

mPlane: an Intelligent Measurement Plane for the Internet

Brian Trammell^{*}, Pedro Casas[†], Dario Rossi[‡], Arian Bär[†], Zied Ben Houidi[§]
 Ilias Leontiadis[¶], Tivadar Szemethy^{||}, Marco Mellia^{**}

^{*}ETH Zurich [†]FTW Vienna [‡]Telecom ParisTech [§]Alcatel-Lucent Bell Labs
[¶]Telefonica I+D ^{||}Netvisor ^{**}Politecnico di Torino

Corresponding Author: trammell@tik.ee.ethz.ch

Abstract—The Internet’s universality is based on its decentralization and diversity. However, its distributed nature leads to operational brittleness and difficulty in identifying the root causes of performance and availability issues, especially when the involved systems span multiple administrative domains. The first step to address this fragmentation is coordinated measurement: we propose to complement the current Internet’s data and control planes with a measurement plane, or *mPlane* for short. *mPlane*’s distributed measurement infrastructure collects and analyses traffic measurements at a wide variety of scales to monitor the network status. Its architecture is centered on a flexible control interface, allowing the incorporation of existing measurement tools through lightweight *mPlane* proxy components, and offering dynamic support for new capabilities. A focus on automated, iterative measurement makes the platform well-suited to troubleshooting support. This is supported by a reasoning system, which applies machine learning algorithms to learn from success and failure in drilling down to the root cause of a problem. This paper describes the *mPlane* architecture and shows its applicability to several distributed measurement problems involving Content Delivery Networks (CDNs) and Internet Service Providers (ISPs). A first case study presents the tracking and iterative analysis of cache selection policies in Akamai, whereas a second example focuses on the cooperation between ISPs and CDNs to better orchestrate their traffic engineering decisions and jointly improve their performance.

Index Terms—Distributed Measurements; Internet Monitoring; Automatic Troubleshooting Support; Machine Learning.

I. INTRODUCTION

The decentralized nature of the Internet leads simultaneously to its resilience and to the difficulty of identifying and tracking the root causes of performance and availability issues. The first step to improve this situation is via measurement: illuminating the currently obscure dynamics of the Internet. To address this issue, we advocate a measurement plane, or *mPlane*¹, alongside Internet’s data and control planes.

mPlane is a distributed measurement infrastructure to perform passive, active, and hybrid² network measurements, continuously or on-demand, at a wide variety of scales, with built-in support for iterative measurement and automated iteration. *mPlane* components consist of flexible probes, which perform measurements. Repositories, which store, aggregate, correlate,

and analyze them, distilling raw measurements into knowledge about the network. And a supervisor for coordinating measurements. An intelligent reasoning system (reasoner from now on) provides support for automating iteration, drilling down to find the root cause of an observed phenomenon by successively ordering increasingly finer measurements.

By enabling measurement throughout the Internet, *mPlane* benefits everyone: Internet Service Providers (ISPs) get a fine-grained picture of the network status, empowering effective management and operation. Application providers gain powerful tools for optimizing performance of their application by accessing to network layer information. Regulators and end-users can verify adherence to SLAs, even when these involve multiple parties crossing the boundaries of the ISP. Finally, customers of all kinds can objectively compare network performance, improving competition in the market.

Many tools and platforms have been proposed in this space in the past, such as PerfSONAR [2] and RIPE Atlas³. RIPE Atlas only considers the case of active measurements, thus highly limiting its monitoring capabilities. And while *mPlane*’s architecture might a-priori look similar to the one of PerfSONAR, the complete *mPlane* framework goes beyond the latter by also focusing on the analysis of the gathered data to diagnose service and network problems and to monitor the performance of inter-domain paths and networks. *mPlane* differs from these tools in that the platform is specifically designed to adapt to the heterogeneity of existing measurement tools as well as to new, advanced analytic components. The development of the reasoner is a key result that will allow structured, iterative, and automated analysis; finally, an emphasis on open interfaces will speed up adoption and increase the impact of the *mPlane* already during the project lifetime.

In this paper, we first examine the open interfaces that make up the *mPlane* architecture, focusing on the *mPlane* workflow, the role of the reasoner therein, and the *mPlane* model for cross-domain measurement. To illustrate the benefits of the *mPlane* approach, we present two case studies involving ISPs and CDNs. The former considers the problem of tracking and iteratively analyzing the cache selection policies employed by a major CDN, from the perspective of an ISP. The latter focuses on the problem of cross-domain collaboration between

¹*mPlane* is a three-year long Integrated Project started in November 2012, funded by the European Commission. See <http://www.ict-mplane.eu/>

²Passive observation of active probing traffic.

³The RIPE Atlas Internet measurement network, <https://atlas.ripe.net/>

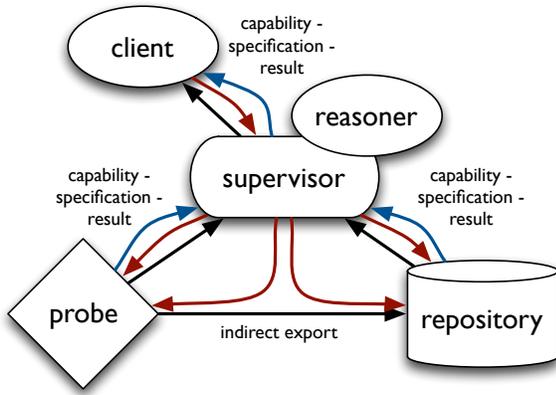


Figure 1. Components and interactions in the mPlane architecture. Blue lines are capabilities announcements, red lines indicate control messages, and black lines correspond to data flows.

ISPs and CDNs as a relevant scenario. For completeness, we examine the application of mPlane to other use cases.

II. ARCHITECTURE

Fig. 1 depicts a schematic view of mPlane. An mPlane infrastructure consists of a set of components spanning over multiple domains, whose overall workflow is (i) *flexible*, supporting continuous background as well as on-demand and iterative measurement; (ii) *compatible*, through the integration of existing measurement tools and platforms; and (iii) *widely interoperable*, through the application of standards to the export formats and protocols as well as to the metrics supported by the platform themselves. In mPlane, *everything is a component*, and the architecture is largely defined by the protocol among the components.

A. Components

Components are the building blocks that perform passive, active and hybrid measurements, and that store and analyze the resulting data. Components can be roughly divided into *probes* and *repositories*. Probes perform raw measurements, and may pre-process them. Repositories store, correlate, and analyze the measurements provided by multiple probes. These components are coordinated by a *supervisor*, which handles the mechanics of distributed measurement (e.g., component discovery, capabilities management, access control, etc.) as well as final analysis, correlation, and compilation of results from multiple repositories and probes. The supervisor performs actions on behalf of its *clients*, which represent mPlane end users, whether a human operator acting through a user interface, or an automatic management process. A special client, optionally integrated within the supervisor, called the *reasoner*, is responsible for automated iteration, and is described in detail in section II-C.

Probes and repositories advertise their *capabilities* in terms of the types of data they can produce and consume, and the operations they can perform thereon, to the supervisor (blue arrows in Fig. 1). The supervisor can in turn expose

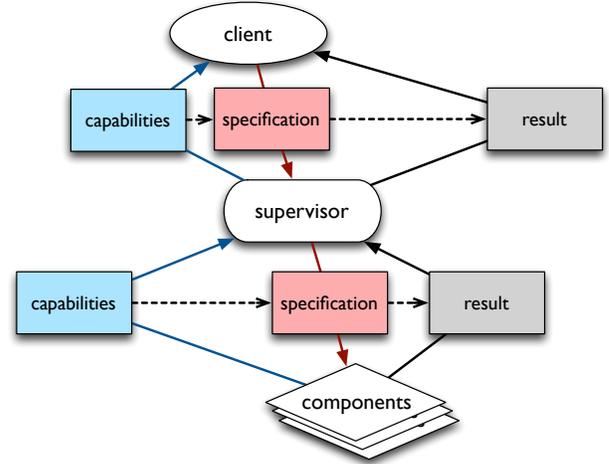


Figure 2. The mPlane workflow. *Capabilities* define the tasks a component can perform. *Specifications* consist of a description of which measurement have to be performed, how, and when.

higher-level capabilities based on these to the clients. In turn, the client controls the probes and repositories via control messages sent through the supervisor (red arrows). Finally, data generated by a probe (black arrows) can directly be sent to the supervisor, or stored into repositories for later analysis.

The division of roles among components need not be this strict in practice. As mPlane components explicitly advertise their capabilities, any meaningful subset of the probe, repository, and supervisor functions can be co-located or even supported by the same component. More generally, any entity that implements the mPlane protocol is an mPlane component.

