Gauging the network friendliness of P2P applications

Dario Rossi, Elisa Sottile, Silvio Valenti, Paolo Veglia
TELECOM ParisTech, Paris, France
firstname.lastname@enst.fr

ABSTRACT

This demo presents a software tool whose aim is to quantify the level of network awareness and friendliness of the currently deployed Internet P2P applications. Considering such systems as black-boxes, we passively measure the traffic they generate and we gather, by means of both passive inference and active-probing techniques, some quantities relevant for their description.

Observed metrics include, e.g., the amount of traffic that the peer under observation exchanges with peers belonging to the same Autonomous System or geographical Country. Similarly, we inspect the traffic exchanged in terms of its proximity in the underlying IP networks, expressed e.g., as RTT delay or number of IP hops.

A number of statistics are then computed, in real-time, over the peer database inferred from measurement (e.g., ranging from empirical probability mass function, to correlation among the measured parameters, etc.). A flexible user interface displays a variety of data representations such as pmf, scatter plots, geographical maps and exploits Kiviat charts to represent, at a glance, the full range of measured values in a compact and visually-intuitive way.

1. INTRODUCTION

Peer-to-peer software now spans a rather large offer of services, and many valuable works already exist which focus on the measurement and analysis of live-TV streaming [2–4], P2P applications. As far as methodology is concerned, the above works can be roughly divided into two classes. The first approach is to use active crawlers, which allow to gather very detailed information from the whole network. This is however a daunting task especially for proprietary systems, since a partial reverse engineering of the application is required. A second set of works adopts a black-box approach, measuring and analyzing the traffic generated by the application: this ensures that the approach is widely applicable, at the expense however of the level of details of the information (e.g., overlay topology cannot be gathered without crawling).

Our work fits in the latter class, focusing on the analysis of P2P application friendliness and awareness toward the underlying network, as [3, 4]. From a high level perspective, we can define two categories of metrics able to express network awareness: *path-wise metrics* (such as RTT delay, HOP count, bottleneck bandwidth) are determined by the conditions on the path between two peers in the overlay. Conversely, *peer-wise metrics* (such as Autonomous Systems, geographical location, /16 IP prefix, access capacity, etc.) only depend on properties of a single peer.

Path-wise metrics are explored in [3], by means of an *active testbed* where authors enforce artificial bandwidth limitations, packet loss and delay, and examine P2P-TV reaction to adverse network conditions. We instead explored peer-wise metrics in [4], by adopting a purely *passive approach*: by inferring from measurement the main properties of content exchange, we assess which parameters mostly influence the download preference of P2P-TV application.

The proposed tool merges both [3, 4] approaches, exploiting passive as well as active techniques to gather full-relief results.

2. DEMO SOFTWARE

We note that UDP is becoming the largely preferred transport layer protocol by P2P applications[1], to which we restrict our attention in the following. Notice that applications typically run on a (random) UDP port, over which they multiplex all incoming and outgoing signaling and data traffic. During the demonstration a probe machine P runs one or more P2P applications, whose traffic is sniffed by the demo software running on the analyzer machine A. The software lets the user select an application on P, identified by a specific $(IP, port)$ pair: once an endpoint has been selected, the software starts analyzing the P2P traffic.

As previously mentioned, the software exploits a mix of active and passive methodology to gather path-wise and peer-wise information respectively. Prior to overviewing the metrics and their representation, let us stress an important implication of this choice. As far as passive methodology is concerned, our software tool gathers the properties of contacted peers either from a local database [7] (e.g., geolocalization and AS number) or through inference and analysis (e.g., throughput, hop-count). The analysis does not interfere with the observed P2P application traffic, but we are rather limited by database access speed. Since the DB allows us more than 40,000 queries per second, this does not constitute a bottleneck.

Conversely, the tool performs active measurements to gather *path-wise* properties, thus possibly interfering with the observed P2P traffic: as such, active path-wise measurement should be limited as much as possible (notice that although measurements are performed by a machine collocated with the probe peer, they likely share the same access link). Consider for instance the issue of path capacity estimation: expensive active-path probing techniques (such as bandwidth measurement by means of packet trains) are not suitable for our purposes, and we rather need light-weight measurement technique (such as those based on packet-pair dispersion). In reason of this observation, we resort to the use of CapProbe [6] to actively estimate the bottleneck capacity, the RTT delay and the IP time-to-live (from which we can infer the path distance). For each peer, we perform $N = 100$ measurements by sending pairs of back-to-back ICMP packets (each packet pair is spaced by $\Delta T = 0.5$ seconds), upper-bounding the number of concurrently active path-probing processes at $C = 50$.

Although the amount of active-probing traffic is limited to $R =$

[1]We point out that file transfer in BitTorrent moved to uTP (a closed loop protocol, controlled at the application layer, working over UDP) since December 2008.

Table 1: Metric gauging methodology

Metric		Type	Method
AS	Autonomous System	Peer	Passive (DB)
CC	geographical Country	Peer	Passive (DB)
NET	IP address similarity	Peer	Passive
RTT	Round Trip Delay	Path	Active
CAP	Capacity	Path	Active
HOP	IP hop-count distance	Path	Active

$2C/\Delta T = 200$ packets per second, performing active experiments for the whole peer population may be a prohibitive task. To this extent, we point out that a large number of peers is only contacted once during the discovery phase, but is not contacted later on, thus is not involved in the exchange of content: while such peers may constitute a significant percentage of the peer population, they are nevertheless irrelevant as far as the traffic volume is concerned. We thus limit active measurements only to peers that actively contribute to the video stream; i.e. those who send at least two packets in a time window W. This simple heuristic still allows to focus on the bulk of the traffic volume, while reducing the amount of active probing traffic.

