Hybrid Circuit/Packet Network Scheduling with Multiple Composite Paths

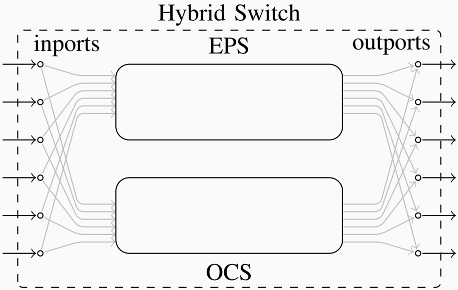
Shih-Hao Tseng[1], (pronounced as "She-How Zen")
joint work with Bo Bai[2] and John C. S. Lui[3]

April 18, 2018

[1]School of Electrical and Computer Engineering, Cornell University
[2]Future Network Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd.
[3]Department of Computer Science and Engineering, The Chinese University of Hong Kong

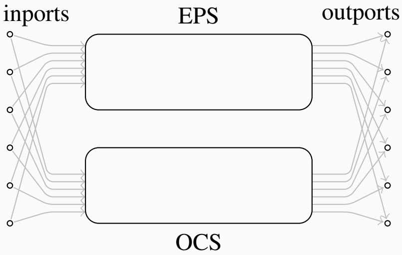
Hybrid Switches

- A hybrid switch (h-switch) combines an electronic packet switch (EPS) and an optical circuit switch (OCS).



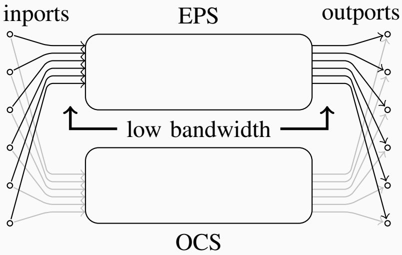
Hybrid Switches

- A hybrid switch (h-switch) combines an electronic packet switch (EPS) and an optical circuit switch (OCS).
- EPS can switch among many-to-many routing patterns swiftly; and OCS provides high-bandwidth one-to-one routing.



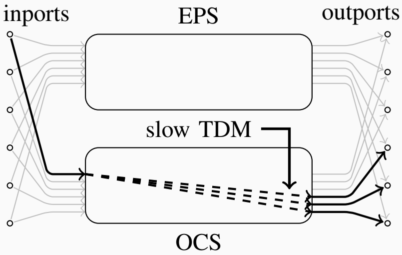
Drawbacks of Hybrid Switches

- EPS suffers low bandwidth;



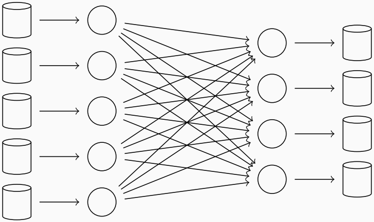
Drawbacks of Hybrid Switches

- EPS suffers low bandwidth; and OCS suffers slow time division multiplexing (TDM) when mapping many-to-one or one-to-many.



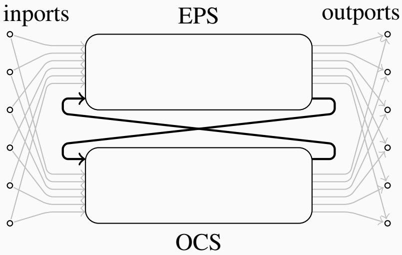
Drawbacks of Hybrid Switches

- EPS suffers low bandwidth; and OCS suffers slow time division multiplexing (TDM) when mapping many-to-one or one-to-many.
- The drawbacks restrict the use of hybrid switches in data-parallel applications, such as MapReduce.



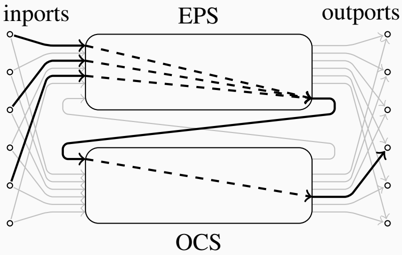
Composite Paths

- Since EPS nowadays supports heterogeneous port bandwidth, with many low-bandwidth ports and few high bandwidth ports, one can connect an OCS outport to an EPS inport (and vice versa) to create a *composite path* (Vargaftik et al., 2016).



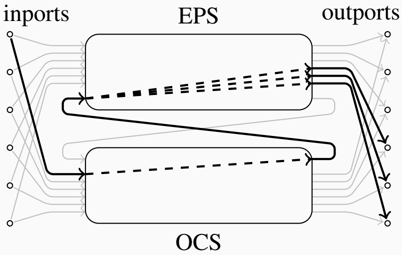
Advantages of Composite Paths

- Composite paths allow EPS to send more data to the outports under many-to-one mapping.



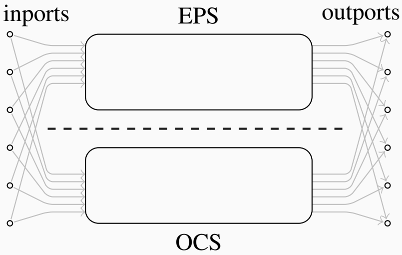
Advantages of Composite Paths

- Composite paths allow EPS to send more data to the outports under many-to-one mapping.
- Composite paths avoid OCS TDM but still provide higher input bandwidth for one-to-many scenarios.



