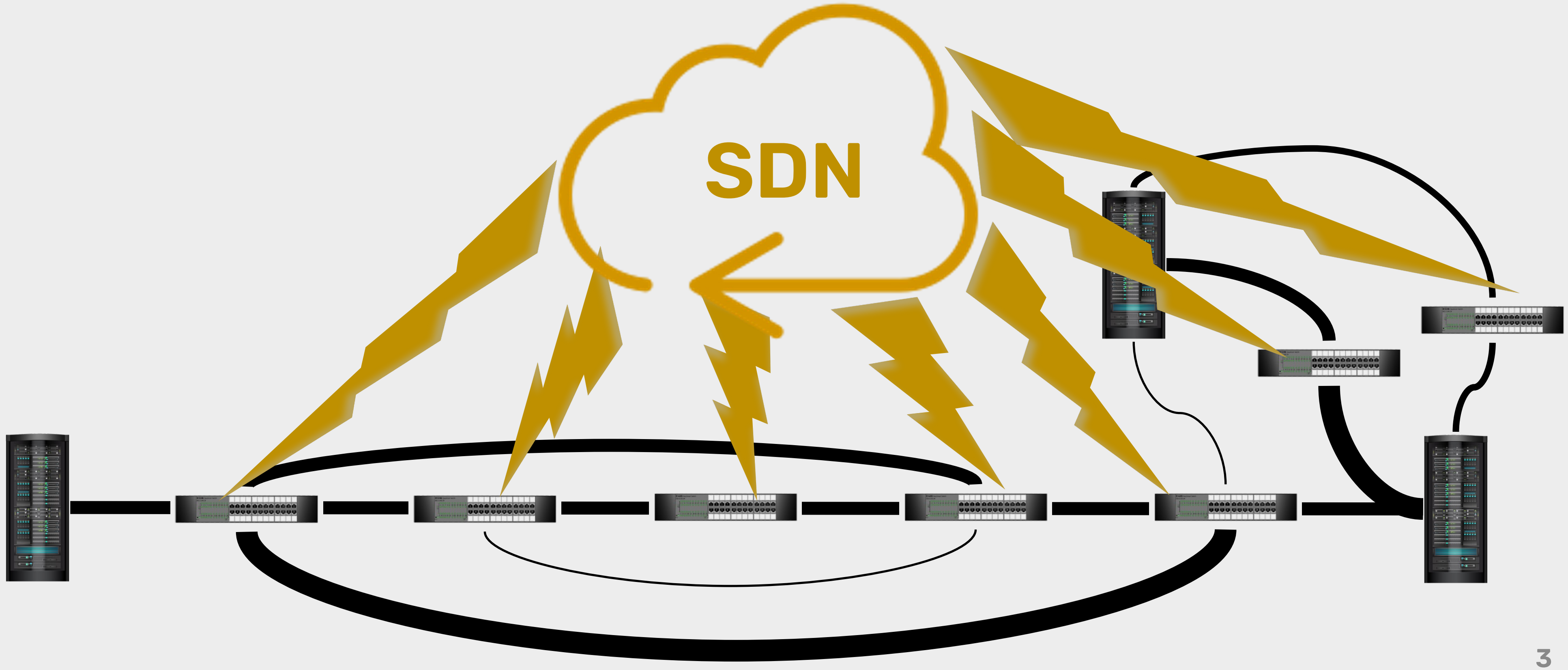
Network updates via SDN

- Networks are prone to be more dynamic



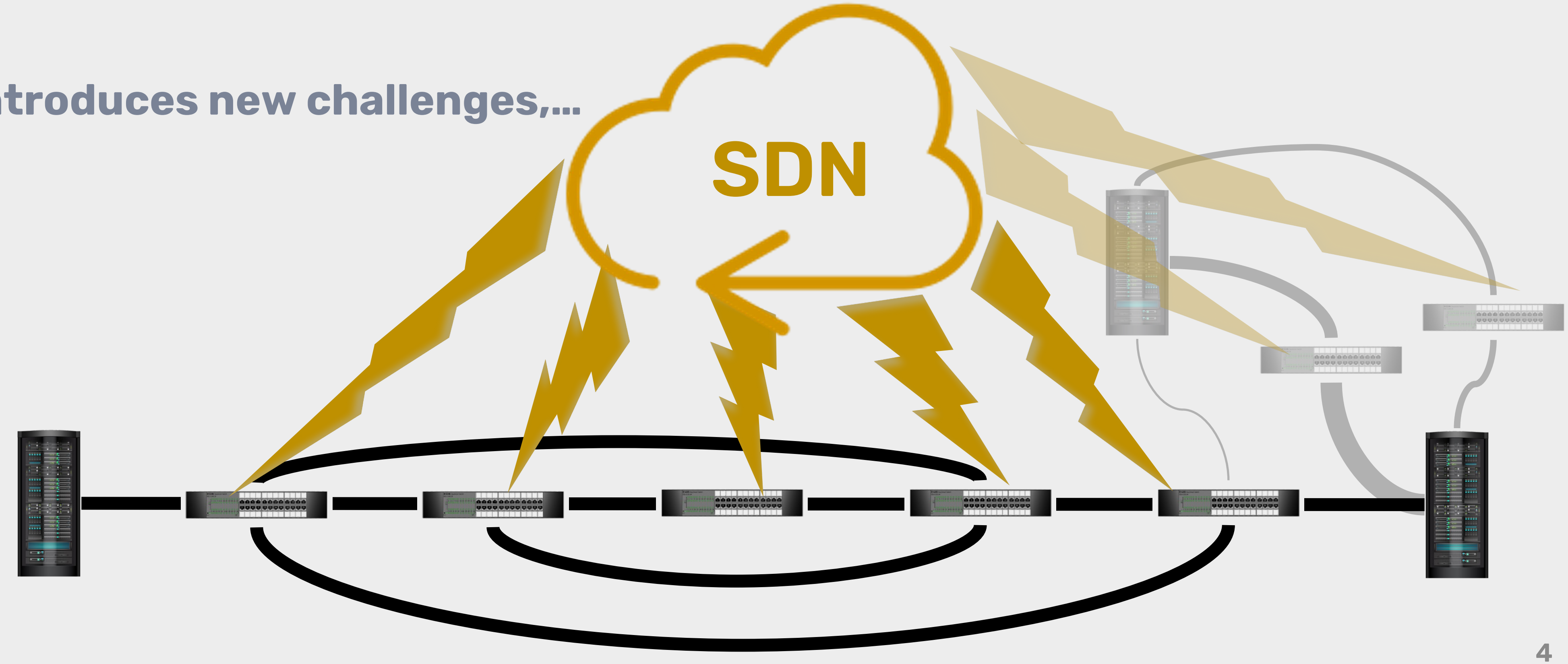
Network updates via SDN

- Networks are prone to be more dynamic
- SDN simplifies and allows for fast updates

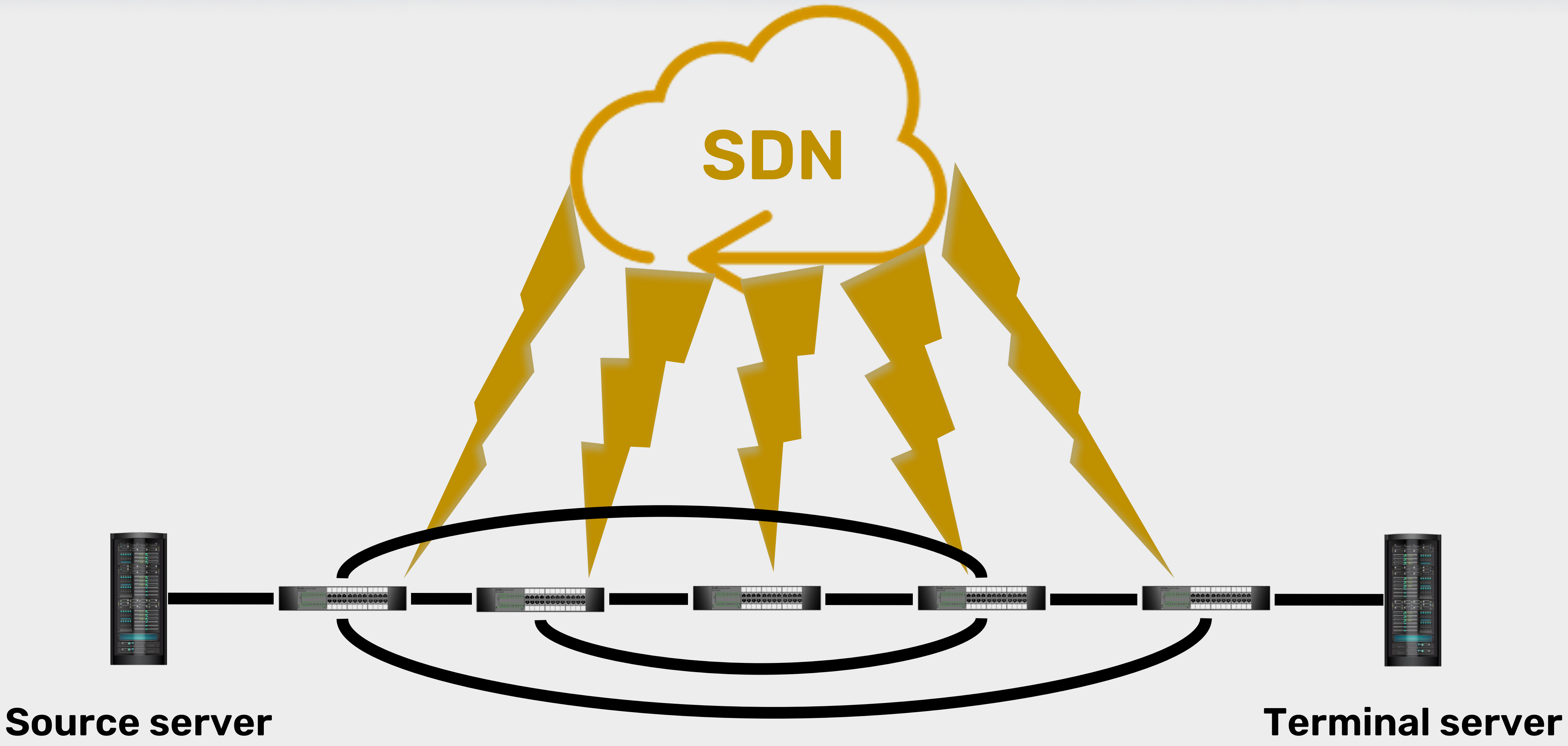


Network updates via SDN

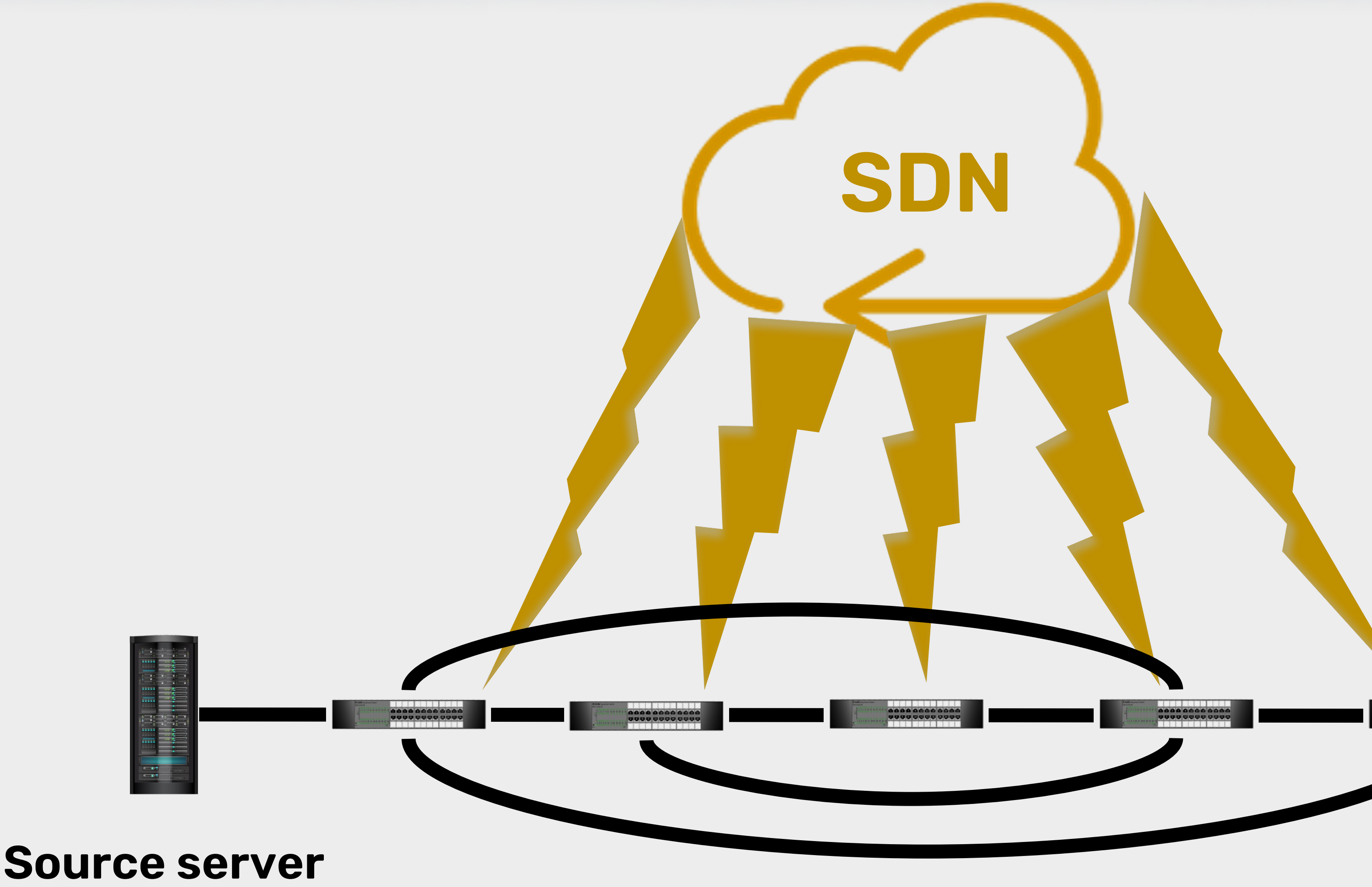
- Networks are prone to be more dynamic
- SDN simplifies and allows for fast updates
- However, SDN introduces new challenges,...