B. Workflow

Assessing complex traffic behaviors is not an easy task, and measurement points must be coordinated to gain a solid understanding of, e.g., the root causes of an issue. A key feature of most troubleshooting workflows is *iterative measurement*: the results of one measurement eliminate possible causes of an issue and point to others, triggering the next measurement to perform. Iterative measurement is especially useful for root cause analysis applications. However, not all measurement workflows are iterative, and iterative measurements often require comparison to background information generated by continuous or periodic measurements. The data and control flow within the architecture must therefore support both an inherently cyclic workflow in the “foreground”, as well as the management of a large set of continuous/periodic measurement in the “background”, whether autonomously performed by probes or directed by supervisors. Flexible interfaces facilitate the flow of control messages to trigger new measurements and get the data in return, supporting both synchronous and asynchronous operation modes.

As shown in Fig. 2, interaction in mPlane begins from a set of component capabilities. From the bottom (blue path), a supervisor collects the capabilities of the components it supervises, and presents capabilities to its clients (e.g., representing measurements it can perform or queries it can answer with its

components). From the top (red path), a client selects some set of capabilities and sends a *specification* to the supervisor, i.e., a description of which measurement to perform, how, where, and when. The supervisor authenticates the client, checks its authorization to perform the measurements called for in the specification, and sends corresponding specifications to the appropriate components. Results (black path) can be returned instantaneously, in which case they are presented over the same channel, or retrieved later. Each result contains all the parameters of the specification used to generate it, so that it is self contained. This simplifies management in large-scale deployments, while reducing the amount of state that each component has to store while waiting for one of its specification to return. Table I shows the capability, specification, and result for a simple active probe measuring two way delay via ICMP (i.e., the venerable `ping` utility).

Key to scaling iterative measurement is the realization that state can be distributed across components by the protocol, as opposed to requiring multiple-component state synchronization. Note in the example that the result contains all the information required to generate it: it is a complete declaration about the state of the world as measured. Long-running and background measurement tasks generally involve probes sending data directly to repositories; this “indirect export” as shown in Fig. 1 is brokered by the supervisor, which tells the probe to send and the repository to accept the data, transmitting specifications to both components. Since the interface a supervisor presents to a client is much like the interface a component presents to a supervisor, measurement can be federated by nesting supervisors in larger organizations. In this arrangement, data volumes become lower and information content higher as one rises up in the measurement stack.

C. The Reasoner

The inclusion of a reasoner in the architecture supports automated iteration. The reasoner accesses mPlane as a client, though, depending on implementation, it may be integrated within the supervisor. It relies on knowledge-rules and learning techniques to automate the iterative measurement process. Its main purpose is to provide detailed insights and draw intermediate conclusions on the results provided by the analysis of disparate highly-aggregated measurements, additionally triggering new actions to improve analysis and drill down to more-specific measurements issues.

The mPlane reasoner comes with a set of *analysis modules* which act as “plugins” to extend its functionality, improving the analysis of the measurements gathered by the probes and stored/pre-process at the repositories. Current mPlane platform provides a broad set of analysis modules, which can be roughly grouped into the following categories: (i) classification and filtering (e.g., of flows, applications, content), (ii) estimation/prediction (e.g., of Quality of Experience, popularity, path metrics, topology), (iii) detection (e.g., of anomalies, threshold-based changes, interfering middle-boxes, hidden relationships between policy rules), (iv) correlations discovery (e.g. between measurements and QoE, traffic directions and caches/servers) and (v) diagnosis (e.g., of QoE or web degradation, lack of connectivity).

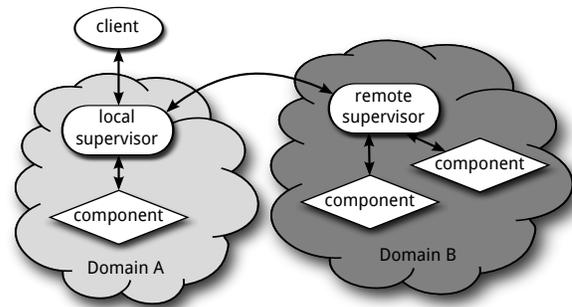


Figure 3. Federation in mPlane through inter-supervisor connections. Supervisors in each domain handle supervisors in external domains as clients.

By learning from previous root cause analysis workflows, the reasoner is able to automatically isolate trouble spots in the network, minimizing human intervention to the strictly necessary (e.g., asserting if a new identified root cause is plausible or not). In terms of the mPlane workflow, the reasoner learns to map expected results to the most likely next specification to generate a high-confidence result.

The reasoner has access to a set of domain-knowledge-based rules that guide its decisions on the iterative analysis process. These rules are conceived as a knowledge structure, which can be augmented by domain-expert end-users. The set of rules is not static or only adapted by domain knowledge information, but expands by learning from past experiences. Current reasoning and learning approaches in mPlane are based on rule-based reasoning, in which decision-tree like graphs are constructed for the specific use cases to tackle. Rule-based reasoning represents a simple and direct association between the diagnosed root cause and the evidence(s) of a problem for better interpretation. We provide a simple example of a diagnosis tree in the case study of section III.

D. Cross-domain Federation

Given the presence of multiple independent administrative boundaries in any end-to-end path across the Internet, an Internet-wide measurement plane must carefully address the issue of measurement spanning multiple domains, where economic and legal relationships can trump technical ones.

Fully distributed measurement provides the greatest flexibility, but it also leads to an authentication and authorization nightmare, as the results can reveal sensitive information about the involved entities and their clients. Measurement involving distributed components in multiple interconnected domains, and domains cooperating on measurements as if part of a larger domain, require the specification of trust relationships and security policies. Defining such policies is especially challenging for components in remote domains, on which one may have no real visibility. Neither does fully distributed measurement reflect how modern networks are managed: inter-domain services require gateways for high-level policy reasons as well as practical, day-to-day security concerns.

mPlane minimizes such issues, by allowing measurements to be collected and owned separately by each involved domain. In this paradigm, multi-domain measurement are handled

Table I
 NOTIONAL EXAMPLE OF CAPABILITY, SPECIFICATION AND RESULT FOR A SIMPLE ICMP PING-LIKE ACTIVE MEASUREMENT PROBE.

capability:measure	specification:measure	result:measure
parameters: start: now...+inf end: now...+inf source.ip4: 192.0.2.3 destination.ip4: * octets.count: 28...65535 period.s: * results: - delay.twoway.icmp.ms.min - delay.twoway.icmp.ms.mean - delay.twoway.icmp.ms.max - delay.twoway.icmp.ms.count	parameters: start: now end: now + 30s source.ip4: 192.0.2.3 destination.ip4: 192.0.2.67 octets.count: 80 period.s: 1 results: - delay.twoway.icmp.ms.min - delay.twoway.icmp.ms.mean - delay.twoway.icmp.ms.max - delay.twoway.icmp.ms.count	parameters: start: 2013-11-27 09:31:18 end: 2013-11-27 09:31:48 source.ip4: 192.0.2.3 destination.ip4: 192.0.2.67 octets.count: 80 period.s: 1 results → values: - delay.twoway.icmp.ms.min → 31 - delay.twoway.icmp.ms.mean → 37 - delay.twoway.icmp.ms.max → 92 - delay.twoway.icmp.ms.count → 30

as communication among supervisors, as shown in Fig. 3. Therefore, the supervisors in each domain essentially handle supervisors in requesting domains as clients. In most cases, the measurement data crossing domain boundaries can be reduced in volume by allowing high aggregation. In other cases only troubleshooting requests can be forwarded (e.g., issue a query to the supervisor of an ISP to verify whether a certain phenomenon is originating within their domain).

Such cross-domain high-level communication simplifies cross-domain management, access control, and data protection, reducing the cross-domain measurement problem to a manageable one. Furthermore, each domain can internally utilize its own measurement infrastructure and apply its specialized knowledge over the network in order to apply its own reasoning algorithms. Requests from external domains can be governed by manageable security policies, with trust relationships evaluated on a per-domain basis, and the supervisor granting or denying access to specific capabilities and selectively applying data protection, such as partial omission or anonymization.

E. Interoperability

To allow rapid scaling of the mPlane platform without duplication of efforts, the mPlane design facilitates the incorporation of the wide variety of existing measurement tools. The simplest way to integrate an existing measurement tool into the mPlane platform is the implementation of an adapter between mPlane and the tool native interfaces. In this case, what the tool can do is represented as a static capability to the supervisor, specifications are translated into the tool native configuration interface, and the tool native output are translated into measurement results.

Many tools have a “probe” part and a “collector” part, and use a common protocol (e.g., IPFIX) to inter-operate with others. Therefore, mPlane is designed to broker the exchange of data between probes and repositories in their native protocols, reducing losses due to protocol translation, and by augmenting the information by the actual description of the data. Measurement names are lexicographically scoped, so definitions can be taken from different fora as appropriate, without colliding or being mistaken for each other. This arrangement will ensure that each measurement result is well-defined and will not allow for possible confusion. The set of protocols supported by probes and repositories for data

exchange and access (e.g., FTP, SSH, HTTP, GridFTP, HFile, etc.) are also represented as capabilities.

