A Real-Time Multicast Routing Scheme for Multi-Hop Switched Fieldbuses

Lixiong Chen*, Xue Liu†, Qixin Wang‡, Yufei Wang‡
*Dept. of ECE, †School of CS, McGill Univ.
‡Dept. of Computing, The Hong Kong Polytechnic Univ.
March 30, 2011

Content


Demand


Background


Problem Definition and Complexity


Heuristic Algorithm


Evaluation


Related Work

Fieldbus is fundamentally different from the Internet, hence requires different solutions.

Specialized networks used in industrial, mining, medical, vehicular, avionic environments

Fieldbus:	**The Internet:**
Hard Real-Time	Best Effort
Periodic Stable Traffic	Random Bursty Traffic
Bounded w/ Global Info	Unbounded w/o Global Info

Fieldbus is evolving from shared medium to multi-hop switched due to scalability needs.

Concord

1976

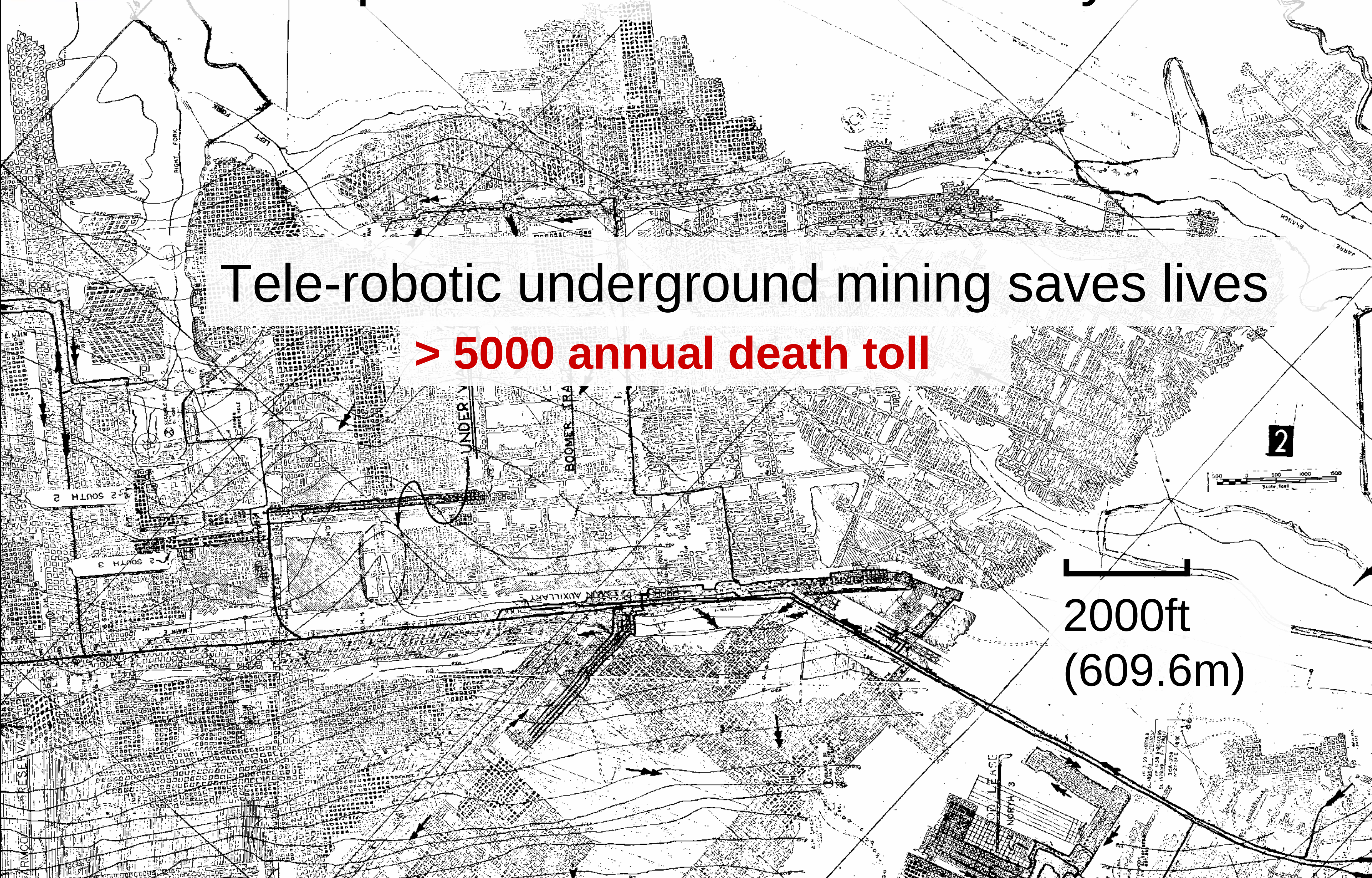
Fieldbus is evolving from shared medium to multi-hop switched due to scalability needs.

A380

2007,

Several Hundreds of computing nodes

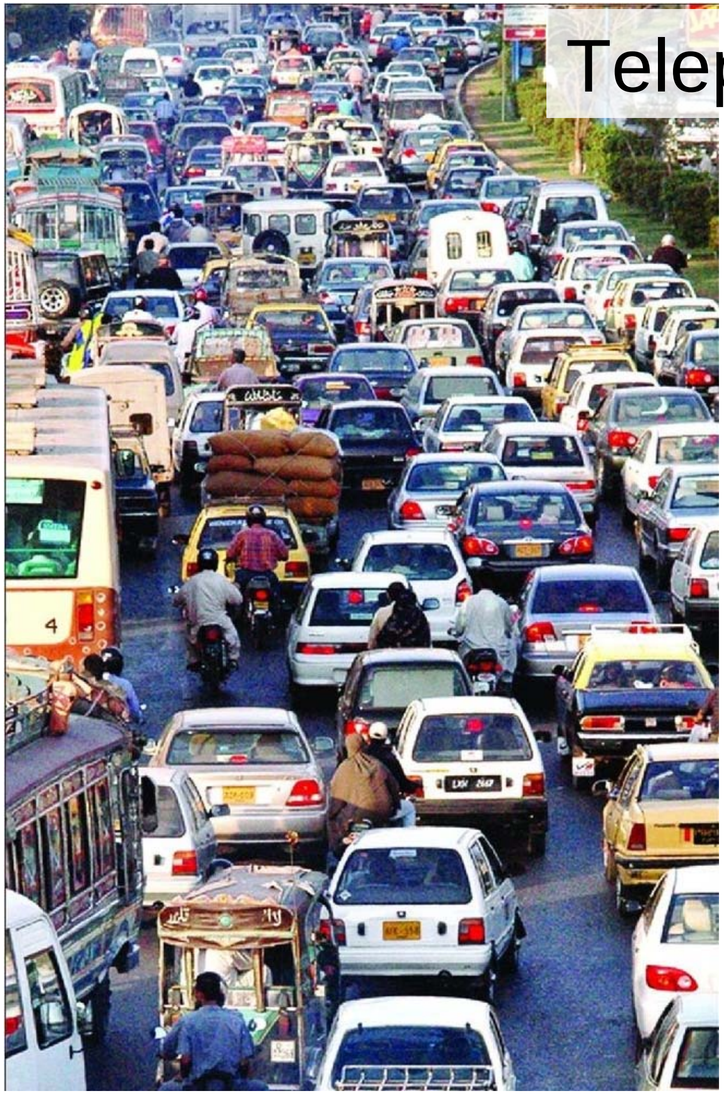
Fieldbus is evolving from shared medium to multi-hop switched due to scalability needs.



Tele-robotic underground mining saves lives

> 5000 annual death toll

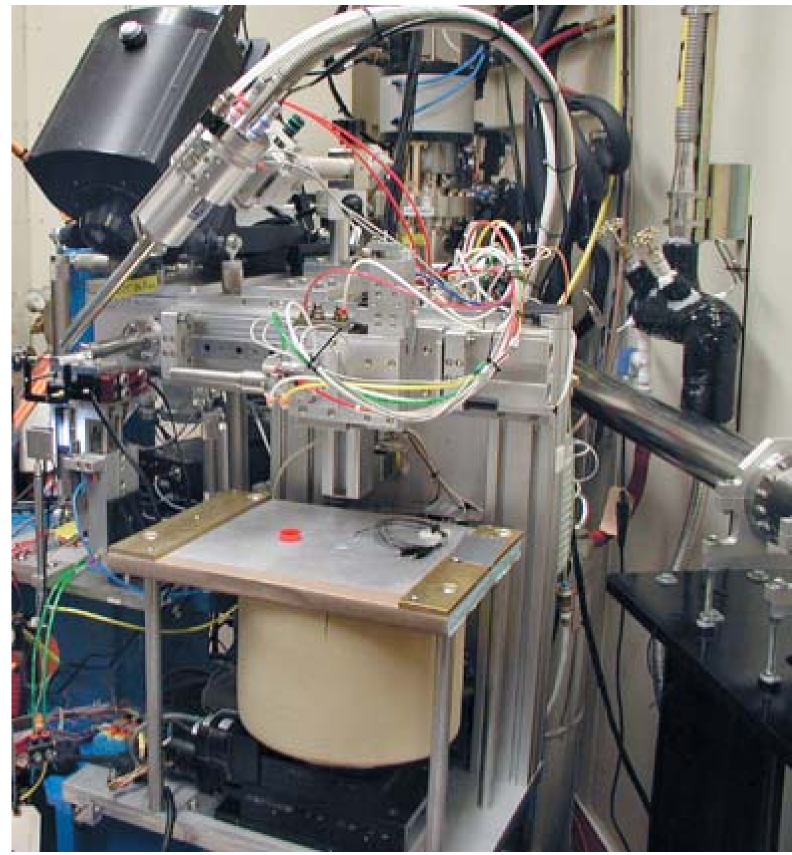
Fieldbus is evolving from shared medium to multi-hop switched due to scalability needs.



Telepresence

Fieldbus is evolving from shared medium to multi-hop switched due to scalability needs.

Robotic Manufacturing



Shared medium → multi-hop switched: real-time multicast becomes a problem.

$$\dot{x}(t)_{n\times 1} = A_{n\times n}x(t)_{n\times 1} + B_{n\times m}u(t)_{m\times 1}$$

$$u(t)_{m\times 1} = -K_{m\times n}x(t)_{n\times 1}$$

Modern control assumes MIMO →
Real-Time Multicast btw
Sensors-Controllers-Actuators/Observers

Shared medium → multi-hop switched: real-time multicast becomes a problem.

$$\dot{x}(t)_{n\times 1} = A_{n\times n}x(t)_{n\times 1} + B_{n\times m}u(t)_{m\times 1}$$

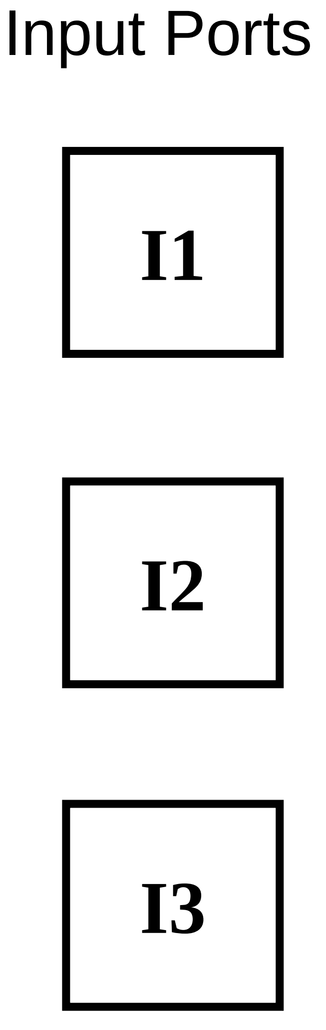
$$u(t)_{m\times 1} = -K_{m\times n}x(t)_{n\times 1}$$

Shared Medium Real-Time Multicast: Easy

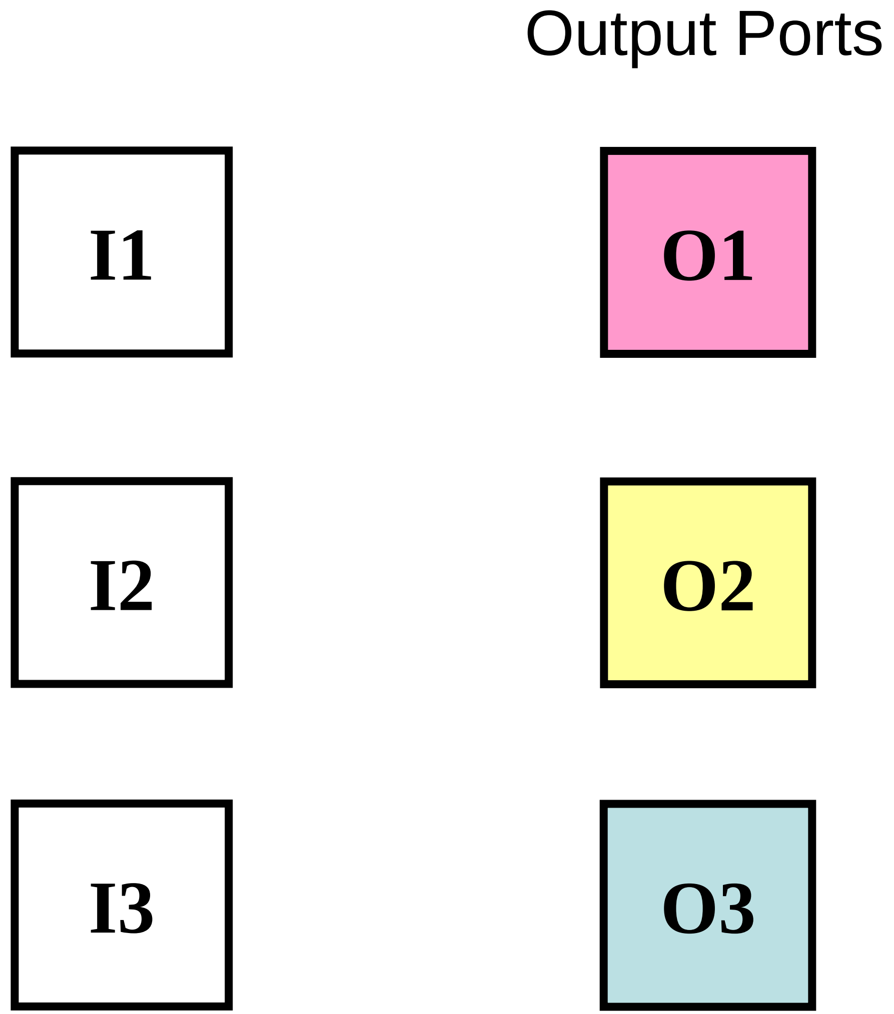
Multi-Hop Switched Real-Time Multicast: ?