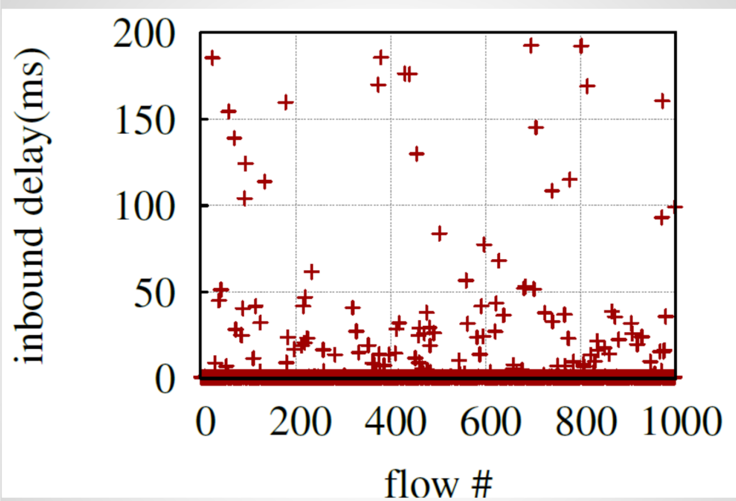


A challenge in SDN updates: non-consistent update times!



A challenge in SDN updates: non-consistent update times!

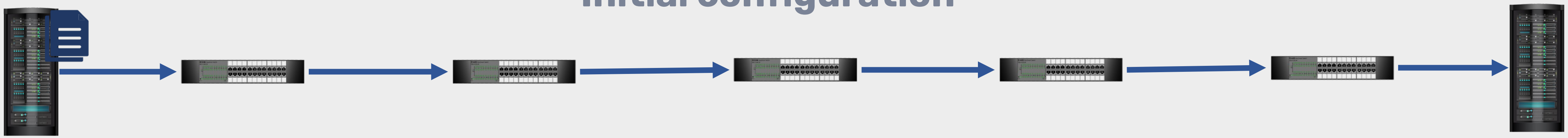




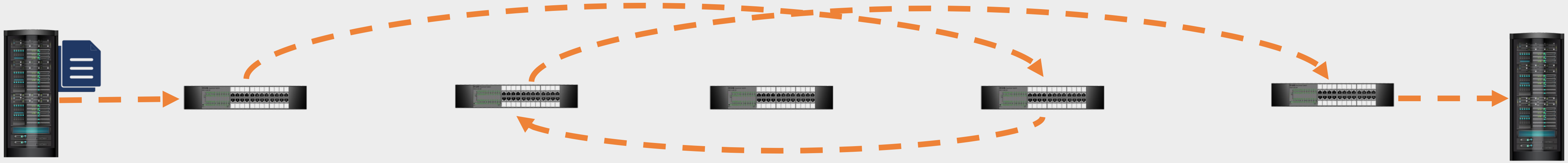
He et al., ACM SOSR 2015: without network latency
Jin et al., ACM SIGCOMM 2014: even higher variance

