Reliable Real-Time Communication in Time-Sensitive Networking with Static and Dynamic Network Traffic

Lisa Maile

Computer Networks and Communication Systems,
Friedrich-Alexander Universität, Erlangen-Nürnberg, Germany
and
Institute of Computer and Network Engineering,
Technische Universität Braunschweig, Germany

Cyber-Physical Systems (CPS)

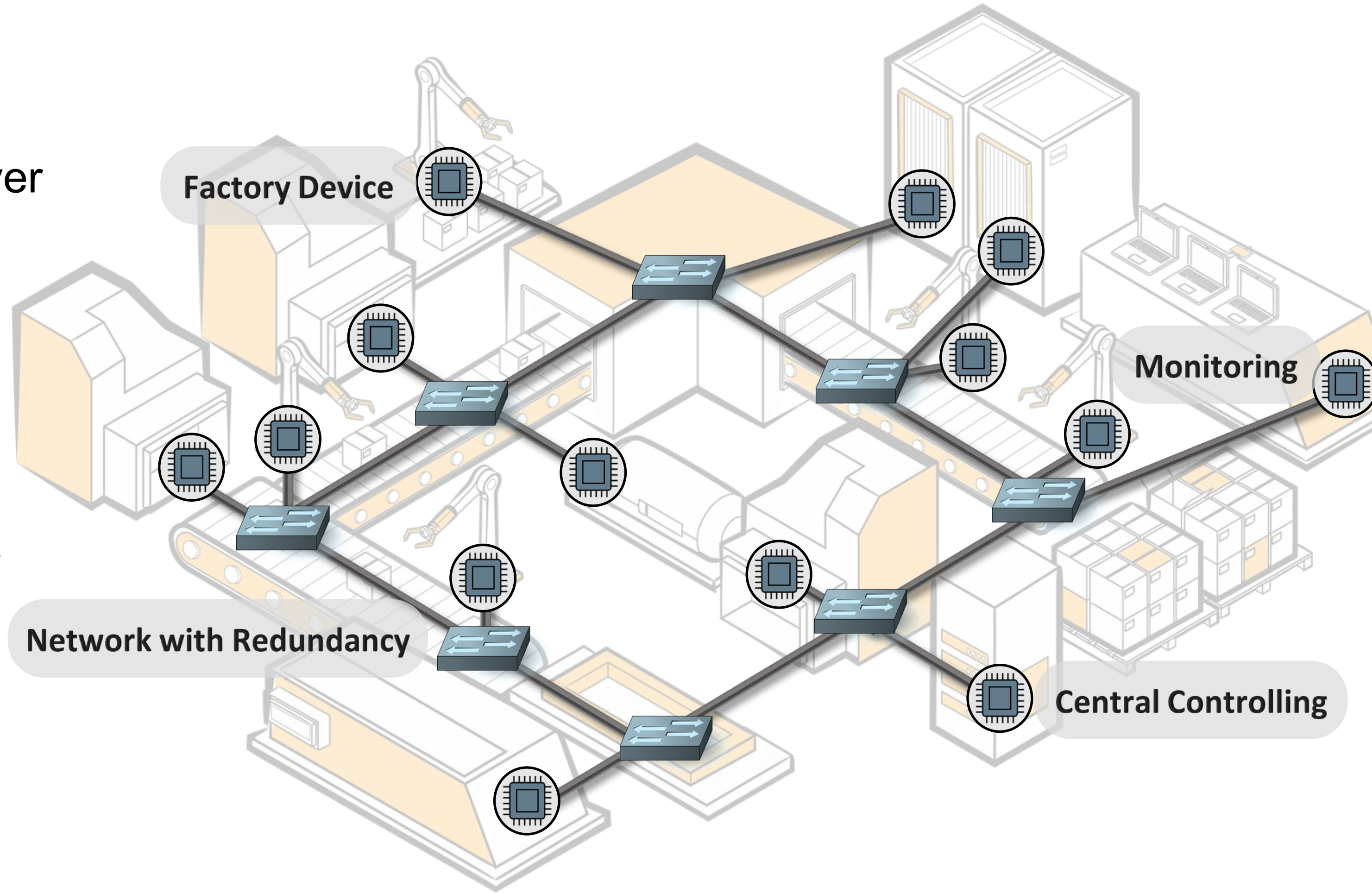
Systems with tight connections of physical & computational resources that communicate over networks.

Dynamic changes:

- switch of production process
- new or updated machinery
- mobile devices, such as laptops and robots
- …

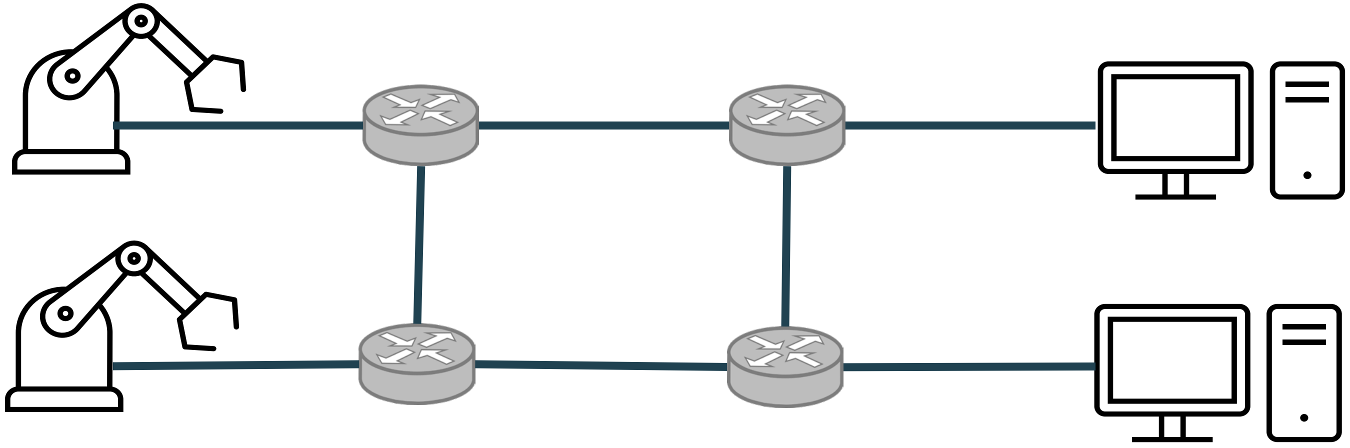
Challenges

Safety-critical & real-time tasks:
bounded delays and no information loss.



Overview

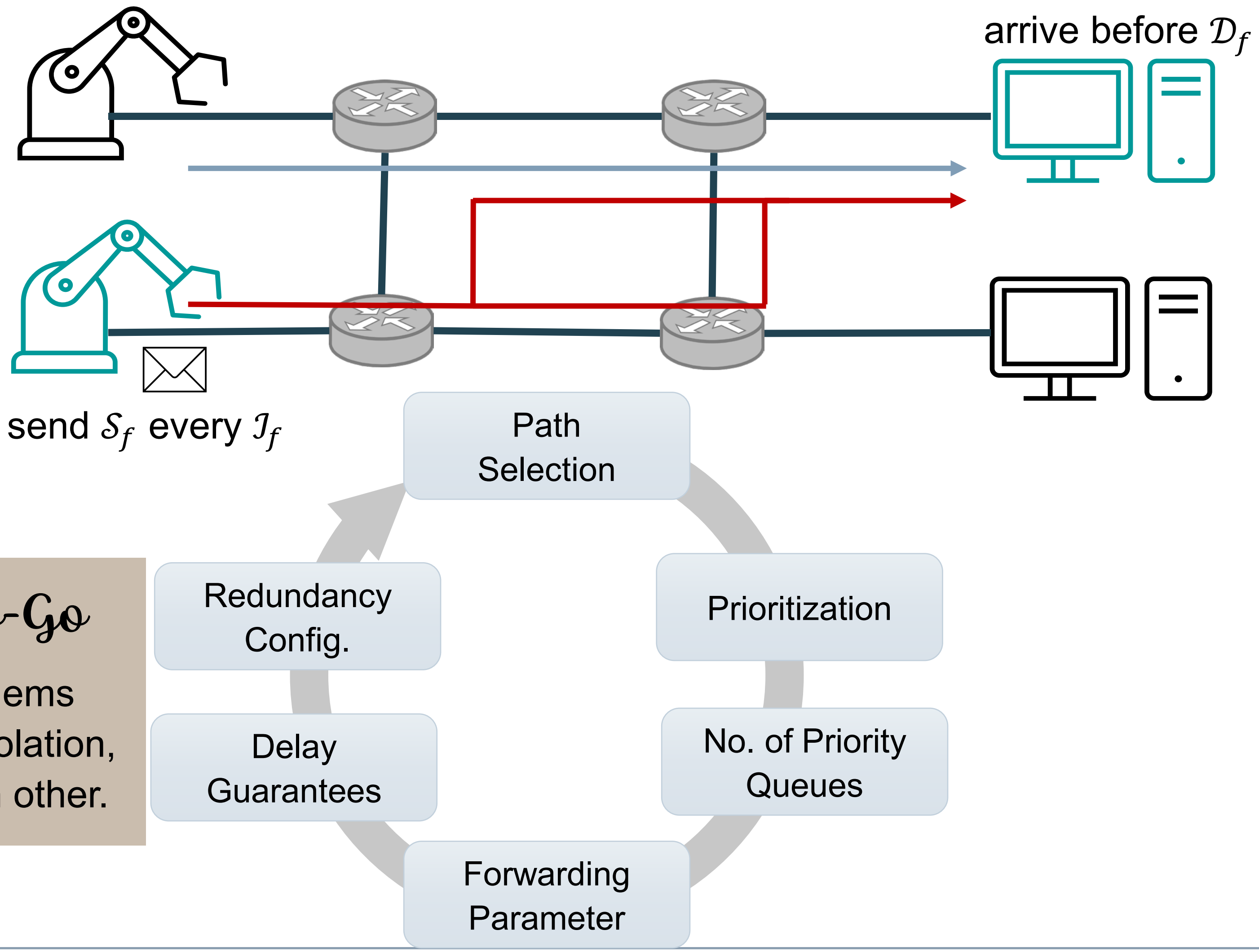
Network Planning



Overview

Network Planning

Flow f
Source
Destination



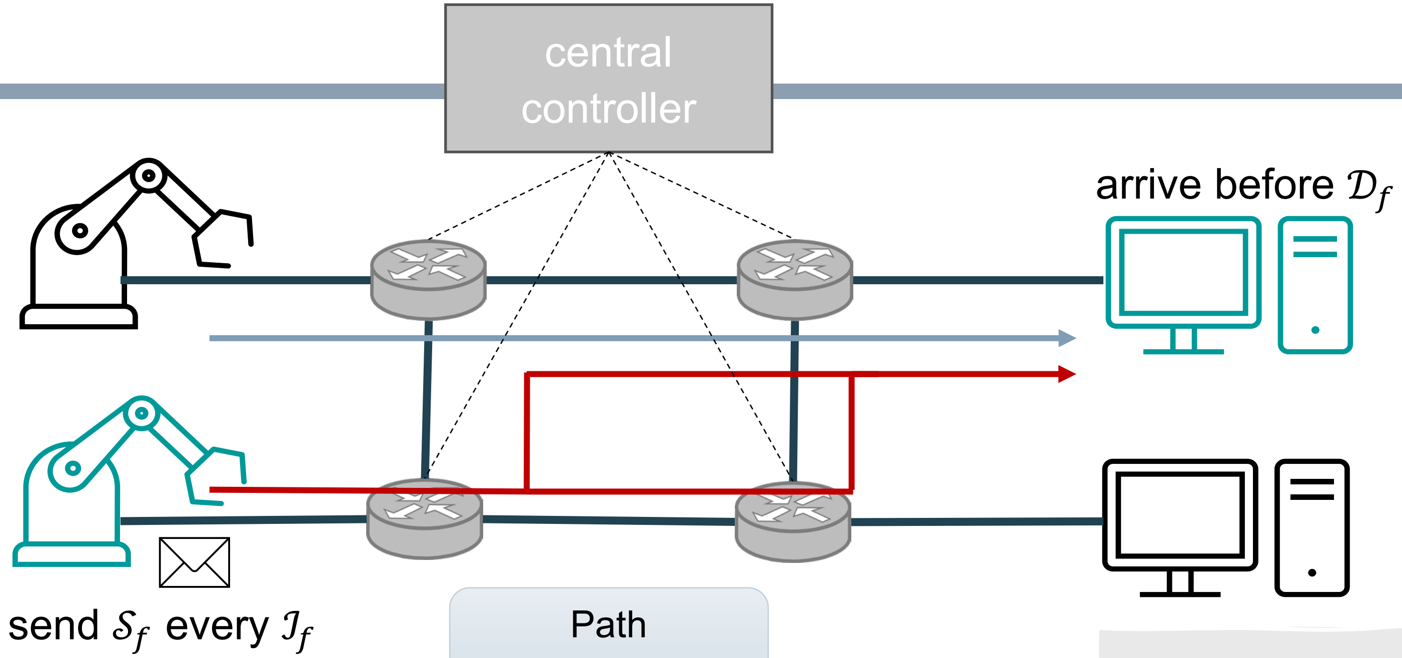
Information To-Go

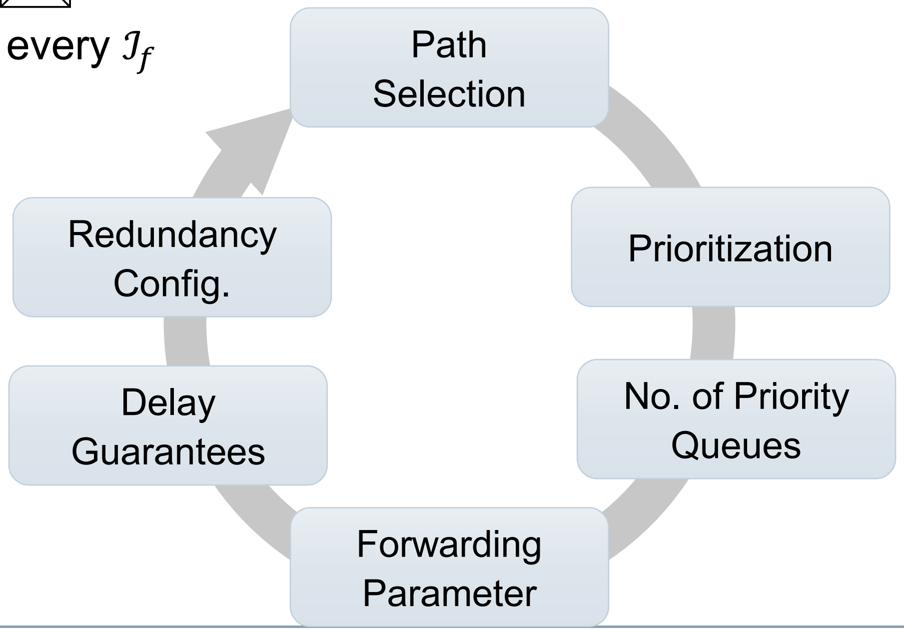
Network design problems have been studied in isolation, but directly affect each other.

Overview

Network Planning

Flow f
Source
Destination
Data Size $\mathcal{S}_f$
Sending Interval $\mathcal{I}_f$
Deadline $\mathcal{D}_f$





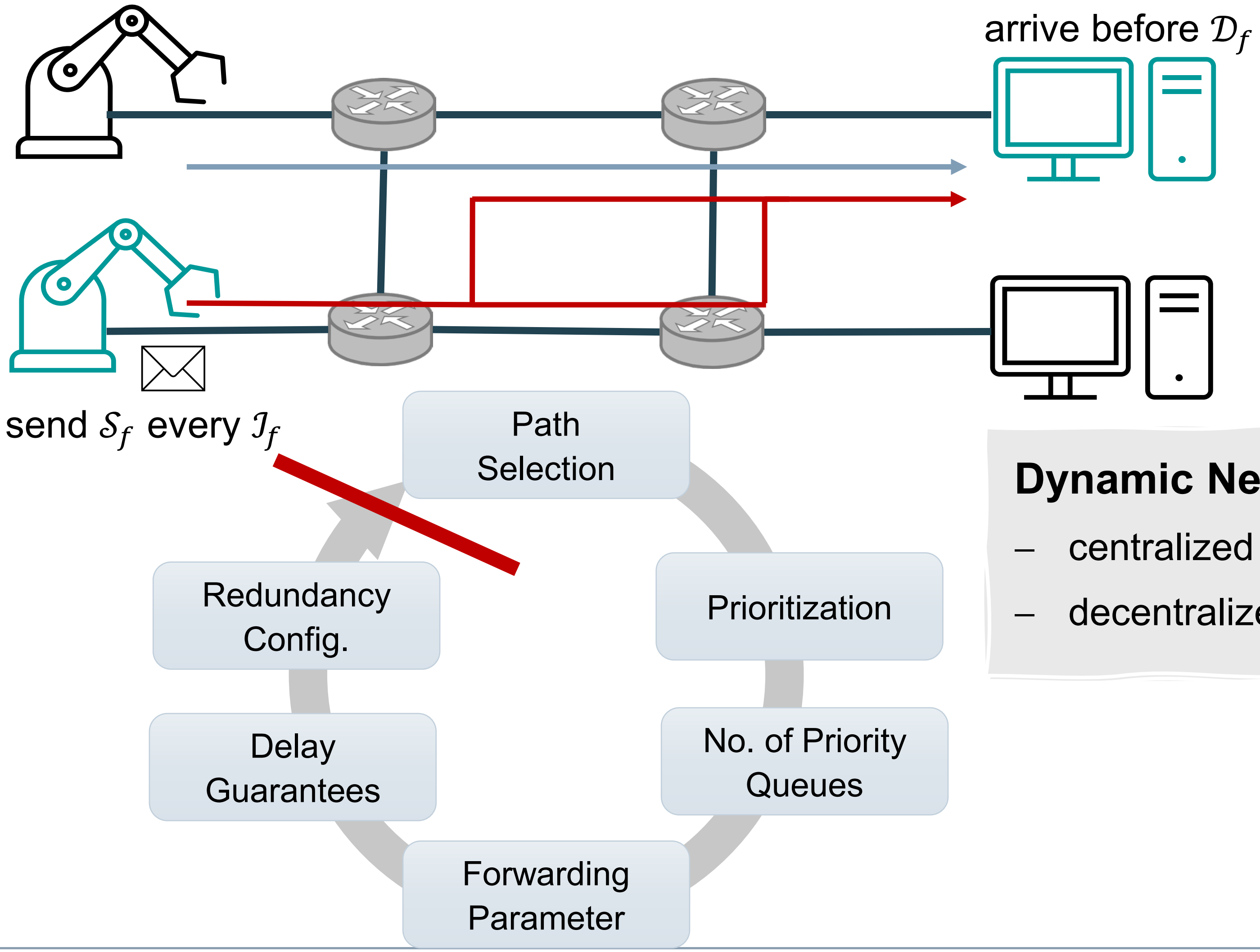
Dynamic Network

- centralized control units
- decentralized reservation protocols

Overview

Network Planning

Flow f
Source
Destination
Data Size $\mathcal{S}_f$
Sending Interval $\mathcal{I}_f$
Deadline $\mathcal{D}_f$



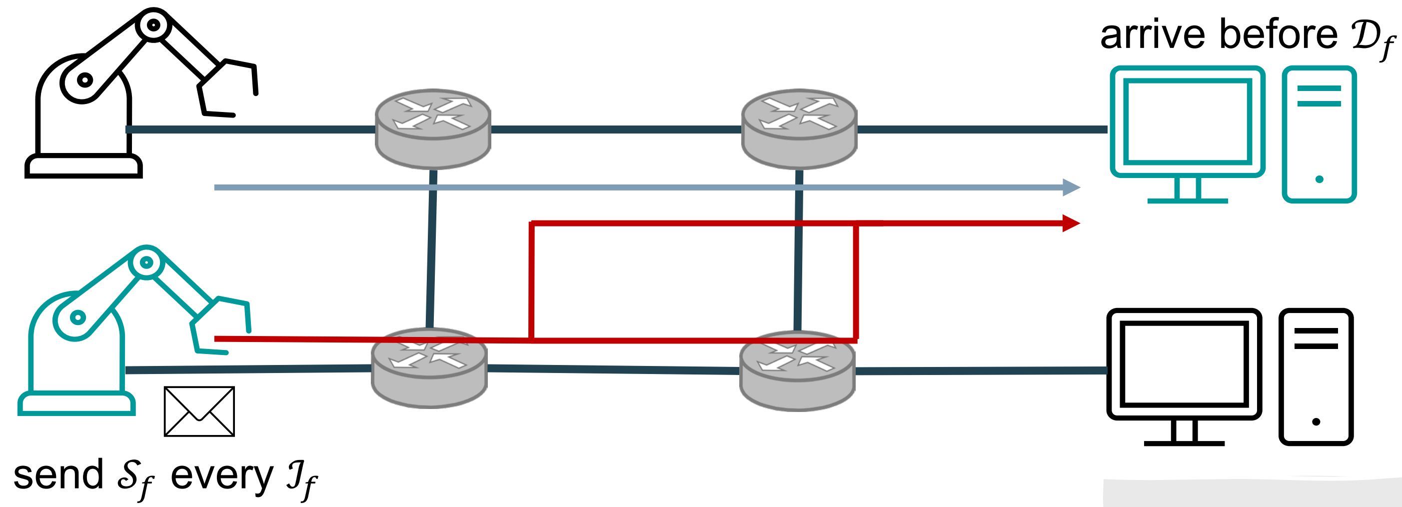
Dynamic Network

- centralized control units
- decentralized reservation protocols

Overview

Network Planning

Flow f
Source
Destination
Data Size $\mathcal{S}_f$
Sending Interval $\mathcal{I}_f$
Deadline $\mathcal{D}_f$



Dynamic Network

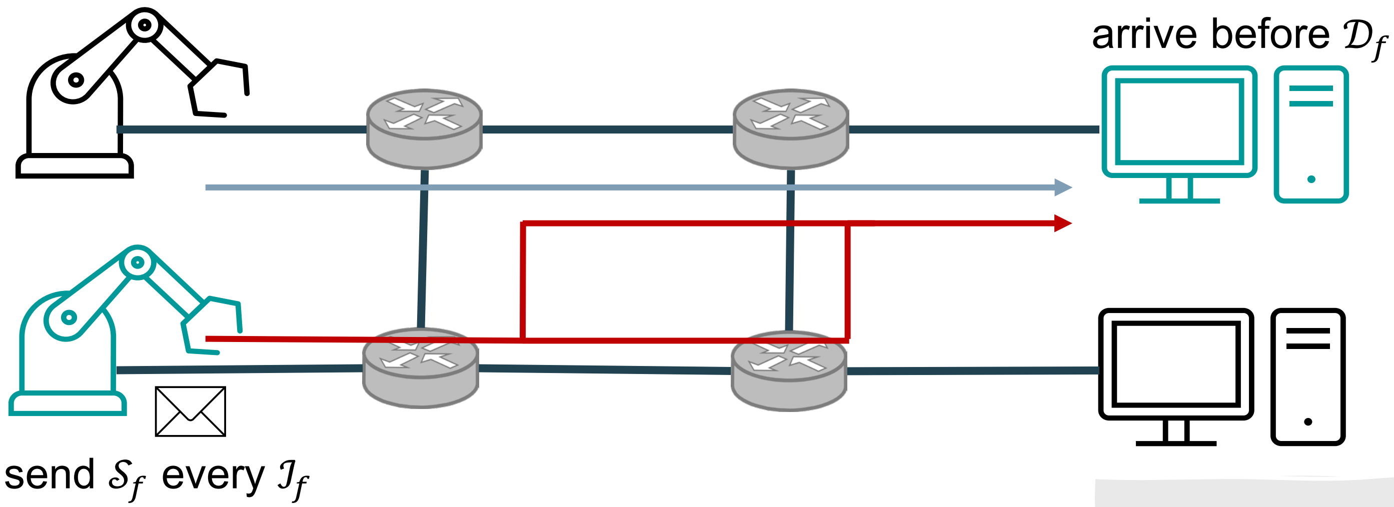
- centralized control units
- decentralized reservation protocols

Delay Guarantees → Forwarding Parameter → No. of Priority Queues → Prioritization → Path Selection → Redundancy Config.