De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

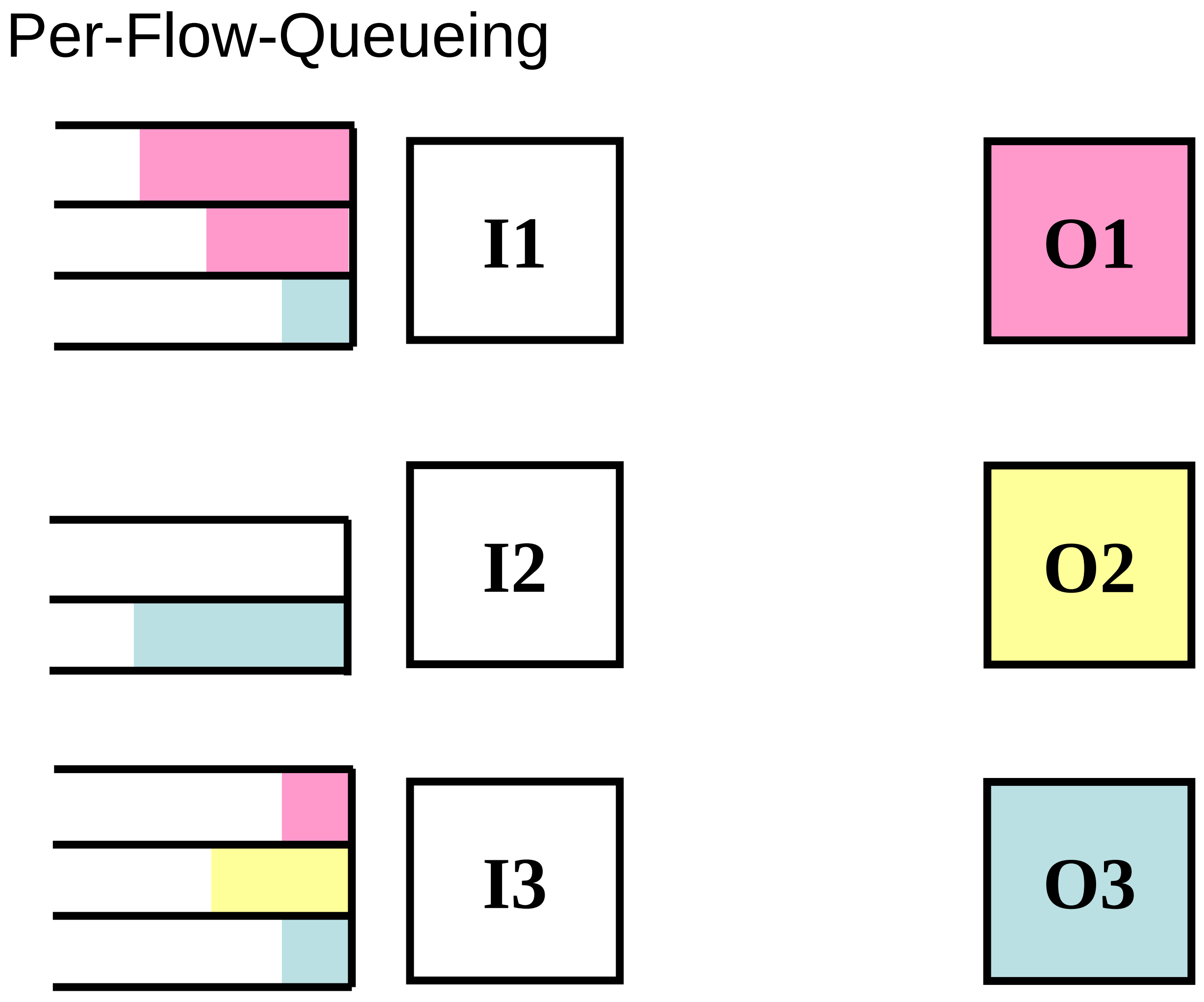
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



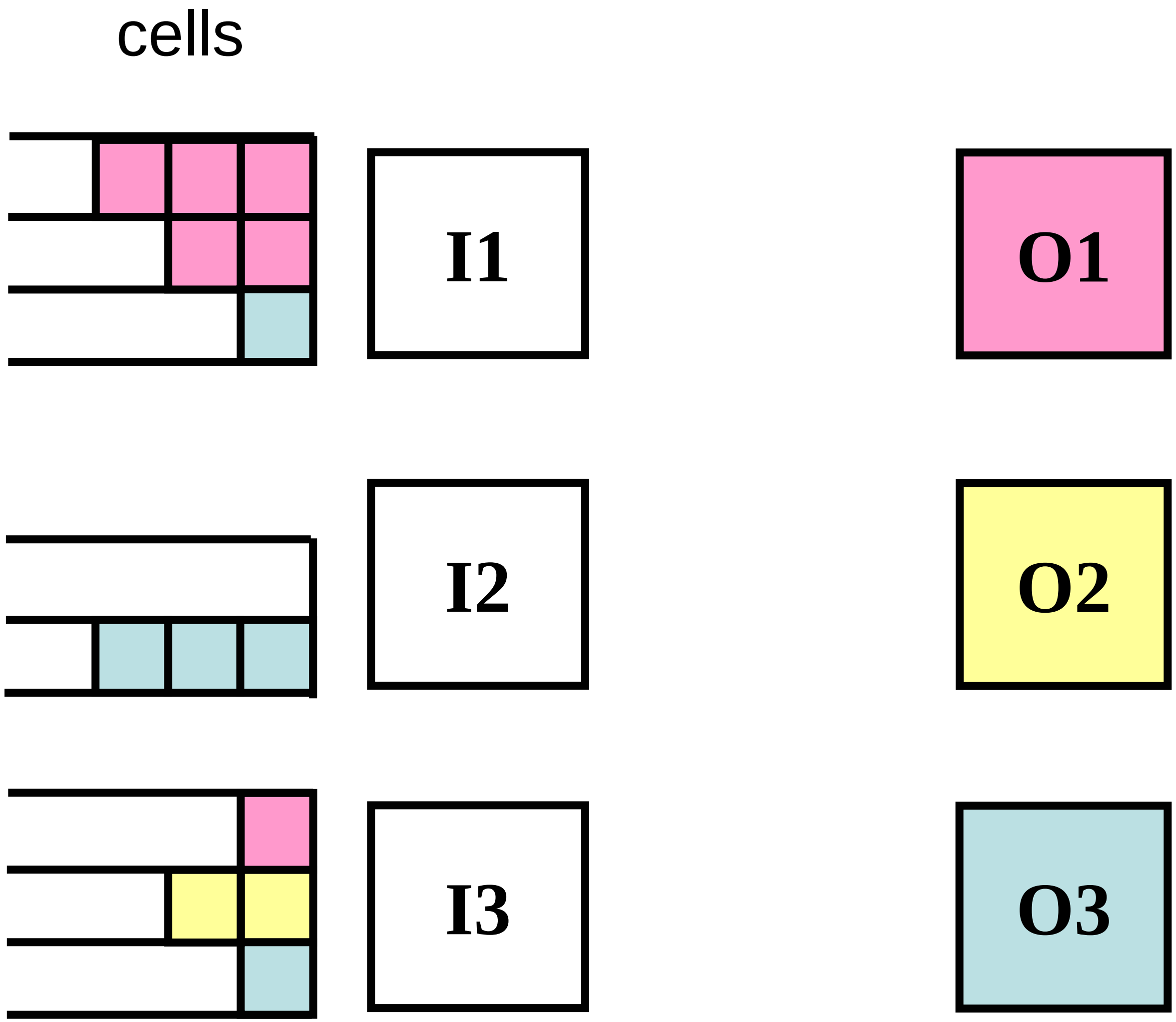
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



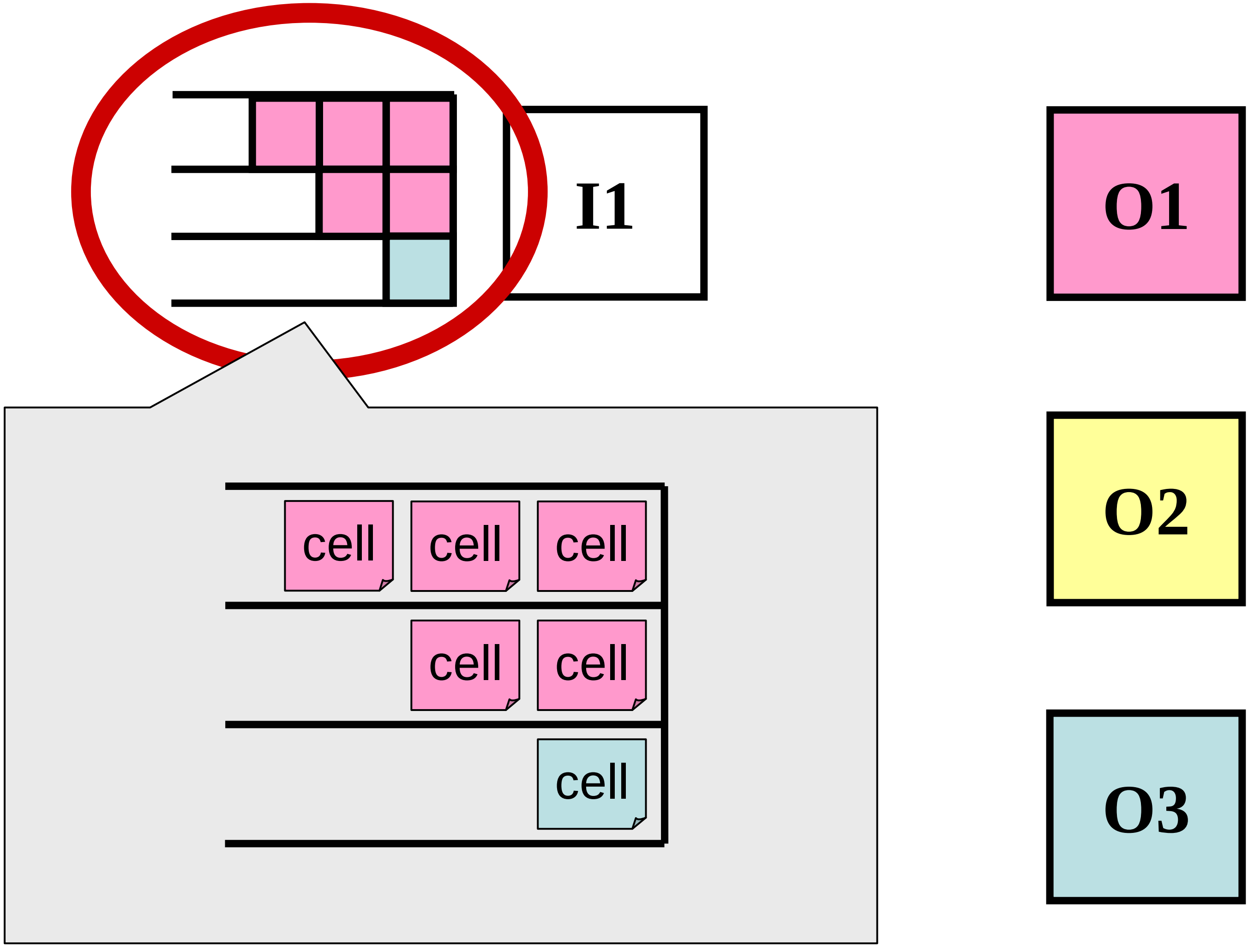
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



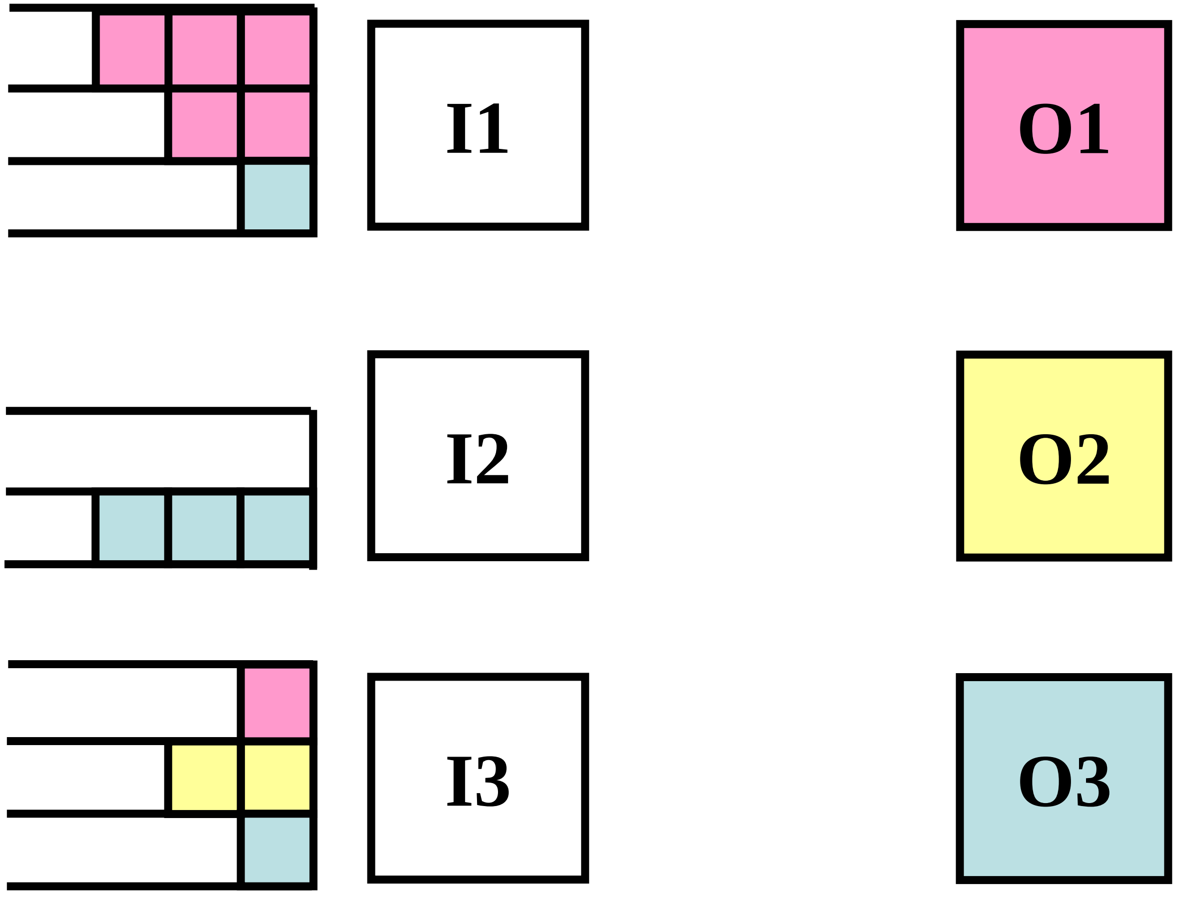
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



