Modeling Communications in Current Platforms

Master 2 Research Lecture: Parallel Systems

Vincent Danjean, MCF UJF, LIG/INRIA/Moais
Derick Kondo, CR INRIA, LIG/INRIA/Mescal
Arnaud Legrand, CR CNRS, LIG/INRIA/Mescal
Jean-François Méhaut, PR UJF, LIG/INRIA/Mescal
Bruno Raffin, CR INRIA, LIG/INRIA/Moais
Jean-Louis Roch, MCF ENSIMAG, LIG/INRIA/Moais
Alexandre Termier, MCF UJF, LIG/Hadas

LIG laboratory, arnaud.legrand@imag.fr

October 13, 2008

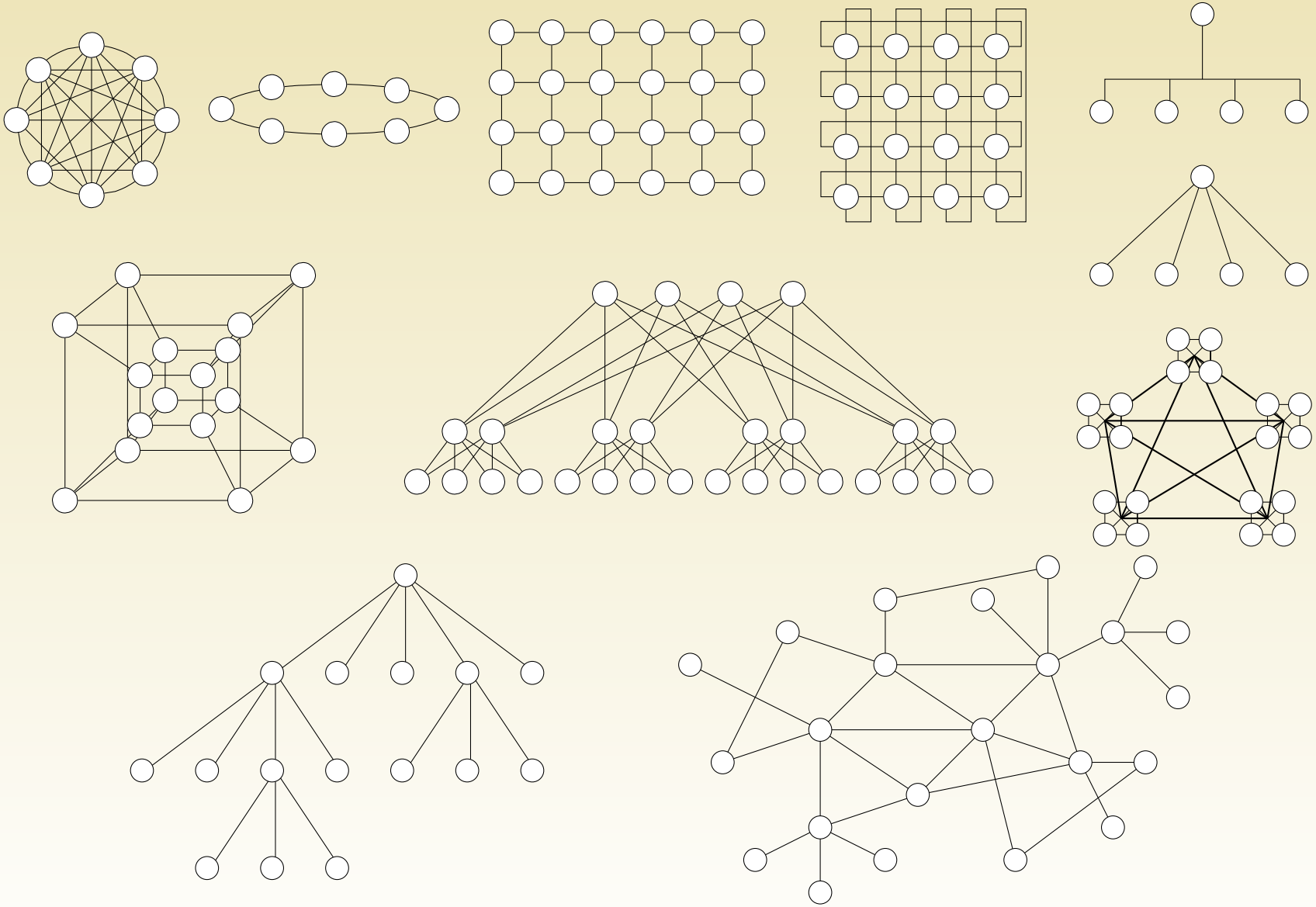
Motivation

- Scientific computing : large needs in computation or storage resources.
- Need to use systems with "several processors":
  - Parallel computers with shared/distributed memory
  - Clusters
  - Heterogeneous clusters
  - Clusters of clusters
  - Network of workstations
  - The Grid
  - Desktop Grids
- When modeling platform, communications modeling seems to be the most controversial part.
- Two kinds of people produce communication models: those who are concerned with scheduling and those who are concerned with performance evaluation.
- All these models are imperfect and intractable.

Outline

Outline

Various Topologies Used in the Litterature



Outline

UET-UCT

