From BitTorrent to Future Networks

GANG achievements and perspective

INRIA Evaluation
GANG Project-Team

http://gang.inria.fr/

March 22, 2012

PLAN

PLAN

Content Distribution

Content distribution is mainly:

- Filesharing
- Streaming
  - OnDemand (YouTube, Netflix)
  - Live (Sport events)

Content Distribution

Content distribution is mainly:

- **Filesharing**
- Streaming
  - OnDemand (YouTube, Netflix)
  - Live (Sport events)

Content Distribution

Content distribution is mainly:

- Filesharing
- **Streaming**
  - OnDemand (YouTube, Netflix)
  - Live (Sport events)

Content Distribution

Content distribution is mainly:

- Filesharing
- Streaming
  - **OnDemand** (YouTube, Netflix)
  - Live (Sport events)

Content Distribution

Content distribution is mainly:

- Filesharing
- Streaming
  - OnDemand (YouTube, Netflix)
  - **Live** (Sport events)

Content Distribution

Content distribution is mainly:

- Filesharing
- Streaming
  - OnDemand (YouTube, Netflix)
  - Live (Sport events)

Things in common:

- Bandwidth is a key parameter
- Lot of stress for the network
- P2P solutions exist

Content Distribution

Here we focus on:

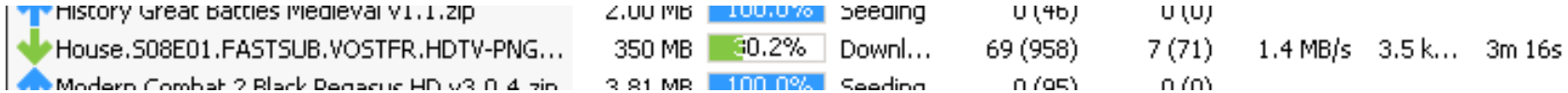
- **Filesharing**
- Streaming
  - OnDemand (YouTube, Netflix)
  - Live (Sport events)

BitTorrent

- Most popular P2P filesharing protocol of last decade
- Simple (relatively), yet effective
- Content is the core of the protocol
  - chunk: atomic* unit of data
  - `.torrent`: atomic* unit of content ($\Rightarrow$ Swarm)

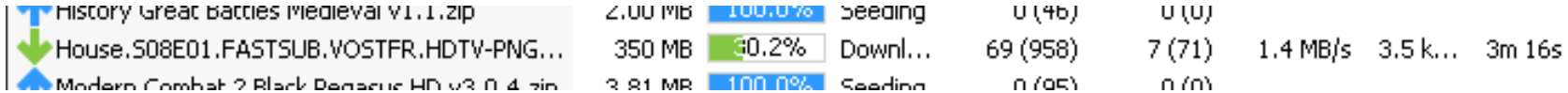
BitTorrent

- Most popular P2P filesharing protocol of last decade
- Simple (relatively), yet effective
- Content is the core of the protocol
  - chunk: atomic* unit of data
  - `.torrent`: atomic* unit of content ($\Rightarrow$ Swarm)
- BT is rather effective



BitTorrent

- Most popular P2P filesharing protocol of last decade
- Simple (relatively), yet effective
- Content is the core of the protocol
  - chunk: atomic* unit of data
  - `.torrent`: atomic* unit of content ($\Rightarrow$ Swarm)
- BT is rather effective



- Why?

PLAN

Tit-for-Tat

- Upload preferentially from your best downloaders
- Main reason proposed for BT success

Tit-for-Tat

- Upload preferentially from your best downloaders
- Main reason proposed for BT success
- TfT can be seen as a collaboration where each partner try to maximize their best interest

Tit-for-Tat

- Upload preferentially from your best downloaders
- Main reason proposed for BT success
- TfT can be seen as a collaboration where each partner try to maximize their best interest
- Reminds of stable marriage theory

Modeling with acyclic preference-based systems

- Stable matchings (1962-today) for P2P?
  - Acceptable partners $\leftrightarrow$ overlay
  - Preferences $\leftrightarrow$ "utility"
  - Quotas $\leftrightarrow$ limited number of connections
- Challenges: adapt for P2P
  - Dynamics matter
  - Specific preferences distribution:
    - Storage, bandwidth, CPU... $\rightarrow$ Total order
    - Proximity (RTT, co-uptime)... $\rightarrow$ Weights on edges
    - Differences, complementarity $\rightarrow$ Linear combination
    - Common thing: *acyclicity* property

Acyclic preference-based system: dynamic *matching* system with acyclic preferences

Modeling with acyclic preference-based systems

- Stable matchings (1962-today) for P2P?
  - Acceptable partners $\leftrightarrow$ overlay
  - Preferences $\leftrightarrow$ "utility"
  - Quotas $\leftrightarrow$ limited number of connections
- Challenges: adapt for P2P
  - Dynamics matter
  - Specific preferences distribution:
    - Storage, **bandwidth**, CPU... $\rightarrow$ Total order
    - Proximity (RTT, co-uptime)... $\rightarrow$ Weights on edges
    - Differences, complementarity $\rightarrow$ Linear combination
    - Common thing: *acyclicity* property

Acyclic preference-based system: dynamic *matching* system with acyclic preferences

Interest

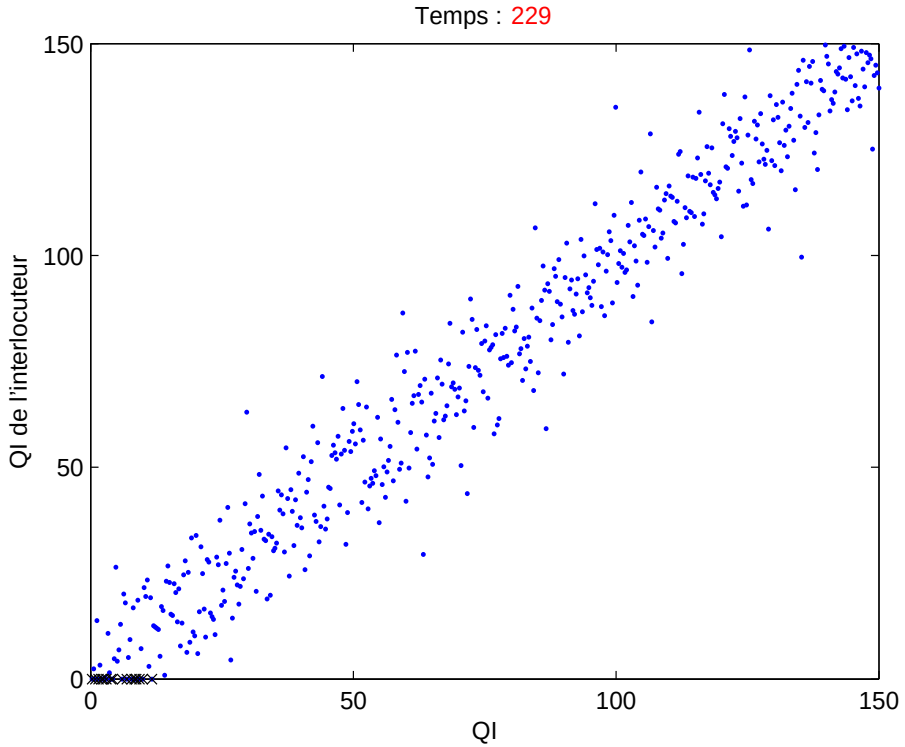
Theorem (INOC, 2007)

An acyclic preference-based system admits one, and only one, stable configuration, which is self-stabilizing.

Interest

Theorem (INOC, 2007)

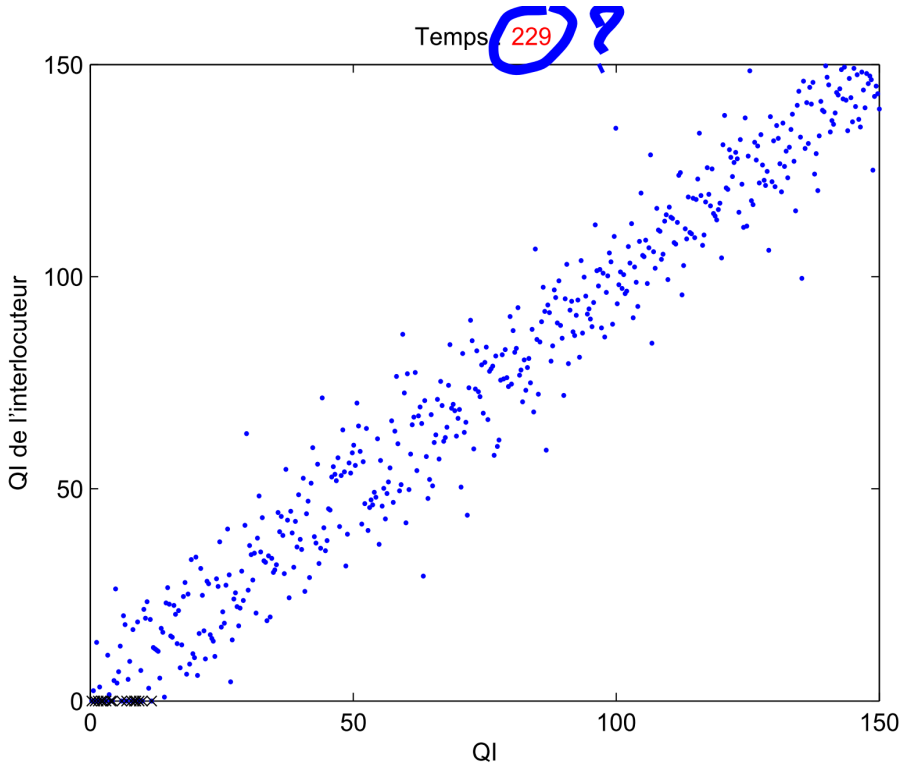
An acyclic preference-based system admits one, and only one, stable configuration, which is self-stabilizing.