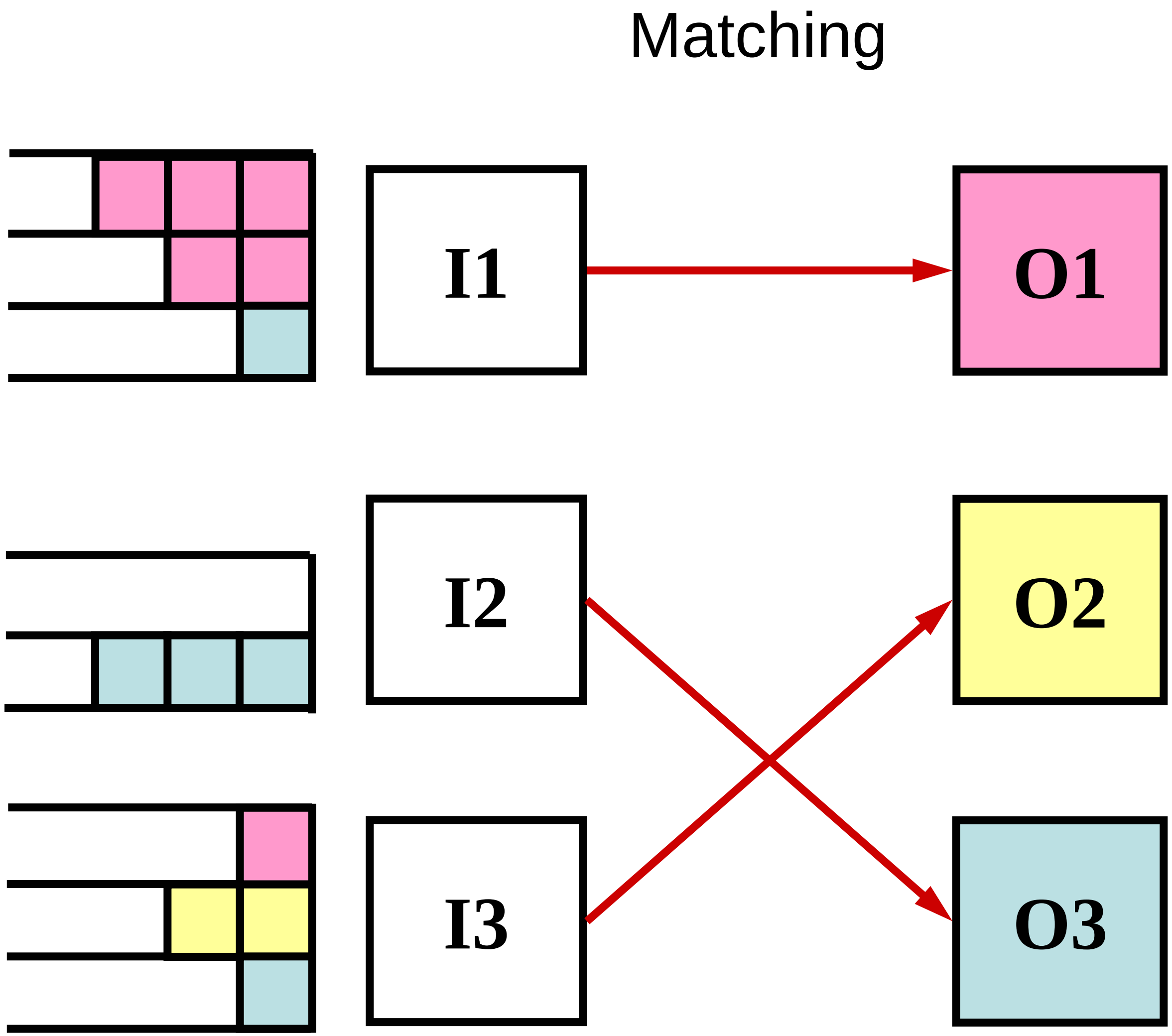
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

Synchronous periodic cell forwarding

Cell-Time

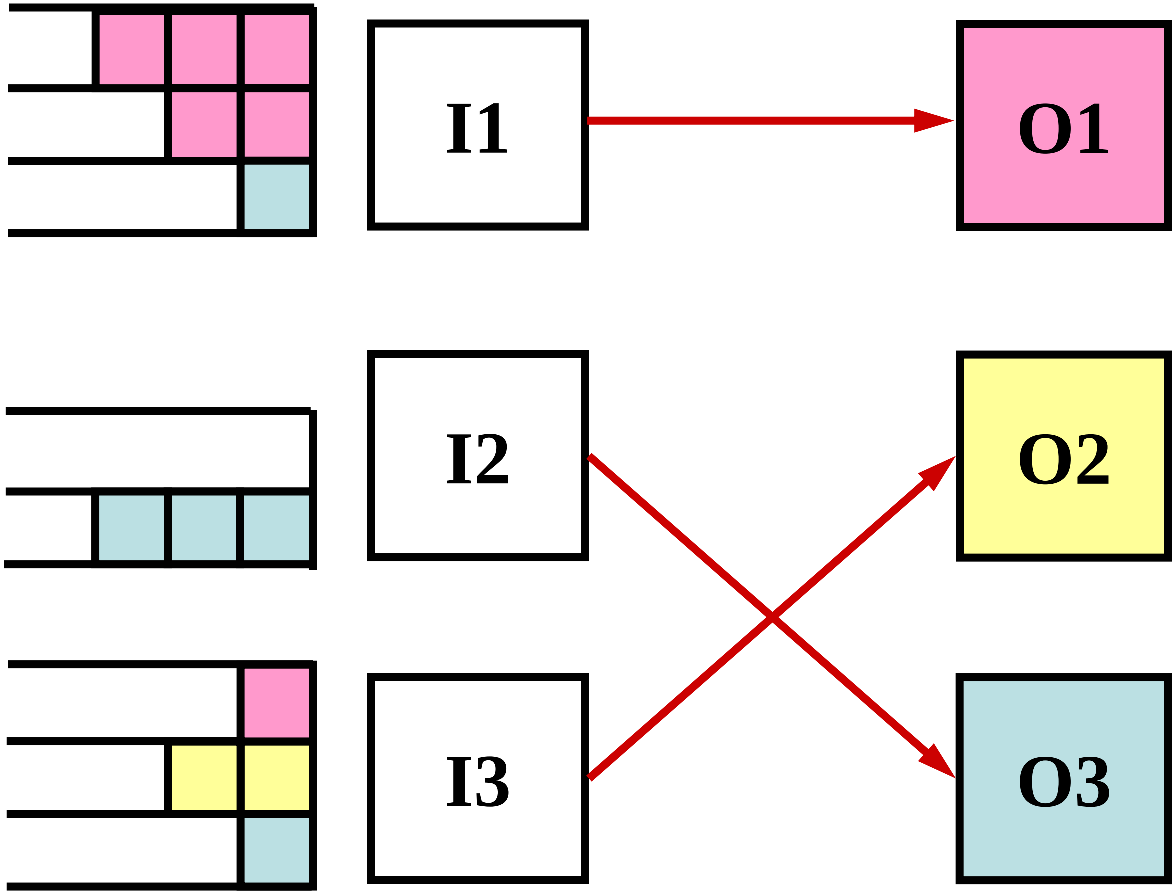


De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



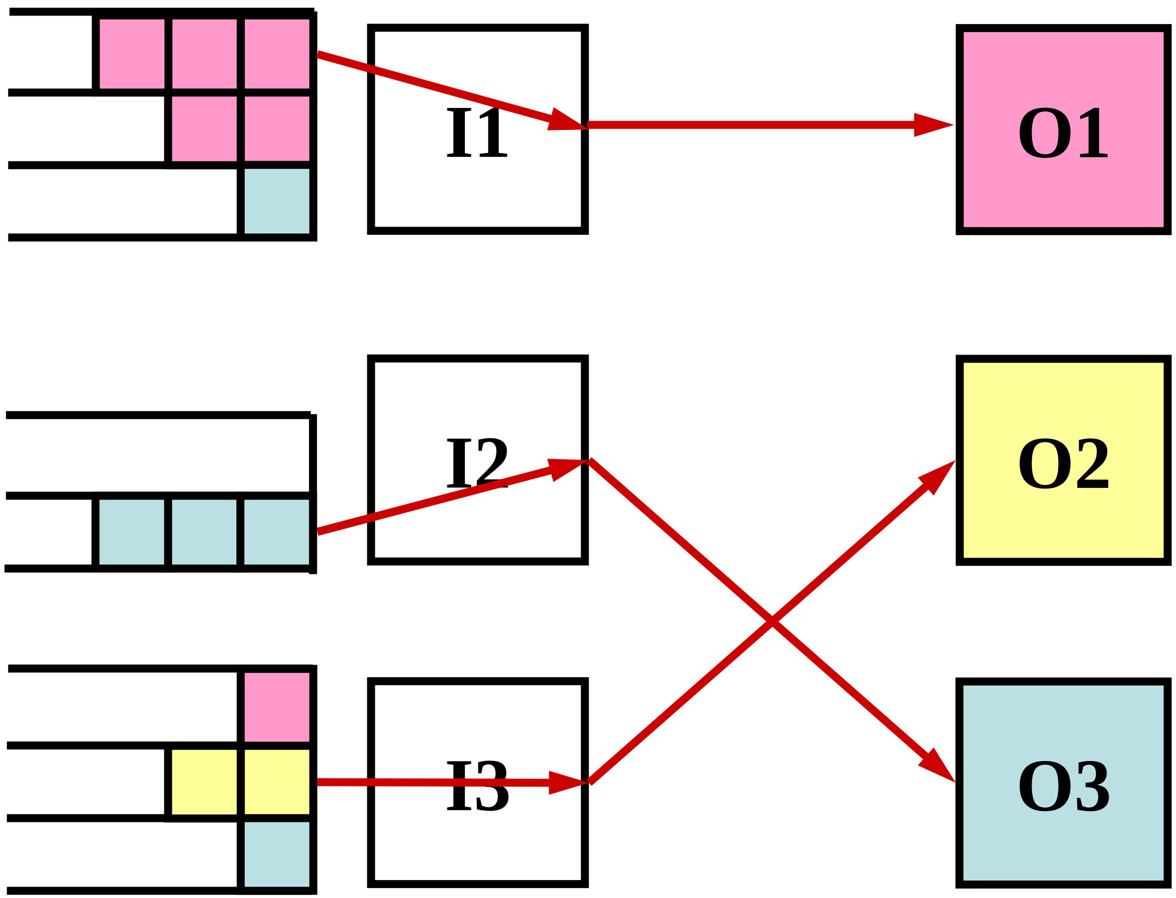
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