Hem. . . This one is mainly used by scheduling theoreticians to prove that their problem is hard and to know whether there is some hope to prove some clever result or not.

"Hockney" Model

Hockney [Hoc94] proposed the following model for performance evaluation of the Paragon. A message of size m from P_i to P_j requires:

$$t_{i,j}(m) = L_{i,j} + m/B_{i,j}$$

In scheduling, there are three types of "corresponding" models:

- Communications are not "splitable" and each communication k is associated to a communication time t_k (accounting for message size, latency, bandwidth, middleware, ...).
- Communications are "splitable" but latency is considered to be negligible (linear divisible model):

$$t_{i,j}(m) = m/B_{i,j}$$

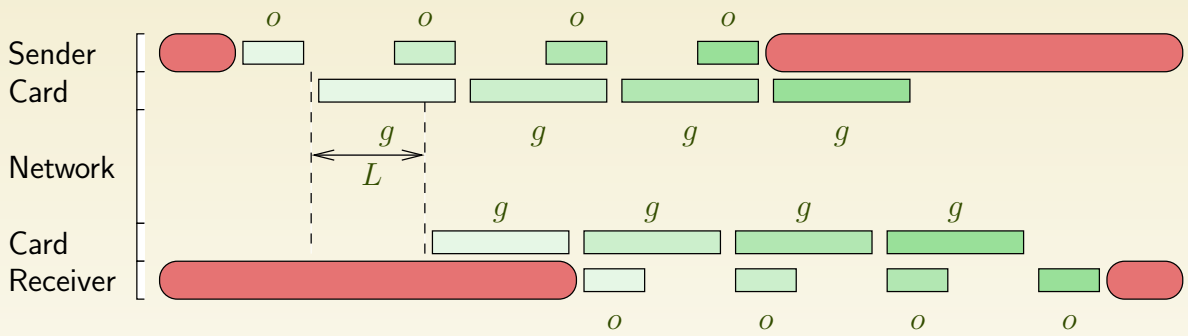
- Communications are "splitable" and latency cannot be neglected (linear divisible model):

$$t_{i,j}(m) = L_{i,j} + m/B_{i,j}$$

LogP

The LogP model [CKP+96] is defined by 4 parameters:

- L is the network latency
- o is the middleware overhead (message splitting and packing, buffer management, connection, ...) for a message of size w
- g is the gap (the minimum time between two packets communication) between two messages of size w
- P is the number of processors/modules



- Sending m bytes with packets of size w:

