

# Using Video-Based Measurements to Generate a Real-Time Network Traffic Map

Yi Sun<sup>⊗</sup>, Junchen Jiang<sup>†</sup>, Vyas Sekar<sup>†</sup>, Hui Zhang<sup>†</sup>, Fuyuan Lin<sup>⊗</sup>, Nanshu Wang<sup>⊗</sup>

<sup>⊗</sup> Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

<sup>†</sup> Carnegie Mellon University, Pittsburgh, PA, USA

{sunyi, linfuyuan, wangnanshu}@ict.ac.cn,

{junchenj, hzhang}@cs.cmu.edu, vsekar@andrew.cmu.edu

## ABSTRACT

We envision a real-time network traffic map for the Internet, where each network link is annotated with its capacity and its current utilization, with an interface that networked applications can query to inform their control decisions. While this goal is simple to state, it has been out of our reach due to concerns over measurement overhead and coverage. Our insight is that the rise of Internet video and the availability of measurements from video players present an unprecedented opportunity to address these issues. We outline a preliminary roadmap to build on this opportunity to realize a global traffic map.

## Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed systems—*Distributed applications*

## General Terms

Performance, Measurement

## 1 Introduction

Many Internet applications can benefit from a service that provides a real-time *traffic map* of the Internet [8, 19]. For instance, CDNs could improve server selection and peer-to-peer applications can choose peers more intelligently. Similarly, websites can be optimized to customize content for their clients based on the network state. Such a service may also be useful for network diagnosis and troubleshooting. In the absence of such a service, each application today deploys custom home-grown solutions or operate “in the dark” via trial-and-error solutions.

While this high-level vision of a *global traffic map* is not new, this goal has proved remarkably elusive. The main challenge that prior efforts have faced can be captured along three key dimensions:

- *Coverage*: Obtaining a global view of the network necessarily entails deploying many millions of vantage points running some measurement logic to obtain path-level metrics of interest. While “crowdsourcing” such measurements via P2P applications is promising [7, 4, 21], even the largest deployed efforts have limited visibility.
- *Overhead*: While reachability or latency are easy to measure, metrics such as available bandwidth or capacity or the location of bottlenecks have traditionally required algorithms with non-trivial overhead (e.g., few 100s of KBs per run) per path [24].
- *Real-time views*: To reflect current conditions, this traffic map also needs to be updated in near real-time. This raises further concerns in conjunction with the above concerns; i.e., we need millions of vantage points continuously running non-trivial measurements all the time.

Thus, this vision of a global traffic map has been out of our reach. In this context, we observe that the growing volume of Internet video traffic [1] and the ability to instrument video players to measure performance of video sessions in near real-time [17, 11] offer an unprecedented opportunity to address the above challenges. Specifically, we have (perhaps for the first time) the capability to obtain real-time measurements of the network state from millions of vantage points without any additional probing overhead.

This paper presents an preliminary roadmap to leverage this opportunity of using video traffic as the “carrier signal” to generate a real-time traffic map. We envision one or more providers in the video ecosystem who can offer such a service. For instance, many content providers (e.g., Google or Netflix or PPLive), CDNs (e.g., Akamai, ChinaCache), and third-party analytics providers (e.g., Conviva, Ooyala) already collect such measurements and are in a good position to build (and monetize) this service.

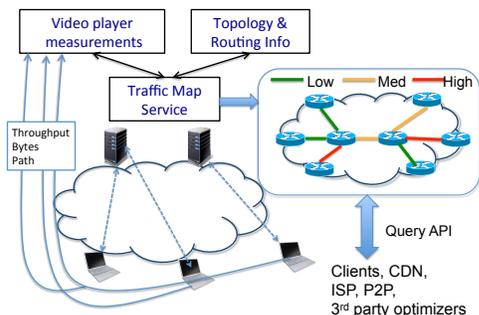
Specifically, our traffic map service annotates each link in the network<sup>1</sup> with its *capacity* and its *current utilization*. A networked application can query the service to obtain the state of links pertinent to its users’ performance. Even though

<sup>1</sup>The granularity of a link depends on the topology view; e.g., Layer 2 vs Layer 3, router vs. PoP. We currently use a PoP-level view as it offers more robust topology measurements. Our vision and algorithms, however, are quite general.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Hotnets '14*, October 27–28, 2014, Los Angeles, CA, USA.