Interest

Theorem (INOC, 2007)

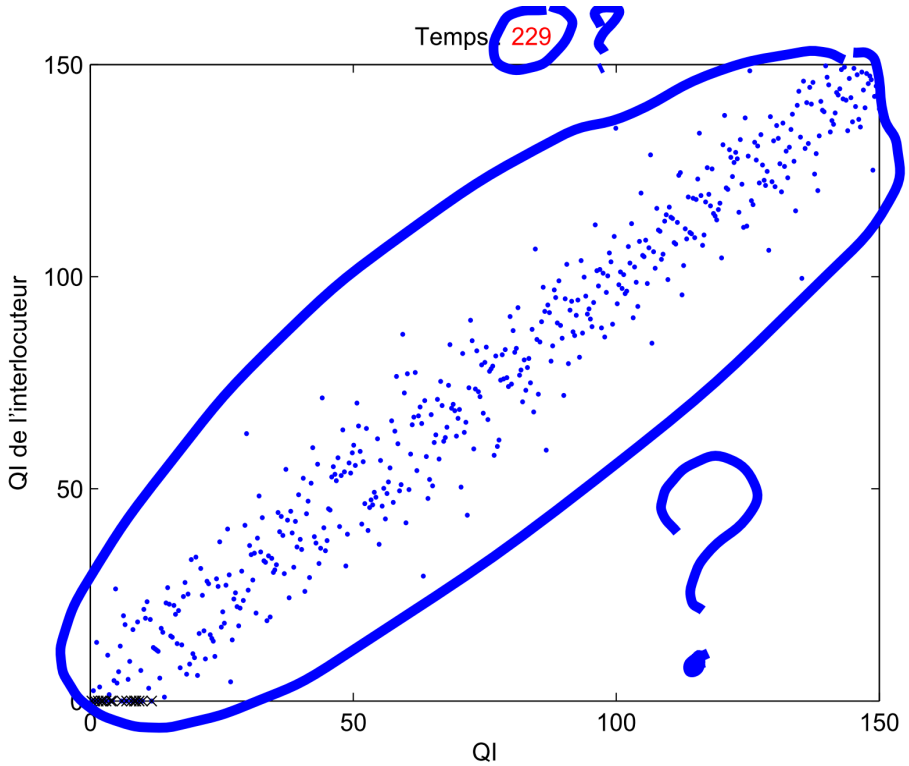
An acyclic preference-based system admits one, and only one, stable configuration, which is self-stabilizing.



Interest

Theorem (INOC, 2007)

An acyclic preference-based system admits one, and only one, stable configuration, which is self-stabilizing.



Convergence

- Worst case (adversarial asynchronous scheduling): exponential (SSS 2007)
- In practice (pseudo-synchronous, e.g. Round-Robin or Poisson): linear (/degree) for total order, logarithmic w.h.p. otherwise (P2P 2007 –best paper–, PPNA 2008).

Stable configuration (BT total order): recipe

- Start with a necessary and sufficient condition

Two peers are partners if they know each other and none of them has a better worst effective partner

Stable configuration (BT total order): recipe

- Start with a necessary and sufficient condition
- Use a simple overlay model

Random graph (Erdös-Rényi) $\mathcal{G}(n, p)$

Stable configuration (BT total order): recipe

- Start with a necessary and sufficient condition
- Use a simple overlay model
- Make some mean-field-like hypothesis

Independence of the events from the NSC

Stable configuration (BT total order): recipe

- Start with a necessary and sufficient condition
- Use a simple overlay model
- Make some mean-field-like hypothesis
- First result:

$$D(i,j) = pS(i,j)S(j,i), \text{ with}$$
$$S(i,j) = 1 - \sum_{k=1}^{j-1} D(i,k)$$

Stable configuration (BT total order): recipe

- Take some fluid limit

$$D(i,j),\ p \rightarrow \mathcal{D}(\alpha,\beta),\ d$$

$$\mathcal{D}(\alpha,\beta) = -d\mathcal{S}(\alpha,\beta)\mathcal{S}(\beta,\alpha), \text{ with}$$
$$\mathcal{S}(\alpha,\beta) = 1 - \int_0^\beta \mathcal{D}(\alpha,x)dx$$

Stable configuration (BT total order): recipe

- Take some fluid limit
- Solve the continuous equation

$$\mathcal{D}(\alpha,\beta) = \frac{d e^{d|\beta-\alpha|}}{(1 - e^{-d\min(\alpha,\beta)} + e^{d|\beta-\alpha|})^2}$$

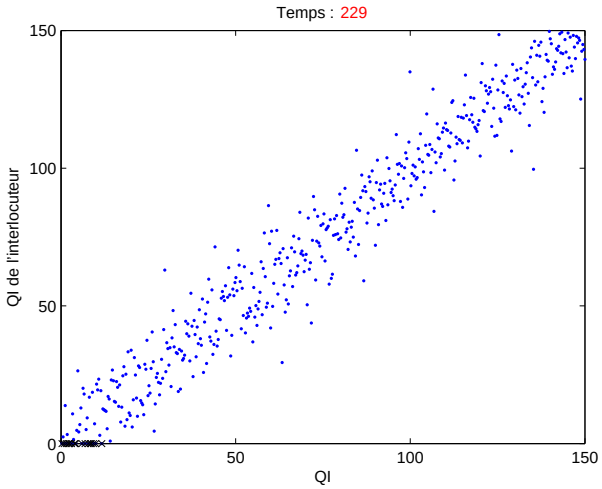
Stable configuration (BT total order): recipe

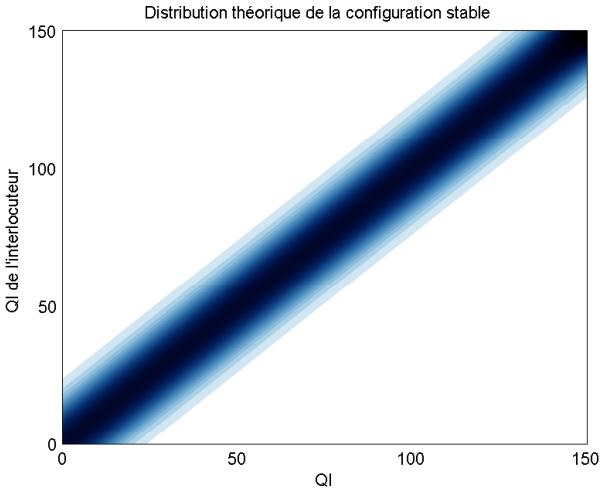
- Take some fluid limit
- Solve the continuous equation
- Back to discrete

$$D(i,j) \approx \frac{pe^{p|j-i|}}{(1-e^{-p\min(i,j)}+e^{p|j-i|})^2}$$

Stable configuration (BT total order): recipe

- Take some fluid limit
- Solve the continuous equation
- Back to discrete
- Validate





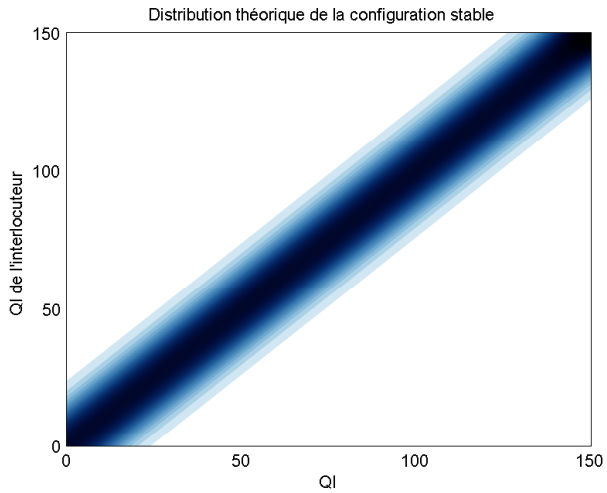
Stratification effects

- Remind: Tit-for-Tat (BitTorrent) $\rightarrow$ upload bandwidth

TfT: give to those who give you the most

Stratification effects

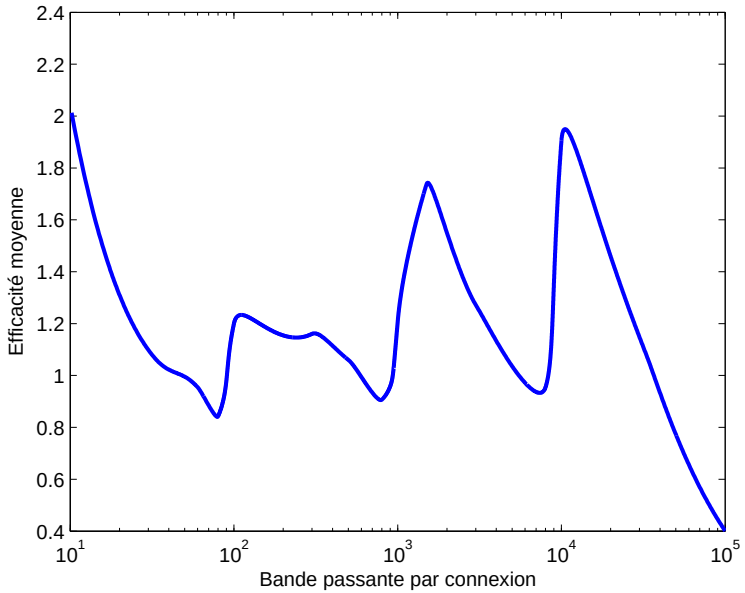
- Remind: Tit-for-Tat (BitTorrent) $\rightarrow$ upload bandwidth
- Stable configuration explains the observed stratification phenomenon



Stratification: interact with peers of a similar level

Stratification effects

- Remind: Tit-for-Tat (BitTorrent) $\rightarrow$ upload bandwidth
- Stable configuration explains the observed stratification phenomenon
- The model allows to study second order effects



Stratification effects

- Remind: Tit-for-Tat (BitTorrent) $\rightarrow$ upload bandwidth
- Stable configuration explains the observed stratification phenomenon
- The model allows to study second order effects
- Method can be applied to many other acyclic preferences

  Distances (ping), similarities, differences,. . .