Overview

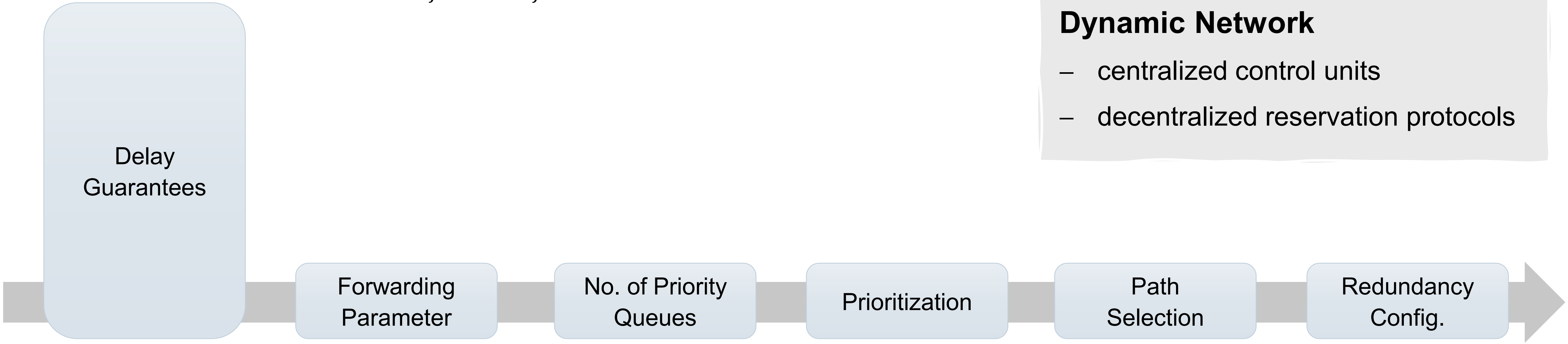
Network Planning

Flow f
Source
Destination
Data Size $\mathcal{S}_f$
Sending Interval $\mathcal{I}_f$
Deadline $\mathcal{D}_f$



Dynamic Network

- centralized control units
- decentralized reservation protocols



State-of-the-Art

Dynamic Networks

Information To-Go

TSN is still under development and many features are currently promised, but not developed.

Dynamic

centralized → no algorithm in standards

decentralized → standardized protocols, but…

1997
Resource Reservation Protocol (RSVP)

mostly relied on IntServ
with per-flow shaping

… only for specialized hardware
(e.g., rate-controlled strict priority, Worst-case Fair Weighted Fair Queueing (WF2Q), …)

2010
Stream Reservation Protocol (SRP)

for Credit-Based Shaper
with per-class shaping
IEEE Std 802.1Qat

… proven wrong (no real-time guarantees) [2]

2018
Resource Allocation Protocol (RAP)

support planned for various schedulers
IEEE Std P802.1Qdd **(draft)**

… no solutions for real-time guarantees yet

Latency Calculation

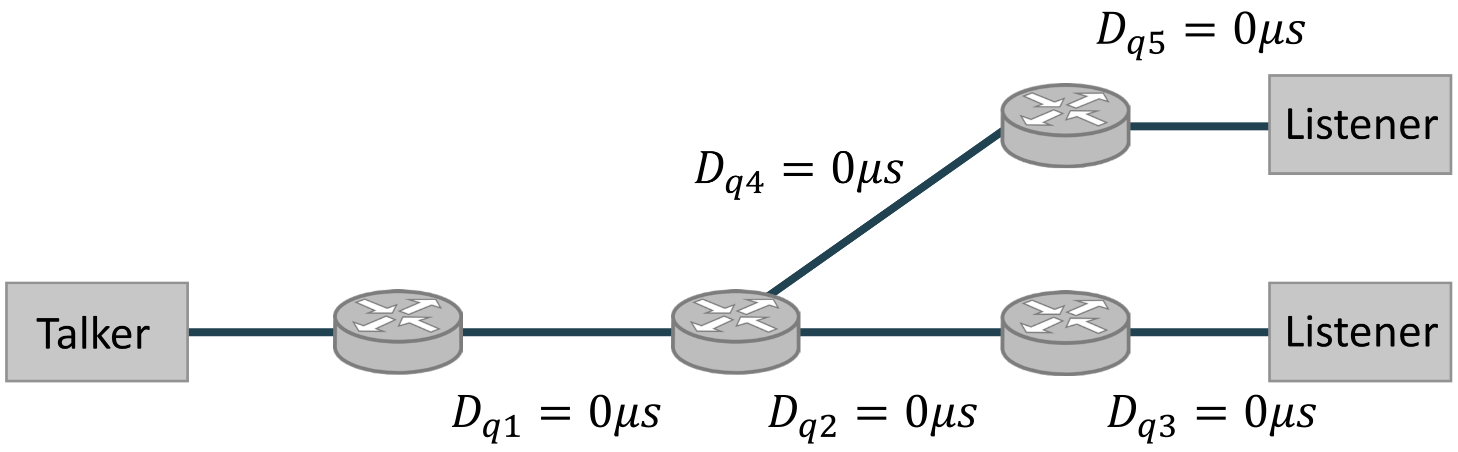
Re-configuration of whole network

Looking back: RSVP/IntServ

Flow latency does not depend on other flows – service guaranteed at all times

But now?

Service offered by TSN schedulers depends on scheduled flows
→ flow burstiness changes, which changes the delay in the subsequent network



Latency Calculation

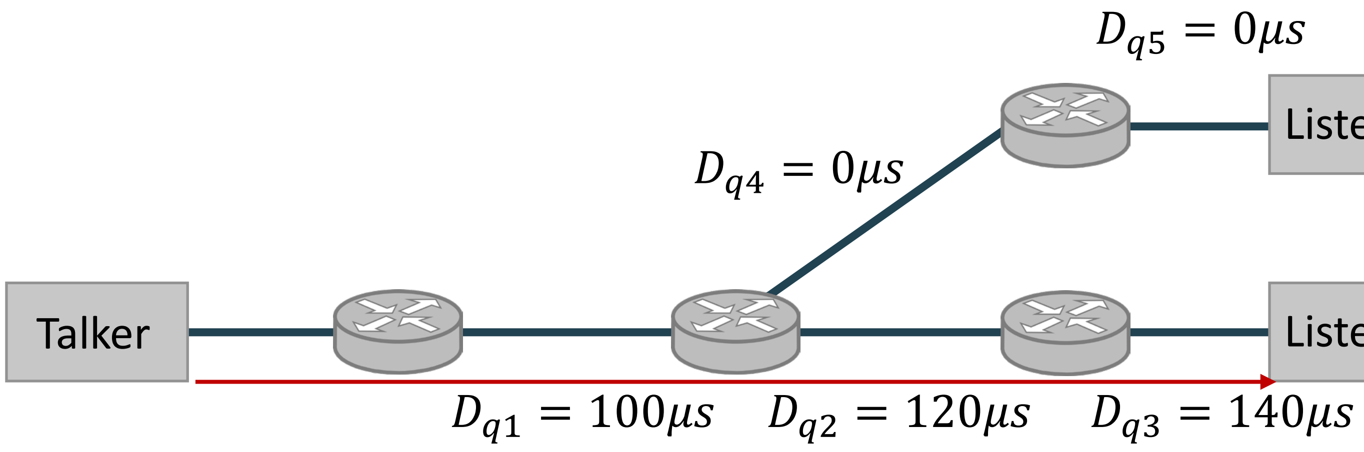
Re-configuration of whole network

Looking back: RSVP/IntServ

Flow latency does not depend on other flows – service guaranteed at all times

But now?

Service offered by TSN schedulers depends on scheduled flows
→ flow burstiness changes, which changes the delay in the subsequent network



Latency Calculation

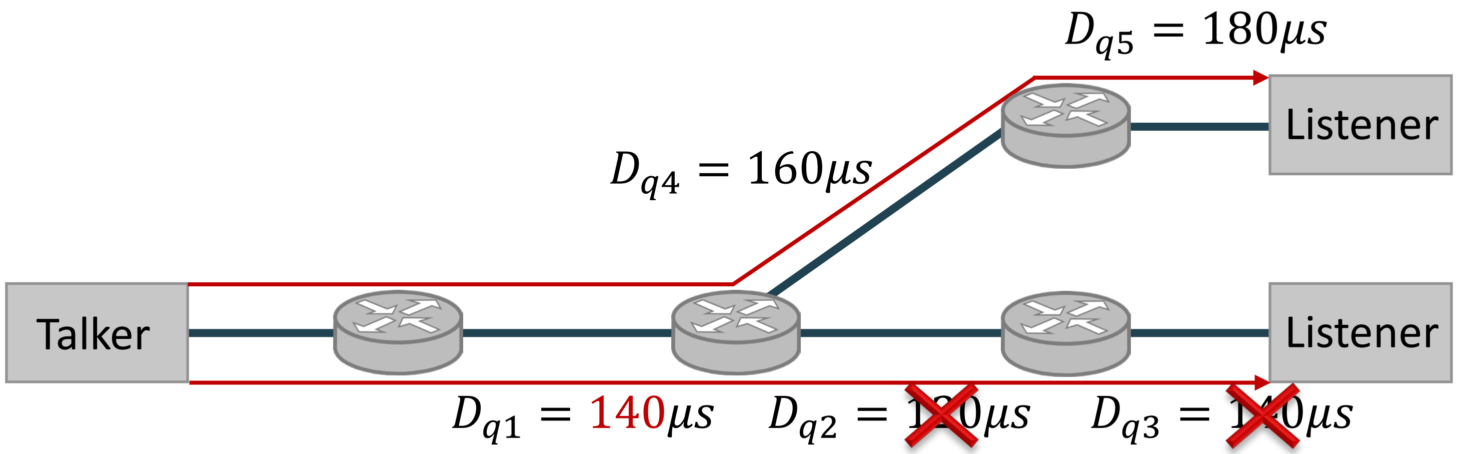
Re-configuration of whole network

Looking back: RSVP/IntServ

Flow latency does not depend on other flows – service guaranteed at all times

But now?

Service offered by TSN schedulers depends on scheduled flows
→ flow burstiness changes, which changes the delay in the subsequent network



Information To-Go

Adding new flows changes the guarantees (delay, buffer size,…) of potentially all existing flows.

Latency Calculation

Solution:

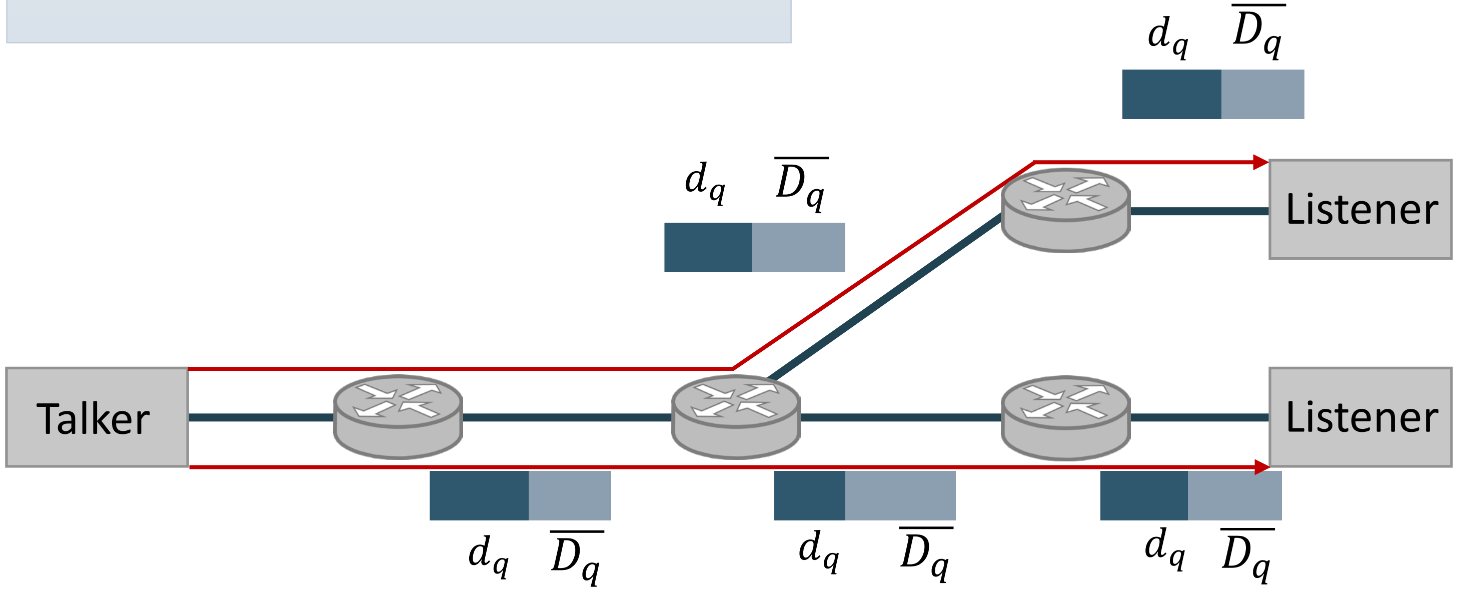
Per-Queue Latency Bounds

- Topology-Independent
- Interference-Independent

Admission Control

- Current Worst-Case Per-Queue Latency d
- Per-Queue Delay Bound $\overline{D_q}$

- Allow flow only if: $d_q \leq \overline{D_q}, \forall q$ on path
- Latency bound of flow: Determined by $\overline{D_q}$



Latency Calculation

State-of-the-Art

Previous Works Assumption: There exists a maximum d_q for Credit-Based Shaper

1) IEEE 802.1BA

$$d = t_{proc} + \underbrace{t_{L_{max}}}_{\text{other pr.}} + \underbrace{\left(\frac{idSl}{C} \cdot CMI - t_{L_{FoI}}\right) \cdot \frac{C}{idSl}}_{\text{same priority}} + t_{L_{FoI}-IPG}$$

2) IEEE 802.1Q

$$d = t_{inQueue} + t_{int} + t_L + t_{prop} + t_{sf}$$

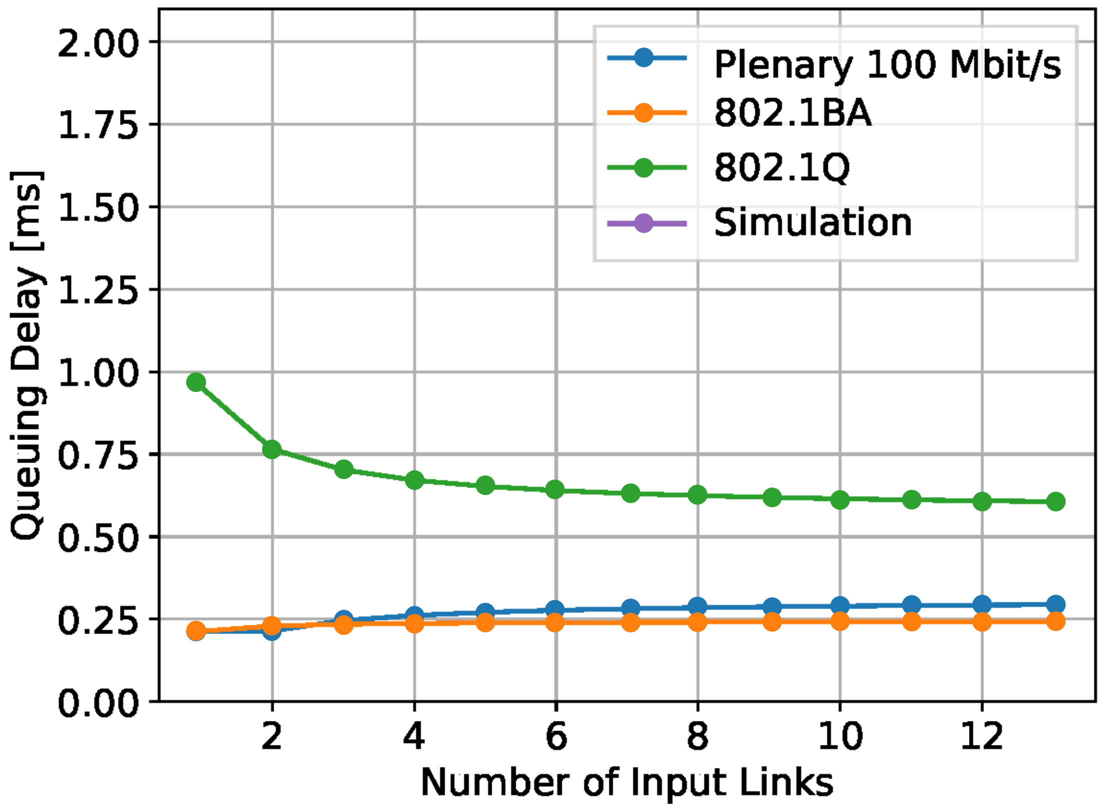
includes

$$t_{queue} = \begin{cases} L_{max}/C & \text{for prio. 7} \\ (L_{max} + L^{(7)})/(C - idSl^{(7)}) & \text{for prio. 6} \end{cases}$$

3) Plenary 100 Mbit/s

$$d = \left(\underbrace{L_{max}}_{\text{other pr.}} + \underbrace{2 \cdot (R_{max} - L_{FoI}) - \left\lceil \frac{R_{max} - L_{FoI}}{N} \right\rceil}_{\text{same priority}} + L_{FoI}\right) \cdot t_{oct}$$

$$R_{max} = \left\lfloor \frac{CMI}{t_{oct}} \cdot \frac{idSl}{C} \right\rfloor, \; N = \min\left(\left|\mathcal{L}^-\right|, \left\lfloor \frac{R_{max} - L_{FoI}}{L_{min}} \right\rfloor\right)$$



results and analysis from [2]