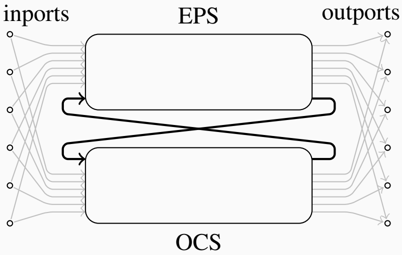
Challenges of Composite-Path Scheduling

- Without composite paths, EPS and OCS can be scheduled in parallel (h-switch).



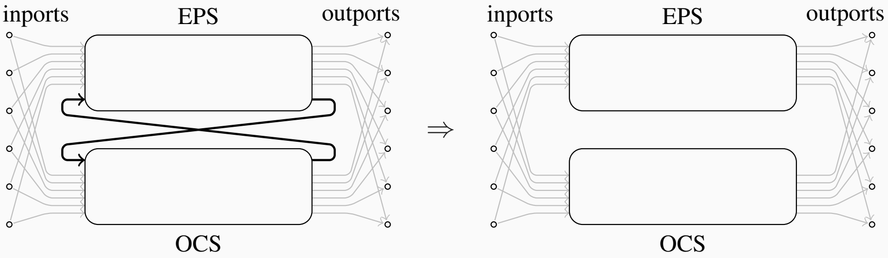
Challenges of Composite-Path Scheduling

- Without composite paths, EPS and OCS can be scheduled in parallel (h-switch).
- However, with composite paths, EPS and OCS are tangled together (cp-switch).



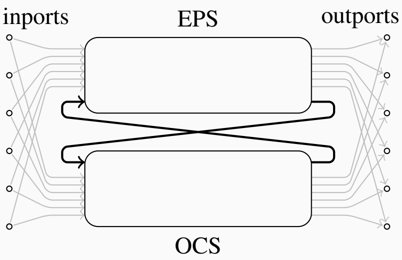
Demand Reduction

- Vargaftik et al. (2016) suggested some heuristics to translate the demand matrix so that we can schedule cp-switches using h-switch scheduling algorithms.



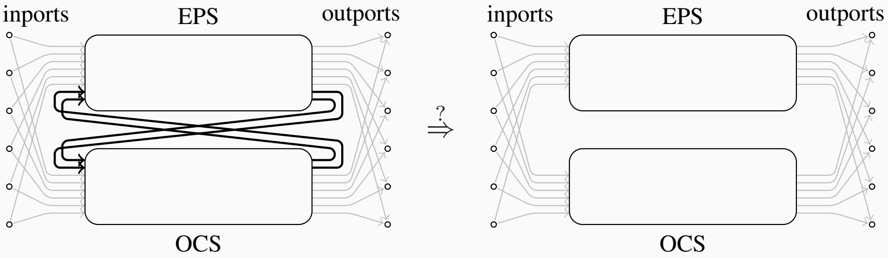
Unresolved Issues

- The translation based algorithm does not provide a theoretical performance guarantee.



Unresolved Issues

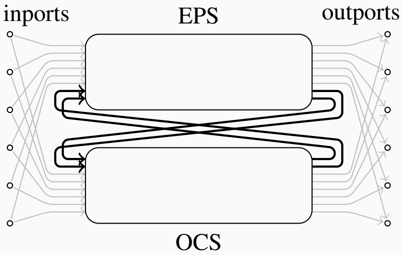
- The translation based algorithm does not provide a theoretical performance guarantee.
- The algorithm only works for one pair of composite paths. In general, we can have more composite paths at the system level.



Composite-Path Switch Scheduling

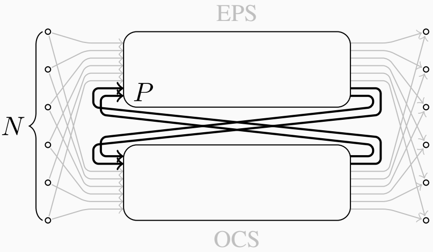
Goal: finding a shortest schedule to satisfy i/o demand.

- Systematic analysis of cp-switch schedules.
- Performance guarantee for cp-switch scheduling algorithms.
- Applicability to multiple composite paths.



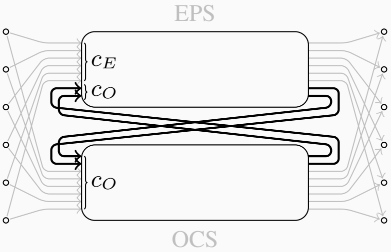
Formulation

- N ports. Each port $n \in [1, N]_{\mathbb{Z}} = \{1, \ldots, N\}$ connects to both EPS and OCS.
- P composite paths (each composite path is a full-duplex line connecting EPS and OCS).



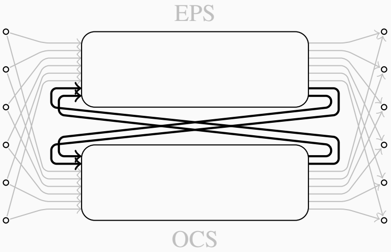
Formulation

- Each port is assumed to have symmetric input/output capacity (bandwidth).
- Each EPS port has capacity c_E; Each OCS port has capacity c_O; And each composite path $p \in P$ is assumed to have capacity c_O as well. In general, $c_O \sim 10c_E \gg c_E$.



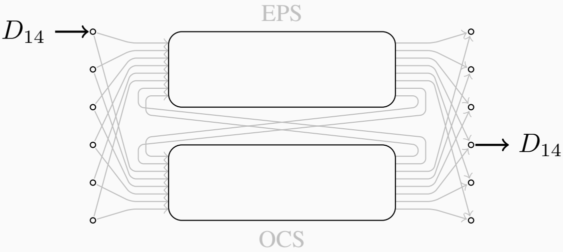
Formulation

- Each OCS inport maps to at most one OCS outport, and each OCS outport can receive data from at most one OCS inport. Such mapping is called an *OCS configuration*.
- No data can be buffered at the inports or the outports of the composite paths.