First side effect

Initial configuration

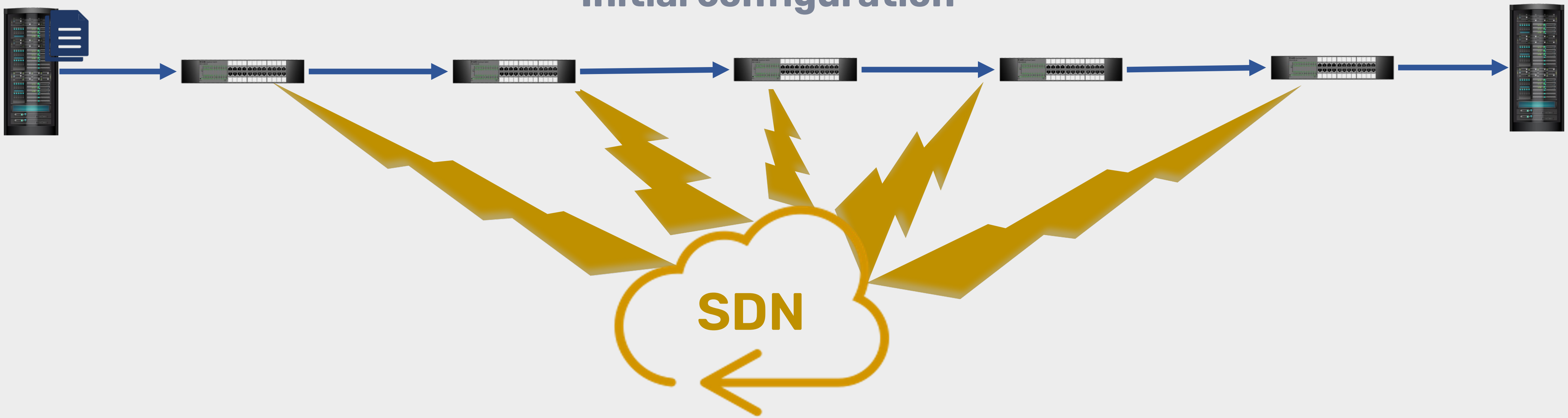


Final configuration

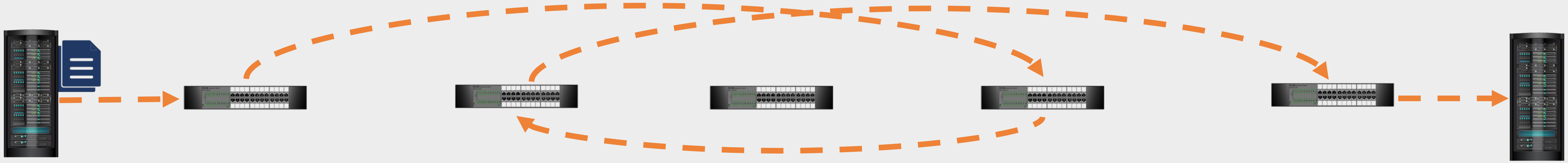


First side effect

Initial configuration

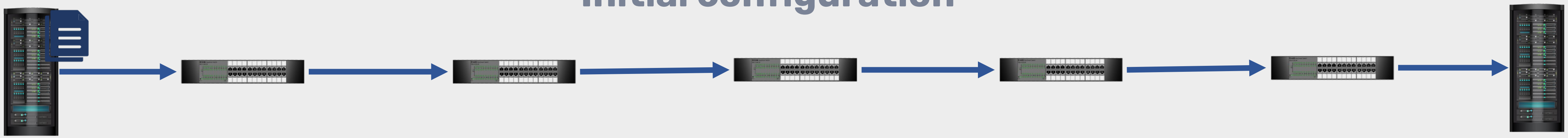


Final configuration

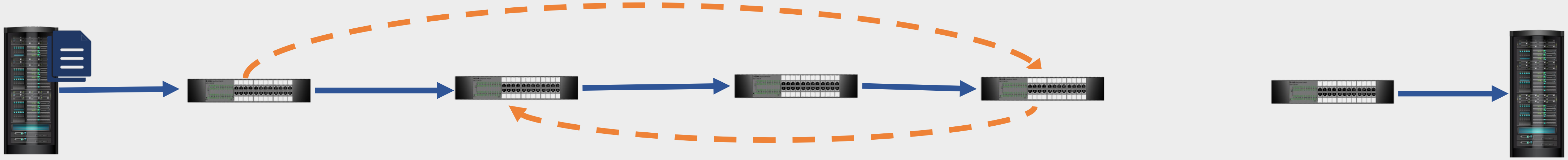


First side effect: transient loops

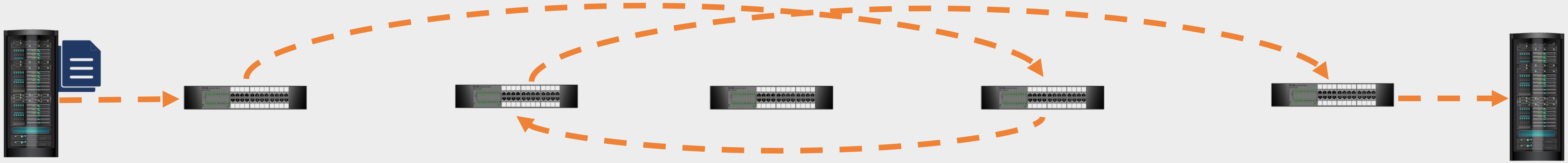
Initial configuration



Possible middle configuration

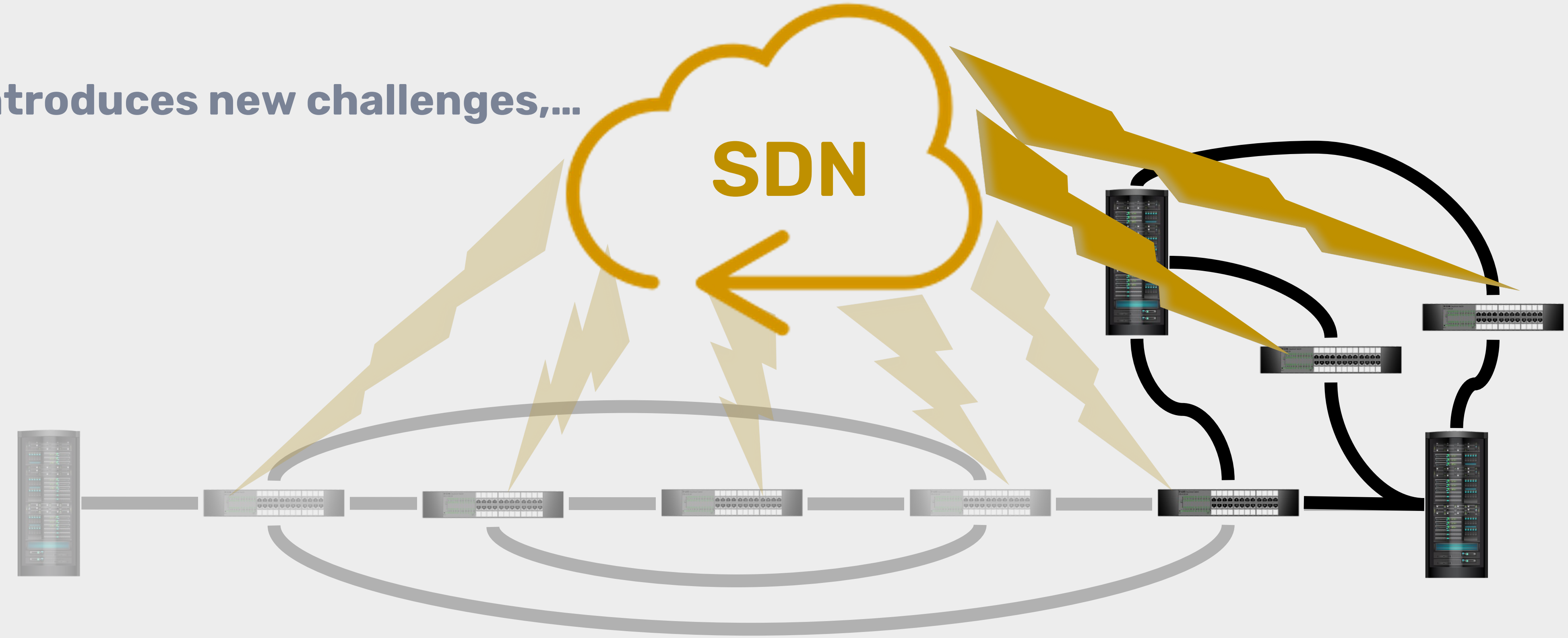


Final configuration

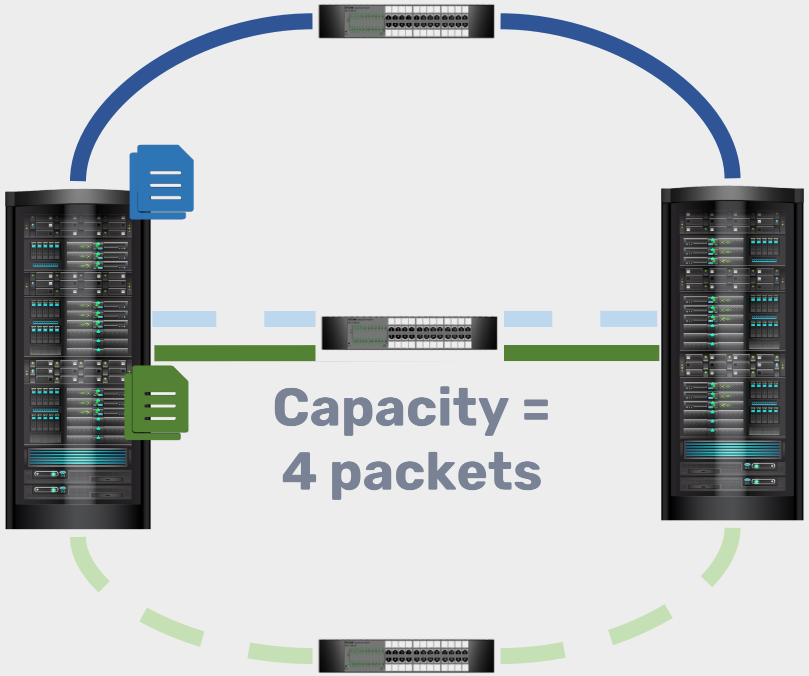


Network updates via SDN

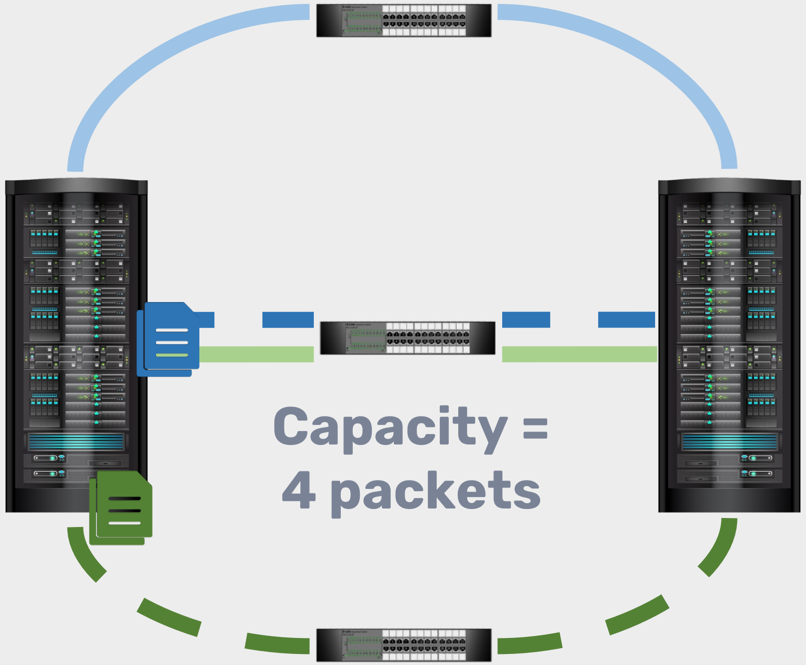
- Networks are prone to be more dynamic
- SDN simplifies and allows for fast updates
- However, SDN introduces new challenges,…