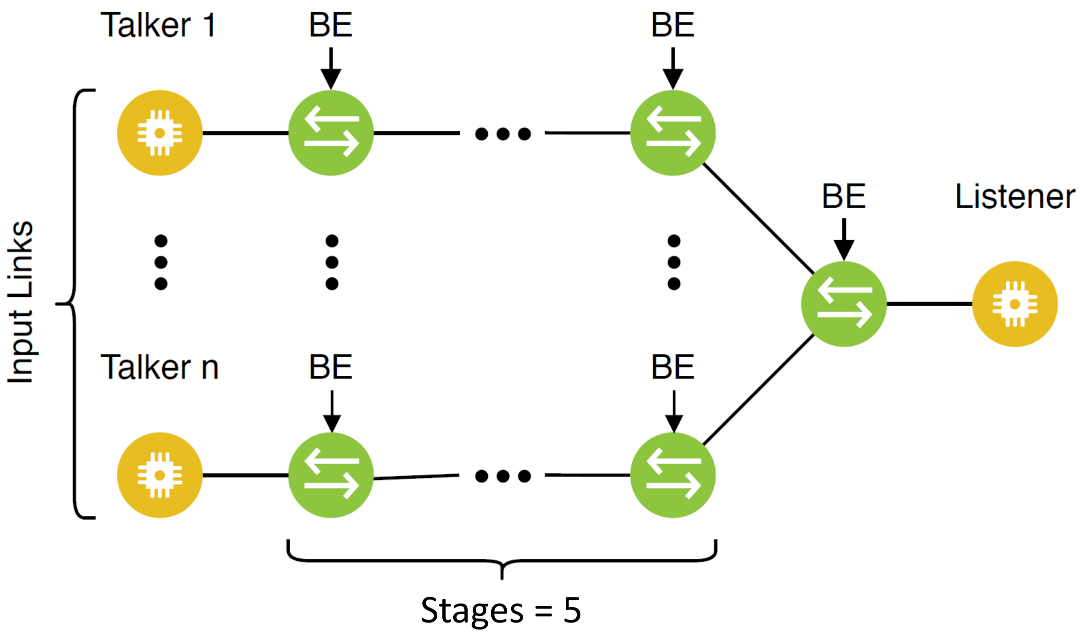
Latency Calculation

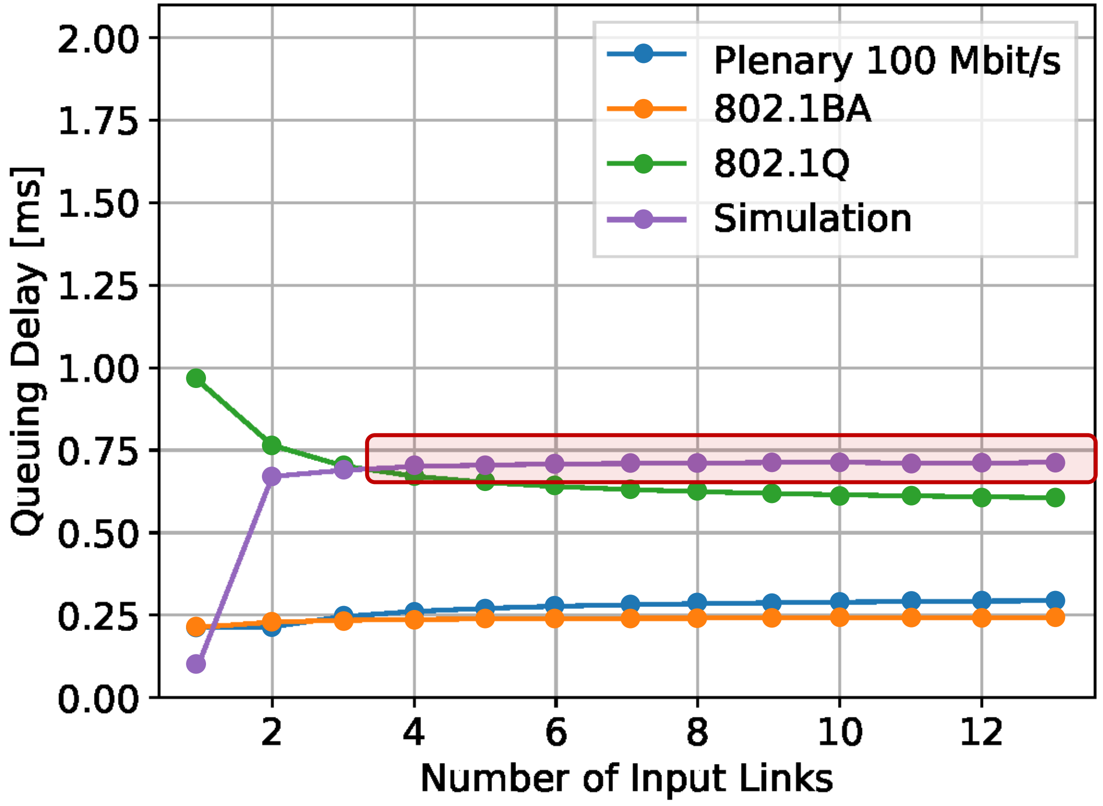
State-of-the-Art

Information To-Go

Delays in a network cannot be „forseen“ in a dynamic environment.

Previous Works Assumption: There exists a maximum d_q for Credit-Based Shaper



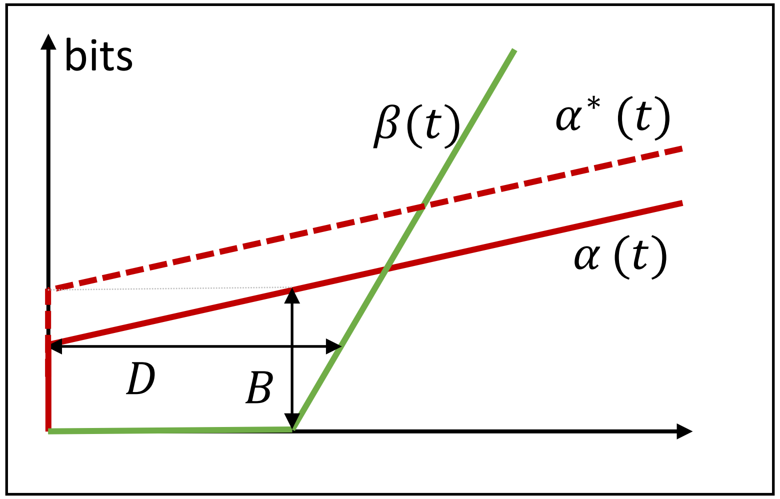


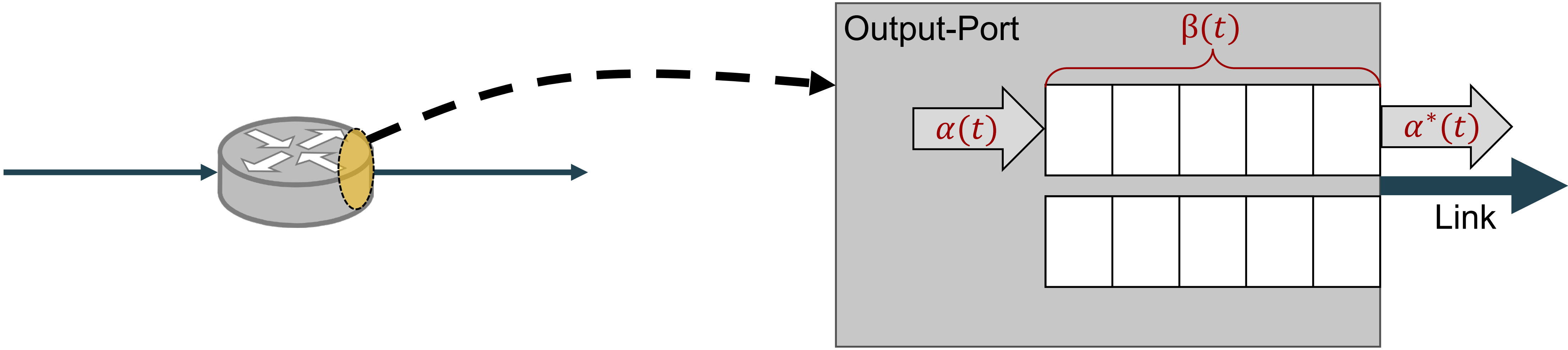
+ analytical proof that d_q cannot be upper bounded in dynamic scenarios [2]

Latency Calculation

New Solution: Network Calculus

- Modeling of Communication Systems
- Worst-Case Performance Guarantees: max. Delays, max. Buffer Sizes, max. Output, …
- Cumulative Functions: max. Arrival Curve $\alpha(t)$ and min. Service Curve $\beta(t)$

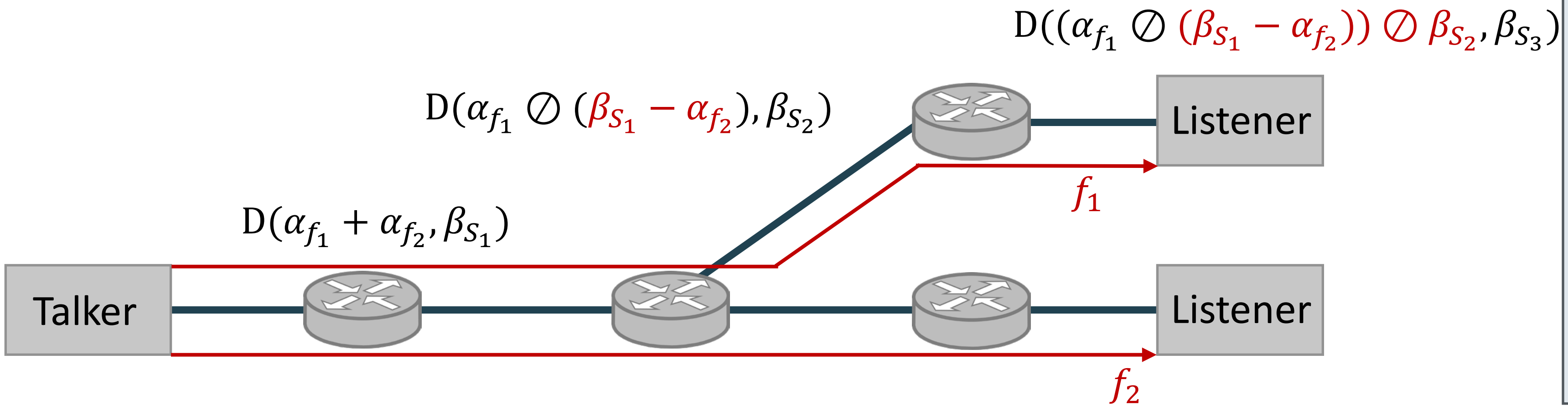




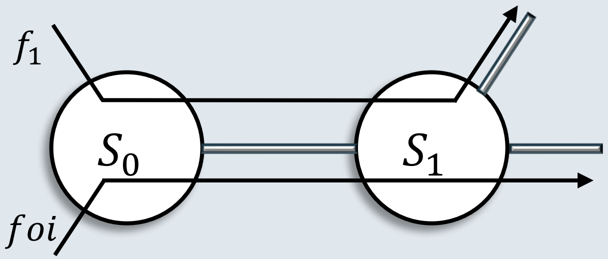
Latency Calculation

Min-Plus Algebra

- Arrival Curve: $\forall s \leq t\colon R(t) - R(s) \leq \alpha(t-s)$
- Strict Service Curve: $\forall$ backlog-period $]s,t]\colon R^*(t) - R^*(s) \geq \beta(t-s)$
- Aggregation: $(\alpha_{f_1} + \alpha_{f_2})(t) = \alpha_{f_1}(t) + \alpha_{f_2}(t)$
- Max. Output: $(\alpha \oslash \beta)(t) = \sup_{u\geq 0}\{\alpha(t+u) - \beta(u)\}$
- Ind. Service Curve: $(\alpha \ominus \beta)(t) = \sup_{u\geq 0}\{\beta(u) - \alpha(u)\}$
- Max. Delay: $\mathrm{D}(\alpha,\beta) = \sup_{t\geq 0}\{\inf_{d\geq 0}\{d : \alpha(t) \leq \beta(t+d)\}\}$



Example



$$D_{foi}^{max} = D_{S_0} + D_{S_1}$$

$$D_{S_0} = D(\alpha_{S_0}, \beta_{S_0})$$

$$\alpha_{S_0} = \alpha^{f_1} + \alpha^{f_{foi}}$$

$$D_{S_1} = D(\alpha_{S_1}, \beta_{S_1})$$

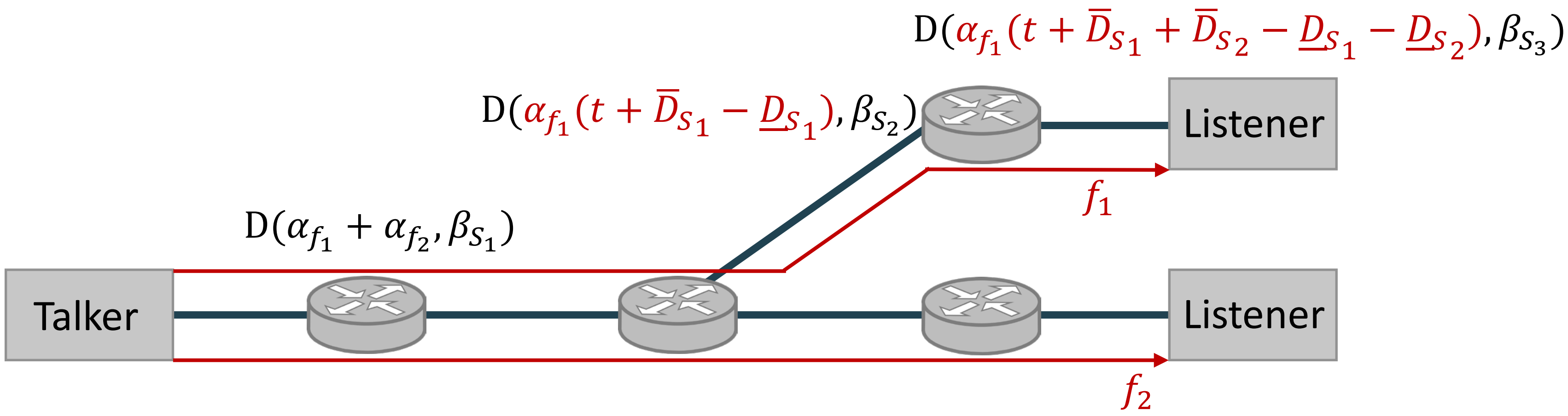
~~$\alpha_{S_1} = \alpha^{f_{foi}} \oslash (\beta_{S_0} \ominus \alpha^{f_1})$~~

$$\alpha_{S_1} = \alpha^{f_{foi}}(t + \overline{D}_{S_0} + \underline{D}_{S_0})$$

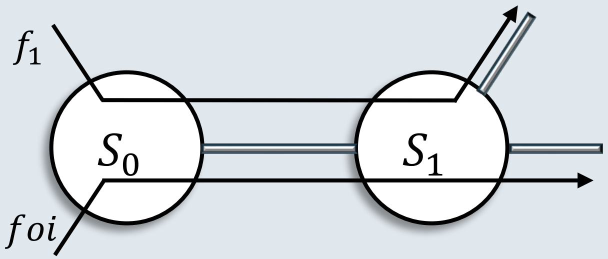
Latency Calculation

Min-Plus Algebra

- Arrival Curve: $\forall s \leq t\colon R(t) - R(s) \leq \alpha(t-s)$
- Strict Service Curve: $\forall$ backlog-period $]s,t]\colon R^*(t) - R^*(s) \geq \beta(t-s)$
- Aggregation: $(\alpha_{f_1} + \alpha_{f_2})(t) = \alpha_{f_1}(t) + \alpha_{f_2}(t)$
- Max. Output: $(\alpha \oslash \beta)(t) = \sup_{u\geq 0}\{\alpha(t+u) - \beta(u)\}$
- Ind. Service Curve: $(\alpha \ominus \beta)(t) = \sup_{u\geq 0}\{\beta(u) - \alpha(u)\}$
- Max. Delay: $\mathrm{D}(\alpha,\beta) = \sup_{t\geq 0}\{\inf_{d\geq 0}\{d : \alpha(t) \leq \beta(t+d)\}\}$



Example



$D_{foi}^{max} = D_{S_0} + D_{S_1}$

$D_{S_0} = D(\alpha_{S_0}, \beta_{S_0})$

$\alpha_{S_0} = \alpha^{f_1} + \alpha^{f_{foi}}$

$D_{S_1} = D(\alpha_{S_1}, \beta_{S_1})$

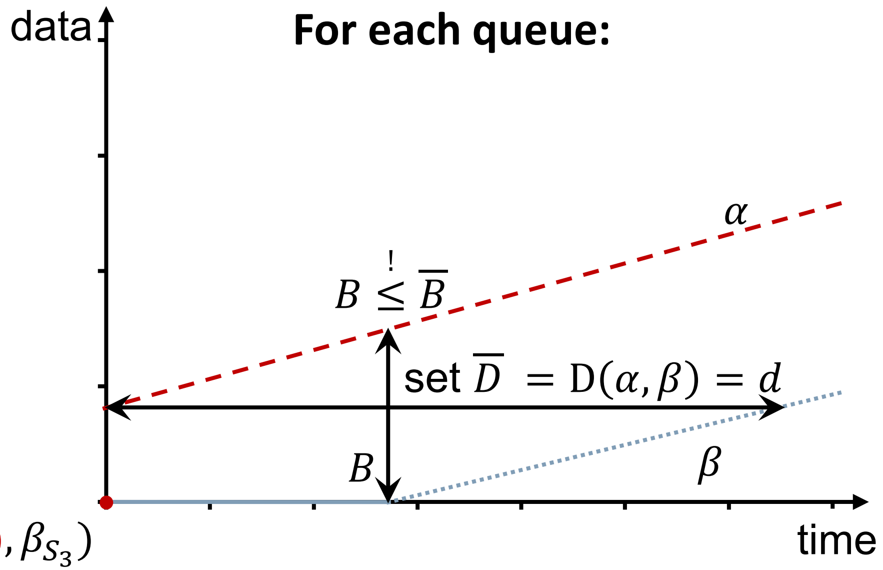
~~$\alpha_{S_1} = \alpha^{f_{foi}} \oslash (\beta_{S_0} \ominus \alpha^{f_1})$~~

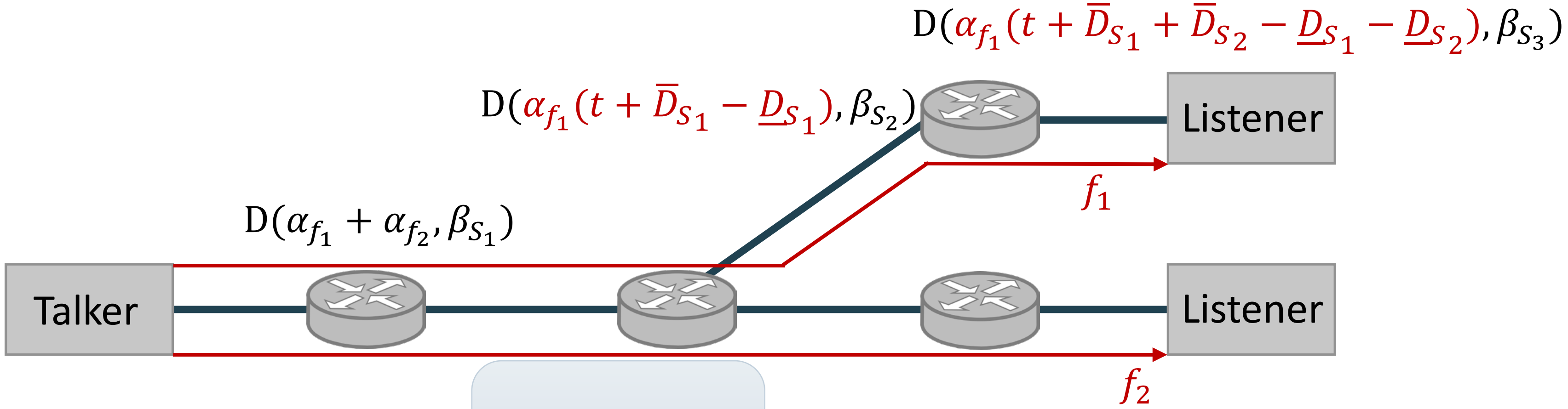
$\alpha_{S_1} = \alpha^{f_{foi}}(t + \overline{D}_{S_0} + \underline{D}_{S_0})$

Latency Calculation

Information To-Go

Network Calculus can be adapted to not depend on flows in other parts of the network.



