Heterogeneity in Data-Driven Live Streaming: Blessing or C

Fabien Mathieu

Hot-P2P, 04/23/2010

Roadmap

Problem statement

Model

Motivation

Optimal broadcasting of a single chunk

Many-to-one

One-to-one

One-to-c

Towards fast broadcasting of a stream of chunks

Curses

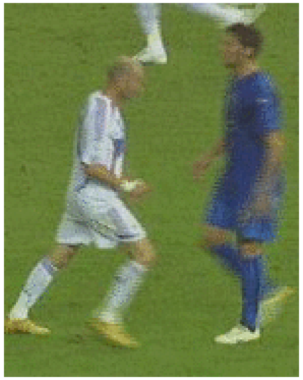
Blessings

The Live Streaming Problem

A live stream (streamrate *s*) to watch...

Injected by a server of capacity $U_c := n_0 s$ s

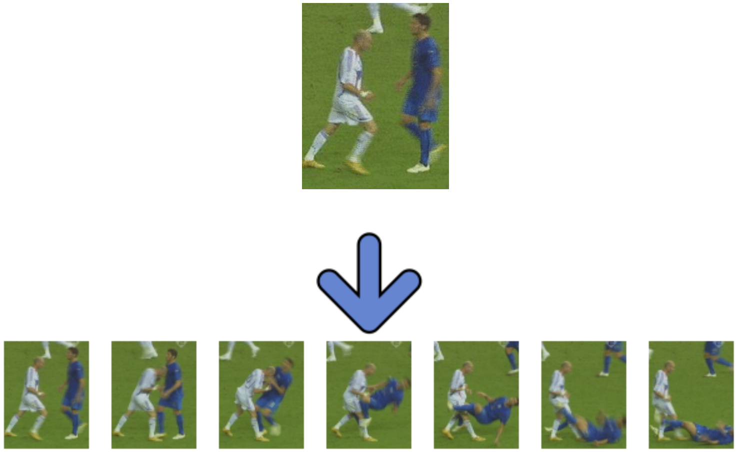
Watched by a set of *N* peers...



Goal #1: make it work!

Goal #2: make it fast!

Chunk-based (aka data-driven) diffusion



Chunk-based: the stream is split into chunks of equal size;

Integrity-rule: only forward fully received chunks;

Upload-constrained: delay comes from upload bandwidth

u1≥...≥⊓𝒩∅

ℬ⊣\⌈⊒⟩⌈⊔⟨⊣∇⌉⌉§√∇⌉∫∫⌉⌈⟩\⌋⟨⊓\∥∫√⌉∇∫⌉⌋≀\⌈

𝒩≀≀⊑⌉∇↕⊣†⌋≀\∫⊔∇⊣⟩\⊔∫

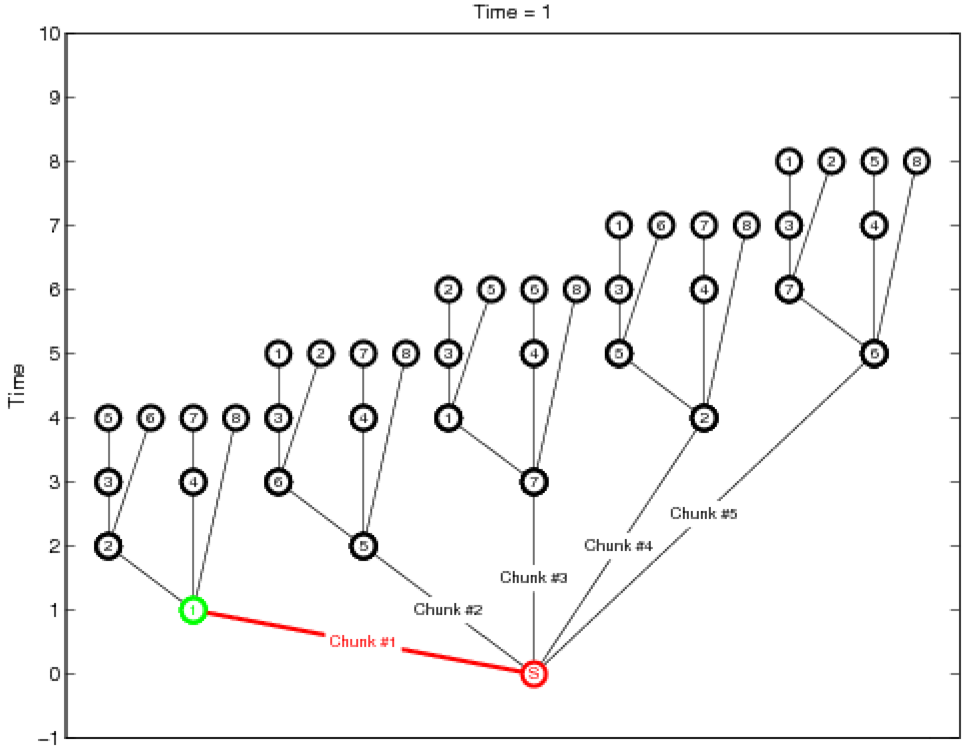
→ Oversimplified model to focus on heterogeneity

