

DataShed: Monitoring and Diagnosis of Large Scale P2P Video Streaming Networks

EECS495-DIC project progress presentation

Vaibhav Rastogi
Yinzhi Cao
Instructor: Ioan Raicu

Problem Overview

- P2P paradigm benefits: scalability
- Monitoring is important
- Problem: how to monitor million-scale P2P media streaming networks
- Need for collecting traces from the entire network to one location

Motivation: A measurement study

- There must be some problems in large scale networks
 - What are they?
- The protocols are proprietary
- Attempted to reverse engineer PPLive protocol
- Things important for us
 - Peer lists: peers of a particular client
 - Buffer maps: ready to play chunks of a client
 - Indicate quality of streaming

Reverse Engineering PPLive

- Only partly successful
- Findings
 - Able to retrieve peer lists in the network
 - PPLive obfuscates buffer maps but these are necessary to infer the quality of service
- A fortunate workaround
 - Collaborated with some researchers in China
 - Use their code to crawl the UUSee network

Measurements

- Massive passive measurements of the Feb 13 Chinese New Year Celebration broadcasts on the UUSee network
- Used 300+ nodes from PlanetLab to monitor the UUSee network
- Collected ~64 GB of compressed logs over a period of 9 hours
- Analysis continuing

Analysis of logs

- Looking for phenomena like *flash crowds*
- Compile buffer maps of each monitored peer and find any specific anomalies which are common to many peers
- Geolocate the peers to observe any unexpected or non optimal peering

Property List

- Completeness
- Efficiency
- Churn resilience
- Real time
- Scalability

Proposed Solution

- Use sufficient redundancy to cope with peer churn
- Organize the network into a hierarchical structure
 - Peers send data to collector through nodes higher up in the hierarchy
 - Reduces the redundancy required by imposing a structure
- Use compression and aggregation at intermediate nodes to limit communication overheads

Partial time line

- Data Analysis: 2-3 weeks
- Theoretical analysis of the protocol: 1-2 weeks

Future Work

- Protocol implementation: 4 weeks
- Performance Evaluation
 - Simulations: 3 weeks
 - Validation by deployment over PlanetLab: 2 weeks