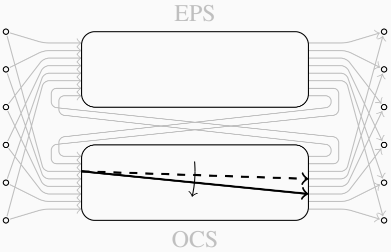
Formulation

- Each entry D_{ij} in the demand matrix $D \in \mathbb{R}^{N \times N}$ refers to the amount of data that should be sent from port i to port j.
- By convention, we assume $D_{nn} = 0$ for each $n \in [1, N]_{\mathbb{Z}}$.



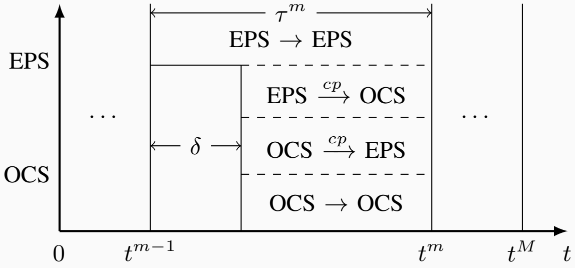
Formulation

- The reconfiguration time of OCS is δ. During the reconfiguration, OCS stops carrying data. In contrast, EPS changes the sending rate seamlessly.



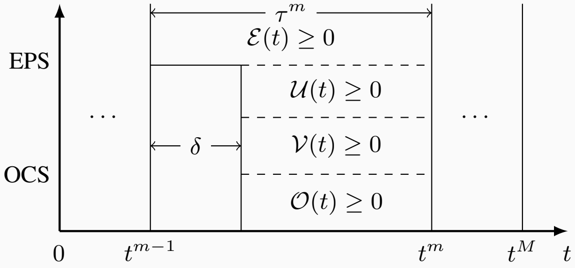
Formulation

- A $M+1$ step schedule is considered: In step 0, only EPS is used; The remaining M steps involve the whole cp-switch.
- Each step $m \in [1, M]_{\mathbb{Z}}$ consists of a reconfiguration phase and a sending phase. The length of the step m is τ^m.



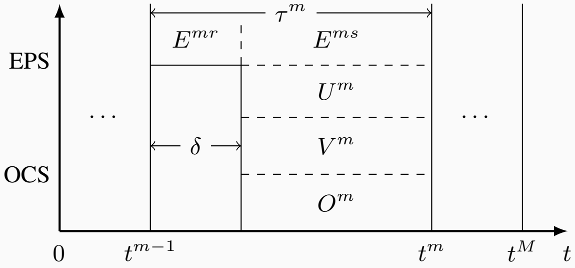
Continuous-Time Control Formulation

- Let $\mathcal{E}(t), \mathcal{U}(t), \mathcal{V}(t), \mathcal{O}(t) \in \mathbb{R}^{N\times N}$ be the *mapping matrices*, which represent the port-to-port sending rates, at time t.
- We can formulate the scheduling problem as a continuous-time control problem.



Mixed Integer Linear Programming (MILP) Formulation

- It turns out the continuous-time control problem can be equivalently transformed into an MILP.
- Instead of the continuous-time mapping matrices, we express the problem in terms of the total data sent during each phase.



Mixed Integer Linear Programming (MILP) Formulation

- For each $\hat{M} \in [0, M]_{\mathbb{Z}}$, solving the following subproblems:

$$\begin{aligned} I(\hat{M}) = \min \quad & \text{length of } \left(\hat{M}+1\right)\text{-step schedule} \\ \text{s.t.} \quad & \text{parameter setup} \\ & \text{demand constraints} \\ & \text{capacity constraints} \\ & \text{operation constraints} \end{aligned}$$

leads to the shortest schedule $\mathrm{OPT} = \min_{\hat{M} \in [0,M]_{\mathbb{Z}}} I(\hat{M})$.

Mixed Integer Linear Programming (MILP) Formulation

- For each $\hat{M} \in [0, M]_{\mathbb{Z}}$, solving the following subproblems:

$$
\begin{aligned}
I(\hat{M}) = \min \quad & \sum_{m=0}^{\hat{M}} \tau^m \\
\text{s.t.} \quad & \text{parameter setup} \\
& \\
& \text{demand constraints} \\
& \\
& \text{capacity constraints} \\
& \text{operation constraints}
\end{aligned}
$$

leads to the shortest schedule $\mathrm{OPT} = \min_{\hat{M} \in [0,M]_{\mathbb{Z}}} I(\hat{M})$.

Mixed Integer Linear Programming (MILP) Formulation

- For each $\hat{M} \in [0, M]_{\mathbb{Z}}$, solving the following subproblems:

$$I(\hat{M}) = \min \sum_{m=0}^{\hat{M}} \tau^m$$
$$\text{s.t. parameter setup}$$
$$E^0 + \sum_{m=1}^{\hat{M}} E^m + U^m + V^m + O^m = D$$
$$\text{capacity constraints}$$
$$\text{operation constraints}$$

leads to the shortest schedule $\text{OPT} = \min_{\hat{M} \in [0,M]_{\mathbb{Z}}} I(\hat{M})$.

