Olivier Beaumont, LaBRI
Arnaud Legrand, Loris Marchal et Yves Robert, ENS Lyon

Pipelining Broadcasts on Heterogeneous Platforms

Journée ORDO du GDR ARP - Jeudi 18 septembre 2003

Introduction

- Complex applications on grid environment require collective communication schemes:

  one to all Broadcast, Multicast, Scatter

  all to one Reduce

  all to all Gossip, All-to-All

- Numerous studies of a single communication scheme, mainly about one single broadcast

- Pipelining communications:
  - data parallelism involves a large amount of data
  - not a single communication, but a series of same communication schemes (e.g. a series of broadcasts from the same source)
  - maximize the throughput of the steady-state operation

Introduction

- Complex applications on grid environment require collective communication schemes:

  one to all Broadcast, Multicast, Scatter

  all to one Reduce

  all to all Gossip, All-to-All

- Numerous studies of a single communication scheme, mainly about one single broadcast
- Pipelining communications:
  - data parallelism involves a large amount of data
  - not a single communication, but a series of same communication schemes (e.g. a series of broadcasts from the same source)
  - maximize the throughput of the steady-state operation

Introduction

- Complex applications on grid environment require collective communication schemes:

  one to all Broadcast, Multicast, Scatter

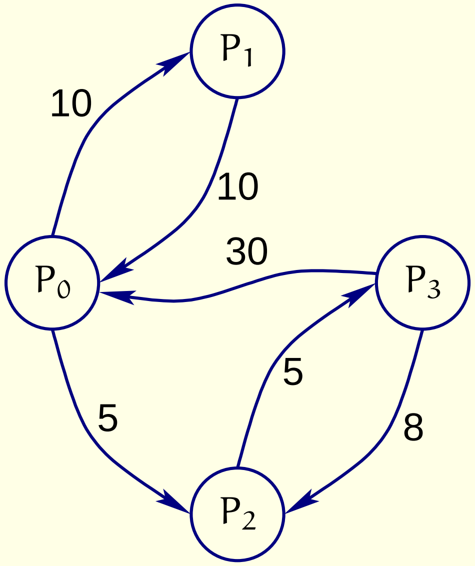
  all to one Reduce

  all to all Gossip, All-to-All

- Numerous studies of a single communication scheme, mainly about one single broadcast
- Pipelining communications:
  - data parallelism involves a large amount of data
  - not a single communication, but a series of same communication schemes (e.g. a series of broadcasts from the same source)
  - maximize the throughput of the steady-state operation

Framework of the platform

- $G = (P, E, c)$
- Let $P_1, P_2, \ldots, P_n$ be the n processors
- $(P_j, P_k) \in E$ denotes a communication link between P_i and P_j
- $c(P_j, P_k)$ denotes the time to transfer one unit message from P_j to P_k
- one-port for incoming communications
- one-port for outgoing communications



Pipelining Broadcasts

- Send n messages from P_0 to all other P_i's
- Let $T_{opt(n)}$ denote the optimal time for broadcasting the n messages
- Asymptotic optimality: $$\lim_{n \to +\infty} \frac{T_{alg}(n)}{T_{opt}(n)} = 1$$
- Usually, broadcast is done on a spanning tree
- What is the best broadcast throughput when using a single tree, a DAG, or a general graph?

Pipelining Broadcasts

- Send n messages from P_0 to all other P_i's
- Let $T_{opt(n)}$ denote the optimal time for broadcasting the n messages
- Asymptotic optimality: $\lim_{n \to +\infty} \frac{T_{alg}(n)}{T_{opt}(n)} = 1$
- Usually, broadcast is done on a spanning tree
- What is the best broadcast throughput when using a single tree, a DAG, or a general graph?

Pipelining Broadcasts

- Send n messages from P_0 to all other P_i's
- Let $T_{opt(n)}$ denote the optimal time for broadcasting the n messages
- Asymptotic optimality: $\lim_{n \to +\infty} \frac{T_{alg}(n)}{T_{opt}(n)} = 1$
- Usually, broadcast is done on a spanning tree
- What is the best broadcast throughput when using a single tree, a DAG, or a general graph?

Pipelining Broadcasts

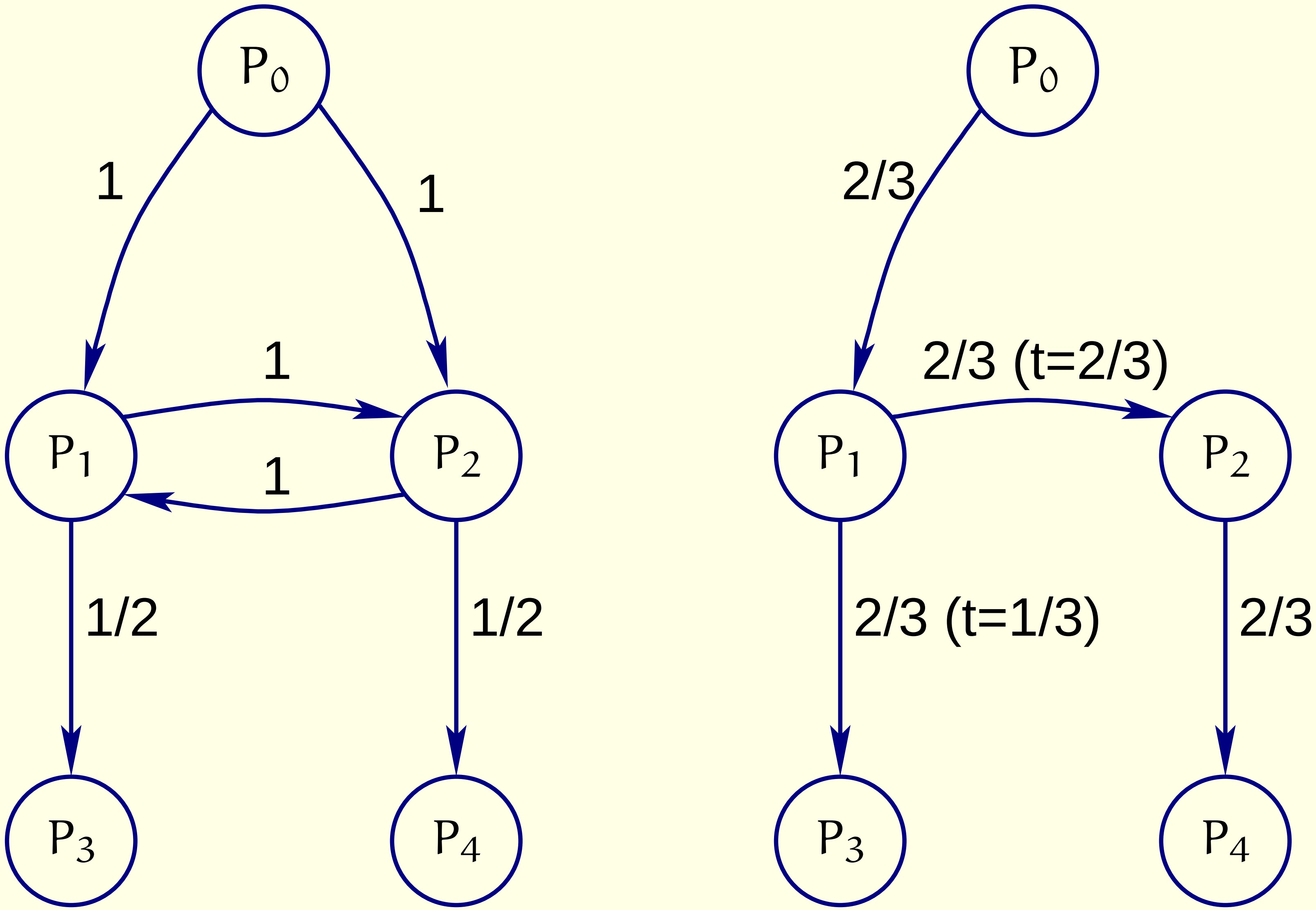
- Send n messages from P_0 to all other P_i's
- Let $T_{opt(n)}$ denote the optimal time for broadcasting the n messages
- Asymptotic optimality: $\displaystyle \lim_{n \to +\infty} \frac{T_{alg}(n)}{T_{opt}(n)} = 1$
- Usually, broadcast is done on a spanning tree
- What is the best broadcast throughput when using a single tree, a DAG, or a general graph?

Pipelining Broadcasts

- Send n messages from P_0 to all other P_i's
- Let $T_{opt(n)}$ denote the optimal time for broadcasting the n messages
- Asymptotic optimality: $$\lim_{n \to +\infty} \frac{T_{alg}(n)}{T_{opt}(n)} = 1$$
- Usually, broadcast is done on a spanning tree
- What is the best broadcast throughput when using a single tree, a DAG, or a general graph?

