HAL
archives-ouvertes.fr

TopHat: supporting experiments through measurement infrastructure federation

Thomas Bourgeau, Jordan Augé, Timur Friedman

▶ To cite this version:

Thomas Bourgeau, Jordan Augé, Timur Friedman. TopHat: supporting experiments through measurement infrastructure federation. TridentCom 2010 - 6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, May 2010, Berlin, Germany. pp.542-557, 10.1007/978-3-642-17851-1_41 . hal-00724834

HAL Id: hal-00724834
https://hal.sorbonne-universite.fr/hal-00724834

Submitted on 22 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TopHat: supporting experiments through measurement infrastructure federation

Thomas Bourgeau, Jordan Augé, and Timur Friedman

LIP6 laboratory, UPMC Paris Universitas and CNRS

Abstract. Researchers use the PlanetLab testbed for its ability to host experimental applications in realistic conditions over the public best-effort internet. Such applications form overlays whose performance is affected by the underlying topology and its evolution. While several topology information services have been proposed for PlanetLab, the TopHat system that we describe here fills a special niche. It is designed to support the entire lifecycle of an experiment: from setup, through run time, to retrospective analysis. TopHat does so in a new way, by drawing upon excellent, proven third party services, notably the DIMES and ETOMIC measurement infrastructures, for specialized measurements. TopHat has been developed as the active measurement component of PlanetLab Europe, the flagship testbed of the OneLab experimental facility. It is part of OneLab's larger effort to pioneer the federation of previously independent testbeds and measurement systems in order to provide a diverse global scale environment for Future Internet research.

1 Introduction

PlanetLab nodes are fully open to the internet, and this allows experimenters to deploy applications such as novel overlays, peer-to-peer systems, content distribution networks, and the like. The collection of topology information is of particular interest to experimenters because the testbed consists only of the nodes at the edges of the network, not the underlying network. The ability to expose these applications to real-world network conditions is one of the prime motivating factors that leads experimenters to use PlanetLab, rather than a simulation or emulation environment. However, it also means that experimenters require information about the network topology in order to guide their experiments and make sense of their results.

There are many tools to measure the interesting properties of network paths. These properties range from the IP topology, which can be obtained thanks to the popular traceroute tool, to the available bandwidth between two nodes, for which there exists several possible tools. A review by the MOME project provides more details on available tools [1].

TopHat proposes an alternative to the deployment and use of such tools independently by each user by providing a topology monitoring service for applications running on the PlanetLab testbed. TopHat's originality in this regard lies in its support of the entire lifecycle of an experiment. During the setup phase,

it assists PlanetLab users in choosing the nodes on which the experiment will be deployed, allowing them to base decisions on measured characteristics of the network as seen from each node. At run time, it provides live information to support adaptive applications and experiment control, providing measurements via a simple query interface and through the use of callbacks. The measurement data collected by the system are archived, and are thus available for retrospective analysis of an experiment, as well as being available for the community at large. Sec. 3 of this paper gives further insight into how TopHat supports users.

Following this description of the service that TopHat provides, Sec. 4 gives an overview of the system's architecture. In particular, it presents the user interfaces, focusing on the web services API.

Another specificity of TopHat is that it draws upon third party services – notably the DIMES and ETOMIC measurement infrastructures – that have a proven track record of excellence in providing specialized measurements to the research community. TopHat tunnels information from these systems to its users transparently. This interconnection is an instance of the larger effort, pioneered by the OneLab experimental facility [2], to federate previously independent testbeds and measurement systems in order to provide a diverse global scale environment for Future Internet research. Sec. 5 presents the details.

TopHat gets its inspiration from a number of proposed and existing systems, which are described in Sec. 2, the related work section of this paper.

2 Related work

Network measurement systems draw on two principal sources of information to learn about the network topology: BGP feeds, which describe how the IP address space is divided into routable prefixes and which provide coarse-grained routes, at the autonomous system (AS) level, to reach those prefixes; and active measurements, starting with the well-known traceroute tool, which learns about routes at the IP interface level.

Infrastructures making BGP information publicly available include Route-Views [3], Team Cymru [4], and pWhoIs [5]. The Route Views project, headquartered at the University of Oregon, provides much of the data that researchers use to study BGP routing tables and their dynamics. Route Views servers get their information by peering directly with BGP routers, typically at large ISPs. Team Cymru is a not-for-profit network security firm that provides an IP to AS number mapping service based on information collected from a large number of BGP feeds (including Route Views). The corporately-run pWhoIs service is similar to that offered by Team Cymru, with the specificity that it offers geographical information about ASes and IPs. TopHat currently sources its IP to AS translations from Team Cymru.

Two notable active measurement infrastructures offered as a public service are Scriptroute and perfSONAR. Scriptroute [6], from the Universities of Maryland and Washington, consists of a set of ready-to-use tools deployed on PlanetLab nodes that a user can access through a simple scripting language.

