Traffic analysis methods to support decisions at the knowledge plane

PÁL VARGA
BME, Department of Telecommunications and Media Informatics
pvarga@tmit.bme.hu

LÁSZLÓ GULYÁS
AITIA International Inc.
lgulyas@aitia.ai

Keywords: knowledge plane, monitor plane, traffic mix, traffic matrix

Traffic analysis of network segments is an effective method to reveal suboptimal configuration, hidden faults and security threats. If the analysis results are promptly acted upon, improvements in service quality are experienced by both the network operator and the end user. The concept of the Knowledge Plane (KPlane), and later the Monitor Plane (MPlane) has been introduced to support Autonomous Networking goals. The tasks of processing the network element, service and traffic information belong to the MPlane. It feeds the KPlane with valuable information, based on which configuration changes are actuated. Although the concept of KPlane is widely used in various levels of network and service management, general traffic analysis is not yet utilized to support decision making procedures. Traffic mix and traffic matrix analysis results are of major interest in the decision making process at the KPlane. In this paper the issues of traffic sensing at the high speed interfaces of the Monitoring Plane are covered, followed by a discussion on traffic mix and traffic matrix analysis methods.

1. Introduction

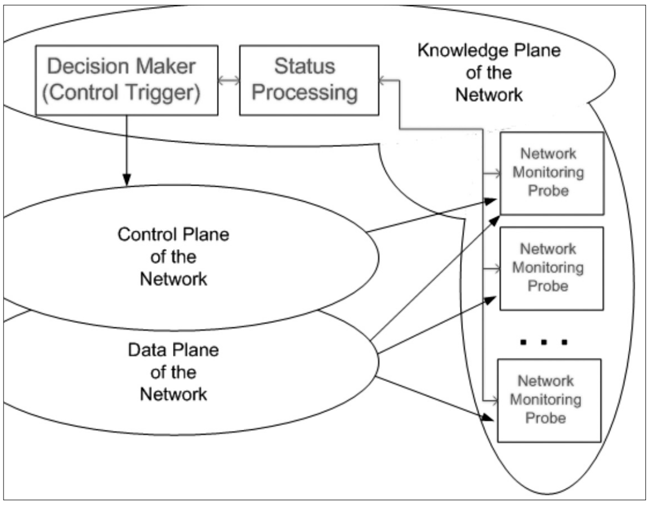
The optimization of network and service resources and the maximization of end-user experience are not necessarily conflicting terms. The reason for such belief lies in the fact that current network operators and service providers lack of up-to-date, usable information on their traffic. The questions of “how much” of “what” actually are traversed on the various network segments, where is that traffic “originated from” and where is it “distributed toward” are rarely answered.

According to the main argument of [1], the users and the operators suffer from the lack of a serious, purposeful optimization effort in the traditional Internet. The transparent core has no knowledge about the data transported, and even if the intelligent edge nodes realize that there is a problem, the core might not be aware of what should be done. The low-level decisions (at the edge) are rarely relate to the higher-level goal (of the core). On the user side this results in meeting the service level agreement only in coarse granularity: it is measured in long periods and more at a network level, rather than on a per-service basis.

The solution for gaining knowledge about network status and traffic characteristics is to gather and process such data, which then provide a basis to trigger corrective actions. The authors of [1] suggest to handle this knowledge in the Knowledge Plane (KP), an abstract entity that completes a triad together with Data Plane and Control Plane (see *Fig. 1)*.

In the original KPlane concept, the input is taken by *sensors* and the output is given by *actuators*. A practical variation of this architecture, detailed in [2], splits the KPlane into *monitoring plane* and *knowledge plane*. The separation of those is an obvious step: the actual “network monitoring units” (sensors) that capture and pre-process traffic data represent the “monitoring plane”, similarly as depicted in Fig. 1. There are further variations and additions to this architecture; we will review these in the section of Related Works, together with a short review of decision making methodologies and practical examples from the field. Fig. 1 depicts the relation between the Knowledge, Control, and Data Planes. The probes/sensors take data from both the control and data planes, and report pre-processed information for the status processing module, where further analysis takes place. The actuator in the model is the de-

Figure 1. Functions of the Knowledge Plane and its connections to Control and Data Planes


Decision Maker (Control Trigger)
Status Processing
Knowledge Plane of the Network
Network Monitoring Probe
Control Plane of the Network
Network Monitoring Probe
Data Plane of the Network
. . .
Network Monitoring Probe

cision maker module, which provides triggers for the control plane, completing the self-management cycle.