Why Matching? An input/output can only send/receive one cell per cell-time

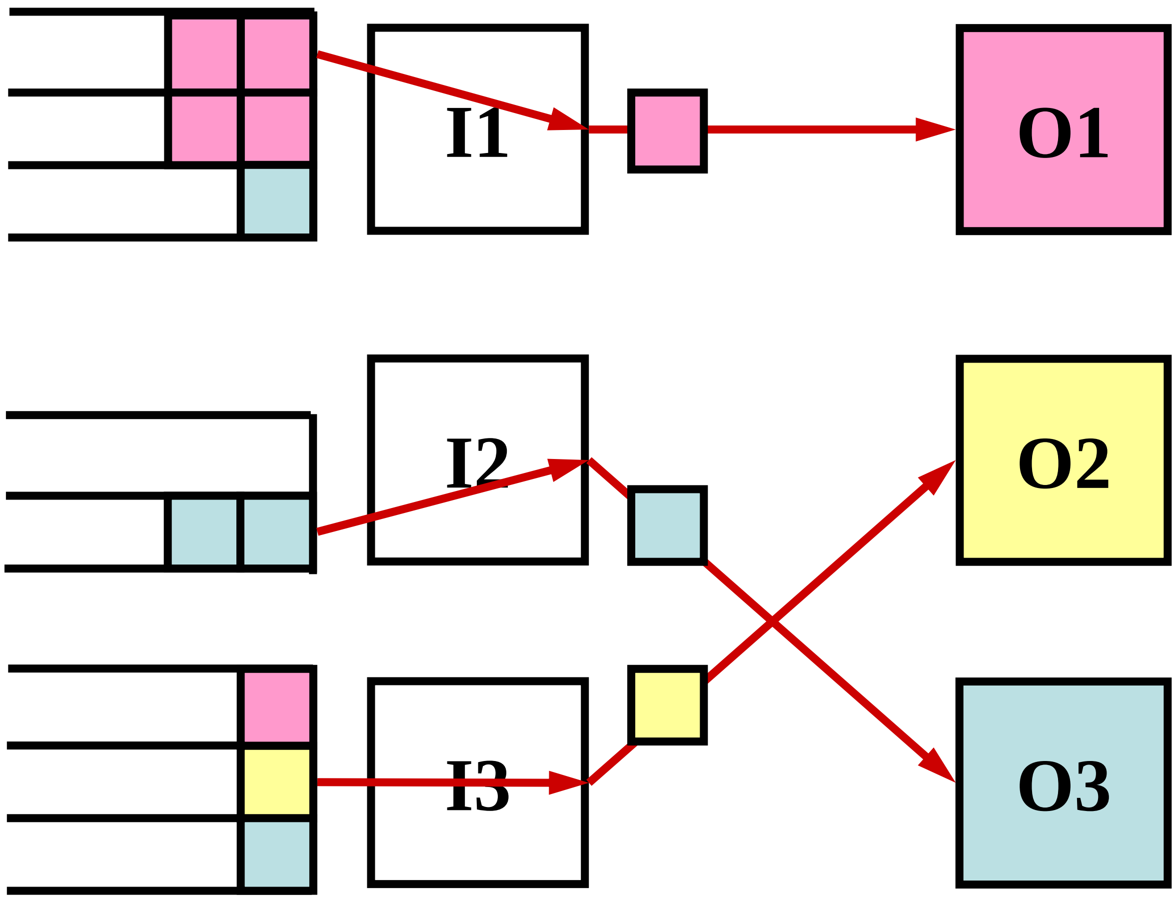


De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

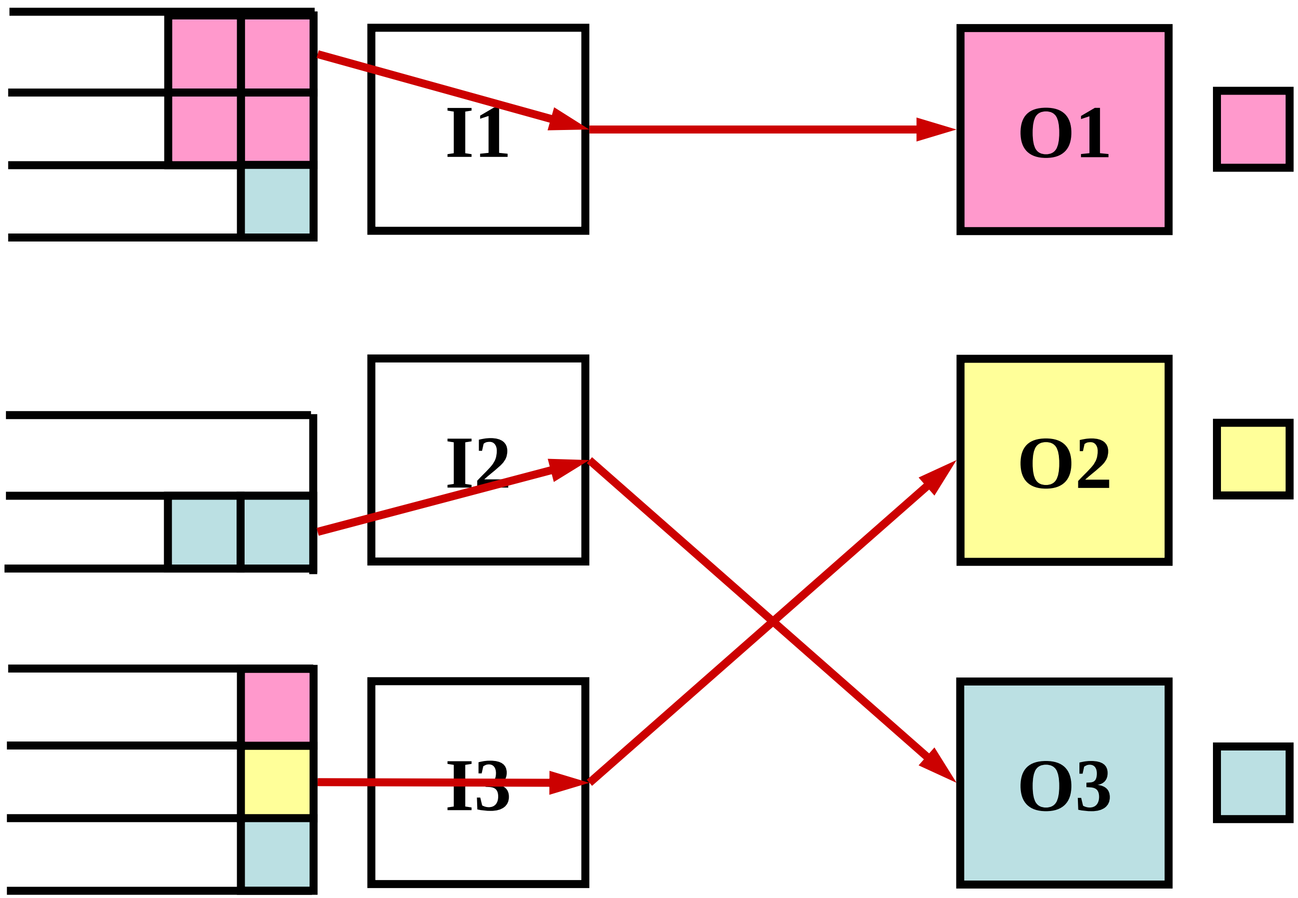
Internal Matching: if an input has multiple per-flow-q for the same output, only one is picked every cell-time.



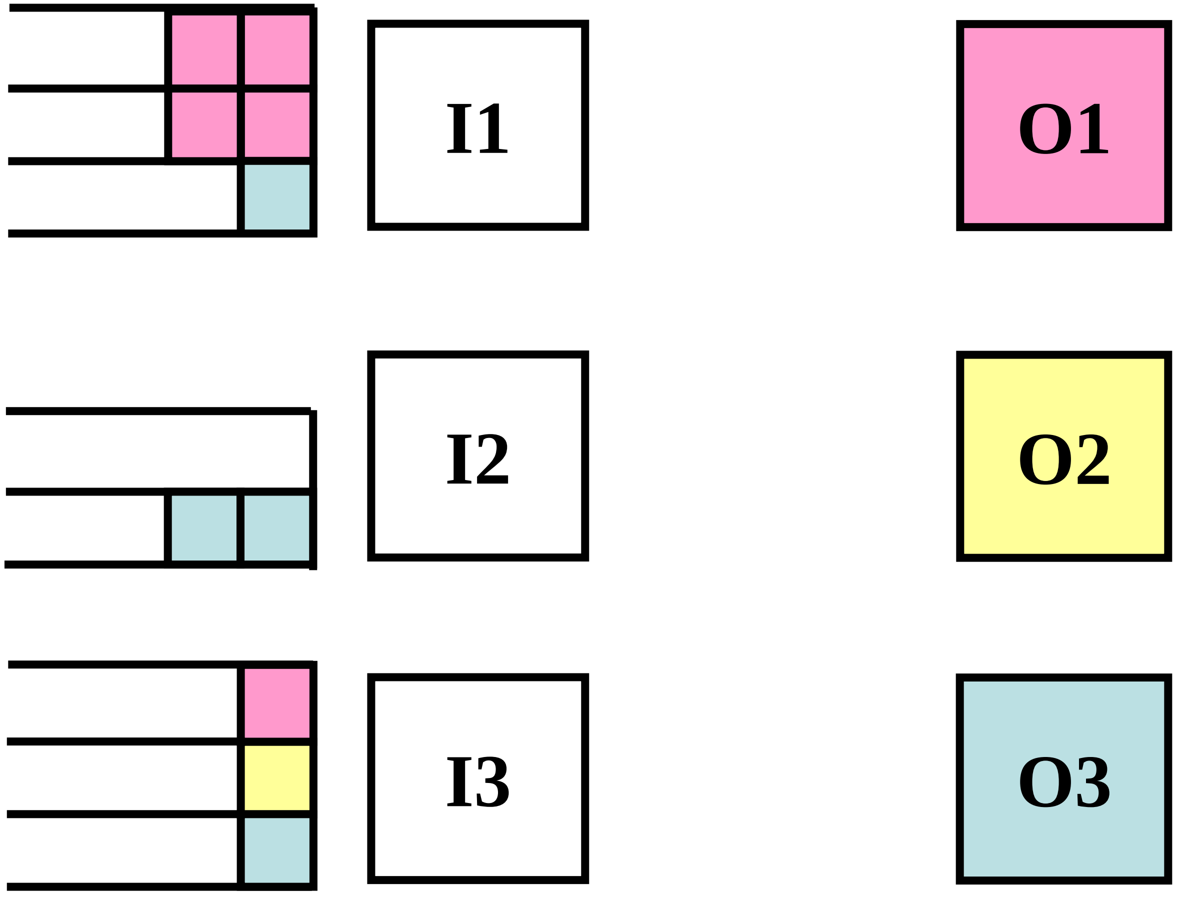
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



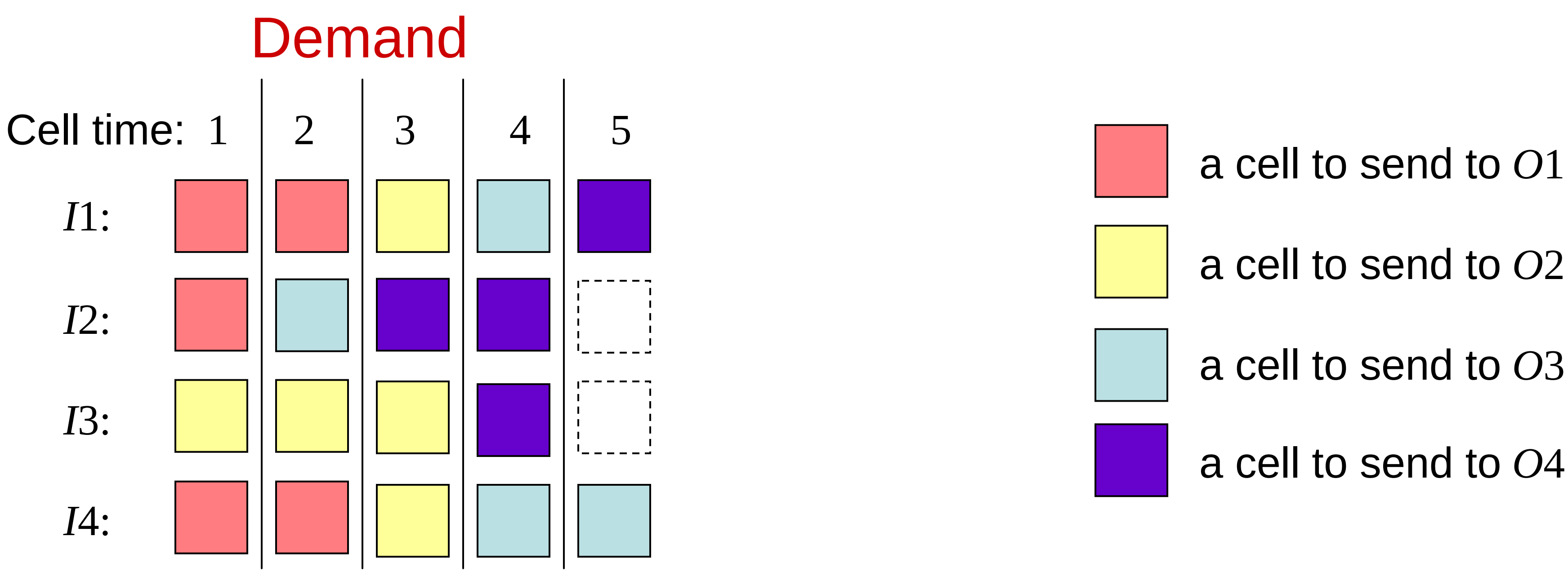
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

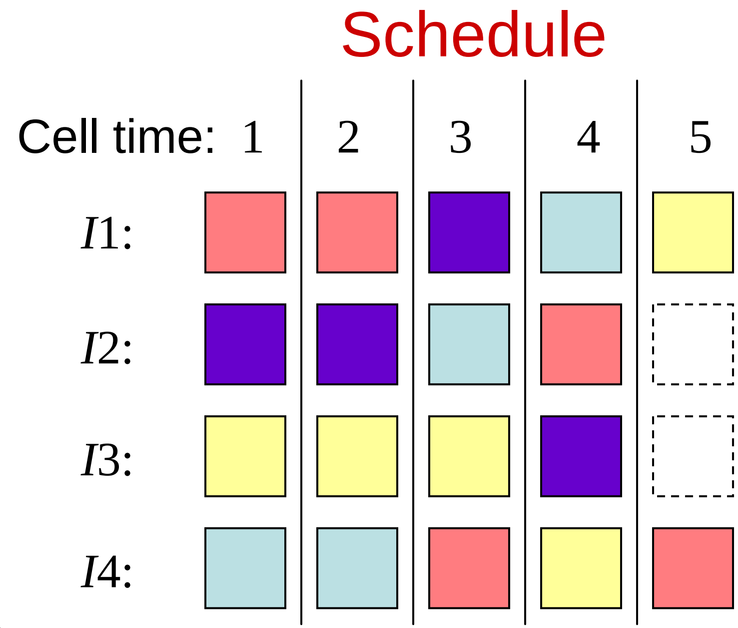
TDMA scheduling frame of $\boldsymbol{M}$ cell-time, e.g., $\boldsymbol{M} = 5$

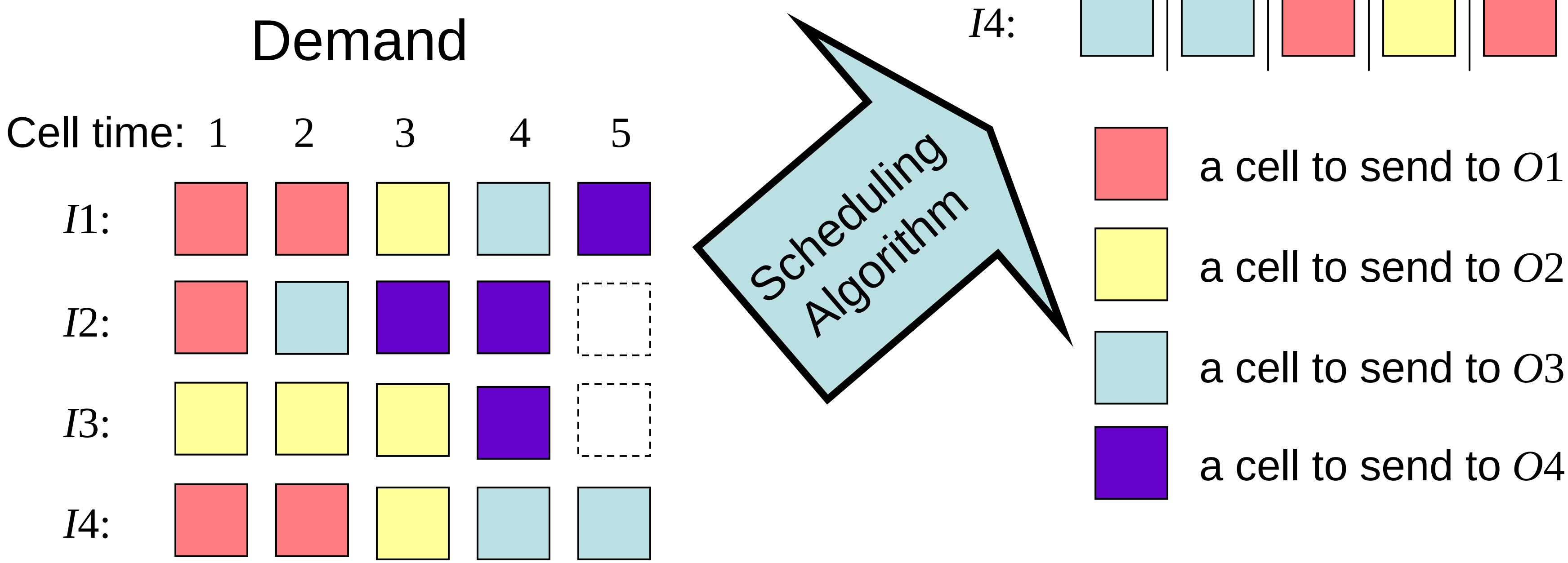
Fit all real-time flows' periods into frame,
e.g., (11, 3) → (5, 2), i.e., (10, 4)