Interoperability is also promoted by the use of existing standards. The largest problem in interoperability of measurement tools is not the control protocols they use, but rather the comparability of the measurements they perform. One tool definition of delay, for example, may not mean the same thing as another. In mPlane, measurements are represented as capabilities, whose core vocabulary is taken from standard definitions, relying heavily on the metrics defined by standardization bodies, e.g., the IETF IP Performance Metrics (IPPM) working group.

III. CASE STUDY I: TRACKING CDN BEHAVIOR

CDNs are a vital part of the current Internet infrastructure. By deploying servers in multiple data centers across the Internet, content can be served to end-users with high availability and performance. However, CDNs pose challenges for ISPs, since changes in server allocation policies can cause sudden changes to the traffic carried by ISPs, impacting traffic engineering and possibly impairing end-user quality of experience. As such, ISPs need advanced tools to track and diagnose shifts in the traffic served by CDNs. Among CDN companies, Akamai is the leading CDN provider.

We thus instrumented mPlane to track the traffic served by the Akamai CDN servers as seen from a large ISP. In this case study, mPlane is instrumented using multiple Tstat passive probes [6], an advanced repository for continuous and large-scale data analysis referred to as DBStream [7], and a single supervisor hosted by the ISP. Tstat probes provide per-flow statistics from three Points-of-Presence (PoP) aggregating 45,000 end-users connected to the Internet, which are exported to the DBStream repository for further processing. Flows are pre-filtered at DBStream by correlating the server IPs with the external MaxMIND databases⁴. Specifically, we focus the analysis on a single /25 subnet hosting Akamai caches, which serve the majority of the flows. Servers in this network are reached by a direct peering agreement between the ISP and Akamai. Nodes are very close, typically less than 5ms far away from customers in the monitored PoPs. We refer to this subset as “preferred” in the following analysis.

⁴MaxMIND GeoIP Databases, <http://www.maxmind.com>.

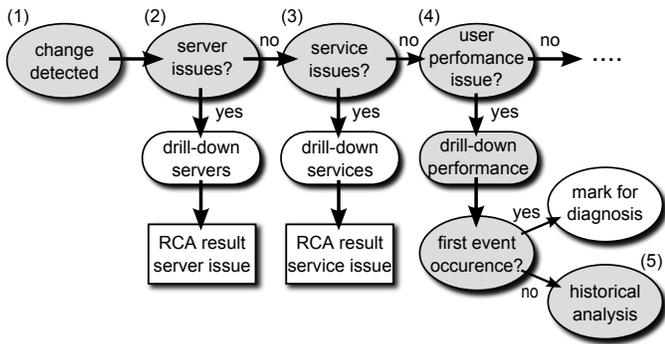


Figure 4. A rule-based reasoning approach for CDN cache selection analysis.

The iterative analysis performed by the reasoner follows the tree-like structure depicted in Fig. 4. We do not provide a full description of the involved processes and analysis modules due to space limitations, but rather follow the sequence of analysis steps involved in the diagnosis of cache selection shifts observed at the Akamai preferred cache. We note that the main goal of this case study is to exemplify the iterative, guided measurements analysis process.

Overall daily patterns and change detection: Fig. 5 (top) details the evolution of the number of flows served by the Akamai CDN on two consecutive days as seen from one vantage point. The preferred cache serves about 30% of traffic at peak time. Surprisingly, traffic served by the preferred cache presents occasional drops. These are effects of the CDN server selection policies shifting traffic back and forth among CDN nodes. Fig. 5 (bottom) reports the evolution of the difference of the number of flows served by the preferred cache in two consecutive time windows of 5 minutes. The iterative analysis process is triggered by the detected abrupt changes in the number of served flows, marked as step (1) in Fig. 4.

Single servers load: The step (2) of the analysis corresponds to checking if the sudden traffic shifts are due to some server failure in the preferred subnet. Fig. 6(a) reports a heatmap of the load for each IP address in the /25 subnet over the 2 days. For each IP address, DBStream computes the fraction of served flows in 5 min. time windows. A color scale is used to represent each cell. The smaller the value, the lighter the color. Only 40 servers are active and constantly used, and few servers handle up to 62% of requests (darker red lines). All servers show lighter colors in correspondence of the traffic shift, thus the server failure hypothesis is ruled out.

Per-service analysis: CDN nodes host very different content, e.g., the same CDN server can host both Facebook and iTunes/AppleStore objects. Tstat exposes this information by snooping the Full Qualified Domain Name (FQDN) of the requested content [8]. At step (3), the reasoner checks if the observed traffic shifts are due to the CDN moving some specific content, reflecting some service-related issues. DBStream filters flows per service, and computes the fraction of requests served by the preferred and other caches. The obtained values are represented by the heatmap shown in Fig. 6(b). The most popular services are reported, sorted by the probability of being served by the preferred cache. The results clearly show two groups: the bottom 300 services are

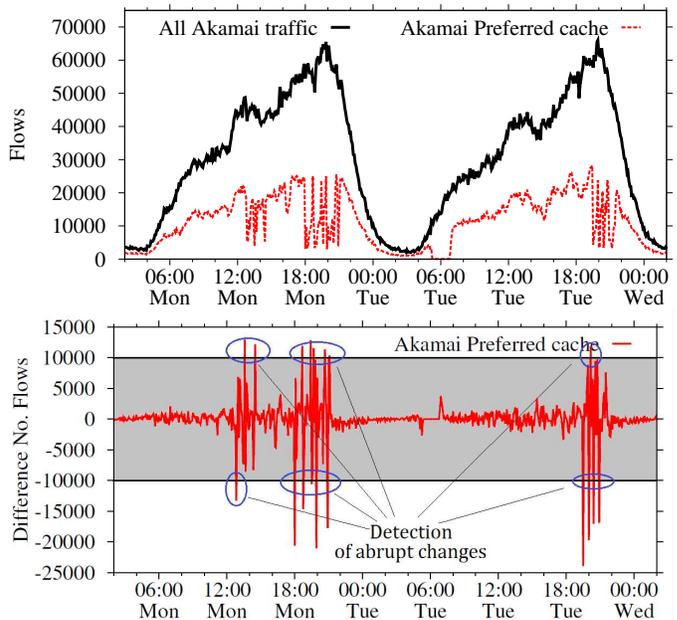


Figure 5. Evolution of number of flows served by Akamai CDN (top) and difference of number of flows served in consecutive 5 minutes time windows (bottom).

normally served by some server at the preferred cache (red dots). The other 200 services are served exclusively by other Akamai CDN servers (green dots). At the same time as the traffic shifts occur, practically all services are migrated to other caches, indicated by the green vertical bars in the plot. Results indicate that the traffic shifts are not related to some particular service, but are rather the effect of changes in the server allocation policies impacting all services.

Impact on performance: Step (4) corresponds to the verification of the end-user performance. The analysis is performed both in terms of downlink throughput and delay. The analysis of the downlink throughput does not reveal any interesting evidence, thus we move on to the analysis of the elaboration time⁵. Fig. 6(c) reports the evolution of the 5th, 25th, 50th, and 75th percentiles of the elaboration time for the considered time period (y-axis is in log scale). Results show that during the traffic shifts on Monday, some impairment of the elaboration time is visible. In particular, the 50th percentile grows from about 10 ms to about 20 ms before and during the shifts happening at 18:00. Even if the same traffic shifts occur also on Tuesday, the 50th percentile of the elaboration time does not increase. The analysis of a whole week of traffic before the event (i.e., step (5), historical analysis) reveals that the same 50th percentile increase happens on all days before Tuesday, but does not occur on Tuesday and the following days. Still, also on Tuesday and on the days after, the same traffic shifts from the preferred cache occur.

The historical analysis does not provide a final root cause for the flagged traffic shifts. Yet, it allows to reveal the occurrence of a maintenance event on Tuesday, visible at Fig. 5

⁵The time between the client first packet with payload, and the server first packet with payload. In case of HTTP, it corresponds to the time between the HTTP-request and the HTTP-response.