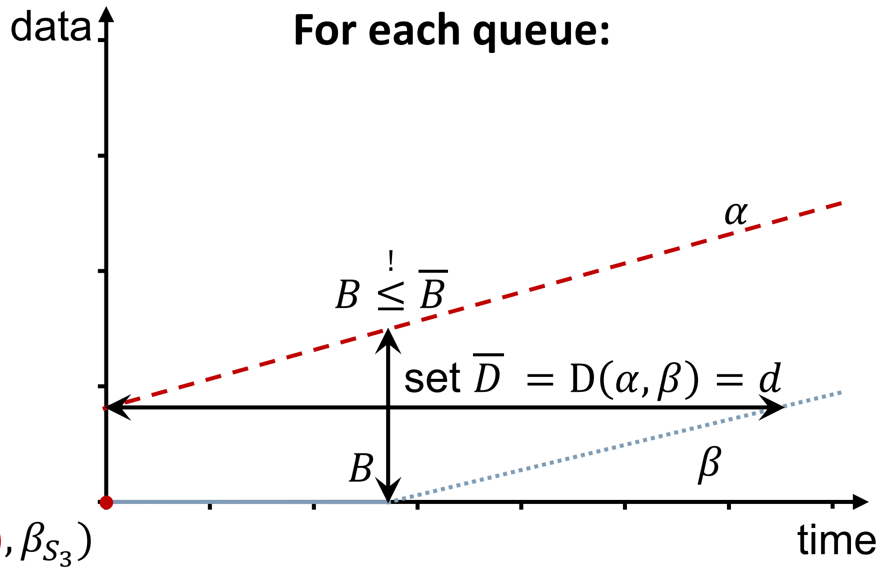


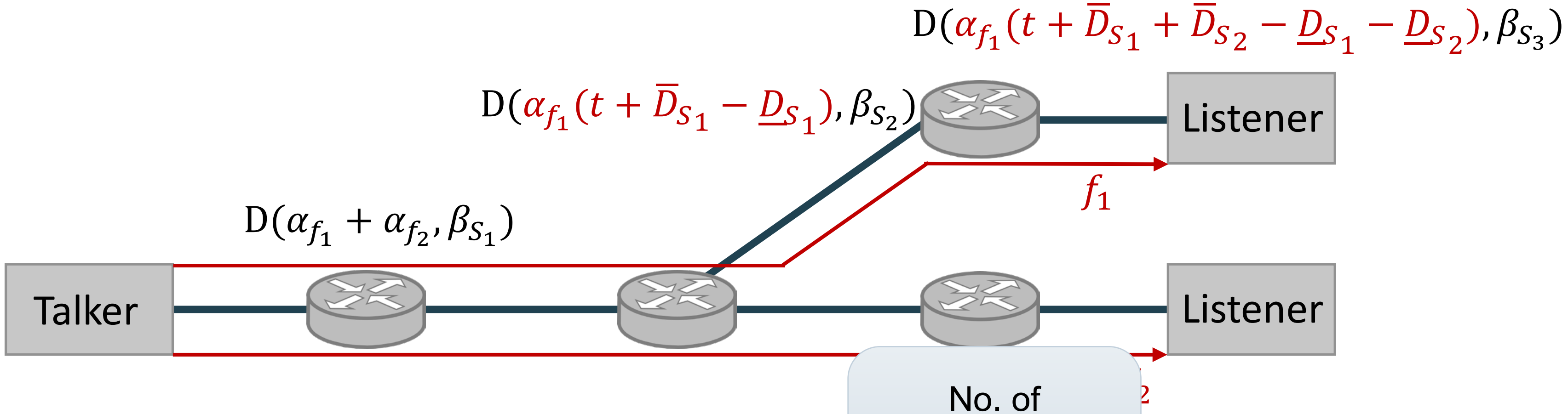
Delay Guarantees | **Forwarding Parameter** | No. of Priority Queues | Prioritization | Path Selection | Redundancy Config.

Latency Calculation

Information To-Go

Network Calculus can be adapted to not depend on flows in other parts of the network.

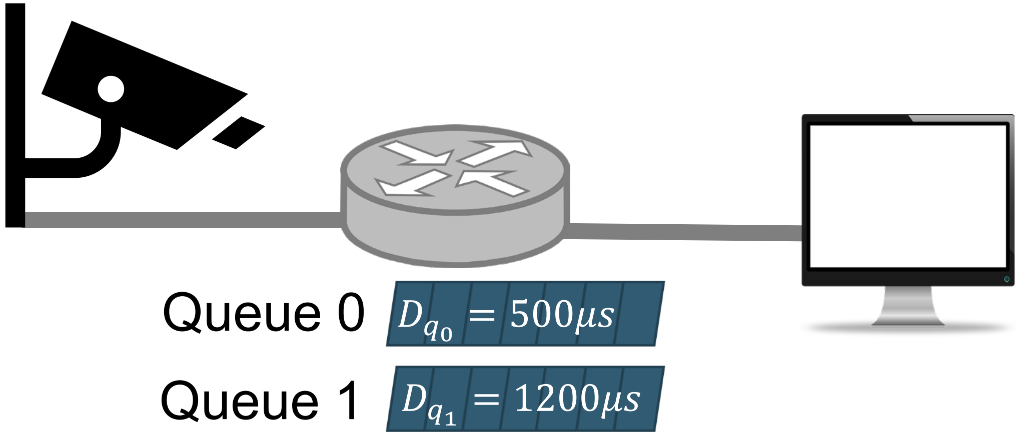


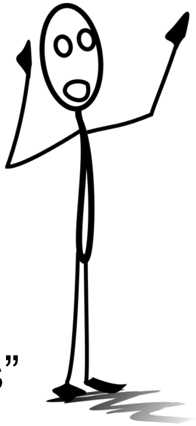


Delay Guarantees → Forwarding Parameter → No. of Priority Queues → Prioritization → Path Selection → Redundancy Config.

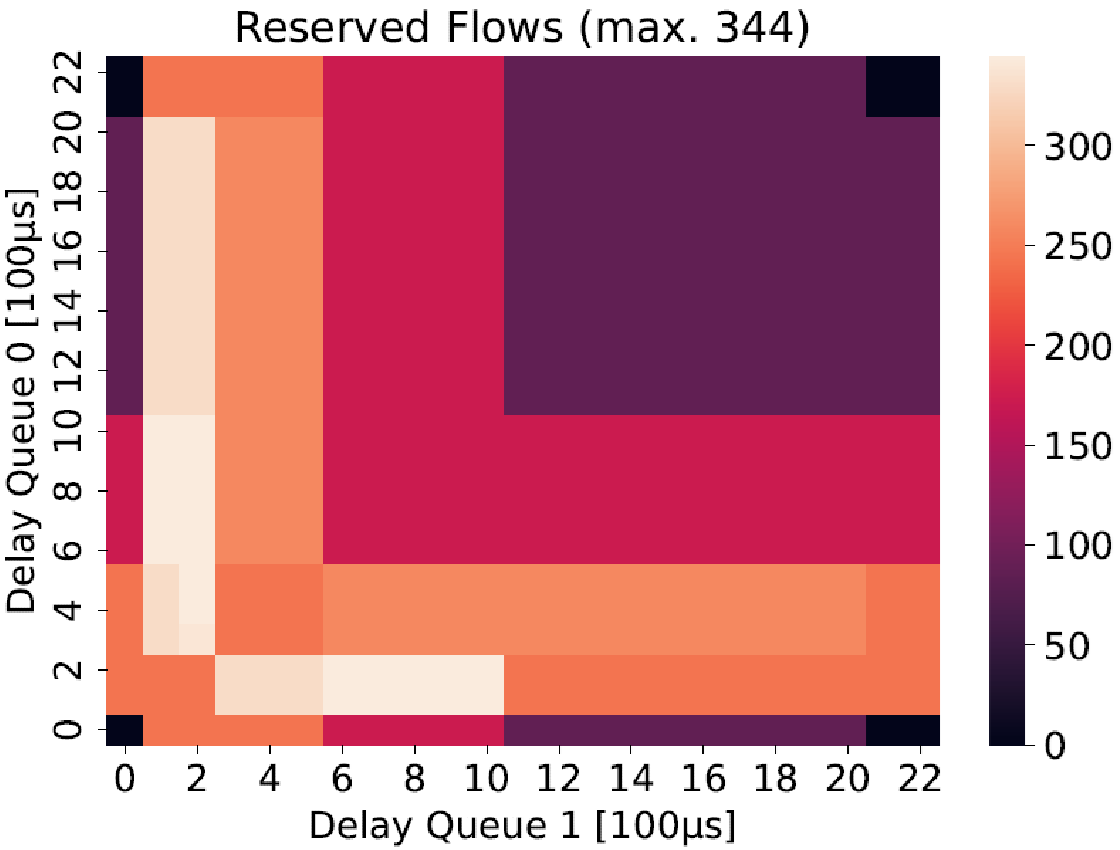
Delay Budgets

Importance

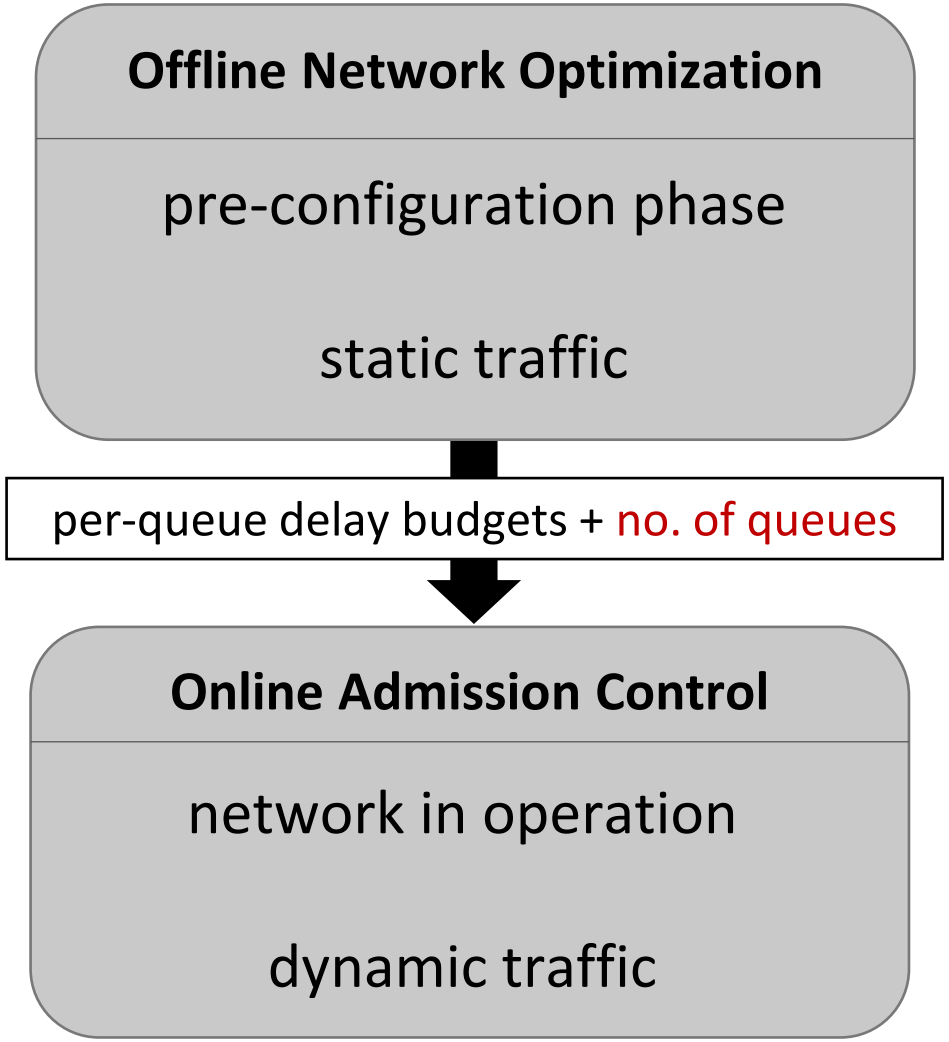




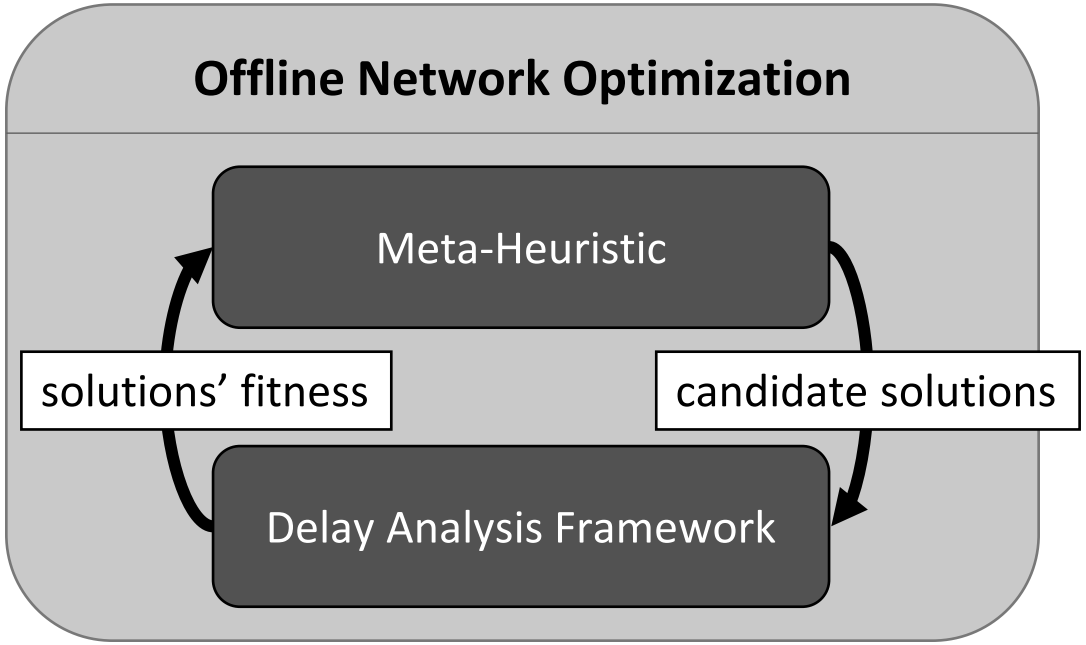
State of the Art: “Educated Guess”



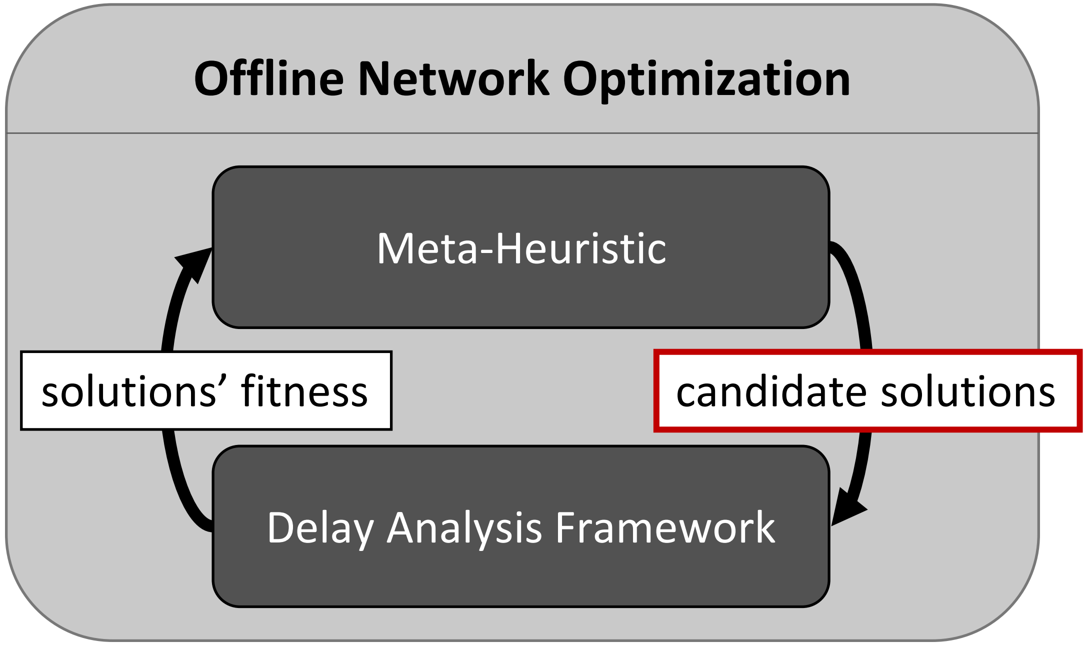
Framework

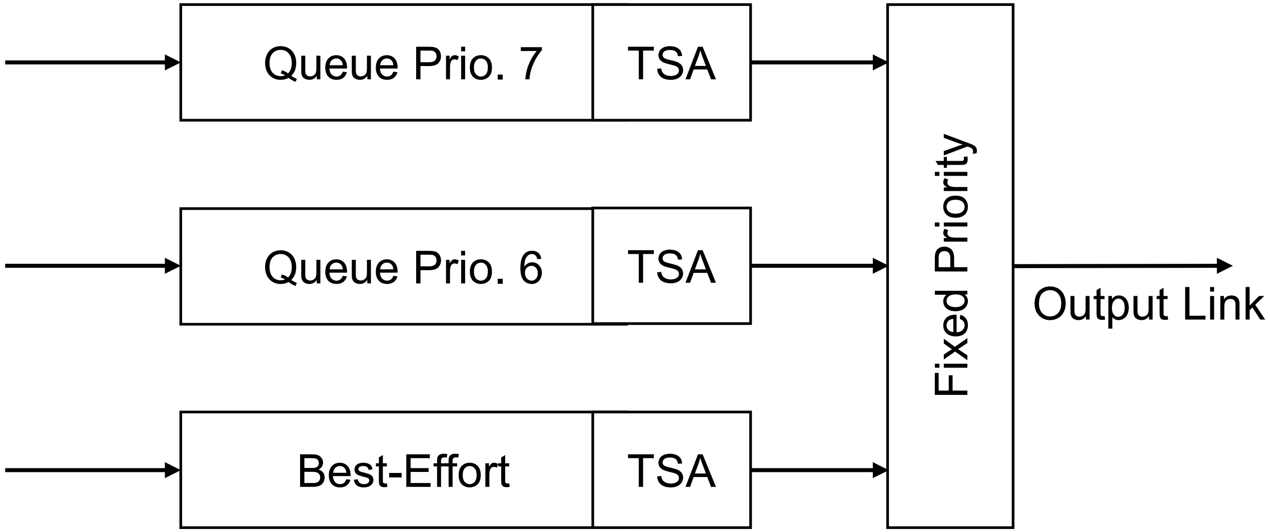

Offline Network Optimization
pre-configuration phase
static traffic
per-queue delay budgets + no. of queues
Online Admission Control
network in operation
dynamic traffic

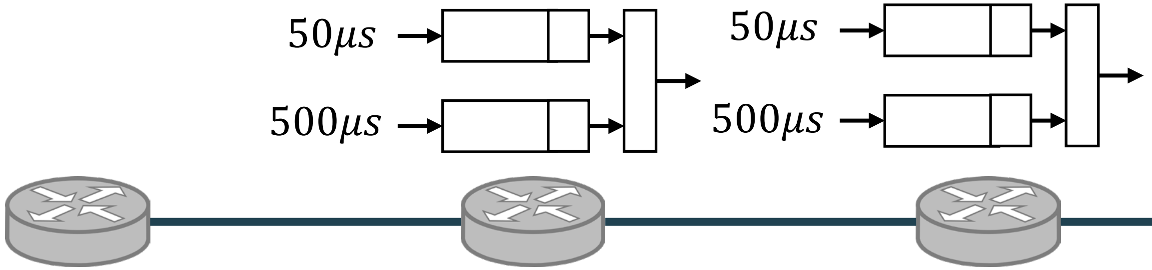
Candidate Solutions



Candidate Solutions



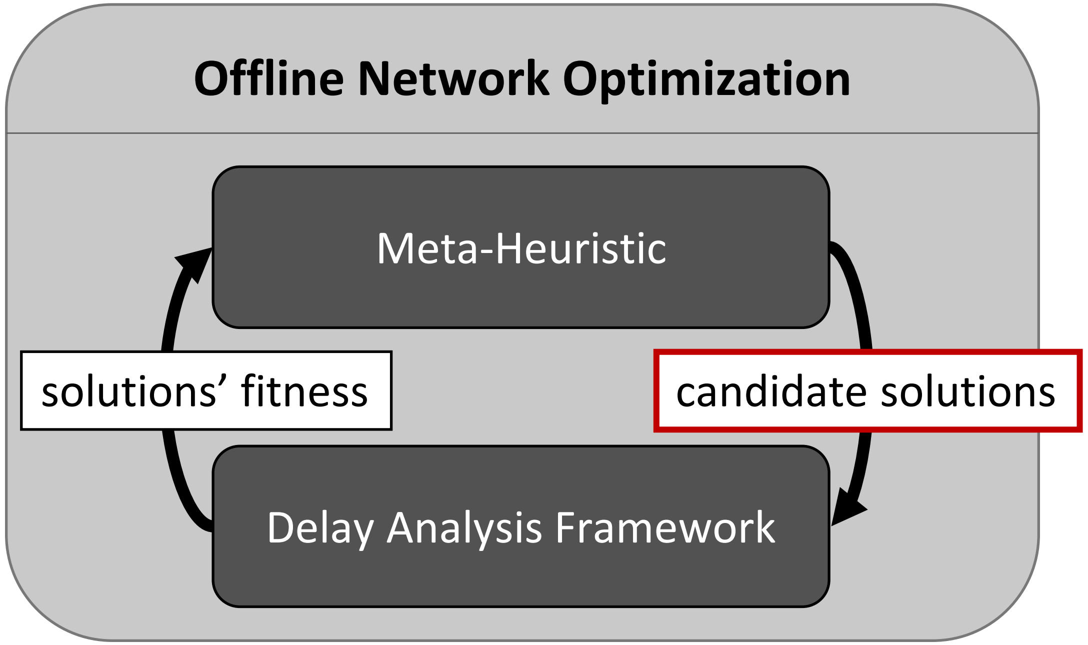


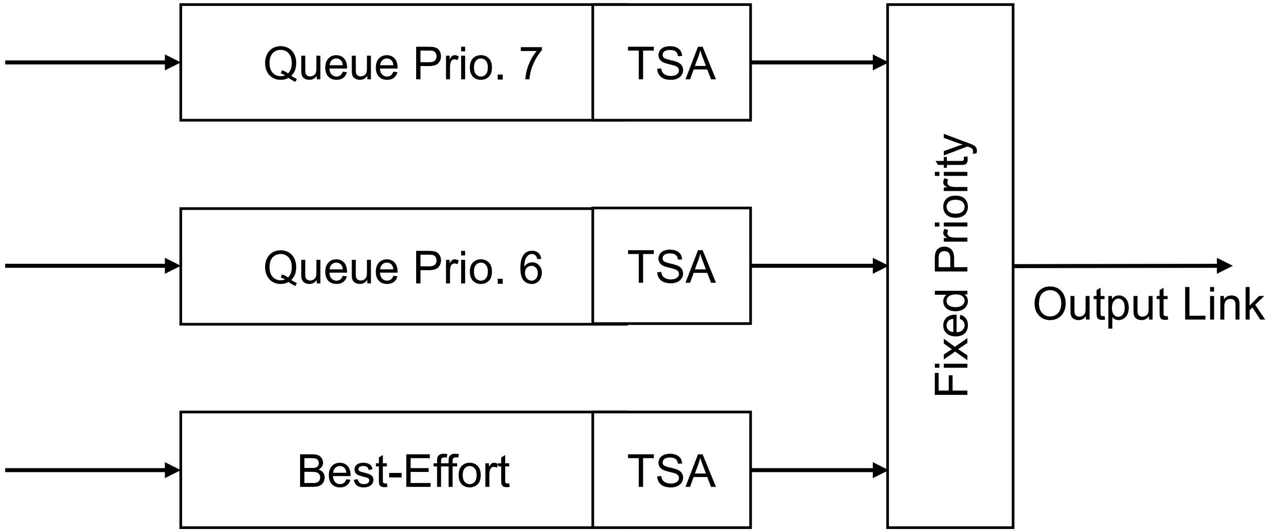


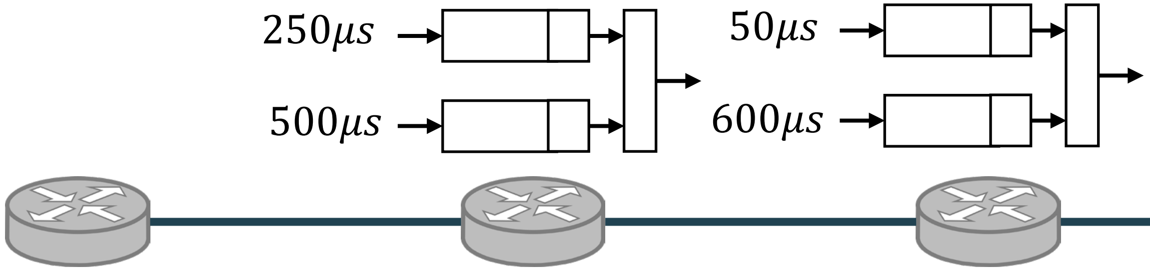
candidate solution with uniform delays:

$[50\mu s, 500\mu s]$

Candidate Solutions



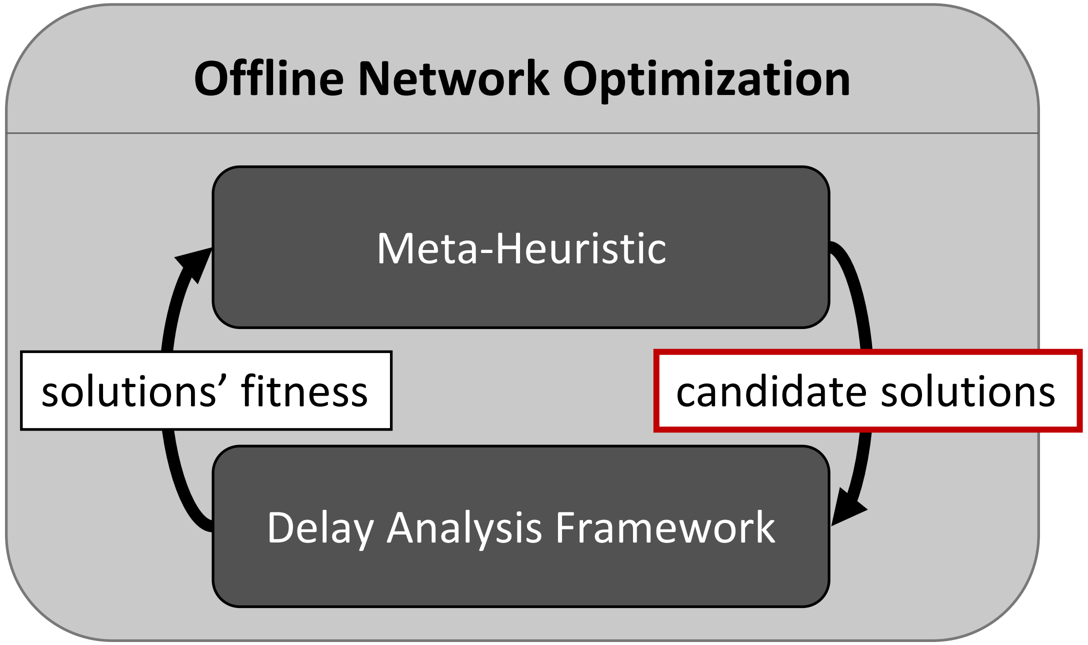


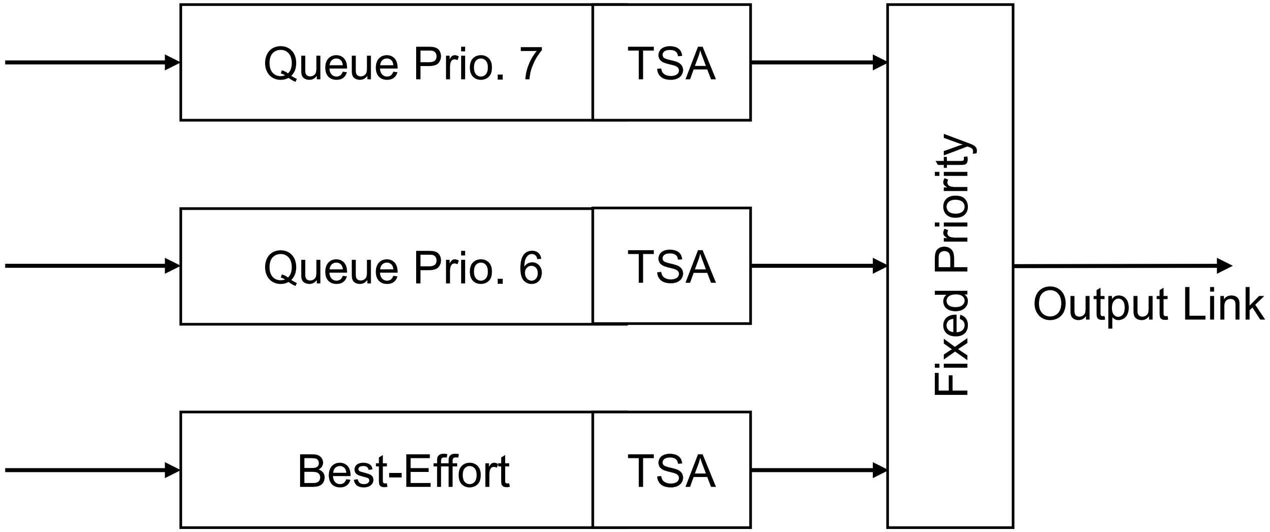


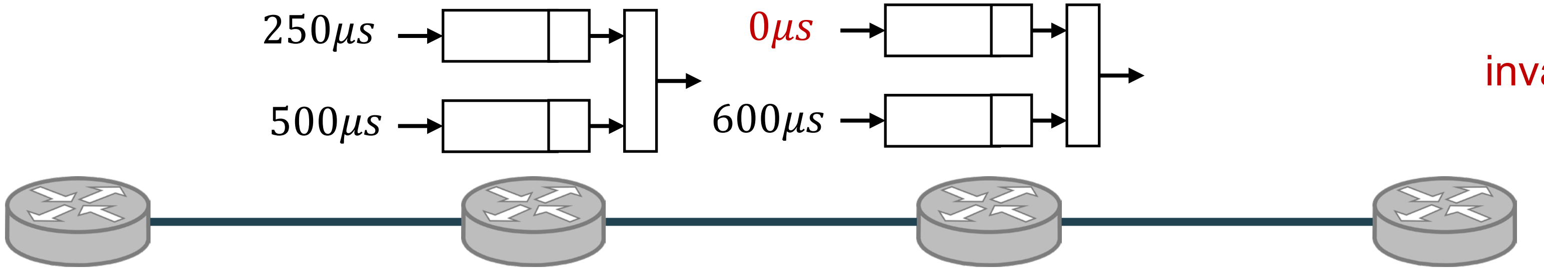
candidate solution with individual delays:

$[250\mu s, 500\mu s, 50\mu s, 600\mu s]$

Candidate Solutions

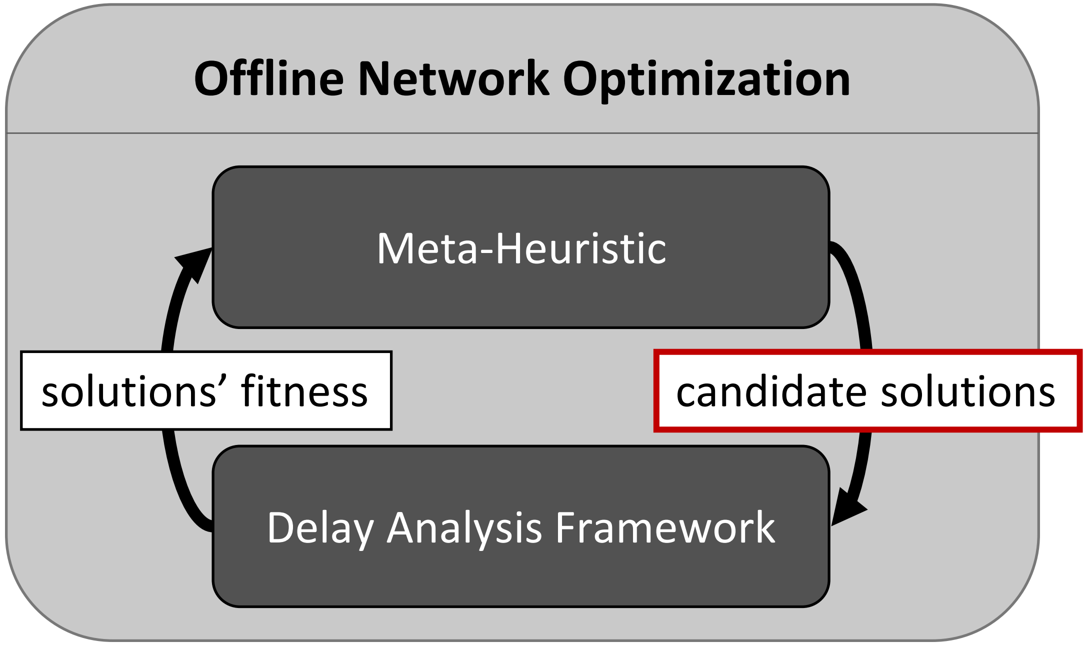


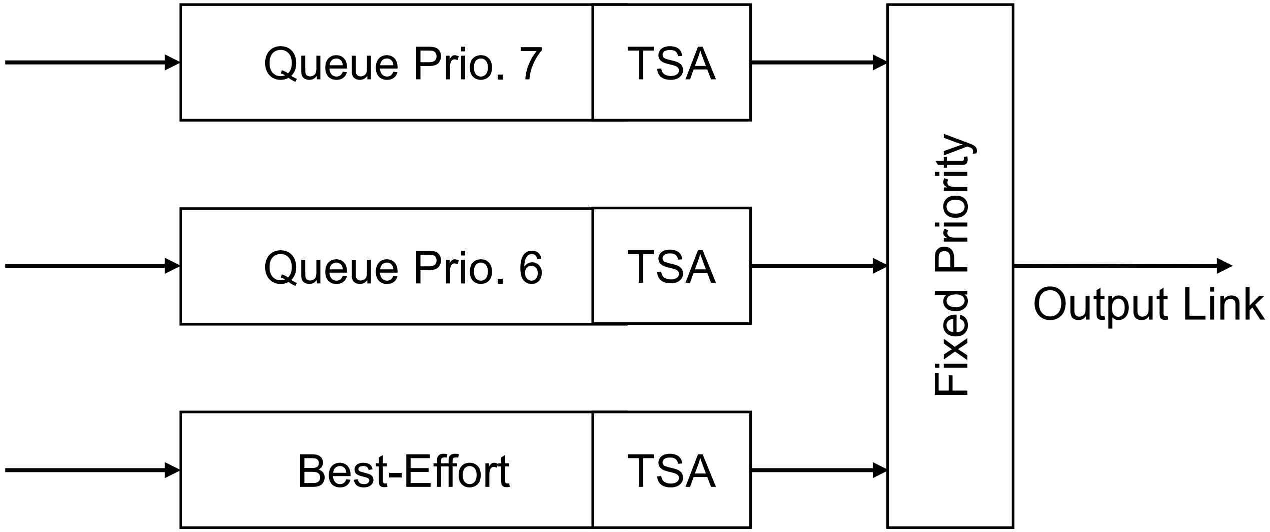


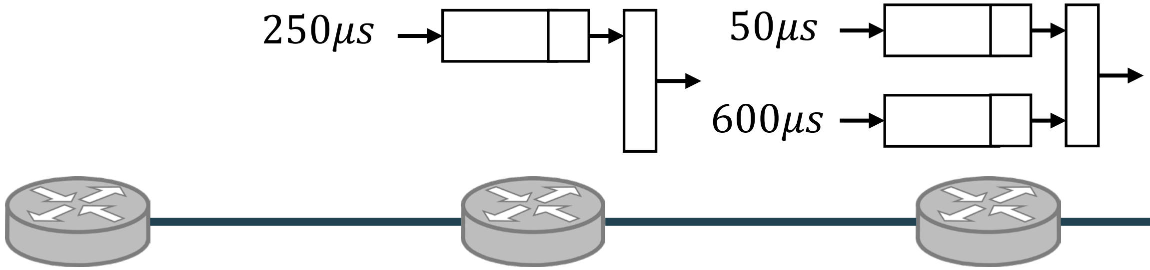


invalid candidate solution

Candidate Solutions



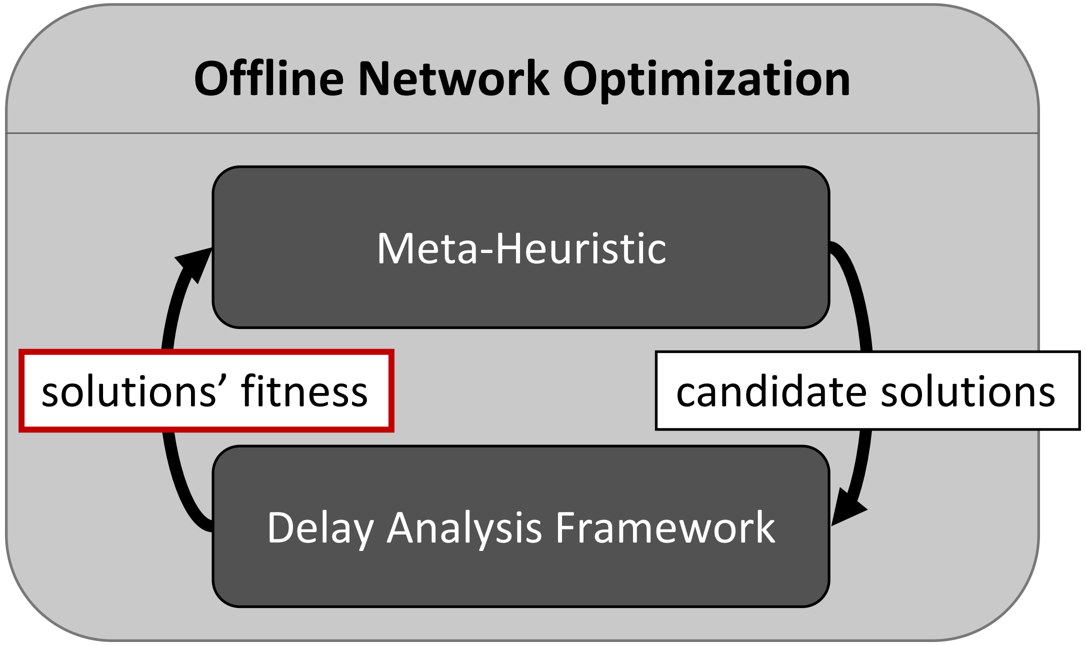




valid candidate solution
(defines no. of priorities)

Solution's Fitness

Fitness / Reward Function



Solution‘s Fitness

Fitness / Reward Function

$$f(s) = \omega_1 \cdot f_R(s) + \omega_2 \cdot f_A(s) + \omega_3 \cdot f_D(s), \; \omega_1 + \omega_2 + \omega_3 = 1$$

$f_R(s) = 1$

if all static flows are reserved

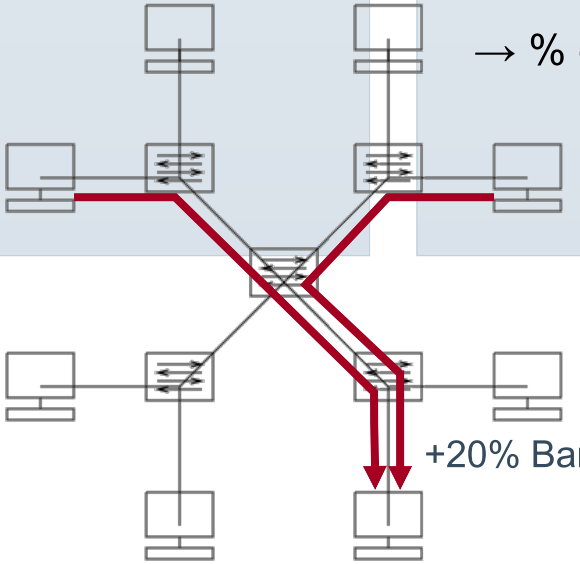
$f_A(s) = 1$

if dynamic future flows “have space”
→ % of bandwidth for defined traffic classes

$f_D(s) = 1$

if end-to-end delays match deadline requirements (within 30% interval)

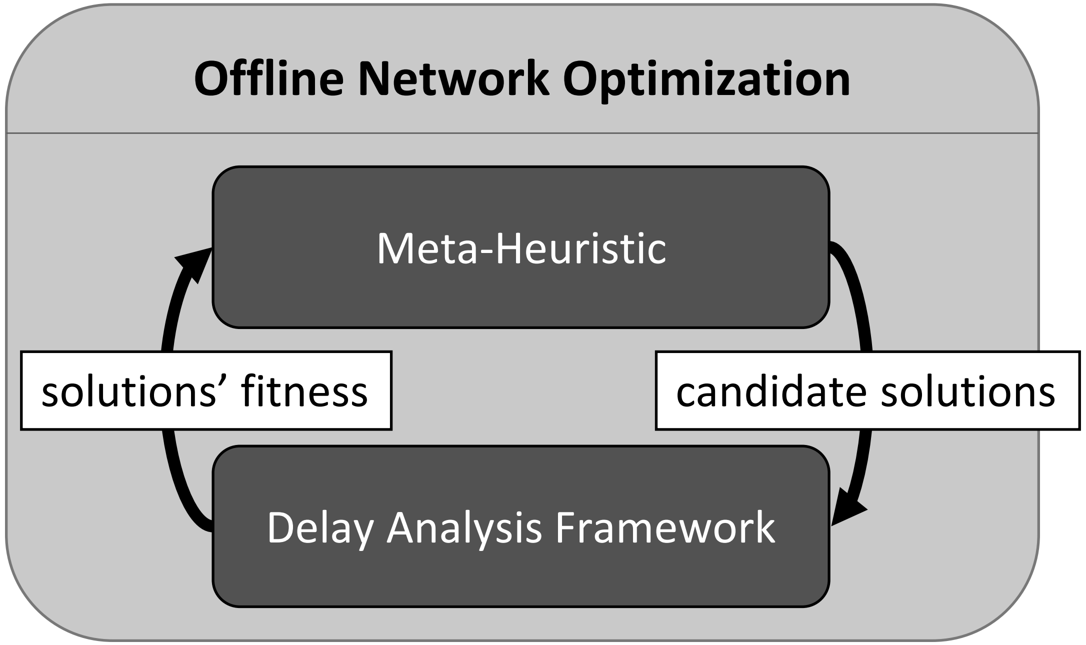
→ because low delays require more resources