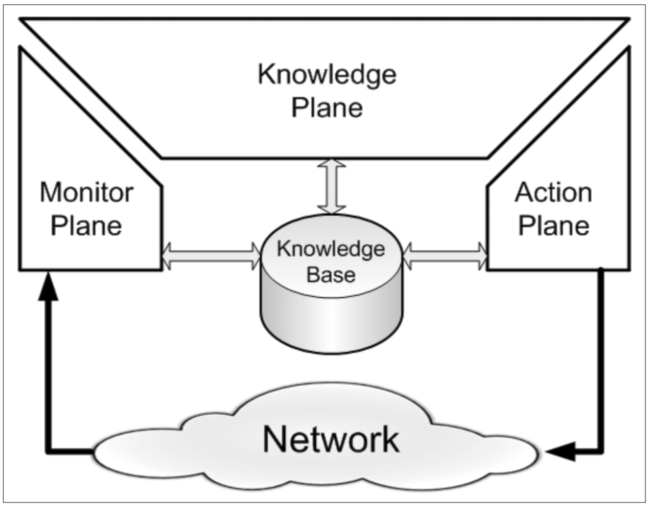
The main source of "knowledge" is the actual traffic of the Control and Data Planes. Although some traffic characteristics can be gathered by analyzing the Control Plane messages, many important applications – such as peer-to-peer (P2P) downloads, video streaming, or interactive voice – hide their control messages, hence their identification is only possible through Deep Packet Inspection (DPI) of the traversed traffic. The aim of Traffic Mix analysis is to determine the distribution of volumes for services and applications utilizing the network. Similarly, Traffic Matrix analysis provides results about traffic volumes – and if possible, further characteristics – broken down by route directions. The second part of this paper discusses our proposed, unique method of Traffic Mix and Traffic Matrix analysis.

2. Related work

Kim et al. summarize the research and development ideas and efforts in management of the Future Internet in [4], specifically reviewing the research activity in the EU, USA and Korea. The authors emphasize the common interest and importance of measurements, monitoring, knowledge representation and reasoning. The original idea of introducing a higher level intelligence to the core about its traffic and general status first appeared in [1], where Clark et al. introduced the concept of the Knowledge Plane. Besides providing very clear motivations, this groundbreaking paper suggested to solve networking issues by using methods devised in the field of Artificial Intelligence (AI). Since then, experts of both area – Network Management, and AI – elaborated various versions of the KPlane concept in great depth.

Li described a layered architecture in [3], where NetKP – the network layer – organizes agents to gather and provide valuable information to the higher-level entities, specKPs, which handle and act upon their own interest, i. e., routing optimization or intrusion detection.

Figure 2.
The original Knowledge Plane concept extended with Monitor and Action Planes



Another variant of splitting is suggested by [6], motivated by the need to get to the kernel of self-functions defined by autonomous networking drives. Hence, the processes in KPlane are based on two loops: a collaborative loop and an adaptation loop. The KPlane itself includes a knowledge base, a reasoning engine, a knowledge sharing process, and a machine learning process. In this model, Monitoring functions remain outside the KPlane.

Dietterich et al. found that the application of distributed, model-based reasoning agents is a feasible and successful approach for certain fault diagnostics tasks that involve the KPlane. In their report (see [5]) one of the main motivations was to involve Machine Learning in KPlane. Although their findings show that these methods can contribute to the KPlane, they do not suggest to have machine learning as a key element of KPlane. Their paper also includes interesting reports on fault detection case studies, including DNS diagnosis, and a scenario where a typo in BGP (Border Gateway Protocol) tables was revealed.

The IST-MUSE project resulted in many ideas and implementations in relation to KPlane. Besides separating the Monitoring Plane from the KPlane in [2,7, and 8], they further introduced the Action Plane (APlane). They also defined a knowledge base that is commonly reachable by KPlane, MPlane and APlane. *Fig. 2* depicts their connection and relation to the network. The main motivation in these papers is to eliminate QoS (Quality of Service) and QoE (Quality of Experience) issues in the access network for VoIP, IPTV and other multimedia services. Instead of gathering knowledge from overall data plane traffic, these papers rely on designated protocols (i.e. RTP, Real-time Transport Protocol) and protocol analysis of the control messages.