Mixed Integer Linear Programming (MILP) Formulation

- For each $\hat{M} \in [0, M]_{\mathbb{Z}}$, solving the following subproblems:

$$
\begin{aligned}
I(\hat{M}) = \min \quad & \sum_{m=0}^{\hat{M}} \tau^m \\
\text{s.t.} \quad & \text{parameter setup} \\
& E^0 + \sum_{m=1}^{\hat{M}} E^m + U^m + V^m + O^m = D \\
& \text{capacity constraints} \\
& \text{OCS must map one-to-one}
\end{aligned}
$$

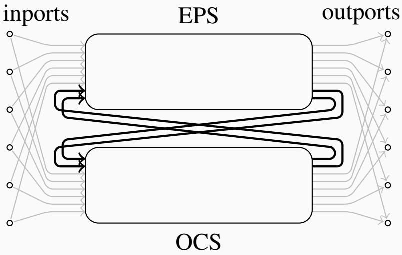
leads to the shortest schedule $\mathrm{OPT} = \min_{\hat{M} \in [0,M]_{\mathbb{Z}}} I(\hat{M})$.

Composite-Path Switch Scheduling

Goal: finding a shortest schedule to satisfy i/o demand.

✓ Systematic analysis of cp-switch schedules.

- Performance guarantee for cp-switch scheduling algorithms.
- Applicability to multiple composite paths.



NP-Hardness

- Unfortunately, $I(\hat{M})$ is NP-hard.

$$
\begin{aligned}
I(\hat{M}) = \min \quad & \sum_{m=0}^{\hat{M}} \tau^m \\
\text{s.t.} \quad & \text{parameter setup} \\
& E^0 + \sum_{m=1}^{\hat{M}} E^m + U^m + V^m + O^m = D \\
& \text{capacity constraints} \\
& \underline{\text{OCS must map one-to-one}}
\end{aligned}
$$

$\rightarrow$ NP-hardness

Linear Relaxation

- We then linear-relax $I(\hat{M})$ to be $L(\hat{M})$:

$$
\begin{aligned}
L(\hat{M}) = \min \quad & \sum_{m=0}^{\hat{M}} \tau^m \\
\text{s.t.} \quad & \text{parameter setup} \\
& E^0 + \sum_{m=1}^{\hat{M}} E^m + U^m + V^m + O^m = D \\
& \text{capacity constraints} \\
& \text{OCS can map many-to-many}
\end{aligned}
$$

Linear Relaxation: Special Case

- The schedule that uses EPS only :

$$
\begin{aligned}
L(0) = \min \quad & \tau^0 \\
\text{s.t.} \quad & \text{parameter setup} \\
& E^0 = D \\
& \text{capacity constraints}
\end{aligned}
$$

- Especially, $L(0) = I(0)$.

Approximation Ratio

Lemma 1

Any cp-switch scheduling algorithm adopting $L(0)$ as an upper bound is a $\frac{c_E+c_O}{c_E}$-approximation algorithm.

- Lemma 1 implies that a scheduling algorithm that produces shorter schedule than EPS only schedule is an approximation algorithm with approximation ratio $\frac{c_E+c_O}{c_E}$.
  $\Rightarrow$ Comparing with $L(0)$ is a naive way to have performance guarantee.

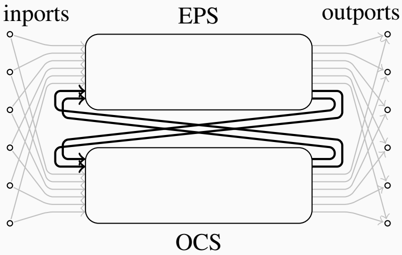
Composite-Path Switch Scheduling

Goal: finding a shortest schedule to satisfy i/o demand.

✓ Systematic analysis of cp-switch schedules.

✓ Performance guarantee for cp-switch scheduling algorithms.

- Applicability to multiple composite paths.



Lemma 2

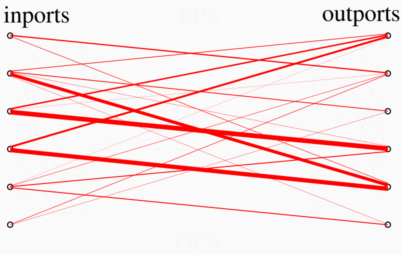
If $L(0) \leq \delta$, $\mathrm{OPT} = L(0)$ *and the shortest time schedule uses EPS only. Otherwise,*

$$L(0) \geq \mathrm{OPT} \geq L(1).$$

- Lemma 2 inspires us to find the shortest time schedule "between" $L(1)$ and $L(0)$.
  $\Rightarrow$ Since $L(1)$ is not feasible to $I(\hat{M})$, can we round it to a feasible schedule shorter than $L(0)$?

Uprounding Procedure

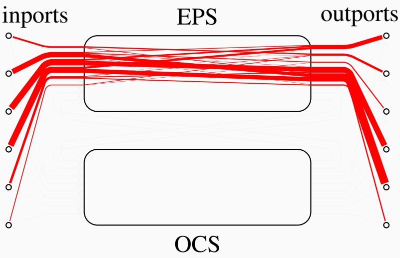
- Taking the demand D, we compute two schedules $L(0)$ and $L(1)$.



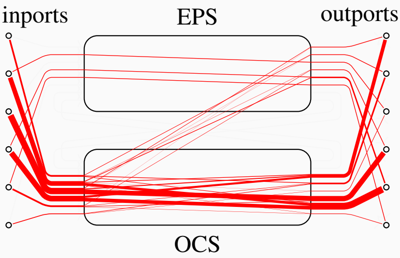
Demand matrix D

Uprounding Procedure

- Taking the demand D, we compute two schedules $L(0)$ and $L(1)$.