Copyright 2014 ACM 978-1-4503-3256-9 ...\$15.00.



**Figure 1: Overview of the ICTM service**

video measurements are an enabler for our vision, there are two key challenges that remain. First, video measurements provide *coarse-grained* estimates of end-to-end throughput and do not directly provide the available bandwidth or utilization per-link. Second, we do not know the *background* traffic (i.e., other unobserved video and non-video traffic).

We present an initial attempt to address these challenges by casting them as network tomography problems [6]. For *capacity inference*, we leverage the history of measurements to compensate for the lack of a fine-grained view. Furthermore, we can use side information such as the ratio of measurement to background traffic; e.g., Netflix is typically 20-30% of peak traffic in US [2]. We can also exploit typical gravity-based traffic structures [26]. Third, to compute the current *utilization* of links we infer the background traffic on each link by approximating the traffic flows in the network as a fair sharing solution. Then, we use a maximum likelihood estimate of the background flows that best explains our current observations.

We evaluate our capacity and utilization inference algorithms on realistic network topologies with synthetic background and measurement workloads. Our initial results are promising. Even with our initial algorithms we are able to predict the capacity with 80% accuracy when the measurement traffic is roughly 20-30% of the total traffic. Furthermore, we observe a positive influence of having more data, wherein adding more historical measurements can substantially boost the capacity inference. We also observe that our background inference algorithm is very accurate.

While these early results are promising, we acknowledge several limitations that need to be addressed before our vision becomes practical. First, we need to understand how critical the different “side” information factors are and how accurate these need to be. Second, our current background inference is slow as it relies on a simple space search algorithm and we need faster algorithms to make ICTM a near real-time service. Finally, we need to validate our ideas using real data feeds from popular video providers.

## 2 ICTM Overview

Our vision is an ICTM (Internet Capacity and Traffic Map) service shown in Figure 1 that applications can query using a public API to obtain a near real-time estimate of different aspects of the network status. Applications may request the

link capacity, the traffic and utilization of the link, and the available bandwidth between pairs of endpoints.

The figure also shows the key inputs into the system. The first input is a set of video-based measurement feeds, where each video client reports throughput and total number of bytes downloaded in an epoch from a specific video server. We assume access to a route measurement platform such as iPlane [19].<sup>2</sup> ICTM does not need any additional active measurements beyond these data feeds. We use the video measurements and the routing data to annotate each PoP-level link with its *link capacity* and *current utilization*.

Many video providers, CDNs, and third-party providers already collect large volumes of such video measurements for their day-to-day operations [11, 17], and this is not a significant additional burden. The coverage of the ICTM service operated by different parties may vary; e.g., CDNs have the view of all traffic from its own servers, while a content provider can offer views across multiple CDNs. As such our focus in this paper is on establishing the viability of such an ICTM service irrespective of the choice of the provider and economic/monetization issues.

### 2.1 Feasibility Study

The key advantage of Internet video is the unprecedented coverage, both across space and time. Here, we present measurements on the *coverage* observed by two independent providers with different vantage points: a third-party video analytics provider serving clients in the US (TPA) and a large video provider in mainland China (ChinaVid).<sup>3</sup> TPA runs a measurement plug-in that runs on many affiliate content providers, while ChinaVid routinely collects throughput and video quality measurements of its clients.

**ISP/AS coverage:** First, we analyze the temporal coverage in terms of the client-side AS and ISP.<sup>4</sup> Note that a single ISP such as AT&T or Comcast may span multiple AS numbers. For example, for AS (similarly for ISP), we divide the one-day period into 1, 10, and 60 minute epochs. Then, for each AS (ISP) that appears at least once in the dataset, we calculate the fraction of epochs in which the AS (ISP) has at least 50 near-concurrent sessions. We show inverse CDF of the coverage in Figure 2. Across both datasets,  $\geq 50\%$  of the ISPs can be measured with very fine time granularity (1 minute), while a substantial fraction of ASes ( $\geq 20\%$ ) can be measured with very fine time granularity (1 minute) as well. We also see that there is a significant increase using 10-minute vs. 1-hour window, which shows an explicit trade-off between coverage and accuracy.

**Client/server-side coverage:** Next, we examine the client/server-side coverage in terms of client/server-side /24 IP prefixes.

<sup>2</sup>The design of this route measurement platform and how it handles issues like router aliasing etc., are outside the scope of this paper.

