Flow-Level Modeling of Parallel Download in Distributed Systems

Abdulhalim Dandoush [1] Alain Jean-Marie [2]

[1]INRIA Sophia-Antipolis-Méditerranée
[2]INRIA Sophia-Antipolis-Méditerranée and LIRMM CNRS/Univ. Montpellier 2

CTRQ 2010, Athens, 15 June 2010

Work funded by the French National Research Agency Grant VOODDO, Multimedia Program

Outline

Progress

The Problem and the question

Distributed storage system:

- storage locations (servers), replication of data
- clients doing parallel download.

Traffic arrives continuously, randomly at client nodes.
The Transport protocol is TCP.

Two related questions:

- What is the response time of file transfers?
- How is the bandwidth shared between flows?

Genesis of the present work

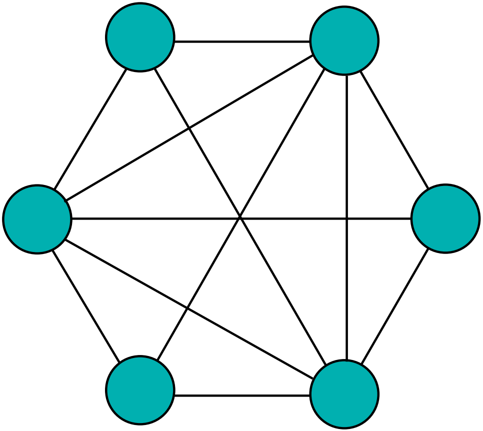
Two independent research actions meet:

- Grid Delivery Network: contents distribution infrastructure developed by the VodDnet company
  http://www.voddnet.com/
  $\Longrightarrow$ development of a flow-level simulator for dimensioning
- Optimization of data replication and redundancy schemes
  $\Longrightarrow$ development of a ns2-based simulator

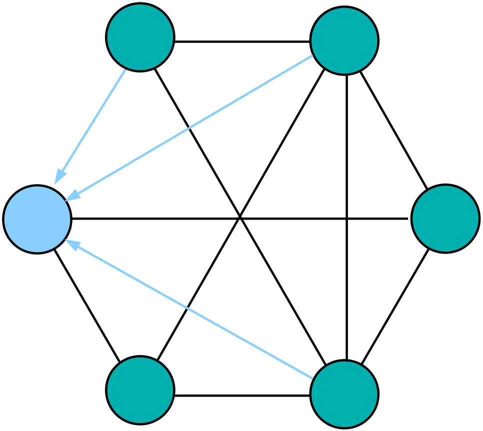
A. Dandoush, S. Alouf, and P. Nain, "A realistic simulation model for peer-to-peer storage systems," in *Proc. of 2nd International ICST Workshop on Network Simulation Tools (NSTOOLS09)*, Pisa, Italy, October 19 2009.

$\Longrightarrow$ opportunity for validating the results of simulations.

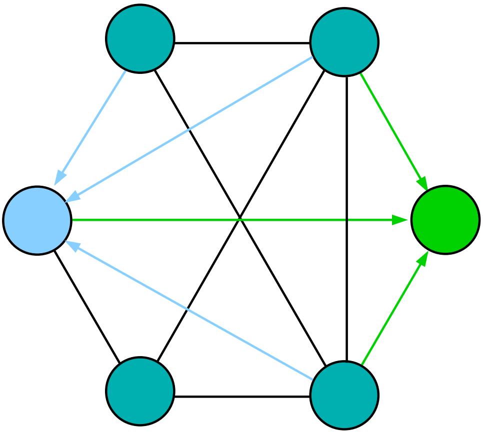
A not-so-simple problem



A not-so-simple problem



A not-so-simple problem



Related Literature

The question of bandwidth sharing in data networks (including the Internet) has, of course, been addressed before.
Several, partly contradictory findings:

- TCP may exhibit a chaotic behavior: e.g. Veres and Boda (invalidated by Figuereido *et al.*), Baccelli and Hong;
- but also may share a link quite fairly: Heyman *et al.*, Ben Fredj *et al.*

Fairness of bandwidth sharing

- Bertsekas and Gallager introduce max/min fairness in networking (Rawls' criterion) and the "progressive filling" algorithm
- Many concepts of fairness: see the survey of Le Boudec