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

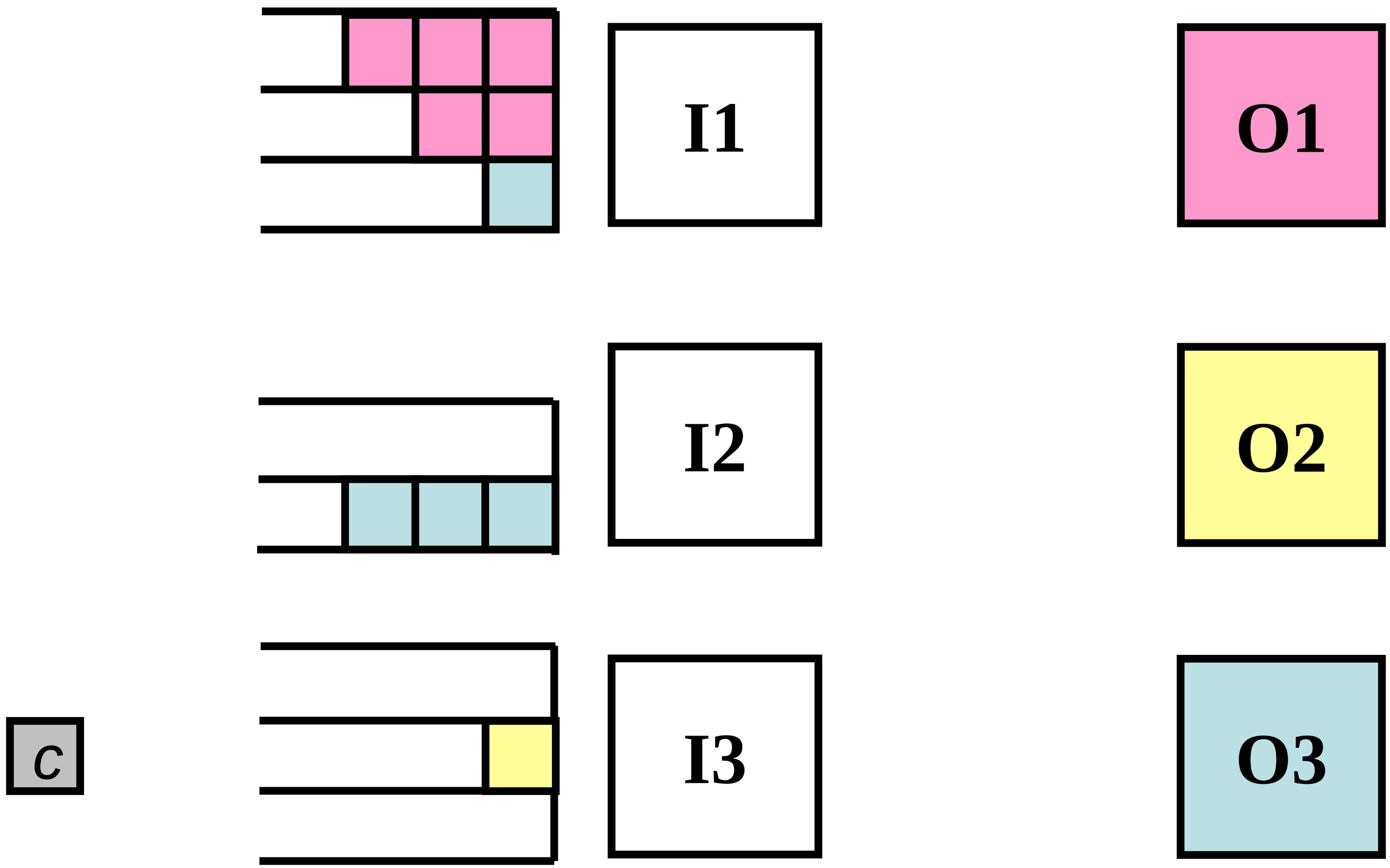
Theorem 1: If demand matrix' every color $\leqslant \boldsymbol{M}$ cell, then have config. time scheduler with $O(N^4)$ time cost [wang10].





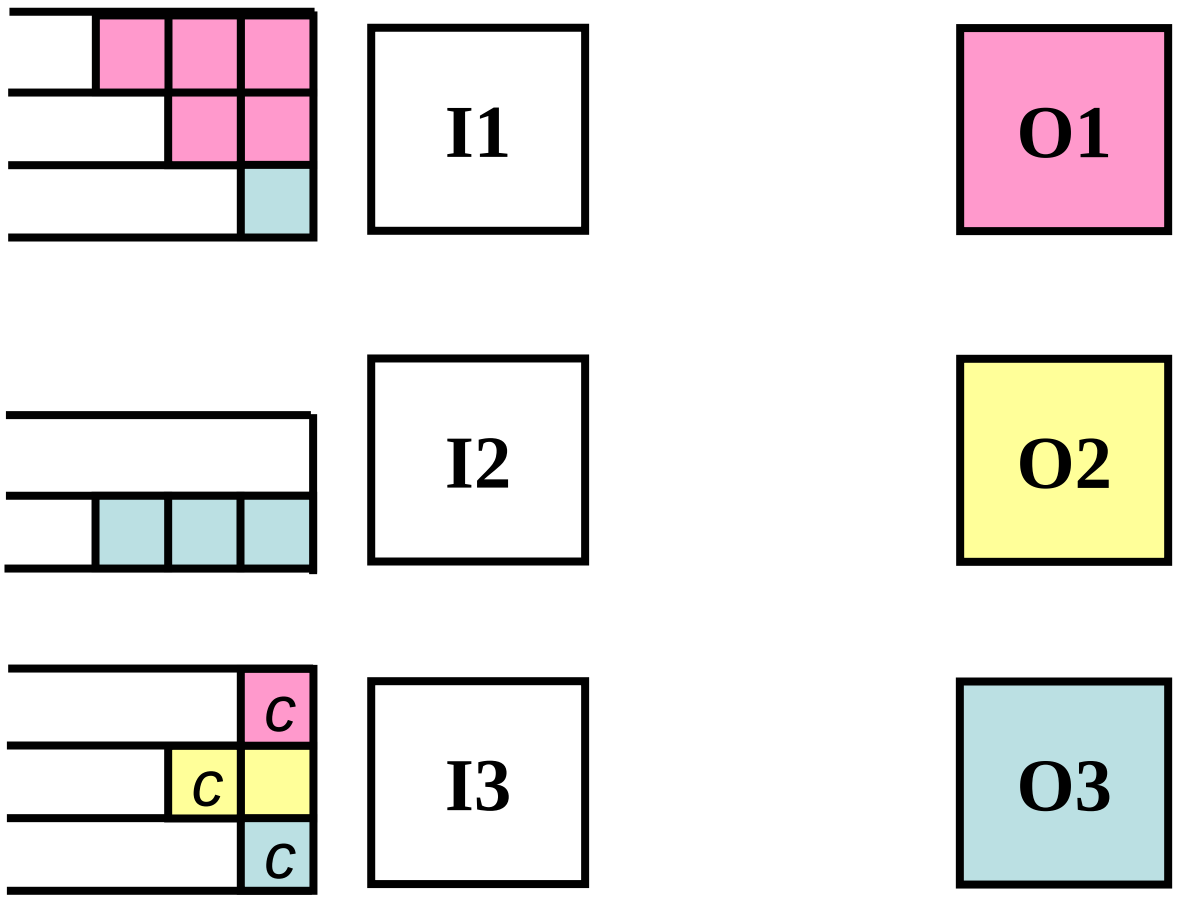
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

Support for Multicast



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA

Support for Multicast



Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V,\vec{E})$$

$$m=(s,D,w,T,H)$$

$$\mathcal{M}=\{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V, \vec{E})$$

$$m = (s, D, w, T, H)$$

$$\mathcal{M} = \{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V,\vec{E})$$

Graph of the switched real-time network (fieldbus)

$$m=(s,D,w,T,H)$$

$$\mathcal{M}=\{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V, \vec{E})$$

Graph of the switched real-time network (fieldbus)

Switches (nodes) of the network

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V, \vec{E})$$

Graph of the switched real-time network (fieldbus)

Switches (nodes) of the network

Edges of the network

$$m = (s, D, w, T, H)$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V, \vec{E})$$

$$m = (s, D, w, T, H)$$

$$\mathcal{M} = \{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V,\vec{E})$$

$$m=(s,D,w,T,H)$$

A (real-time) multicast group

$$\mathcal{M}=\{m_i\}$$

$$V,\vec{E}),\mathcal{M})$$

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V,\vec{E})$$

$$m = (s, D, w, T, H)$$

$$\mathcal{M} = \{m_i\}$$

A (real-time) multicast group

Source End

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$m = (s, D, w, T, H)$$

Destination Ends

A (real-time) multicast group

Source End

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

Destination Ends

$$m = (s, D, w, T, H)$$

A (real-time) multicast group

Source End

Cells to multicast every period

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$m = (s, D, w, T, H)$$

- m: A (real-time) multicast group
- s: Source End
- D: Destination Ends
- w: Cells to multicast every period
- T: Period (unit: cell-time)

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$m = (s, D, w, T, H)$$

- Destination Ends (D)
- Period (unit: cell-time) (T)
- Deadline (relative, unit: cell-time) (H)
- A (real-time) multicast group (m)
- Source End (s)
- Cells to multicast every period (w)

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V,\vec{E})$$

$$m=(s,D,w,T,H)$$

$$\mathcal{M}=\{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V, \vec{E})$$

The set of all real-time multicast groups

$$= (s, D, w, T, H)$$

$$\mathcal{M} = \{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$\vec{G}(V,\vec{E})$

The set of all real-time multicast groups

The ith real-time multicast group

$= (s, D, w, T, H)$

$$\mathcal{M} = \{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$$\vec{G}(V,\vec{E})$$

$$m=(s,D,w,T,H)$$

$$\mathcal{M}=\{m_i\}$$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

$\vec{G}(V,\vec{E})$

An instance of RTMS problem

$= (s, D, w, T, H)$

$\mathcal{M} = \{m_i\}$

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

An instance of RTMS problem

The network(fieldbus) topology

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

Real-Time Multicast Scheduling (RTMS) problem in multi-hop switched fieldbuses

An instance of RTMS problem

The network(fieldbus) topology

The set of RT multicast groups that user demands

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

RTMS Problem: Given a q, how to schedule each switch s.t. every m_i meets needs?

$$\vec{G}(V, \vec{E})$$

$$m = (s, D, w, T, H)$$

$$\mathcal{M} = \{m_i\}$$

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

RTMS Problem: Given a q, how to schedule each switch s.t. every m_i meets its needs?

Theorem 2: RTMS Problem is NP-Hard.

$$\vec{G}(V, \vec{E})$$

$$m = (s, D, w, T, H)$$

$$\mathcal{M} = \{m_i\}$$

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

M-slot Periodic RTMS: a subset of RTMS problems.

$$\mathcal{Q} = \{q \mid q \text{ is an RTMS instance}\}$$

$$\mathcal{Q}' = \{q' \mid q' = (\vec{G}, \mathcal{M}) \in \mathcal{Q},$$
$$\mathcal{M} = \{m_i\}, m_i = (s_i, D_i, w_i, T_i, H_i),$$
$$\text{and } \forall i, T_i \equiv M\}$$

M-slot Periodic RTMS: a subset of RTMS problems.

$$\mathcal{Q} = \{q \mid q \text{ is an RTMS instance}\}$$

Proposition 1: M-slot Periodic RTMS is NP-Hard.

$$\mathcal{Q}' = \{q' \mid q' = (\vec{C}, \mathcal{M}) \in \mathcal{Q},$$

$$\mathcal{M} = \{m_i\}, m_i = (s_i, D_i, w_i, T_i, H_i),$$

$$\text{and } \forall i, T_i \equiv M\}$$

M-slot Periodic RTMS: a subset of RTMS problems.

$\mathcal{Q} = \{q \mid q \text{ is an RTMS instance}\}$

Proposition 1: M-slot Periodic RTMS is NP-Hard.

$\mathcal{Q}' = \{q' \mid q' = (\vec{C}, \mathcal{M}) \in \mathcal{Q},$

$\mathcal{M} = \{m_i\}, m_i = (s_i, D_i, w_i, T_i, H_i),$

Search for Heuristic Solutions.

$\text{and } \forall i, T_i \equiv M\}$

Transform an M-slot Periodic RTMS instance into a Real-Time Multicast Routing (RTMR) instance.

Given

$$q' = (\vec{G}, \mathcal{M}) \in \mathcal{Q}',$$

where $\mathcal{M} = \{m_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), H_i)\}$, define

$$\tilde{q} = (\vec{G}, \tilde{\mathcal{M}}),$$

where $\tilde{\mathcal{M}} = \{\tilde{m}_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), \tilde{H}_i)\}$, and

$$\tilde{H}_i = \max\left\{\left\lfloor \frac{H_i - M}{M+1} \right\rfloor, 0\right\}.$$

Transform an M-slot Periodic RTMS instance into a Real-Time Multicast Routing (RTMR) instance.

M-slot Periodic RTMS instance

Given

$$q' = (\vec{G}, \mathcal{M}) \in \mathcal{Q}',$$

where $\mathcal{M} = \{m_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), H_i)\}$, define

$$\tilde{q} = (\vec{G}, \tilde{\mathcal{M}}),$$

where $\tilde{\mathcal{M}} = \{\tilde{m}_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), \tilde{H}_i)\}$, and

$$\tilde{H}_i = \max\left\{\left\lfloor \frac{H_i - M}{M+1} \right\rfloor, 0\right\}.$$

Transform an M-slot Periodic RTMS instance into a Real-Time Multicast Routing (RTMR) instance.

M-slot Periodic RTMS instance

Given

$$q' = (\vec{G}, \mathcal{M}) \in \mathcal{Q}',$$

where $\mathcal{M} = \{m_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), H_i)\}$, define

RTMR instance

$$\tilde{q} = (\vec{G}, \tilde{\mathcal{M}}),$$

where $\tilde{\mathcal{M}} = \{\tilde{m}_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), \tilde{H}_i)\}$, and

$$\tilde{H}_i = \max\left\{\left\lfloor \frac{H_i - M}{M+1} \right\rfloor, 0\right\}.$$

Transform an M-slot Periodic RTMS instance into a Real-Time Multicast Routing (RTMR) instance.

M-slot Periodic RTMS instance

Given

$$q' = (\vec{G}, \mathcal{M}) \in \mathcal{Q}',$$

where $\mathcal{M} = \{m_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), H_i)\}$, define

RTMR instance

$$\tilde{q} = (\vec{G}, \tilde{\mathcal{M}}),$$

Max Multicast Tree Height

where $\tilde{\mathcal{M}} = \{\tilde{m}_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), \tilde{H}_i)\}$, and

$$\tilde{H}_i = \max\left\{\left\lfloor \frac{H_i - M}{M+1} \right\rfloor, 0\right\}.$$

Transform an M-slot Periodic RTMS instance into a Real-Time Multicast Routing (RTMR) instance.

Given

$$q' = (\vec{G}, \mathcal{M}) \in Q',$$

where $\mathcal{M} = \{m_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), H_i)\}$, define

$$\tilde{q} = (\vec{G}, \tilde{\mathcal{M}}),$$

where $\tilde{\mathcal{M}} = \{\tilde{m}_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), \tilde{H}_i)\}$, and

$$\tilde{H}_i = \max\left\{\left\lfloor \frac{H_i - M}{M+1} \right\rfloor, 0\right\}.$$

A solution to $\tilde{q} = (\vec{G}, \tilde{\mathcal{M}})$ is a solution to $q' = (\vec{G}, \mathcal{M})$.

Transform an M-slot Periodic RTMS instance into a Real-Time Multicast Routing (RTMR) instance.

A solution to $\tilde{q} = (\vec{G}, \tilde{\mathcal{M}})$ is a solution to $q' = (\vec{G}, \mathcal{M})$.

RTMScheduling problem becomes a Routing problem.

