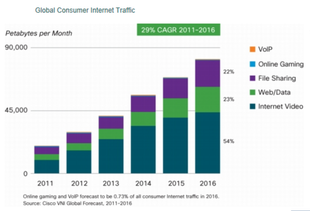
Gradient: Balancing Cost and User Utility for Live Network Streaming

Ymir Vigfusson[§], Annie Liu[†], Winnie Liu[†], Yuanchao Zhu[†], Gunnar S. Sigurbjornsson[§], Kristjan V. Jonsson[§], Qi Huang[‡], Ken Birman[‡], Daniel A. Freedman[‡†]

[§]School of Computer Science, Reykjavik University, Reykjavik, Iceland; [‡]Department of Computer Science, Cornell University, Ithaca, New York, USA; [†]School of Electronics Engineering and Computer Science, Peking University, Beijing, China

Context



* Trend: Skyrocketing demand of bandwidth

- Video takes up most of global consumer Internet Traffic
- One important type of stream is live stream

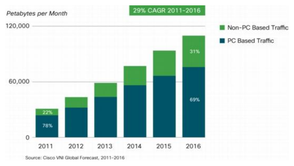
Weakness of traditional way to save bandwidth

* Caching:
  - cannot cache live stream [1]
* IP-level multicast
  - Wide-area deployment of IPmulticast has failed
* Application-level multicast
  - generally assumes a homogeneous population in terms of devices, bandwidth, costs and utility

Observation

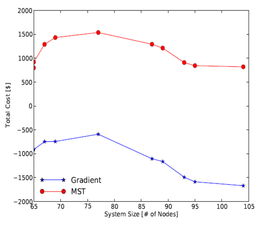
Heterogeneous end devices

Different maximum resolution, thus different demand for stream rate

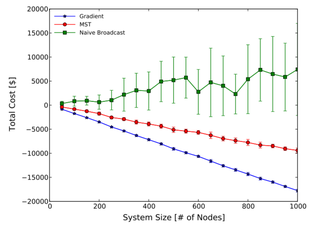


It may be better to send low-quality video to many users then high-quality to only a select or lucky few.

Emulation and Simulation Results



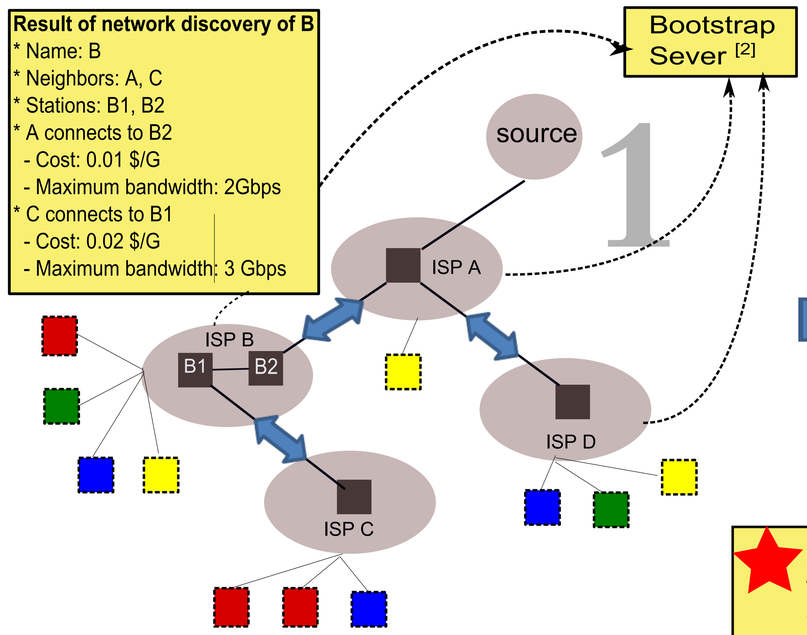
Experiment on PlanetLab. GRADIENT *uses substantially less bandwidth without sacrificing end-user utility compared to the* MST *approach.*