Second side effect: congestion

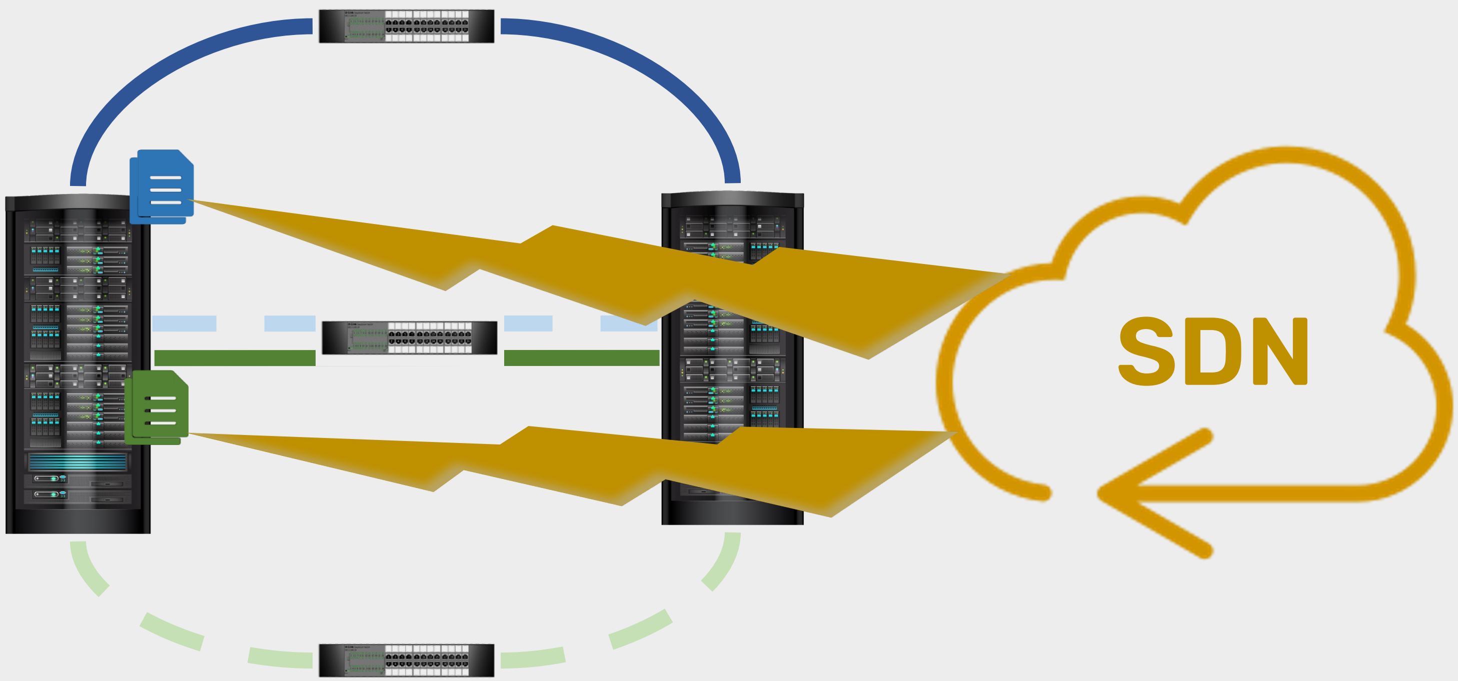


Initial configuration

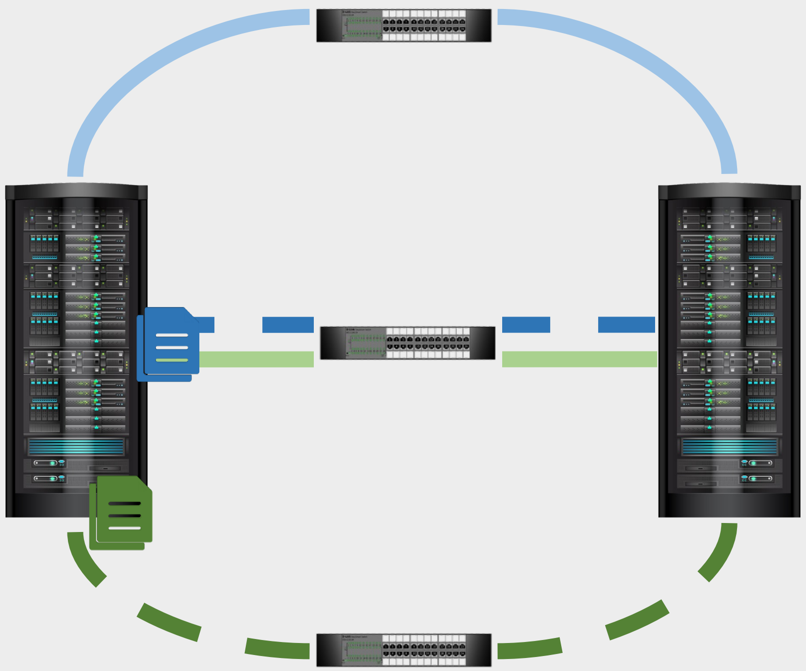


Final configuration

Second side effect: congestion

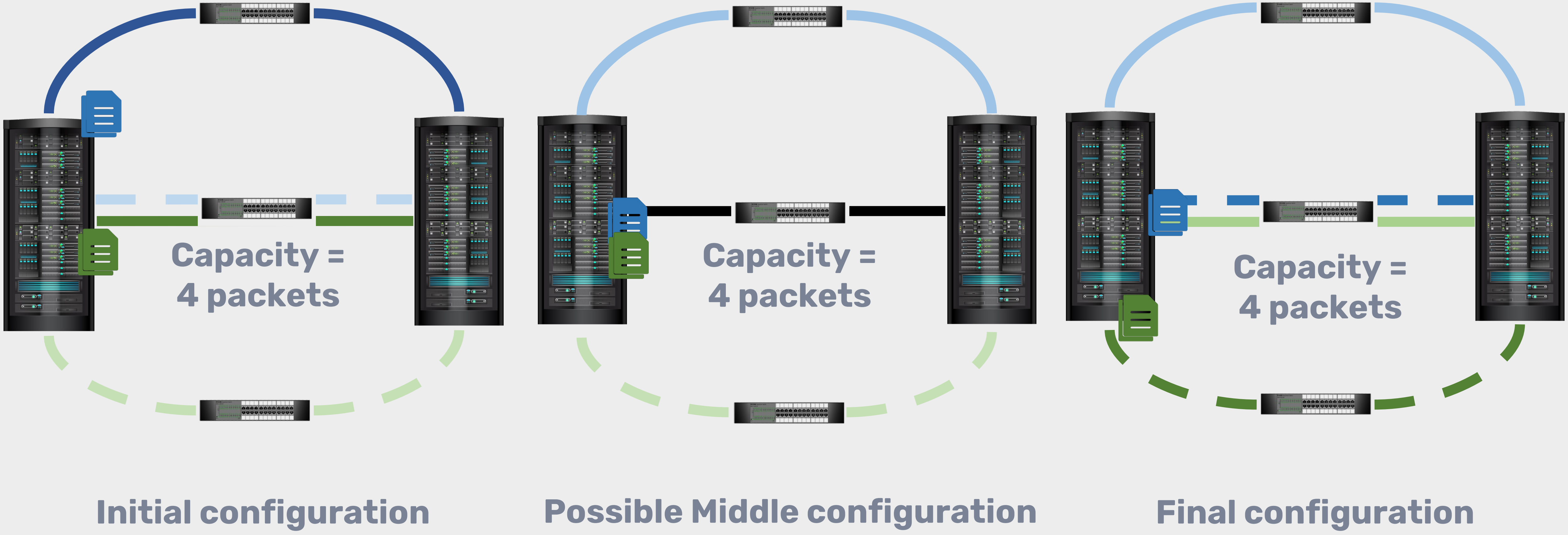


Initial configuration



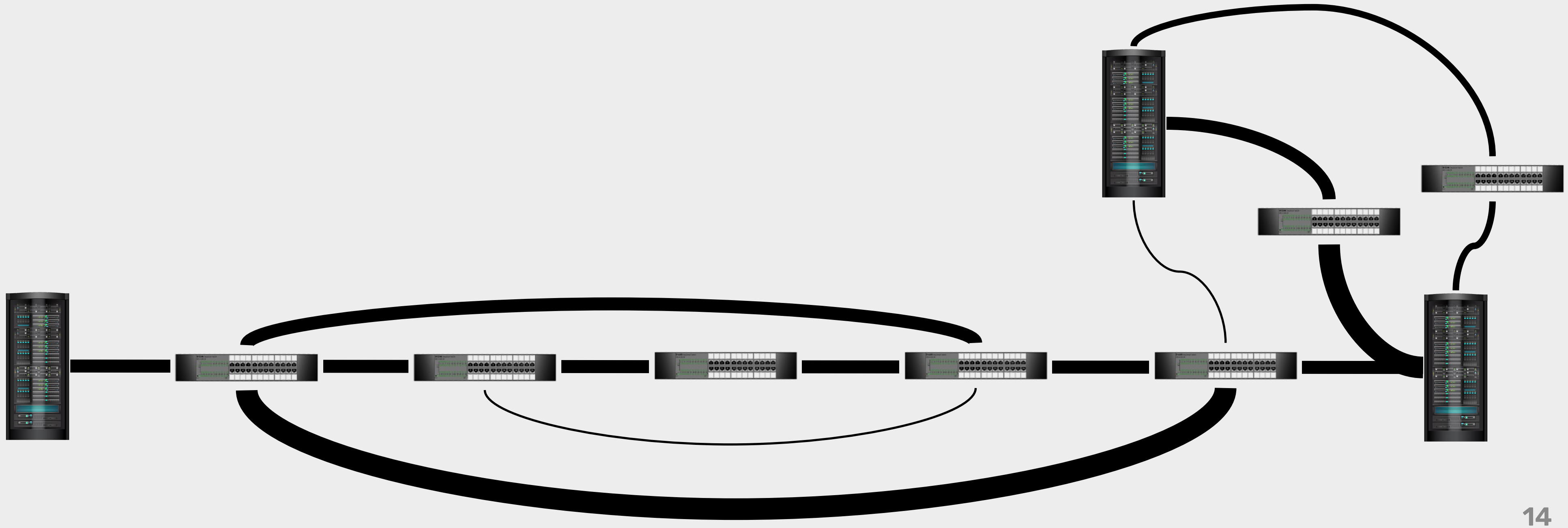
Final configuration

Second side effect: congestion



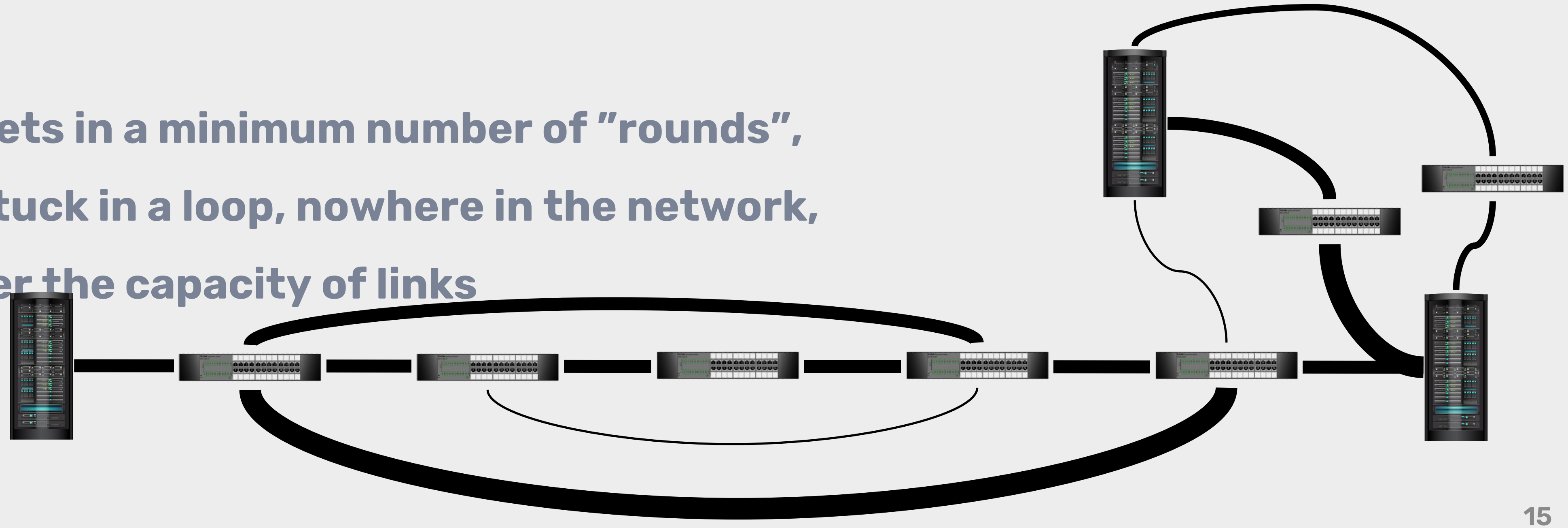
Problem definition

- **Input:** given a network with:
  - multiple unsplittable flows with different demands from different sources and terminals
  - different capacity on each link
  - unknown update delays on each switch

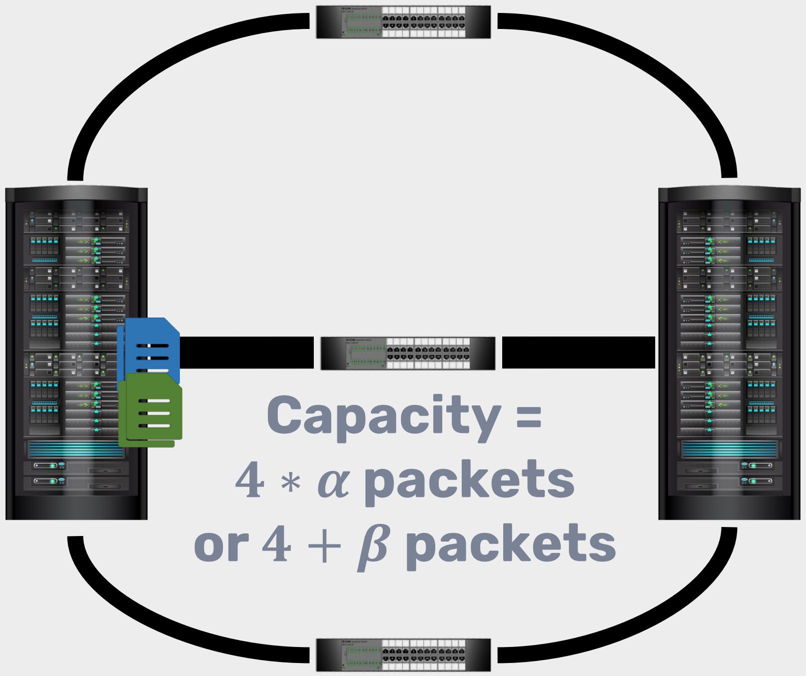


Problem definition

- **Input:** given a network with:
  - multiple unsplittable flows with different demands from different sources and terminals
  - different capacity on each link
  - unknown update delays on each switch
- **Goal:**
  - routing packets in a minimum number of ”rounds”,
  - no packets stuck in a loop, nowhere in the network,
  - not going over the capacity of links

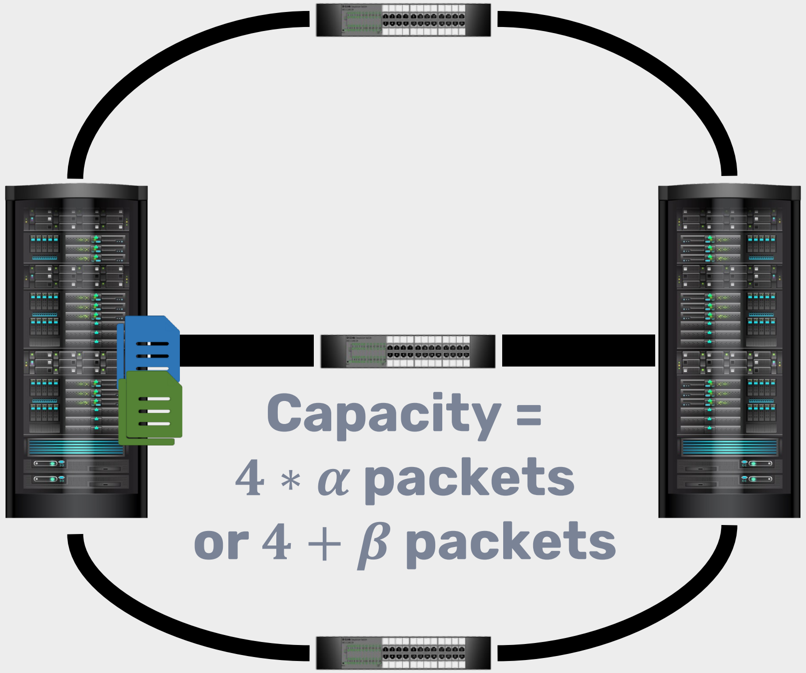


Our proposed Solution: Augmentation



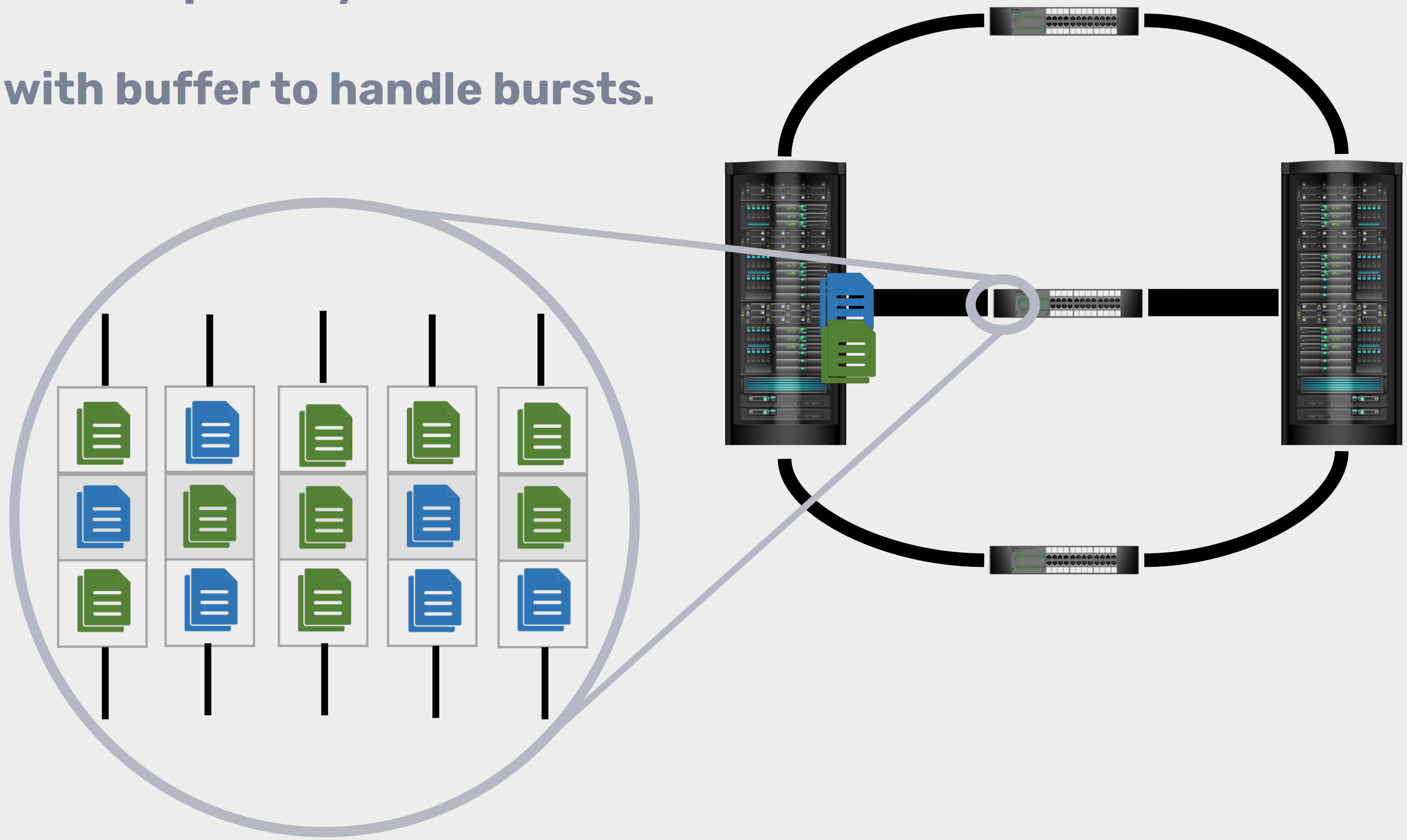
How to realize augmentation?

- **Augmentations are needed temporarily.**



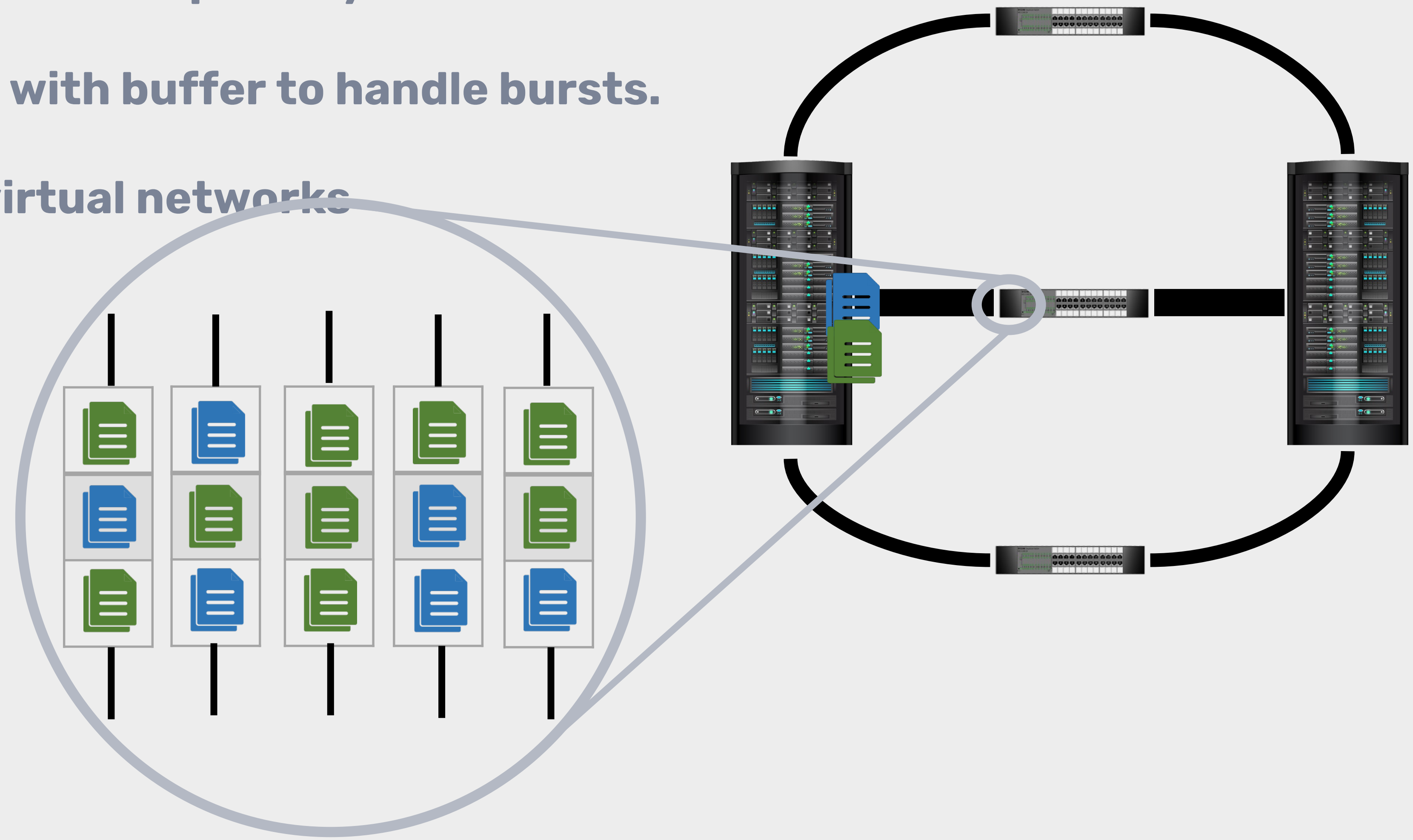
How to realize augmentation?