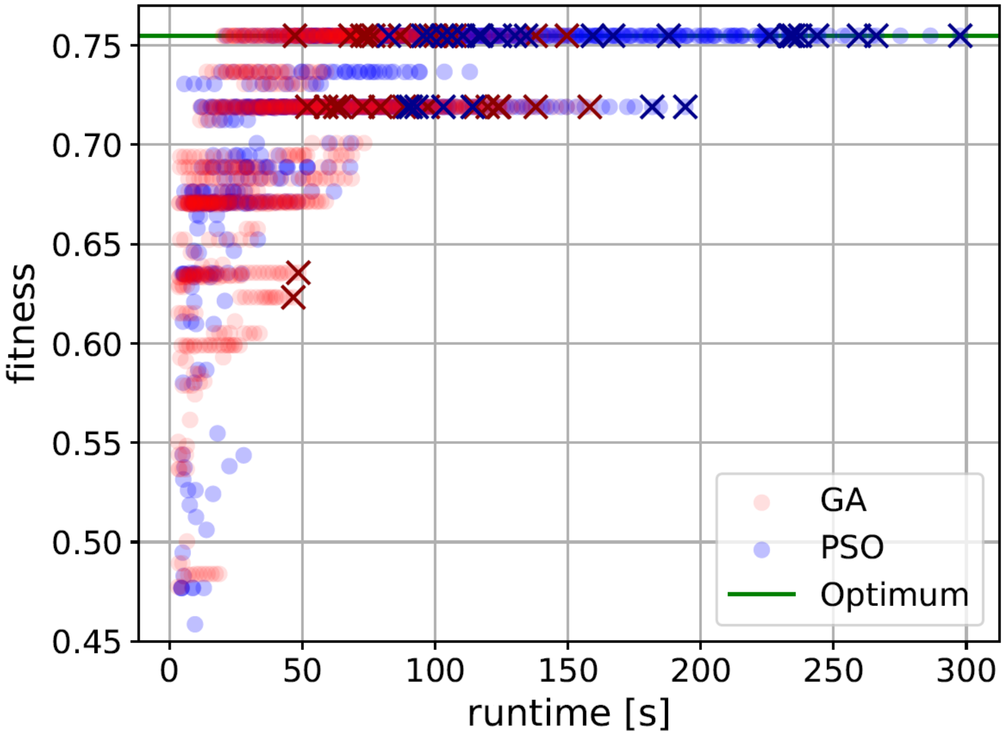
	Sending Interval	Max. Frame Size	Max. Latency
Profile 1	250 µs	64B	250 µs
Profile 2	500 µs	128B	500 µs
Profile 3	1000 µs	256B	1000 µs
Profile 4	2000 µs	512B	2000 µs
Profile 5	4000 µs	1024B	4000 µs

$0.7 \cdot D_\tau$ $\quad D_\tau$ $\quad D_p(s)$

Meta-Heuristic



Genetic Algorithm (GA)
vs.
Particle Swarm Optimization (PSO)



Benchmark Algorithms

Genetic Algorithm (GA)
Particle Swarm Optimization (PSO)

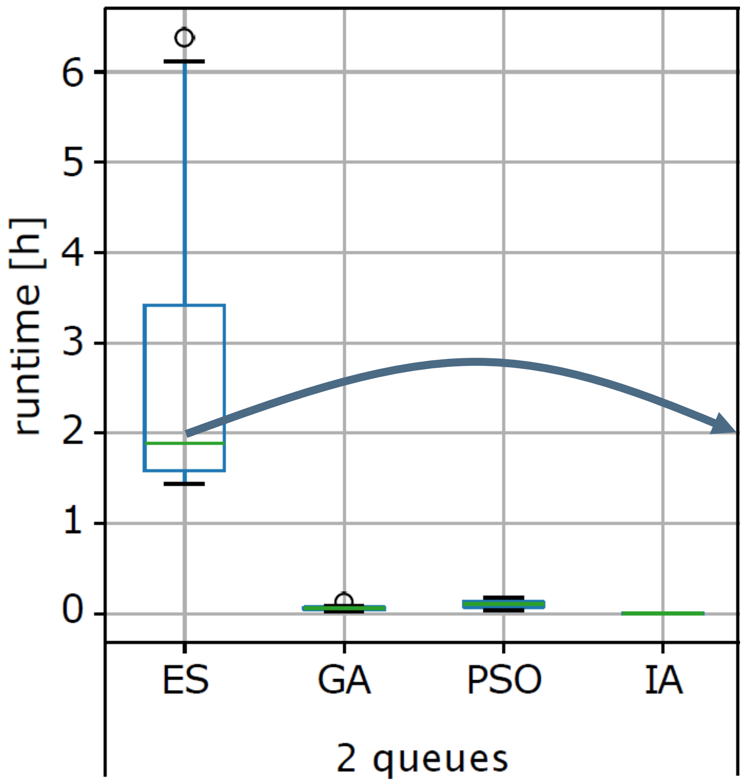
new solution → to be evaluated

Exhaustive Search (ES)

all solutions → optimum

Intuitive Approach (IA)

deadline of static flows uniformly distributed over path
→ “educated guess”

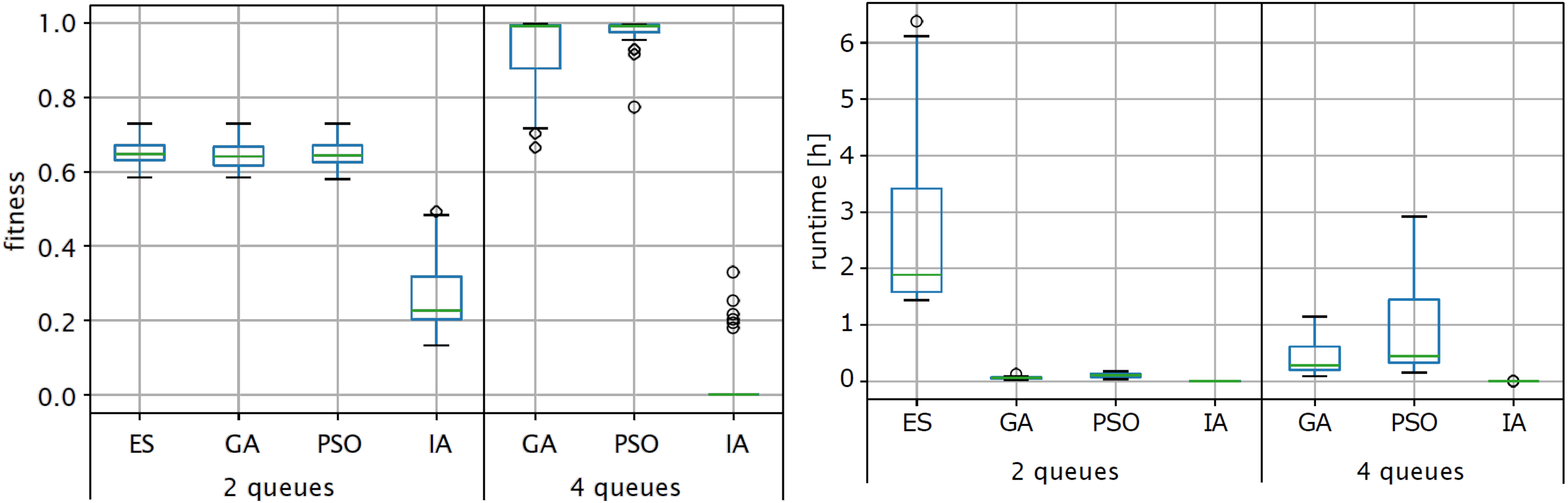


Evaluation

Benchmark Algorithms

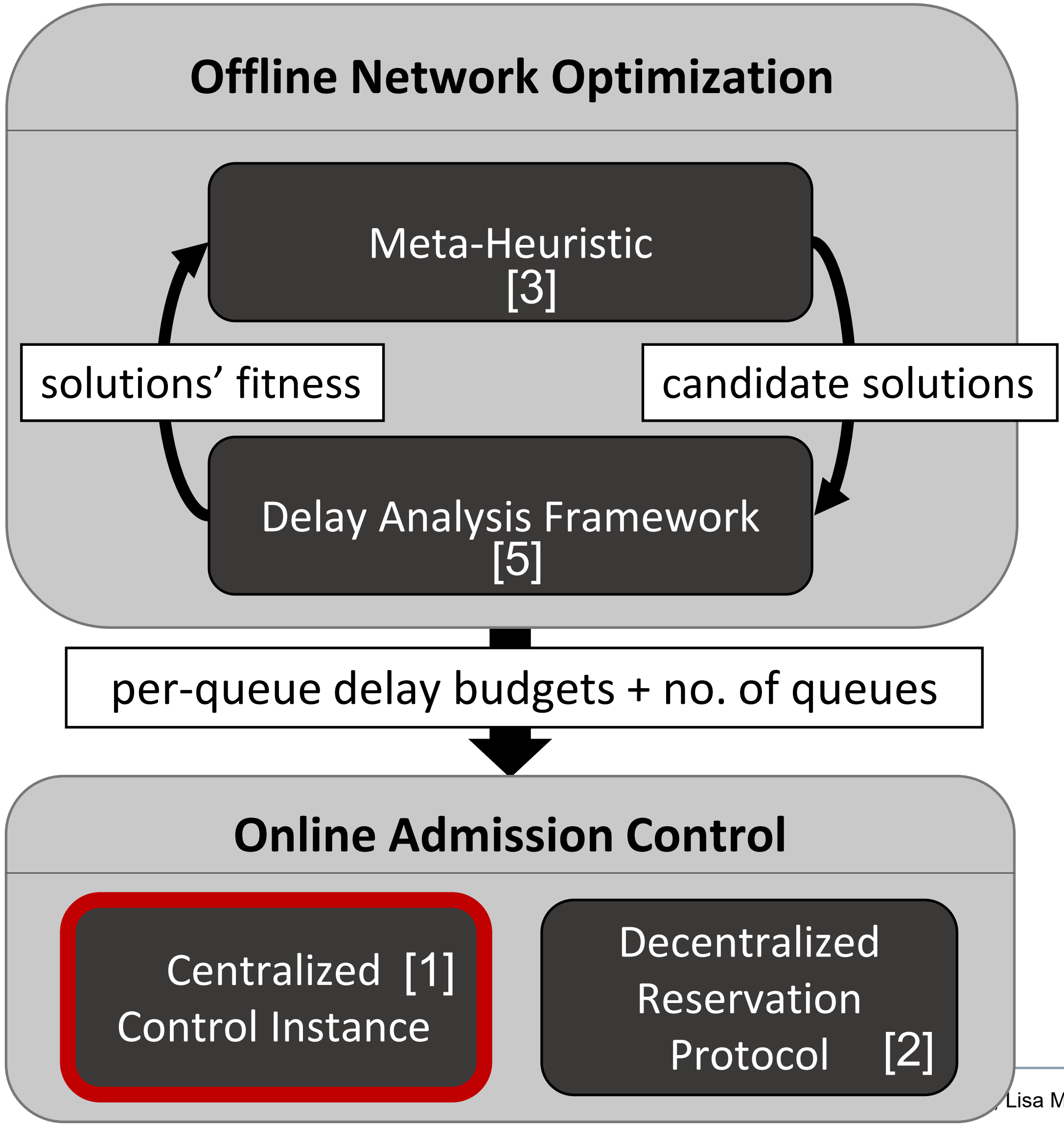
Information To-Go

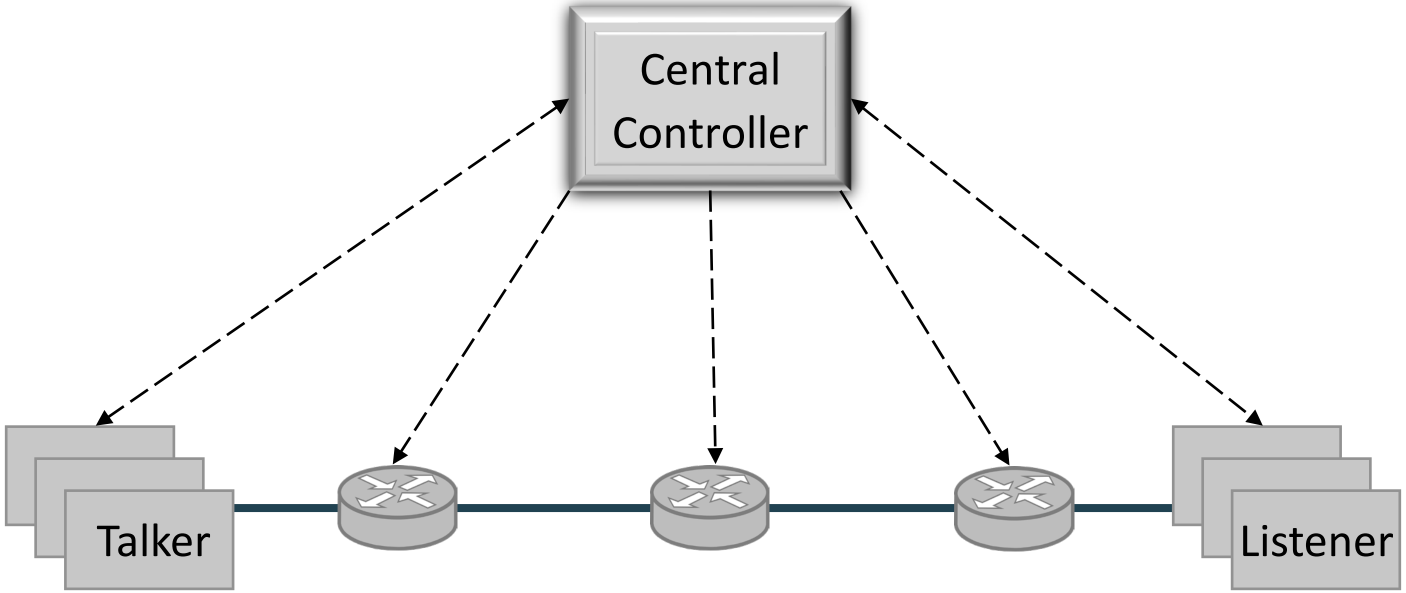
Meta-heuristics are highly efficient in testing a large variety of network setups.



Framework Overview

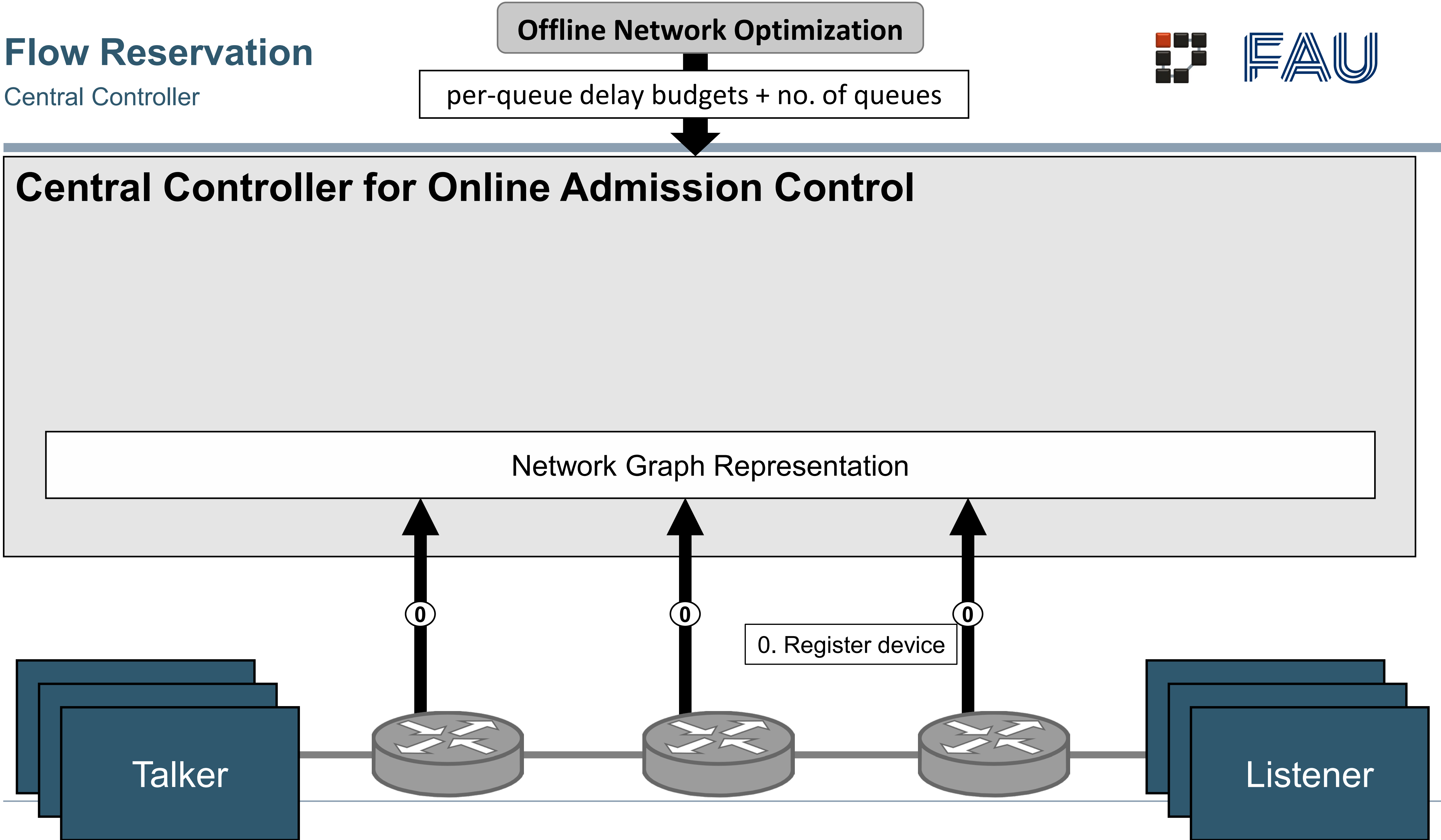
Combination of Offline and Online Control