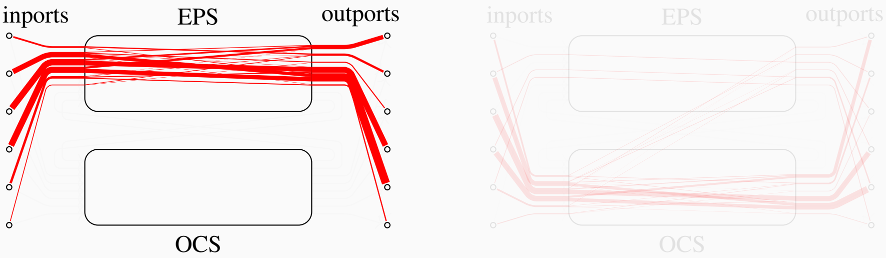
1-step linear-relaxed schedule $L(0)$



2-step linear-relaxed schedule $L(1)$

Uprounding Procedure

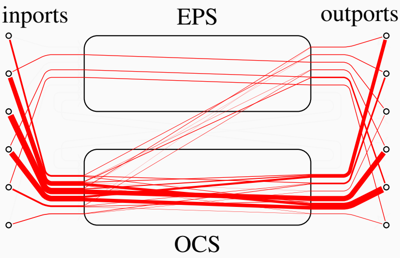
- Taking the demand D, we compute two schedules $L(0)$ and $L(1)$.
- If $L(0) \le \delta$, we have found the shortest schedule.



1-step linear-relaxed schedule $L(0)$

Uprounding Procedure

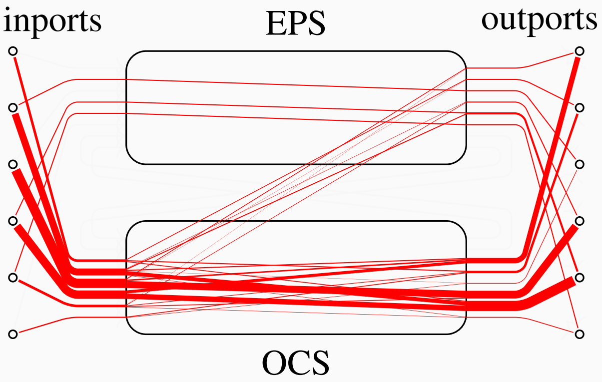
- Taking the demand D, we compute two schedules $L(0)$ and $L(1)$.
- If $L(0) \leq \delta$, we have found the shortest schedule.
- Otherwise, we upround $L(1)$ to a feasible schedule.



2-step linear-relaxed schedule $L(1)$

Uprounding Procedure

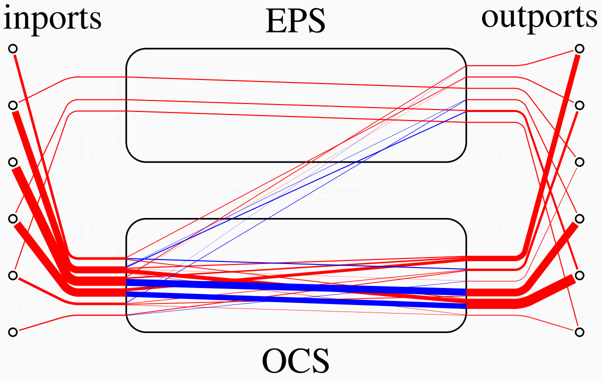
- Find OCS configuration that can send as much relaxed traffic as possible.



Uprounding Procedure

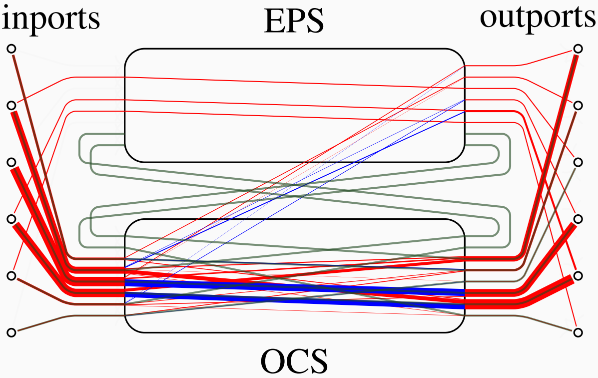
- Find OCS configuration that can send as much relaxed traffic as possible.

  $\Rightarrow$ Use maximum weight matching algorithm.



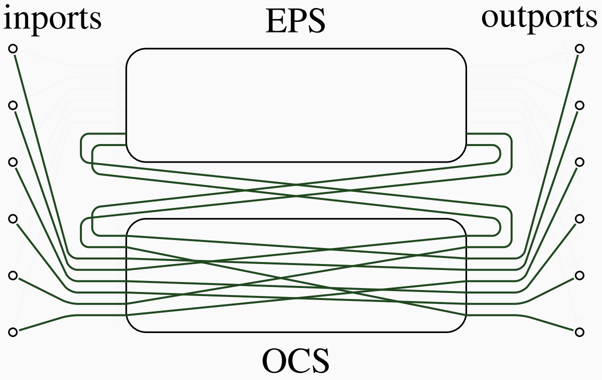
Uprounding Procedure

- The OCS Configuration that supports the most relaxed traffic.