It supports queries for traceroute information and measurements of delay and available bandwidth. As Scriptroute is accessible to any internet user, it places an emphasis on measurement safety. TopHat support a similar set of measurement types, but sources them from third party services if those services can offer superior measurements. Measurement safety in TopHat is achieved with the PlanetLab model [7] of restricting users to those whose institutions have committed to an acceptable use policy, and by ensuring traceability of each measurement back to its originator; in exchange, there are no hard-coded limits on what can be measured.

perfSonar [8], a product of the national research and education network (NRen) community, is deployed in the NRen networks and provides uniform access to measurements to users in multiple administrative entities. It serves as an example to TopHat of a system that interconnects measurement systems. Whereas perfSonar offers a uniform set of tools from a set of peer entities, TopHat federates heterogeneous systems. The focus of perfSonar is on troubleshooting across network boundaries; TopHat's is on experiment support. In addition to performing standard active measurements, perfSonar has direct access to router information such as queue lengths and packet drop rates. Because of this unique source of measurements, we consider it a prime candidate for future TopHat interconnection.

TopHat draws upon the Dimes [9] infrastructure for its large number of measurement vantage points. Dimes consists of thousands of software agents hosted by volunteers under the coordination of researchers at Tel-Aviv University. There are also agents on PlanetLab. The Dimes web service interface provides access to IP level traces and AS information. The Ono project [10] marshals a much larger number of agents: there have been more than 650,000 downloads of its plugin for the popular Vuze BitTorrent client. These agents conduct traceroutes between clients in order to support better peer selection. However, the sensitive nature of measurements conducted by individuals' peer-to-peer clients means that the Ono data would not be freely available to TopHat users.

Whenever possible, TopHat conducts delay and available bandwidth measurements from Etomic boxes. Etomic [11] is an infrastructure consisting of GPS-synchronized servers equipped with measurement cards that are capable of measuring delays to a precision of tens of nanoseconds. There are a few dozen Etomic boxes, many of them collocated with PlanetLab nodes. Since these boxes must be reserved, another platform has been deployed alongside Etomic to allow on-demand measurements. This platform is called Sonoma [12], and it offers medium-precision resolution (tens of microseconds) measurements through a webservice interface. TopHat interconnects with Sonoma as well. The Ripe TTM infrastructure [13] also provides GPS-synchronized measurement boxes. Interconnection with TTM would be of interest to TopHat because there are a few hundred of these, located primarily with network operators (including commercial operators), and providing regular measurements in a full mesh between boxes. As opposed to Etomic, though, TTM does not provide on-demand measurements or the same degree of precision.

In designing TopHat, we have been inspired by the Nakao et al. paper [14] that argues for the deployment of a topology information service within a testbed, aimed at providing users with a variety of measurements through a common API. The basic service can then be used to offer higher-level functionalities. One interest of such a service is that aggregation of requests allows for measurement reuse, thus reducing the strain on the network. Also, the user can benefit from best-of-breed tools and measurements, leaving him to focus on developing his overlay application. iPlane [15] is an infrastructure that implement this sort of service. Run by researchers at the University of Washington, iPlane provides overlays with predictions of network path characteristics such as delays, loss rates, and available bandwidth. The focus is on making predictions concerning paths between endpoints for which direct measurements are not possible or would be costly to obtain. It follows in the line of other predictive services such as IDMaps and Vivaldi (see the iPlane paper for further references). iPlane makes use of its own agents on PlanetLab nodes and within BitTorrent clients, as well as connecting to public traceroute servers. TopHat's federation with external measurement infrastructures can be seen as an extension of iPlane's drawing upon external traceroute servers. As a service for PlanetLab users, TopHat does not face the same necessity for measurement prediction as does a universal measurement service such as iPlane; in most cases, there are TopHat dedicated agents available to directly perform measurements from the PlanetLab nodes. TopHat could benefit, though, by adding predictions in circumstances where on-demand measurements are not possible.

TopHat also shares characteristics with Atmen [16]. Atmen reduces measurement overhead through reuse of measurements taken from similar vantage points or at points in time close to those that have been requested. The system supports a mechanism to trigger alarms (which in turn can start up active measurements) when it detects topological changes. TopHat's callback mechanism operates on a similar principle. Atmen is not available to the research community at large.

TopHat provides historical measurement data through the Network Measurement Virtual Observatory [17]. The best-used source of historical data comes from the CAIDA center. CAIDA's Archipelago, or Ark, measurement infrastructure [18] (the successor of the well-known Skitter), consists of a few dozen monitors worldwide. Ark aims for regular, comprehensive measurements to all /24 network prefixes, whereas TopHat provides measurements focused on testbed users' demands.