<sup>3</sup>We do not reveal the names for reasons of anonymity.

<sup>4</sup>Unfortunately, we do not have AS-path measurements concurrent with the video players and thus we cannot measure coverage for transit providers.

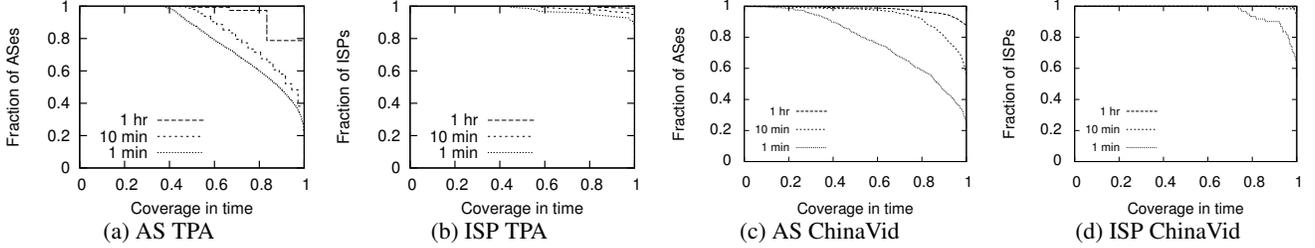


Figure 2: Inverse CDF of ISP/AS coverage.

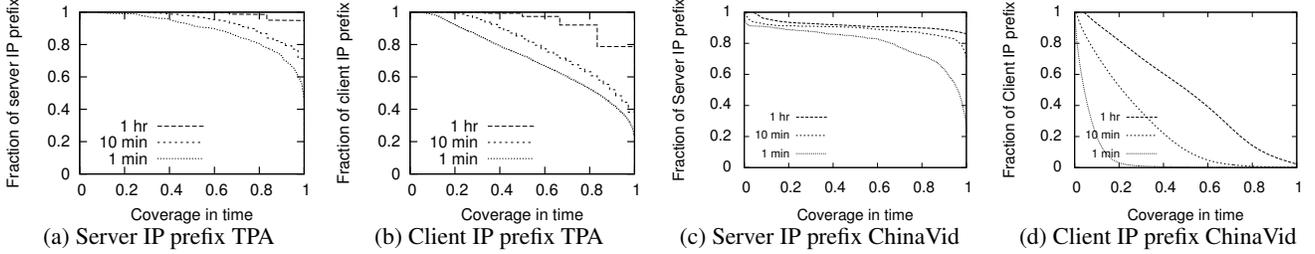


Figure 3: Inverse CDF of server/client IP prefix coverage.

Similar to Figure 2, Figure 3 shows the inverse CDF of fraction of epochs in which each client or server IP prefix (/24) has sufficient samples. We see that  $\geq 40\%$  of US server prefixes,  $\geq 20\%$  of China server IP prefixes, and  $\geq 20\%$  of US client IP prefix have enough sessions in every 1-minute windows. (The coverage of China client IP prefixes is low due to the extremely skewed distribution of the IP addresses within each IP prefix). Again we see a significant increase by using larger windows.

## 2.2 Problem Definition

Next, we formally define the inference problem that the ICTM service needs to solve.

We divide time into discrete epochs with a size  $|e|$  on the order of tens of minutes (as suggested by our coverage measurements). The subscript  $e$  refers to a specific discrete epoch in which our measurements are made.<sup>5</sup> In each epoch, we have client-side video measurements between different source and destination pairs. Each source and destination are mapped to a corresponding “node” in the network topology map; e.g., an ingress/egress PoP. Let  $M_{m,s,d,e}$  denote a measurement between source  $s$  and destination  $d$  in epoch  $e$ . (There may be many measurements during a given epoch for the same src-dst pair.) Each measurement  $m$  has two key attributes:  $B_{m,s,d,e}$  denotes the total bytes downloaded and  $T_{m,s,d,e}$  denotes the average throughput during this epoch. Let  $\mathcal{P}_{s,d,e}$  be the route used for traffic between  $s$  and  $d$  during the epoch  $e$  (obtained from the route measurement engine). Each path consists of multiple links;  $l \in \mathcal{P}$  denotes that this link lies on this path.