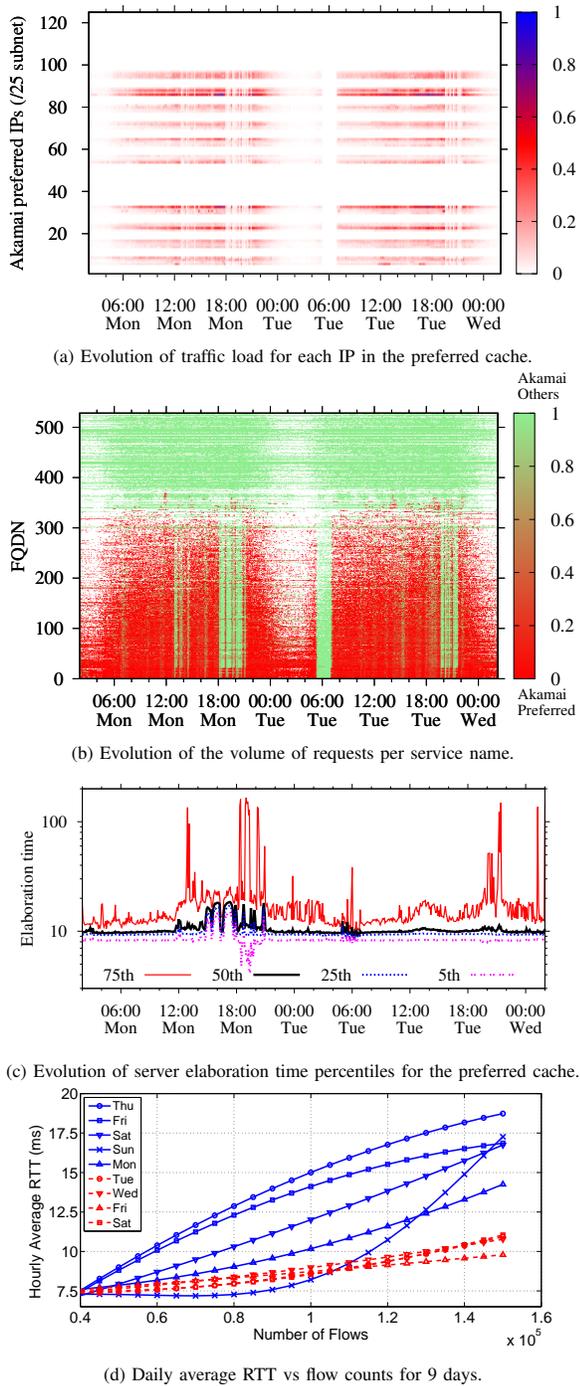


Figure 6. Iterative analysis of cache selection policies in Akamai.

from 5am to 7am as a CDN outage. The most interesting observation from such flagged maintenance event is that the end-performance issues in terms of delay are solved after it, as confirmed in Fig. 6(d), which compares the per-day average RTT between users and serves, before and after this maintenance event.

IV. CASE STUDY II: ISP/CDN COLLABORATION

The second case study illustrates the benefits of mPlane in a cross-domain collaboration scenario, which involves both ISPs and CDNs. The context is similar to the one of previous

example, but includes also the CDN as a participant of the mPlane framework. As previously mentioned, CDN cache selection policies pose difficult traffic management challenges to the ISP. However, the ISP is not the only player affected in this ecosystem. Indeed, the distributed nature of CDNs also poses challenges regarding content management and server selection strategies. CDNs have to dynamically map end-users to appropriate servers without being fully aware of the underlying network topology and network conditions between their servers and the end-users.

This results in a situation in which both ISPs and CDNs take uncoordinated traffic engineering decisions, which can be harmful for both actors, as well as for the end-users. The ISP/CDN collaboration problem has been treated in the literature to date [1], [3]–[5]. While providing different solutions, all these studies claim that the challenges that ISPs and CDNs face separately can be turned into an opportunity for collaboration. On the one hand, CDNs can take advantage of the detailed view of the network topology (e.g., paths, number of hops) and network condition (e.g., delay, bandwidth, overloaded paths) only available to the ISP for better optimizing their server selection strategies. On the other hand, ISPs can regain partial control on the routes followed by its traffic.

The mPlane platform can natively address the ISP/CDN collaboration problem. mPlane provides means to (1) measure both players' networks, (2) analyze these measurements, and (3) exchange useful information without revealing sensitive operational information about either party to the other.

Required Measurements: On the ISP side, the simplest and most important information is the traffic volume on the paths between the end-users and the CDN. Link load information is available via mPlane-wrapped SNMP interface counters or network flow metering, and paths from a repository built around BGP listeners, IGP monitoring results, and static topology information provided by the ISP. Active probes provide additional information about delay along these paths.

The ISP's supervisor is responsible for orchestrating the ISP probes and repositories; collected data are stored in repositories local to the ISP for data persistence and further analysis. As such, the ISP keeps an up-to-date view of the performance its network offers on each of its end-to-end paths, as well as a historical view of its intra-domain performance. This network performance map is the main source of information for improving CDN performance too. ISP information is additionally complemented with high-level information provided by the network administrator concerning planned maintenance events, network upgrades, and upcoming agreements. This allows the local ISP reasoner to predict possible upcoming changes that might affect end-to-end path conditions.

On the CDN side, probes operating on server logs or passive query monitoring expose trends in content popularity, and store them in a CDN-local repository. The CDN supervisor correlates this with CDN topological information, and, as in the ISP case, planned maintenance and upgrade events.

Note that the ISP and the CDN can currently try to infer the data exchanged by the other side using measurements on their local networks, as performed in the previous case study. However, such measurements are necessarily less accurate

and less efficient than those obtained through a collaborative framework.

Inter-Supervisor Federation: We assume that the ISP and the CDN interconnect at multiple different locations. Both the ISP and the CDN run an mPlane supervisor which orchestrates the measurements within their respective networks; these exchange information with each other through the mPlane interfaces. Two collaboration models are possible: in the first model, the ISP supervisor makes its network map available to the CDN supervisor on demand. The CDN supervisor provides a “find best cache location” capability based on this map, which uses cached information about the ISP’s network map to select a cache location for a given user request. In the second model, the ISP considers its network map sensitive; therefore, the CDN supervisor monitors the current load and performance of each CDN cache, and provides the ISP supervisor with the ordered list of cache locations that can satisfy end-user requests. The ISP supervisor couples this information with the ISP network map to rank caches and to satisfy its own traffic engineering objectives. The ISP supervisor then returns this ranking to the CDN supervisor. This approach is comparable to the one proposed in [1].

V. ADDITIONAL APPLICATIONS

In this section we describe two additional application scenarios where mPlane can provide significant improvements.

A. Multi-ISP SLA Monitoring

The current economic situation and the services-guided Internet ecosystem is pushing ISPs to further interact and jointly look for new business models that provide higher revenues out of their networks. One of this use-cases is represented by large-scale, end-to-end service differentiation. Consider, for example, an international bank with a presence in multiple countries, that needs to interconnect these sites. For security, it opts for a VPN-based solution spanning the networks of the ISPs providing the service at each site; this VPN has additional quality of service (QoS) requirements expressed by end-to-end Service Level Agreement (SLA).

The monitoring of an end-to-end SLA for this QoS requirement, and how to pinpoint the ISP responsible for a violation, is an open question to which we propose to apply mPlane. In this case, probes within each ISP would verify the status and QoS parameters of the portion of the monitored network, which would be reported to a supervisor at each ISP. The end-customer’s ISP would then query each ISP for these parameters, correlating them with timing information about perceived quality of experience (QoE) problems.

To have a common view of the overall service performance, the report issued by the end-customer’s supervisor is periodically forwarded to each of the involved ISPs’ supervisors. Internet eXchange Point (IXPs) can be easily included in the picture. By combining the active, passive and even hybrid measurements with the aforementioned SLA verification reports, the end-customer is able to monitor and verify the established SLA, and the ISPs are able to detect which of them is not fulfilling it in case of degradation.

B. Mobile Video Streaming Troubleshooting

The use of video-on-demand on mobile devices is increasing, resulting in significant strain on mobile operators, which often results in degraded performance. Troubleshooting performance issues in video streaming services is not a trivial task, given the wide variety of possible causes. From the end-user side, the user’s device might not be able to correctly load and display the video, for example due to missing codecs, poor CPU, poor storage capacity, etc. In addition, the user might be in an area with poor cellular reception. From the mobile provider perspective, the mobile network might be congested, either at the access or the backbone. Core network misconfiguration may also cause problems for a seemingly random group of users. Finally, the problem may reside within the CDN used by the service itself.

mPlane distributed architecture spanning different domains can be well exploited to track down such issues. Statistical analysis of software probes primitive measurements (mass-deployed on end-user devices) helps to identify the set of users or regions with problems. These measurements include metrics that are relatively easy to obtain from end-user devices without imposing heavy additional load (e.g., signal strength, OS-level network statistics counters, reconnect events, etc.).

Once the impacted parts of the network topology are found, active probes in the operator network perform targeted measurements, e.g., downloading “problematic” videos while performing fine-grained analysis on network transport characteristics. At the same time, passive upstream flow analyzers focus on these streams (e.g., by address), and provide detailed monitoring data. Border probes can also be utilized to find out whether stream degradation happened within, or outside of the provider’s network. For a given set of parameters describing a problem, the operator’s reasoner can find the most-likely cause among the aforementioned ones.