3 An infrastructure in support of testbed applications

This section presents the topology information services that TopHat offers to support PlanetLab applications, from setup through completion. It also discusses how usage monitoring in TopHat might help us gain a better understanding of users' needs and how this could provide insight into how the platform should evolve.

3.1 Supporting experiments from setup through completion

TopHat offers its users four broad services that follow the experiment lifecycle:

Setup A large part of the interest of deploying an application on PlanetLab is to expose it to a diversity of network locations and conditions. Examples of characteristics that a researcher might seek include: locations in Europe, Asia, and North America; nodes that are far from each other in terms of traceroute hops, AS path, or delay; nodes that are collocated with high-precision measurement boxes; particularly stable routes between nodes; paths that have load balancing routers; or a range of available bandwidths. The core PlanetLab services do not aid researchers in choosing their nodes on such bases, leaving it to a service such as TopHat.

Live The underlying topology between PlanetLab nodes, and characteristics of that topology, will typically evolve during an experiment, due to network anomalies, such as path failures, the emergence of bottlenecks, and other sources of network dynamism. These changes are of interest for experiment control: for instance, a researcher might want to restart an experiment if certain paths have changed. They are also of interest for the applications themselves. A peer-to-peer application might adapt its overlay as a function of changing delays and available bandwidth in the underlay. TopHat offers measurements on demand. Also, to avoid the need for polling, TopHat offers a callback service. Sec. 3.2 provides more details.

Rewind The service we brand "rewind" offers a researcher access to measurements related to an experiment once it is finished. He can use these data to understand application performance. A user will typically repeat the same experiment several times while varying some control parameters. The retrospective data can help him tease apart the effects that are due to changes in the parameters from those that are due to evolutions of the network topology. The data could also serve as inputs to a simulation, allowing the changes to be replayed while further parametric variations are explored.

Viz This service consists of a collection of visualization tools that a researcher can use to obtain graphical representations of his experimental data.

3.2 User and application interfaces

To promote ease of use, TopHat presents the user with a simple measurement query interface. The user need not concern himself with cumbersome details if he does not wish to do so. For instance: he can simply ask for an available bandwidth measurement without specifying which tool TopHat should use; he can request a one-way delay measurement without himself synchronizing the measurement agents on two hosts; he does not need to parse the output of diverse tools, or handle the various error conditions that might arise. TopHat

provides an opportunity for the user to benefit from best-of-breed tools while only focusing on the core of his experiment.

The user can specify a class of measurement, such as traceroute, latency, available bandwidth, or topological distance at the IP or AS level, and TopHat itself will select the specific tool for that class. For example, when asked for a traceroute, TopHat would normally select our own team's Paris Traceroute tool [19] because of its ability to avoid many of the measurement artifacts that the standard traceroute tool encounters in the presence of load-balancing routers. The user does not need to know this, but he can learn it if he so wishes: by requesting the information, the user can obtain the name and the version of the tool that TopHat has selected. Furthermore, if the user does wish to request a particular tool, among the tools that are available, he is free to do so.

In addition to providing direct responses to user requests, TopHat allows requests to be registered for later reply. These can be either requests that require some time to fulfill, and therefore are more adapted to an asynchronous reply; requests for periodic updates; or requests for callbacks to be triggered by measurements and other events. These latter two forms of reply allow the user or his experiment to take actions in response to change. Such actions might include starting or stopping an experiment, readjusting an overlay topology, or triggering a set of measurements.

Examples of events that can trigger a callback are: a routing change as measured by traceroute, a delay increase of more than 20% along a given path, or the availability of a new available bandwidth measurement from the background measurement service. The callback information consists of the change that has occurred, a reference to the callback conditions that the change triggered, and a timestamp. Channels to inform the user of changes include: the XML-RPC interface, updates to in the user's space on the TopHat website, e-mail alerts, and RSS feeds. A user can browse the event history on the TopHat website.

3.3 Leveraging historical data

TopHat regularly conducts its own background measurements, compiling a general use archive. It can provide data to those interested in the long term evolution of network topology, or to those who want to look back to a specific point in time, for example to see what happened during a network failure or an attack. These measurements are also available to serve users' requests.

A user may specify a time frame when requesting a measurement, indicating the period over which he considers the measurement to be valid. He might be interested in a very recent measurement (no older than one minute, for instance) or be more flexible (if, say, a measurement from any time during the past day would do). If TopHat has an archived measurement that corresponds to the requested time frame, it can serve the user with that measurement, avoiding the need to launch additional probe traffic. In cases where measurements require significant time to carry out, serving the result from the archive provides the user with a faster response.