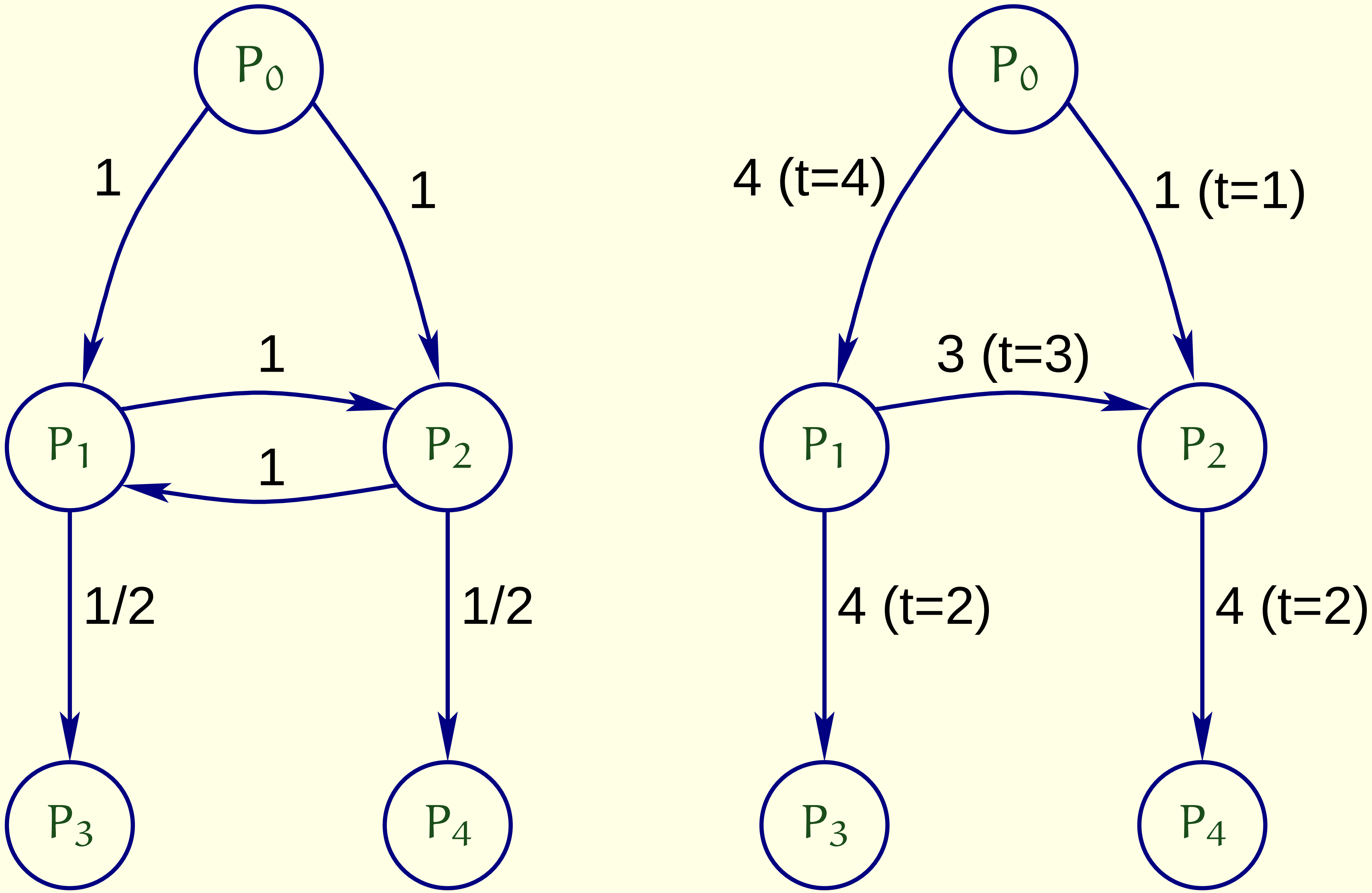
With a tree

The throughput with the best tree is 2 messages every 3 tops



With a DAG

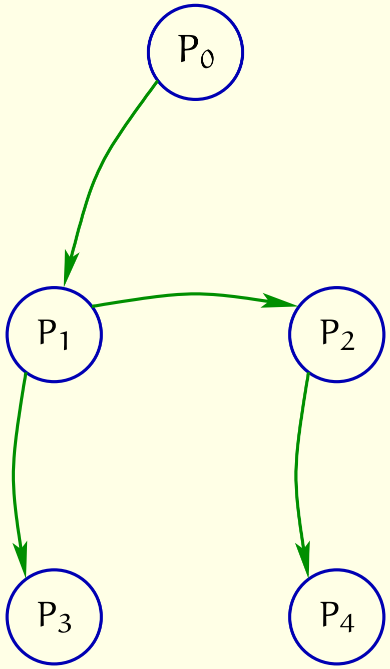
The throughput with the best DAG is 4 messages every 5 tops



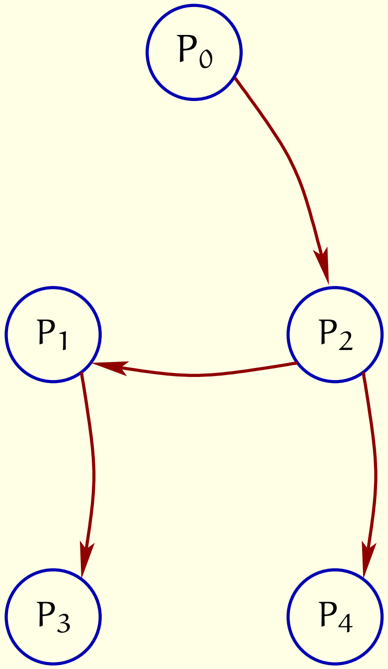
With a general graph

- Throughput with the best graph: 2 messages every 2 tops
- Two different sorts of messages (even/odd numbered)
- $m_1(i)$ denotes the message sent from P_0 to P_1 during period i
- $m_2(i)$ denotes the message sent from P_0 to P_2 during period i

path for m_1 messages



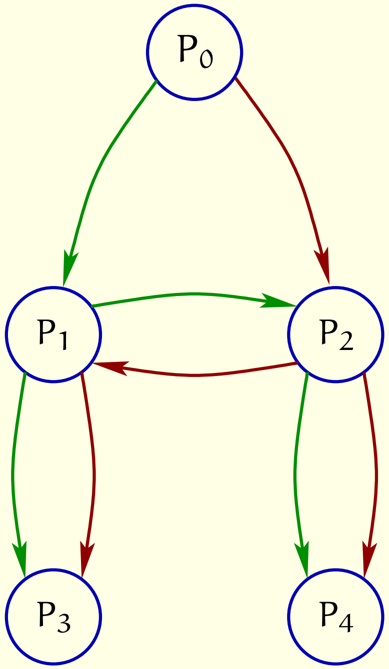
path for m_2 messages



With a general graph

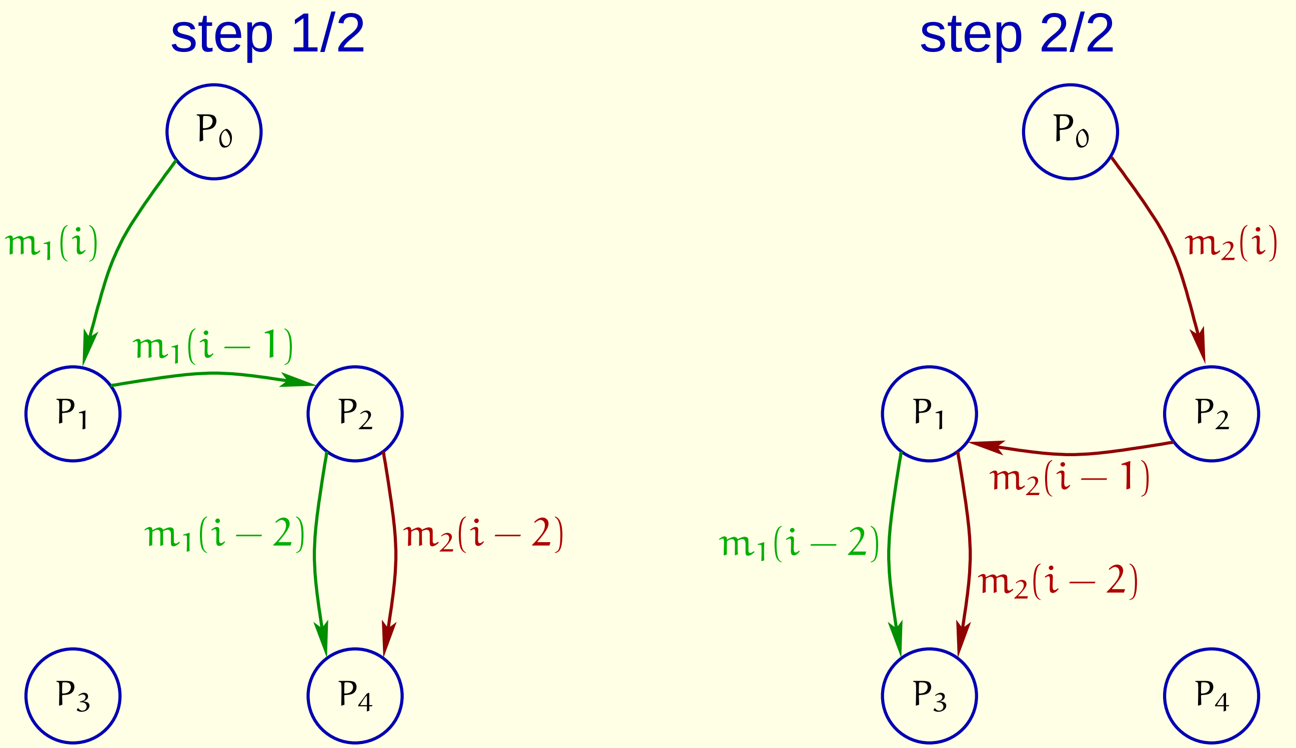
- Throughput with the best graph: 2 messages every 2 tops
- Two different sorts of messages (even/odd numbered)
- $m_1(i)$ denotes the message sent from P_0 to P_1 during period i
- $m_2(i)$ denotes the message sent from P_0 to P_2 during period i

all communications



With a general graph

- Throughput with the best graph: 2 messages every 2 tops
- Two different sorts of messages (even/odd numbered)
- $m_1(i)$ denotes the message sent from P_0 to P_1 during period i
- $m_2(i)$ denotes the message sent from P_0 to P_2 during period i