VI. CONCLUSIONS

To shed light on the Internet operational obscurity, we advocate the development of a measurement plane alongside the data, control and management planes: efforts carried out in the mPlane project point precisely in this direction.

Rather than proposing yet another clean-slate approach, bound to limited adoption and hence likely to fail, the mPlane architecture allows maximum reuse of existing measurement tools. At the same time, building on existing tools do not hinder development of new ones, as mPlane dynamically supports new capabilities.

The mPlane supervisor offers a flexible control interface, with a focus on automated, iterative measurement that makes the platform well-suited to troubleshooting support. Raw measured data is just the starting point of mPlane. Its core inner-working are represented by repositories, that store, filter and process the measurements data; and by a reasoner, which iteratively drills down to the root cause of a problem.

The mPlane platform was developed with the tiered structure of the Internet in mind, and makes it easy for complex measurements tasks to span multiple administrative domains. Indeed, as measurements are already coordinated by an mPlane

supervisor in a single administrative entity, supervisor-to-supervisor communication transparently allows measurement to extend between neighboring mPlane domains. The platform's flexibility allows the implementation of multiple solutions depending on the constraints and specific needs of the different interacting players, encompassing and extending the solution space with respect to the current state of the art.

Finally, as shown by the mPlane use-cases shortly presented in this paper, the mPlane architecture not only allows multiple solutions from a technical standpoint, but also potentially introduce new viable business models in the Internet ecosystem.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627 (Integrated Project "mPlane").

REFERENCES

- [1] I. Poese et al., "Enabling Content-aware Traffic Engineering", in *ACM SIGCOMM Computer Communication Review*, 2012.
- [2] A. Hanemann et al., "PerfSONAR: A Service Oriented Architecture for Multi-Domain Network Monitoring", in *ICSOC*, LNCS 3826, pp.241–254, 2005.
- [3] R. Alimi et al., "ALTO Protocol draft-ietf-alto-protocol-13.txt", in *IETF Internet draft*, 2012.
- [4] W. Jiang et al., "Cooperative Content Distribution and Traffic Engineering in an ISP Network", in *ACM SIGMETRICS*, 2009.
- [5] P. Xia et al., "Distributed Joint Optimization of Traffic Engineering and Server Selection", in *IEEE Packet Video Workshop*, 2010.
- [6] A. Finamore et al., "Experiences of Internet Traffic Monitoring with Tstat", in *IEEE Network*, 2011.
- [7] A. Bär et al., "Continuous Analytics for Traffic Monitoring and Applications to CDN", FTW-TECHREPORT-129, 2013.
- [8] I. Bermudez et al., "DNS to the Rescue: Discerning Content and Services in a Tangled Web", in *ACM IMC*, 2012.

VII. BIOGRAPHIES

Brian Trammell is a Senior Researcher at ETH Zurich's Communication Systems Group, where his primary research interests are Internet measurement, performance, and security. He is a member of the Internet Architecture Board, co-chair of the Internet Engineering Task Force's IP Performance Metrics working group, and an active contributor to measurement-related standards in the IETF. He leads the mPlane architecture effort.

Pedro Casas is a Senior Researcher at FTW. He got a Ph.D. in computer science from Telecom Bretagne in 2010, and joined CNRS as a research fellow between 2010 and 2011. His research interests span the analysis of network traffic, QoE modeling and assessment, and data mining in networking. He has co-authored more than 50 papers (40 as first author) in major conferences and journals, and received 6 best paper awards in the last 5 years.

Dario Rossi received his PhD (2005) degree from Politecnico di Torino and his HDR (2010) degree from Université Pierre et Marie Curie (UPMC), and is currently a Professor at Telecom ParisTech and Ecole Polytechnique. He has co-authored over 100 papers in leading conferences and journals, holds 6 patents and he participated in the program committees of over 40 conferences. His research interests include Internet traffic measurement, information centric networks, green networking and peer-2-peer networks.

Arian Bär is a PhD candidate in Computer Science in the second year at FTW in Vienna. He received his Diploma degree in Computer Science from the Friedrich-Alexander Universität Erlangen-Nürnberg in 2009. His PhD topic is about the application of data base approaches to big and fast data streams common in network monitoring environments. His research interests include network monitoring and analytics, data stream warehousing, query scheduling, data mining and machine learning.

Zied Ben Houidi, Ph.D, is a researcher at Bell Labs France since 2011, working on how to enrich the network with added-value services and functionalities. Prior to joining Bell Labs, he worked at Orange Labs on management and routing in core networks, where his work on the Border Gateway Protocol (BGP) routing table transfers has led to significant improvements to BGP implementations (one to two orders of magnitude faster transfers).

Ilias Leontiadis is currently a Research Associate at Telefonica Research. In the past he was a researcher at University of Cambridge and received his PhD from University College London (UCL). His research interests include mobile systems, pervasive computing, wireless networks, sensor networks, privacy and mobility modelling.

Tivadar Szemethy received his M.Sc. from Budapest University of Technology and his PhD from Vanderbilt University, Nashville, TN, USA. After working with academic research institutions in Hungary and in the USA, he joined NETvisor Plc, Hungary as Chief Architect, focusing on network and systems operation, management, monitoring and measurements products and research.

Marco Mellia (SM'08), Ph.D., is the coordinator of the mPlane Integrated Project that focuses on building an Intelli-

gent Measurement Plane for Future Network and Application Management. His research interests are in the design of energy efficient networks (green networks), in the area of traffic monitoring and analysis, and in cyber monitoring in general. He has co-authored over 200 papers published in international journals and presented in leading international conferences.

VIII. FIGURES AND TABLES (SEPARATED)

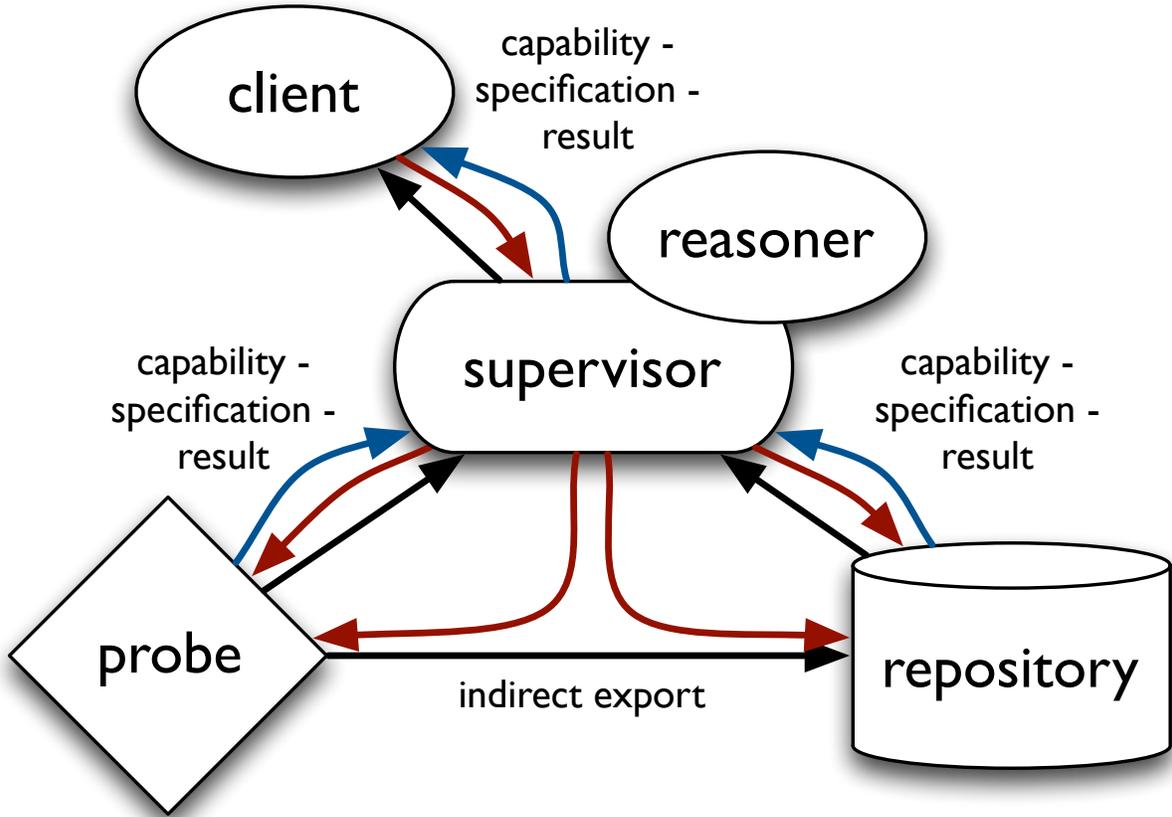


Figure 1. Components and interactions in the mPlane architecture. Blue lines are capabilities announcements, red lines indicate control messages, and black lines correspond to data flows.