In addition to being able to specify a time frame for measurement validity, the user can specify a time interval for measurement aggregation. For instance, a user might be interested in collecting a set of traceroutes between a source and a destination. If the motivation is to understand the current state of the network, he might want the past few hours' traceroutes. On the other hand, if the motivation is to uncover rarely seen alternate links between routers, he might want information from several weeks of measurements.

Similarly, a user can request summary information from aggregated data. Requests might include: an average value, its variance, or the most frequently seen values. Such information can be used for node selection in the setup phase of an experiment: is some path characteristic between two nodes, such as delay or available bandwidth, stable or not? If the experiment is to be a short one, perhaps only the current state of the network is of interest. If, on the other hand, a user plans to deploy a long term service then he might wish to select nodes based upon measurements that indicate historic stability.

3.4 Understanding user requirements

Although TopHat's main objective is to provide a service for users and the applications that they run on PlanetLab, we currently have no direct information about how people use the testbed. Our understanding comes to us through experiment descriptions in the literature and from what hints we can extract from traffic that originates from testbed nodes (the proportion of traceroute traffic, etc.) We therefore instrument TopHat to give us a better picture of users' measurement needs.

Logging of TopHat usage helps us to determine which features users exploit regularly and which ones either do not interest them or are too complicated or inaccessible to be much used. We can shape our future design of the system accordingly. The logs are also important for us to report to our sponsors, to indicate the extent to which the system that they paid for is in fact being used, and to what ends. And the usage logs are of interest to those who study testbed usage in general. Finally, usage logs help us to monitor the service and debug any problems, as well as engineer the system over the long term.

Beyond automated usage monitoring, we plan to work closely with application developers to understand their needs and to integrate the features that they find most useful.[1]

4 Description of the TopHat measurement infrastructure

4.1 Architecture overview

Fig. 1 presents a global overview of the TopHat architecture, divided into functional blocks. The black boxes represent tasks run by the system, and the arrows indicate the flow of data through the system. Users and applications are at the

[1] We welcome inquiries.

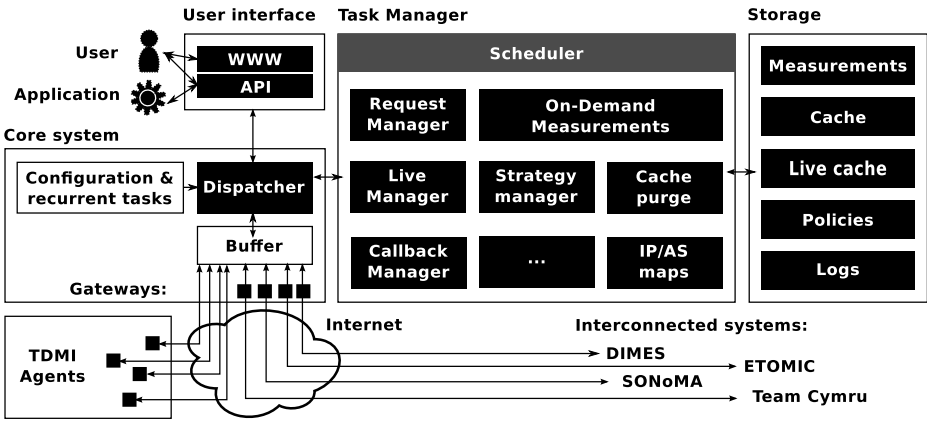

Fig. 1. TopHat architecture

top left. Users access TopHat either through the web interface or, via the XML-RPC API, at the command line. Applications use the XML-RPC API. The measurement infrastructures that conduct the measurements are at the bottom. On the bottom left are TopHat's own measurement agents, which are deployed within a slice on PlanetLab nodes. This is the TopHat Dedicated Measurement Infrastructure (TDMI), supplying measurements when no other system can do so. Other measurement infrastructures are on the bottom right. These are interconnected to TopHat via gateways. Mediating between the user and application requests and the measurement infrastructures is the core system, at center left. The core system dispatches requests and measurements to the task manager, top center. Data are stored in the storage subsystem, top right.

Origin of measurements The measurements originate either from TDMI or, via gateways, from the interconnected measurement systems. TopHat's dedicated agents are modular daemons that consist of wrappers around common measurement tools and basic services, like file upload and packet forging. The tools are invoked by dynamically loadable modules that perform tasks ranging from periodically starting a set of measurements to providing an XML-RPC interface to the agents. TDMI agents incorporate some improved measurement mechanisms that authors of this paper have helped develop, including Paris Traceroute [19], which more accurately measures internet paths that contain load balancing routers, and Doubletree [20], an algorithm to enhance the efficiency of a distributed probing infrastructure.

Gateways are specialized versions of the agents just described. They authenticate themselves to the external measurement infrastructures and ensure the exchange of data with TopHat. Gateways translate the requests originating from TopHat into platform-specific requests and wrap the results in a format that TopHat can understand, appending metadata to the measurements.