Stratification effects

- Remind: Tit-for-Tat (BitTorrent) $\rightarrow$ upload bandwidth
- Stable configuration explains the observed stratification phenomenon
- The model allows to study second order effects
- Method can be applied to many other acyclic preferences

References : Europar 2007, ICDCS 2007, Infocom 2009

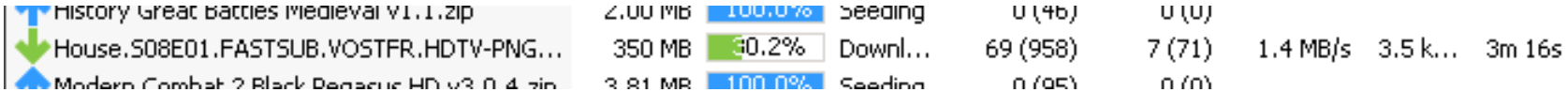
PLAN

Tit-for-Tat is not enough

- It gives an incentive for people to share their bandwidth (give and you shall receive)

Tit-for-Tat is not enough

- It gives an incentive for people to share their bandwidth (give and you shall receive)
- Not sufficient to explain actual performance

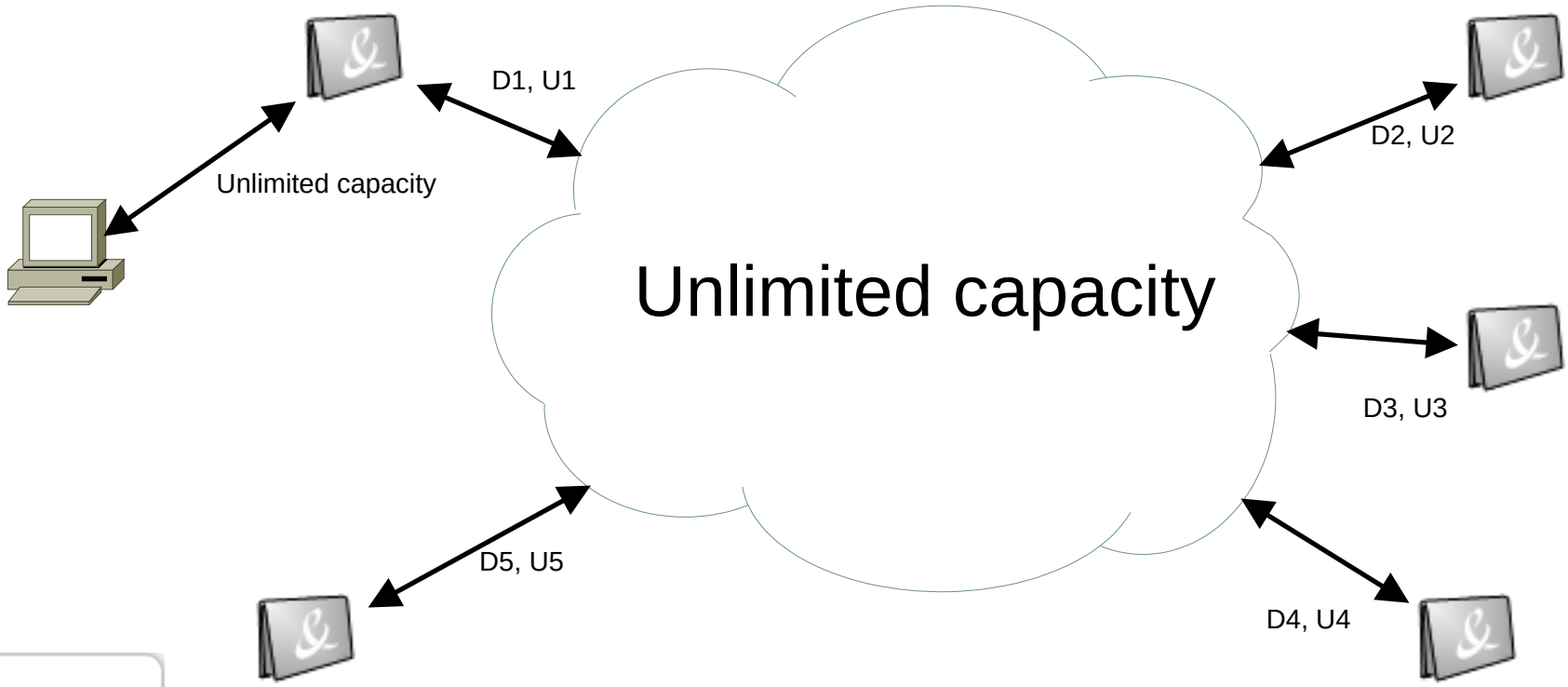


Bandwidth bottleneck

Basis of P2P model:

- Access is limited (physical/software)
- Everything else is not

$\Rightarrow$ Set of $\{d_i, u_i\}$ bottlenecks.



CLS model

Three main types of nodes

- Servers (C): provide, don't scale up
- Leechers (L): Need, provide
- Seeders (S): provide, scale

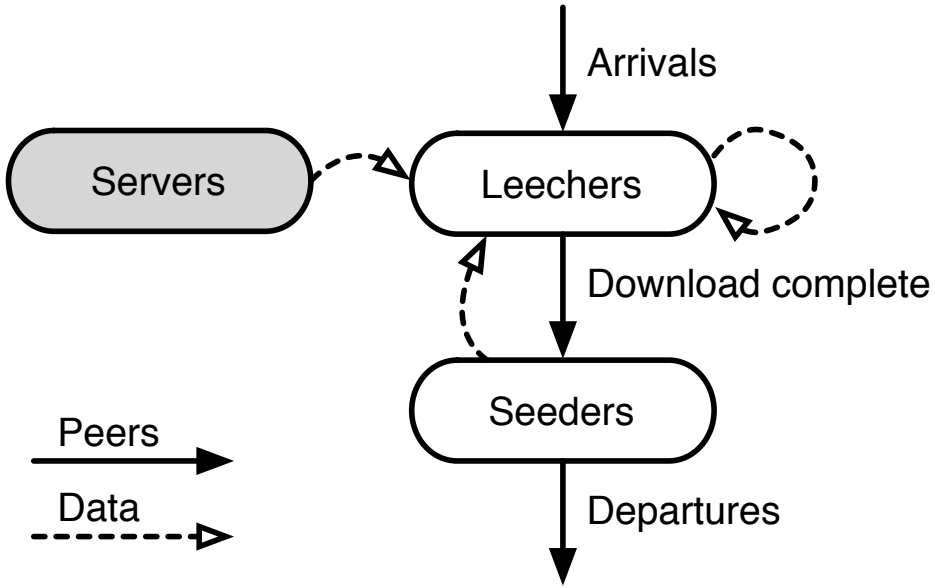
CLS model

Three main types of nodes

- Servers (C): provide, don't scale up
- Leechers (L): Need, provide
- Seeders (S): provide, scale

Often seen as an open network

CLS model



Often seen as an open network

Seeders: a reason for BT performance

BCL in steady state tells a lot

- **Scalability** (elastic rate)

$$d = (1 + \beta)u$$

Seeders: a reason for BT performance

BCL in steady state tells a lot

- Scalability (elastic rate)
- **Leverage** (fixed rate)

$$N_L \leq \frac{N_C}{1 - (\alpha_L + \beta\alpha_S)}$$

Seeders: a reason for BT performance

BCL in steady state tells a lot

- Scalability (elastic rate)
- Leverage (fixed rate)
- **Overprovisioning** (seeders)

$$\bar{T}_S \bar{u}_S + \frac{U_C}{\lambda} > C(1 - \frac{\bar{u}}{d_{\max}})$$

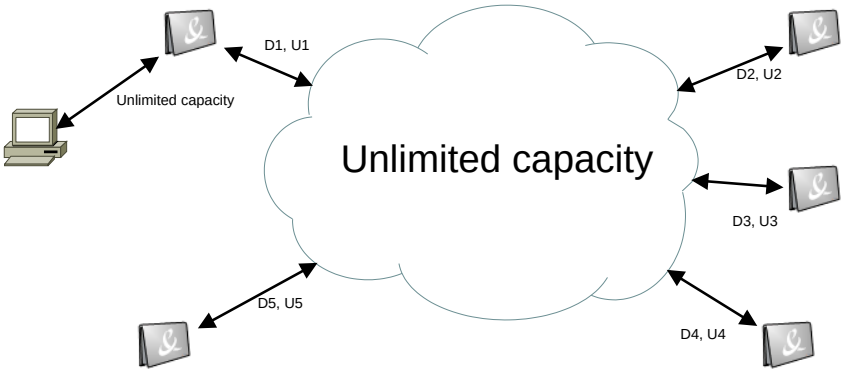
Bandwidth modeling: are we done?

Modeling P2P systems: big picture

Basis: bottleneck 1.0 model

- Access is limited (physical/software)
- Everything else is not

$\Rightarrow$ Set of $\{d_i, u_i\}$ bottlenecks.



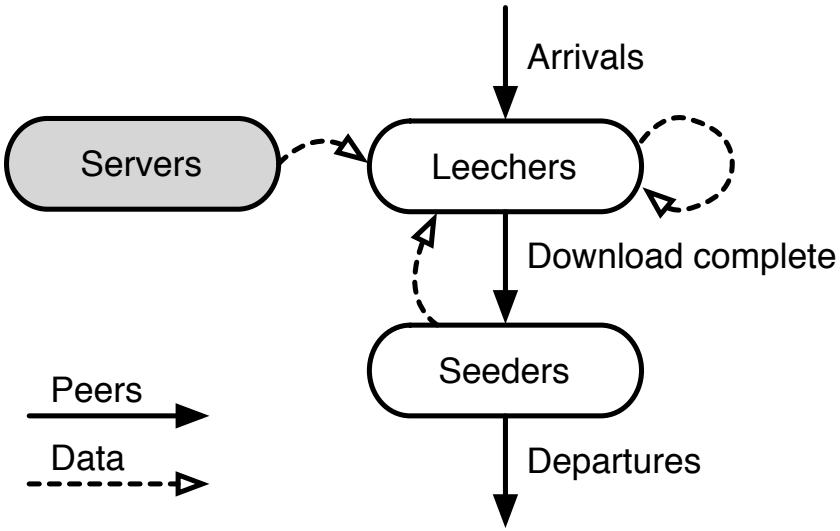
Access Bottleneck

Comprehension of P2P systems

Modeling P2P systems: big picture

Three main types of nodes

- Servers (C): provide, don't scale up
- Leechers (L): Need, provide
- Seeders (S): provide, scale