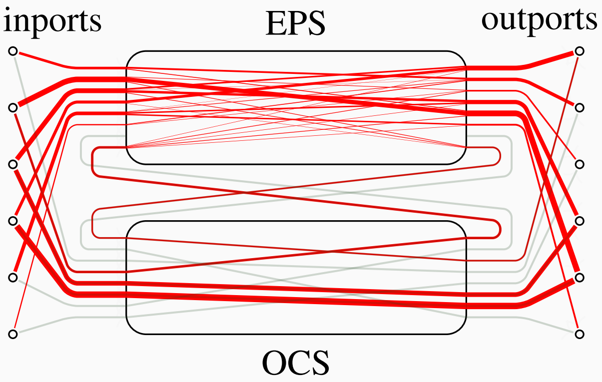
Uprounding Procedure

- The OCS Configuration that supports the most relaxed traffic.



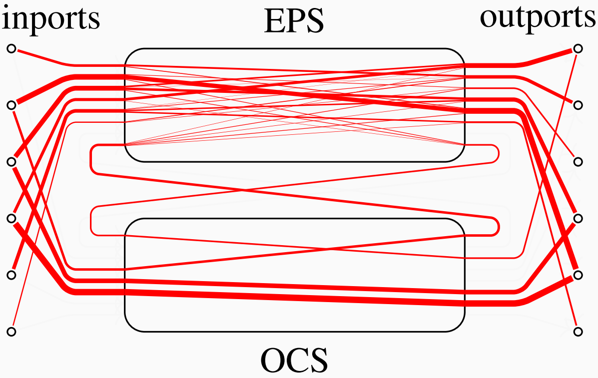
Uprounding Procedure

- Once the OCS configuration is decided, the shortest 2-step schedule can be obtained by a linear program.



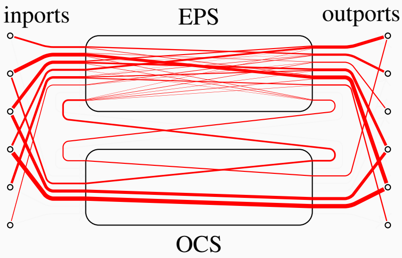
Uprounding Procedure

- Once the OCS configuration is decided, the shortest 2-step schedule can be obtained by a linear program.



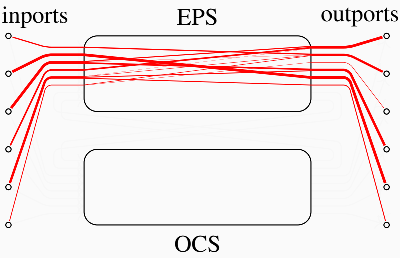
Uprounding Procedure

- Once the OCS configuration is decided, the shortest 2-step schedule can be obtained by a linear program.

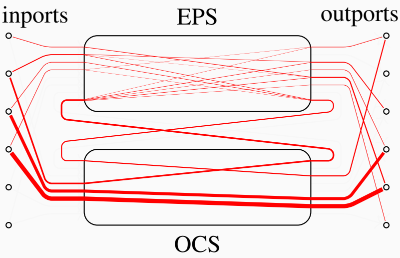


Uprounding Procedure

- Once the OCS configuration is decided, the shortest 2-step schedule can be obtained by a linear program.



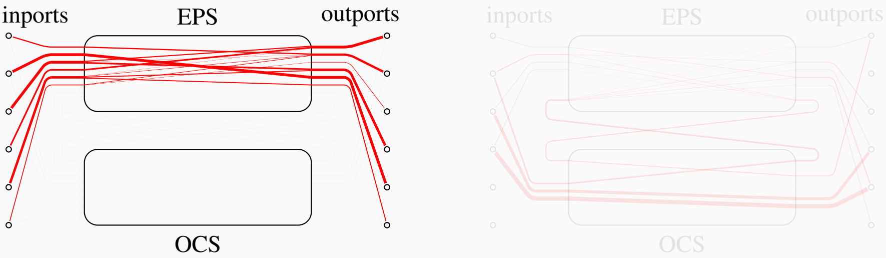
Step 0



Step 1

Uprounding Procedure

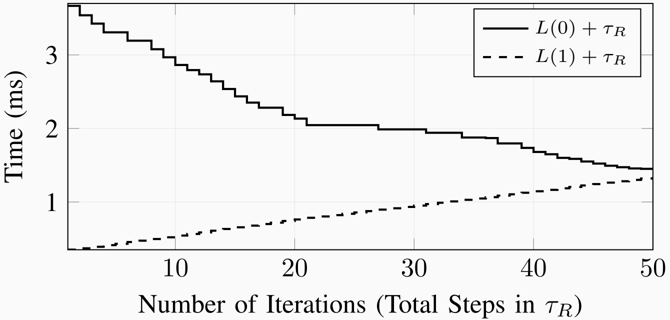
- Once the OCS configuration is decided, the shortest 2-step schedule can be obtained by a linear program.
  $\Rightarrow$ The algorithm can work online by repeating the same procedure on the resulting step 0.



Step 0

Bound Shrinking

- Let τ_R be the current length of the schedule without step 0.
- Upper bound: $L(0) + \tau_R$.
- Lower bound: $L(1) + \tau_R$.



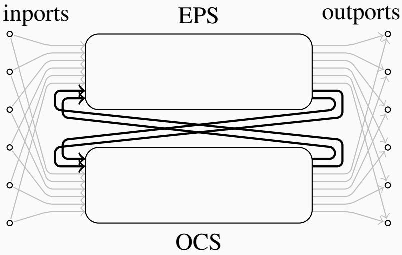
Composite-Path Switch Scheduling

Goal: finding a shortest schedule to satisfy i/o demand.