The Monitor Plane is extensively used in [9] as well, where a complete, "access control list"-based VoIP service management system is described and evaluated. The KPlane in this paper is put in a different context: its functionalities include Call Data Record generation and visualization.

Although KPlane was not mentioned in [11] all of its features appear in the service management framework described in the paper: measurements, monitoring, data processing/mining, decision making, knowledge bases and machine learning. The presented framework has been effectively used for fault detection and elimination for Ethernet services and for VoIP services [11] as well.

A specialized KPlane is suggested in [12] in order to handle current QoS problems with protection routing algorithms in GMPLS over WDM (Generalized Multiprotocol Label Switching over Wavelength Division Multiplexing) networks. This is a clear example of using a variation of the KPlane concept to enhance concrete routing methods' speed and effectiveness.

It is clear that the concept of Knowledge Plane is widely used in various levels of network and service management. Nevertheless, general traffic analysis is not yet utilized in order to support decision making in the KPlane. In the following sections we describe the suggested management architecture, traffic analysis concept and two methods to extract valuable information about the traffic mix and the traffic matrix.

3. The Monitor Plane

In this paper we follow the architecture suggested in [8] (see Fig. 2), and closely examine the functions and requirements of the Monitor Plane. This function is crystallized at the original definition of autonomous networks, in [13], defining the foursome of "Monitor-Analyze-Plan-Execute" (MAPE) functions. The core function of the MPlane is to provide complete and detailed view of the network and its services. Probes at every element (access nodes, routers, switches, content servers, links, etc.) monitor the element status as well as traffic parameters.

Although built-in probe modules seem convenient, passive probing is more desirable. Active network elements (such as routers or switches) keep their processing priorities to their main job, occasionally leaving the Knowledge Base without information. These occasions of degradation in the status reporting function happen at the worst time from the KPlane's point-of-view - for practical reasons. It gets degraded at the time when the element is getting overloaded. Coincidentally, such detailed reports of overloading would be the most beneficial for the KPlane. This is why passive probing is more desirable to gather information on these elements.

After capturing the raw data, processed, grouped, and filtered traffic information gets inserted into the Knowledge Base by the probes. Both packet- and flow-level analysis reveal important characteristics on losses, delays, and jitters in the traffic, routing specialties, network structure changes and violations of the SLS (Service-Level Specification).

We are focusing on gathering these characteristics by passive monitoring. In the following subsections we briefly describe the basic requirements and mechanisms enabling this method.

3.1. Basic functions of the probes

The inevitable function of the network monitoring probes is catching, filtering, and preprocessing the traffic. These tasks should be completed for the whole network. Since installing and maintaining such a monitoring network could be an enormous effort for the operator, introducing the MPlane at the highest aggregation parts (i.e. monitoring the fastest links) can be a good decision. Monitoring these relatively few points allows gathering all packets that traverse the network, although some locally looping traffic could be left out of the analysis.

The probes should have the following crucial functionalities:

• *Creating timestamps for the packets.* Time-stamping done by hardware (firmware) facilitates much more precision than by software, since it avoids possible latencies due to the operating system.

• *Filtering on hardware level.* High-speed traffic (i.e. currently 10 Gbps or above) presently allows no option for on-the-fly filtering in software. Clearly defined, low level filters are very useful: they can dramatically decrease the data to be analyzed.

• *Truncating incoming packets.* For the majority of the network analysis functions, statistics-counting, or fingerprint analysis, it is not necessary to use the whole IP packet, only the first portion of it. A practical example is truncating at 128 bytes, which keeps TCP and IP headers as well as the beginning part of application headers that are helpful for identification, since it contains fingerprints for P2P or video.

• *Traffic processing.* The main traffic processing functionalities are briefed in the next section.