Access Bottleneck

C/L/S Model

Comprehension of P2P systems

Modeling P2P systems: big picture

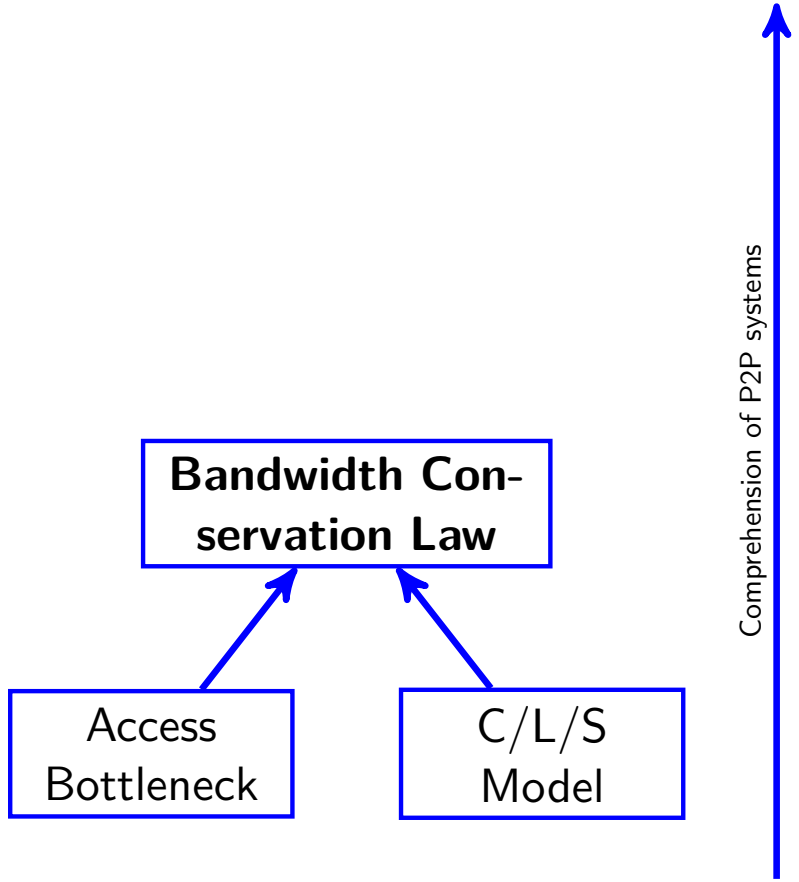
Bandwidth Conservation Law

- Scalability (elastic rate)
- Leverage (fixed rate)
- Overprovisioning (seeders)

$$d = (1 + \beta)u$$

$$N_L \leq \frac{N_C}{1 - (\alpha_L + \beta\alpha_S)}$$

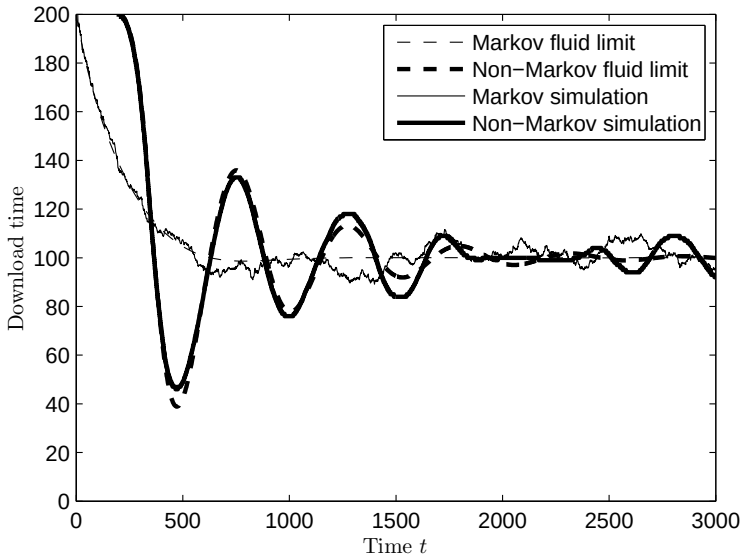
$$\bar{T}_S \bar{u}_S + \frac{U_C}{\lambda} > C(1 - \frac{\bar{u}}{d_{\max}})$$