✓ Systematic analysis of cp-switch schedules.

✓ Performance guarantee for cp-switch scheduling algorithms.

✓ Applicability to multiple composite paths.

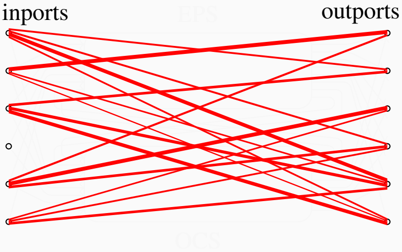


Issues to be Evaluated

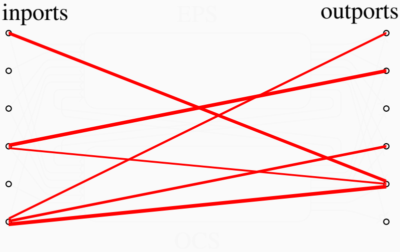
- Do more composite paths lead to shorter schedule?
- How well do the algorithms schedule one-to-many traffic?
- How does OCS reconfiguration time δ influence the length of the schedule?

Benefits of Multiple Composite Paths

- Meshed multicast traffic: Pick each inport-outport pair with probability 0.5.
- Skewed multicast traffic: Pick each source with probability 0.5 to install multicast traffic.

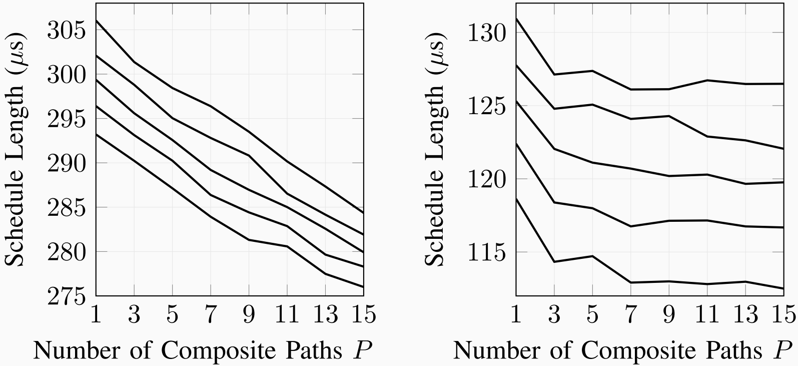


Meshed multicast traffic.



Skewed multicast traffic.

Benefits of Multiple Composite Paths



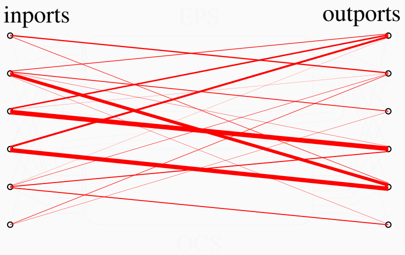
(a) Meshed multicast traffic.

(b) Skewed multicast traffic.

Figure 1: The 30^{th}, 40^{th}, 50^{th}, 60^{th}, and 70^{th} percentiles of the schedule length given by the proposed algorithm.

Performance Improvement under Skewed Demand

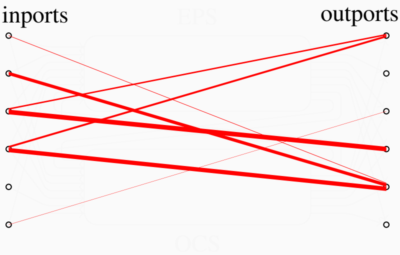
- We compare our algorithm with the state-of-the-art scheduling algorithm CPSwitchSched (Vargaftik et al., 2016).



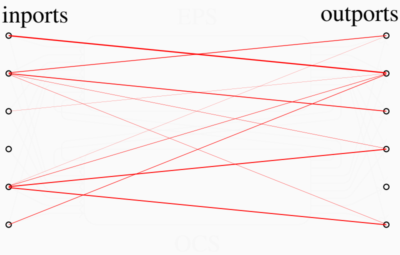
Demand matrix D

Performance Improvement under Skewed Demand

- We compare our algorithm with the state-of-the-art scheduling algorithm CPSwitchSched (Vargaftik et al., 2016).



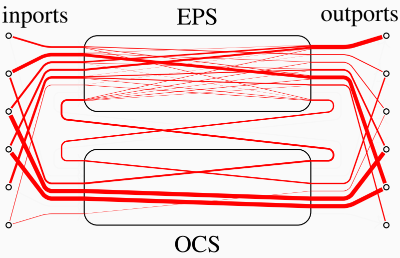
Traffic through EPS and OCS



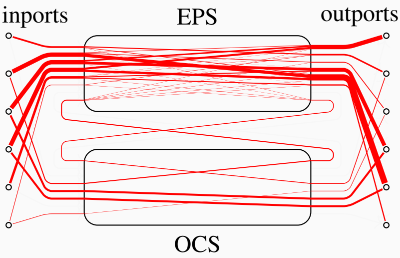
Traffic through the composite path

Performance Improvement under Skewed Demand

- We compare our algorithm with the state-of-the-art scheduling algorithm CPSwitchSched (Vargaftik et al., 2016).
    - Solstice (Liu et al., 2015).
    - Eclipse (Venkatakrishnan et al., 2016).