There are two quantities we want to infer: (1) the capacity of each link  $C_l$  and (2) the background load of each link in a given epoch  $bg_{l,e}$ . Formally, the ICTM problem is to estimate the values of the  $C_l$  and  $bg_{l,e}$  given the  $M_{m,s,d,e}$

<sup>5</sup>We expect this time granularity is sufficient for many application-layer control decisions [18].

and  $\mathcal{P}_{s,d,e}$  as inputs. Note that while the capacity does not change across time on the scale of days or weeks, the background load is much more variable. At a high-level, this is a *tomographic inference* problem, where we are inferring hidden variables based on some observed measurements. The key difference is that classical network tomography focuses on delay, loss etc., whereas our focus is on capacity and link utilization.

There are other ways to formulate an ICTM service that we do not consider here. For example, the service could report the available bandwidth for a given source-destination IP pair, without giving the link capacity or link utilization. However, we choose a more extreme formulation to infer per link capacity and utilization, because we can reconstruct these other end-to-end measures with per-link estimates.

## 2.3 Challenges

**Coarse measurement:** With video-based measurements, the measured bandwidth between two hosts is more coarse-grained, because the video players run at the application-layer within “sandboxes” that do not have access to packet-level statistics. Thus, the throughput will be measured over a coarser time duration than packet dispersion techniques (e.g., Spruce [24], IGI [15]). Consequently, unlike these active measurement approaches, we cannot infer the amount of *hidden* background traffic.

**Scale and responsiveness:** Typical active measurement platforms handle updates from a few (hundreds of) vantage points. In contrast, ICTM has to handle measurements from millions of video viewers, maintain the most up-to-date information of paths and links, and process a large number of queries in near real-time.

## 3 Link Capacity Inference

In this section, we sketch an initial algorithm for inferring the link capacities  $C_l$  given historical measurements  $M_{m,s,d,e}$

$$\begin{aligned}
\text{Minimize : } & \sum_l Cost_l & (1) \\
\forall l, e : & \sum_{s,d,l \in \mathcal{P}_{s,d}} \frac{B_{m,s,d,e}}{|e|} + \sum_{s,d,l \in \mathcal{P}_{s,d}} bg_{s,d,e} \leq C_l & (2) \\
\forall l : & C_l \geq T_{m,s,d,e}, \text{ if } l \in \mathcal{P}_{s,d} & (3) \\
\forall l : & C_l = \sum_{c \in CapVals} (d_{l,c} \times Cap_c) & (4) \\
\forall l : & Cost_l = \sum_{c \in CapVals} (d_{l,c} \times Cost_c) & (5) \\
\forall l : & \sum_{c \in CapVals} d_{l,c} = 1 & (6) \\
\forall l, c : & d_{l,c} \in \{0, 1\} & (7)
\end{aligned}$$

**Figure 4: Initial and underconstrained ILP formulation for inferring capacities**

and the paths  $\mathcal{P}_{s,d,e}$  over a number of epochs. We begin with simple solutions and progressively build insights toward the final solution.

A simple strawman solution is to map each measurement to its path and set the capacity to be the maximum throughput observed over the measurement history. That is,

$$\forall l : C_l = \max_{e,s,d, s.t. l \in \mathcal{P}_{s,d,e}} T_{m,s,d,e}$$

While easy to compute, there are several obvious problems with this strawman. First, it does not account for the background traffic and may significantly underestimate link capacities. Second, by viewing each measurement independently, it fails to utilize the confluence of multiple flows on a link within the same epoch. Finally, it ignores other practical aspects; e.g., links are likely drawn from some discrete set of values (e.g., 1Gbps or 10Gbps) and links are provisioned with some expected demand and cost budgets.

We can address these by introducing other hidden variables capturing the rate of background traffic between each src-dst pair in the historical epochs:  $bg_{s,d,e}$  and solving the optimization problem shown in Figure 4. This optimization formulation makes a few key improvements over the above naive solution. First, Eq (2) considers the total bytes across all measurement flows on a link during a given epoch. (The  $|e|$  factor is to normalize the bytes by the epoch duration to make it a rate measure.) Second, it also incorporates the new background variables. Third, it captures the observation that capacities have to be drawn from a discrete set of values  $CapVals$  in Eq (4) by introducing binary variables  $d_{l,c}$  indicating whether the link  $l$  has the discrete capacity  $Cap_c$ . (Note that at most one of these indicator variables can be 1.) Finally, it models capacity inference as a cost minimization problem based on the intuition that operators have likely chosen a low cost provisioning solution in expectation of observed demands.