$$2o + L + \left\lceil \frac{m}{w} \right\rceil \cdot \max(o, g)$$

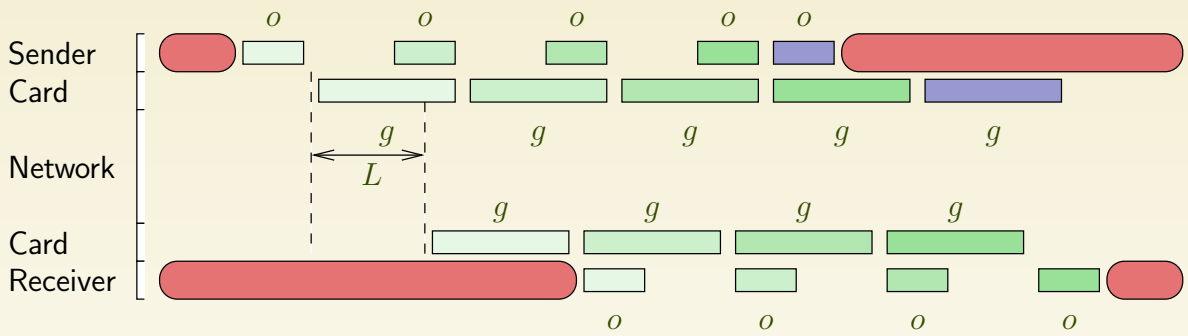
- Occupation on the sender and on the receiver:

$$o + L + \left(\left\lceil \frac{m}{w} \right\rceil - 1 \right) \cdot \max(o, g)$$

LogP

The LogP model [CKP+96] is defined by 4 parameters:

- L is the network latency
- o is the middleware overhead (message splitting and packing, buffer management, connection, ...) for a message of size w
- g is the gap (the minimum time between two packets communication) between two messages of size w
- P is the number of processors/modules



- Sending m bytes with packets of size w:

$$2o + L + \left\lceil \frac{m}{w} \right\rceil \cdot \max(o, g)$$

- Occupation on the sender and on the receiver:

$$o + L + \left(\left\lceil \frac{m}{w} \right\rceil - 1 \right) \cdot \max(o, g)$$

LogGP & pLogP

The previous model works fine for short messages. However, many parallel machines have special support for long messages, hence a higher bandwidth. LogGP [AISS97] is an extension of LogP:
G captures the bandwidth for long messages:

short messages $2o + L + \left\lceil \frac{m}{w} \right\rceil \cdot \max(o, g)$

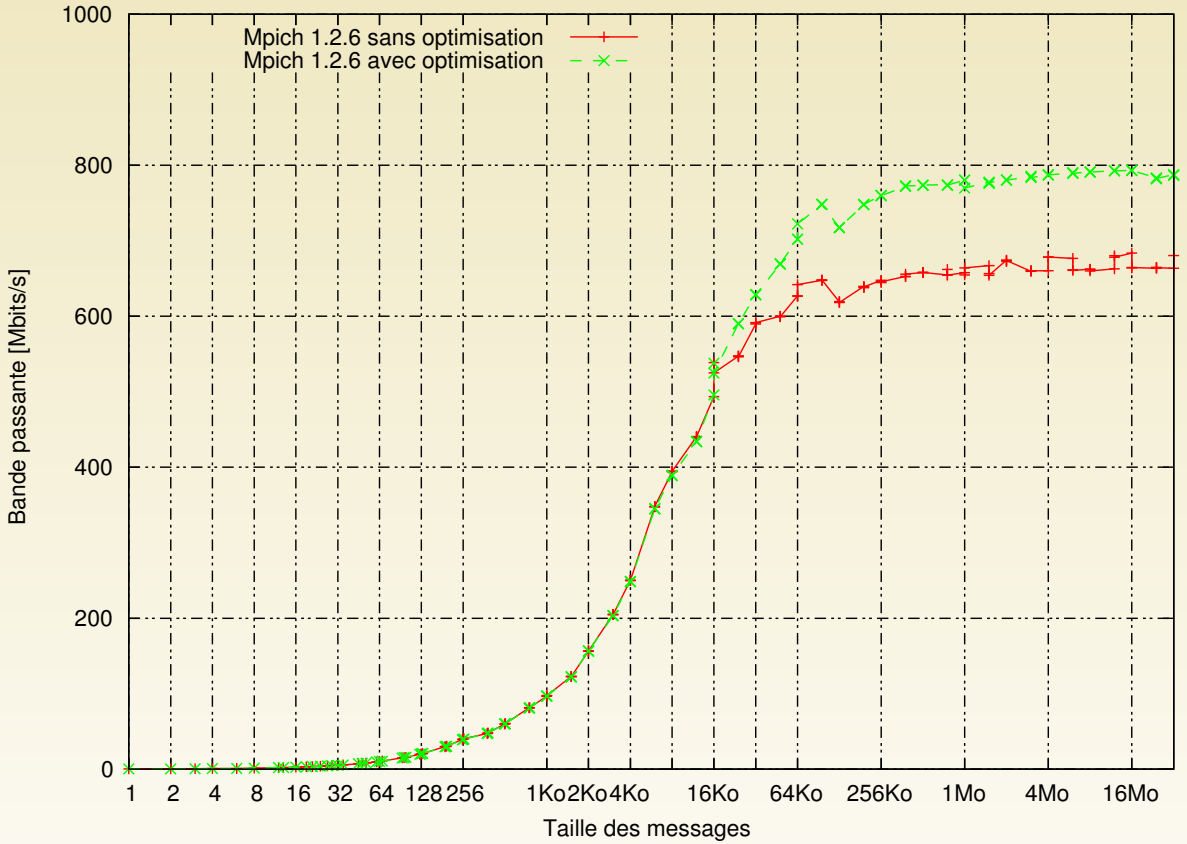
long messages $2o + L + (m-1)G$

There is no fundamental difference...

OK, it works for small and large messages. Does it work for average-size messages ? pLogP [KBV00] is an extension of LogP when L, o and g depends on the message size m. They also have introduced a distinction between o_s and o_r. This is more and more precise but concurency is still not taken into account.

Bandwidth as a Function of Message Size

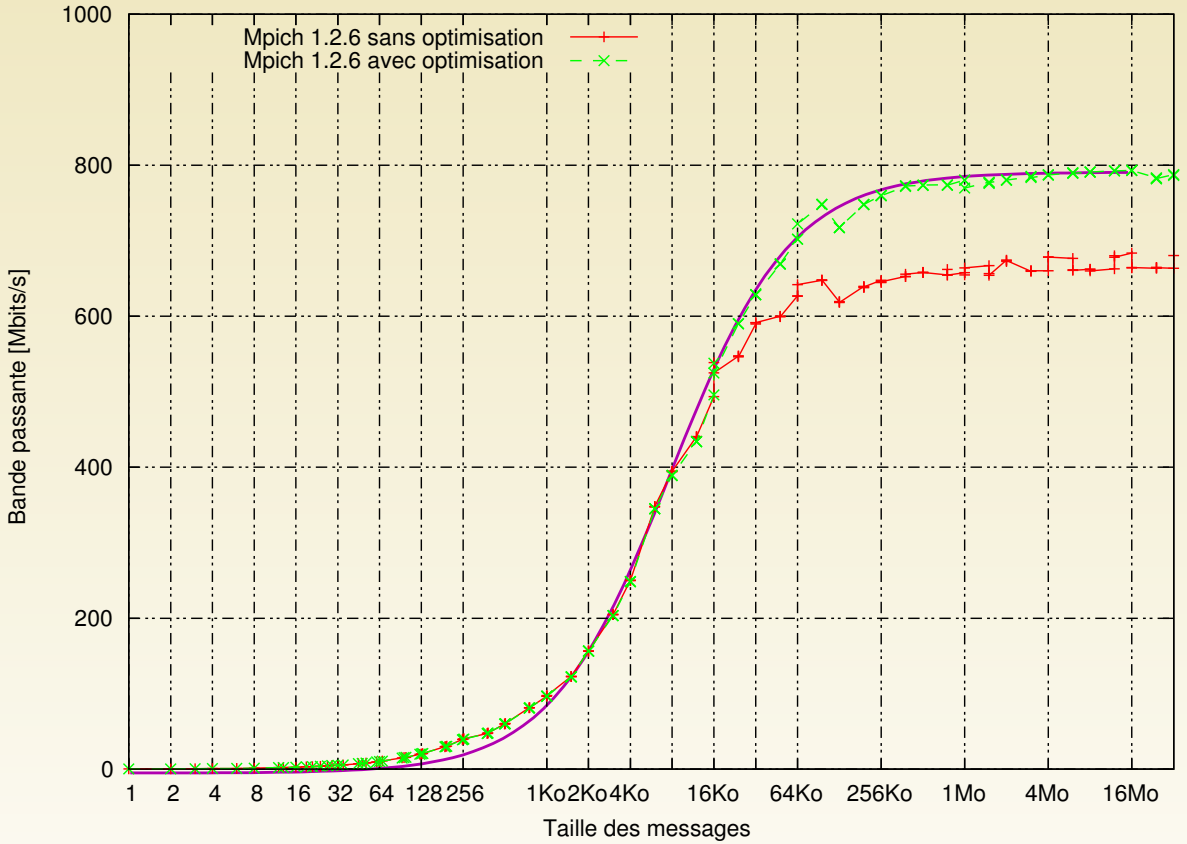
With the Hockney model: $\frac{m}{L+m/B}$.