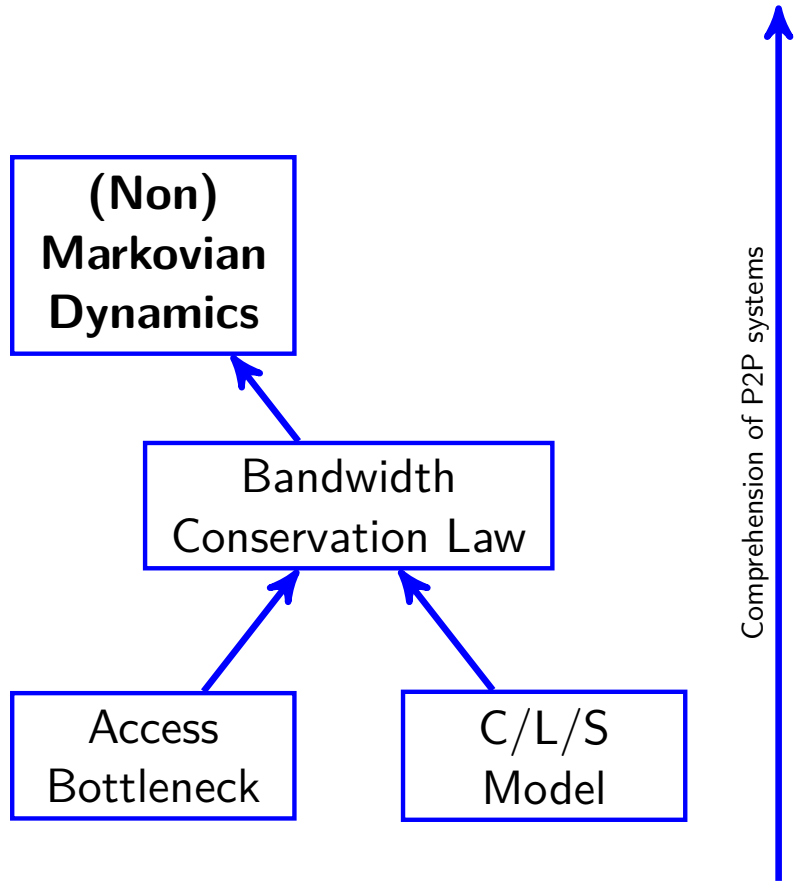


Modeling P2P systems: big picture

We build from there

- Dynamics of the system

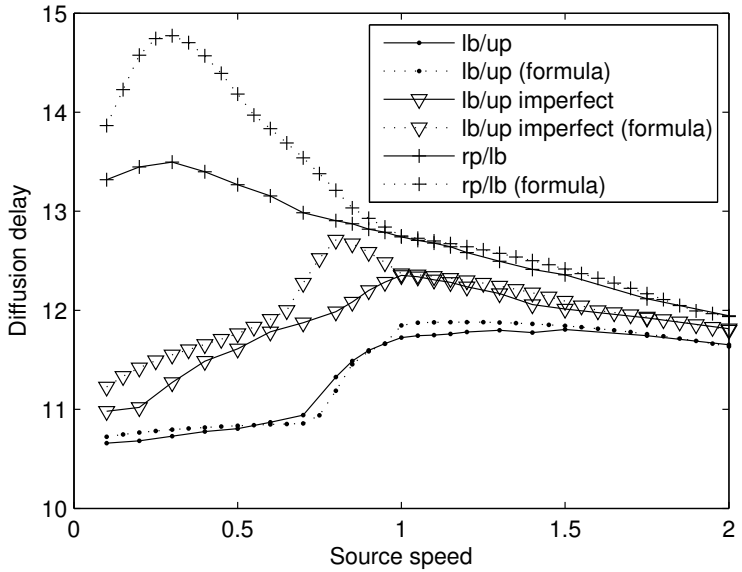


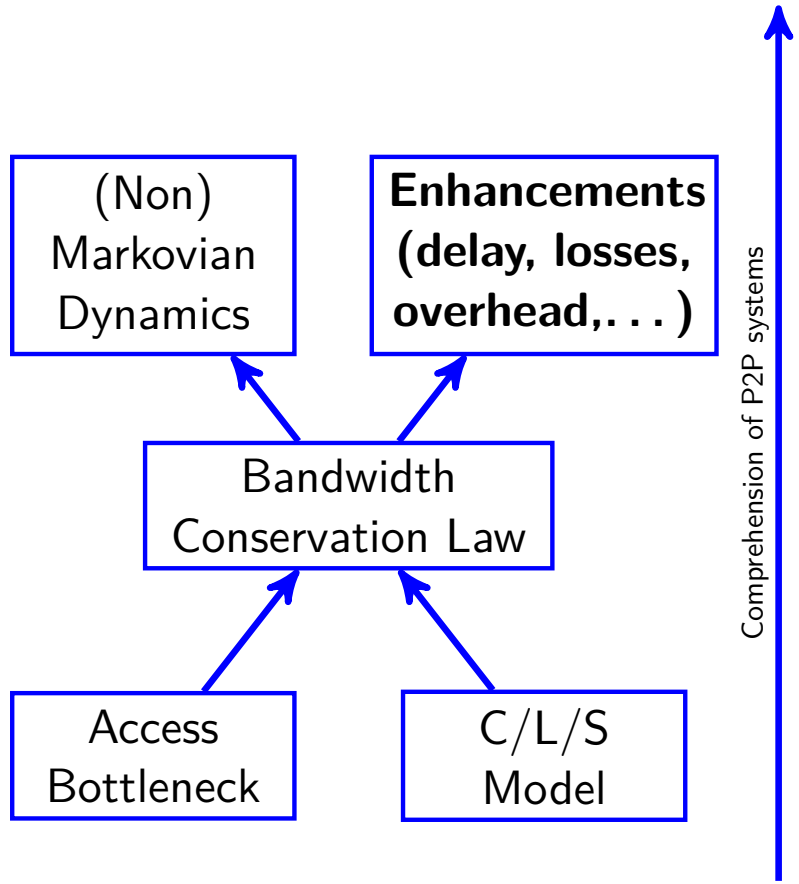


Modeling P2P systems: big picture

We build from there

- Dynamics of the system
- QoS analysis

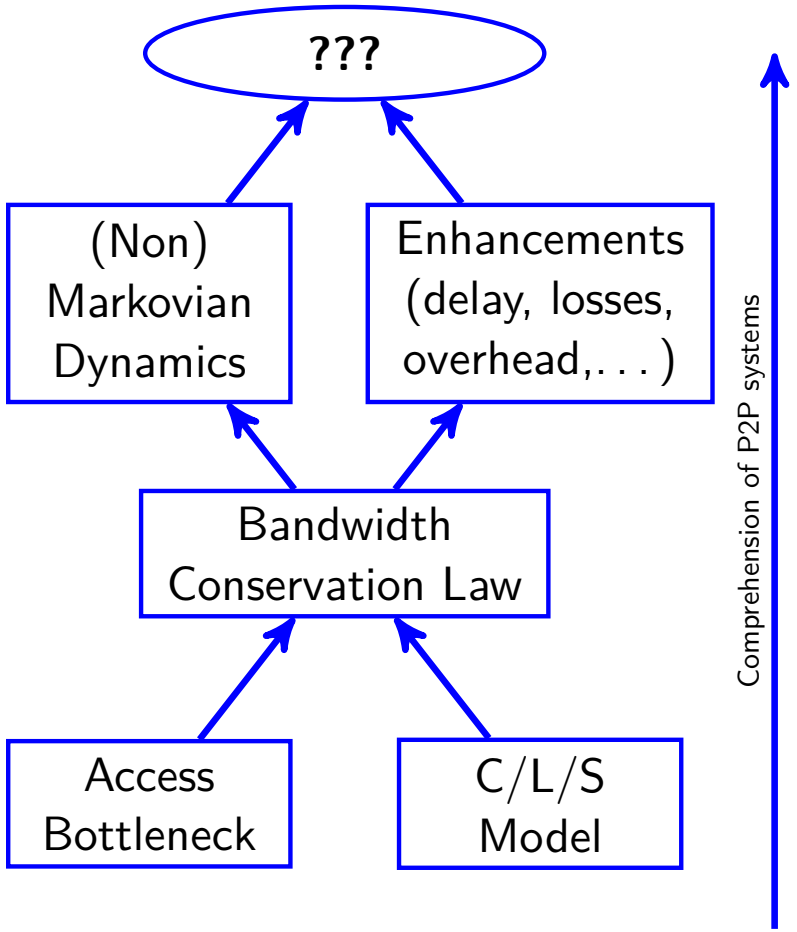




Modeling P2P systems: big picture

We build from there

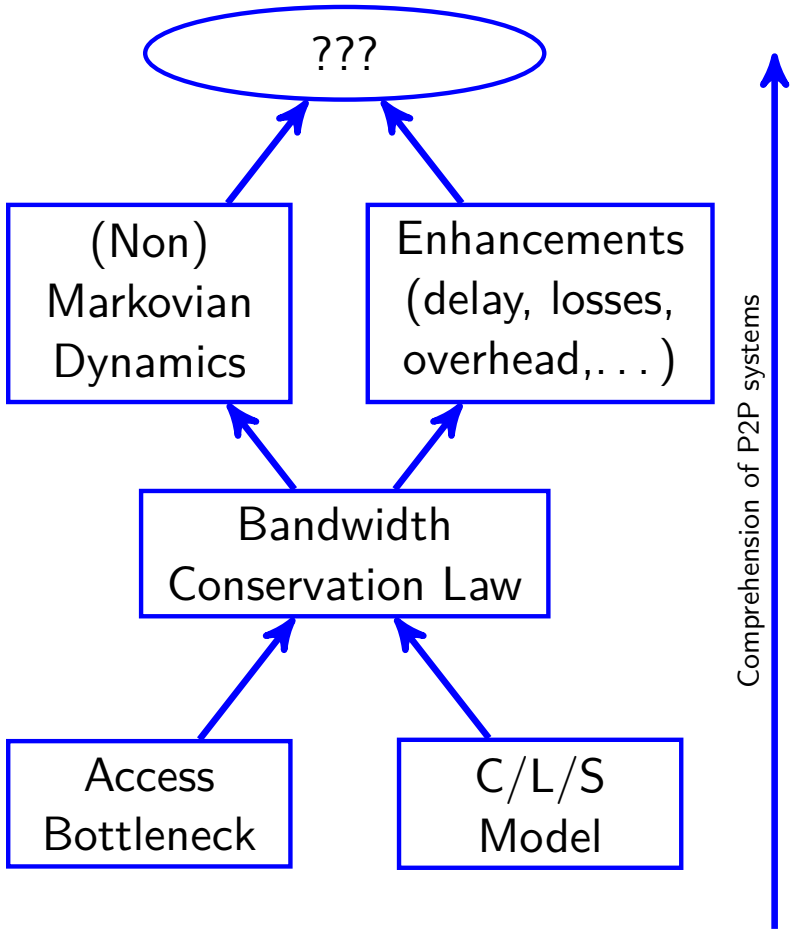
- Dynamics of the system
- QoS analysis
- And so on...



Modeling P2P systems: big picture

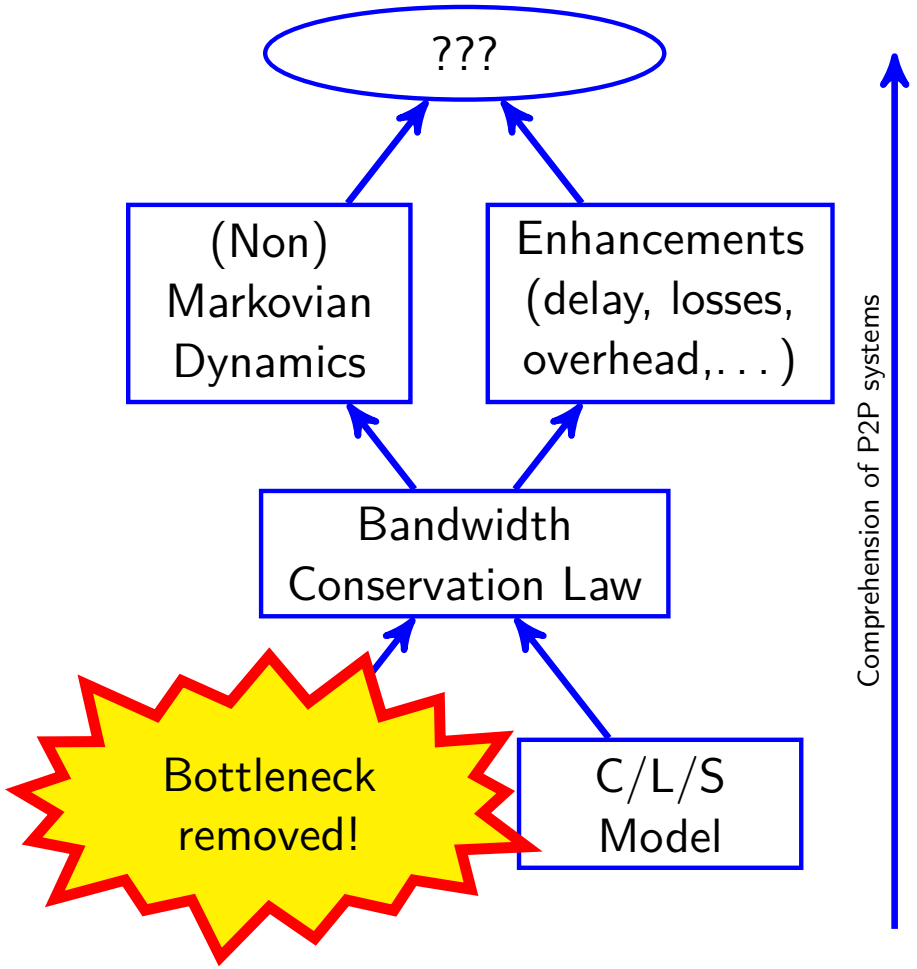
Limitations of current bandwidth model:

- Congestion in the core?
- Access is boosted (FTTH)
- Other paradigms (wireless NC)



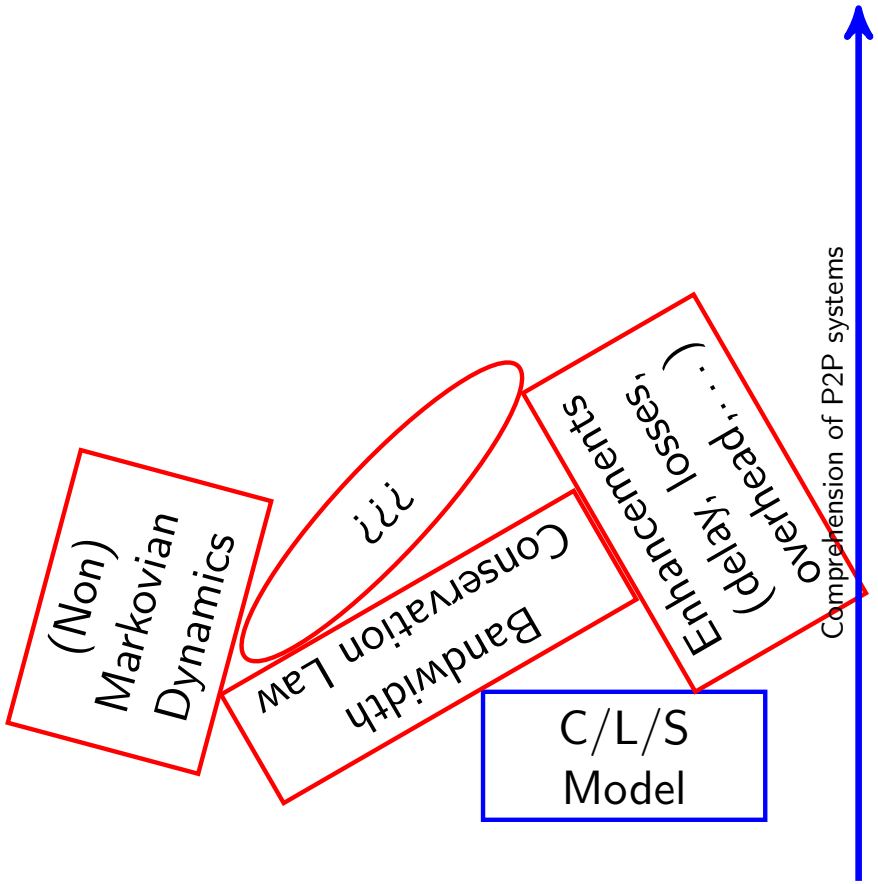
Modeling P2P systems: big picture

Q: What if bandwidth model does not hold anymore?



Modeling P2P systems: big picture

Q: What if bandwidth model does not hold anymore? A: Oops... It's all broken!



