A Content Propagation Metric for Efficient Content Distribution

Ryan S. Peterson†*, Bernard Wong‡*, and Emin Gün Sirer†*

† Department of Computer Science, Cornell University
‡ School of Computer Science, University of Waterloo
* United Networks, LLC

August 18, 2011
Content Distribution

- Origin server
- Cache servers
- End users
BW in Client-Server

- Origin server
- Cache servers
- End users
BW in Peer-to-Peer

origin server

cache servers

end users in swarms
BW in **Antfarm**

- **Origin server**
- **Cache servers**
- **End users in swarms**
Goal

Efficiently use all available bandwidth
Problem Definition

- The general multi-swarm content distribution problem
  - **given**: hosts, swarms, and swarm memberships
  - **find**: allocation of each host’s upload bandwidth among its swarms that maximizes system-wide bandwidth
Approach

New metric that steers hosts toward a globally efficient allocation of resources

Enables each host to measure its impact on each swarm and adjust its bandwidth allocations accordingly
Approach

New metric that steers hosts toward a globally efficient allocation of resources

Content Propagation Metric
Outline

The CPM

V-Formation

Evaluation
Benefit of a Block

$p$'s choice: upload the next block to $s_1$ or $s_2$?

Which swarm will benefit more?
Determining Benefit

- What block $p$ uploads
- Distribution of blocks in the swarms
- Sizes of the swarms
- Network conditions among peers
- The direct recipient of $p$'s block

Use history to predict the future
Intuition

Measure how “fast” $p$’s blocks propagate in each swarm

Use the result as an estimate of the benefit that the swarms derive from $p$’s blocks
Content Propagation Metric

**Block propagation bandwidth**: rate that an uploaded block propagates in a fixed time interval $\tau$

**CPM**: rolling average of a peer’s recent block propagation bandwidths for a swarm
Using the CPM

• Each host measures random uploaded blocks to maintain a CPM value for each swarm

• Hosts upload to swarms with the largest CPM values when faced with competing requests

• Hosts proactively probe new swarms and swarms with stale CPM values
CPM Case Study
competition for block propagation
CPM Case Study

- Bandwidth from cache
- Time

Network nodes and bandwidth visualization.
CPM Overview

• Identifies neediest swarms
• Easy to measure
• Can allocate bandwidth from a single server
• Accounts for interference from competing hosts
Outline

The CPM

V-Formation

Evaluation
V-Formation

- Based on our hybrid architecture
- A logically centralized coordinator provides efficient bookkeeping
- A token protocol enables the coordinator to track blocks and monitor peers
Coordinator

- Measures swarm dynamics
  - tracks block transfers based on spent tokens
- Computes peers’ CPM values
  - periodically sends updates to peers
- Provides accountability
  - detects and blocks misbehaving peers
Wire Protocol Goals

• Track block transfers among peers
• Disseminate CPM values and peer lists
• Enforce peer behavior
Wire Protocol

- coordinator
- join $s_i$
- peerlist

Diagram showing network topology and protocol interactions.
Wire Protocol

cordinator

tokens

get tokens
Wire Protocol

coordinator

want block

block
Wire Protocol

- coordinator
- token
- deposit tokens
Wire Protocol

coordinator

coordinator's state

b1

time
Wire Protocol

coordinator

coordinator’s state

CPM value

announce

b1

time
Coordinator Design

- Stores membership info, propagation data, and CPMs.
- Distributed, shared state.
- Web server handles peer requests, records block propagation data continuously.
- Processors continuously process block propagation data.
Coordinator State

• **Soft state stored in memcached**
  
  • **Swarm**: peers, number of blocks
  
  • **Peers**: addr, swarms, block propagation bandwidths, CPMs
  
  • **Blocks**: swarm, propagation graph with timestamped, peer-identified nodes

• **Updated via atomic CAS operations**
Outline

The CPM

V-Formation

Evaluation
Evaluation

• Built and deployed V-Formation as a video-sharing service called FlixQ
• Uses the CPM to achieve high performance
• Coordinator scales to large deployments
Experimental Setup

• Coordinator on Amazon EC2
• 380 peers on PlanetLab with realistic bandwidth capacities
• 200 swarms based on IMDb movie popularities and sizes
• 20% of peers belong to multiple swarms
• 2 caches with different subsets of content
End-to-End Performance

- BitTorrent
- Antfarm
- V-Formation

aggregate bandwidth (KB/s) vs. time (s)
Performance of Heuristics

- V-Formation
- Largest swarm
- Global rarest
- Random
- Smallest swarm

aggregate bandwidth (KB/s)

time (s)
Scalability

- bandwidth
- state size

Number of peers vs.
Coordinator bandwidth (KB/s)
Coordinator state (MB)
Related Work

• Content Distribution Networks
  - Antfarm, Akamai, CoBlitz, CoDeeN, ECHOS, Coral, Slurpie, YouTube, Hulu, GridCast, Tribler, Joost, Huang et al. 2008, Freedman et al. 2008, ...

• P2P Swarming
  - BitTorrent, BitTyrant, PropShare, BitTornado, BASS, Annapureddy et al. 2007, Guo et al. 2005, Pouwelse et al. 2005, Zhang et al. 2011, OneSwarm, ...

• Incentives and microcurrencies
  - Dandelion, BAR Gossip, Samsara, Karma, SHARP, PPay, Kash et al. 2007, Levin et al. 2009, iOwe, ...
Conclusions

• New hybrid approach for efficient bandwidth allocation

• Decentralized metric enables hosts to measure their global benefit

• Centralized implementation drives hosts toward globally efficient use of resources

http://flixq.com