MPICH, TCP with Gigabit Ethernet

Bandwidth as a Function of Message Size

With the Hockney model: $\frac{m}{L+m/B}$.



MPICH, TCP with Gigabit Ethernet

What About TCP-based Networks?

The previous models work fine for parallel machines. Most networks use TCP that has fancy flow-control mechanism and slow start. Is it valid to use affine model for such networks?
The answer seems to be yes but latency and bandwidth parameters have to be carefully measured [LQDB05].

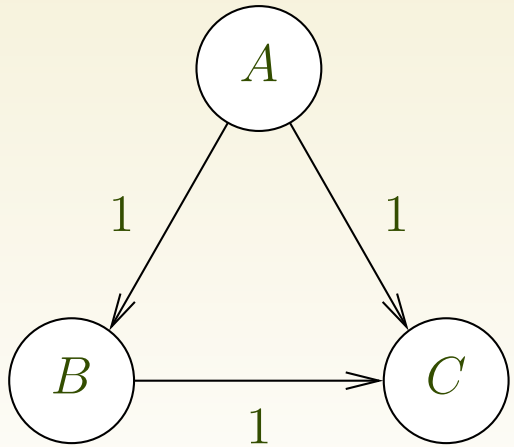
- Probing for $m = 1$b and $m = 1$Mb leads to bad results.
- The whole middleware layers should be benchmarked (theoretical latency is useless because of middleware, theoretical bandwidth is useless because of middleware and latency).

The slow-start does not seem to be too harmful.
Most people forget that the round-trip time has a huge impact on the bandwidth.

Outline

Multi-ports

- A given processor can communicate with as many other processors as he wishes without any degradation.
- This model is widely used by scheduling theoreticians (think about all DAG with commmunications scheduling problems) to prove that their problem is hard and to know whether there is some hope to prove some clever result or not.
  Some theoreticians feel like this model is borderline, especially when allowing duplication or when trying to design algorithms with super tight approximation ratios [Yves Robert 01–??].
  Frankly, such a model is totally unrealistic.
- Using MPI and synchronous communications, it may not be an issue. However, with multi-core, multi-processor machines, it cannot be ignored...

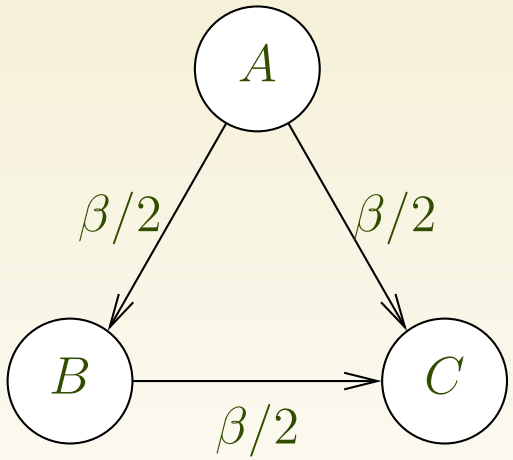


Multi-port

Bounded Multi-port

- Assume now that we have threads or multi-core processors.

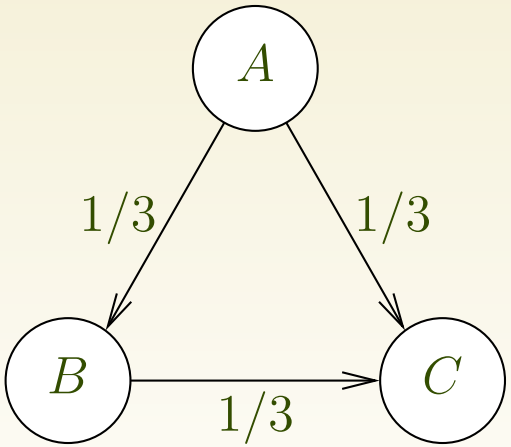
  We can write that sum of the throughputs of all communications (incomming and outgoing). Such a model is OK for wide-area communications [HP04].
- Remember, the bounds due to the round-trip-time must not be forgotten!