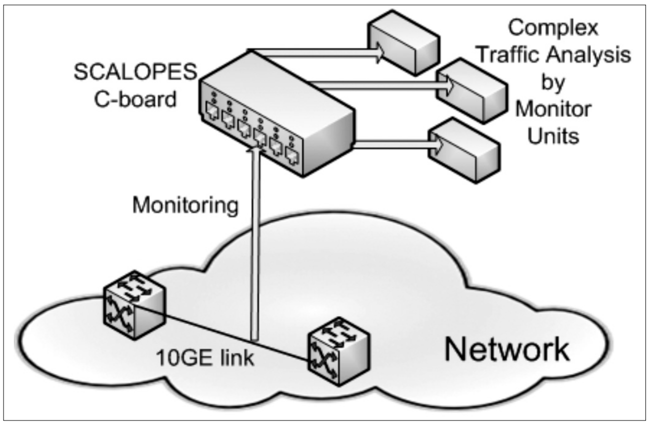
• *Encapsulation and presentation of preprocessed data.* The traffic analysis results must be structured and packed when passed over to the Knowledge Base.

3.2. Traffic processing

The time-stamped, filtered, truncated packets must be processed in order to reveal network and service statuses. Depending on the traffic volume, and the depth of the analysis, this processing can be fed into one or many processors. In order to keep up with the ever increasing traffic and the demand for complex analysis, the processing system must be highly scalable. As discussed earlier, monitoring core links has the advantage of utilizing all through-traffic (that traverses the network), although it requires equipment being able to monitor high-speed (currently 10 Gbps Ethernet) links without frame loss.

For low analysis demand (when one CPU can deal with the challenges), a highly reliable monitoring card, such as SGA10GED can be used to capture the traffic. (This card has been developed as part of the CELTIC TIGER2 project, partially funding our research.) It fits into a PCI slot of an industrial grade PC, where it captures, timestamps, filters, and truncates packets before passing it to the main CPU where Traffic Analysis is performed.

Figure 3. A scalable solution for Traffic Analysis of high-speed network links



In cases where on-the-fly, complex analysis is required on highly utilized links, the SCALOPES C-board is a highly scalable solution. (The C-board has been developed as part of the ARTEMIS SCALOPES project, partially funding our research.). It is a standalone, FPGA-based hardware, equipped with 2x10 Gbps Ethernet interfaces and 16x1 Gbps Ethernet interfaces. When used as part of the Monitor Plane, it is also preprocessing the packets, but rather than passing their data to one CPU, it distributes them among many monitor units through its 1 Gbps Ethernet Interface. The standalone Monitor Units then carry out traffic analysis, and present the results to the knowledge base. *Fig. 3* depicts such a scenario. Detailed description of this system can be found in [14].

The distinct analysis tasks – such as flow separation, application identification, QoS-related parameter calculation per flow/application/route – are managed by separated modules, so the parallel tasks can be run on distinct processors in the same time. Moreover, the inactive modules can be turned off to save power.

The tasks of the monitor units in this architecture are the following:

- collect and decode all the incoming information continuously (in 7/24 manner),
- check filtering rules predefined by the network operator, execute conditional controlled orders/ commands (conditional packet saving, alarming),
- structured data storage (raw data, statistics, assays, alerts)
- generation of packet- and flow-level counters on volume, loss, delay, jitter
- generation of specialized traffic reports, such as traffic mix and traffic matrix
- database handling, remote access/query (Remote Capture, Session/Flow Trace)

4. Methods for retrieving traffic-specific knowledge

4.1. Traffic Matrix calculations

Traffic Matrix is a network planning and development tool. During Traffic Matrix analysis, basic QoS statistics are periodically created on flow-level, and matched to originating and destination routes, network segments, or endpoint pairs (such as IP address(-range) pairs, MPLS tunnel endpoints, etc.). The first step of the analysis is determining the flows by an n-tuple (i.e., “5-tuple”: from-IP, to-IP, from-port, to-port, protocol), and building/refreshing the flow-database. Once the targeted data structure is clarified, the algorithms of Traffic Matrix calculation are of low complexity. Such algorithms are described in [15]. The result of the measurement can be used to display periodical statistics that support network planning or service marketing activities.

The actual Traffic Matrix can easily contain endpoint-pairs in the magnitude of 105. It is challenging to display such huge amount of data in a way that humans understand. While the raw results should be made available for reference in the Knowledge Base, some kind of data grouping should also be applied for visual presentation. One example of a good solution is to group the matrix elements into network segments, based on their destination addresses. The aim of the grouping algorithm is never to display high, invisible amount of segments (e.g. more than 15). When the operator wishes to peek inside a segment’s statistics, he/she get it displayed as a deeper layer of the matrix.