2.1 Metric Definition

Tab. 1 summarizes some of the metrics measured by the application, highlighting whether an active or passive methodology is used. Notice that in some cases, it may be possible to measure the same metric (e.g., IP TTL, RTT, etc.) with either methodology. Yet, notice that is difficult to infer RTT by passive measurement of UDP traffic, since reverse engineering is needed to match data packets with the corresponding application-layer acknowledgements. Shortly, peer-wise properties are either gathered through local database [7], or inferred by purely passive measurement, while path-wise properties are collected through active measurements.

For each of the above metric X, the demo software partitions the contributing peers set $\mathcal{P}$ in two disjoint groups $\mathcal{P} = \mathcal{P}_{close(X)} \cup \mathcal{P}_{far(X)}$, so that peers that are "closer" to the peer under observation are grouped altogether. For instance, in case of AS (CC) metric, the $\mathcal{P}_{close}$ set will be constituted by peers belonging to the same Autonomous System (Country). Netmask similarity will group together peers based on their IP/16 prefix, while for RTT metric, peers whose RTT will be smaller than the median RTT will fall into $\mathcal{P}_{close}$ (and similarly for HOP).

For each metric, we count the number of peers and bytes in either group [4]: e.g., we evaluate the percentage P_{CC} of peers that belong to the same country over the total number of contacted peers, and the percentage of bytes B_{CC} exchanged with them. Similarly, we evaluate how many bytes have been exchanged with peers having a RTT lower than the median RTT, etc. Traffic directionality is also taken into account, meaning that it is possible to either jointly or separately analyze the download/upload application behavior. As another mean to express network awareness, we evaluate the correlation $\rho(B, X)$ between the amount of bytes B exchanged and the values of metric X: as correlation does not imply causation, we just use it to *relatively weight* the importance of each metric.

2.2 Metric Representation

The computed statistics are shown, in real-time, with different types of representations (e.g., scatter plots, geographical maps, empirical probability distribution function, etc.). In order to show all of the relevant statistics at a glance, we also adopt a Kiviat [5] representation to report noteworthy characteristics of different classes of applications (e.g., Web, interactive, VoIP, etc.).

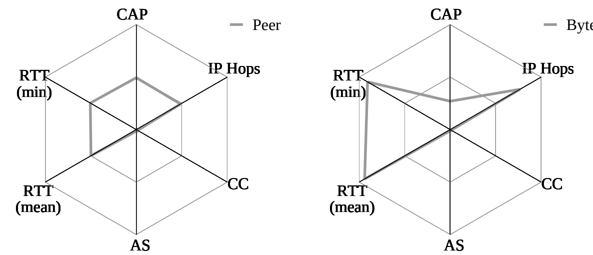

Figure 1: Example of network-awareness of a PPLive endpoint: percentage of close peers (left) and bytes (right) are reported by means of a Kiviat representation

For illustration purposes, Fig. 1 shows the network awareness as Kiviat charts for a PPLive endpoint, showing both the percentage of "close" peers (left) and bytes (right) computed over a 15 minutes experiment. A single curve is used to join the percentage of different metrics reported on the six radial axes. Consider the left peer plot first: since a median threshold is used for the capacity CAP, HOP count, minimum and median RTT, this means that exactly half of the contacted peers are "close" with respect to each of these metrics. Conversely, practically none of these close peers belong to the same AS or CC of the endpoint under observation.

Let us now consider the right byte plot: intuitively, the bigger the closed curve, the friendlier is the application toward the IP network. If we look at the percentage of bytes we see that, at least in this experiment, while the percentage of CC and AS bytes is negligible, a rather large fraction of bytes is exchanged with the half of peers that are close in terms of IP hops and RTT delay. Clearly, no definitive conclusions can be gathered from a single observation: many factors indeed affect the above metris (e.g., channel popularity, access type, time of day, etc.) so that results have to be averaged over a large number of observations prior that any general and relevant conclusions can be made. Yet, we point out this behavior to be rather unexpected, as [3] showed older versions of PPLive to be insensitive to peer RTT delay and IP hop count. This possibly suggests that newer versions may implement proximity techniques, and also confirms that characterization of P2P applications needs to be constantly updated.

3. REFERENCES

[1] Demo software, available online at `http://www.infres.enst.fr/~drossi/P2PGauge`

[2] X. Hei, C. Liang, J. Liang, Y. Liu, K.W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Transactions on Multimedia*, Vol.9, No.8, 2007

[3] E. Alessandria, M. Gallo, E. Leonardi, M. Mellia, M. Meo, "P2P-TV systems under adverse network conditions: a measurement study," *In IEEE INFOCOM'09*, Apr. 2009

[4] A. Horvath, M. Telek, D. Rossi, P. Veglia, D. Ciullo, M. A. Garcia, E. Leonardi and M. Mellia, "Network Awareness of P2P Live Streaming Applications," *In HOTP2P Workshop at IEEE IPDPS'09*, Rome, Italy, May 2009.

[5] K. Kolence, P. Kiviat, "Software Unit Profiles and Kiviat Figures," *ACM Performance Evaluation Review*, Vol. 2, No. 3, pp. 2–12, September 1973.

[6] R. Kapoor, L. J. Chen, Li Lao, M. Gerla, and M. Y. Sanadidi. "CapProbe: A Simple and Accurate Capacity Estimation Technique," *In ACM SIGCOMM'04*, Portland, USA, 2004

[7] MaxMind GeoLite, `http://www.maxmind.com/`