- Augmentations are needed temporarily.
- Networks are equipped with buffer to handle bursts.

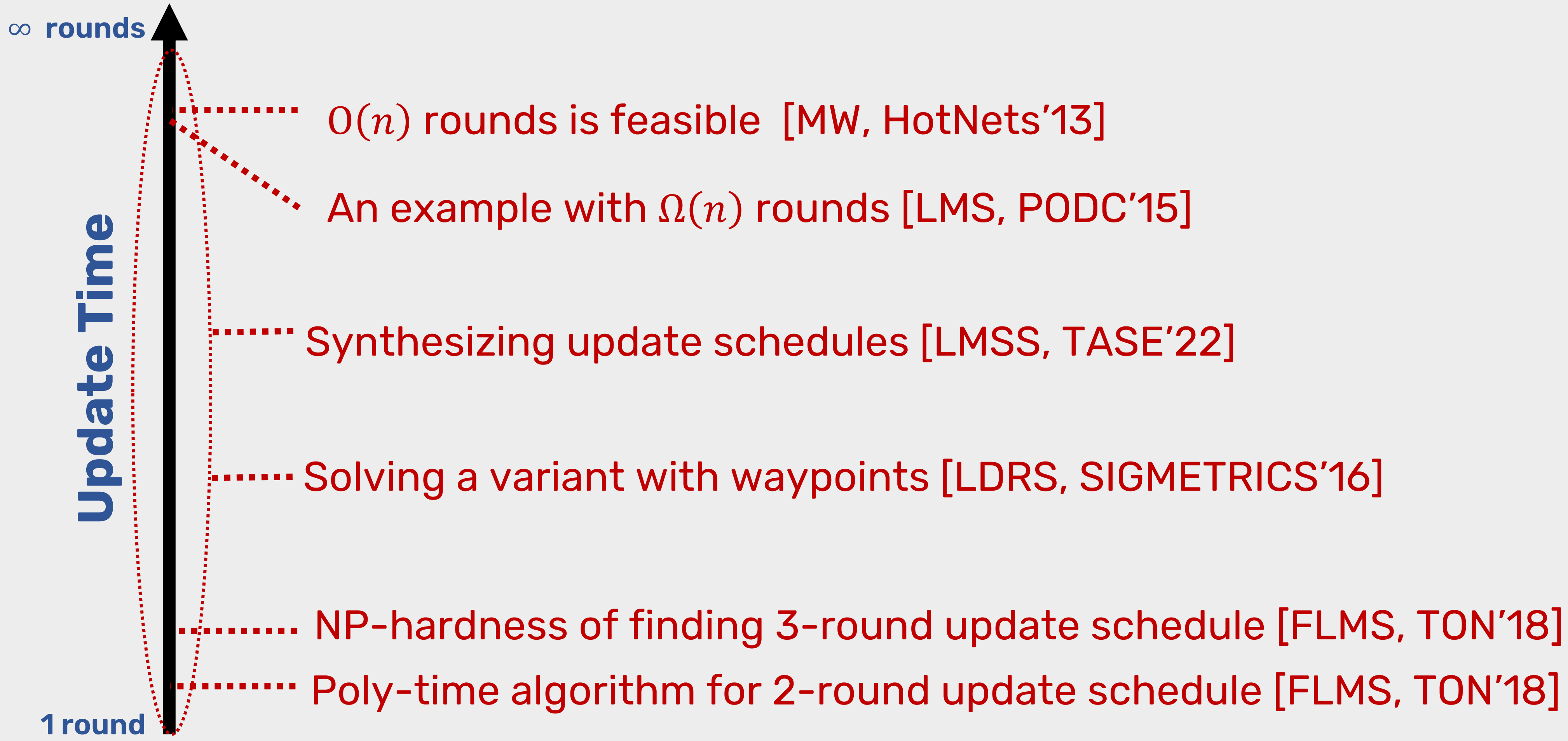


How to realize augmentation?

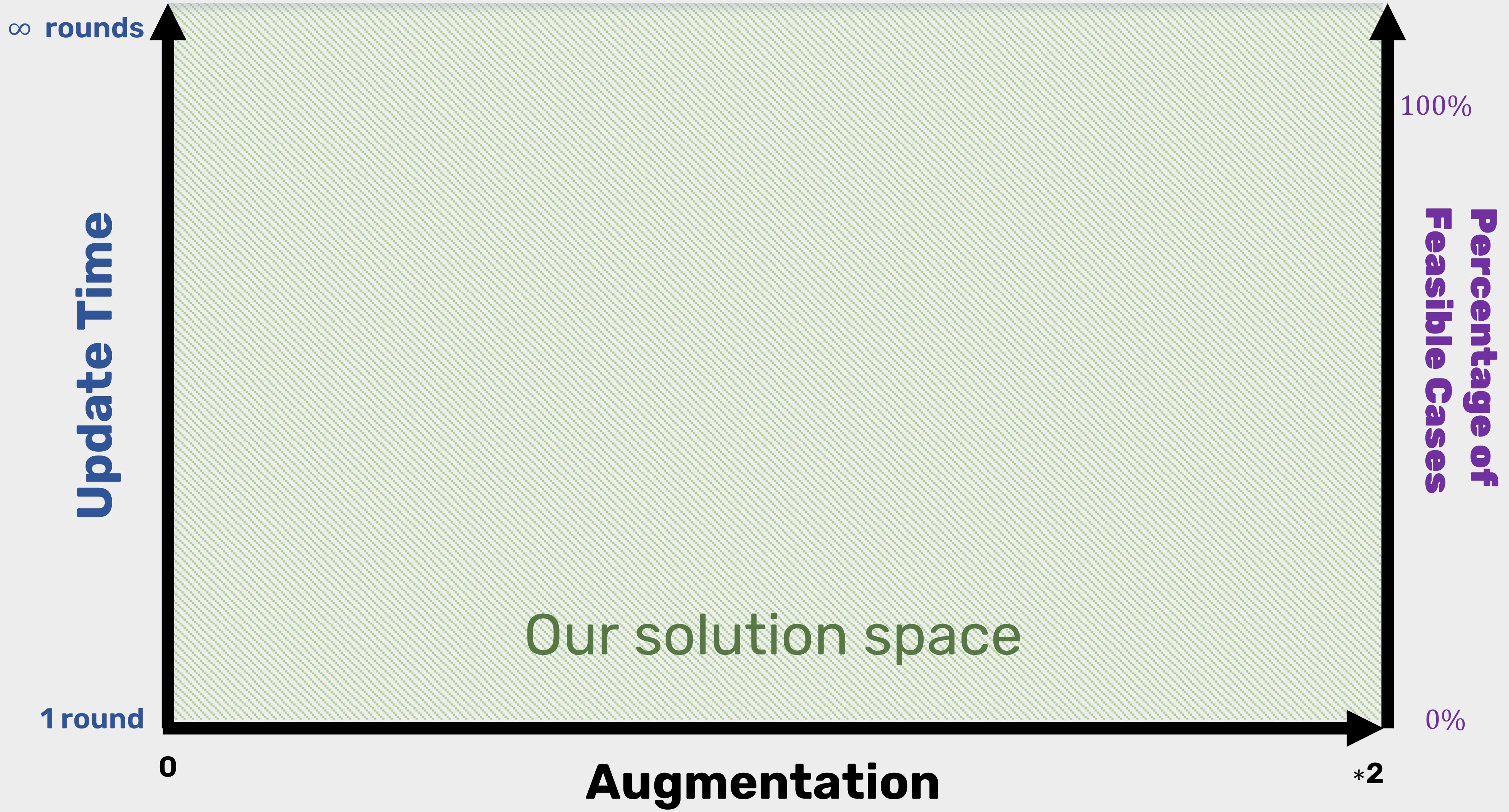
- Augmentations are needed temporarily.
- Networks are equipped with buffer to handle bursts.
- Congestion control in virtual networks



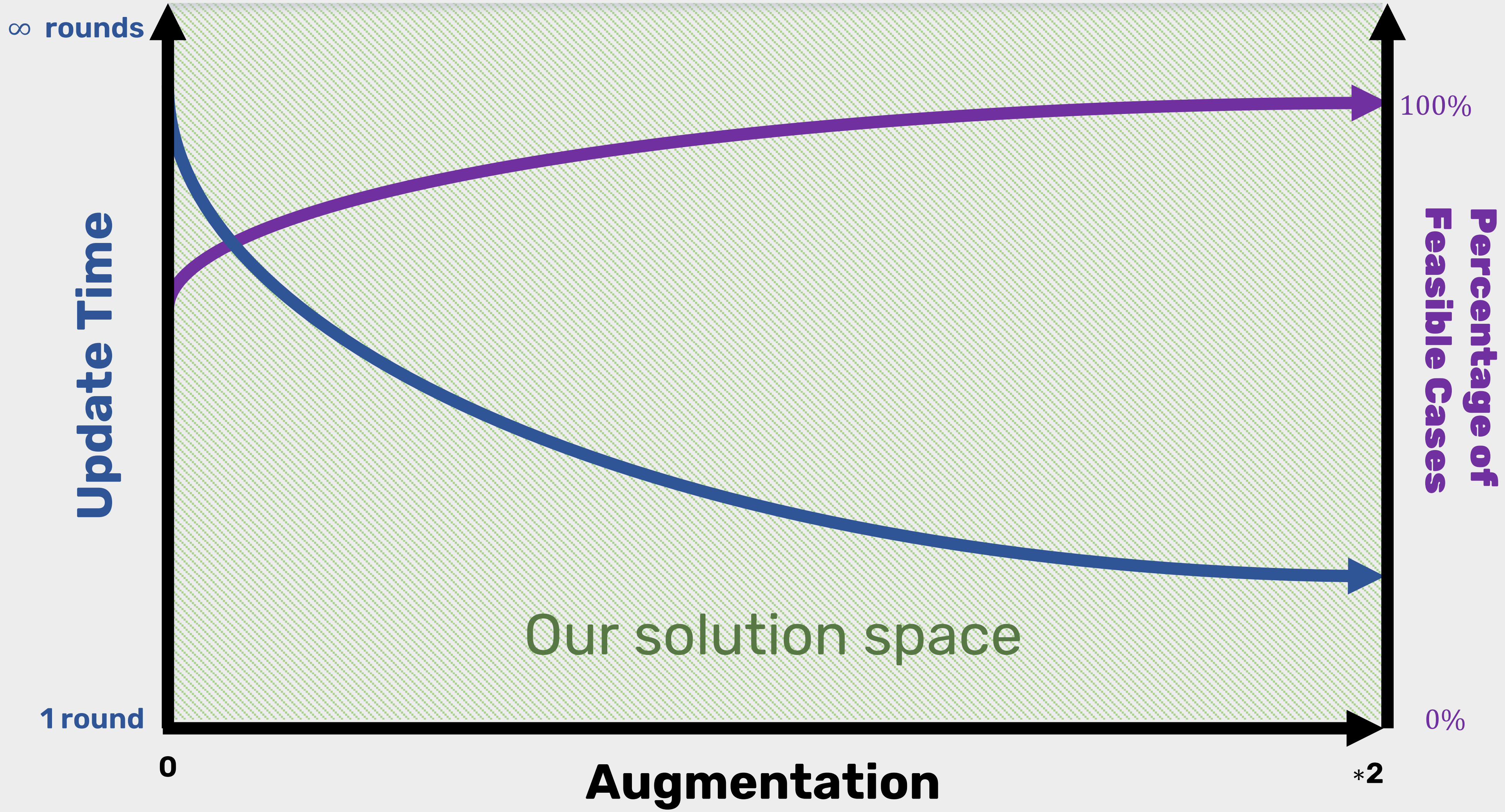
Selected previous works



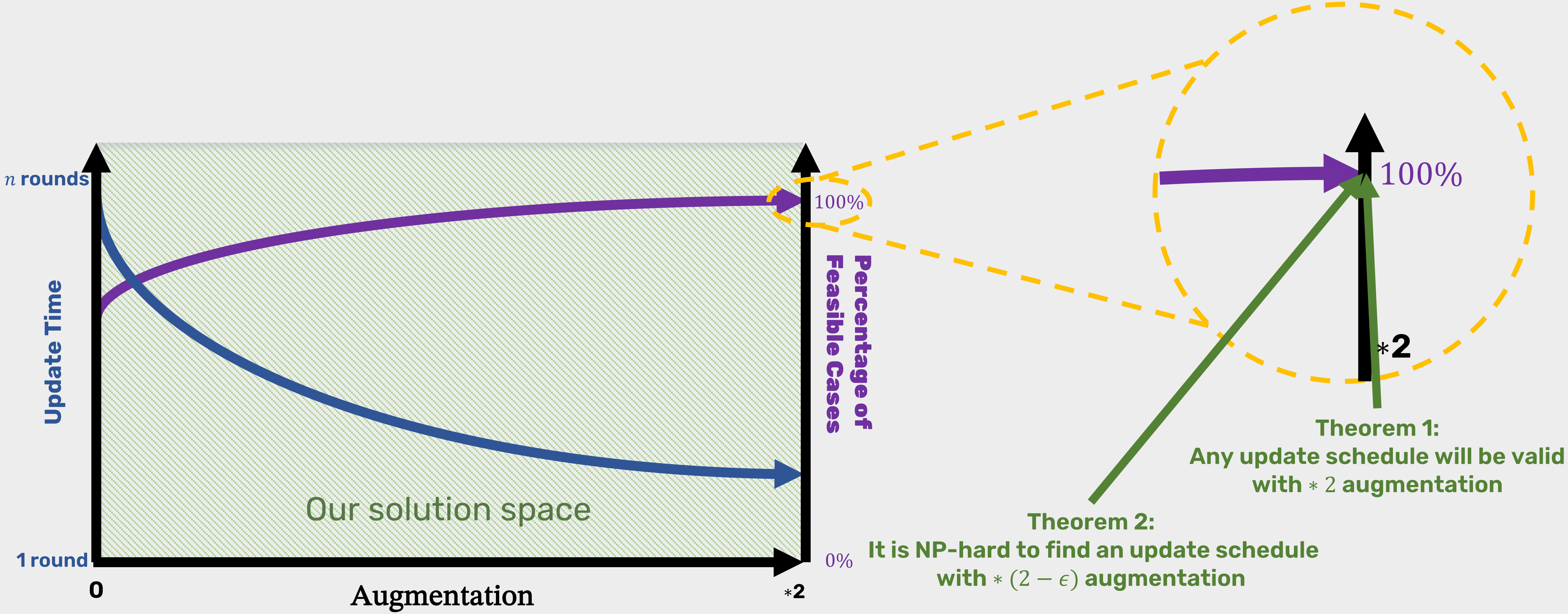
Our contribution: introducing a new dimension