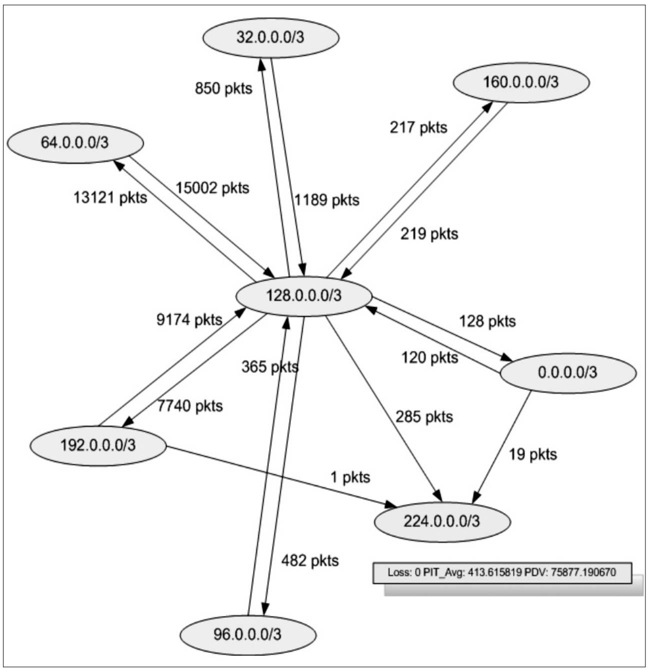
This way the calculated QoS parameters show up in an aggregated manner in the segment-to-segment relation. If the system allows manual definition of segment-creation rules, operators can gather valuable information by grouping their endpoints into various segments. An example screenshot from a solution integrated in our system is shown in *Fig. 4*.

4.2. Traffic Mix Statistics

Traffic mix analysis is the classification of traffic flows into application types, and then evaluating these for the service parameters important for the given application type. Flows are classified by means of statistical indicators and, if necessary, behavior heuristics. The most important flow types include video stream, video conference, or simple download of videos, audio stream, VoIP, and peer-2-peer.

An application belonging to a traffic-class can be identified by using static identifiers (e.g. port-based), dynamic identifiers (e.g. changing ports, fingerprints) or by applying packet-level statistics-based evaluation methods (i.e., Naive Bayes). Powerful identification methods for VoIP, video and p2p applications are described in [17-19] respectively. We used these methods successfully during the CELTIC TIGER2 project – see [20] for details.

Figure 4.
Screenshot of a Traffic Matrix visualization application [16]


Loss: 0 PIT_Avg: 413.615819 PDV: 75877.190670

Once a traffic flow is identified (i.e., based on 5-tuple), various metrics are calculated in order to help identifying the traffic-class. These metrics are the following:

– throughput: transferred data bytes per second,
– packet loss: the rate of received packets and total transmitted packets in a given time interval, or during the connection,
– packet delay: depending on the network topology and link load it takes a certain amount of time to receive a packet after it was sent; there is also a gap (delay) between packets on the wire,
– jitter: network load is not always static: as conditions and usage changes over time, packet delay changes as well – this is called jitter,
– round-trip-time: interactive applications require fast replies, which can be characterized with this parameter,
– out of order/duplicated packets.

Fig. 5 depicts a partial result of one of our measurements at a major ISP. It visualizes the number of parallel VoIP sessions (upper diagram) and the traffic volume (in kbps). The different kind of VoIP traffic are represented with different colors, which are – from bottom to top – a) Skype over UDP, end-to-end; b) Skype over UDP, end-to-office; c) other type of VoIP, d) Skype over TCP.

5. Decision Making

Since processing of network status is continuous at the KPlane, and faults/attacks may happen at any time, so decisions on corrective actions have to be made on-the-fly as well. The Action Plane should be notified (instructed) about these actions for execution. Although the accuracy of decisionmaking process is the key, it is limited by the variety of the input information – which is in this case merely traffic-related. Beside the accuracy, speed is also a key factor.