However, this problem is *underconstrained* similar to many other network tomography formulations (e.g., [26]). That is, these equations do not impose sufficient structure on the capacity values. To address this issue, network tomography

approaches typically rely on out-of-band (side) information to impose more structure. In our context, we identify some natural candidates to serve this role:

- *Gravity models*: The traffic volumes between a  $s, d$  pair is roughly proportional to the products of the population (or total traffic volume) originating in  $s$  and  $d$  [26].
- *Measurement vs. Background ratio*: We may additionally know that our measurement traffic is roughly  $\beta$  of the total traffic; e.g., reports suggest YouTube is  $\approx 18\%$  and Netflix is  $\approx 30\%$  of peak traffic [2].
- *Overprovisioning*: Finally, we can use knowledge that the network core links are typically overprovisioned to run at  $\gamma=30-40\%$  link utilization on average.

Thus, we update our optimization function to incorporate this side information and attempt to minimize the deviation of our estimates from these additional factors:<sup>6</sup>

$$\sum_l Cost_l + \omega_1 \times GravDev + \omega_2 \times BGRatioDev + \omega_3 \times OverProvDev$$

Specifically, the new terms in the objective to capture these deviations are defined as follows:

$$BGRatioDev = (totmeas - \beta \times totbg)^2$$

$$OverProvDev = \sum_l \sum_e (totmeas_{l,e} + totbg_{l,e} - \gamma \times C_l \times |e|)^2$$

$$GravDev = \sum_{s,d,e} (bg_{s,d,e} \times |e| - \mathcal{G}_{s,d} \times totbg_e)^2$$

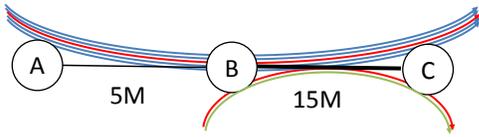
Here,  $totmeas$  and  $totbg$  denote the total volume of the measurement and (unobserved) background traffic over the measurement period. Similarly  $totmeas_{l,e}$  and  $totbg_{l,e}$  denote the volume of measurement and background traffic on link  $l$  for the epoch  $e$ .  $\mathcal{G}_{s,d}$  represents the gravity coefficient, where we expect the total volume between a given  $s, d$  to be  $\mathcal{G}_{s,d} \times totbg_e$ , where  $totbg_e$  is the total background traffic in the epoch.

We can solve this optimization problem using many historical measurements to obtain the  $C_l$  values. This optimization is, however, non-trivial as it has both discrete (binary) variables and quadratic terms in the objective. We also need suitable weights  $\omega_1-\omega_3$  to determine the importance of the different side information factors. As a simple starting point, we solve a practical relaxation of this optimization where we treat the  $d_{l,c}$  as fractional values in the range  $[0, 1]$ . Then, we simply “round up” the obtained capacity values to the nearest higher value in  $CapVals$ . In Section 5 we show that even this simplified solution can yield accurate results for capacity inference. We plan to investigate exact optimizations as the capacity inference is an offline step.

## 4 Estimating Current Utilization

In this section, we move to the second problem of inferring the  $bg_{l,e}$  for each  $l$  at a given  $e$ .

<sup>6</sup>The choice of adding these as constraints vs. terms in the objective is somewhat arbitrary. We add them to the objective because we may not know the exact values.



**Figure 5: An example for background traffic inference. There are 5 flows from A to C and 2 flows from B to C, and 2 of them in red are selected as measurement traffic.**

At first glance, this might seem a subproblem we solved in the previous step. However, there are two main differences. First, while capacity inference can rely on historical measurements, link utilization needs to be more real-time and based on the current epoch  $e$ . Second, the side information we used such as gravity model or the ratio of background-to-measurement traffic are aggregate effects that only manifest over long timescales but not per-epoch.

This suggests we need a different approach. Fortunately, since we have already run the capacity inference step, we can additionally use the  $C_l$  values. Thus, we reformulate our goal to infer  $bg_{l,e}$  given  $M_{m,s,d,e}$ ,  $\mathcal{P}_{s,d,e}$ , and  $C_l$ .