Related Literature (ctd)

We retain that:

- The "fairness" acheived by TCP is not max/min for infinite-living flows sharing a single bottleneck
- Bonald and Proutière suggest that when the traffic is more dynamic, the differences tend to blur

Finally:

No consensus, no operational methods, no dynamic traffic, no batch arrivals

Purpose of this talk

In the talk, we:

- investigate whether the max/min way of sharing bandwidth gives "good enough" results
- introduce a flow-level simulation algorithm
- perform comparisons with packet-level simulations
- show that the results are good.
- discuss a queuing theoretic (Processor-Sharing) approximation

Progress

Flow-Level Modeling

Abdulhalim Dandoush , Alain Jean-Marie

Introduction
The problem
Why isn't it simple?
The literature
This talk

The Flow-Level Algorithm

Experiments
Experimental setup
Results

Analysis

Conclusion

Questions

The Progressive-Filling Flow-Level Algorithm

Notation: $f \nabla a$ for "flow f goes through link a"

Algorithm PFFLA

Data: Set of links $\mathcal{A}$ with capacities C_a; set of flows $\mathcal{F}$
Result: A throughput value for each flow
begin
  remove from $\mathcal{A}$ nodes without flows ;
  while $\mathcal{A}$ *not empty* **do**
    foreach $a \in \mathcal{A}$ **do** $N_a \leftarrow \#\{f \in \mathcal{F} | f \nabla a\}$;
    calculate $\theta^* = \min_{a \in \mathcal{A}} C_a / N_a$; $a^* = arg \min_{a \in \mathcal{A}} C_a / N_a$
    foreach $f, f \nabla a^*$ **do**
      set $\theta_f = \theta^*$;
      foreach $a \in \mathcal{A}, f \nabla a$ **do** $C_a \leftarrow C_a - \theta^*$;
      remove f from $\mathcal{F}$;
    remove from $\mathcal{A}$ links without flows ;
  return $\{\theta_f\}$

Properties

Definition of max/min fairness (Bertsekas & Gallager, Rawls)

A throughput allocation is max/min fair if an increase in some flow's share must result in the decrease of another flow's share that had already less throughput.

Theorem

The algorithm computes a max/min fair sharing of the bandwidth

Proof: (indirect) the algorithm does basically the same operations as the "progressive filling" algorithm of Bertsekas & Gallager.
Proof: (direct) there is a necessary and sufficient condition for an allocation to be max/min fair, see Bertsekas & Gallager. It is satisfied by the algorithm.

Progress

Simulation Experiments

Experiments:

- Embed the PFFLA into a dynamic network simulator
- Simulate the same setting with ns-2
- Compare the distributions, averages
- Compare with a Processor-Sharing Queueing Model

Flow-Level Simulator

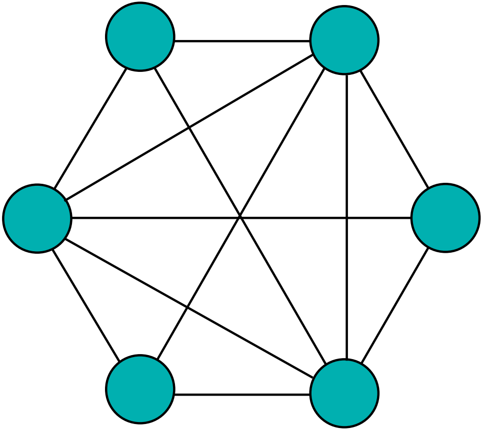
An event-driven simulator at the flow level.

Flow-Level File Transfer Simulator

```
begin
    arr ← nextArrival(); dep ← +∞ ;
    repeat
        if arr ≤ dep then  // this is an arrival
            create s flows ;
            arr ← nextArrival()
        else  // this is a flow completion
            terminate the flow ;
            perform statistics
        apply PFFLA ;
        dep ← nextCompletion()
    until terminal condition;
```