Table I
NOTIONAL EXAMPLE OF CAPABILITY, SPECIFICATION AND RESULT FOR A SIMPLE ICMP PING-LIKE ACTIVE MEASUREMENT PROBE.

capability:measure	specification:measure	result:measure
parameters: start: now...+inf end: now...+inf source.ip4: 192.0.2.3 destination.ip4: * octets.count: 28...65535 period.s: * results: - delay.twoway.icmp.ms.min - delay.twoway.icmp.ms.mean - delay.twoway.icmp.ms.max - delay.twoway.icmp.ms.count	parameters: start: now end: now + 30s source.ip4: 192.0.2.3 destination.ip4: 192.0.2.67 octets.count: 80 period.s: 1 results: - delay.twoway.icmp.ms.min - delay.twoway.icmp.ms.mean - delay.twoway.icmp.ms.max - delay.twoway.icmp.ms.count	parameters: start: 2013-11-27 09:31:18 end: 2013-11-27 09:31:48 source.ip4: 192.0.2.3 destination.ip4: 192.0.2.67 octets.count: 80 period.s: 1 results → values: - delay.twoway.icmp.ms.min → 31 - delay.twoway.icmp.ms.mean → 37 - delay.twoway.icmp.ms.max → 92 - delay.twoway.icmp.ms.count → 30

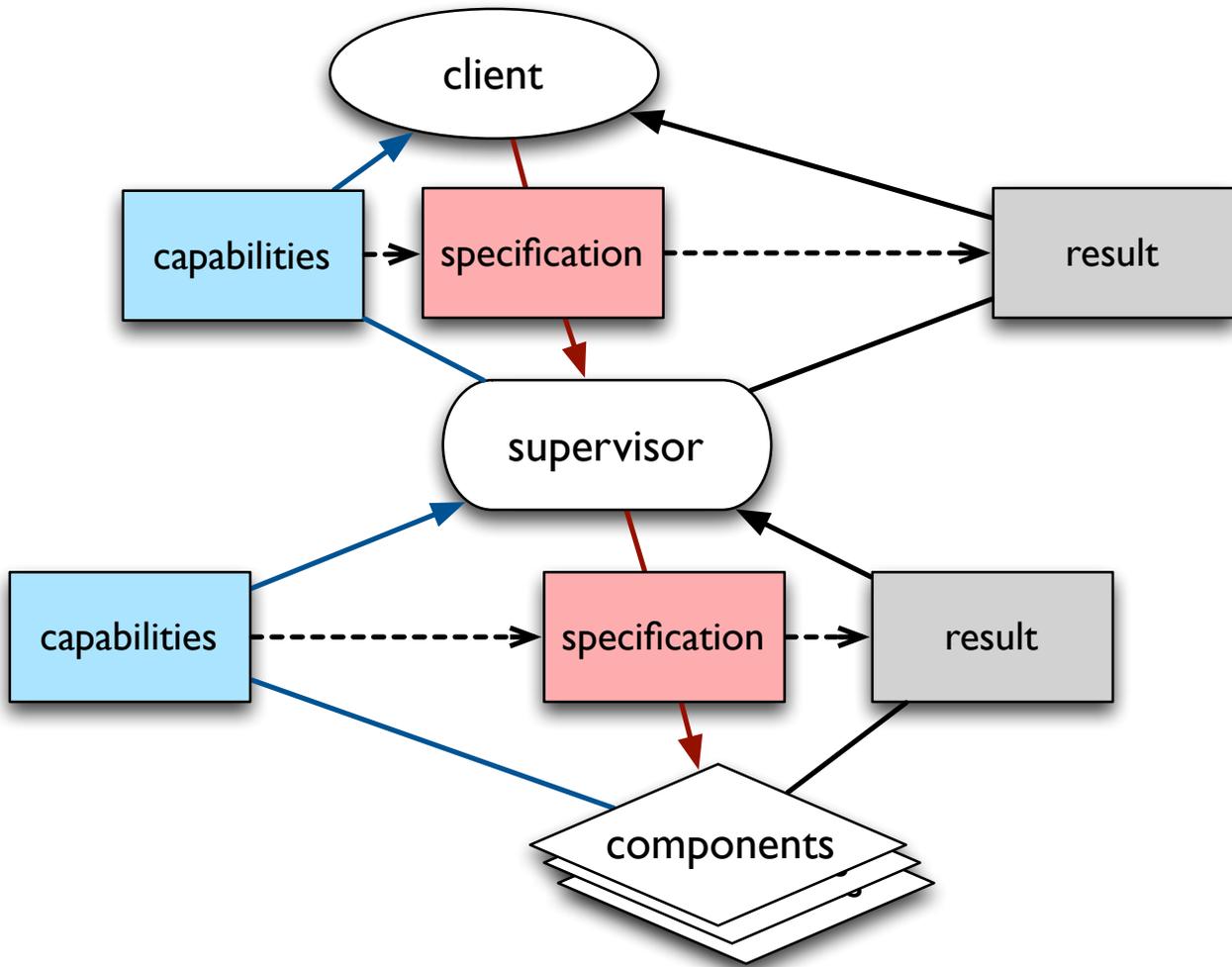


Figure 2. The mPlane workflow. *Capabilities* define the tasks a component can perform. *Specifications* consist of a description of which measurement have to be performed, how, and when.

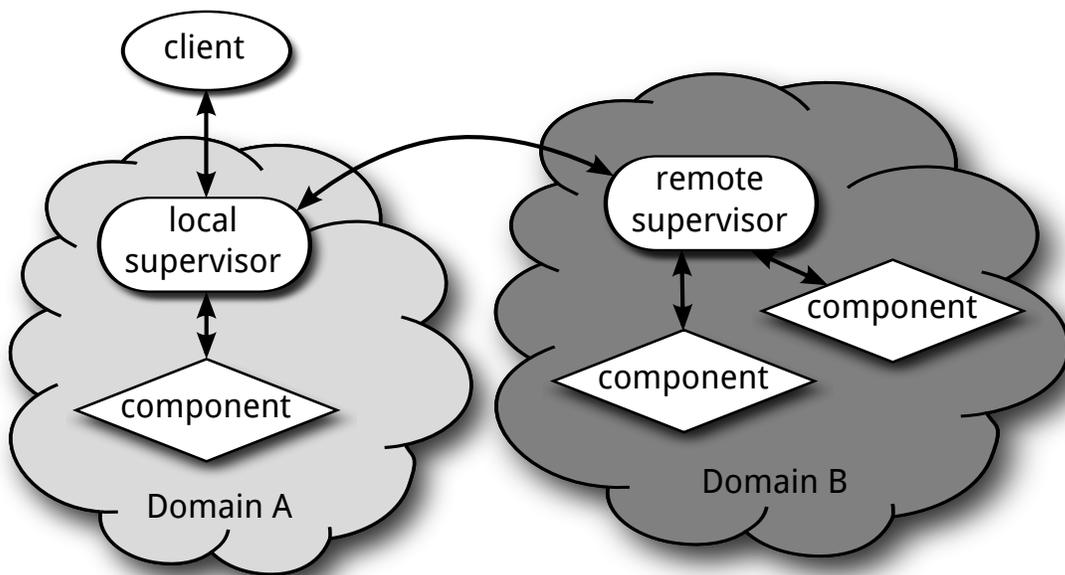


Figure 3. Federation in mPlane through inter-supervisor connections. Supervisors in each domain handle supervisors in external domains as clients.