Multi-port (β)

Single-port (Pure)

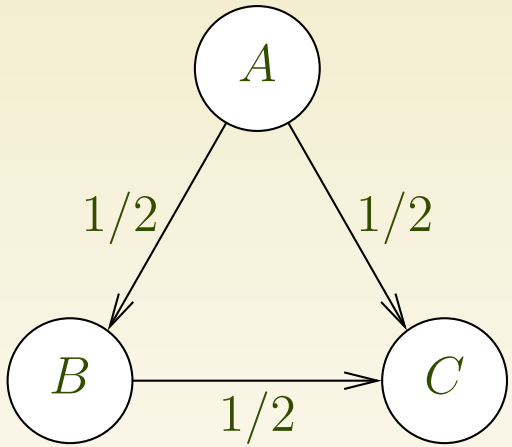
- A process can communicate with only one other process at a time. This constraint is generally written as a constraint on the sum of communication times and is thus rather easy to use in a scheduling context (even though it complexifies problems).
- This model makes sense when using non-threaded versions of communication libraries (e.g., MPI). As soon as you're allowed to use threads, bounded-multiport seems a more reasonnable option (both for performance and scheduling complexity).



1-port (pure)

Single-port (Full-Duplex)

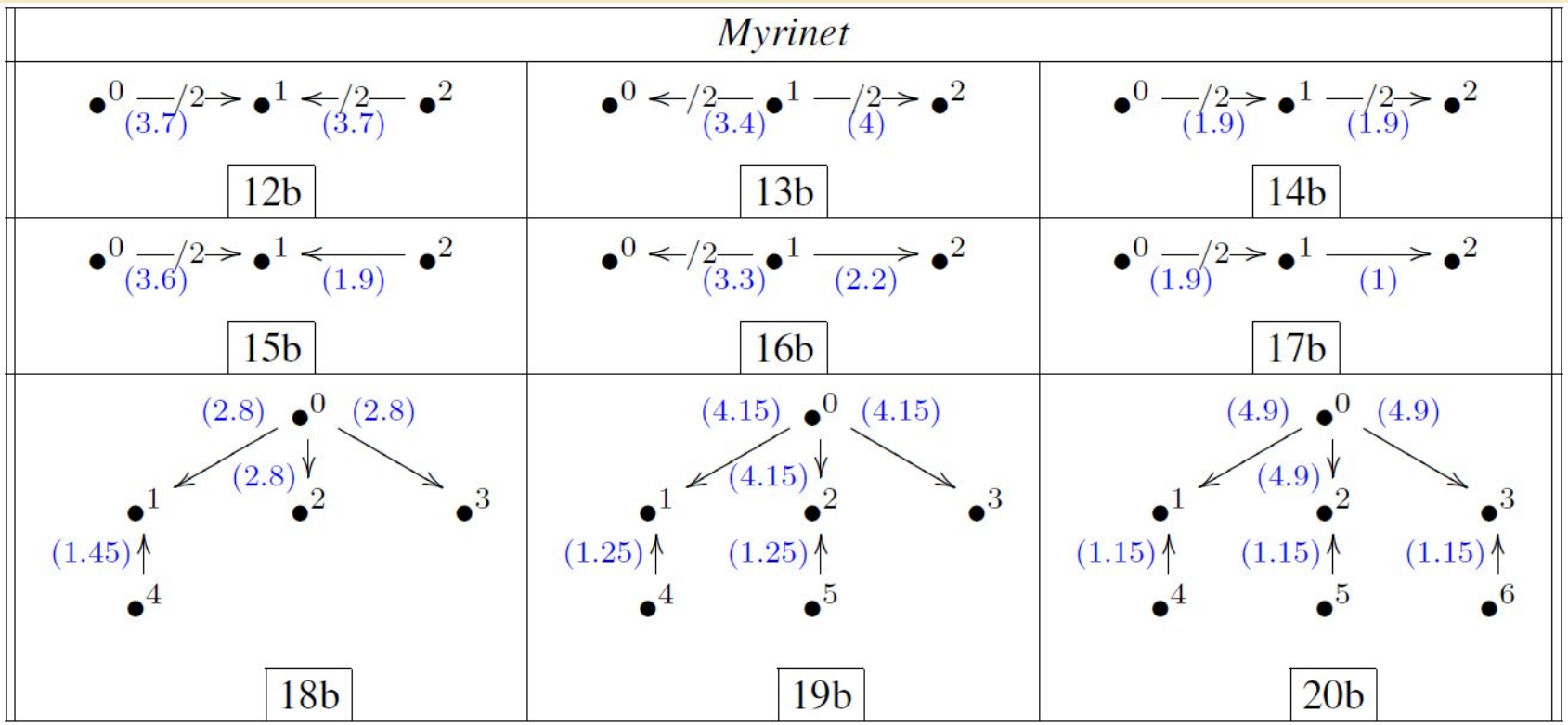
At a given time, a process can be engaged in at most one emission and one reception. This constraint is generally written as two constraints: one on the sum of incomming communication times and one on the sum of outgoing communication times.