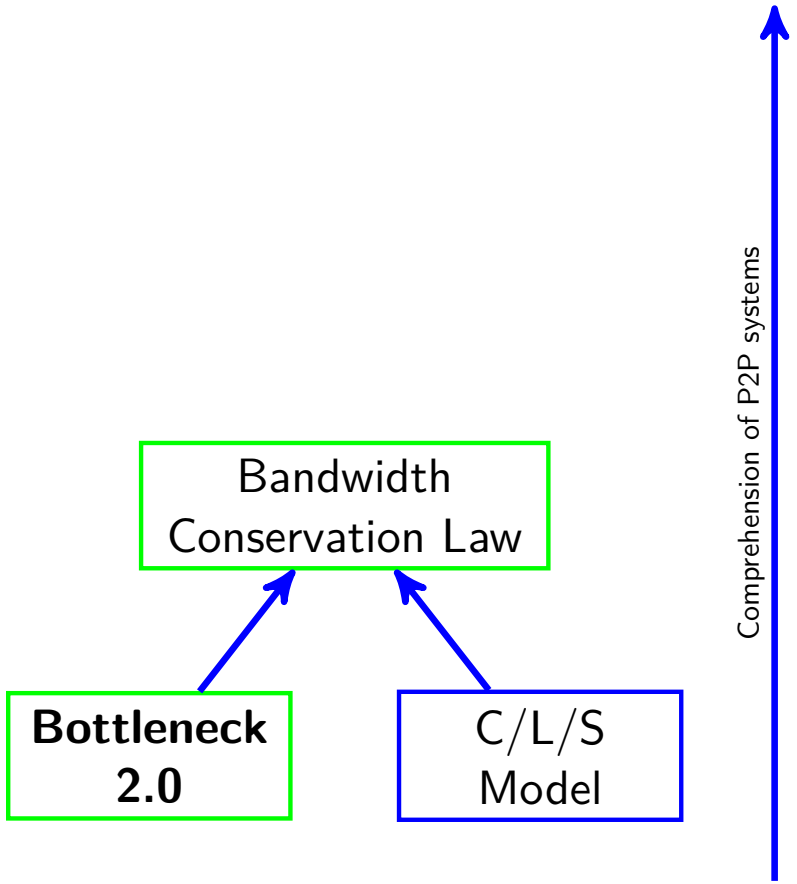
Modeling P2P systems: big picture

Our goal:

- change bottleneck model (complexity/realism trade-off)
- *And stoop and build 'em up with worn-out tools*

Remark

- 1.0 results don't apply
- But they may give ideas



Bottleneck 2.0

- Bottleneck 1.0 was weight on nodes
- Complete physical model: too complex
- Compromise: weight on (logical) edges

Additional hypothesis

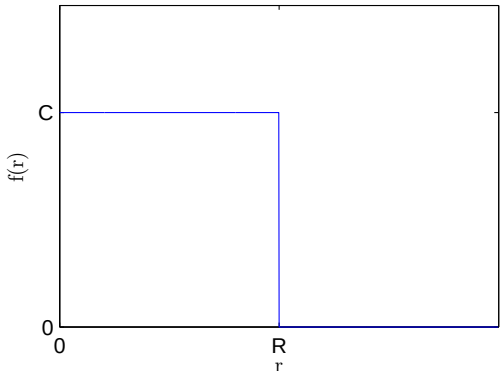
- Independence of capacities
- Symmetric exchanges between peers
- Rates can be derived from a metric

$\Rightarrow$ rate between two peers is defined by some function $f(r)$.

Examples of rate functions

- Constant radius

$$f(r) = C1_{r \leq R}$$

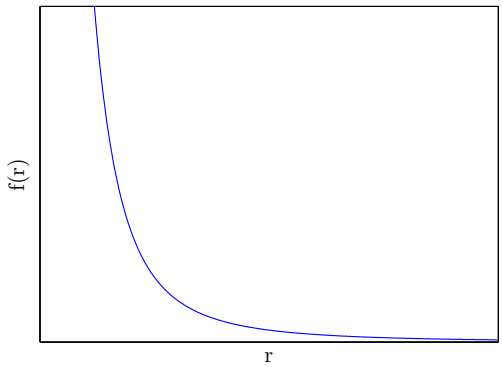


Examples of rate functions

- Constant radius
- Noisy wireless channel

$$f(r) = C_1 \log(1+\frac{C_2}{r^\alpha}),$$

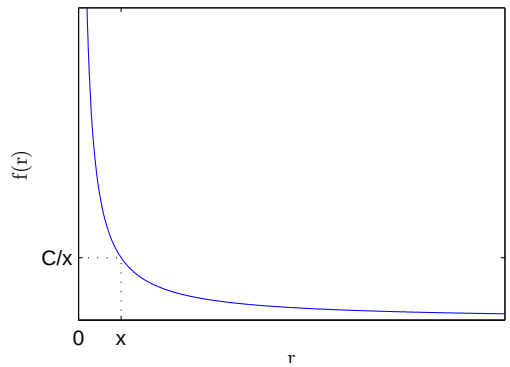
with $2 \leq \alpha \leq 4$



Examples of rate functions

- Constant radius
- Noisy wireless channel
- TCP-like function

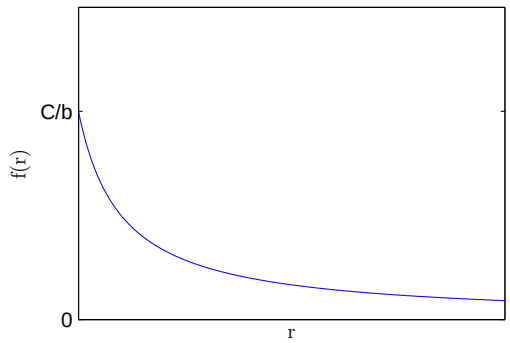
$$f(r) = \frac{C}{r}$$



Examples of rate functions

- Constant radius
- Noisy wireless channel
- TCP-like function
  - with delay offset

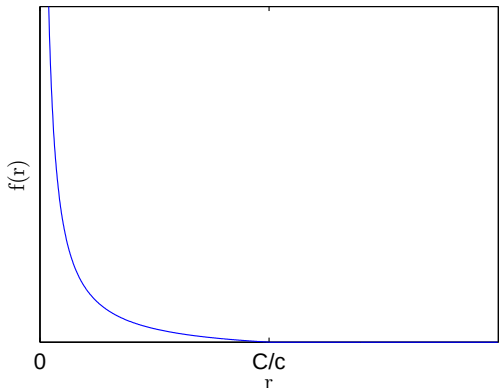
$$f(r) = \frac{C}{r+b}$$



Examples of rate functions

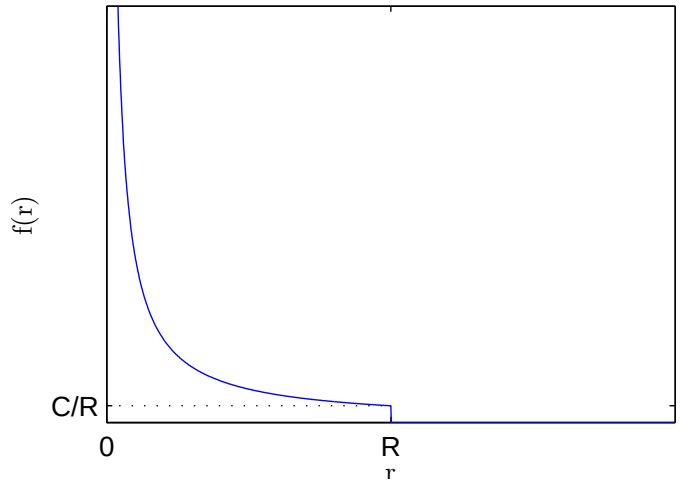
- Constant radius
- Noisy wireless channel
- TCP-like function
  - with delay offset
  - with required overhead

$$f(r) = (\frac{C}{r} - c)^+$$



In this Talk

- TCP-like rate with constant radius
  $\rightarrow$ download bandwidth
  $d(r) = \frac{C}{r} 1_{r \leq R}$



In this Talk

- TCP-like rate with constant radius
  $\rightarrow$ download bandwidth
  $d(r) = \frac{C}{r} 1_{r \leq R}$
- Euclidian plan (2D)

In this Talk

- TCP-like rate with constant radius
  $\rightarrow$ download bandwidth
  $d(r) = \frac{C}{r} 1_{r \leq R}$
- Euclidian plan (2D)
- Leechers arrive with some Poisson intensity λ

In this Talk

- TCP-like rate with constant radius
  $\rightarrow$ download bandwidth
  $d(r) = \frac{C}{r} 1_{r \leq R}$
- Euclidian plan (2D)
- Leechers arrive with some Poisson intensity λ
- Each leecher needs a random amount of data of mean F

In this Talk

- TCP-like rate with constant radius $\rightarrow$ download bandwidth $d(r) = \frac{C}{r} 1_{r \leq R}$
- Euclidian plan (2D)
- Leechers arrive with some Poisson intensity λ
- Each leecher needs a random amount of data of mean F
- Leaves as soon as download completes (freerider case)

In this Talk

- TCP-like rate with constant radius $\rightarrow$ download bandwidth $d(r) = \frac{C}{r} 1_{r \leq R}$
- Euclidian plan (2D)
- Leechers arrive with some Poisson intensity λ
- Each leecher needs a random amount of data of mean F
- Leaves as soon as download completes (freerider case)
- How data can be available is not discussed here

Warm-up: wishful thinking

- Imagine a “fluid” stationary state, described by constants

Warm-up: wishful thinking

- Imagine a "fluid" stationary state, described by constants
- There is a density β_f of peers

Warm-up: wishful thinking

- Imagine a "fluid" stationary state, described by constants
- There is a density β_f of peers
- A peer downloads at speed $d_f = \int_0^R \frac{C}{x} \beta_f 2\pi x dx = 2\pi RC\beta_f$

Warm-up: wishful thinking

- Imagine a "fluid" stationary state, described by constants
- There is a density β_f of peers
- A peer downloads at speed $d_f = \int_0^R \frac{C}{x} \beta_f 2\pi x dx = 2\pi RC\beta_f$
- Its life expectancy is $W_f = \frac{F}{d_f} = \frac{F}{2\pi RC\beta_f}$

Warm-up: wishful thinking

- Imagine a "fluid" stationary state, described by constants
- There is a density β_f of peers
- A peer downloads at speed $d_f = \int_0^R \frac{C}{x} \beta_f 2\pi x dx = 2\pi RC\beta_f$
- Its life expectancy is $W_f = \frac{F}{d_f} = \frac{F}{2\pi RC\beta_f}$
- Little's law: $\beta_f = \lambda W_f$, so $\beta_f = \sqrt{\frac{\lambda F}{2\pi RC}}$