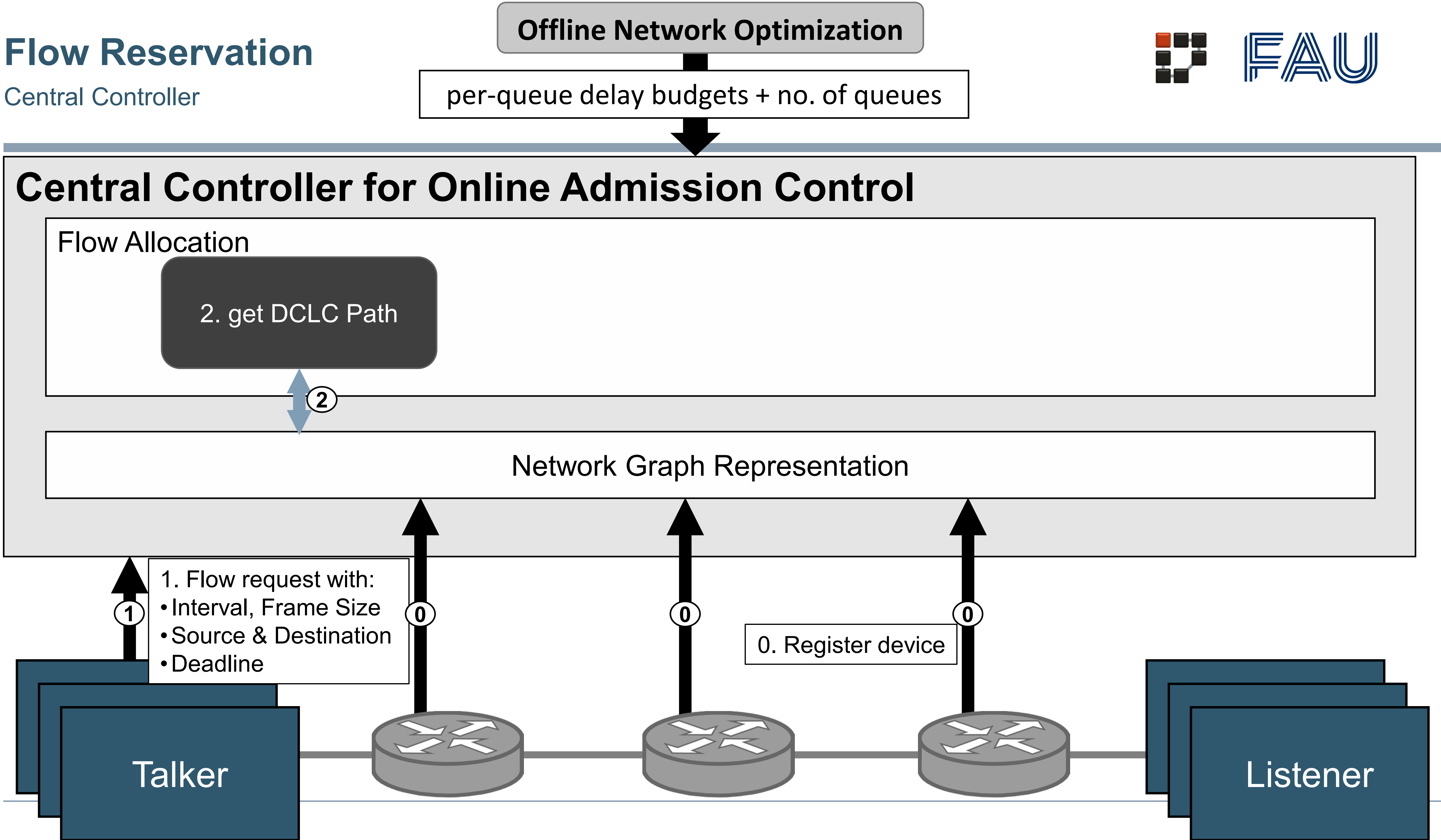
Flow Reservation

Central Controller



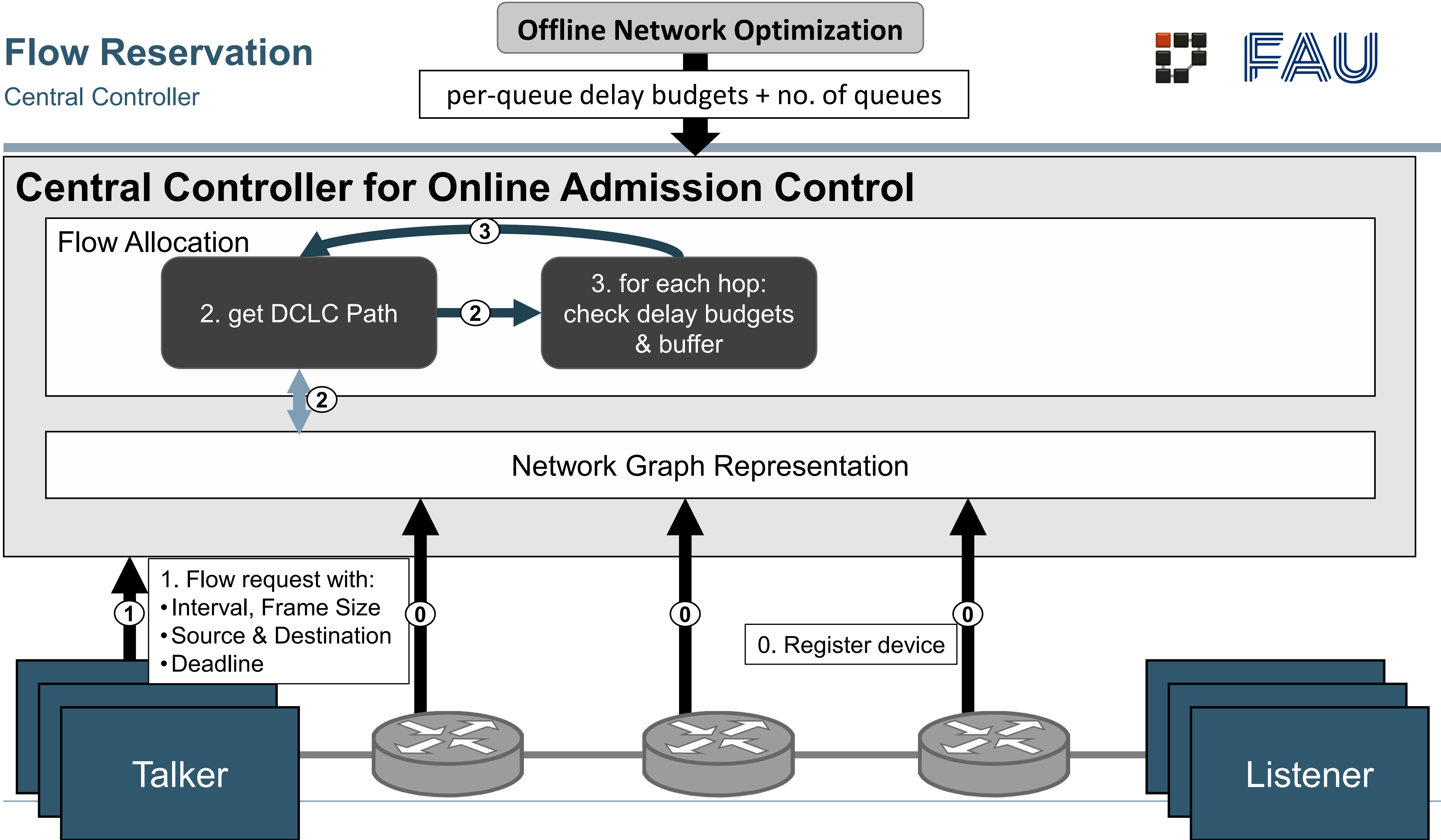
Flow Reservation

Central Controller



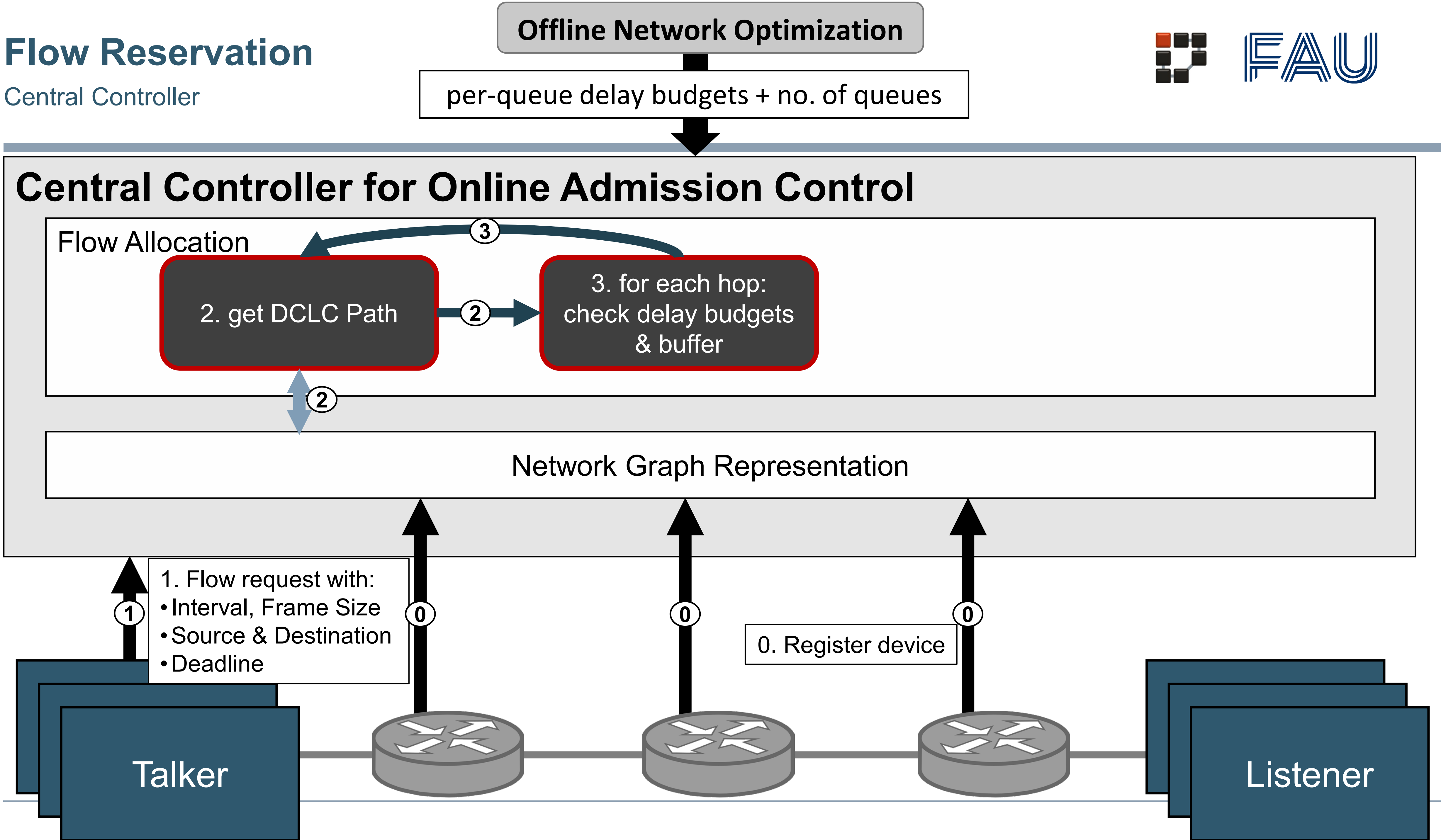
Flow Reservation

Central Controller



Flow Reservation

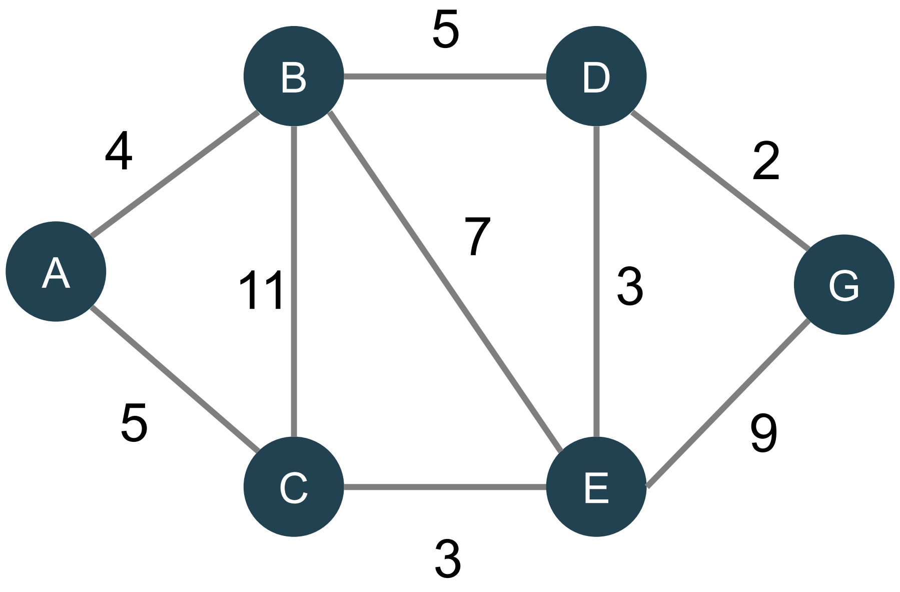
Central Controller



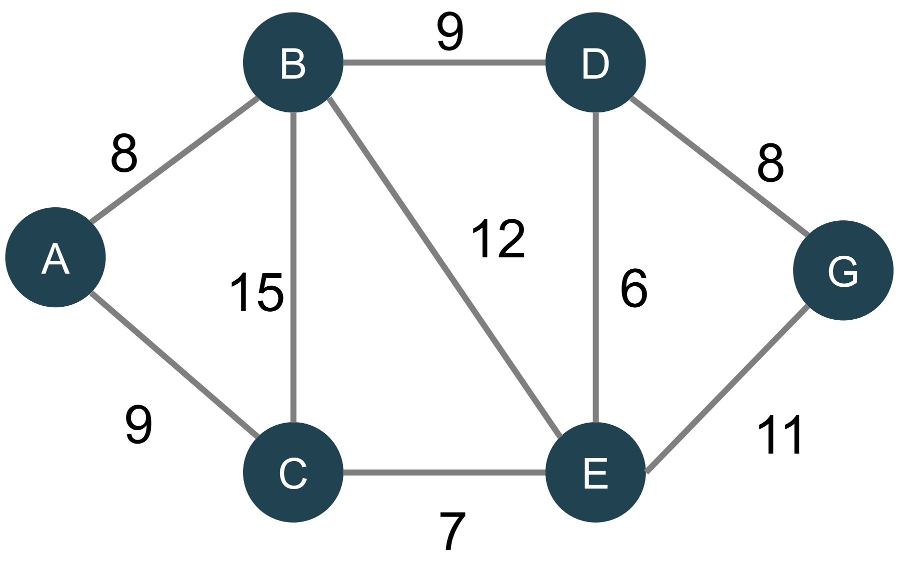
Flow Allocation

How to find the path(s)?

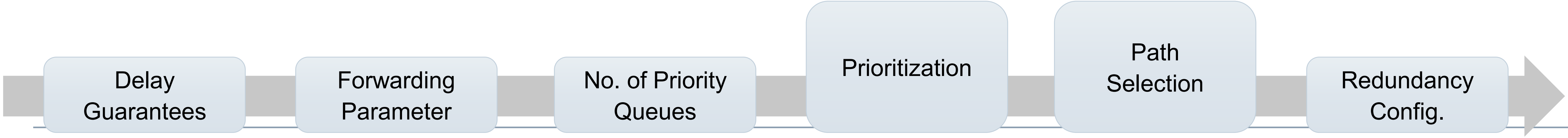
1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph



priority 7



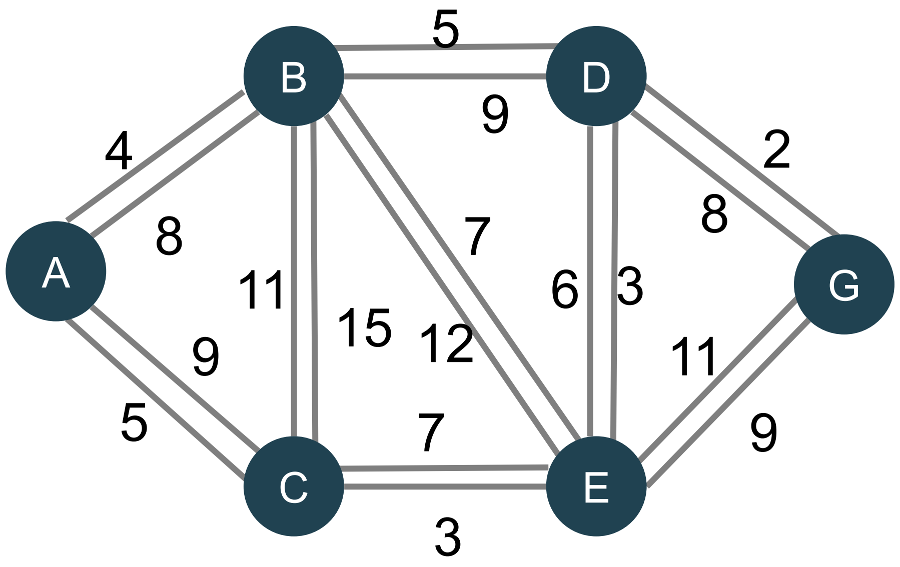
priority 6

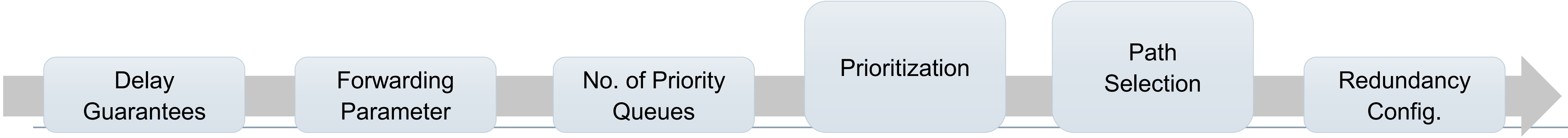


Flow Allocation

How to find the path(s)?

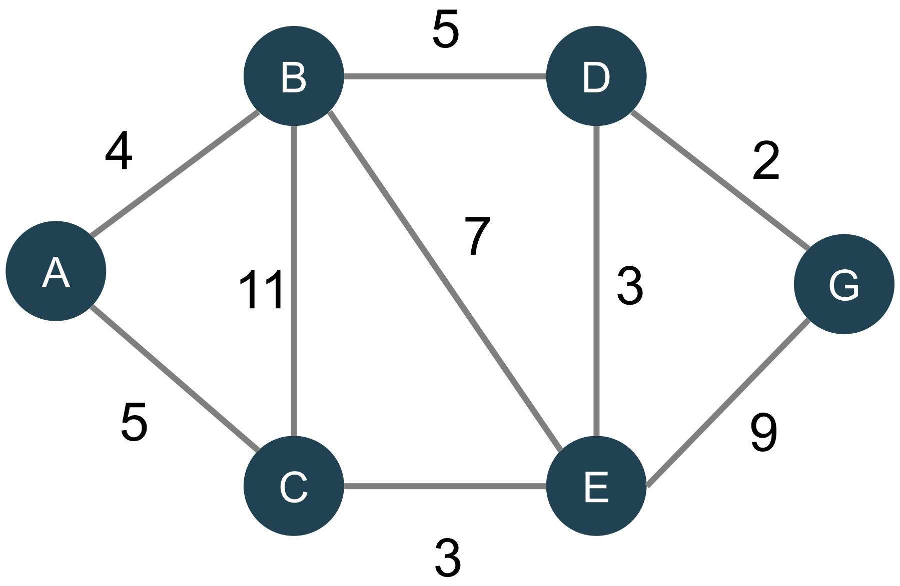
1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph **or Combined**



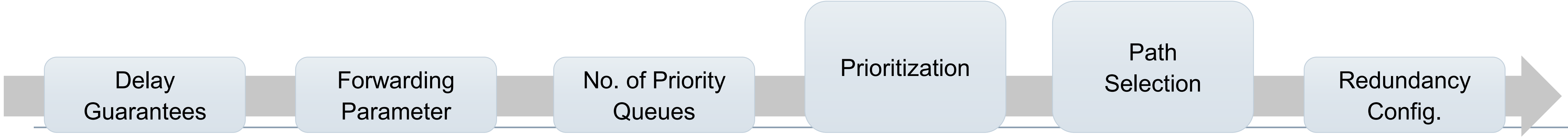


Flow Allocation

How to find the path(s)?

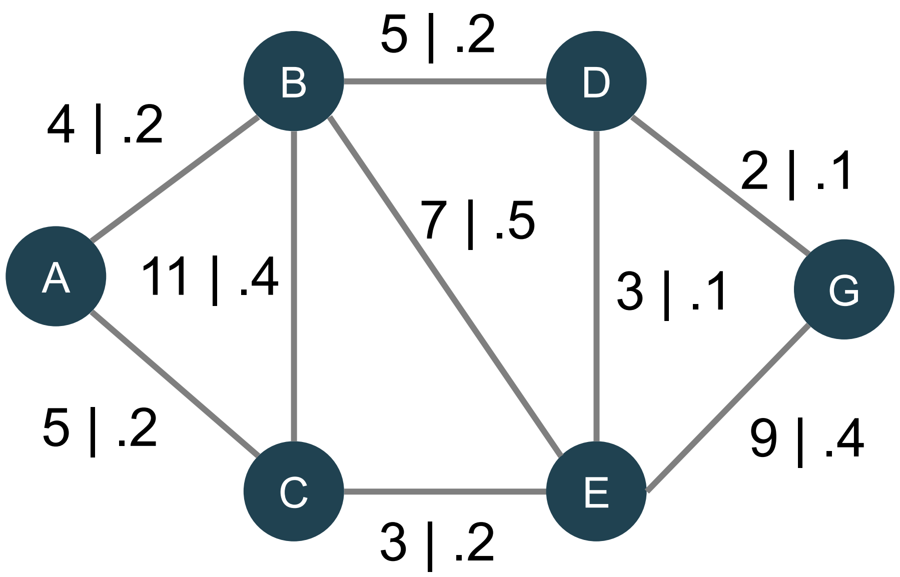


1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph or Combined

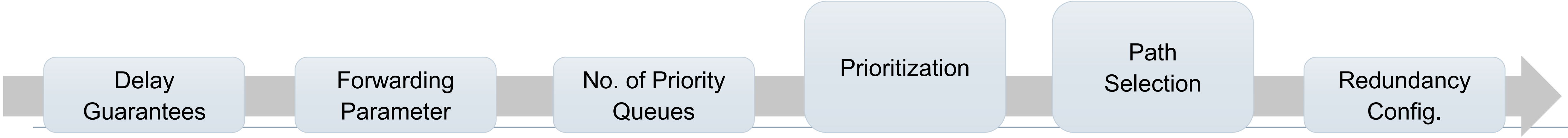


Flow Allocation

How to find the path(s)?

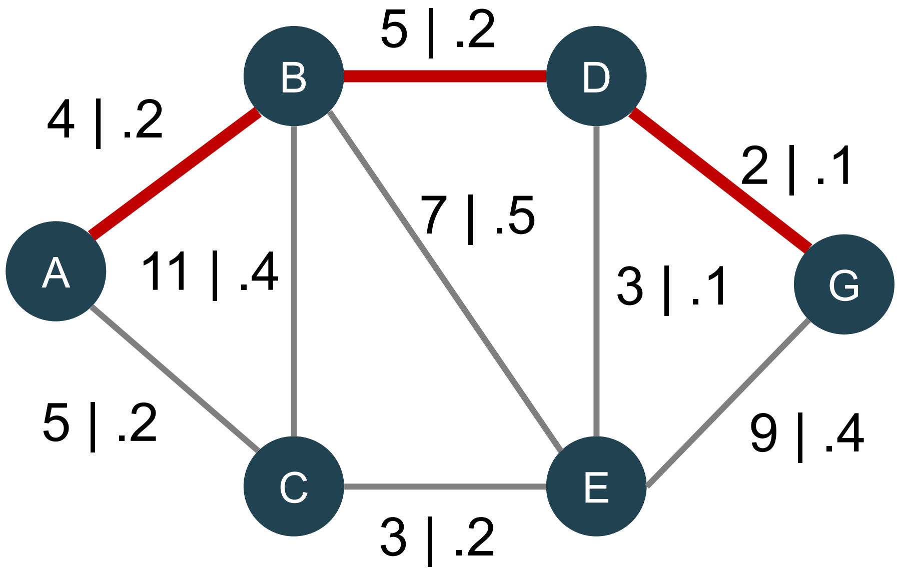


1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph or Combined
3. “Bandwidth” Weights to Balance the Network



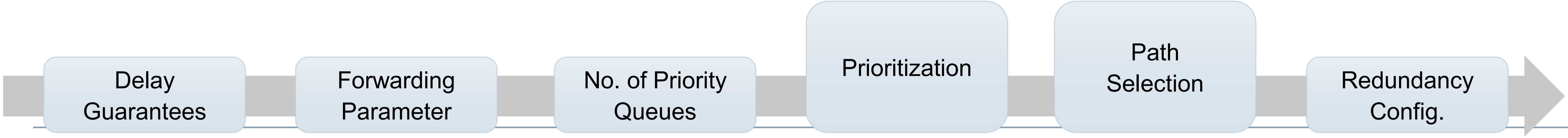
Flow Allocation

How to find the path(s)?