1-port (full duplex)

Single-port (Full-Duplex)

This model somehow makes sense when using networks like Myrinet that have few multiplexing units and with protocols without control flow [Mar07].

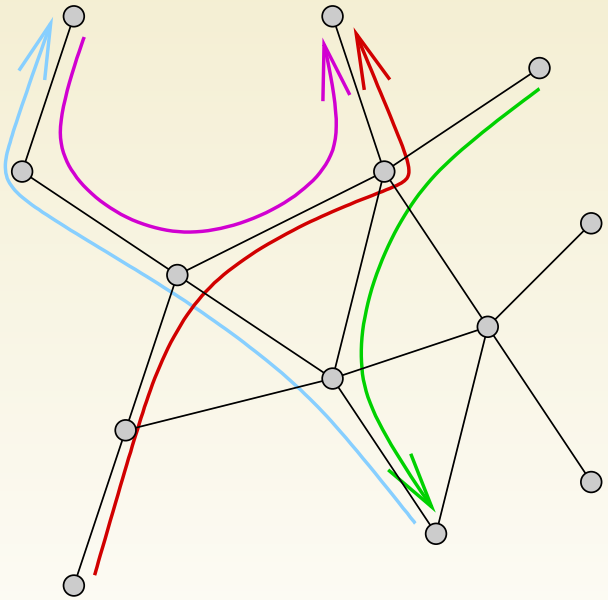


Even if it does not model well complex situations, such a model is not harmfull.

Fluid Modeling

When using TCP-based networks, it is generally reasonnable to use flows to model bandwidth sharing [MR99, Low03].

$$\forall l \in \mathcal{L}, \sum_{r \in \mathcal{R} \text{ s.t. } l \in r} \rho_r \leqslant c_l$$



Income Maximization maximize $\sum_{r \in \mathcal{R}} \rho_r$

Max-Min Fairness maximize $\min_{r \in \mathcal{R}} \rho_r$

Proportional Fairness maximize $\sum_{r \in \mathcal{R}} \log(\rho_r)$

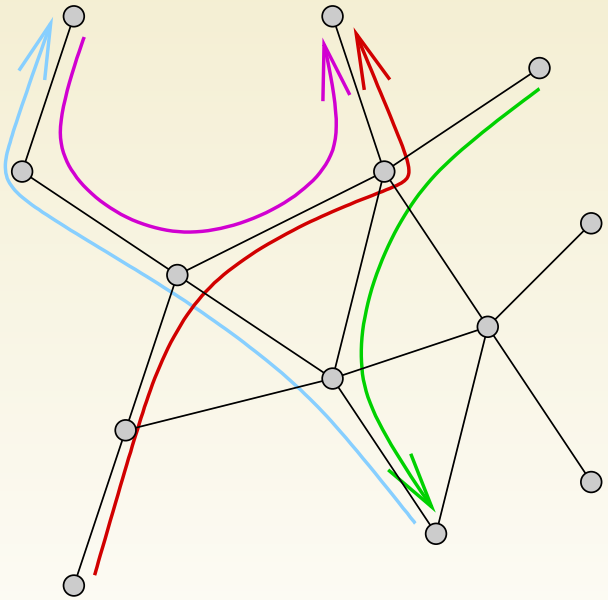
Potential Delay Minimization minimize $\sum_{r \in \mathcal{R}} \frac{1}{\rho_r}$