Optimal delay in the homogeneous case

Homogeneity allows to work in slotted time

The best way to broadcast one chunk takes $\frac{\lceil \log_2(N/n_0) \rceil}{u}$

Extends to a stream of chunks by permutation of the single chunk tree



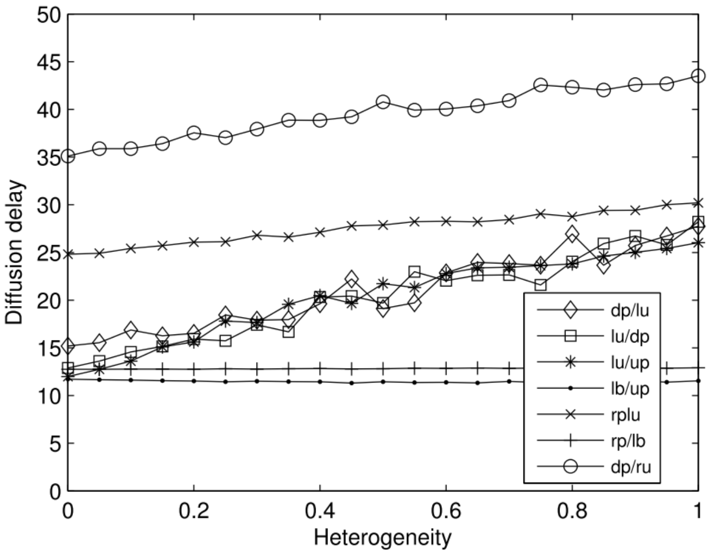
Optimal delay in the heterogeneous case

Nice formula for the optimal single chunk transmission: failed

Direct link single chunk / stream of chunks: failed

Intuition: should be faster (centralization is the extreme case)

In practice:



Summary: heterogeneity is a b....

Model extension: forwarding policies

Authorize collaborations, or force parallelism:

Many-to-one: any set of peers with a chunk can forward that chunk in time 1/Σιυι.

Νοτ ρεαλιστιχ ατ αλλ, βυτ σο εασιερ το χομπυτε;

Ωιλλ ηελπ υνδερστανδ τηε οτηερ μοδελσ.

Ονε–το–ονε (μονο–σουρχε): α πεερ *ι* ωιτη α χηυνκ χαν φορωαρδ ιτ ιν τιμε 1/υι.

τηε γενυινε μοδελ;

σλοωερ τηαν μανψ–το–ονε, βυτ μορε ρεαλιστιχ.

Ονε–το–*χ* (παραλλελισμ): α πεερ *ι* νεεδσ ατ λεαστ χ/υι το φορωαρδ α χηυνκ, υπ το χ πεερσ σιμυλτανεουσλψ.

Εξτενδ τηε χλασσιχαλ μοδελ;

παραλλελισμ χαν αϖοιδ βανδωιδτη ωαστε, βυτ ιτ σλοωερ.

Delays under the model

Focus on full dissemination (lossless) delay

- D: minimal delay for a single chunk transmission (no competition);
- $\rightarrow D_m, D_1, D_c$.
- $\tilde{D}$: min-max delay for an infinite stream of chunk ($\tilde{D} \geq D$);
- $\tilde{D}_m, \tilde{D}_1, \tilde{D}_c$.