The flow of data through the system Data files are uploaded to a buffer in the core system which, upon reception, associates each one with a task for further processing. The tasks are modular entities that can dynamically be added to the task manager to add new functionalities. Two classes of task are scheduled according to two levels of priority, depending on whether they are part of the live flow of information or whether they can be delayed to some extent. High priority tasks are those that are mandatory to ensure interactive communication with the user: measurement requests, updates from an agent that might trigger a callback, etc. Low priority tasks are generally those regularly created by the server as part of background activities such as ongoing IP and AS-level mapping, updates to the dataset that informs the probing strategy, cache purges, etc.

Both the core system and the agents maintain caches adequate to avoid the transmission and processing of redundant information, and to support high rates of measurement data transfer. Thus, the task manager only asks the agents for a new on-demand measurement when it is unable to fulfill the request from its own cache. Similarly, a traceroute measurement that has not changed since the previous measurement won't trigger a new database entry, but simply an update (first in the cache, then when the information is synchronized to the database). This extends the notion of measurement reuse by accounting for user needs. A user can obtain a measurement more quickly if his request specifies a time interval tolerance sufficient to serve the response out of the cache. The mechanism could be adapted in the future to serve predictions, such as those proposed by iPlane [15] and Atmen [16]. While the system does not limit measurements to avoid abuse, as Scriptroute [6] does (in TopHat, this is handled by the acceptable use policy and the ability to trace measurements back to users), policies are implemented to protect the system itself from being overwhelmed by measurement demands.

Data storage The TopHat database records measurements as well as metadata such as agent system logs. Certain types of measurements, such as traceroutes, typically don't change much, if at all, from one measurement to the next. TopHat reduces the load on its database in such circumstances by avoiding rewrite of the entire measurement and only writing the differences and new timestamps. TopHat also employs caches at the agents, for robustness, and along the paths from agent to database. Caching is important in our architecture as it allows quick responses to the most frequent requests, the storage of snapshots representing the state of measurements over a given time interval, and, when a set of data is going to be subject to calculation, ready access to that set. TopHat also stores information such as its own system logs, the dataset of IPs it is currently probing, and policy-related information such as blacklists, rate limits, etc.

Scalability considerations TopHat uses a centralized server architecture much like other monitoring systems, such as CoMon [21], that are currently deployed on PlanetLab. It does so for the same reason: the architecture is simpler and therefore more robust. System logs allow us to uncover bottlenecks, and none have been insurmountable so far. We will continue to study the system's

scalability as it grows. TDMI agents conduct measurements in a full mesh, which increases server load as the square of the number of agents. As the number of agents increases, we will look to improved algorithms for avoiding measurement redundancy, along the lines of Doubletree [20], to improve the system's scaling characteristics.

4.2 User interface and API

TopHat provides two main interfaces: an XML-RPC API that allows an application, or a sophisticated experimenter (using a command-line interface) to interrogate the system, and a web interface that provides greater ergonomy for many of an experimenter's tasks. Typically, the web interface is more convenient during experiment setup for such tasks as node selection, while the API will be used by the application to perform measurements when it is running, or to react to changes in the underlying network.

The core API is the set of functions made available to the user, and that allow him to benefit from the functionalities presented in Sec. 3. The most important functions are:

Get allows the user to request information and measurements about nodes and paths in the monitored topology. A `Get` can be a standard request, an asynchronous request, or a request for periodic updates, as described in Sec. 3.2. In the latter two cases, the system responds via a callback.

Filter is a convenience function that filters a set of nodes or paths according to specified criteria. For instance, suppose the experiment requires twenty nodes that are each at least ten IP hops away from all of the others. The user can request a long list of nodes via the `Get` function and then pass that list to the `Filter` function along with the conditions on path length and number of nodes.

SetCallback is used to configure conditions on which the system will react by triggering a callback function, as described in Sec. 3.2. The API also features a set of related functions to help the user manage his list of callbacks (list, deletion, etc.)

An full description of the up-to-date API is available on TopHat's website.[2] This paper restricts itself to illustrating the `Get` function.

4.3 Requesting a measurement

The prototype of the `Get` function is as follows:

```
RET = Get(Auth, Method, Timestamp, Input, Output, Callback)
```

[2] `http://www.top-hat.info/`

The parameters of the request The first parameter, `Auth`, is an authentication token similar to the one used to authenticate with PlanetLab [22]. Authentication can be password based or key based. We are currently working on a common authentication mechanism for all OneLab platforms, and the use of this parameter will be updated accordingly.

`Method` describes the type of information or measurement we are requesting. A simple request is generic, using a keyword from a high level taxonomy (e.g., *traceroute* or *nodeinfo*). A more sophisticated request asks for a specific measurement tool.