Some weird function minimize $\sum_{r \in \mathcal{R}} \arctan(\rho_r)$

Fluid Modeling

When using TCP-based networks, it is generally reasonnable to use flows to model bandwidth sharing [MR99, Low03].

$$\forall l \in \mathcal{L}, \sum_{r \in \mathcal{R} \text{ s.t. } l \in r} \rho_r \leqslant c_l$$



Income Maximization maximize $\sum_{r \in \mathcal{R}} \rho_r$

Max-Min Fairness maximize $\min_{r \in \mathcal{R}} \rho_r$ ATM

Proportional Fairness maximize $\sum_{r \in \mathcal{R}} \log(\rho_r)$

TCP Vegas

Potential Delay Minimization minimize $\sum_{r \in \mathcal{R}} \frac{1}{\rho_r}$

Some weird function minimize $\sum_{r \in \mathcal{R}} \arctan(\rho_r)$

TCP Reno

Flows Extensions

- Note that this model is a multi-port model with capacity-constraints (like in the previous bounded multi-port).
- When latencies are large, using multiple connections enables to get more bandwidth. As a matter of fact, there is very few to loose in using multiple connections...
- Therefore many people enforce a sometimes artificial (but less intrusive) bound on the maximum number of connections per link [Wag05, MYCR06].

Outline

Remind This is a Model, Hence Imperfect

- The previous sharing models are nice but you generally do not know other flows...
- Communications use the memory bus and hence interfere with computations. Taking such interferences into account may become more and more important with multi-core architectures.
- Interference between communications are sometimes... surprising.

Modeling is an art. You have to know your platform and your application to know what is negligeable and what is important. Even if your model is imperfect, you may still derive interesting results.


A. Alexandrov, M. Ionescu, K. Schauser, and C. Scheiman.
LogGP: Incorporating long messages into the LogP model for parallel computation.
Journal of Parallel and Distributed Computing, 44(1):71–79, 1997.


D. Culler, R. Karp, D. Patterson, A. Sahay, E. Santos, K. Schauser, R. Subramonian, and T. von Eicken.
LogP: a practical model of parallel computation.
Communication of the ACM, 39(11):78–85, 1996.


R. W. Hockney.
The communication challenge for mpp : Intel paragon and meiko cs-2.
Parallel Computing, 20:389–398, 1994.


B. Hong and V.K. Prasanna.
Distributed adaptive task allocation in heterogeneous computing environments to maximize throughput.

In *International Parallel and Distributed Processing Symposium IPDPS'2004*. IEEE Computer Society Press, 2004.

T. Kielmann, H. E. Bal, and K. Verstoep.
Fast measurement of LogP parameters for message passing platforms.
In *Proceedings of the 15th IPDPS. Workshops on Parallel and Distributed Processing*, 2000.

Steven H. Low.
A duality model of TCP and queue management algorithms.
IEEE/ACM Transactions on Networking, 2003.

Dong Lu, Yi Qiao, Peter A. Dinda, and Fabián E. Bustamante.

Characterizing and predicting tcp throughput on the wide area network.
In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005.

Maxime Martinasso.

Analyse et modélisation des communications concurrentes dans les réseaux haute performance.
PhD thesis, Université Joseph Fourier de Grenoble, 2007.


Laurent Massoulié and James Roberts.
Bandwidth sharing: Objectives and algorithms.
In *INFOCOM (3)*, pages 1395–1403, 1999.


Loris Marchal, Yang Yang, Henri Casanova, and Yves Robert.
Steady-state scheduling of multiple divisible load applications on wide-area distributed computing platforms.
Int. Journal of High Performance Computing Applications, (3), 2006.


Frédéric Wagner.
Redistribution de données à travers un réseau haut débit.
PhD thesis, Université Henri Poincaré Nancy 1, 2005.