In order to understand the complexity of the decision making problem, a short review the main challenges are necessary. Clark et al. [1] points out three significant issues that need to be addressed by the Knowledge Plane.

1. The KPlane needs to operate in the presence of incomplete and inconsistent information, with the possibility of even misleading or malicious pieces of data.

2. The KPlane needs to be able to handle conflicting or inconsistent high level goals.

3. The KPlane needs to be general and future proof, i.e., the introduction of new technologies and novel applications should be possible. Moreover, the environment in which optimization needs to take place is highly dynamic, where both short and long term changes are possible in the structure and complexity of the network system.

Such challenges are not uncommon in the research and applications of the last decades of Artificial Intelligence (AI) literature. In particular, multi-agent systems (see [21]) are often proposed to handle such challenges. A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents, where intelligent agents, shortly put, mean autonomous decision making entities with individual information processing capabilities and individual goals. Such agents can naturally incorporate different viewpoints or goals in a system and also provide a natural way to embody components with different levels of data access.

As a consequence, however, the goals and actions of agents in a multi-agent system may partially be aligned or conflicting. Also, even if conflicts are missing or resolvable, information may be unevenly distributed among the agents. Therefore, agents interact and try to resolve conflicts and collaborate according to various protocols and methods. A vast body of the recent AI and MAS literature deals with conflict management, collaboration and cooperation, and distributed optimization in such systems (see [22-24]).

It is worth pointing out that the agent metaphor is a natural abstraction layer to describe conflicting or inconsistent goals – independent of the particular problem at hand. This is also true for matters of trust (cf., malicious information). This way, these issues can be handled by general solution methods and need not be developed for each particular application domain. In other words, these challenges of the Knowledge Plane may be handled by “canned solutions” developed in other research domains.

Multi-agent systems are often said to provide a solution for the introduction of novel applications as well. The idea behind this proposal is that if a new application or requirement appears, a new agent (or bunch of new agents) may be introduced to the system at any point in time. With the general conflict resolution and collaboration protocols in place, the new goals and requirements represented by the new agents will be seamlessly integrated in the system. Similarly, should some of the goals rendered outdated by time, the sets of agents can be gracefully eliminated from the multi-agent system.

Still, in order to proceed towards a decision making solution in the autonomous networking field, further research is required. Although recent AI-related research should be exploited in the area of network management, currently there are no real-time, scalable solutions available. The canned multi-agent solutions have not yet broken into the network management field, and the few prototype systems (e.g. the one described in [25]) remained prototypes up to now.

Due to the aforementioned limiting factors, a scalable, high-performing, yet less accurate solution is suggested for decision making: rule-based reasoning. It is used with success in many areas; see [10] as an example. In connection with the KPlane, we continue future research in the AI-field, and further developments and integration toward a scalable, rule-based reasoning engine that is applied in a distributed manner throughout the KPlane.