Problem Formalization

- Input: $G = (P, E, c)$
- Output:
  - The best throughput $\frac{p}{q}$
  - A "compact" description of the behiavior of the nodes.

During q time steps

- step 1: $P_{i_1}^{(1)}$ sends 1 mess to $P_{j_1}^{(1)}$
- step 1: $P_{i_2}^{(1)}$ sends 1 mess to $P_{j_2}^{(1)}$
- $\vdots$
- step q: $P_{i_n}^{(q)}$ sends 1 mess to $P_{j_n}^{(q)}$

This may not be polynomial since the size of the description is a priori of order $O(nq)$

During q time steps

- step 1: $P_{i_1}^{(1)}$ sends $\alpha_{i_1}^{(1)}$ mess to $P_{j_1}^{(1)}$
- step 1: $P_{i_2}^{(1)}$ sends $\alpha_{i_2}^{(1)}$ mess to $P_{j_2}^{(1)}$
- $\vdots$
- step q: $P_{i_n}^{(q)}$ sends $\alpha_{i_n}^{(q)}$ mess to $P_{j_n}^{(q)}$

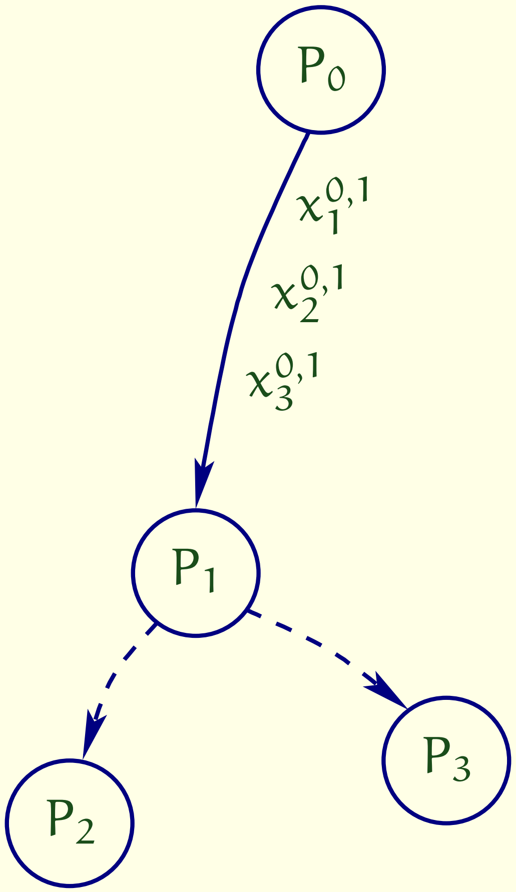
The size of such a description may be polynomial

Broadcast: Linear Program (1)

$x_i^{j,k}$ denotes the fraction of the message from P_0 to P_i that uses edge (P_j, P_k)

The conditions are

- $\forall i, \quad \sum x_i^{0,k} = 1$
- $\forall i, \quad \sum x_i^{j,i} = 1$
- $\forall j \neq 0, i, \quad \sum_k x_i^{j,k} = \sum_k x_i^{k,j}$



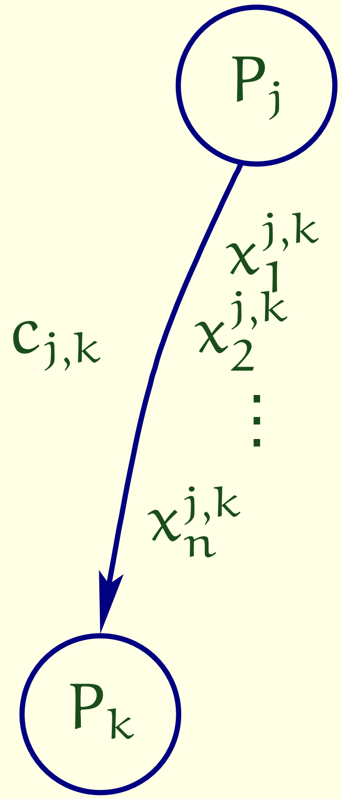
Broadcast: Linear Program (2)

$t_{j,k}$ denotes the time to transfer all the messages between P_j and P_k

- $t_{j,k} \leqslant \sum x_i^{j,k} c_{j,k}$????
- may be too pessimistic since $x_{i_1}^{j,k}$ and $x_{i_2}^{k,j}$ may be the same message
- not good for for a lower bound

or

- $\forall i, \quad t_{j,k} \leqslant x_i^{j,k} c_{j,k}$????
- may be too optimistic since it supposes that all the messages are sub-messages of the largest one
- OK for a lower bound, may not be feasible



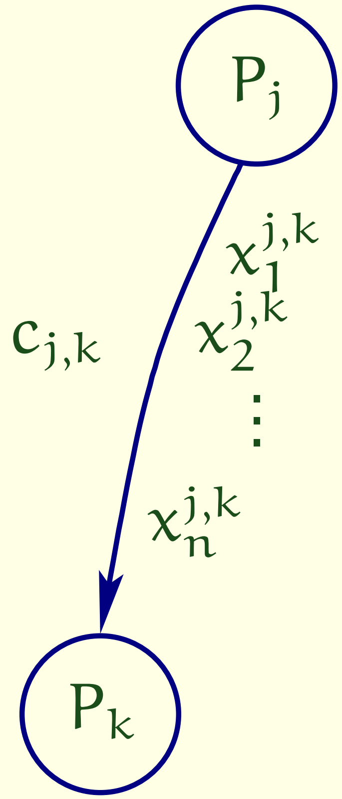
Broadcast: Linear Program (2)

$t_{j,k}$ denotes the time to transfer all the messages between P_j and P_k

- $t_{j,k} \leqslant \sum x_i^{j,k} c_{j,k}$????
- may be too pessimistic since $x_{i_1}^{j,k}$ and $x_{i_2}^{k,j}$ may be the same message
- not good for for a lower bound

or

- $\forall i, \quad t_{j,k} \leqslant x_i^{j,k} c_{j,k}$????
- may be too optimistic since it supposes that all the messages are sub-messages of the largest one
- OK for a lower bound, may not be feasible



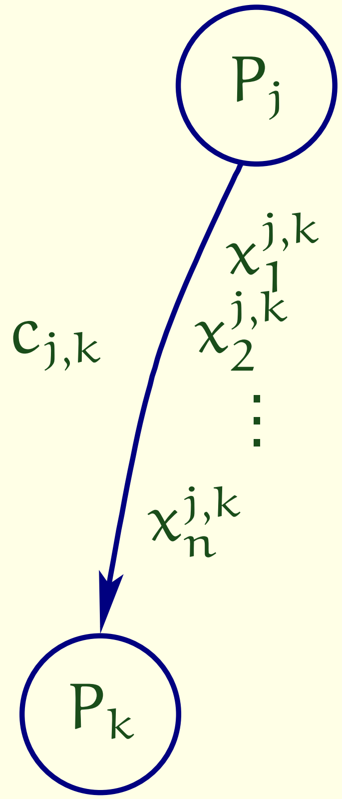
Broadcast: Linear Program (2)

$t_{j,k}$ denotes the time to transfer all the messages between P_j and P_k

- $t_{j,k} \leqslant \sum x_i^{j,k} c_{j,k}$????
- may be too pessimistic since $x_{i_1}^{j,k}$ and $x_{i_2}^{k,j}$ may be the same message
- not good for for a lower bound

or

- $\forall i, \quad t_{j,k} \leqslant x_i^{j,k} c_{j,k}$????
- may be too optimistic since it supposes that all the messages are sub-messages of the largest one
- OK for a lower bound, may not be feasible