The `Timestamp` parameter specifies the time that the request refers to. This can be a simple textual description (e.g., *now*, *latest*, or *today*), a UNIX timestamp (to get the closest measurement), or an interval. In the case of an interval, the user can ask for a variety of information, such as the first or last measurement in the interval, a list of measurements, or an average value.

`Input` specifies the object or objects to be measured, such as a path, or a set of paths, a node, or a set of nodes. The allowed values depend upon `Method`. The standard way to specify a node is to give its hostname or its IP address.

Each method returns a set of fields that are particular to that method. For example, the *traceroute* method returns the source and destination IP addresses (*src_ip* and *dst_ip*); a list of entries for each hop, consisting of the hop number (*hops.ttl*), the IP address (*hops.ip*), and the DNS name (*hops.hostname*) of the node; as well as additional information such as the presence of load balancing on the path, a timestamp, the platform the measurement originates from, etc. The *nodeinfo* method returns the IP address and hostname of a node (*ip* and *hostname*); the autonomous system that it is part of (*asn* and *as_name*); the city in which it is located (*city*); a *precision* field indicating the type, if any, of high-precision measurement equipment at that location (thanks to collocation with an ETOMIC node, for example); etc. The user specifies which fields he wants to receive by providing a set of their names to the `Output` parameter.

Finally, the `Callback` parameter is used for asynchronous requests, which typically take some time to answer, or requests for periodic updates. The `Timestamp` specifies the desired frequency update. For the simplest requests, this parameter will go unused, as in the sample query below.

Sample query Fig. 2 illustrates a Python query. The request calls for traceroutes from two nodes. One of the nodes belongs to the TDMI platform, the other to SONOMA.

This sample query returns a list of associative arrays that each describe a traceroute with the requested fields: source and destination IP, then, for each hop, the TTL, the IP address, and the corresponding hostname, and finally the platform that performed the measurement. Additional fields such as *tool*, *version* and *timestamp* can be added to obtain further information about the measurements; for the first traceroute this would have given for instance: *tool*='Paris Traceroute' and *version*='0.92b'.

Note how supplementary information can communicate the provenance of the measurements, which is an important feature for an interconnected measurement

Query:

```
path_list = [('planet2.elte.hu', 'planetlab-europe-02.ipv6.lip6.fr'),
             ('ape.onelab.elte.hu', 'planetlab-europe-02.ipv6.lip6.fr')]
print TopHat.Get(auth, 'traceroute', 'now', path_list,
    ['src_ip', 'dst_ip', 'hops.ttl', 'hops.ip', 'hops.hostname', 'platform_name'])
```

Result:

```
[{'src_ip': '157.181.175.248',  'dst_ip': '132.227.62.19',
    'hops': [ {'ttl': '1', 'ip': '157.181.175.254',  'hostname': None},
              {'ttl': '2', 'ip': '157.181.126.45',  'hostname': 'taurus.taurus-leo.elte.hu'}, ...],
    'platform_name': 'TDMI'},
 {'src_ip': '157.181.175.247', 'dst_ip': '132.227.62.19',
    'hops': [ ... ],
    'platform_name': 'SONoMA'}
]
```

Fig. 2. Sample traceroute request dispatched to two platforms

system: a point elaborated upon in Sec. 5. This issue of provenance also arises when supplying inferred data to the user. For instance, when a set of IP aliases to a router has been inferred, the user might want a pointer to the technique and/or data source that was used. (This inferred information is also distinguished from raw measurements in the database.)

5 Interconnection

The example in the previous section shows how TopHat makes use of its connections with other platforms to satisfy measurement requests. This section elaborates on the systems with which TopHat is currently connected, DIMES, ETOMIC, SONOMA, and Team Cymru, which were briefly described in Sec. 2, explaining the motivations for this interconnection. It also sets forth the case for a future connection with perfSONAR. The section wraps up by describing ways in which we can generalize our approach to interconnection.

5.1 Infrastructures connected to TopHat

DIMES [9] is notable for the large number of vantage points that it offers (1700 measurement agents were active on a recent day), and the fact that many of these vantage points are on people's home computers, providing diversity compared to the NREN environment in which most PlanetLab nodes are located. DIMES is able to freely share its data with other measurement infrastructures, which is not the case for other systems of this type, such as Ono [10]. The interest of having access to measurement agents located outside of the testbed stems

from PlanetLab's openness to the internet as a whole. Experimental applications on PlanetLab make use of this openness. For instance, a content distribution network (CDN) deployed on PlanetLab might be designed to serve content to a user via its nearest PlanetLab node, with distance calculated at the AS or IP level. Outside measurements can help determine these distances. Similarly, a network coordinate system can benefit from measurements taken from a large number of vantage points.