Our contribution: introducing new optimal & feasible schedules



Our contribution: theoretical proofs



NP-Hardness of finding an optimal

A 3SAT Problem

$$C_i = \left(x_j \vee \neg x_{j'} \vee x_{j''}\right)$$

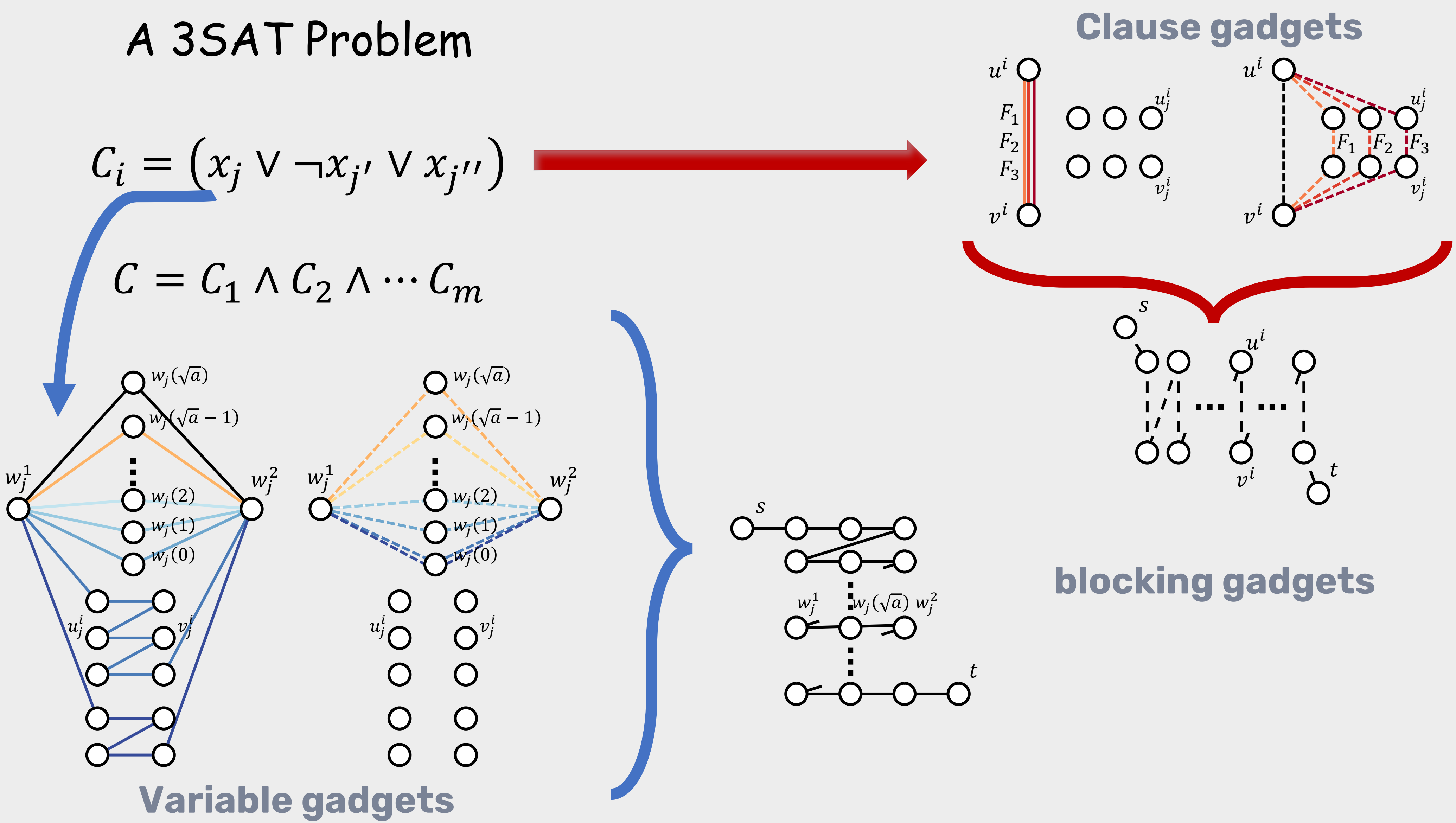
$$C = C_1 \wedge C_2 \wedge \cdots C_m$$

NP-Hardness of finding an optimal

A 3SAT Problem

$C_i = (x_j \vee \neg x_{j'} \vee x_{j''})$

$C = C_1 \wedge C_2 \wedge \cdots C_m$



An optimal solution based on MIP

Minimize R (or α, β)

for all $i \in [|P|]$

$$\textstyle\sum_{r\in[R]} x^r_{v,i} = 1 \qquad \forall v \in V(F^o_i \cup F^u_i) \setminus \{t_i\}$$

$$y^0_{(v,w),i} = 1 \qquad \forall (v,w) \in F^o_i$$

$$y^0_{(v,w),i} = 0 \qquad \forall (v,w) \notin F^o_i$$

for all $r \in [R]$

$$R \geq r \cdot x^r_{v,i} \qquad \forall v \in V(F^o_i \cup F^u_i) \setminus \{t_i\}$$

$$y^r_{(v,w),i} = 1 \qquad \forall (v,w) \in F^o_i \cap F^u_i$$

$$y^r_{(v,w),i} = \textstyle\sum_{r' \leq r} x^{r'}_{v,i} \qquad \forall (v,w) \in F^u_i \setminus F^o_i$$

$$y^r_{(v,w),i} = 1 - \textstyle\sum_{r' \leq r} x^{r'}_{v,i} \qquad \forall (v,w) \in F^o_i \setminus F^u_i$$

for all $\forall (v,w) \in F^o_i \cup F^u_i$

$$Y^r_{(v,w),i} \geq y^{r-1}_{(v,w),i}$$

$$Y^r_{(v,w),i} \geq y^r_{(v,w),i}$$

$$Y^r_{(v,w),i} \leq \frac{o^r_{w,i} - o^r_{v,i} - 1}{|V|-1} + 1$$

for all $\forall v \in P_i$

$$\Lambda^r_{v,i} = x^r_{v,i} \qquad \exists (v,w) \in F^o_i \wedge (v,w') \in F^u_i$$

$$\Lambda^r_{v,i} = 0 \qquad \nexists (v,w) \in F^o_i \wedge (v,w') \in F^u_i$$

$$\Upsilon^r_{v,i} \leq f^r_{(w,v),i}, f^r_{(w',v),i} \qquad \exists (w,v) \in F^o_i \wedge (w',v) \in F^u_i$$

$$\Upsilon^r_{v,i} = 0 \qquad \nexists (w,v) \in F^o_i \wedge (w',v) \in F^u_i$$

$$f^r_{(v,w),i} \leq Y^r_{(v,w),i} \qquad \forall (v,w) \in F^o_i \cup F^u_i$$

$$\textstyle\sum_{(s_i,v)} f^r_{(s_i,v),i} = 1 + \Lambda^r_{s_i,i} \qquad s_i \in P_i$$

$$\textstyle\sum_{(v,t_i)} f^r_{(v,t_i),i} = 1 + \Upsilon^r_{t_i,i} \qquad t_i \in P_i$$

$$\textstyle\sum_{(v,w)} f^r_{(v,w),i} - \sum_{(w',v)} f^r_{(w',v),i} = \Lambda^r_{v,i} - \Upsilon^r_{v,i}$$

$$\forall v \in v \in V(F^o_i \cup F^u_i) \setminus \{s_i, t_i\}$$

$$(v,w), (w',v) \in F^o_i \cup F^u_i$$

$$\textstyle\sum_{i\in[|U|]} f^r_{(v,w),i} \cdot d_i \leq \alpha \cdot c_{(v,w)} + \beta \qquad \forall (v,w) \in E$$

An optimal solution based on MIP: breakdown

Minimize R (or α, β)

for all $i \in [|P|]$

$$\sum_{r \in [R]} x^r_{v,i} = 1 \qquad \forall v \in V(F_i^o \cup F_i^u) \setminus \{t_i\}$$
$$y^0_{(v,w),i} = 1 \qquad \forall (v,w) \in F_i^o$$
$$y^0_{(v,w),i} = 0 \qquad \forall (v,w) \notin F_i^o$$

for all $r \in [R]$

$$R \geq r \cdot x^r_{v,i} \qquad \forall v \in V(F_i^o \cup F_i^u) \setminus \{t_i\}$$
$$y^r_{(v,w),i} = 1 \qquad \forall (v,w) \in F_i^o \cap F_i^u$$
$$y^r_{(v,w),i} = \sum_{r' \leq r} x^{r'}_{v,i} \qquad \forall (v,w) \in F_i^u \setminus F_i^o$$
$$y^r_{(v,w),i} = 1 - \sum_{r' \leq r} x^{r'}_{v,i} \qquad \forall (v,w) \in F_i^o \setminus F_i^u$$

Loop-freedom

for all $\forall (v,w) \in F_i^o \cup F_i^u$

$$Y^r_{(v,w),i} \geq y^{r-1}_{(v,w),i}$$
$$Y^r_{(v,w),i} \leq \frac{o^r_{w,i} - o^r_{v,i} - 1}{|V|-1} + 1$$

Split-avoidance

for all $\forall v \in P_i$

$$\Lambda^r_{v,i} = x^r_{v,i} \qquad \exists (v,w) \in F_i^o \wedge (v,w') \in F_i^u$$
$$\Lambda^r_{v,i} = 0 \qquad \nexists (v,w) \in F_i^o \wedge (v,w') \in F_i^u$$
$$\Upsilon^r_{v,i} = 0 \qquad \nexists (w,v) \in F_i^o \wedge (w',v) \in F_i^u$$
$$f^r_{(v,w),i} \leq Y^r_{(v,w),i} \qquad \forall (v,w) \in F_i^o \cup F_i^u$$

Congestion-freedom

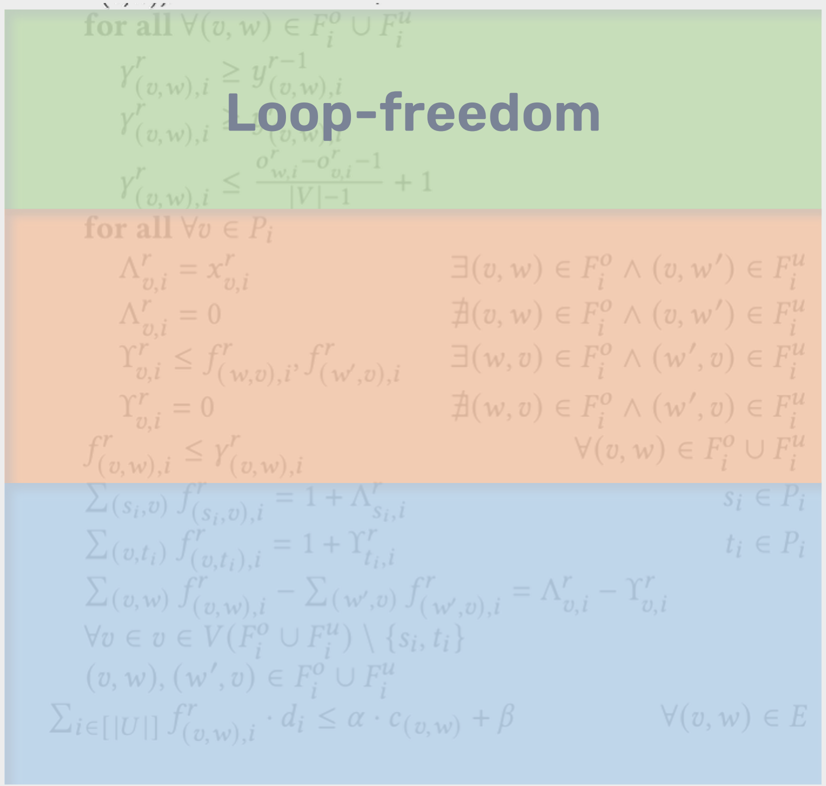
$$\sum_{(s_i,v)} f^r_{(s_i,v),i} = 1 + \Lambda^r_{s_i,i} \qquad s_i \in P_i$$
$$\sum_{(v,t_i)} f^r_{(v,t_i),i} = 1 + \Upsilon^r_{t_i,i} \qquad t_i \in P_i$$
$$\sum_{(v,w)} f^r_{(v,w),i} - \sum_{(w',v)} f^r_{(w',v),i} = \Lambda^r_{v,i} - \Upsilon^r_{v,i}$$
$$(v,w),(w',v) \in F_i^o \cup F_i^u$$
$$\sum_{i \in [|U|]} f^r_{(v,w),i} \cdot d_i \leq \alpha \cdot c_{(v,w)} + \beta \qquad \forall (v,w) \in E$$