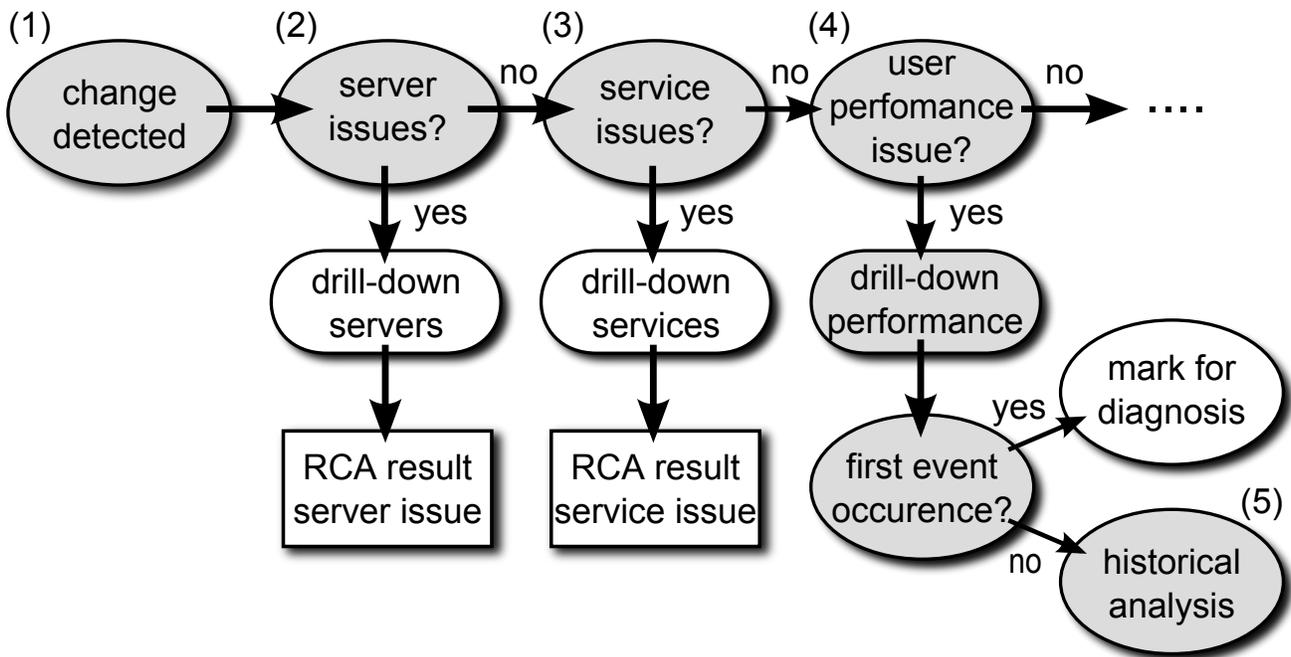


Figure 4. A rule-based reasoning approach for CDN cache selection analysis.

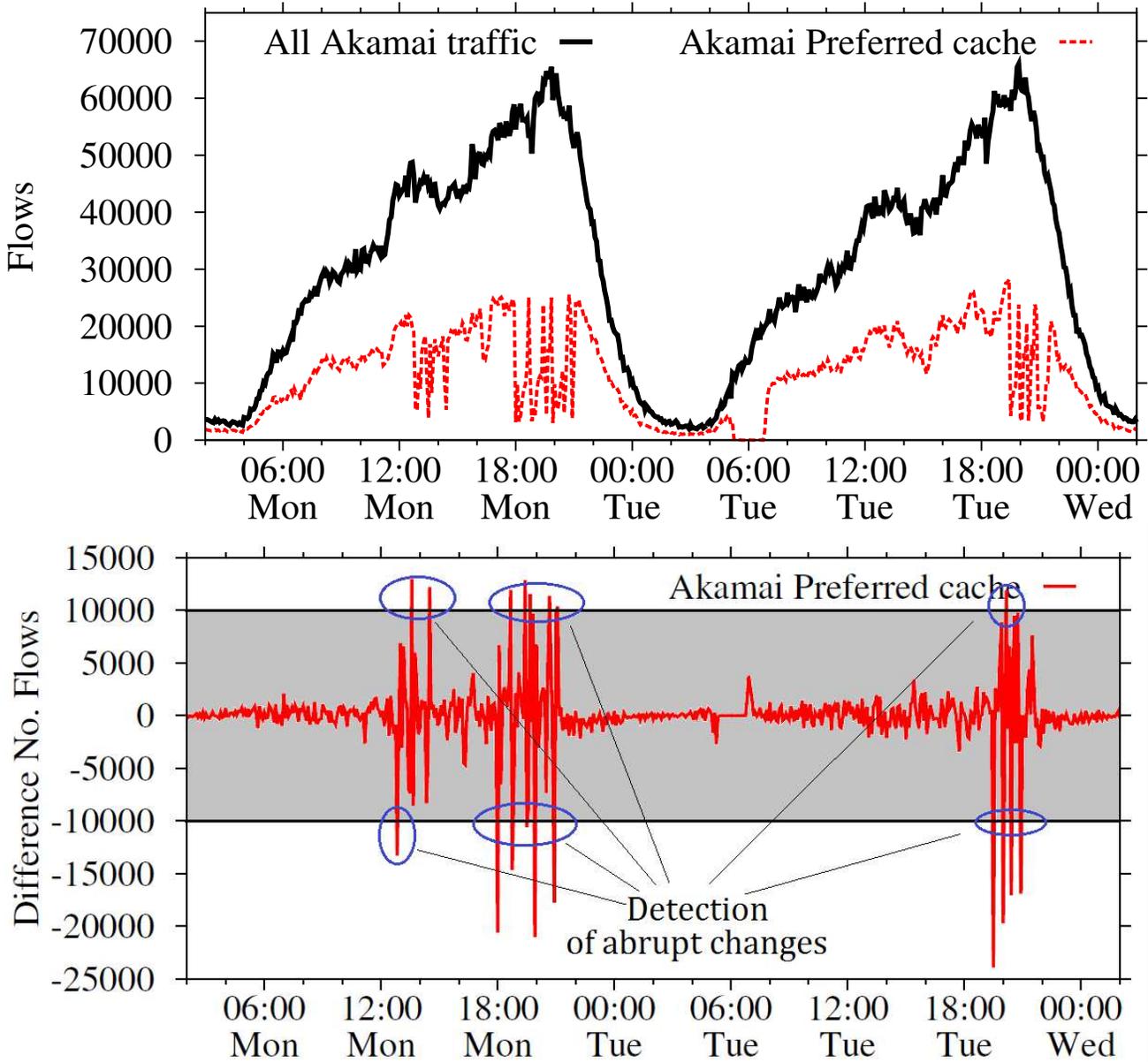


Figure 5. Evolution of number of flows served by Akamai CDN (top) and difference of number of flows served in consecutive 5 minutes time windows (bottom).

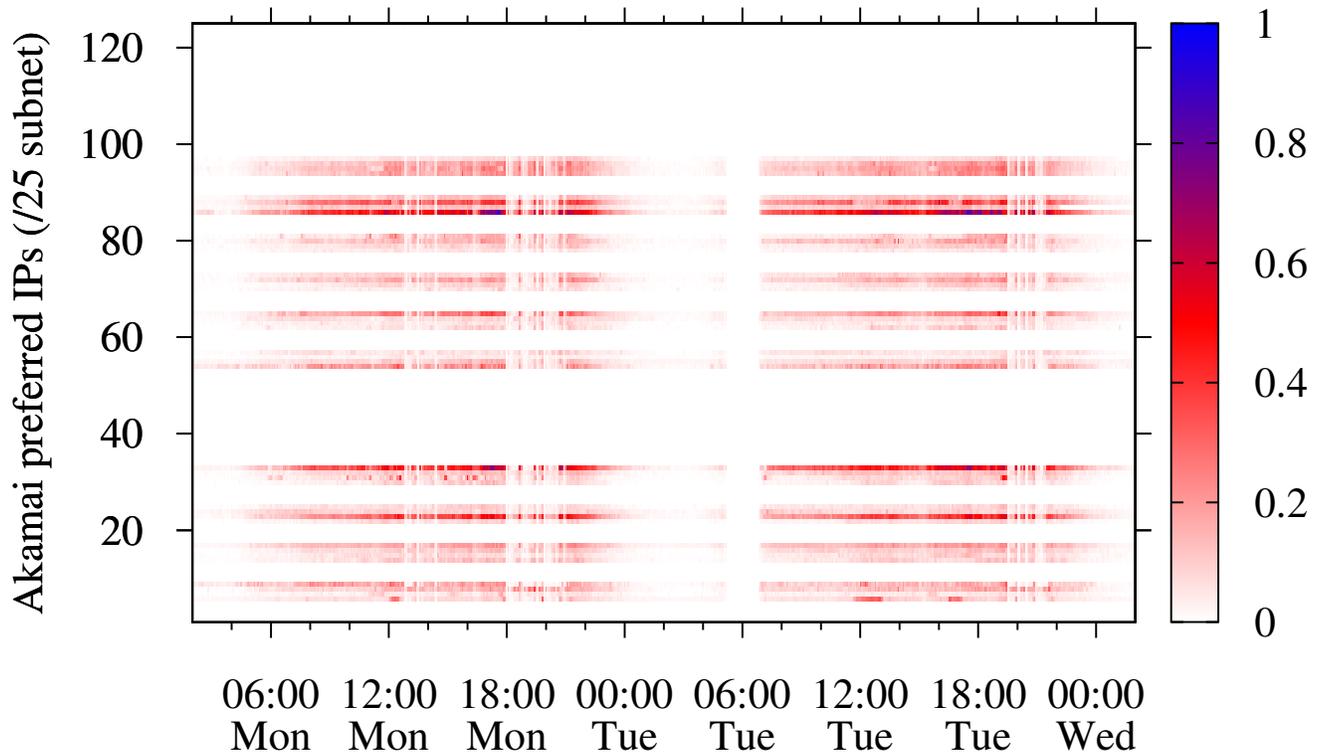


Figure 6. (a) Evolution of traffic load for each IP in the preferred cache.