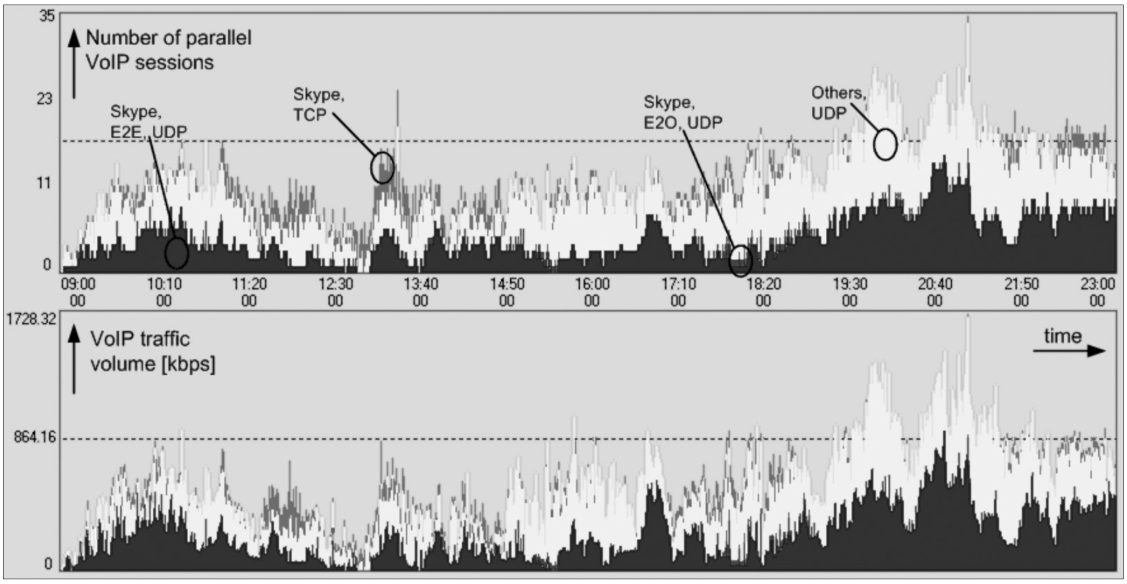


Figure 5. VoIP Portion visualization of a Traffic Mix analysis

6. Conclusions

Network efficiency and service quality are required to be kept at high standards for both the network operators' and the users' point of view. This can be achieved by keeping the network and service status under continuous monitoring. When inefficiencies become evident, or failures appear, corrective actions should be orchestrated. A recent concept to cover the autonomous loop of "Monitor-Analyze-Plan-Execute" (MAPE) is to utilize a Monitor Plane to gather and process information, introduce a Knowledge Plane to continuously process network and service status according to the requirements, and carry out commands for corrective actions by Action Plane entities.

In this paper we closely examined the tasks of the Monitor Plane, and suggested a scalable architecture to gather and process network traffic in a distributed manner. Since decisions at the Knowledge Plane should be partially made by traffic information, two important traffic analysis methods have been introduced to support decision making. Traffic Mix analysis requires a flow-based approach, where flows get classified into application types based on their characteristics, and then evaluated by related QoS metrics. Traffic Matrix analysis is important for both network and service planning, since it outputs the traffic volumes and characteristics correlated with the traffic endpoints. This information can efficiently support status processing and decision making at the KPlane, since currently these are the most sophisticated traffic-related analysis methods that human experts use during network/ service evaluation and planning.

The brief review of distributed multi-agent systems suggests that based on their problem statement, such systems – when available, – should be able to cover the requirements of the ideal KPlane. Nevertheless, this field requires further applied research, since a scalable, high-performing, tangible MAS – that could serve as a KPlane – is still missing.

Acknowledgements

This work has been partially funded in the framework of the CELTIC TIGER2 project (CP5-024) as part of the EUREKA cluster program.

Parts of the research leading to these results has also received funding from the ARTEMIS Joint Undertaking under grant agreement n° 100029 and from the Hungarian National Office for Research and Technology.

The authors would like to thank the contribution of Gyorgy Horvath (BME-TMIT), Laszlo Kovacs (AITIA) and Peter Tatai (AITIA).

Authors

PÁL VARGA is a lecturer at Department of Telecommunications and Media Informatics, at Budapest University of Technology and Economics, Hungary, where he got his MSc from. He is currently proceeding toward his PhD there. Besides, he is the head of the Telecommunications division at AITIA International Inc. His main research interests are service and network management, and traffic analysis. He focuses his research to network performance measurements, fault localization, traffic classification, end-to-end QoS and SLA issues.

LÁSZLÓ GULYÁS is a graduate of the Lóránd Eötvös University, Hungary, from where he received his PhD, MSc and BSc degrees, all in Computer Science. He is assistant professor at the Department of History and Philosophy of Science, Lóránd Eötvös University, Budapest. He is also a research partner at AITIA International Inc and a fellow at Collegium Budapest (Institute for Advanced Study). He is a member of the Scientific Advisory Board of the Simulation Center of the Informatics Cooperative Research and Education Center of the Lóránd Eötvös University. Dr. Gulyás has authored several book chapters (5+) and journal articles (5+), and published many conference papers (40+). He participated in two international research consortia under the European Commission's 6th Framework Programme and was project leader or participant in 4 R&D projects funded by the Hungarian Government. His main research interests are computational multi-agent systems where he has worked on 'engineering' desired emergent phenomena. He is currently working on agent-based models of social systems.