The ETOMIC [11] and SONOMA [12] systems are notable for the higher than ordinary precision that they bring to measurements. They are GPS-synchronized dedicated measurement boxes. ETOMIC boxes, which must be reserved, provide delay measurements with a precision of tens of nanoseconds. SONOMA boxes can run measurements concurrently with a precision of tens of microseconds. A couple of dozen PlanetLab sites currently house both ETOMIC and SONOMA boxes, which clearly makes them of interest to TopHat. The precision of these boxes is useful for calculating one-way delays and available bandwidth, and for geolocalization.

The Team Cymru IP to AS mapping service [4] is notable for offering information drawn from a large number of BGP feeds and making it easily accessible to be queried over the internet. By connecting with the Team Cymru service, TopHat avoids the need to receive and process these BGP feeds itself.

We are currently exploring the possibility of interconnecting TopHat with perfSONAR [8], which is notable for offering extensive measurements from privileged vantage points within the network. perfSONAR obtains measurements directly from routers and from agents located at network points of presence (POPs). Some of the information that it provides, such as router queue lengths and packet drop rates, is not normally available to outside researchers and can at best be inferred. The perfSONAR information is of particular interest to researchers on PlanetLab because communication between most PlanetLab nodes crosses NREN infrastructure that is instrumented by perfSONAR.

5.2 Generalizing our approach to interconnection

Our work on TopHat takes place in the context of the larger effort to federate computer networking testbeds. Initiatives such as FIRE [23] in Europe and GENI [24] in the United States are pursuing the vision of a worldwide federation of testbeds that will allow experimentation with new networking technologies at a global scale. OneLab [2] pioneered this vision, starting its work on federation in September 2006. An effort to develop measurement infrastructures for these testbeds has been a part of OneLab since the beginning, and TopHat is one of the results.

Our approach with TopHat has been to build on the interconnection mechanisms that emerge from the work on testbeds, as TopHat exists to be at the service of these testbeds. We see a first example of this in the TopHat authentication mechanism. TopHat operates on PlanetLab, which as a result of OneLab is now a federation of testbeds, PlanetLab Europe having joined the original

PlanetLab in the United States. The same mechanisms that allow users to authenticate themselves to the global PlanetLab system serve to authenticate them with TopHat.

We plan to extend this authentication mechanism to encompass the infrastructures with which TopHat interconnects. At present, each interconnection has its own particular authentication mechanism, encapsulated in our gateway architecture. However, some of these platforms, such as ETOMIC, are testbeds in their own right, with users who can log in and run experiments. The potential exists, with a common authentication mechanism, to allow users of these systems access to TopHat and the OneLab facility, just as TopHat now has access to these systems. Rakotoarivelo et al. [25] have underlined how researchers are ever more inclined to deploy their experiments across different testbeds on different administrative entities, or to repeat the same experiment on different testbeds.

Other aspects of interconnection can also be standardized. For instance, the language that a system uses to describe the resources that one system requires from another. Here too, we can borrow from work currently being done on testbed interconnection. Various proposals for generic resource specifications (RSpecs) are currently emerging from PlanetLab [26] and OMF [25], among others, and efforts are taking place, notably within GENI [27], to harmonize them.

Another area that can benefit from standardization is the way in which measurements are described. Work in the IP Performance Metrics Working Group at the IETF has led to an XML specification for traceroute information [28], for instance.

Finally, we believe it is important to standardize a system for usage accounting across systems. Since active network measurements are potentially disruptive, these systems can only function if there is accountability, meaning the ability to trace a measurement back to the user who requested it. Accounting is also valuable to system operators, to enable them to better understand who is using their systems and for what purposes, and plan system development accordingly.

6 Conclusion

This paper has presented TopHat, a topology information service for the PlanetLab testbed. TopHat is oriented specifically towards the support of experiments running on the testbed, from setup through completion. The service provides data about the underlying network that help a PlanetLab user to choose the nodes that will be part of his experimental overlay. In so doing, it enables users to exploit the geographical and topological and diversity that PlanetLab uniquely offers. The service also assists a running experiment by offering live measurements through a simple query language, and by allowing the user to define callbacks that will keep him informed of significant changes, such as a change in the topology, or increased delays along a path. Other aspects of the service include measurement archiving and data visualization.

TopHat aggregates measurements coming from several infrastructures, including DIMES, which has measurement agents at a large number of vantage points, and ETOMIC and SONOMA, which offer high precision measurements. The user benefits transparently from this range of capabilities, accessing them through a simple common interface with PlanetLab based authentication.

This interconnection of measurement systems is an instance of the more general issue of testbed federation. TopHat provides one model for handling common authentication of users, description of resources, and the exchange of control messages.

As new features emerge for TopHat, they are being deployed in PlanetLab Europe, the European arm of the global PlanetLab testbed, in preparation for roll-out worldwide. This is part of a larger development effort: PlanetLab Europe is the flagship testbed of the OneLab experimental facility, and TopHat is a component in MySlice, a more general experiment management facility for OneLab.