Given

$$q' = (\vec{G}, \mathcal{M}) \in Q',$$

where $\mathcal{M} = \{m_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), H_i)\}$, define

$$\tilde{q} = (\vec{G}, \tilde{\mathcal{M}}),$$

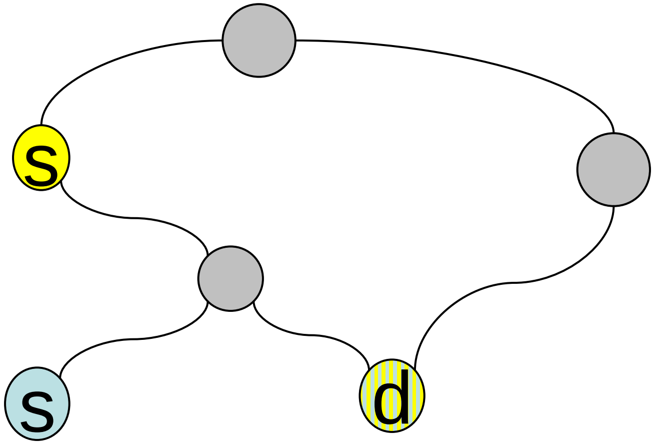
where $\tilde{\mathcal{M}} = \{\tilde{m}_i\} = \{(s_i, D_i, w_i, T_i(\equiv M), \tilde{H}_i)\}$, and

$$\tilde{H}_i = \max\left\{ \left\lfloor \frac{H_i - M}{M+1} \right\rfloor, 0 \right\}.$$

Heuristic Routing Algorithm

Existing mainstream internet multicast routing algorithms become Dijkstra when network is static and global info is available

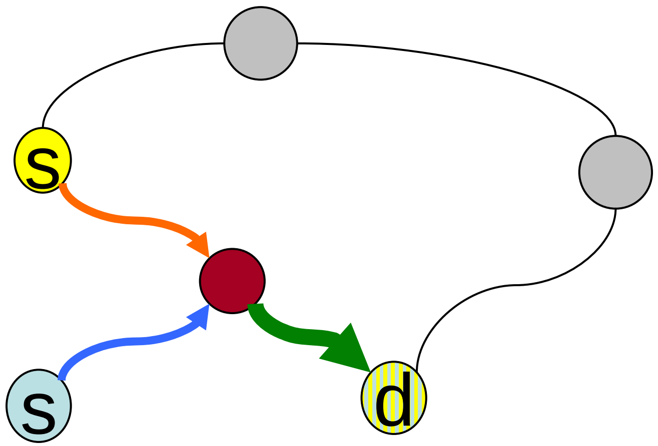
Dijkstra’s short-coming: only cares about # of hops, ignores congestion.



Heuristic Routing Algorithm

Existing mainstream internet multicast routing algorithms become Dijkstra when network is static and global info is available

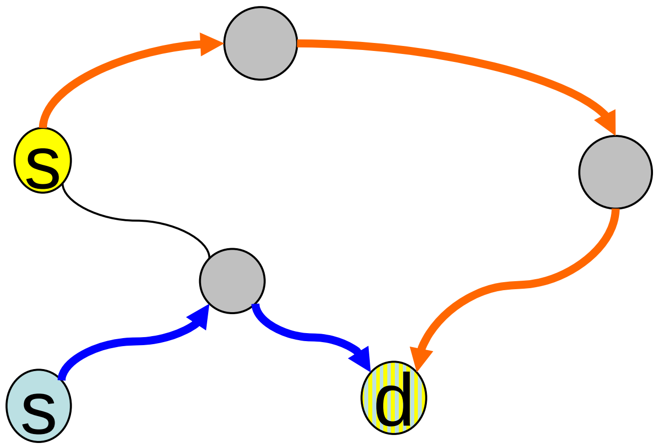
Dijkstra's short-coming: only cares about # of hops, ignores congestion.



Heuristic Routing Algorithm

Existing mainstream internet multicast routing algorithms become Dijkstra when network is static and global info is available

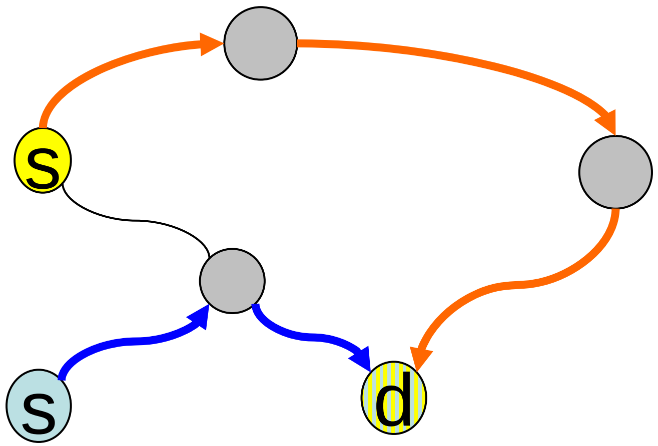
We want a heuristic routing algorithm that considers both (hops and congestion).



Heuristic Routing Algorithm

Existing mainstream internet multicast routing algorithms become Dijkstra when network is static and global info is available

We want a heuristic routing algorithm that considers both (hops and congestion).



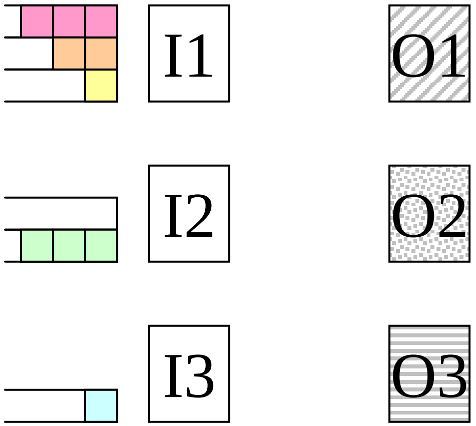
Heuristic Routing Algorithm

Grow the multiple trees simultaneously with multiple iterations.

In each iteration, $\leqslant$1 link is added to each tree.

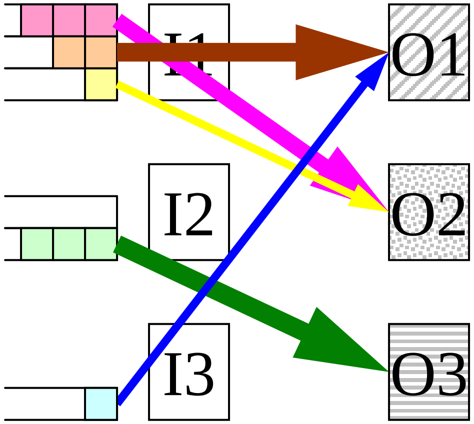
When multiple tree contends in a same switch, we carry out a job-hunting-like negotiation to let only $\leqslant$1 contending tree grow through an output in this iteration.

The job-hunting-like contention negotiation is itself an iterative algorithm.



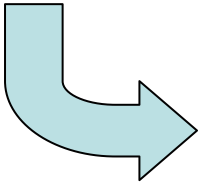
different colors for
different trees

The job-hunting-like contention negotiation is itself an iterative algorithm.

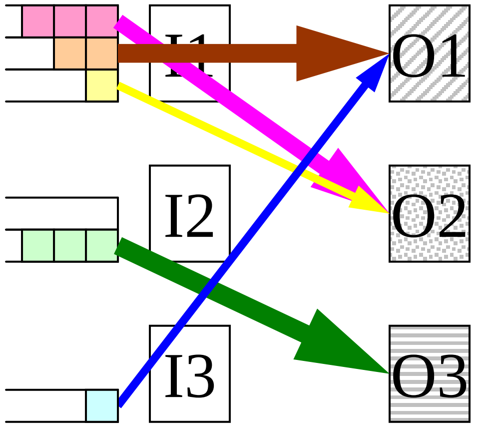


different colors for different trees

Each tree ranks all outputs and only apply to its favorite output

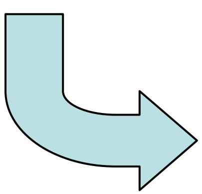


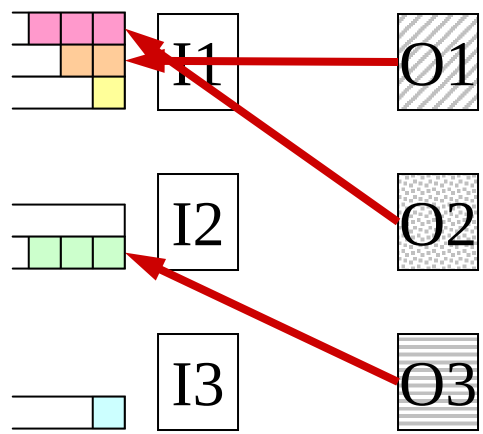
The job-hunting-like contention negotiation is itself an iterative algorithm.



different colors for
different trees

Each tree ranks all outputs and only apply to its favorite output

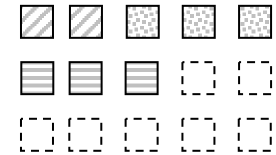


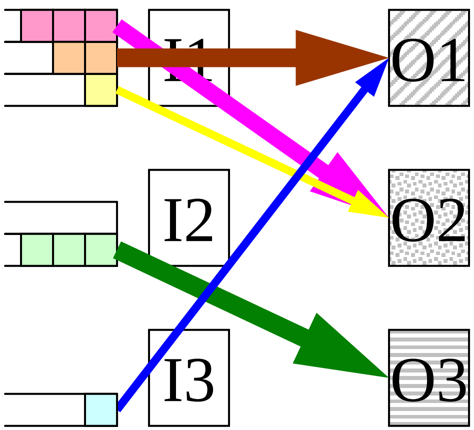


Each output offers job to the most loyal applicant

The job-hunting-like contention negotiation is itself an iterative algorithm.

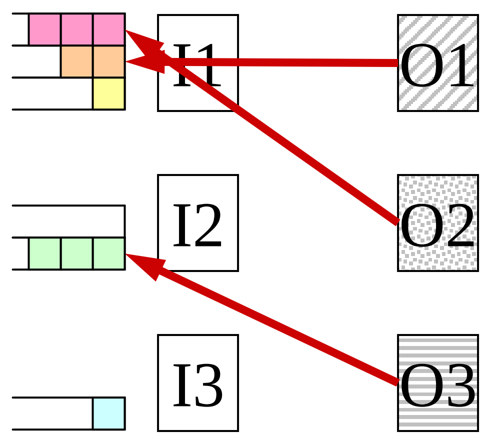
updated demand matrix



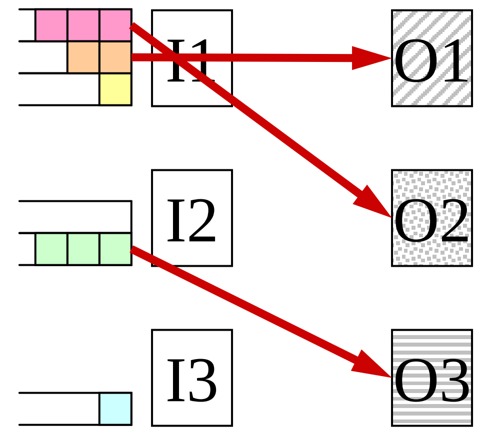


different colors for different trees

Each tree ranks all outputs and only apply to its favorite output



Each output offers job to the most loyal applicant



Accept job by reserving corresponding frame slots.

Ranking function design: considers both E2E delay (hops) and congestion.

$$r_{t,o} = \frac{\gamma_o(\tilde{H} - H(t,v))}{\mathrm{dis}(v, t.target)}$$

$$\gamma_o = \begin{cases} 1 & (\text{if } \max_{j \in \mathcal{O}_u}\{N_j\} \\ & = \max_{j \in \mathcal{O}_u}\{N'_j\}), \\ \exp(\max_{j \in \mathcal{O}_u}\{N_j\} & (\text{otherwise}) \\ - \max_{j \in \mathcal{O}_u}\{N'_j\}) & \end{cases}$$

Ranking function design: considers both E2E delay (hops) and congestion.

rank of o to tree t

$$r_{t,o} = \frac{\gamma_o(\tilde{H} - H(t,v))}{\mathrm{dis}(v, t.target)}$$

$$\gamma_o = \begin{cases} 1 & (\text{if } \max_{j\in\mathcal{O}_u}\{N_j\} \\ & = \max_{j\in\mathcal{O}_u}\{N'_j\}), \\ \exp(\max_{j\in\mathcal{O}_u}\{N_j\} & (\text{otherwise}) \\ - \max_{j\in\mathcal{O}_u}\{N'_j\}) & \end{cases}$$

Ranking function design: considers both E2E delay (hops) and congestion.

rank of o to tree t

routing flexibility: slack to reaching max hop (max e2e delay) bound

$$r_{t,o} = \frac{\gamma_o(\tilde{H} - H(t,v))}{\mathrm{dis}(v, t.target)}$$

$$\gamma_o = \begin{cases} 1 & (\text{if } \max_{j \in \mathcal{O}_u}\{N_j\} \\ & = \max_{j \in \mathcal{O}_u}\{N'_j\}), \\ \exp(\max_{j \in \mathcal{O}_u}\{N_j\} & (\text{otherwise}) \\ - \max_{j \in \mathcal{O}_u}\{N'_j\}) & \end{cases}$$

Ranking function design: considers both E2E delay (hops) and congestion.

rank of o to tree t

routing flexibility: slack to reaching max hop (max e2e delay) bound

shortest distance to target end

$$r_{t,o} = \frac{\gamma_o(\tilde{H} - H(t,v))}{\mathrm{dis}(v, t.target)}$$

$$\gamma_o = \begin{cases} 1 & (\text{if } \max_{j \in \mathcal{O}_u}\{N_j\} \\ & = \max_{j \in \mathcal{O}_u}\{N'_j\}), \\ \exp(\max_{j \in \mathcal{O}_u}\{N_j\} & (\text{otherwise}) \\ - \max_{j \in \mathcal{O}_u}\{N'_j\}) & \end{cases}$$

Ranking function design: considers both E2E delay (hops) and congestion.

rank of o to tree t

routing flexibility: slack to reaching max hop (max e2e delay) bound

$$r_{t,o} = \frac{\gamma_o(\tilde{H} - H(t,v))}{\mathrm{dis}(v, t.target)}$$

shortest distance to target end

congestion↑ ⇔↓

$$\gamma_o = \begin{cases} 1 & (\text{if } \max_{j \in \mathcal{O}_u}\{N_j\} \\ & = \max_{j \in \mathcal{O}_u}\{N'_j\}), \\ \exp(\max_{j \in \mathcal{O}_u}\{N_j\} & (\text{otherwise}) \\ - \max_{j \in \mathcal{O}_u}\{N'_j\}) & \end{cases}$$

Definition of "favorite", a.k.a., "loyalty".

$$r_{t,o} = \frac{\gamma_o(\tilde{H} - H(t,v))}{\mathrm{dis}(v, t.target)}$$

$$o^{(t,1)} \stackrel{def}{=} \mathrm{argmax}_{o \in \mathcal{O}}\{r_{t,o}\}$$

$$o^{(t,2)} \stackrel{def}{=} \mathrm{argmax}_{o \in \mathcal{O} - \{o^{(t,1)}\}}\{r_{t,o}\}$$

$$r_{t,o^{(t,1)}} - r_{t,o^{(t,2)}}$$

t 's loyalty to $o^{(t,1)}$

Evaluation Setup.