Example

3 peers, s=1, n0=u1=1, u2=u3=1/2

Exactly the bandwidth required according to Bandwidth Conservation Law (BCL)

Equivalent homogeneous system: n0=1, u=2/3

	Many-to-1	1-to-1	1-to-2	Eq. homo.
D	$\frac{5}{3}$	2	2	3
$\tilde{D}$	3	3	4	3

Streaming is the issue

Single CHUNK

Results for the single chunk transmission

Dm is given by a simple greedy algorithm:

Gives

$$D_m = \sum_{k=s}^{N-1} \frac{1}{U_k}, \text{ with } U_k = \sum_{i=1}^{k} u_i$$

Absolute, tight, lower bound for chunk transmission.

Homogeneous case:

$$D_m^u \approx \frac{\ln(\frac{N}{n_0})}{u}$$

$$D_m^{u_1} \leq D_m \leq D_m^{\bar{u}}$$

Results for the single chunk transmission

D1 is also given by a simple greedy algorithm.

No simple, closed formula.

Theorem: $D_m \leq D_1 < 2D_m + \frac{n_0}{U_{n_0}}$

Conjecture (sigh!): $D_m \leq D_1 < \frac{D_m}{\ln(2)} + \frac{n_0}{U_{n_0}}$

Price of atomicity is $\frac{1}{\ln(2)}$

Extension to parallelism: $D_m \leq D_c < c\frac{D_m}{\ln(1+c)} + c\frac{n_0}{U_{n_0}}$

Price of parallelism is $\frac{c}{\ln(1+c)}$

STREAM OF CHUNKS

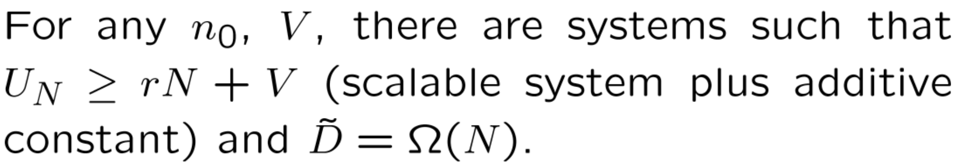
Streaming case: feasibility

- A Sig'08 papers based on substreams can be adapted to chunk-based diffusion.
- Good news: proves the feasibility of chunk diffusion as long as BCL is OK.
- Bad news: designed for latency-based transmissions, not chunk-based ones
- → poor delay guarantee: $\tilde{D}_1 < 2\frac{N-1}{\min_{u_i>0} u_i}$

Bad case scenario

In the one-to-one model, poor peer may affect the delay
if you have to use them at some time

This can happen even in overprovisionned scenarios



For any n_0, V, there are systems such that $U_N \geq rN + V$ (scalable system plus additive constant) and $\tilde{D} = \Omega(N)$.

Good case 1: emulating homogeneity

Assume we can find *u* such that

$$u \quad \frac{N-n_0}{N}\varepsilon$$ (BCL of the emulated system)

$$\sum_i \left\lfloor \frac{u_i}{u} \right\rfloor \quad N$$ (emulation condition)

Then we have

$$D_1 \leq \frac{\lceil \log_2(M/n_0) \rceil + 1}{u}, \text{ with } M := \#\{i / u_i \quad u\} \quad (M \quad N)$$

Poor peers (ui<u) are never used

Sufficient condition for *u* to exist:
 factor 2 BW provisioning for quantification

Good case 2: avoiding competition

- A sufficient condition for $\tilde{D} = D$ is $Dr \leq 1$ (time-disjoint trees)

- Necessary for the $\infty/1$ case with $u_1 > u_2$.