Broadcast: Linear Program (3)

one-port model, during one time unit

- at most one sending operation: $\sum_{(P_j, P_k) \in E} t_{j,k} \leqslant t_j^{out}$
- at most one receiving operation: $\sum_{(P_k, P_j) \in E} t_{k,j} \leqslant t_j^{in}$

and at last,

- $\forall j, \quad t_j^{out} \leqslant t^{broadcast}$
- $\forall j, \quad t_j^{in} \leqslant t^{broadcast}$

Broadcast: Linear Program (3)

one-port model, during one time unit

- at most one sending operation: $\displaystyle\sum_{(P_j,P_k)\in E} t_{j,k} \leqslant t_j^{out}$
- at most one receiving operation: $\displaystyle\sum_{(P_k,P_j)\in E} t_{k,j} \leqslant t_j^{in}$

and at last,

- $\forall j, \quad t_j^{out} \leqslant t^{broadcast}$
- $\forall j, \quad t_j^{in} \leqslant t^{broadcast}$

Broadcast: Linear Program (4)

MINIMIZE $t^{broadcast}$,

SUBJECT TO

$$
\begin{cases}
\forall i, & \sum x_i^{0,k} = 1 \\
\forall i, & \sum x_i^{j,i} = 1 \\
\forall i,\ \forall j \neq 0, i, & \sum x_i^{j,k} = \sum x_i^{k,j} \\
\forall i, j, k & t_{j,k} \leqslant x_i^{j,k} c_{j,k} \\
\forall j, & \sum_{(P_j, P_k) \in E} t_{j,k} \leqslant t_j^{out} \\
\forall j, & \sum_{(P_k, P_j) \in E} t_{k,j} \leqslant t_j^{in} \\
\forall j, & t_j^{out} \leqslant t^{broadcast} \\
\forall j, & t_j^{in} \leqslant t^{broadcast}
\end{cases}
$$

A few remarks

- The linear program provides a lower bound for the broadcasting time of a unit-size divisible message
- It is not obvious that this lower bound is feasible since we considered that all the messages using the same communication link are sub-messages of the largest one.

Let us consider the multicast of a message:

- Some nodes do not need to receive the whole message
- We use the same inequalities but if P_i does not belong to the multicast set, then $\sum x_i^{0,k} = 1$ and $\sum x_i^{j,i} = 1$ are removed

A few remarks

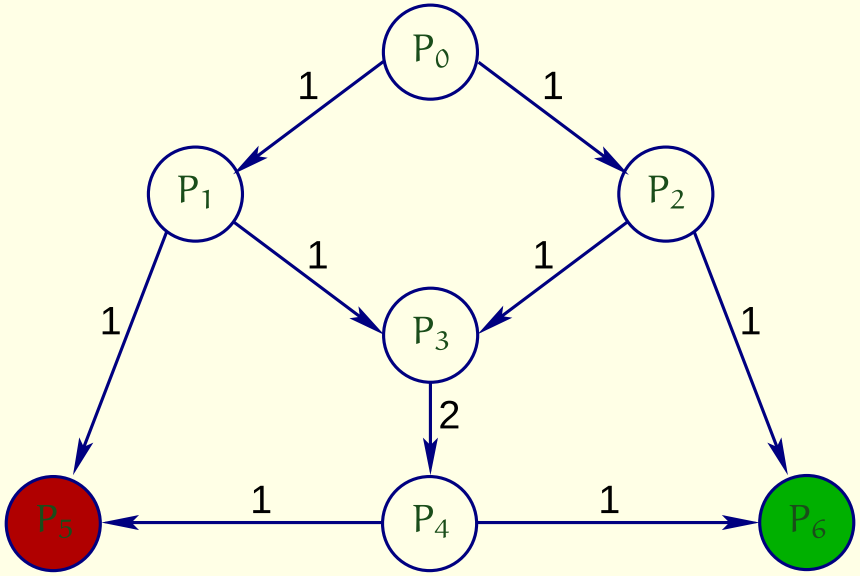
- The linear program provides a lower bound for the broadcasting time of a unit-size divisible message
- It is not obvious that this lower bound is feasible since we considered that all the messages using the same communication link are sub-messages of the largest one.

Let us consider the multicast of a message:

- Some nodes do not need to receive the whole message
- We use the same inequalities but if P_i does not belong to the multicast set, then $\sum x_i^{0,k} = 1$ and $\sum x_i^{j,i} = 1$ are removed

Lower Bound ??? Multicast Example (1)

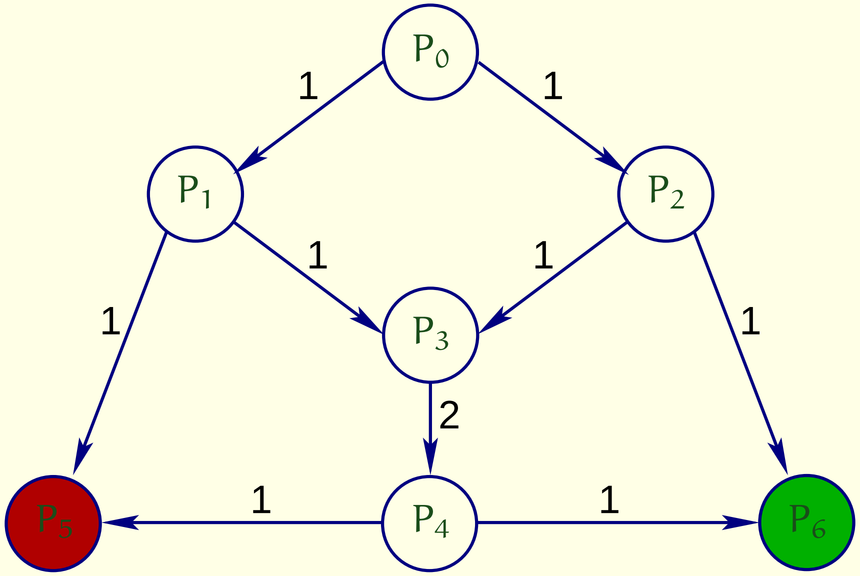
Consider the following platform, where the multicast set consists in the colored nodes:



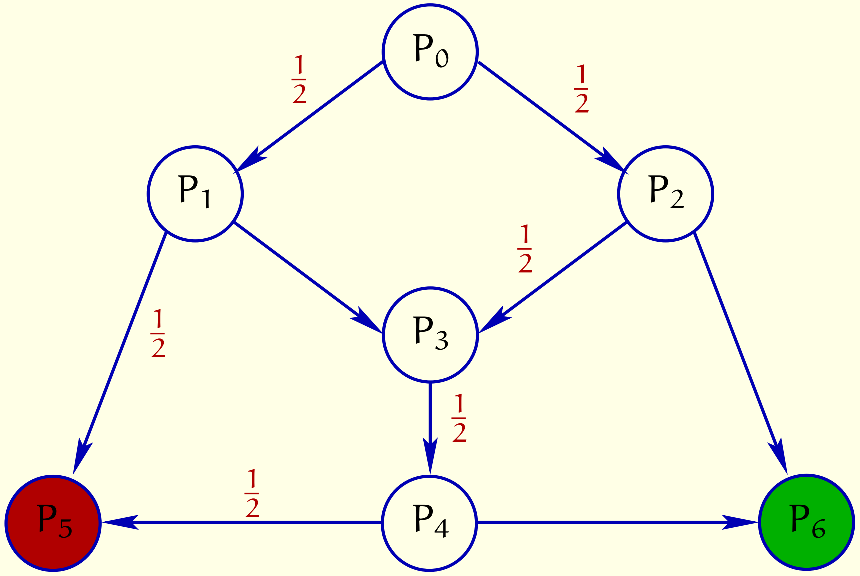
The linear program provides the following solution with throughput 1:

Lower Bound ??? Multicast Example (1)

Consider the following platform, where the multicast set consists in the colored nodes:

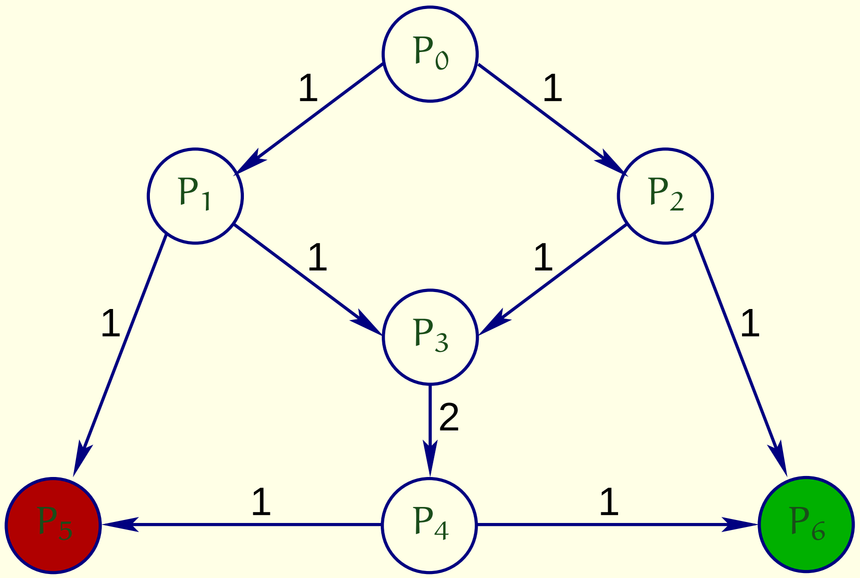


The linear program provides the following solution with throughput 1:

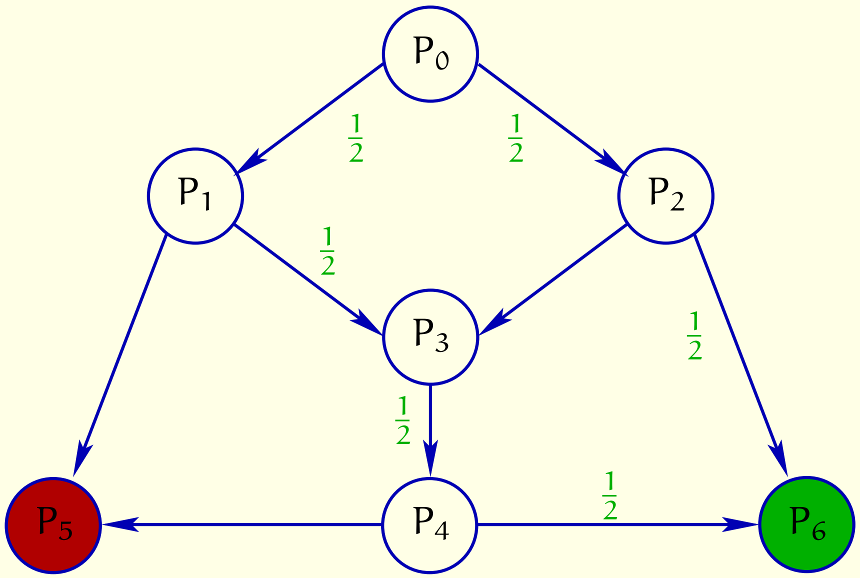


Lower Bound ??? Multicast Example (1)

Consider the following platform, where the multicast set consists in the colored nodes:

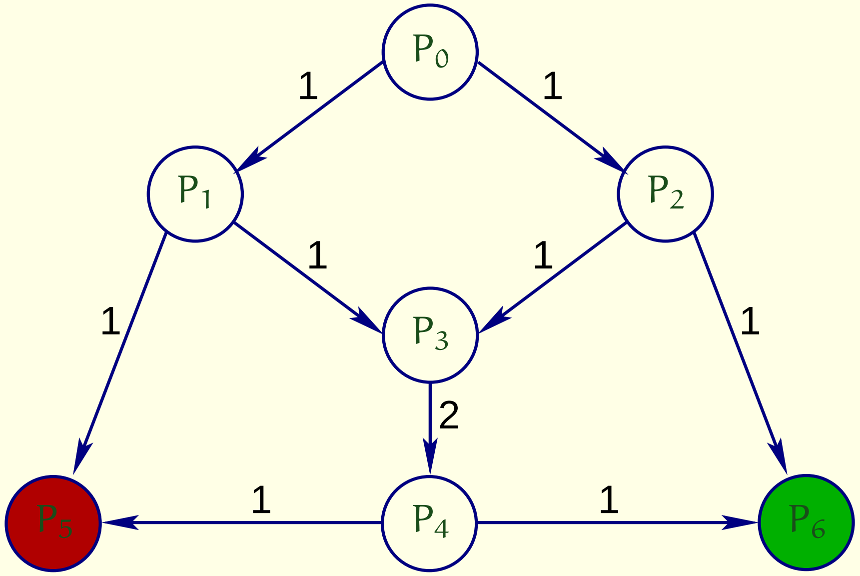


The linear program provides the following solution with throughput 1:

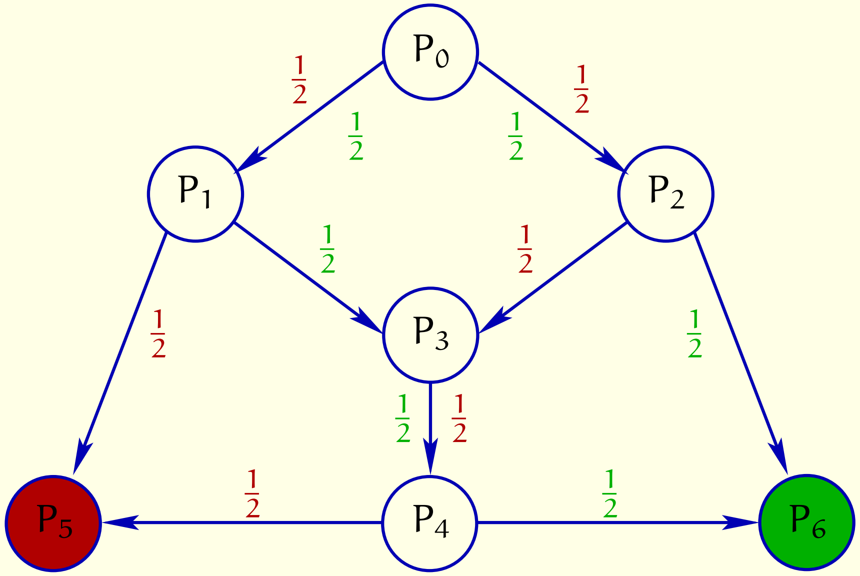


Lower Bound ??? Multicast Example (1)

Consider the following platform, where the multicast set consists in the colored nodes:

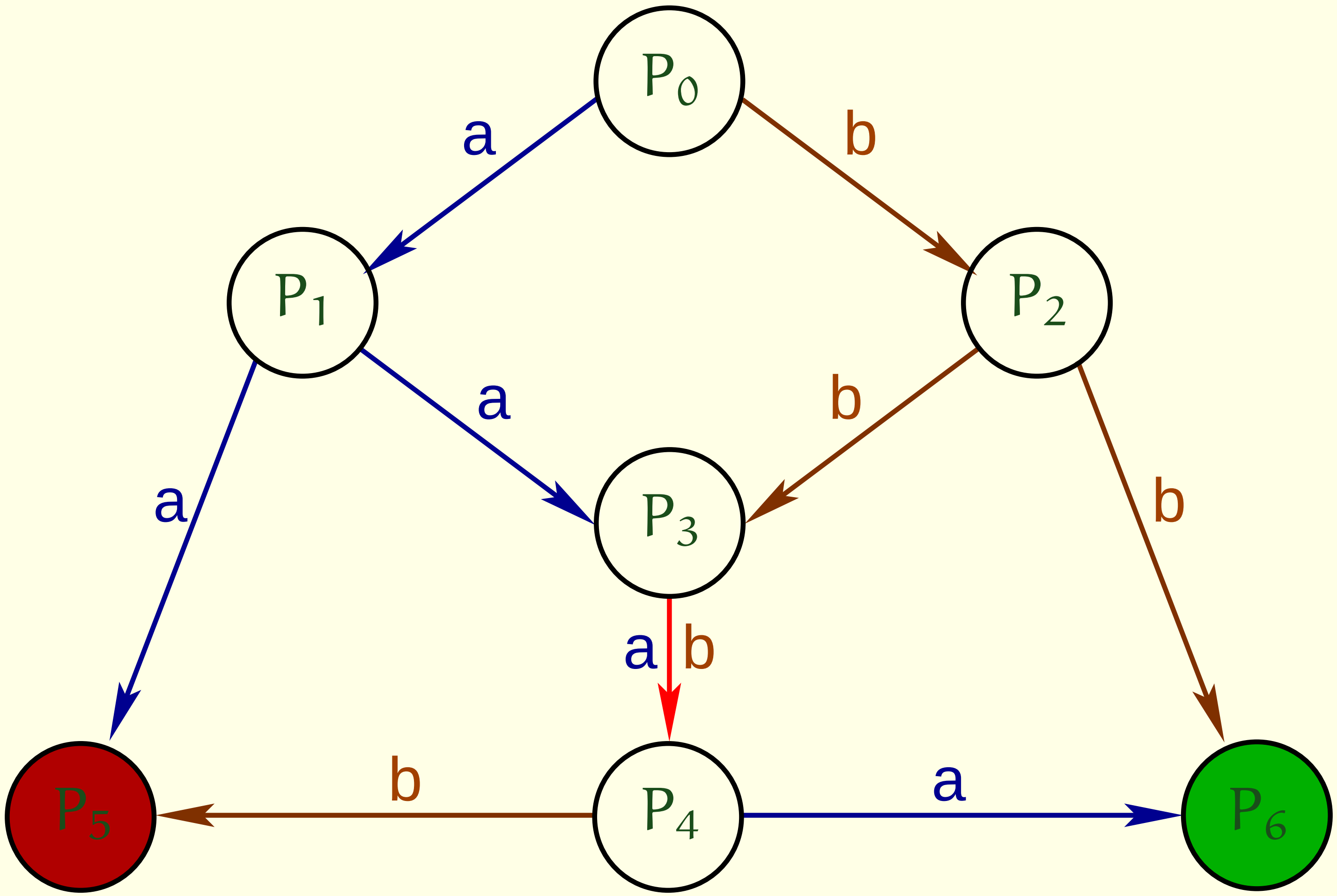


The linear program provides the following solution with throughput 1:



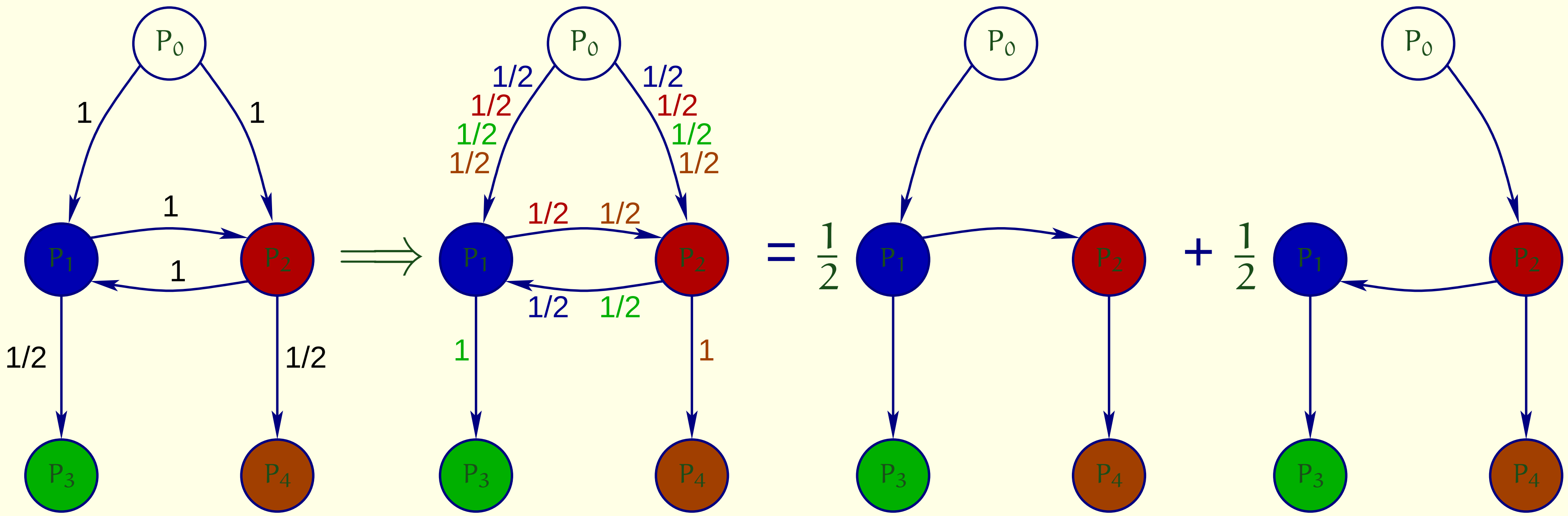
Lower Bound ??? Multicast Example (2)

Nevertheless, the obtained throughput is not feasible:



Lower Bound ??? Broadcast Example

For broadcast, the bound is nevertheless tight:



2 disjoint broadcast trees T_1 and T_2, of weight $\frac{1}{2} \Longrightarrow$ 1 message broacast at every top.

- How to find the trees ?
- How to keep the number of (weighted) trees relatively low ?

How many paths from P_0 to P_i (1)

$x_i^{j,k}$ denotes the fraction of the message from P_0 to P_i that uses edge (P_j, P_k)

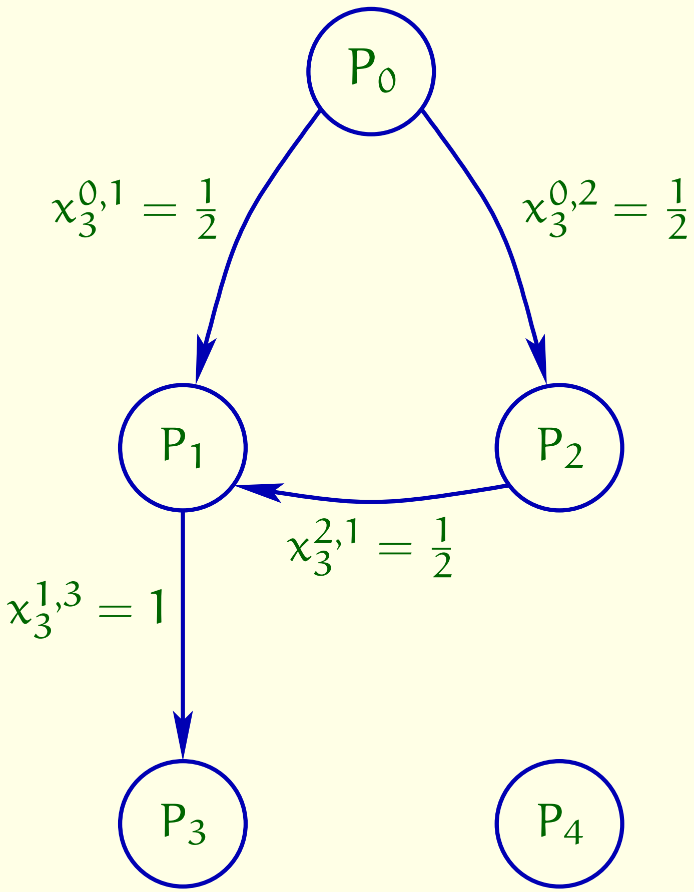
We know that

$$\begin{cases} \text{fraction of messages leaving } P_0 & \sum x_i^{0,k} = 1 \\ \text{fraction of messages arriving at } P_i & \sum x_i^{j,i} = 1 \\ \text{conservation law at } P_i \neq P_0,\ P_i & \sum x_i^{j,k} = \sum x_i^{k,j} \end{cases}$$

The x_i's define a flow in G of total weight 1.

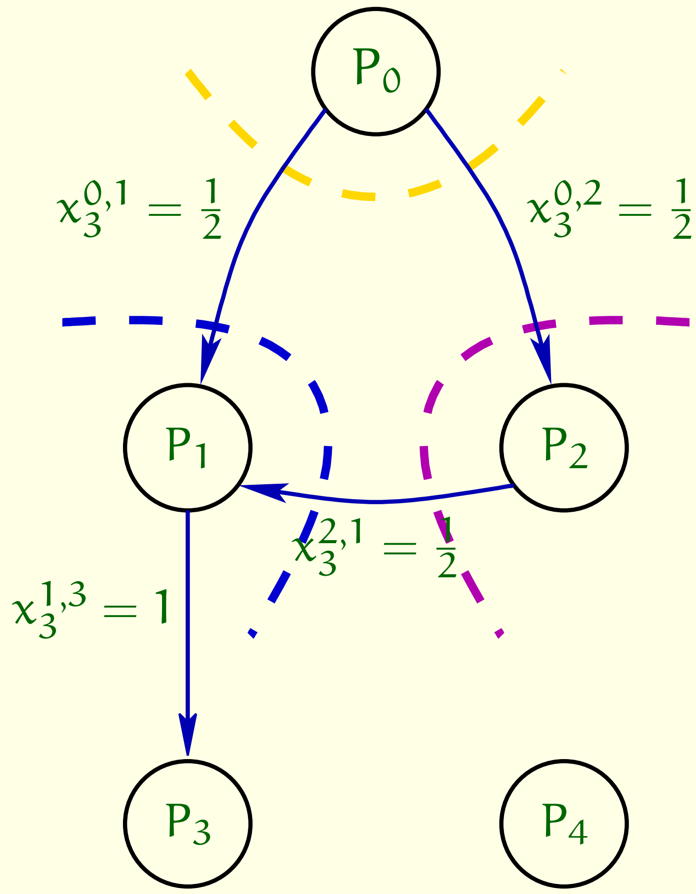
How many paths from P_0 to P_i (2)

- The x_3's define a flow in G of total weight 1
- In order to disconnect P_3 from P_0, a total weight of 1 has to be removed



How many paths from P_0 to P_i (2)

- The x_3's define a flow in G of total weight 1
- In order to disconnect P_3 from P_0, a total weight of 1 has to be removed



A nice graph theorem

- $c(P_0, P_i)$ miniumum weight to remove to disconnect = 1
- $c(P_0) = \min c(P_0, P_i)$ = 1
- $n_{j,k} = \max_i \left\{ x_i^{j,k} \right\}$ is the fraction of messages through (P_j, P_k).