4x4 port real-time switches

15x15 square grid network topology

Per port capacity: 1Gbps,

M = 2000 (cell/frame), 1 cell = 500 bit

10000 Trials, in each trial:

- Random number of multicast groups,
- w_i = 1~20 cell/frame (500K~10Mbps)

Evaluation Setup.

4x4 port real-time switches

15x15 square grid network topology

Per port capacity: 1Gbps,

M = 2000 (cell/frame), 1 cell = 500 bit

10000 Trials, in each trial:

- Random number of multicast groups,
- w_i = 1~20 cell/frame (500K~10Mbps)

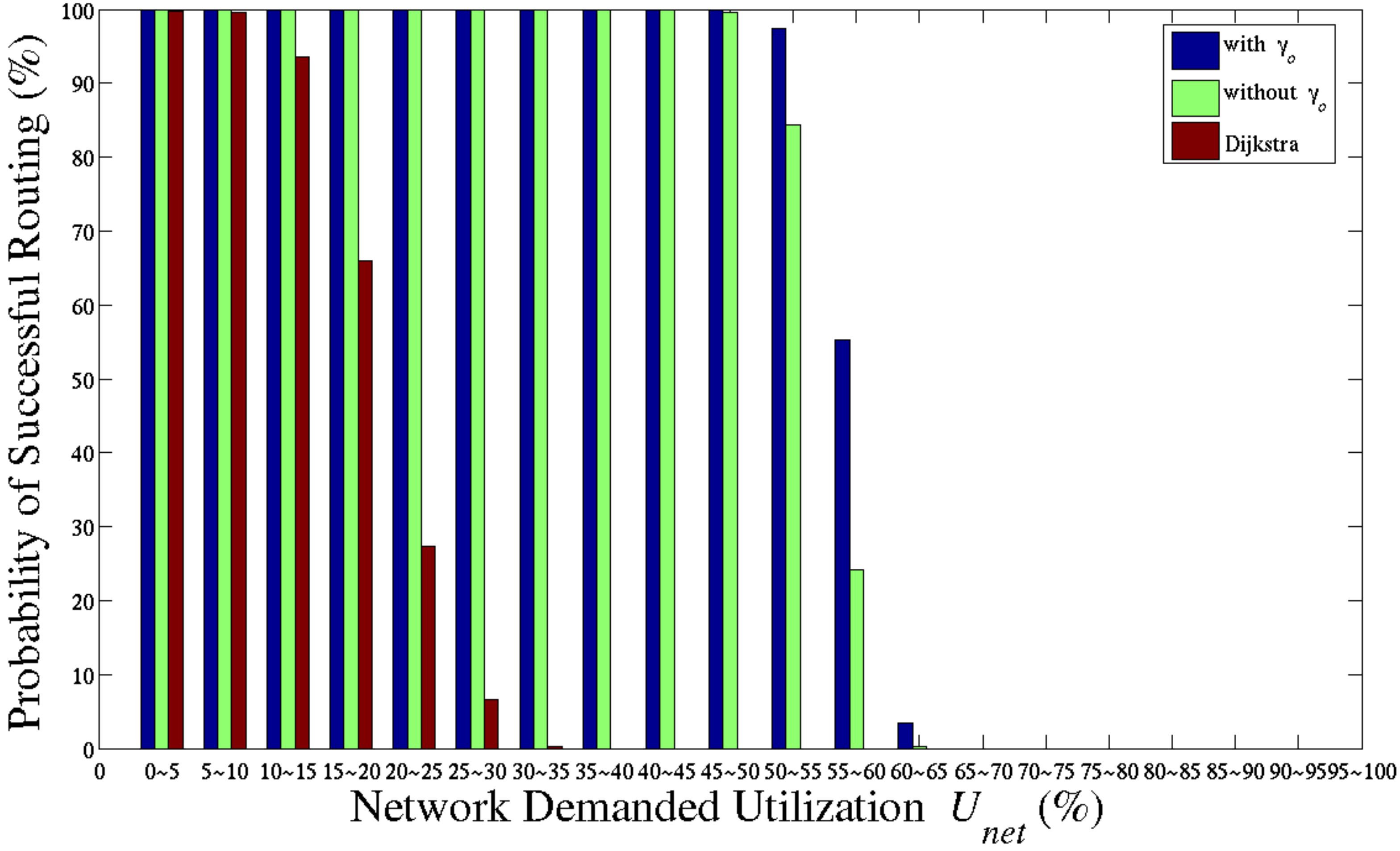
Evaluation Setup.

$$\tilde{H}_i = \eta H_i$$

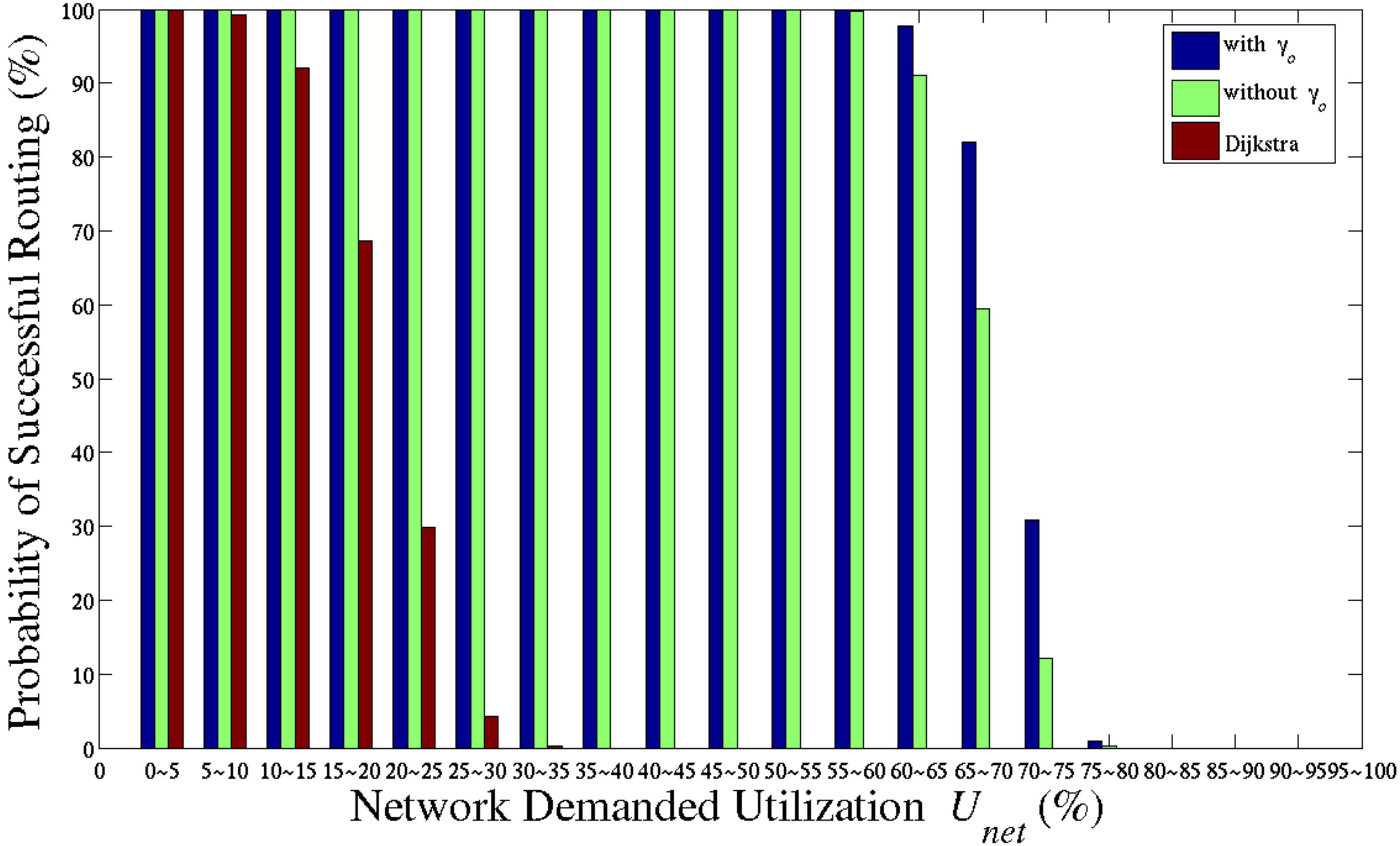
Network Demanded Utilization
(Application Layer E2E Utilization)

$$U_{net} \stackrel{def}{=} \frac{\sum_i (|D_i| w_i)}{|\bigcup_i \{D_i\}| M}$$

Evaluation Results: η=3



Evaluation Results: η=9



Mainstream Internet multicast routing algorithms mainly concern about dynamic distributed group management.

Reverse Path Broadcasting/Multicasting (RPB/RPM) [semeria97]

Truncated Reverse Path Broadcasting (TRPB) [semeria97]

Distance Vector Multicast Routing Protocol (DVMRP) [waitzman88]

Multicast Extension to Open Shortest Path First (MOSPF) [moy94a][moy94b]

Protocol Independent Multicast (PIM) [fenner06][adams05]

Core-Based Tree Multicast Routing (CBT) [ballardie97]

Mainstream Internet multicast routing algorithms become Dijkstra for static network with global info.

Reverse Path Broadcasting/Multicasting (RPB/RPM) [semeria97]

Truncated Reverse Path Broadcasting (TRPB) [semeria97]

Distance Vector Multicast Routing Protocol (DVMRP) [waitzman88]

Multicast Extension to Open Shortest Path First (MOSPF) [moy94a][moy94b]

Protocol Independent Multicast (PIM) [fenner06][adams05]

Core-Based Tree Multicast Routing (CBT) [ballardie97]

Others

P2P and Overlay Network Multicast are concerned with statistical performance instead of hard real-time E2E delay bound.

Thank You!

References

[adams05] A. Adams, J. Nicholas, and W. Siadak, Protocol Independent Multicast– Dense Mode (PIM-DM): Protocol Specification (Revised), RFC 3973, Jan., 2005.

[ballardie97] A. Ballardie, Core Based Tees (CBT) Multicast Routing Architectre, RF 2201, Spe., 1997.

[fenner06] B. Fenner, M. handley, H. Holbrook, and I. Kouvelas, Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised), RFC 4601, Aug., 2006.

[moy94a] J. Moy, Multicast Extensions to OSPF, RFC 1584, Mar., 1994.

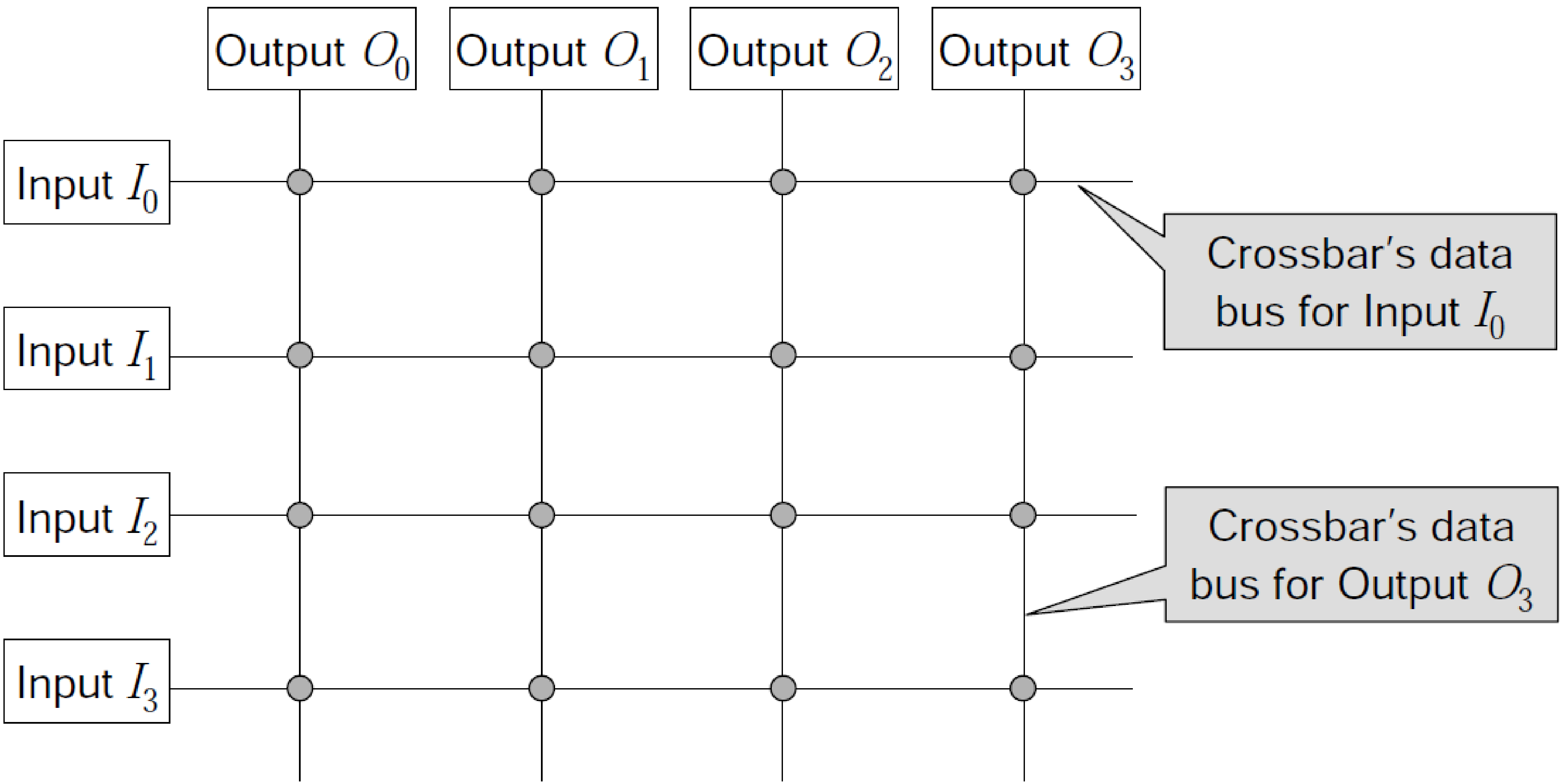
[moy94b] J. Moy, MOSPF: Analysis and Experience, RFC 1585, Mar., 1994.

[semeria97] C. Semeria and T. Maufer, Introduction to IP Multicast Routing, IETF draft-ietf-mboned-intro-multicast-03.txt, Jul., 1997.