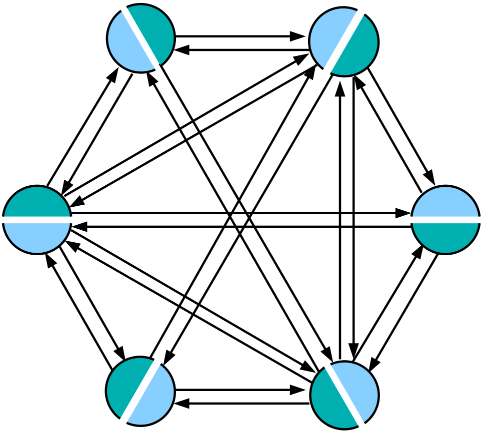
Topological setup

All-to-all symmetric communications

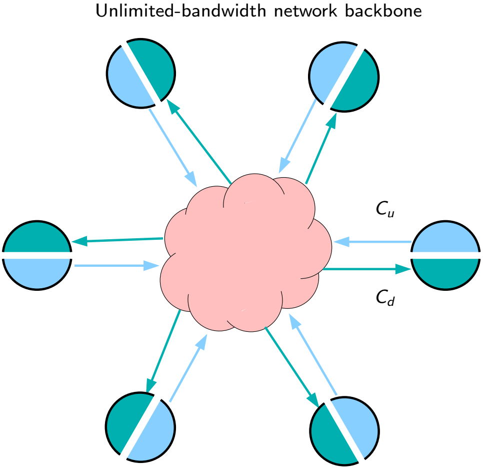


Topological setup

Independence of upload/download bandwidth



Topological setup



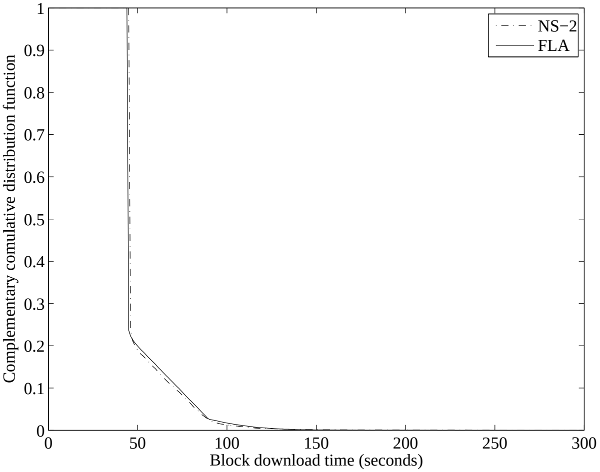
Other experimental data

- Arrival of download requests: Poisson process rate λ
- 4 blocks per file
- Block size: 1 MB or 2 MB
- $\mathcal{N}$ nodes total: $\mathcal{N}/2$ upload, $\mathcal{N}/2$ download.
- Upload/Download link capacity C_u/C_d: symmetric or asymmetric.
  When asymmetric, ratio $C_d \div C_u \simeq 4$
- Ns-2 specific parameters: standard TCP parameters, equal link latencies (2ms), large buffer size (500 packets), 1500 B per packet.

Parameters of the experiments

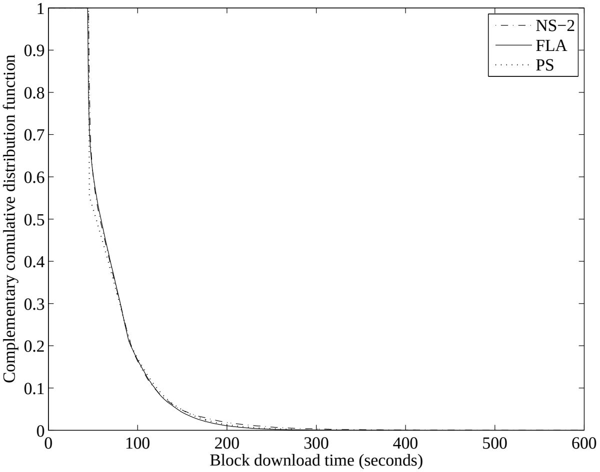
Experiment number	$\mathcal{N}/2$ peers	C_d/C_u kbps	S_B/S_F MB	$1/\lambda$ sec.	ρ %
1	25	384/384	4/1	60	6
2	250	576/576	8/2	1.913	25
3	250	1500/1500	8/2	0.510	36
4	250	1500/1500	8/2	0.367	50
5	250	1500/1500	8/2	0.306	60
6	250	1500/1500	8/2	0.262	70
7	25	1500/384	8/2	59.81	12
8	250	1500/384	8/2	5.98	12
9	500	1500/384	8/2	2.99	12
10	500	1500/384	8/2	0.718	50
11	500	2000/384	8/2	0.718	50