Theorem 1. *(Weighted version of Edmond's branching Theorem) Given a directed weighted $G = (P, E, n)$, $P_0 \in P$ the source we can find P_0-arborescences $T_1, \ldots, T_k$ and weights $\lambda_1, \ldots, \lambda_k$ with $\sum \lambda_i \delta(T_i) \leqslant n$ with*

$$\sum \lambda_i = c(P_0) = 1,$$

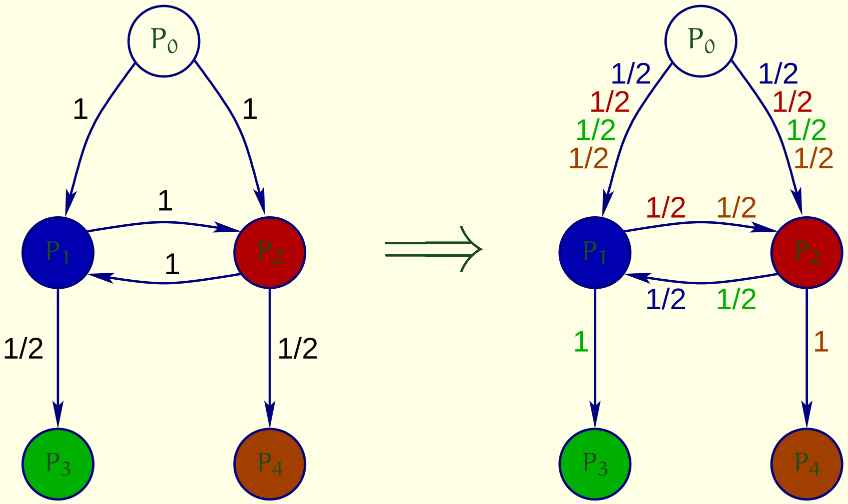
in strongly polynomial time, and $k \leqslant |E| + |V|^3$.

This theorem provides:

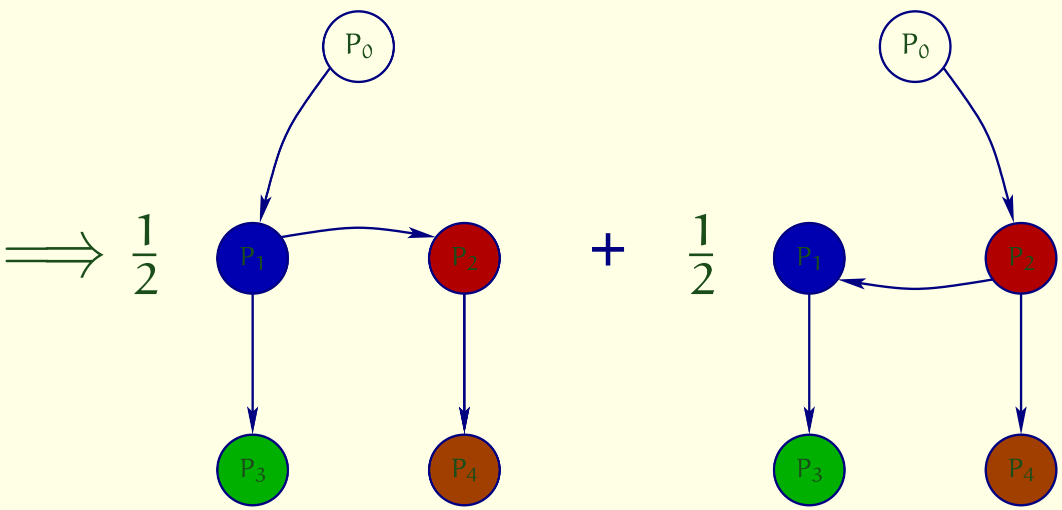
- the set of trees, their weights
- and the number of trees is "low": $\leqslant |E| + |V|^3$.

A nice graph theorem (2)

1. Linear program:



2. Schrijver's algorithm for weighted Edmond's theorem



Compact description of the solution?

- Period duration = 2 $(=\text{lcm(denominators tree coeff.)})$
- P_0 sends even-numbered messages to P_1 and odd-numbered messages to P_2
- Complete description for time-steps $2i$ and $2i+1$:
  - P_0 sends m_{2i} to P_1 and m_{2i+1} to P_2
  - P_1 sends m_{2i-2} (recvd. from P_0 at previous step) to P_2 and P_3
  - P_1 sends m_{2i-3} (recvd. from P_2 at previous step) to P_3
  - P_2 sends m_{2i-1} (recvd. from P_0 at previous step) to P_1 and P_4
  - P_2 sends m_{2i-4} (recvd. from P_1 at previous step) to P_4
- Solution size: number of communications within one period bounded by:

$$\begin{array}{cc} \text{number of trees} & \leqslant |E| + |V|^3 \\ \times & \\ \text{number of edges of one tree} & \leqslant |V| \end{array}$$

Compact description of the solution?

- Period duration = 2 $(= \text{lcm(denominators tree coeff.)})$
- P_0 sends even-numbered messages to P_1 and odd-numbered messages to P_2
- Complete description for time-steps $2i$ and $2i+1$:
  - P_0 sends m_{2i} to P_1 and m_{2i+1} to P_2
  - P_1 sends m_{2i-2} (recvd. from P_0 at previous step) to P_2 and P_3
  - P_1 sends m_{2i-3} (recvd. from P_2 at previous step) to P_3
  - P_2 sends m_{2i-1} (recvd. from P_0 at previous step) to P_1 and P_4
  - P_2 sends m_{2i-4} (recvd. from P_1 at previous step) to P_4
- Solution size: number of communications within one period bounded by:

$$\begin{array}{cc} \text{number of trees} & \leqslant |E| + |V|^3 \\ \times & \\ \text{number of edges of one tree} & \leqslant |V| \end{array}$$

Compact description of the solution?

- Period duration = 2 $(=$ lcm(denominators tree coeff.)$)$
- P_0 sends even-numbered messages to P_1 and odd-numbered messages to P_2
- Complete description for time-steps $2i$ and $2i+1$:
  - P_0 sends m_{2i} to P_1 and m_{2i+1} to P_2
  - P_1 sends m_{2i-2} (recvd. from P_0 at previous step) to P_2 and P_3
  - P_1 sends m_{2i-3} (recvd. from P_2 at previous step) to P_3
  - P_2 sends m_{2i-1} (recvd. from P_0 at previous step) to P_1 and P_4
  - P_2 sends m_{2i-4} (recvd. from P_1 at previous step) to P_4
- Solution size: number of communications within one period bounded by:

$$\text{number of trees} \leqslant |E| + |V|^3$$

$$\times$$

$$\text{number of edges of one tree} \leqslant |V|$$

Compact description of the solution?

- Period duration = 2 $(=$ lcm(denominators tree coeff.)$)$
- P_0 sends even-numbered messages to P_1 and odd-numbered messages to P_2
- Complete description for time-steps $2i$ and $2i+1$:
  - P_0 sends m_{2i} to P_1 and m_{2i+1} to P_2
  - P_1 sends m_{2i-2} (recvd. from P_0 at previous step) to P_2 and P_3
  - P_1 sends m_{2i-3} (recvd. from P_2 at previous step) to P_3
  - P_2 sends m_{2i-1} (recvd. from P_0 at previous step) to P_1 and P_4
  - P_2 sends m_{2i-4} (recvd. from P_1 at previous step) to P_4
- Solution size: number of communications within one period bounded by:

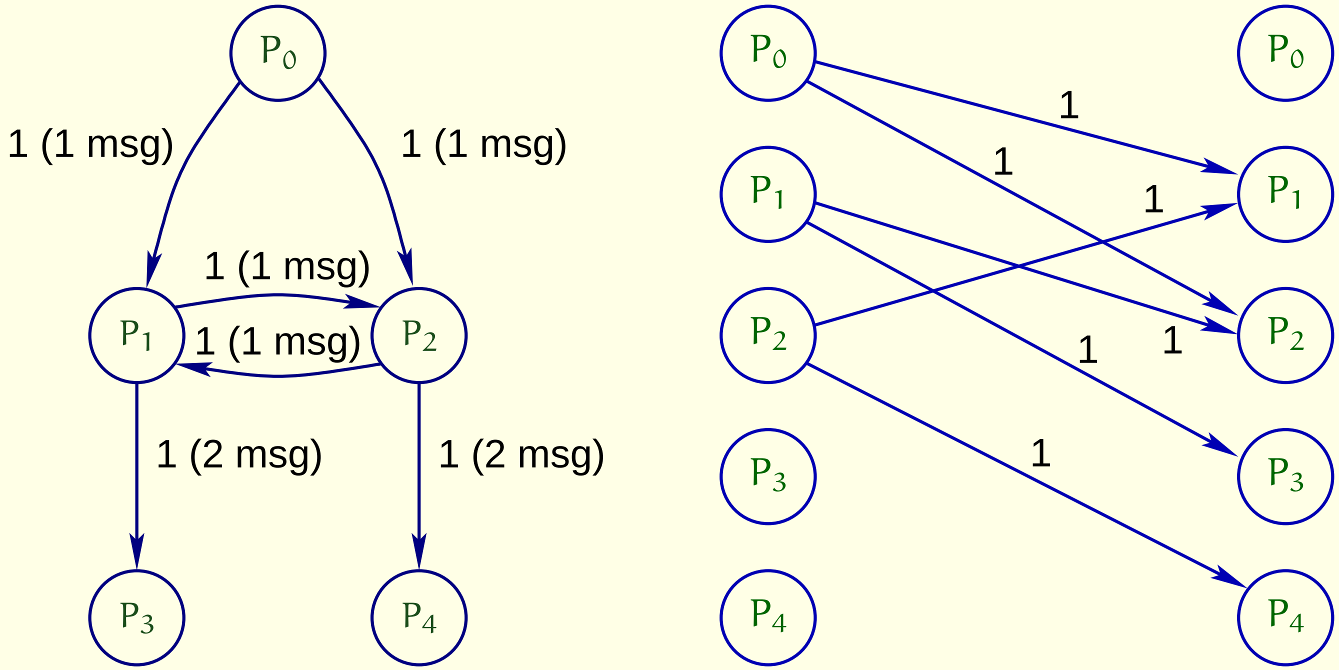
number of trees $\leqslant |E| + |V|^3$

$\times$

number of edges of one tree $\leqslant |V|$

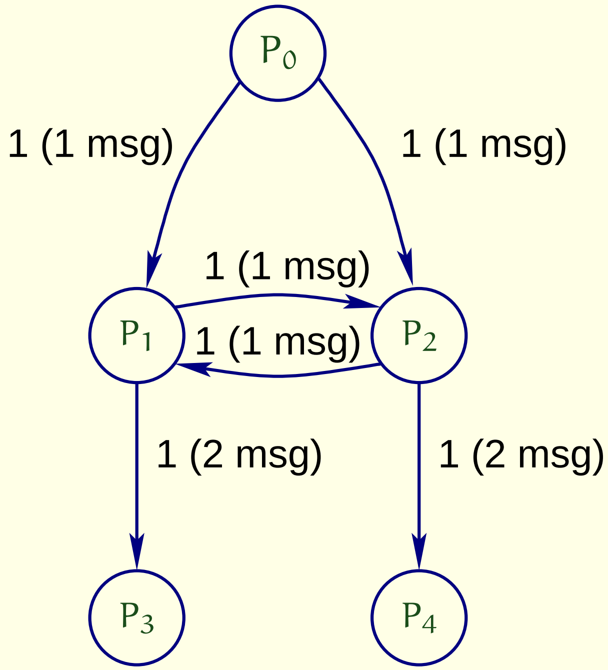
From local to global (1)

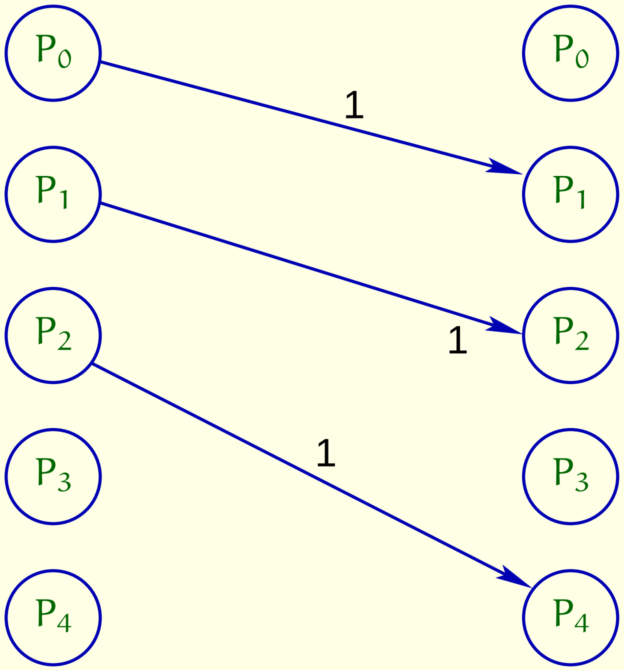
1. Set of communications to execute within period T
2. One-port equations $\longrightarrow$ local constraints
3. Pairwise-disjoint communications to be scheduled simultaneously
   $\Rightarrow$ extract a collection of matchings



From local to global (1)

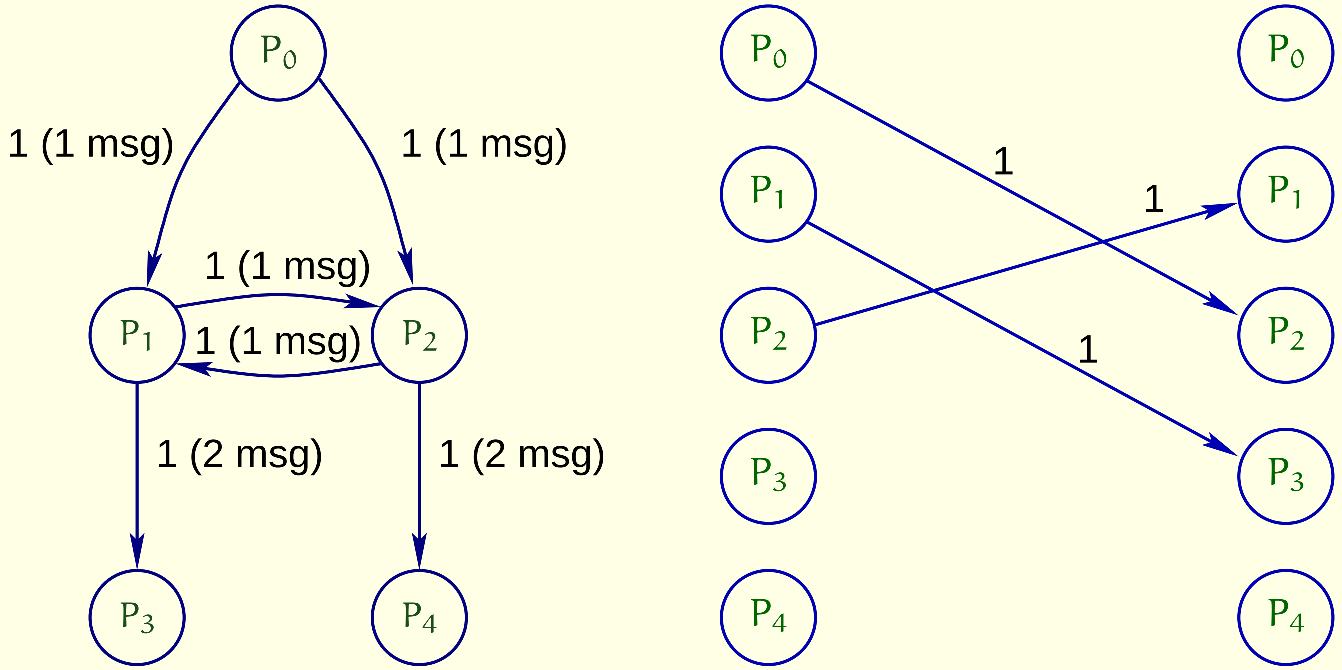
1. Set of communications to execute within period T
2. One-port equations $\longrightarrow$ local constraints
3. Pairwise-disjoint communications to be scheduled simultaneously
   $\Rightarrow$ extract a collection of matchings





From local to global (1)

1. Set of communications to execute within period T
2. One-port equations $\longrightarrow$ local constraints
3. Pairwise-disjoint communications to be scheduled simultaneously
   $\Rightarrow$ extract a collection of matchings



From local to global (2)

Solution

- Peel off bipartite communication graph
- **Idea 1** Split each communication of length L into L communications of length 1 and use König's edge-coloring algorithm (but not polynomial)
- **Idea 2** Use Schrijver's weighted edge-coloring algorithm:
  - extract a matching and substract maximum weight from participating edges
  - zero out at least one edge for each matching
  - strongly polynomial

From local to global (2)

Solution

- Peel off bipartite communication graph
- **Idea 1** Split each communication of length L into L communications of length 1 and use König's edge-coloring algorithm (but not polynomial)
- **Idea 2** Use Schrijver's weighted edge-coloring algorithm:
  - extract a matching and substract maximum weight from participating edges
  - zero out at least one edge for each matching
  - strongly polynomial

From local to global (2)

Solution

- Peel off bipartite communication graph
- **Idea 1** Split each communication of length L into L communications of length 1 and use König's edge-coloring algorithm (but not polynomial)
- **Idea 2** Use Schrijver's weighted edge-coloring algorithm:
  - extract a matching and substract maximum weight from participating edges
  - zero out at least one edge for each matching
  - strongly polynomial

Conclusion

Complexity of steady-state problems

Ask biased question:

Can we determine best throughput and characterize a solution achieving it, all that in polynomial time?

1. Broadcast: yes
2. Multicast: no, NP-complete
3. Scatter: yes (easier)
4. Reduce: yes (complicated too)

Makespan minimization versus throughput

Everything NP-hard.

Conclusion

Complexity of steady-state problems

Ask biased question:

Can we determine best throughput and characterize a solution achieving it, all that in polynomial time?

1. Broadcast: yes
2. Multicast: no, NP-complete
3. Scatter: yes (easier)
4. Reduce: yes (complicated too)

Makespan minimization versus throughput

Everything NP-hard.