# Building a Real-Time Internet Traffic Map

Paper # 123


## Abstract

This paper presents the design and evaluation of an Internet traffic map system. The core enabling technique is massive and continuous measurement from end-host applications (e.g., Internet video).


## 1 Introduction

- Real-time traffic information is increasingly critical for applications to make the best decision for two reasons:
  1. More choices mean more potentials for QoS improvement but also hard to explore locally.
  2. Internet traffic becomes more and more dynamic.
  3. Need for such a real-time Internet traffic is shared by many applications, suggesting the huge impact of a service like this.
- Real-time traffic map for Internet has three requirements
  1. Coverage
  2. Overhead
  3. Real-time view
- Existing approaches do not meet all three requirement.
- Our argument is that it is feasible to build a real-time Internet traffic map (RITM) by using the massive and continous measurement from popular and bandwidth intensive applications (e.g., streaming video) running at the end hosts.
- This paper present a design and implementation of such a real-time Internet traffic map through addressing two challenges with this approach:
  1. Link-level traffic inference: we present the techniques to extrapolate link-level traffic statistics by combining different data sources, including static Internet topology information. Note the difference to network tomography.
  2. Scalability: we present a design that is able to handle massive simultaneous measurements from millions of video sessions and maintain a global view of Internet traffic map in near real-time scale.
- We evaluate the performance of RITM system and show that **XXX** (from accuracy-wise and scalability-wise)
- Finally, we use **XXX** applications to demonstrate that RITM system can significantly improvement the QoS of applications.

## 2 Motivation

We motivate the need for an RITM by using real-trace to show significant variability of (link-level and path-level) available bandwidth in both space and time.

### 2.1 Dataset

- Describe the dataset: number of sessions, client-ASNs, video objects, servers, etc.
- Link-level performance dataset: ground-truth.

### 2.2 Path-level Variability

### 2.3 Link-level Variability

## 3 System Overview

### 3.1 RTM Architecture

- Give a high-level schematic figure of RITM system with input and ouput.
- RITM system consists of three parts
  1. Data input: video measurement and topology information
  2. Inference algorithm
  3. Output: RITM
  4. Query interfaces

### 3.2 Video Measurement Input

- Video measurement samples, information associated to each session and granularity of measurement (per-minute, per-second or per-chunk)
- Client-side measurement collection

### 3.3 Topology Information Input

- Explain the static dataset of topology

### 3.4 Annotated Real-time Internet Traffic Map

- Data structure of a link and a node
- Data structure of statistics associated to each link

- Query interface of an RITM.

# 4 Link-level Traffic Inference

## 4.1 Problem Formulation

- Formulate the problem of link-level traffic inference

## 4.2 Bottleneck Inference Algorithm

- Basic algorithm
- Improvement techniques (e.g., aggregation, removal of video-induced noise, etc)

# 5 Scalable Backend

The implementation of the backend of RITM system must be scalable to handle massive simultaneous upates from client-side video sessions and process them efficiently in order to maintain a near real-time view of RTM.

**Measurement sample storage:** Store the measurement samples in Hadoop file system

**Update RTM in parallel:** Process the measurement

**Other issues:**

# 6 Evaluation

## 6.1 Micro-benchmarks

### 6.1.1 Bottleneck Inference Algorithms

Use ns-2/3 simulation to show the inference accuracy

### 6.1.2 Backend Scalability

Use EC2 implementation to test its scalability and process latency.

## 6.2 Real-trace Evaluation

### 6.2.1 Coverage

### 6.2.2 Accuracy

# 7 Applications

## 7.1 Path Selection

Compare with static/random selection in terms of bandwidth as well as standard video quality metrics (buffering ratio, join time, etc).

## 7.2 Edge Server/Cache Selection

Compare with static/random selection in terms of bandwidth as well as standard video quality metrics (buffering ratio, join time, etc)

## 7.3 Peer Selection

This represents an application beyond video. Compare with static/random selection in terms of bandwidth.

# References