Adaptive HTTP streaming is a simple yet increasingly popular means to deliver streaming media over broadband access.

But...

Congestion can happen anywhere

CDN over-provisioning is expensive [Leighton]

User experience can still suffer from short quality drops, unless we use a large reception buffer

So...

Can distributed streaming [Nguyen et al.] be applied to adaptive HTTP streaming and what would the benefits be?
Adaptive HTTP streaming (recap)

- The content is encoded in different qualities/bit rates
- Each of the resulting streams is split into chunks of same duration (e.g. 2 seconds) and GOP aligned
Compared to legacy adaptive streaming, the principal specificity is concurrent operation of multiple servers by requesting adjacent byte ranges of the same chunk from all servers and managing timeouts.
Multipath with multihomed client

- Single server
- Client has multiple access networks
- Routing configured properly
### Bandwidth estimation - for each iteration

\[
B_i = \text{Bytes} \cdot 8 / \text{elapsed}
\]

\[
\text{avg}_{i+1} = (1 - \alpha) \cdot \text{avg}_i + \alpha \cdot B_i
\]

\[
\Delta_i = |B_i - \text{avg}_i|
\]

\[
\text{var}_{i+1} = (1 - \beta) \cdot \text{var}_i + \beta \cdot \Delta_i
\]

\[
\hat{B}_i = \text{avg}_i - c \cdot \text{var}_i
\]

- Smaller values for \(\alpha\) and \(\beta\) make the estimation change slowly
- Large value for \(c\) keeps a safety margin to avoid congesting the link
- In our implementation we have set \(\alpha=1/16\), \(\beta=1/8\) and \(c=1\)
  - The chosen values tend to exploit the link without causing congestion
  - Good choice for HD/3D TV which are bandwidth hungry
- Multipath estimation is the sum of individual paths’ estimations
Download timeout management

1 chunk, e.g. 300kBytes

- Idle servers are reused to download another range of the same chunk
- Downloaded byte ranges are merged and complete chunks are passed to the hardware decoder

Sent to Server1
HTTP/1.1 GET
Range: 0-140kBytes

To Server2
HTTP/1.1 GET
Range: 140-200kBytes

To Server3
HTTP/1.1 GET
Range: 200-kBytes
Experimental setup - Emulation testbed

- Standard Apache servers
- Network emulator: Linux 2.6.31-22 with NetemCLG 1.1 patch
- Multipath-able set-top-box, collecting traces over 500 iterations
- Encoded content uploaded to web hosting servers
- Use ADSL and 3G access network (Orange, SFR and Orange 3G)
Results

- Lowest bit rate/quality is significantly higher with multipath
  - Users remember worst quality [Hamberg and De-Ridder]
- Drop-offs are less frequent and attenuated with multipath
  - Sharp drop-offs have heavy impact on overall experience [Pinson and Wolf]
Results

**Mono path**

- Estimated
- Measured
- Total estimated
- Requested

**Multipath (3)**

- Estimated
- Measured
- Total estimated
- Requested

**Bit rate**

- Time

**Max** = 8924 ms

**Avg** = 2728 ms

- Late
- 1M
- 2M
- 3M
- 4M
- 5M
- 6M

**Time**

- >90%

- 67%

- 30%

- 2%
Observations - Number or paths

- Benefits are significant starting with 2 paths, limit reached with 6 paths
Observations - Loss burst length

- Multipath is especially efficient with large loss bursts where single path would require buffering

![Graphs showing comparisons between single and multipath for different loss burst lengths and delay times, highlighting the reduced re-buffering time with multipath.]
Observations - Unbalanced delays

- Paths with unbalanced delays have a limited performance hit [Wang et al.]

![Graphs showing performance distribution for single and multiple paths with different delay levels.](image)

1. Single path: 1 path 1 conn 15ms delay
   - max=8924ms
   - avg=3006ms

2. 3 slightly unbalanced paths: 3 paths 15/30/45ms delay

3. 3 balanced paths: 3 paths 15ms delay

4. 3 heavily unbalanced paths: 3 paths 15/45/75ms delay

---

14

---

technicolor
Observations – Internet multiserver

- Real world test: provision video chunks to free web hosting servers on the Internet
- Alternatively stream through DSL access
  - From fastest server alone
  - From 3 servers

1 fast server

3 connections to fast server

1 fast + 2 slow servers

3 connections towards fast server
Observations - Internet dual access

- Interconnection of ISPs varies greatly from one access to the other
- Paths through different accesses towards same server seem to be uncorrelated
  - Dispersion of results is amortized by concurrent use multiple paths
  - Aggregate capacity is approximately summation of individual paths’ capacities
Conclusions

- The use of multipath communication has been shown to improve adaptive streaming
- With as few as 2 available path the user experience is dramatically improved *with the same access link capacity*
- Experiments with real Internet configurations have confirmed the results found with our emulation testbed
- Applicability to the case where multipath is provided through multihoming was also demonstrated

Future work

- Fairness should be improved in presence of shared bottlenecks [Ma and Oi]
- Comparisons should be done with other rate adaptation algorithms
- Improve path selection by using correlated paths detection
- Validate network friendliness of distributed adaptive clients crowds