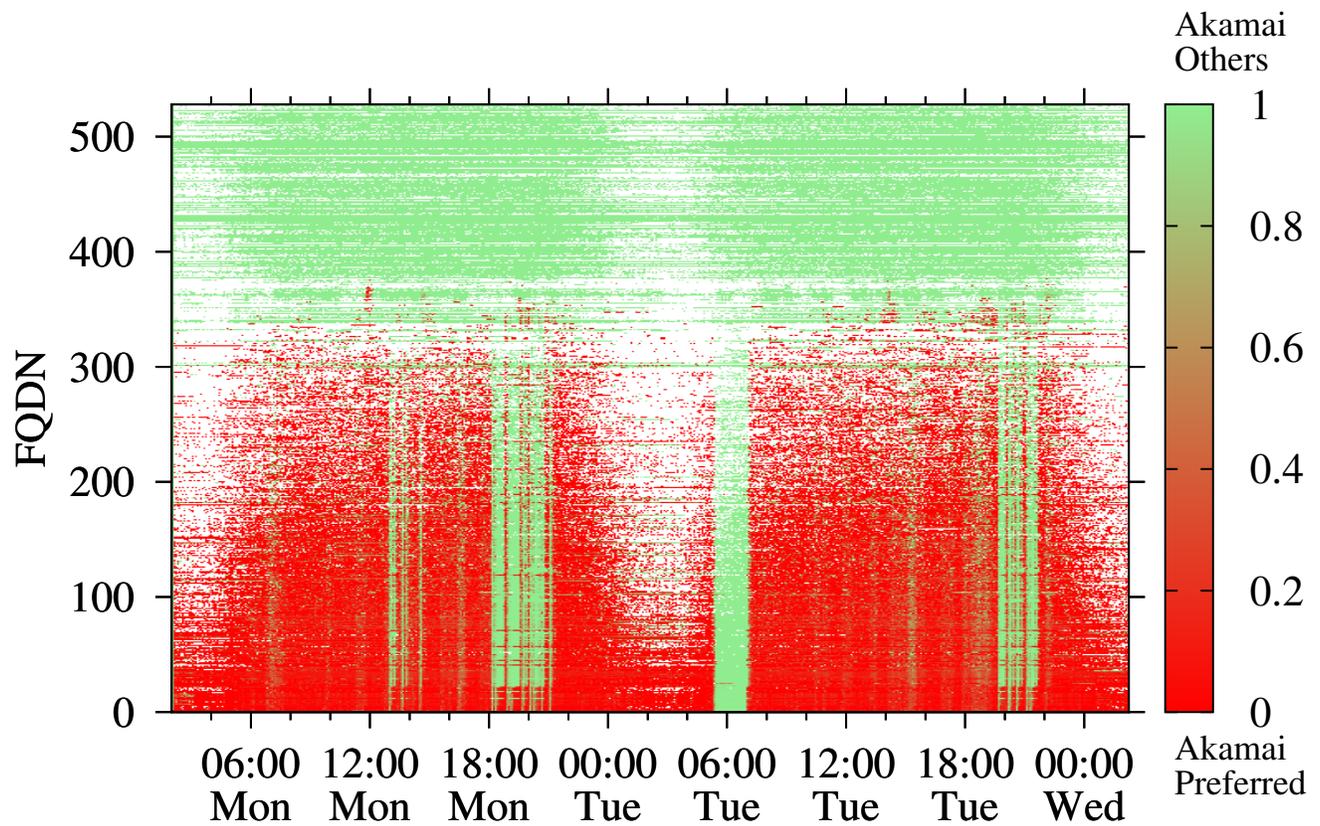


Figure 6. (b) Evolution of the volume of requests per service name.

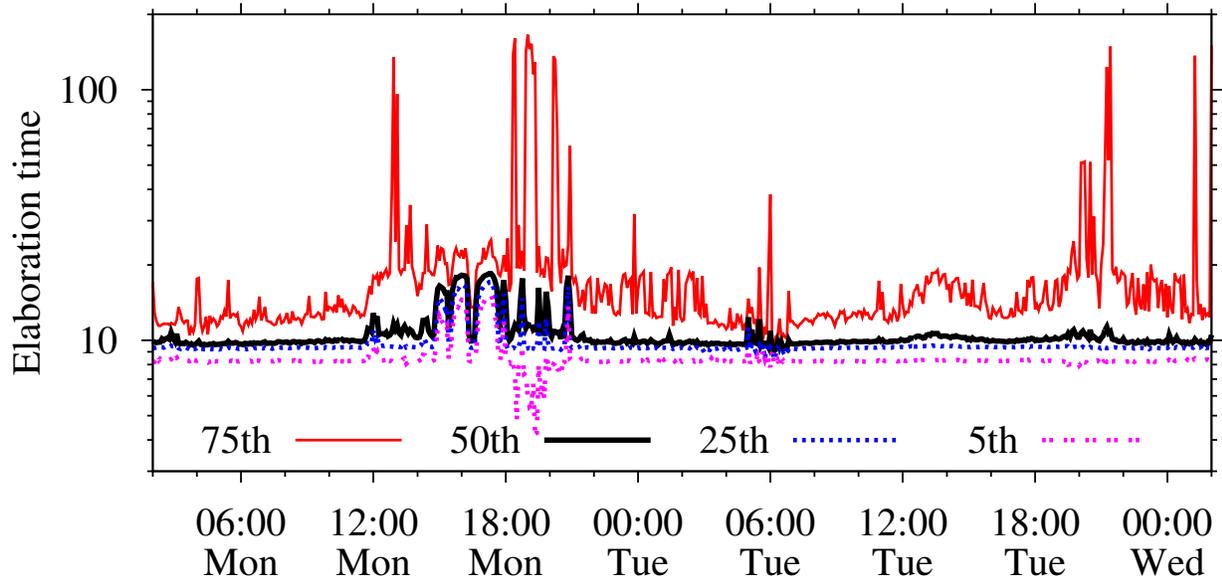


Figure 6. (c) Evolution of server elaboration time percentiles for the preferred cache.

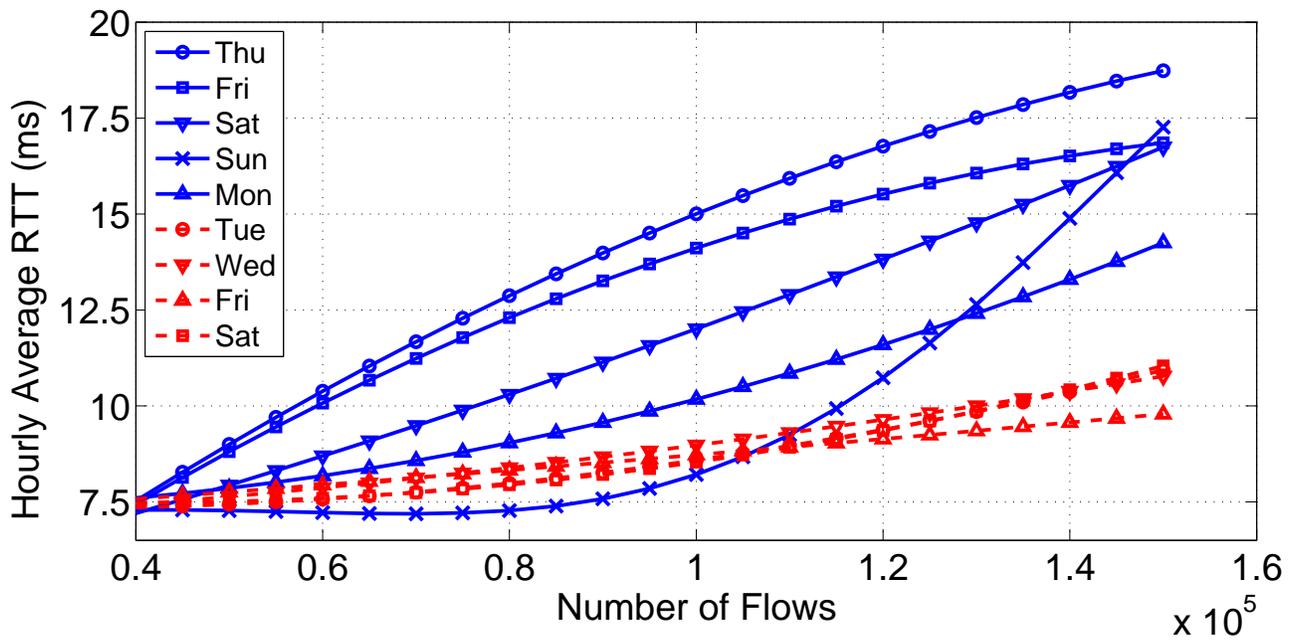


Figure 6. (d) Daily average RTT vs flow counts for 9 days.