An optimal solution based on MIP: key insights

Miller-Tucker-Zemlin formulation

$$\gamma^r_{(v,w),i} \geq y^{r-1}_{(v,w),i}$$
$$\gamma^r_{(v,w),i} \geq y^{r}_{(v,w),i}$$
$$\gamma^r_{(v,w),i} \leq \frac{o^r_{w,i} - o^r_{v,i} - 1}{|V|-1} + 1$$

Enforces ordering among switches



An optimal solution based on MIP: key insights

Branch and merge points

$$
\begin{aligned}
\Lambda^r_{v,i} &= x^r_{v,i} && \exists(v,w) \in F^o_i \wedge (v,w') \in F^u_i \\
\Lambda^r_{v,i} &= 0 && \nexists(v,w) \in F^o_i \wedge (v,w') \in F^u_i \\
\Upsilon^r_{v,i} &\leq f^r_{(w,v),i}, f^r_{(w',v),i} && \exists(w,v) \in F^o_i \wedge (w',v) \in F^u_i \\
\Upsilon^r_{v,i} &= 0 && \nexists(w,v) \in F^o_i \wedge (w',v) \in F^u_i
\end{aligned}
$$

Enforcing strict source-terminal paths

for all $\forall(v,w) \in F^o_i \cup F^u_i$

$$
\begin{aligned}
\gamma^r_{(v,w),i} &\geq y^{r-1}_{(v,w),i} \\
\gamma^r_{(v,w),i} &\geq y^r_{(v,w),i} \\
\gamma^r_{(v,w),i} &\leq \frac{o^r_{w,i} - o^r_{v,i} - 1}{|V|-1} + 1
\end{aligned}
$$

for all $\forall v \in P_i$

$$
\begin{aligned}
\Lambda^r_{v,i} &= x^r_{v,i} && \exists(v,w) \in F^o_i \wedge (v,w') \in F^u_i \\
\Lambda^r_{v,i} &= 0 && \nexists(v,w) \in F^o_i \wedge (v,w') \in F^u_i \\
\Upsilon^r_{v,i} &\leq f^r_{(w,v),i}, f^r_{(w',v),i} && \exists(w,v) \in F^o_i \wedge (w',v) \in F^u_i \\
\Upsilon^r_{v,i} &= 0 && \nexists(w,v) \in F^o_i \wedge (w',v) \in F^u_i
\end{aligned}
$$

Split-avoidance

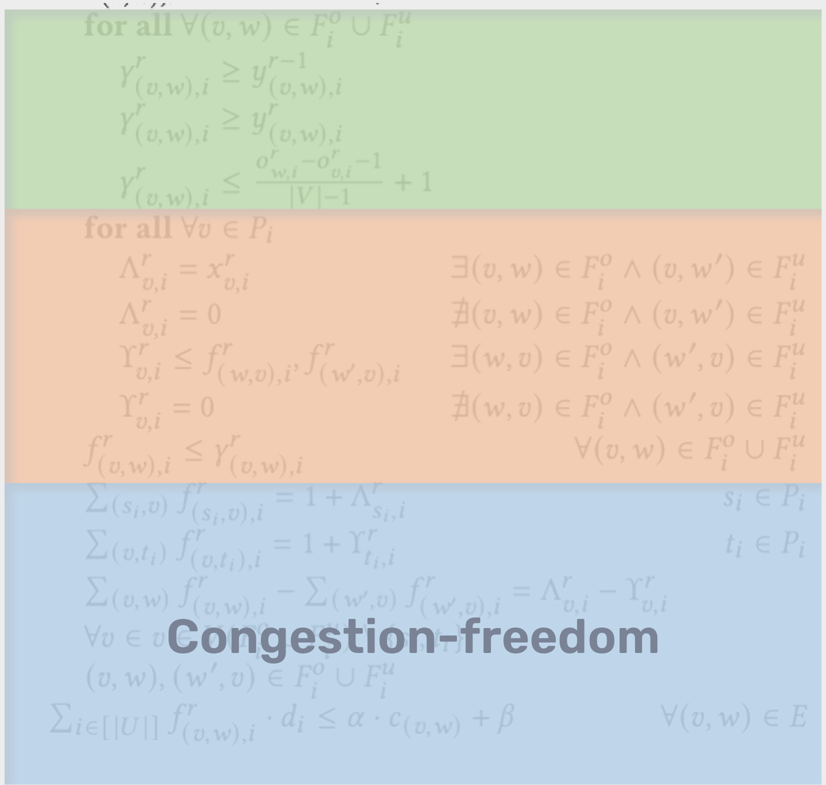
$$
\begin{aligned}
& f^r_{(v,w),i} \leq \gamma^r_{(v,w),i} && \forall(v,w) \in F^o_i \cup F^u_i \\
& \textstyle\sum_{(s_i,v)} f^r_{(s_i,v),i} = 1 + \Lambda^r_{s_i,i} && s_i \in P_i \\
& \textstyle\sum_{(v,t_i)} f^r_{(v,t_i),i} = 1 + \Upsilon^r_{t_i,i} && t_i \in P_i \\
& \textstyle\sum_{(v,w)} f^r_{(v,w),i} - \sum_{(w',v)} f^r_{(w',v),i} = \Lambda^r_{v,i} - \Upsilon^r_{v,i} \\
& \forall v \in v \in V(F^o_i \cup F^u_i) \setminus \{s_i, t_i\} \\
& (v,w), (w',v) \in F^o_i \cup F^u_i \\
& \textstyle\sum_{i \in [|U|]} f^r_{(v,w),i} \cdot d_i \leq \alpha \cdot c_{(v,w)} + \beta && \forall(v,w) \in E
\end{aligned}
$$

An optimal solution based on MIP: key insights

Congestion freedom

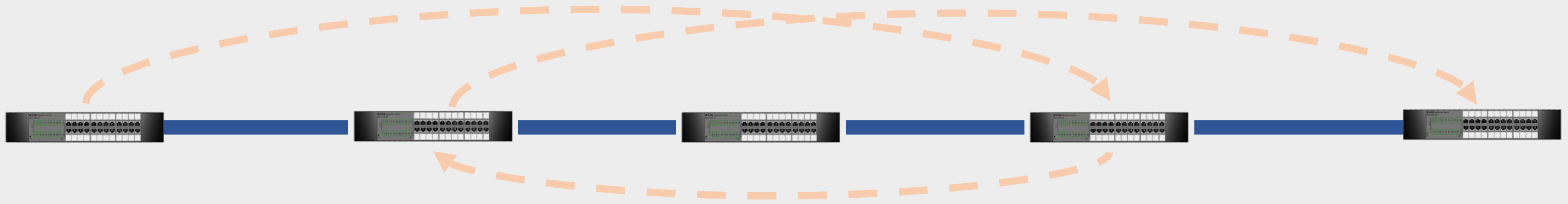
$$
\begin{aligned}
&\textstyle\sum_{(s_i,v)} f^r_{(s_i,v),i} = 1 + \Lambda^r_{s_i,i} && s_i \in P_i \\
&\textstyle\sum_{(v,t_i)} f^r_{(v,t_i),i} = 1 + \Upsilon^r_{t_i,i} && t_i \in P_i \\
&\textstyle\sum_{(v,w)} f^r_{(v,w),i} - \sum_{(w',v)} f^r_{(w',v),i} = \Lambda^r_{v,i} - \Upsilon^r_{v,i} \\
&\forall v \in v \in V(F^o_i \cup F^u_i) \setminus \{s_i, t_i\} \\
&(v,w), (w',v) \in F^o_i \cup F^u_i \\
&\textstyle\sum_{i \in [|U|]} f^r_{(v,w),i} \cdot d_i \le \alpha \cdot c_{(v,w)} + \beta && \forall (v,w) \in E
\end{aligned}
$$

Limiting flows



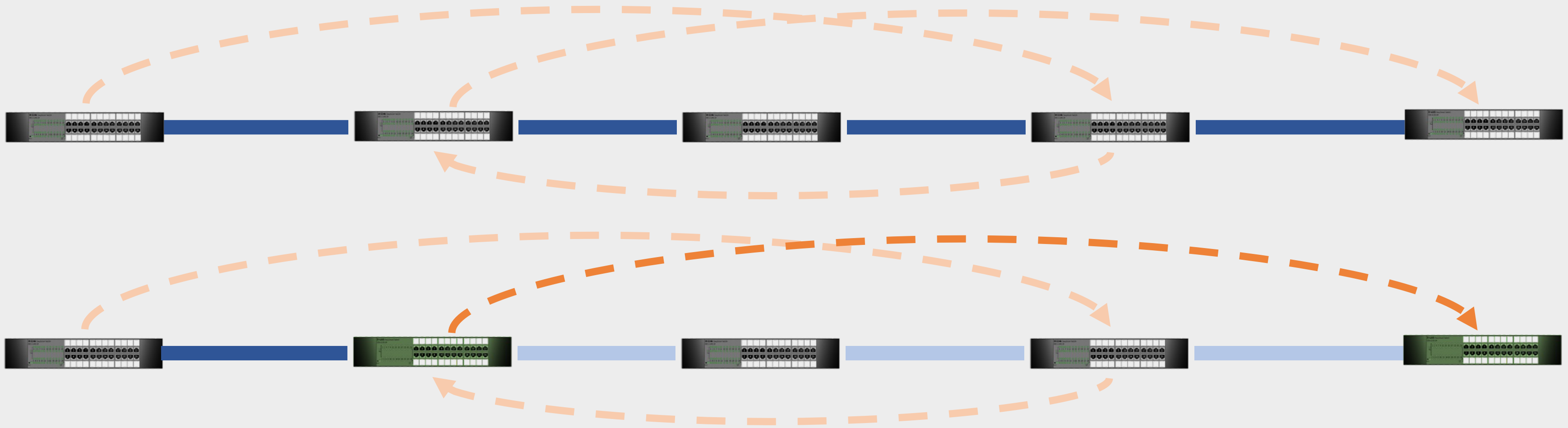
Fast algorithms: Greedy

➢ **Goal:** optimizing the number of rounds



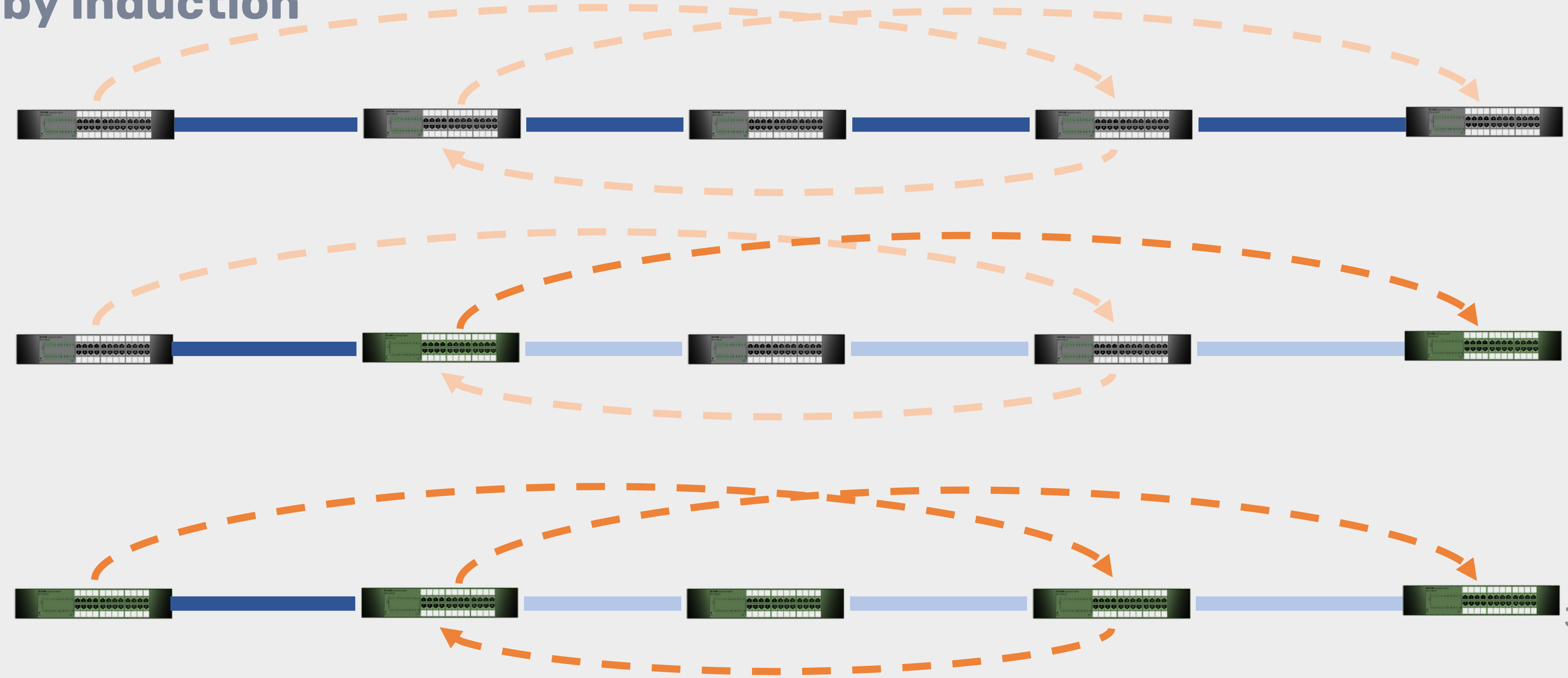
Fast algorithms: Greedy

- **Goal:** optimizing the number of rounds
- **Method:** backward recursions from terminal



