Heterogeneity in Data-Driven Live Streaming: Blessing or Curse?

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 $U_c := n_0 s$.

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

\[ u_1 \geq \ldots \geq u_N \]

Bandwidth are expressed in chunks per second

No overlay constraints

\( \rightarrow \) 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

$\Theta \left( g_2 \left( \frac{N}{n_0} \right) \right)$

The formula $\frac{u}{u}$

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 $1/\Sigma_i n_i$.

Not realistic at all, but so easier to compute; will help understand the other models.

One-to-one (mono-source): a peer $i$ with a chunk can forward it in time $1/n_i$.

the genuine model;

slower than many-to-one, but more realistic.

One-to-$\chi$ (parallelism): a peer $i$ needs at least $\chi/n_i$ to forward a chunk, up to $\chi$ peers simultaneously.

Extend the classical model;

parallelism can avoid bandwidth wastage, but it slower.
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, $s=1$, $n_0=1$, $u_1=1$, $u_2=u_3=1/2$

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

Equivalent homogeneous system: $n_0=1$, $u=2/3$

<table>
<thead>
<tr>
<th></th>
<th>Many-to-1</th>
<th>1-to-1</th>
<th>1-to-2</th>
<th>Eq. homo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>$\frac{5}{3}$</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$\tilde{D}$</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Streaming is the issue
Single CHUNK
Results for the single chunk transmission

$D_m$ 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:

$$D_u \approx \frac{\ln\left(\frac{N}{n_0}\right)}{u}$$

Homogeneous case:

$$D_{\mu_1} \leq D_m \leq D_{\mu}$$
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 \( \frac{1}{\ln(1+c)} \)
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

- \( \tilde{D}_1 < 2 \frac{N-1}{\min_{u_i > 0} u_i} \)
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

$$u \overset{N - n_0}{N}(BCL \text{ of the emulated system})$$

$$\forall \overset{N}{i} \forall$$

(emulation condition)

Then we have

$$D_0 \overset{\log_2(M / n_0) + 1}{u}, \text{ with } M := \#\{i / u_i \leq u\} \ (M \ N)$$

Poor peers ($u_i < u$) 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:
  \[ \bar{u} \geq f(h)r, \text{ with } 1 \leq f(h) \leq \left[ \ln / \log_2 / c \log_{1+c} \right] \left( \frac{N}{n_0} \right) \]
Good case 2: avoiding competition

Idea: protect the early diffusion to emulate the Dr≤∞

Recipe:
- Split the peers into E equal subsets, \( E = \text{Dr} \)
- All subsets should have roughly the original BW distribution.
- Round-robin the chunk injection among the subsets.
- Primary goal: intra-diffusion of the chunk.
- Secondary goal (when idle): broadcast to other subsets.
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 $u \geq \nabla \iff \{ \exists f(1+cst) \} \subseteq \mathcal{E}$

\[
\mathcal{C}(\nabla \mathcal{V}) \iff \{ \exists f(1+cst) \} \subseteq \mathcal{E}
\]

\[
\mathcal{F}(\nabla \mathcal{V}) \iff \{ \exists f(1+cst) \} \subseteq \mathcal{E}
\]
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 come 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)