Results / Small Load



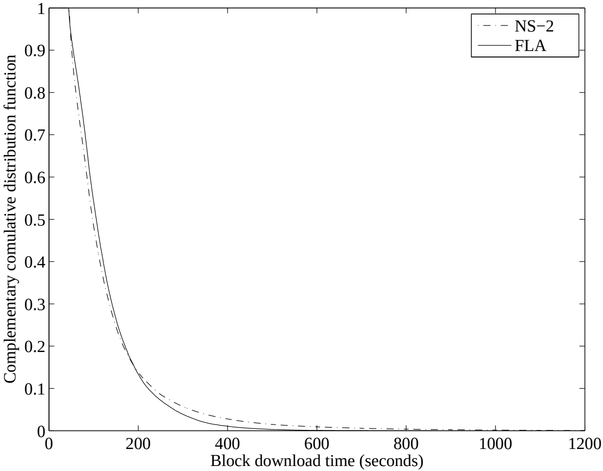
$\rho = 12\%, \mathcal{N}{=}500,\ C_u = C_d = 1500 kb/s$

Results / Intermediate Load



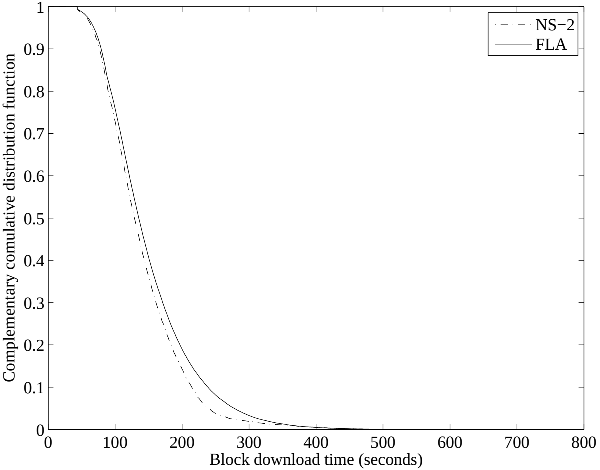
$\rho = 36\%,\ \mathcal{N}{=}500,\ C_u = C_d = 1500 kb/s$

Results / Medium to Large Load



$\rho = 60\%, \mathcal{N}{=}500, \ C_u = C_d = 1500 kb/s$

Results / Medium to Large Load (ctd)



$\rho = 50\%$, $\mathcal{N}$=1000, $C_u = 384kb/s$, $C_d = 1500kb/s$

Comparison of average download times: PFFLA, NS and PS

Ex. nb	$\hat{E}[T_{NS}]$ sec.	$E[T_{FLA}]$ sec.	$RR\%$ NS/FLA	$E[T_{PS}]$ sec.	$RR\%$ NS/PS
1	96.062	95.45	0.6%	95.44	0.6%
2	161.252	160.196	0.6%	166.132	-3%
3	73.547	73.346	0.2%	71.7692	2.4%
4	99.501	97.75	1.7%	91.864	7.6%
5	129.066	127.691	1%	114.83	11%
6	176.45	180.05	-2%	153.107	13.2%
7	61.137	62.901	-2.8%	52.19	17%
8	64.738	64.935	-0.3%	52.19	19.3%
9	65.298	65.182	-0.2%	52.19	20%
10	144.615	152.137	-5.8%	91.865	36%
11	142.1	149.213	-5.1%	68.45	51.8%

Progress

A Processor-Sharing approximation

In some situations, bottlenecks occur at the client side. This is the case:

- when the upload bandwidth is large enough
- when load is small enough

Formula for the response time in the $M/D/1/PS$ queue

$$E[T_{PS}] \;=\; \frac{d}{1-\rho}\;,$$

where:

- d the unitary download time
- $\rho = \lambda\sigma$ the load factor of the link.

Distribution of the response time in the PS queue

According to Yashkov and Yashkova (2007) the distribution of $V(d)$, the response time in a $M/D/1/PS$ queue with:

- arrival rate λ,
- service time d,
- load factor is $\rho = \lambda d$, is:

$$E(e^{-sV(d)}) \;=\; (1-\rho)\,\frac{(s+\lambda)^2 e^{-d(s+\lambda)}}{s^2+\lambda(s+(s+\lambda)(1-\rho))e^{-d(s+\lambda)}}\;,$$

Distribution of the response time in the PS queue (ctd.)