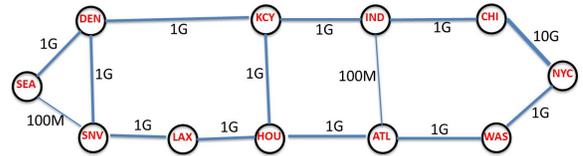
We make two simplifying assumptions here. First, we assume that the background rate is relatively constant during this epoch. Second, we assume that all traffic runs TCP or TCP-friendly algorithms. If all flows are running TCP, then they will saturate the network with a rate allocation that is approximately close to a max-min fair allocation [20].

Building on these insights, we sketch the following inference algorithm. Consider one specific possible value of the background load  $bg_{s,d,e}$  for the current epoch. Now, given this value, and the inferred capacities  $\{C_l\}$ , we can compute what the remaining traffic could have achieved via network-wide max-min fairness allocations. Let this predicted value be  $T_{s,d,e}^{Est}(bg_{s,d,e})$ . Since we know the actual observed values  $B_{m,s,d,e}$  and  $T_{m,s,d,e}$ , we can pick the ‘‘maximum likelihood’’ estimate:

$$\arg \min_{bg_{s,d,e}} |T_{s,d,e}^{Est}(bg_{s,d,e}) - T_{s,d,e}|$$

We can use some suitable summary for  $T_{s,d,e}$  using the  $T_{m,s,d,e}$  measurements. Depending on the choice, we might get lower or upper bounds. For instance, if we pick the min observed throughput for each  $s, d$  during this epoch  $e$  and use that to estimate the background loads, we get a conservative overestimate of the background traffic rates. We currently pick the median value in our experiments.

Figure 5 shows an illustrative example. With max-min fair allocation, the throughput of each A-C flow is 1Mbps, and the B-C flow is 5Mbps. Suppose our measurement set consists of 1 A-C and 1 B-C flow. Then, we enumerate the search space of the possible link utilizations. Consider one of these points where the background traffic occupies 90% of the capacity on both links. Then, it leaves 10% of the bandwidth (that is 0.5Mbps on  $A - B$  and 1.5 Mbps on  $B - C$ ) for our measurement flows. According to max-min fair sharing, these two flows should have throughputs of 0.5Mbps and 1Mbps respectively, clearly deviating from the observation. However, when we consider setting background traffic to occupy 80% of the bandwidth on A-B and 60% of



**Figure 6: Topology used in simulation.**

the bandwidth on B-C, the throughputs of the two measurement flows generated by max-min fair share model match the measurements. Therefore, we can infer the background traffic occupies  $5 * 80\% = 4\text{Mbps}$  on A-B and  $15 * 60\% = 9\text{Mbps}$  on B-C.

There are two other issues here. First, there might be multiple values of the background loads  $bg_{s,d,e}$  that generate the same estimation error. In that case, we select the one which best matches the side information of measurement vs. background ratio. Second, we need fast algorithms for solving this search problem. We currently use a simple but expensive greedy algorithm for searching the space.

## 5 Preliminary Results

**Setup:** Since we do not have public information on background loads or link capacities, we rely on simulations to validate our algorithms. We implemented a flow-level simulator [23] that takes as input a traffic matrix, topology and routing information, flow size and arrival patterns, link capacities, and determines the throughput of each flow.

For the following results, we used a PoP-level topology. We generated synthetic traffic patterns where the traffic volume between a pair of PoPs is proportional to the product of the city populations. We assume a Poisson arrival distribution with fixed flow sizes of 2 epochs. (This is not fundamental and was just for convenience to complete the simulation model.) We choose the link capacities from a discrete set of values  $\{100\text{Mbps}, 1\text{Gbps}, 10\text{Gbps}\}$ , with the expected utilization around 80%. Figure 6 shows the ground truth topology and link capacities.

We randomly selected some of the flows as *measurement* traffic and use their observed throughput and byte counts as the input to our algorithms. The key metric of interest is the *estimation accuracy* for each link. Since the capacities are discrete values, we report the fraction of links for which the inference is correct. For the background ratio, we compute the accuracy metric as  $1 - \frac{|inference_l - groundtruth_l|}{groundtruth_l}$  and report the network-wide average. We vary the number of epochs and fraction of measured flows.

As a simplifying assumption, we use accurate estimates of the side information values – the gravity vector  $\mathcal{G}_{s,d}$ , overprovisioning factor  $\gamma$ , and the background-to-measurement traffic ratio  $\beta$ . We set the values of  $\omega_1 = \omega_2 = \omega_3 = 1$  and leave better choices of these weights for future work.