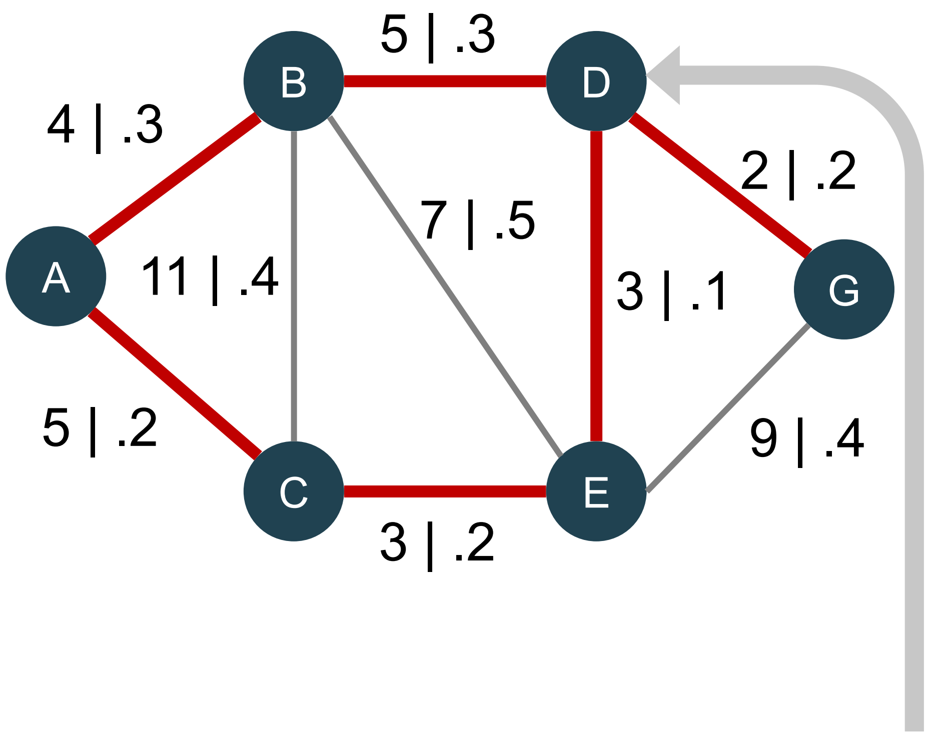
1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph or Combined
3. “Bandwidth” Weights to Balance the Network
4. Algorithm:

```
until #paths found:
    get delay-constrained least-cost (DCLC) path
    check access
```



Flow Allocation

How to find the path(s)?

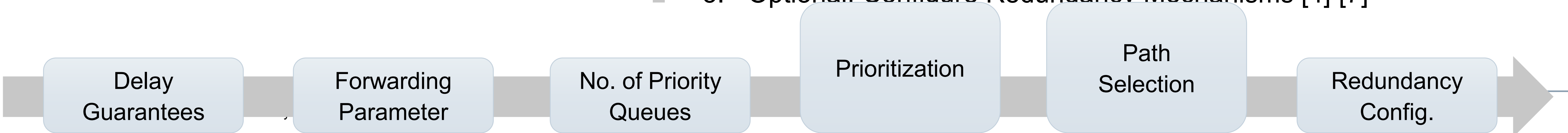


1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph or Combined
3. "Bandwidth" Weights to Balance the Network
4. Algorithm:

```
until #paths found:
    get delay-constrained least-cost (DCLC) path
    check access
```

by adapting disjoint shortest path routing algorithms [6] (e.g., Suurballe, …)

5. Optional: Configure Redundancy Mechanisms [4] [7]

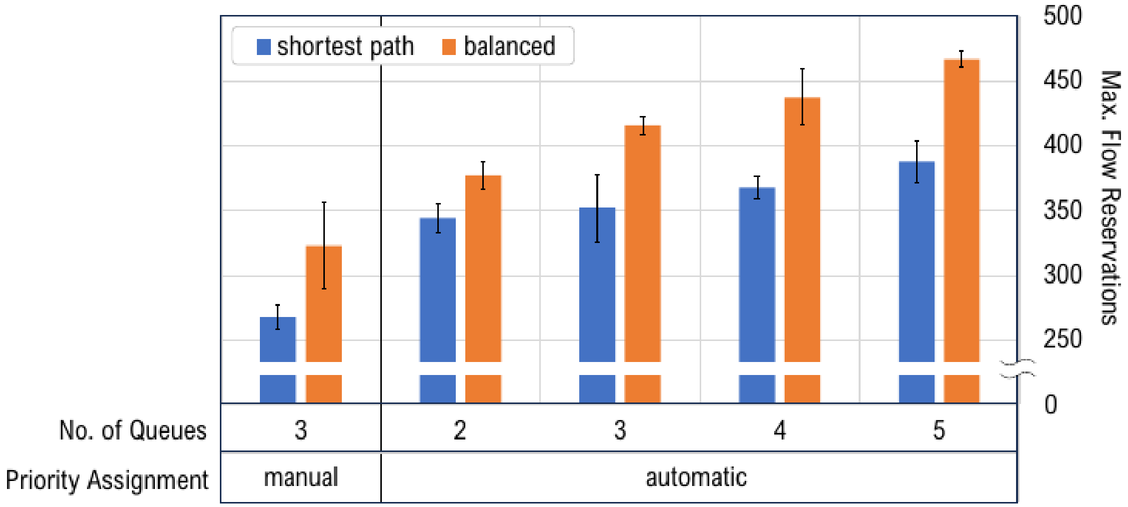


Flow Allocation

Information To-Go

Pre-defined paths and priorities (e.g., by users) are far from optimal. Heuristics can be more efficient.

How to find the path(s)?



1. Delay Budgets can be used for Routing Algorithms ☺
2. Multiple Priorities: Separate Graph or Combined
3. “Bandwidth” Weights to Balance the Network
4. Algorithm:

```
until #paths found:
    get delay-constrained least-cost (DCLC) path
    check access
```

by adapting disjoint shortest path routing algorithms [6] (e.g., Suurballe, …)

5. Optional: Configure Redundancy Mechanisms [4] [7]

Delay Guarantees → Forwarding Parameter → No. of Priority Queues → Prioritization → Path Selection → Redundancy Config.

Flow Reservation

Routing Decision

delay-constrained least-cost (DCLC) routing algorithm, with bandwidth utilization as costs

balances the network load / prevents bottlenecks

no flow-to-priority mapping required as input

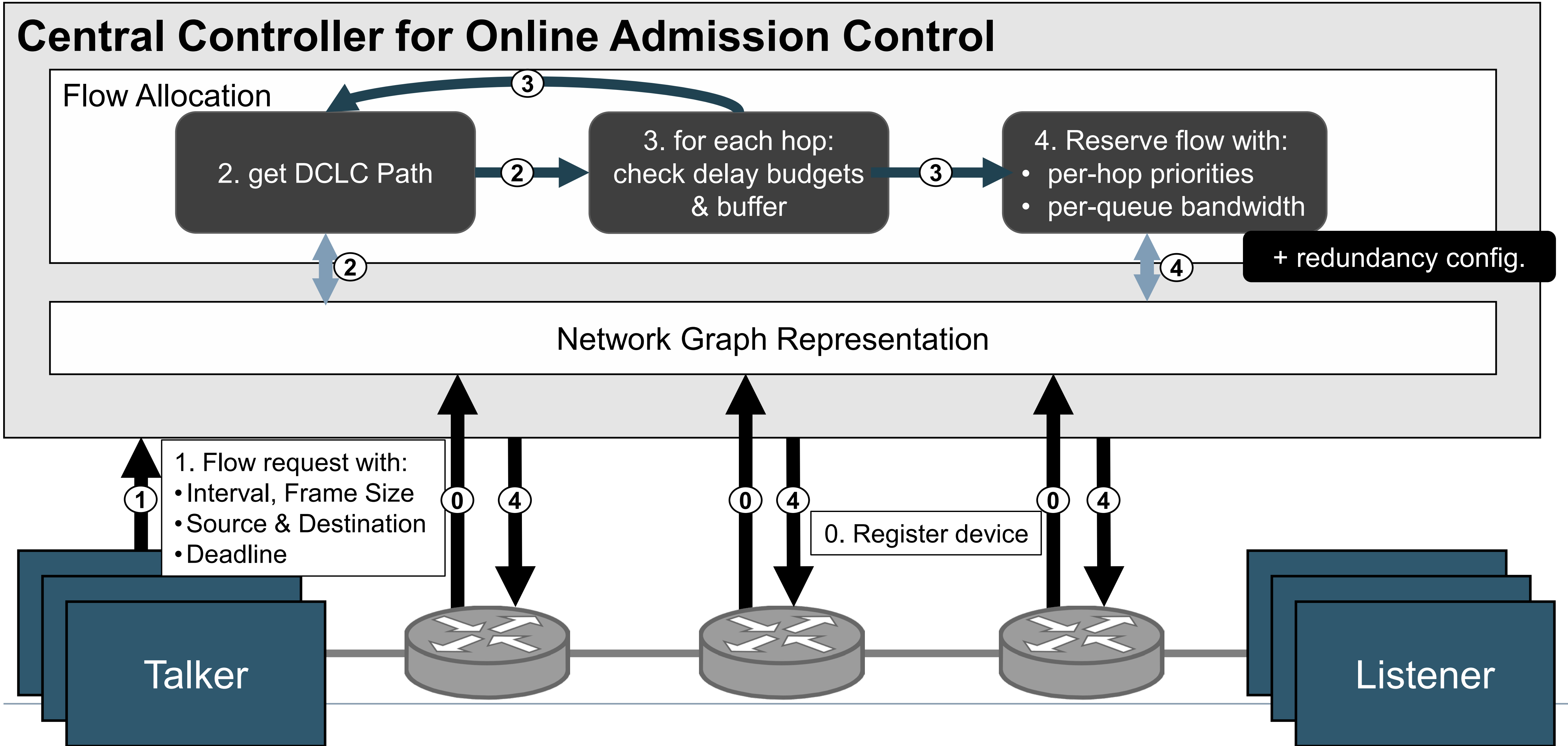
no guarantee on optimality (optimization problems themselves)

future flows unknown

still, highly practical networks with few user input / assumptions on future flows

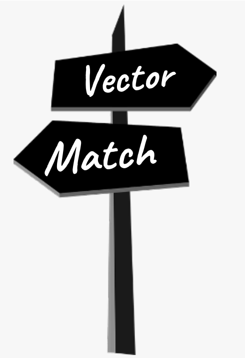
Flow Reservation

Central Controller



Redundancy Configuration

Challenges: Incorrect configuration of parameters for elimination in IEEE Std 802.1CB can result in valid frames to be discarded entirely, passing of duplicates, and unexpected bursts.
Too high and too low values can jeopardize the reliability of FRER [Maile2022].



Match Recovery Algorithm (MRA):

only applicable to **intermittent streams,** otherwise MRA passes duplicates
→ missing support for intermittent stream identification

Reset Timer:
SequenceRecoveryResetMSec

too low: unnecessary resets [Maile2022] & duplicates passed [Varga2023]

too high: discards (non-duplicate) frames [Maile2022]

Vector Recovery Algorithm (VRA) – History Length: *frerSeqRcvyHistoryLength*

too short: discards (new) frames [Hofmann2020]

too high: increased processing time can result in frame loss [Rana2023], $O(n)$ with n window size

Burst & Peak Rate Increase:

delay increase for flow [Thomas2022] and for interfering flows [Hofmann2020]
& buffer must be increased [Hofmann2020]

Delay Guarantees → Forwarding Parameter → No. of Priority Queues → Prioritization → Path Selection → Redundancy Config.

Redundancy Configuration

Stream Characteristics:

- Sending Interval (e.g., CMI)
- Maximum Interval Frames (MIF)
- Maximum Frame Size (MFS)

> CMI is used as variable name,
> it represents **arbitrary** sending intervals,
> possibly **individual** for each stream

A stream sends at most MIF packets during an interval of length CMI.
Each packet is smaller or equal to MFS.

Information To-Go

Redundancy in TSN is save, if configured with the given formulas. Only require TSpec and Δd.

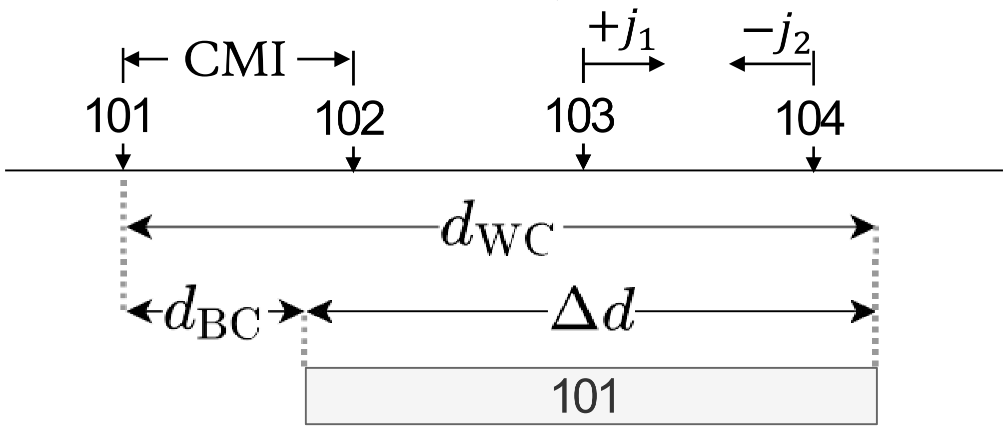
> introduction of optional **jitter** term,
> if the frames are not guaranteed to be separated by full interval (e.g., due to clock inaccuracy)
> with $j_1 + j_2 = J \leq CMI$

Network Characteristics:

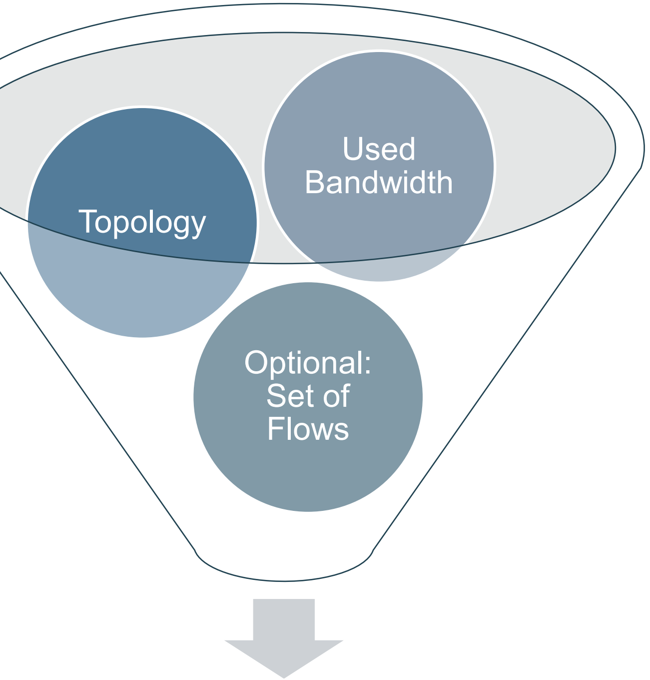
- lowest delay of fastest path d_{BC} (best-case)
- highest delay of slowest path d_{WC} (worst-case)
- reception window: $\Delta d = d_{WC} - d_{BC}$

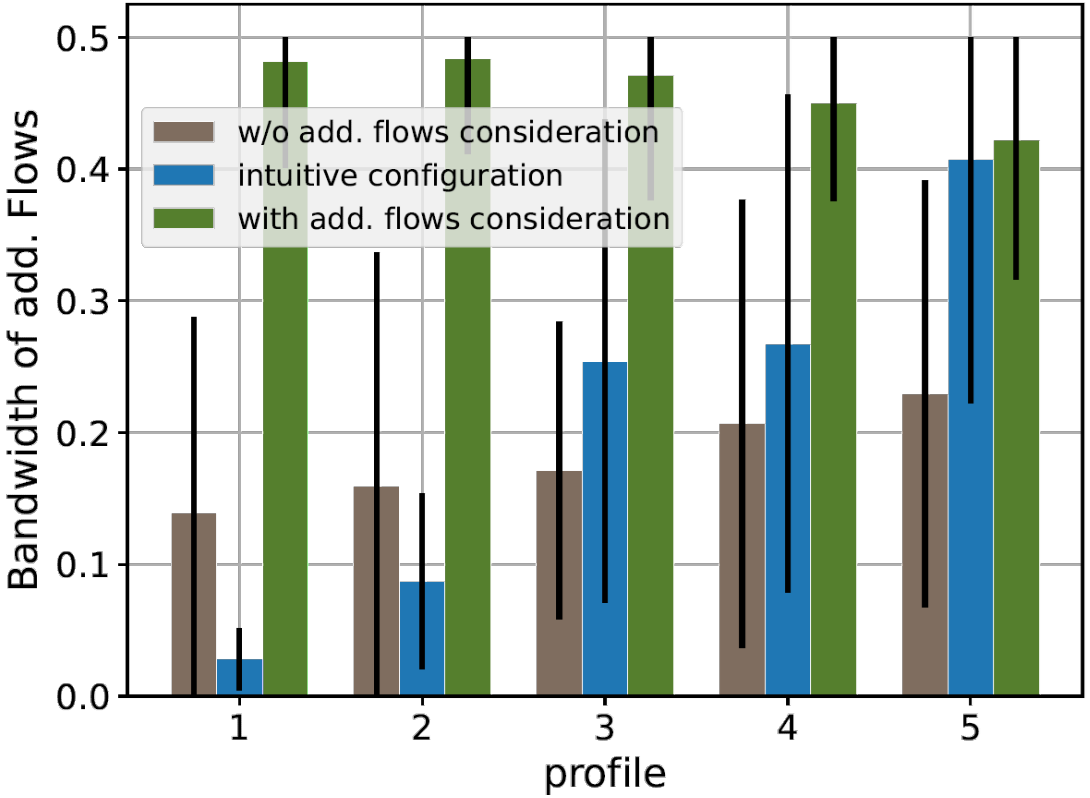
Illustration:

Send packets



Result





Delay Guarantees → Forwarding Parameter → No. of Priority Queues → Prioritization → Path Selection → Redundancy Config.

Conclusion

Conclusion

- Ultra-reliable flows with save guarantees and redundant transmission
- Only minimal user input
- Combining offline and online configuration for TSN networks
- Allows for efficient networks in dynamic scenarios
- Future work: Heterogenous networks

References

Central Flow Reservation / Check Access:

[1] L. Maile, K.-S. J. Hielscher, and R. German, “Delay-Guaranteeing Admission Control for Time-Sensitive Networking Using the Credit-Based Shaper,” *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1834–1852, 2022, doi: 10.1109/OJCOMS.2022.3212939.

Decentral Flow Reservation / Check Access:

[2] L. Maile, D. Voitlein, A. Grigorjew, K.-S. J. Hielscher, and R. German, “On the Validity of Credit-Based Shaper Delay Guarantees in Decentralized Reservation Protocols,” in *Proceedings of the 31st International Conference on Real-Time Networks and Systems*, in RTNS ’23. New York, NY, USA: Association for Computing Machinery, Jun. 2023, pp. 108–118. doi: 10.1145/3575757.3593644.

Delay Budgets Choice / Offline + Online Optimization

[3] L. Maile, K.-S. J. Hielscher, and R. German, "Combining Static and Dynamic Traffic with Delay Guarantees in Time-Sensitive Networking,” in *Proceedings of the 16th EAI International Conference on Performance Evaluation Methodologies and Tools*, in ValueTools’23. Crete, Grece. Forthcoming.

Redundant Configuration:

[4] L. Maile, D. Voitlein, K.-S. Hielscher, and R. German, “Ensuring Reliable and Predictable Behavior of IEEE 802.1CB Frame Replication and Elimination,” in *ICC 2022 - IEEE International Conference on Communications*, May 2022, pp. 2706–2712. doi: 10.1109/ICC45855.2022.9838905.

Network Calculus / Delay Analysis for TSN:

[5] L. Maile, K.-S. Hielscher, and R. German, “Network Calculus Results for TSN: An Introduction,” in *2020 Information Communication Technologies Conference (ICTC)*, Nanjing, China: IEEE, May 2020, pp. 131–140. doi: 10.1109/ICTC49638.2020.9123308.

(Disjoint) Routing:

[6] P. Navade, L. Maile, and R. German, “Multiple DCLC Routing Algorithms for Ultra-Reliable and Time-Sensitive Applications”, KuVS Fachgespräch - Würzburg Workshop on Modeling, Analysis and Simulation of Next-Generation Communication Networks 2023 (WueWoWAS’23). doi: 10.25972/OPUS-32217.

Delay Analysis for Redundant Transmission:

[7] L. Thomas, A. Mifdaoui, and J.-Y. L. Boudec, “Worst-Case Delay Bounds in Time-Sensitive Networks With Packet Replication and Elimination,” *IEEE/ACM Transactions on Networking*, pp. 1–15, 2022, doi: 10.1109/TNET.2022.3180763.

Thank you!