- Requires tremendous bandwidth over-provisioning: $\bar{u} \geq f(h)r$, with $1 \leq f(h) \leq \left[\ln / \log_2 / c \log_{1+c}\right] (\frac{N}{n_0})$

Good case 2: avoiding competition

Idea: protect the early diffusion to emulate the Dr≤∞ ⌋⊣∫⌉↙

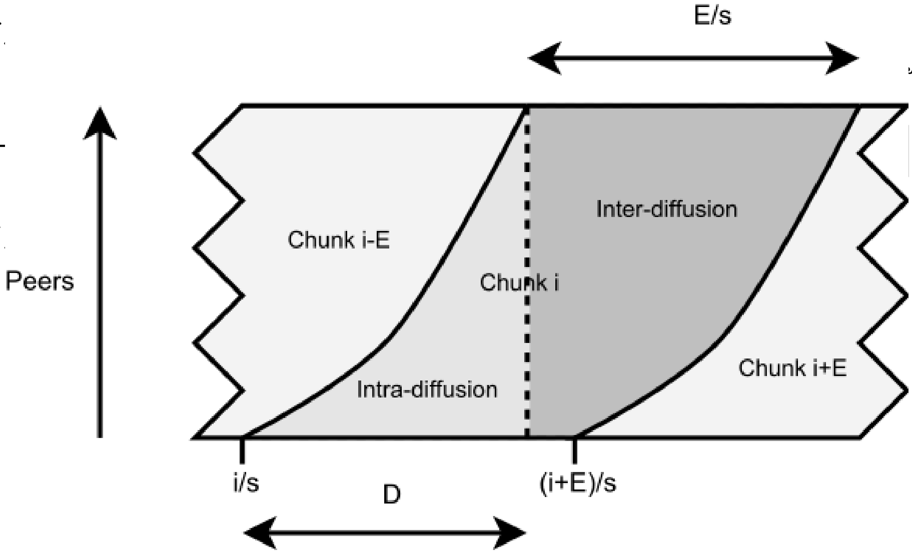
ℛ⌉⌋⟩√⌉¬

𝒮√↕⟩⊔⊔⟨⌉√⌉⌉∇∫⟩\⊔≀ℰ⌉∐⊓⊣↕∫⊓⌊∫⌉⊔∫⇔⇐ℰ≈𝒟∇⇒ 𝒜↕↕∫⊓⌊∫⌉⊔∫∫⟨≀⊓↕⌈⟨⊣⊑⌉∇≀⊓}⟨↕†⊔⟨⌉≀∇⟩}⟩\⊣↕ℬ𝒲 ⌈⟩∫⊔∇⟩⌊⊓⊔⟩≀\↙

ℛ≀⊓\⌈

𝒫∇⟩↕-

𝒮⌉⌋≀\⌈



∫⌉⊔∫↙

↙

⟨⌉∇∫⊓⌊∫⌉⊔∫

Good case 2: avoiding competition

Proper validation of the idea still on-going

Quantification effects can make some BW useless

The subset must be as similar as possible
(one monster peer in a single subset is hard to handle)

Lead to condition u≥∇⇐∞⇓∞∝ℰ⇒⇓⌋∫⊔⇔⊒⟩⊔⟨ℰ≈↕≀}⇐𝒩∝\∨′⇒↙

𝒞≀∇∇⌉∫√≀\⌈⟩\}⌈⌉↕⊣†⊣↕⇕≀∫⊔↕⟩∥⌉𝒟

ℱ≀∇∫≀⇕⌉∫↕⟩}⟨⊔↕†≀⊑⌉∇ √∇≀⊑⟩∫⟩≀\\⌉⌈∫†∫⊔⌉⇕∫⇔⊔⟨⌉∫⊔∇⌉⊣⇕⌈⌉↕⊣†⟩∫⊣↕⇕≀∫⊔↕⟩∥⌉⊔⟨⌉√⟨⊓\∥⌈⌉↕⊣†

Chunk-based model and delay: summary

For homogeneous systems, single delay=stream delay

Collaboration for one chunk transfer is not that useful

Parallelism is not that scary

Heterogeneity speeds up single delay

Some bandwidth overprovisioning seems to be required in the general case

And then? Limits of the chunk-based model

Chunks comes from BitTorrent's world

Integrity purpose

Great for unstructured approaches

Big chunk reduce overhead

Good for theory

Quantification

Delay expressed as bandwidth

But when streaming is concerned...

Need to go down to latency timescales

Limits of the model validity

A possible lead for future work

Do we really need strong integrity mechanisms?

Try to learn from the stripe world

Thank you very many!

No interactive questions, but you can

- Have a look at the paper
- Email me (fabien.mathieu@orange-ftgroup.com)