**Capacity Inference:** Figure 7 shows that the fraction of links with perfect capacity inference improves from  $< 20\%$  when 10% of the traffic serves as measurements to  $> 80\%$  when the ratio increases to 30%. The result also shows the

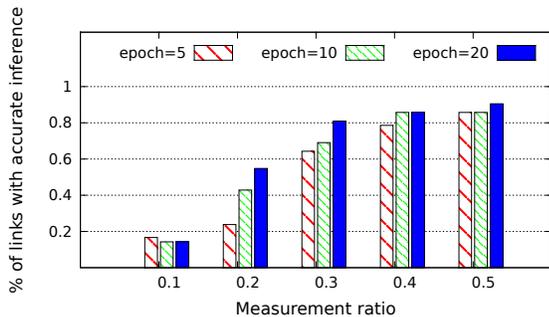


Figure 7: Accuracy of link capacity inference

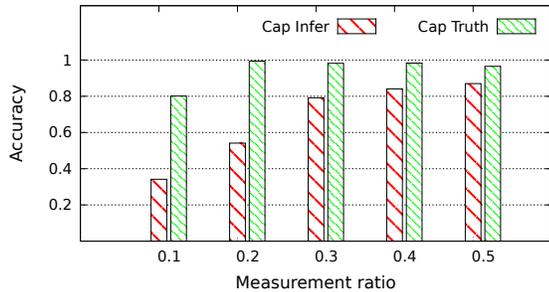


Figure 8: Accuracy of background traffic inference

improvement as we have more historical information by increasing the number of measurement epochs. For instance, keeping the measurement ratio fixed at 0.2 we only infer the capacities accurately for 23% of the links with 5 epochs. However, when we use 20 epochs of information the accuracy is significantly improved to about 60%.

**Utilization inference:** Figure 8 shows accuracy of estimating the link utilization as the measurement traffic ratio increases. We consider two cases with perfect capacity inference and the actual capacity inferred using 20 epochs of measurements. Similar to capacity inference, the accuracy increases with more measurement traffic, e.g. the accuracy is 60% when 30% of the traffic can be measured. One reason for the poor performance when the measurement only occupies 10% or 20% of the total traffic is that the inaccuracy of the link capacity inference impacts the background inference (see Figure 7). With incorrect capacities as the input, our background traffic inference also suffers. However, if the capacity is precisely inferred, the accuracy of the background inference is  $\geq 80\%$  even if the measurement traffic ratio is only 10%. This suggests that additional information from ISPs (e.g., link capacities) can improve our inferences.

## 6 Related Work

**Network Tomography:** Network tomography [6] infers a network’s characteristics using information derived from indirect data sources; e.g., to infer topology [13, 9], traffic matrix [26], latency [25], and packet loss [10, 12, 22]. While we are inspired by this line of research, we focus on different and arguably more challenging inference tasks.

**Crowdsourcing:** Recent works rely on network-intensive

P2P applications running on end systems to estimate and infer network performance [5, 7, 4, 3]. Our work follows in this spirit, but uses video as the data source. Given current trends, video has a much better coverage than that of P2P.

**End-to-end tools:** End-to-end tools such as Pathneck [14], IGI/PTR [15], BRoute [16] are used to measure the network bandwidth and locate the bottleneck link on the path. The main disadvantages of these tools are the high overhead and the need for dedicated vantage points.

## 7 Discussion and Limitations

**Measurement biases:** Because video clients use CDN servers, there might be some selection bias toward picking local servers leading to blind spots into edge links between clients or deep into the network core. That said, this client-CDN segment is arguably the “high impact” part of the network that carries most traffic and ICTM naturally sheds light on this critical segment of the global network.

**Additional instrumentation and information:** In this work we took an explicit stance to rely on the existing measurements made by video clients. This choice was pragmatic as the browser or player-based sandbox limits the types of measurements that can be run. One open question is what additional measurements can boost our accuracy.

**Other “carriers”:** It is conceivable that other application-level carriers (e.g., large web or ad providers [22]) could also use our algorithms to create similar traffic maps. However, we believe that video is uniquely positioned because it offers a continuous view of bandwidth conditions as it is inherently using persistent transfers, unlike pure web traffic.

**Cross-provider collaboration:** Our current work assumes a single provider with access to video measurements. An interesting direction for future work is to consider a semi-federated approach where different video application providers can collaboratively refine individual traffic maps to complement each other’s blind spots. Additionally, some ISPs may provide some capacity estimates to help validate the inference results.