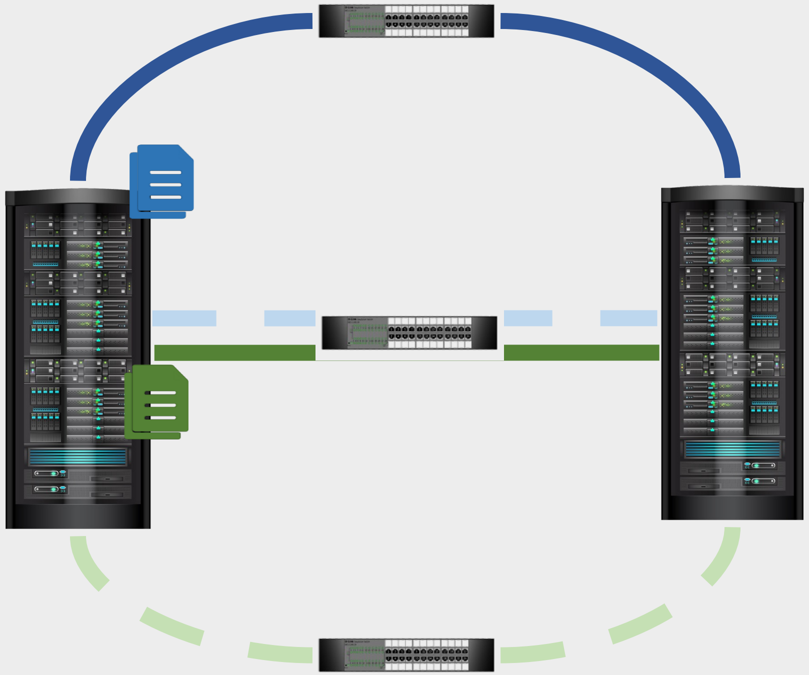
Fast algorithms: Greedy

- **Goal:** optimizing the number of rounds
- **Method:** backward recursions from terminal
- **Proof of termination:** by induction



Fast algorithms: Delay

➢ **Goal:** optimizing congestion



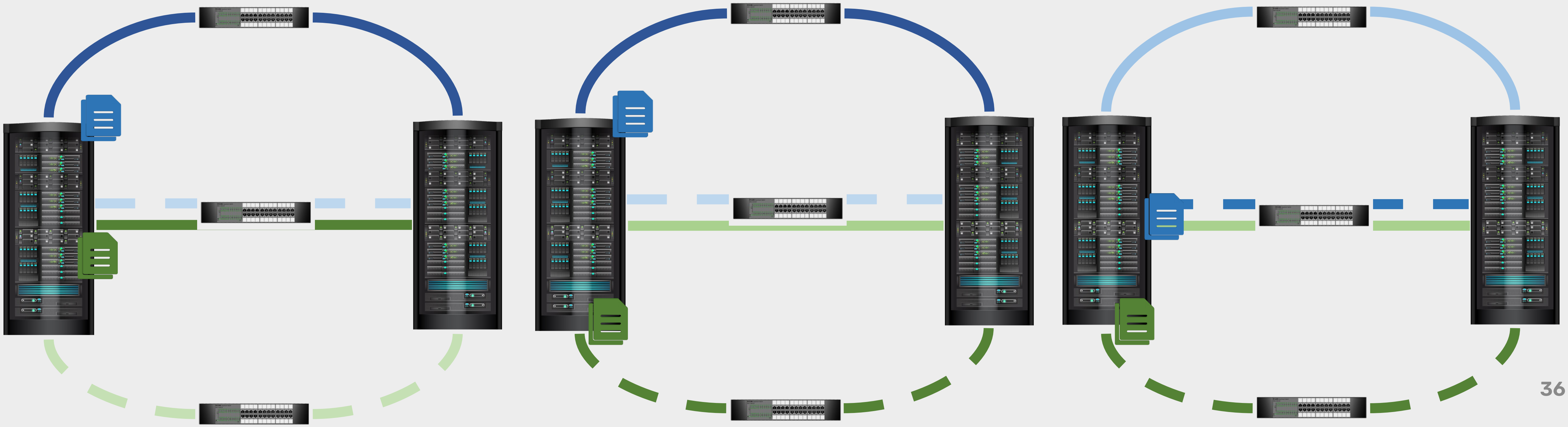
Fast algorithms: Delay

- **Goal:** optimizing congestion

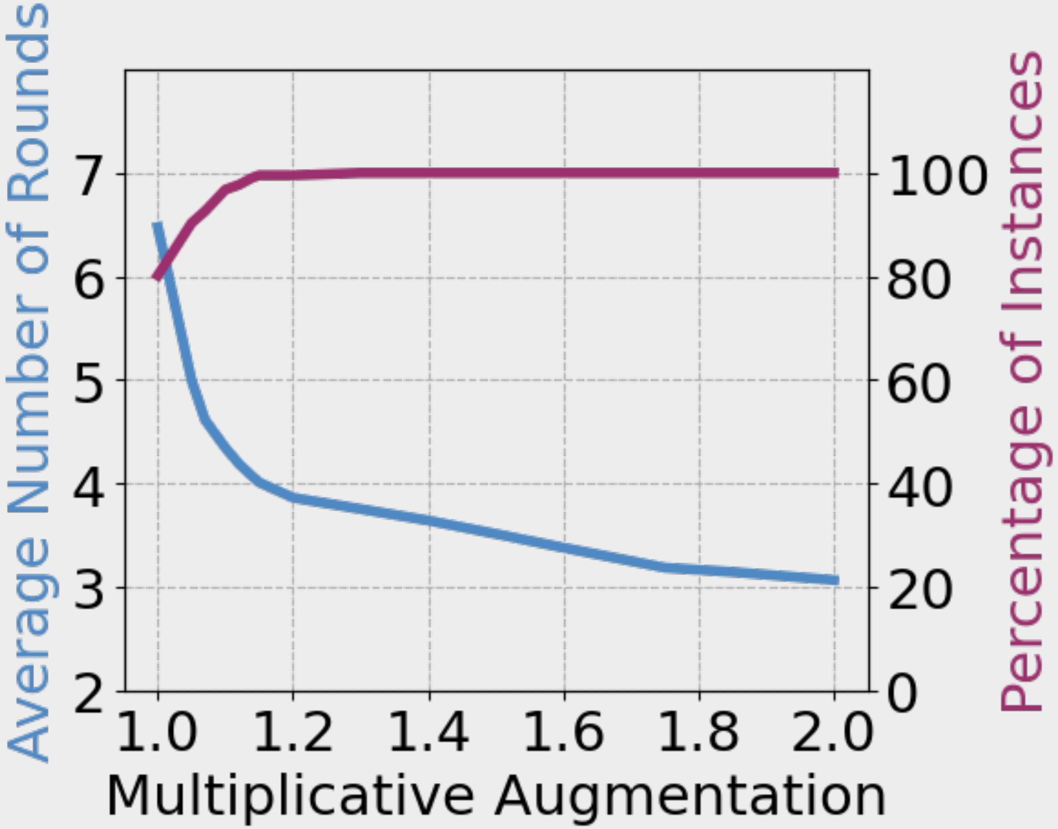
- **Method:** searching for best delayed path

Fast algorithms: Delay

- **Goal:** optimizing congestion
- **Method:** searching for best delayed path
- **Proof of termination:** stops when no changes happen in augmentation



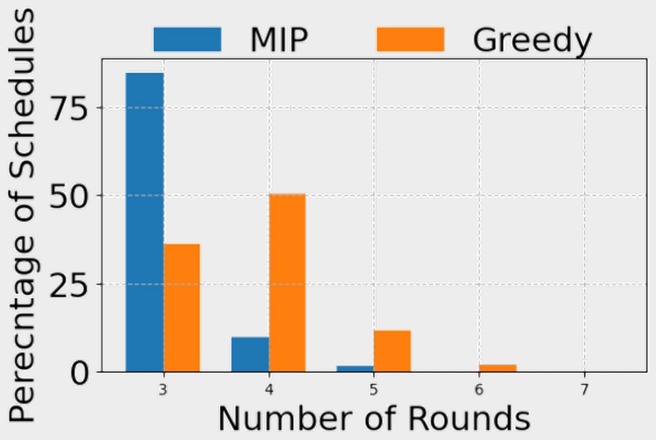
Empirical counter-part of the tradeoff

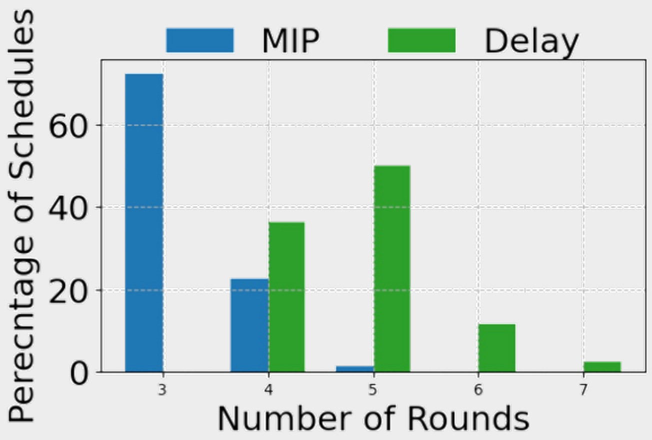


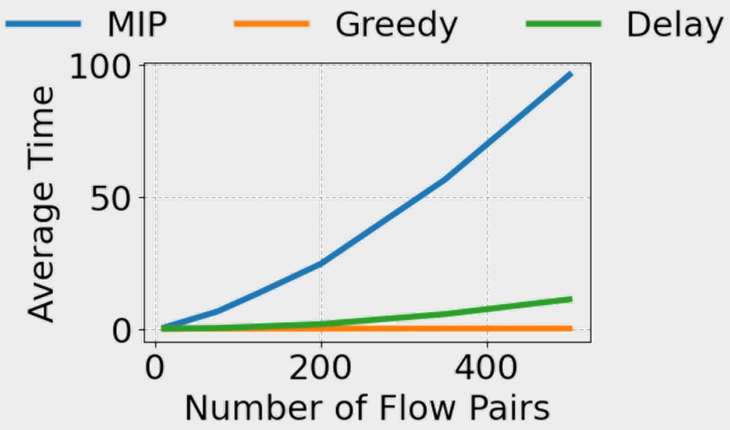
The Internet Topology Zoo

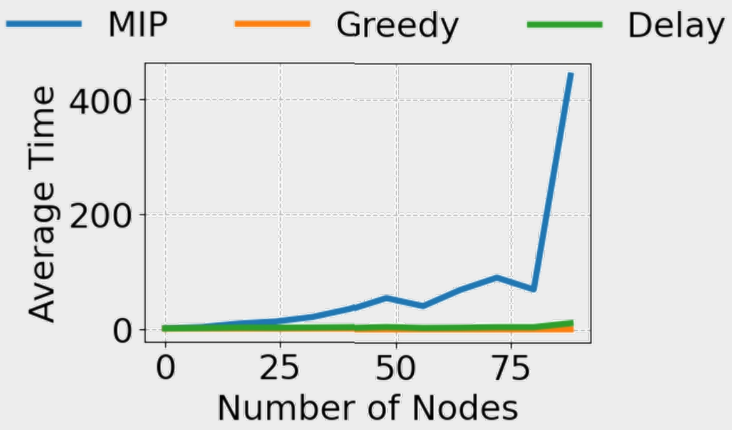
Code is available at github.com/inet-tub/AugmentRoute

MIP vs. Greedy vs. Delay









Summary

- **Concept:** introducing augmentation for consistent updates
- **Theory:**
  - any schedule is consistent with $*2$ augmentation,
  - finding a consistent schedule with $*2-\epsilon$ augmentation is NP-hard
- **Algorithms:**
  - a mixed integer program to find the optimal number of rounds/augmentation
  - fast algorithms minimizing the number of rounds/augmentation
- **Empirical evaluation:** confirming our theories
- **Future work:** Supporting splittable flows or way-pointing

Thank you!

erc
European Research Council
Established by the European Commission

FWF
Der Wissenschaftsfonds.

Bundesministerium
für Bildung
und Forschung