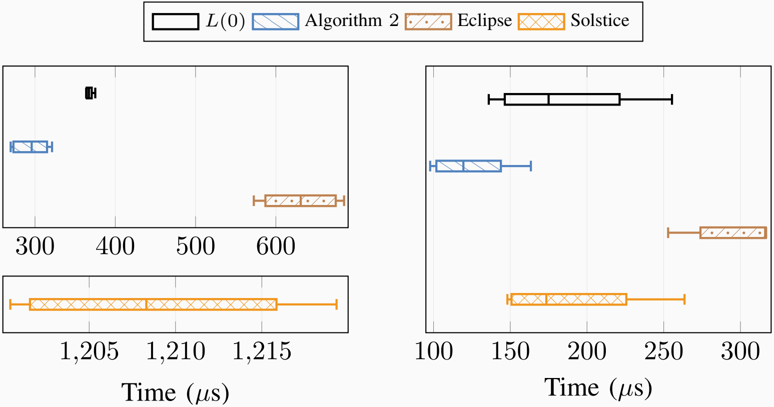
Solstice.



Eclipse.

H. Liu et al., “Scheduling Techniques for Hybrid Circuit/Packet Networks,” 2015.
S. B. Venkatakrishnan et al., “Costly Circuits, Submodular Schedules and Approximate Carathéodory Theorems,” 2016.

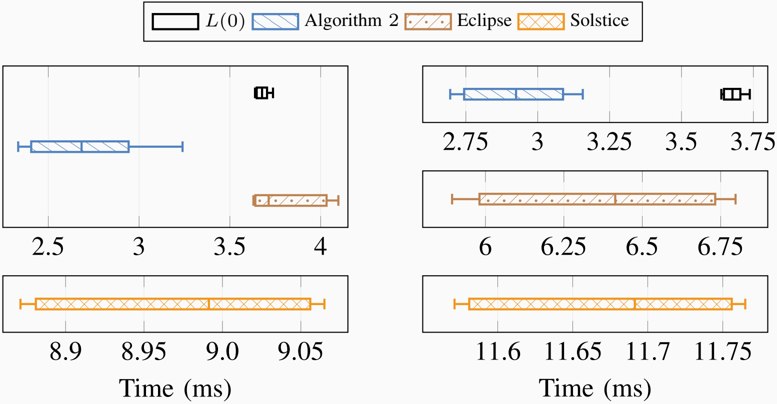
Performance Improvement under Skewed Demand



(a) Meshed traffic, $\delta = 20$ (μs). **(b)** Skewed traffic, $\delta = 20$ (μs).

Figure 4: The 1^{st}-5^{th}-50^{th}-95^{th}-99^{th} percentiles of the schedule lengths.

Effects of OCS Reconfiguration Overhead



(a) $\delta = 20$ (μs).

(b) $\delta = 200$ (μs).

Figure 5: Lighter Loading Condition.

Effects of OCS Reconfiguration Overhead

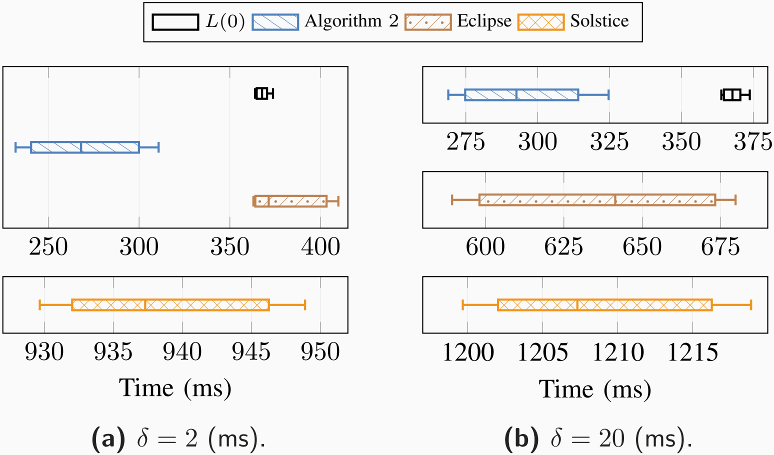


Figure 6: Heavier Loading Condition.

Effects of OCS Reconfiguration Overhead

Loading, δ	Solstice	Eclipse
Lighter, 20 (μs)	70.2%	27.2%
Lighter, 200 (μs)	75.0%	54.4%
Heavier, 2 (ms)	71.4%	27.8%
Heavier, 20 (ms)	75.8%	54.4%

Table 1: Performance improvement of the given algorithm on the 50^{th} percentile over CPSwitchSched with different schedulers.

Conclusion

- We establish a framework to study cp-switch scheduling problems systematically. It supports multiple composite paths.
- Although each cp-switch scheduling subproblem is NP-hard, a fixed approximation ratio is still possible for the overall schedule.
- Our proposed algorithm not only works online but also outperforms existing methods significantly (by 30% to 70%).

Questions & Answers

References

- S. Vargaftik, K. Barabash, Y. Ben-Itzhak, O. Biran, I. Keslassy, D. Lorenz, and A. Orda, "Composite-path switching," in *Proc. CoNEXT*. ACM, 2016, pp. 329–343.
- H. Liu, M. K. Mukerjee, C. Li, N. Feltman, G. Papen, S. Savage, S. Seshan, G. M. Voelker, D. G. Anderson, M. Kaminsky, G. Porter, and A. C. Snoeren, "Scheduling techniques for hybrid circuit/packet networks," in *Proc. CoNEXT*. ACM, 2015, pp. 41:1–41:13.

References

S. Bojja Venkatakrishnan, M. Alizadeh, and P. Viswanath, "Costly circuits, submodular schedules and approximate carathéodory theorems," in *ACM SIGMETRICS*, vol. 44, no. 1. ACM, 2016, pp. 75–88.