This gives:

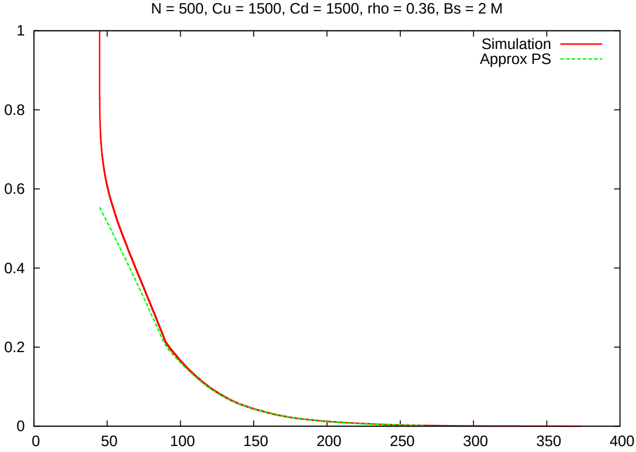
$$P(V(d) \leq d + t) \;=\; (1-\rho)e^{-\rho} \sum_{n=0}^{\infty} (-1)^n e^{-n\rho} 1_{\{t \geq nd\}}$$

$$\sum_{m=0}^{n} \binom{n}{m} (2-\rho)^m (1-\rho)^{n-m} \frac{[\lambda(t-nd)]^{2n-m}}{(2n-m)!}$$

$$\left[1 + 2\lambda \frac{t-nd}{2n-m+1} + \frac{\lambda^2(t-nd)^2}{(2n-m+1)(2n-m+2)}\right]$$

Approximation with PS, medium load

Approximation is very good for loads up to 25%.
Even for 36%:



Progress

Conclusions

Contributions:

- The Flow-Level modeling with max/min fairness works fine:
  - at least for the mean response time
  - up to a load of 50-60%
  - and much faster than packet-level modeling
- Queueing formulas work also up to a load of 40%

More work needed:

- Optimize the algorithm for speed
- Understand the deviation in distributions
- Address the problem of different RTTs
- Find queueing formulas for asymmetric up/down links

Questions?

Bibliography

The complete paper

A. Dandoush and A. Jean-Marie, "Download Process in Distributed Systems, Flow-level Algorithm vs. Packet-level Simulation Model," INRIA, Research Report RR-7159, 2009, last accessed: 24 Mar 2010. [Online]. Available: `http://hal.inria.fr/inria-00442030/en/`.

References on TCP

Veres, A. and Boda, M., "The chaotic nature of TCP congestion control". *Proc. IEEE INFOCOM 2000*.

Figueiredo, D.R., Liu, B., Feldmann, A., Misra, V. and Towsley, D., "On TCP and self-similar traffic", *Performance Evaluation*, **61** (2-3), pp. 129–141, July 2005.

Bibliography (ctd)

Flow sharing and fairness

D. Bertsekas and R. Gallager, *Data Networks, 2nd ed.* Prentice Hall, New Jersey, 1992.

D. P. Heyman, T. V. Lakshman, and A. L. Neidhardt, "A new method for analysing feedback-based protocols with applications to engineering web traffic over the Internet," in *Proc. of 1997 ACM SIGMETRICS Intl. Conf. on Measurement and modeling of comp. systs.*, New York, USA, 1997, pp. 24–38.

S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié, and J. W. Roberts, "Statistical bandwidth sharing: a study of congestion at flow level," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 111–122, 2001.

Bibliography (end)

T. Bonald and A. Proutière, "Insensitive bandwidth sharing in data networks," *Queueing Systems*, vol. 44, pp. 69–100, 2003.

J.-Y. Le Boudec, *Rate adaptation, Congestion Control and Fairness: A Tutorial*, Ecole Polytechnique Fédérale de Lausanne (EPFL), Dec 2008.

The Processor Sharing Queue

Yasshkov, S. F. and Yashkova, A. S., "Processor sharing: A survey of the mathematical theory", *Automation and Remote Control*, **68** (9), pp. 1662–1731, 2007.