**Sensitivity:** There are several assumptions in our inference mechanisms; e.g. gravity model, estimates on measured traffic vs. background, and degree of over-provisioning. As future work, we need to evaluate the sensitivity of the inference to the accuracy of such side information to better understand how critical they are to the ICTM’s success.

**Real-world validation:** Finally, the validation in this paper was based on simulation with synthetic traffic patterns. We are now collecting the real dataset of both the network topology (by trace-route) and video-viewing with the help of a leading Chinese video service provider. We will use these real data to verify our approaches in the near future.

## Acknowledgements

This work is supported in part by the National Basic Research Program (2012CB315802) and the Natural Science Foundation of China (61379133). This work was also funded in part by NSF under award number CNS-1345305.

## 8 References

- [1] Cisco Visual Networking Index. <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>.
- [2] Sandvine Global Internet Phenomena. <https://www.sandvine.com/trends/global-internet-phenomena/>.
- [3] Z. S. Bischof, J. S. Otto, and F. E. Bustamante. Distributed systems and natural disasters: Bittorrent as a global witness. In *Proc. Special Workshop on Internet and Disasters*, 2011.
- [4] Z. S. Bischof, J. S. Otto, and F. E. Bustamante. Up, down and around the stack: Isp characterization from network intensive applications. *ACM SIGCOMM Computer Communication Review*, 42(4):515–520, 2012.
- [5] Z. S. Bischof, J. S. Otto, M. A. Sánchez, J. P. Rula, D. R. Choffnes, and F. E. Bustamante. Crowdsourcing isp characterization to the network edge. In *Proc. ACM SIGCOMM workshop on Measurements up the stack*, 2011.
- [6] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: recent developments. *Statistical science*, pages 499–517, 2004.
- [7] D. R. Choffnes, F. E. Bustamante, and Z. Ge. Crowdsourcing service-level network event monitoring. *ACM SIGCOMM Computer Communication Review*, 41(4):387–398, 2011.
- [8] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A knowledge plane for the internet. In *Proc. SIGCOMM*, 2003.
- [9] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang. Maximum likelihood network topology identification from edge-based unicast measurements. In *Proc. ACM SIGMETRICS*, 2002.
- [10] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *Proc. ITC Conf. IP Traffic, Modeling and Management*, pages 28–1, 2000.
- [11] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. A. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *Proc. SIGCOMM*, 2011.
- [12] N. G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. Inferring link loss using striped unicast probes. In *Proc. IEEE INFOCOM*, 2001.
- [13] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak. Toward the practical use of network tomography for internet topology discovery. In *Proc. IEEE INFOCOM*, 2010.
- [14] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang. Locating internet bottlenecks: Algorithms, measurements, and implications. In *Proc. SIGCOMM*, 2004.
- [15] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications*, 21(6):879–894, 2003.
- [16] N. Hu and P. Steenkiste. Exploiting internet route sharing for large scale available bandwidth estimation. In *Proc. IMC*, 2005.
- [17] S. Krishnan and R. Sitaraman. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. 2012.
- [18] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang. A case for a coordinated internet video control plane. In *to appear in SIGCOMM*, 2012.
- [19] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. In *Proc. OSDI*, 2006.
- [20] D. Nace and M. Pioro. Max-Min Fairness and Its Applications to Routing and Load-Balancing in Communication Networks: A Tutorial. *IEEE Communications Surveys & Tutorials*, 10(4), 2008.
- [21] J. S. Otto, M. A. Sánchez, D. R. Choffnes, F. E. Bustamante, and G. Siganos. On blind mice and the elephant. In *Proc. of ACM SIGCOMM*, 2011.
- [22] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Server-based inference of internet link lossiness. In *Proc. IEEE INFOCOM*, 2003.
- [23] J. Sommers, R. Bowden, B. Eriksson, P. Barford, M. Roughan, and N. Duffield. Efficient Network-wide Flow Record Generation. In *Proc. INFOCOM*, 2011.
- [24] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proc. IMC*, 2003.
- [25] Y. Tsang, M. Coates, and R. D. Nowak. Network delay tomography. *IEEE Transactions on Signal Processing*, 51(8):2125–2136, 2003.
- [26] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads. In *Proc. ACM SIGMETRICS*, 2003.