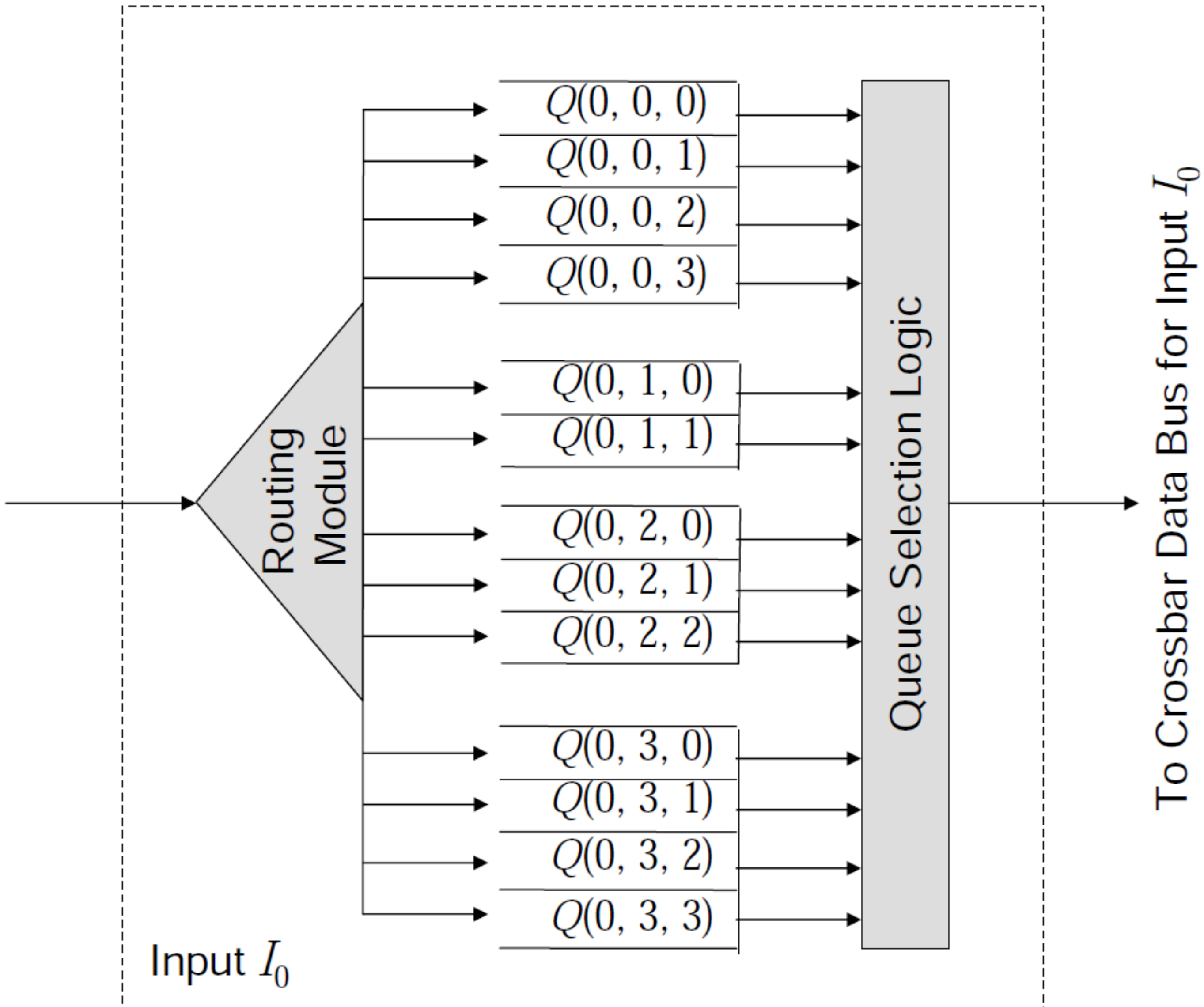
[waitzman88] D. Waitzman, C. Partridge, and S. Deering, Distance Vector Multicast Routing Protocol, RFC 1075, Nov., 1988.

[wang10] Q. Wang and S. Gopalakrishnan, Adapting a Main-Steam Internet Switch Architecture for Multi-Hop Real-Time Industrial Networks, in IEEE Transactions on Industrial Informatics, v6, n3, May, 2010.

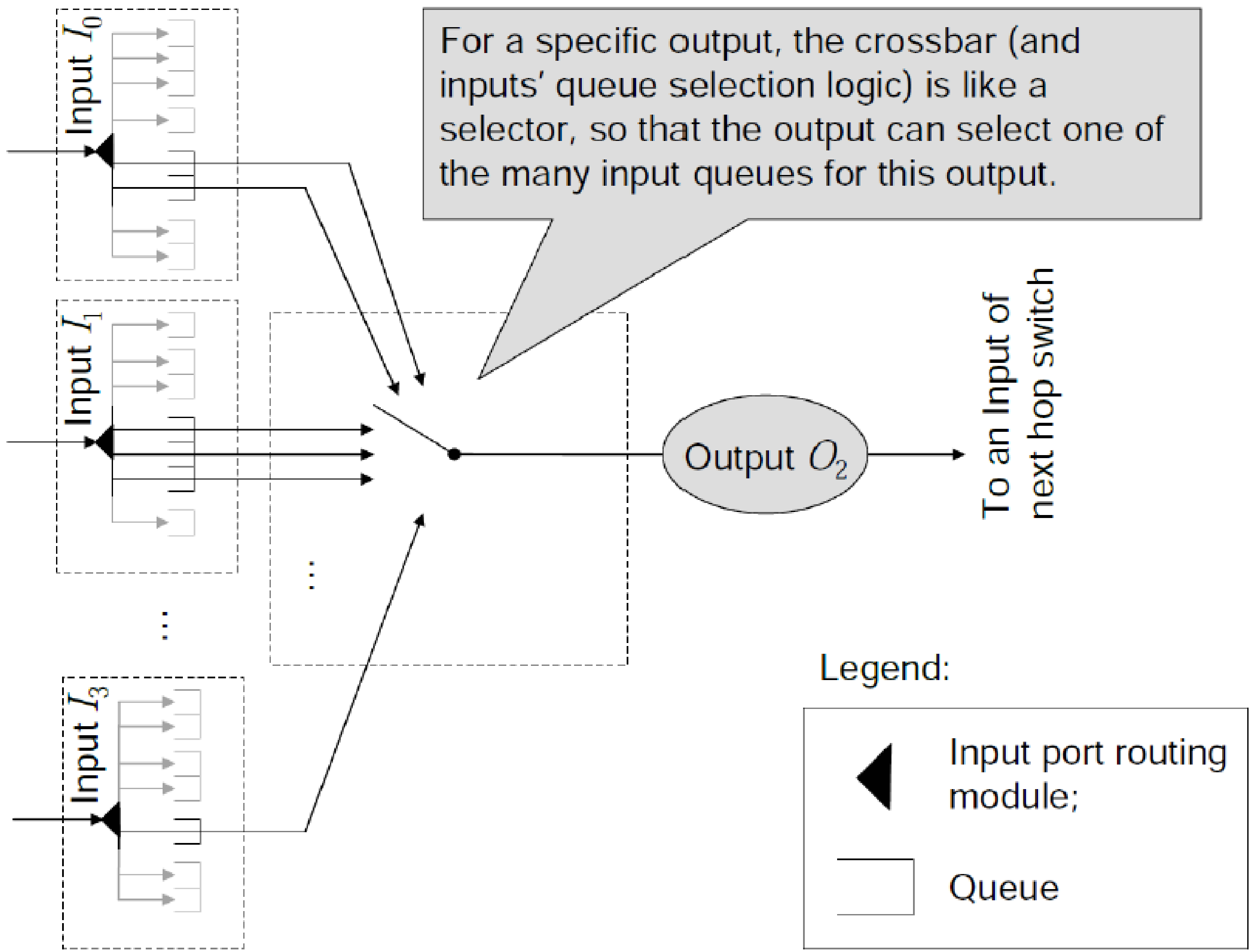
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



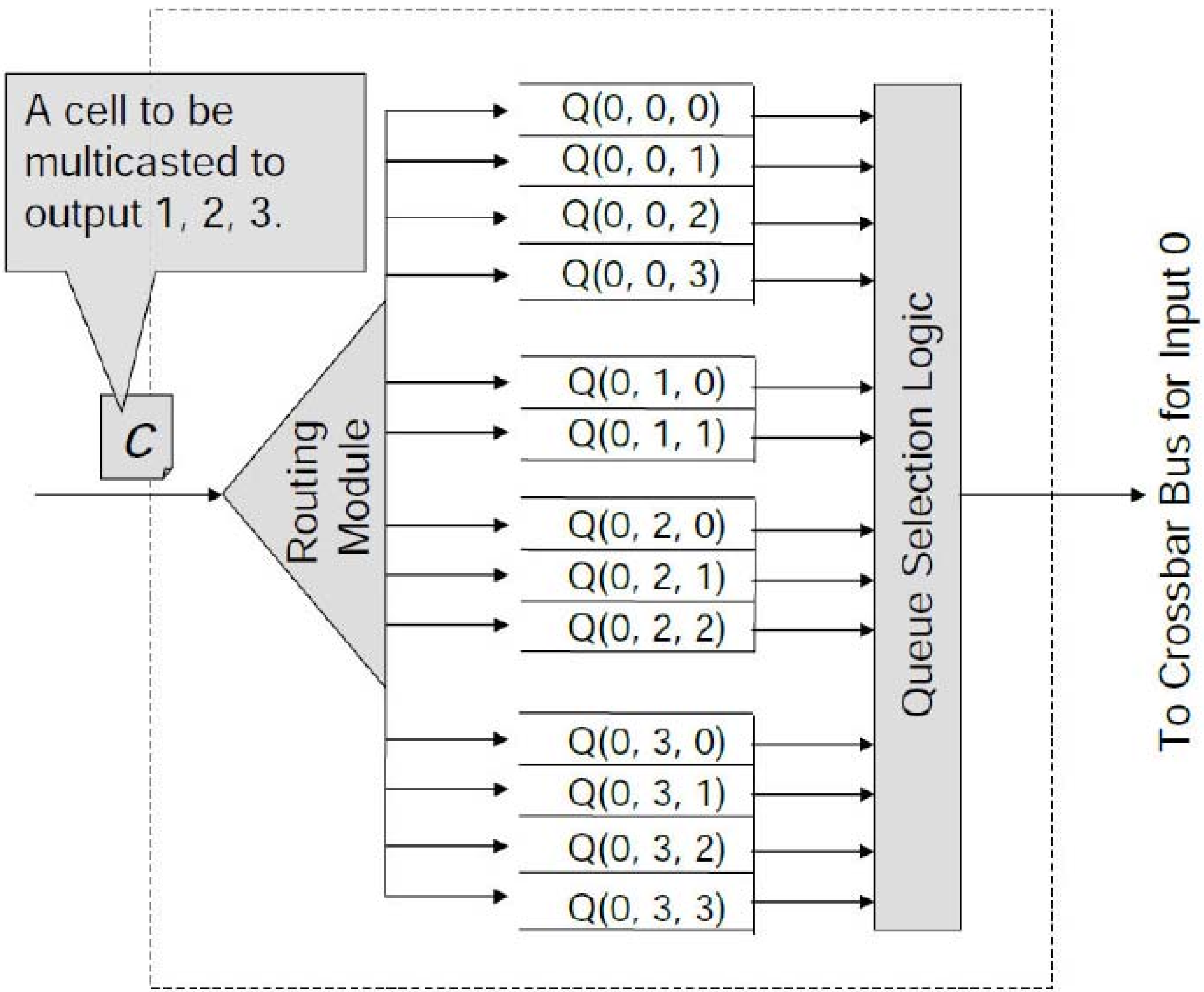
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



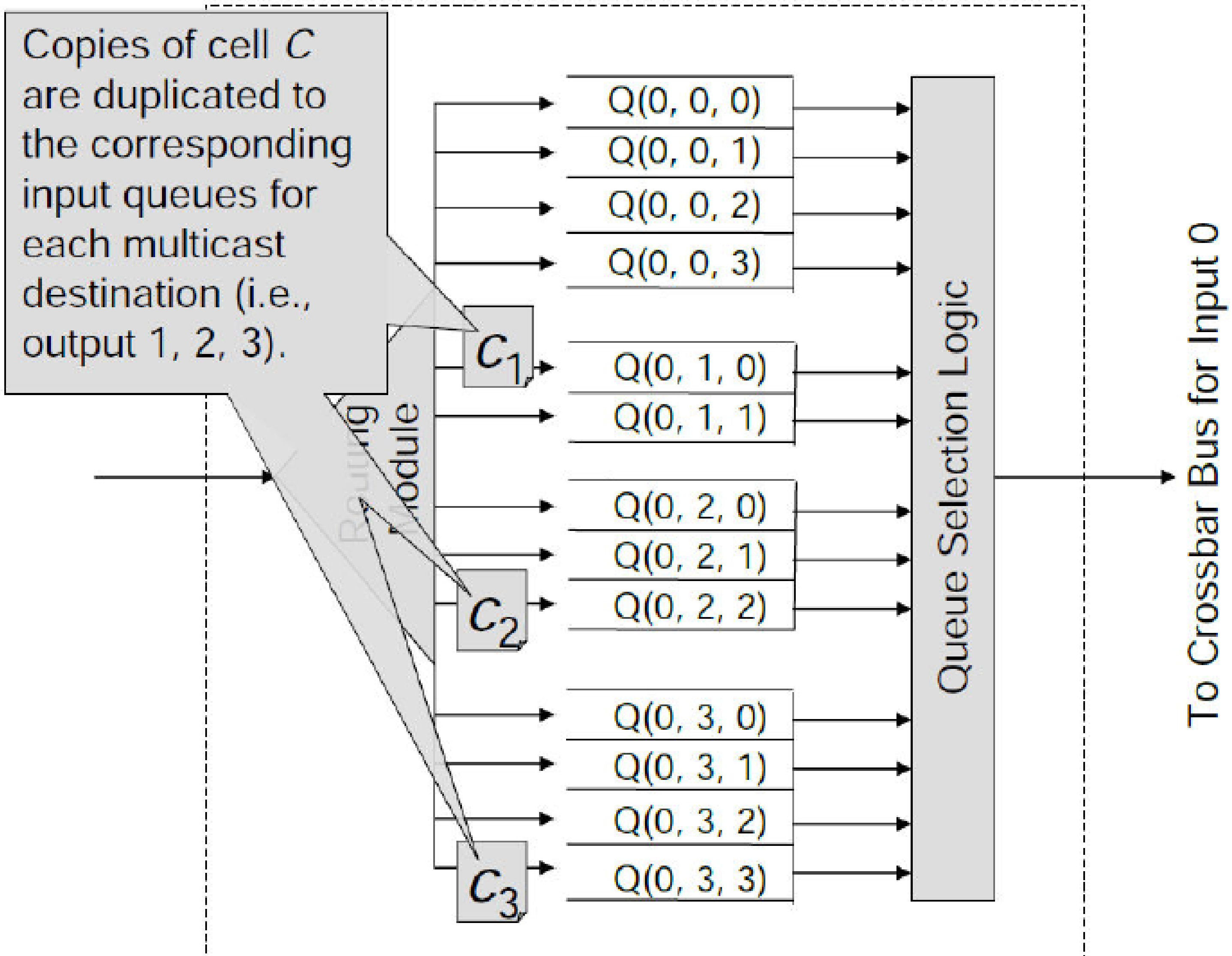
De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA



De facto standard (real-time) fieldbus switch architecture: crossbar per-flow-q TDMA