Acknowledgments

We thank our OneLab colleagues, and in particular those of the ETOMIC, DIMES, and MySlice teams, for their thoughtful comments on TopHat as it has evolved.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°224263-OneLab2.

References

1. C. Schmoll, J. Quittek, A. Bulanza, S. Zander, M. Kundt, E. Boschi, and J. Sliwinski. State of interoperability. Deliverable D11, FP6 IST MOME project, 2004.
2. The OneLab experimental facility. `http://www.onelab.eu/`.
3. University of Oregon Route Views project. `http://www.routeviews.org/`.
4. Team Cymru IP to ASN mapping service. `http://www.team-cymru.org/Services/ip-to-asn.html`.
5. The Prefix WhoIs Project. `http://pwhois.org/`.
6. N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A public internet measurement facility. In *Proc. USITS*, 2003.
7. N. Spring, L. Peterson, A. Bavier, and V. Pai. Using PlanetLab for network research: myths, realities, and best practices. *SIGOPS Oper. Syst. Rev.*, 40(1):17–24, 2006.
8. A. Hanemann, J. W. Boote, E. L. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, D. M. Swany, S. Trocha, and J. Zurawski. PerfSONAR: A service oriented architecture for multi-domain network monitoring. In *Proc. ICSOC*, 2005.
9. Y. Shavitt and E. Shir. DIMES: let the internet measure itself. *ACM SIGCOMM Comput. Commun. Rev.*, 35(5):71–74, 2005.
10. K. Chen, D. Choffnes, R. Potharaju, Y. Chen, F. Bustamante, D. Pei, and Y. Zhao. Where the Sidewalk Ends: Extending the internet AS graph using traceroutes from P2P users. In *Proc. ACM CoNEXT*, 2009.

11. D. Morato, E. Magana, M. Izal, J. Aracil, F. Naranjo, F. Astiz, U. Alonso, I. Csabai, P. Haga, G. Simon, J. Steger, and G. Vattay. The European Traffic Observatory Measurement Infraestructure (ETOMIC): A testbed for universal active and passive measurements. In *Proc. Tridentcom*, 2005.
12. SONoMA measurement infrastructure. `http://www.complex.elte.hu/sonoma/`.
13. F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, H. Uijterwaal, and R. Wilhelm. Providing active measurement as a regular service for ISP's. In *Proc. Passive and Active Measurements Workshop (PAM)*, 2001.
14. A. Nakao, L. Peterson, and A. Bavier. A routing underlay for overlay networks. In *Proc. ACM SIGCOMM*, 2003.
15. V. 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.
16. B. Krishnamurthy, H. V. Madhyastha, and O. Spatscheck. ATMEN: a triggered network measurement infrastructure. In *Proc. WWW*, 2005.
17. P. Matray, I. Csabai, P. Haga, J. Steger, L. Dobos, and G. Vattay. Building a prototype for network measurement virtual observatory. In *Proc. MineNet*, 2007.
18. Archipelago measurement infrastructure. `http://www.caida.org/projects/ark/`.
19. B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *Proc. ACM IMC*, 2006.
20. B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Efficient algorithms for large-scale topology discovery. In *Proc. ACM SIGMETRICS*, 2005.
21. K. Park and V. S. Pai. CoMon: a mostly-scalable monitoring system for PlanetLab. *SIGOPS Oper. Syst. Rev.*, 40(1):65–74, 2006.
22. PlanetLab Central API Documentation: Authentication. `https://www.planet-lab.eu:443/db/doc/PLCAPI.php#Authentication`.
23. FP7 Future Internet Research and Experimentation (FIRE) initiative. `http://cordis.europa.eu/fp7/ict/fire/`.
24. NSF Global Environment for Network Innovations (GENI) initiative. `http://www.geni.net/`.
25. T. Rakotoarivelo, M. Ott, I. Seskar, and G. Jourjon. OMF: a control and management framework for networking testbeds. In *Proc. SOSP Workshop on Real Overlays and Distributed Systems (ROADS)*, 2009.
26. L. Peterson, S. Sevinc, S. Baker, T. Mack, R. Moran, and F. Ahmed. PlanetLab implementation of the Slice-Based Facility Architecture, 2009. Draft technical report. `http://www.cs.princeton.edu/~llp/geniwrapper.pdf`.
27. T. Faber and R. Ricci. Resource description in GENI: Rspec model, Mar. 2008. Presentation given at the Second GENI Engineering Conference.
28. S. Niccolini, S. Tartarelli, J. Quittek, T. Dietz, and M. Swany. Information model and XML data model for traceroute measurements. RFC 5338, IETF, Dec. 2008.