Warm-up: wishful thinking

- Imagine a "fluid" stationary state, described by constants
- There is a density β_f of peers
- A peer downloads at speed $d_f = \int_0^R \frac{C}{x} \beta_f 2\pi x dx = 2\pi RC\beta_f$
- Its life expectancy is $W_f = \frac{F}{d_f} = \frac{F}{2\pi RC\beta_f}$
- Little's law: $\beta_f = \lambda W_f$, so $\beta_f = \sqrt{\frac{\lambda F}{2\pi RC}}$
- Therefore $W_f = \frac{\beta_f}{\lambda} = \sqrt{\frac{F}{\lambda 2\pi RC}}$

Warm-up: wishful thinking

- Imagine a "fluid" stationary state, described by constants
- There is a density β_f of peers
- A peer downloads at speed $d_f = \int_0^R \frac{C}{x} \beta_f 2\pi x dx = 2\pi RC\beta_f$
- Its life expectancy is $W_f = \frac{F}{d_f} = \frac{F}{2\pi RC\beta_f}$
- Little's law: $\beta_f = \lambda W_f$, so $\beta_f = \sqrt{\frac{\lambda F}{2\pi RC}}$
- Therefore $W_f = \frac{\beta_f}{\lambda} = \sqrt{\frac{F}{\lambda 2\pi RC}}$
- Remark: peer sees $N_f = \beta_f \pi R^2 = \sqrt{\frac{\pi \lambda F R^3}{2C}}$

Dimensional analysis

Four parameters, three physical units

- Range R, in m;
- TCP constant C, in $bits.s^{-1}.m$;
- Intensity λ, in $m^{-2}.s^{-1}$;
- File size F, in $bits$.

π-theorem

- system can be described by an equation with just $4 - 3 = 1$ dimensionless parameter ρ.
- Applying theorem gives $\rho = \frac{\lambda F R^3}{C}$.
- For convenience, we use $N_f = \sqrt{\frac{\pi\rho}{2}}$ instead

Dimensional analysis

Using it: recipe

- Compute the properties of ONE single system
- With proper scaling, results apply to ALL (R, C, λ, F) systems with same N_f.

Practical example: average download time W

- DA proves that there is a function $M(N_f)$ such that

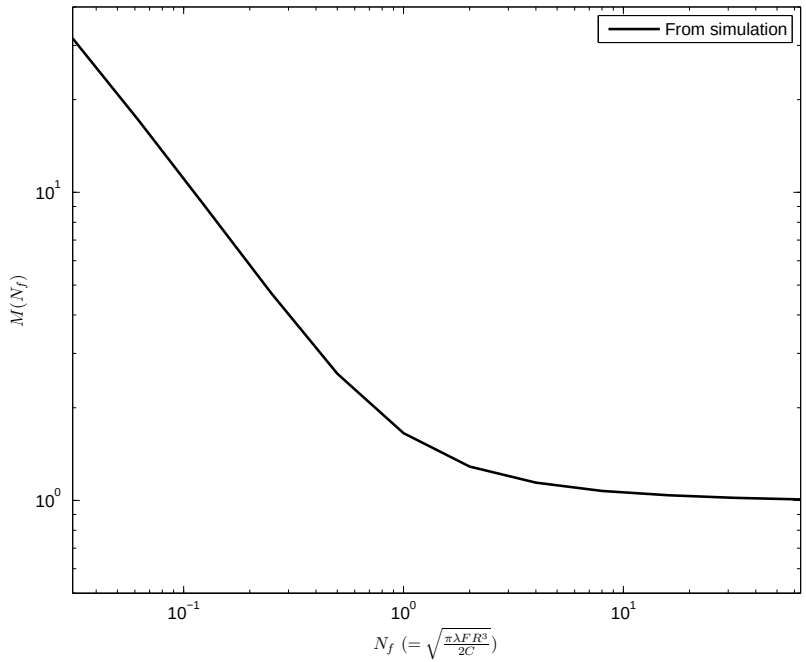
$$W(\lambda, F, C, R) = M\left(N_f(\lambda, F, C, R)\right)\sqrt{\frac{F}{\lambda 2\pi CR}} = M(N_f)W_f$$

- M can be easily sampled (by simulations)
- we get the average latency for all systems of the model!

One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $N_f \ll 1$ is called *hard core*
- N_f inbetween is *intermediate*



A remark: Palm effect

- A leecher increases the speed of its neighbors

A remark: Palm effect

- A leecher increases the speed of its neighbors
- $\rightarrow$ it makes them disappear faster

A remark: Palm effect

- A leecher increases the speed of its neighbors
- $\rightarrow$ it makes them disappear faster
- $\rightarrow$ it "sees" less neighbors than a random point in space

A remark: Palm effect

- A leecher increases the speed of its neighbors
- $\rightarrow$ it makes them disappear faster
- $\rightarrow$ it "sees" less neighbors than a random point in space
- $\rightarrow$ intuitively, interaction leads to some repulsion

A remark: Palm effect

- A leecher increases the speed of its neighbors
- $\rightarrow$ it makes them disappear faster
- $\rightarrow$ it "sees" less neighbors than a random point in space
- $\rightarrow$ intuitively, interaction leads to some repulsion
- Indeed, we proved repulsion $\rightarrow M(N_f) \geq 1$

Fluid limit: idea

- In the fluid limit, leechers have lot of neighbors
- Impact of one single leecher is minimal
- Palm effect can be neglected
- $\rightarrow$ back to wishfulland!

Fluid limit: results

Neglecting Palm gives:

- $M = 1$
- $W = W_f = \sqrt{\frac{F}{\lambda 2\pi CR}}$ (with random distribution)

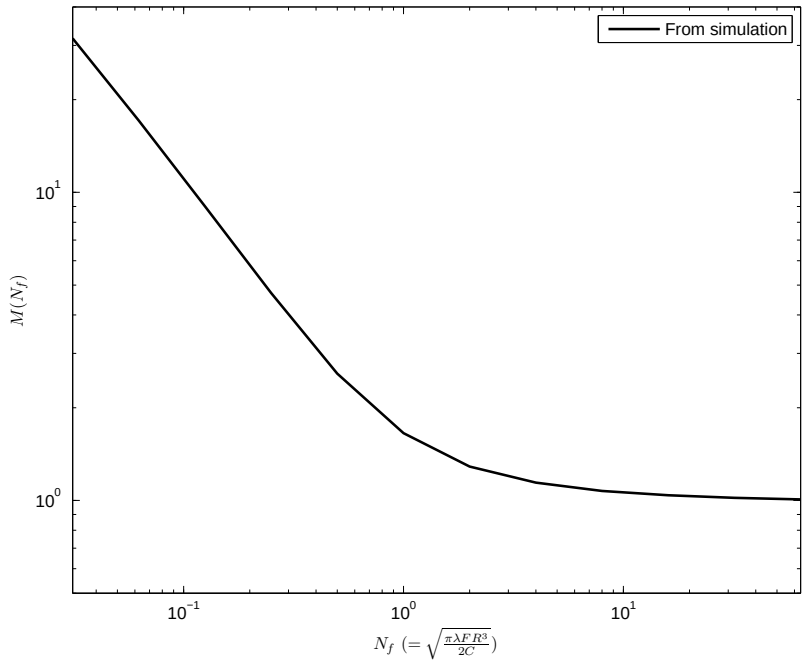
Remark

N_f increases and W decreases with $\sqrt{\lambda}$: supra-scalability of the fluid limit

One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $N_f \ll 1$ is called *hard core*
- N_f inbetween is *intermediate*

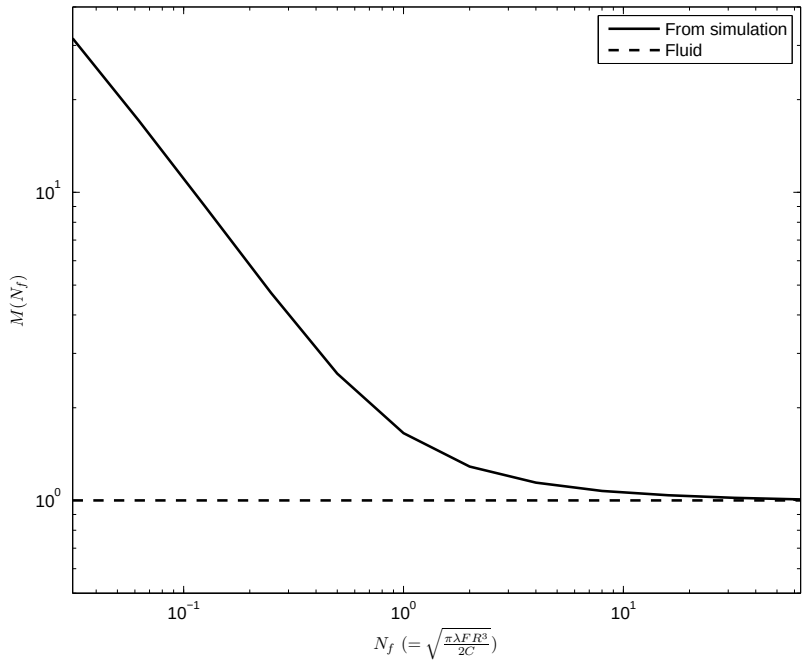


One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $\Rightarrow$ OK!
- $N_f \ll 1$ is called *hard core*

- N_f inbetween is *intermediate*



Hard core: idea

Hard core is

- Almost instant repulsion
- A lot of stillness
- Brief actions

- HC is borderline as a realistic P2P scenario
- HC may be a good model for WSNs and DTNs.