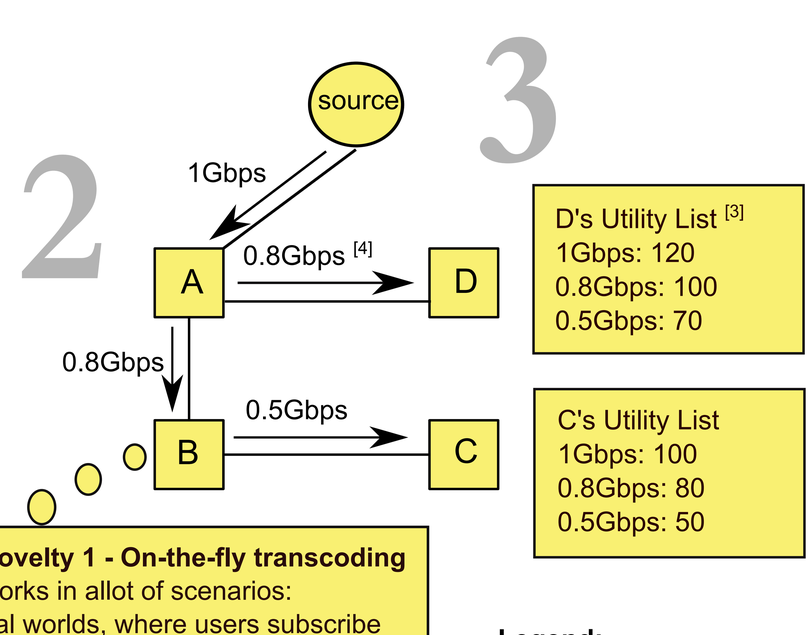
Simulation. *The* GRADIENT *optimization is effective compared to and* MST *even as system sizes scale up. Error bars represent one standard deviation over 25 trials.*

Y-axis is total cost (bandwidth cost - utility gain), measured by $; X-axis is number of Gradient nodes.
Our simulation results agree with an experimental evaluation on PlanetLab and show an improvement at scale.

Real Internet



Gradient Overlay



How Gradient system works

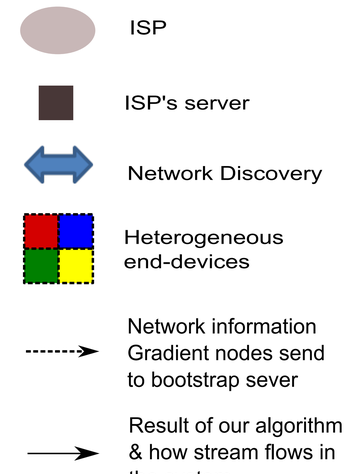
1. Collect information of real Internet
2. Create an Gradient overlay according to the collected information
3. Run the approximation algorithm on the overlay to make a trade-off between users' utility and bandwidth cost
   * *Where the trade-off lies:*
     If we send high rate stream to users, users' utility gain will be high, which we like, but the bandwidth cost is also high, which we dislike

The system can run according to the result of the algorithm to route and transcode

Key Novelty 1 - On-the-fly transcoding
This works in allot of scenarios:
* Virtual worlds, where users subscribe to updates for virtual objects at different rates
* Stock updates, where updates about some portfolios more important than others

Key Novelty 2 - Trade-off between users' utility and bandwidth cost
We derive a constant-factor approximation algorithm for the NP-hard problem of balancing the trade-off between utility and cost.

Legend:



[1] Live stream is strictly defined as fresh stream generated within 1 second
[2] Running a distributed variant of the bootstrap is preferable. we currently use a centralized server but a distributed implementation is in the works
[3] Each Gradient node has a utility list. Numbers following the rate is to measure how users want this rate: the larger the number is, the more strongly users want this rate
[4] The result rate cannot exceed the maximum bandiwidth of the link. The bandwidth cost on the link = selected stream rate * bandwidth cost/G