Hard core: results

Hard core seen as a ball packing process

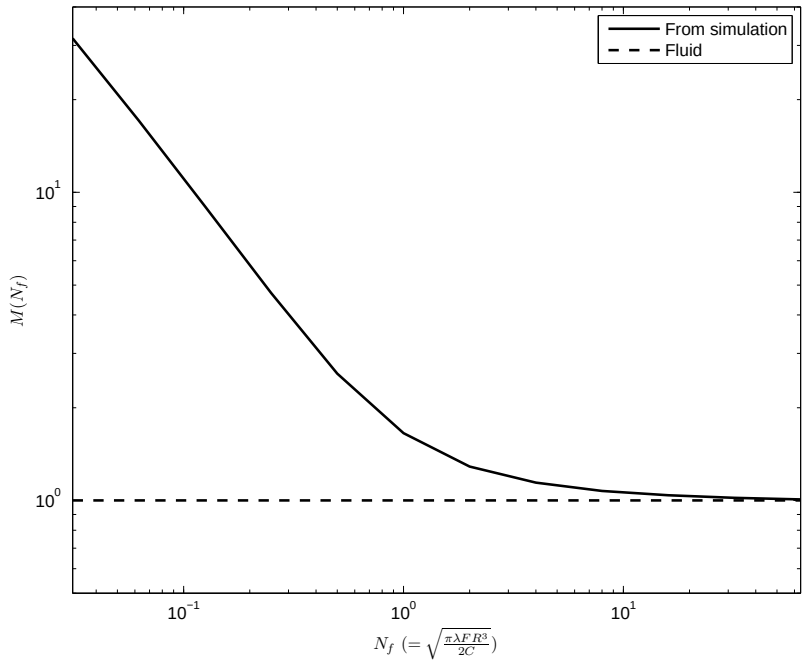
- $M = \frac{1}{N_f}$
- $W = \frac{1}{\lambda \pi R^2}$ (unaffected by C or F)
  - a leecher "instantly" disappears with probability $\frac{1}{2}$
  - Otherwise its download time is random with mean $2W$

One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $\Rightarrow$ OK!
- $N_f \ll 1$ is called *hard core*

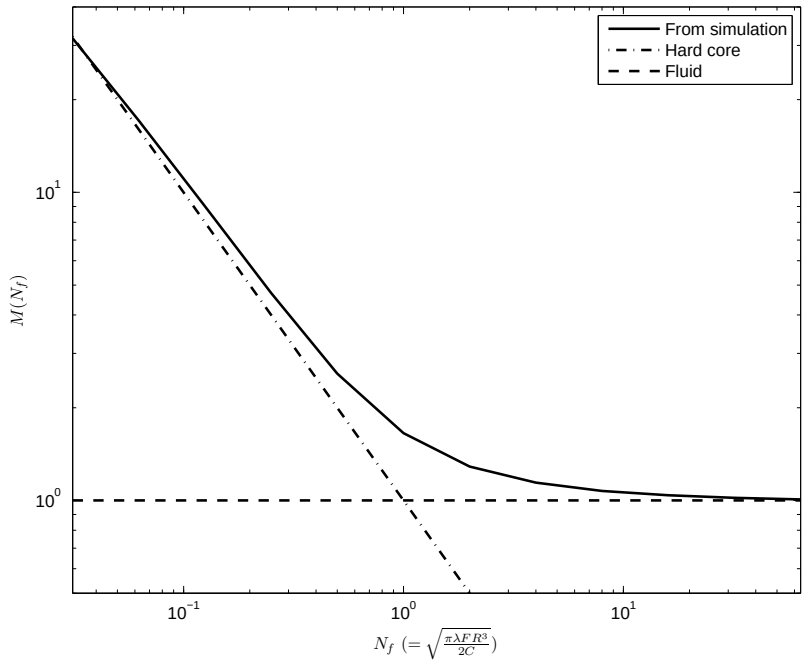
- N_f inbetween is *intermediate*



One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $\Rightarrow$ OK!
- $N_f \ll 1$ is called *hard core*
- $\Rightarrow$ OK!
- N_f inbetween is *intermediate*



Intermediate: idea

- Clear repulsion
- $\rightarrow$ fluid cannot apply
- But this not true hard core
- $\rightarrow$ hard core cannot apply

$\Rightarrow$ Complete description difficult

Intermediate: results

- Heuristic based on considering a first order estimate of Palm
- Gives M solution of

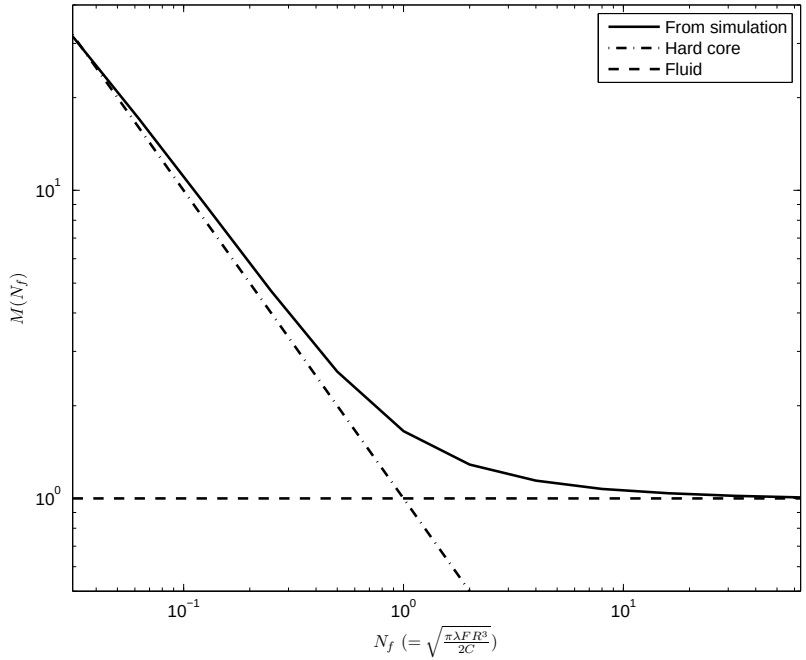
$$M^2 \left(1 - \frac{M}{2N_f} \ln(1 + \frac{2N_f}{M})\right) = 1$$

- Also predicts fluid and hardcore limits
- Just average value, no formula for distribution (yet?)

One function to describe them all

Three cases to look at

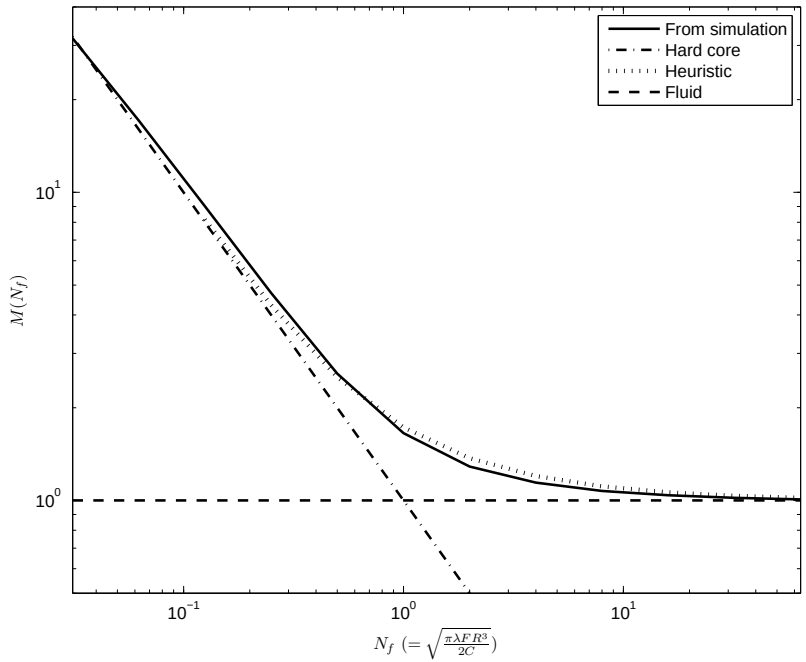
- $N_f \gg 1$ is called *fluid*
- $\Rightarrow$ OK!
- $N_f \ll 1$ is called *hard core*
- $\Rightarrow$ OK!
- N_f inbetween is *intermediate*



One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $\Rightarrow$ OK!
- $N_f \ll 1$ is called *hard core*
- $\Rightarrow$ OK!
- N_f inbetween is *intermediate*
- $\Rightarrow$ almost OK!

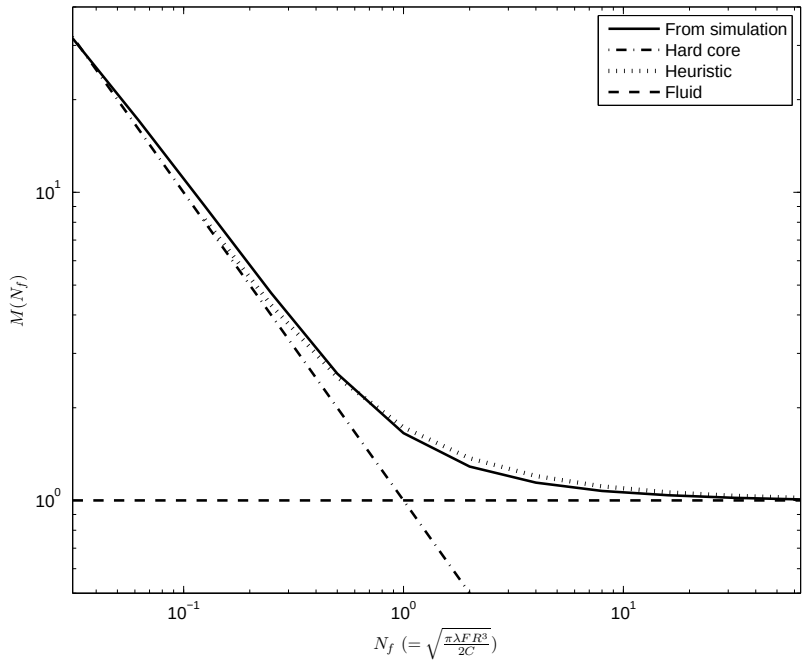


One function to describe them all

Three cases to look at

- $N_f \gg 1$ is called *fluid*
- $\Rightarrow$ OK!
- $N_f \ll 1$ is called *hard core*
- $\Rightarrow$ OK!
- N_f inbetween is *intermediate*
- $\Rightarrow$ almost OK!

$\Rightarrow$ OK!



Variants

Our model is sheer elegance in its simplicity, but

- increasing realism is a good thing
- Warning: π-theorem gets weaker with extra parameters

Variants that still allows some use of DA

- Generalization to $f(r)$ variants
  (including offsets, e.g. overhead, additive latency)
- Any type of metric space
- Abandonment
- Bottleneck 1.0 backward compatibility
- Seeders

PLAN

Conclusion

- We can understand how current things work
- We can anticipate the evolution

Live streaming

Video-on-Demand

Distributed storage

A P2P approach to CCN

Voting Systems