References

[1] Clark, D.D., Partridge C., and Ramming, J.C., "A knowledge plane for the Internet", In Proc. of the 2003 Conference on Applications, Technologies, architectures, and protocols for computer communications, August 25-29, 2003, Karlsruhe, Germany

[2] De Vleeschauwer, B., Van de Meerssche, W., Simoens, P., De Turck F, Dhoedt, B., Demeester, P., Gilon E., Struyve, K., Van Caenegem T., "On the Enhancement of QoE for IPTV Services through Knowledge Plane Deployment", In Proc. of Broadband Europe, December 11-14, 2006, Geneva, Switzerland

[3] Li, J., "Agent Organization in the Knowledge Plane", PhD. Dissertation, MIT, 2008.

[4] Kim, S., Won, Y.J., Choi, M., Hong, J.W., Strassner, J., "Towards Management of the Future Internet", In Proc. of the 1st IEEE/IFIP Workshop on Management of the Future Internet, Long Island, NY, USA, June 5, 2009.

[5] Dietterich, T., Harvey, B.T., Miller, D., Jones, D., Gray, A., Fern, A., Tadepalli, P., "Machine Learning for the Knowledge Plane", DARPA Technical Report, Oregon State University, June 2006.

[6] Mbaye, M., Krief, F., "A Collaborative Knowledge Plane for Autonomic Networks", In Autonomic Communication, Ed. by Vasilakos, A.V., Parashar, M., Karnouskos, S., Pedrycz, W., ISBN 978-0-387-09752-7, Springer, 2010.

[7] Simoens, P., Vleeschauwer, B.D., Van de Meerssche, W., De Truck, F., Dhoedt, B., Demeester, P., Van Caeneghem, T., Struyve, K., Dequeker, H., Gilon, E., "RTP Connection Monitoring for Enabling Autonomous Access Network QoS Management", In Proc. of 12th European Conference on Networks and Optical Communications, Stockholm, Sweden, June 2007.

[8] Latre, S., Simoens, P., Vleeschauwer, B.D., Van de Meerssche, W., De Truck, F., Dhoedt, B., Demeester, P., Van Den Berghe, S., Gilon, E., "Design for a Generic Knowledge Base for Autonomic QoE Optimization in Multimedia Access Networks", In Proc. of 2nd IEEE Workshop on Autonomic Communications and Network Management, Salvador, Brazil, April 2008.

[9] Kobayashi, A., Ishibashi, K., "VoIP Measurement Architecture using Data Mediation", In Proc. of IPOM 2009, Venice, Italy, LNCS 5843.

[10] Varga, P., Moldovan, I., "Integration of Service-Level Monitoring with Fault Management for End-to-End Multi-Provider Ethernet Services", IEEE Transactions on Network and Service Management, Vol.4 No.1, 2007.

[11] Varga, P., Moldovan, I., Molnar, G., "Complex Fault Management Solution for VoIP Services," In Infocommunications Journal, Vol. 60, pp.15-21, December 2005.

[12] Urra, A., Calle, E., Marzo, J.L., "Adding new Components to the Knowledge Plane in GMPLS over WDM Networks", In Proc. of IEEE Workshop on IP Operations and Management, Bejing, China, October 11-13, 2004.

[13] IBM, "Architectural Blueprint for Autonomic Computing", 2003.

[14] Plosz, S., Moldovan, S., Varga, P., Kantor, L., "Dependability of a Network Monitoring Hardware", In Proc. of DEPEND 2010, Venice, Italy

[15] AITIA, BME, "Report on New Architectural Platform and Specification of Example SW Code for Analysis", ARTEMIS SCALOPES Deliverable DA1.3., 2010.

[16] Szendrei, G., "Calculation and visualization of Traffic Matrices", Technical Report, BME-TMIT, 2010.

[17] Bonfiglio D., Mellia, M., Meo, M., Rossi, D., Tofanelli, P., "Revealing Skype Traffic: When Randomness Plays with You", In Proc. of SIGCOMM, Japan, 2007.

[18] Varga, P., Kovacs, L., Moldovan, I., Illes, A.Cs., Kun, G., Sey, G., Turzo, G., "Analysis of Media Communication over the Internet", Technical Report for Hungarian Telecom, 2007.

[19] Karagiannis, T., Broido, A., Faloutsos, M., Kc claffy, "Transport Layer Identification of P2P Traffic", In Proc. of the 4th ACM SIGCOMM Conference on Internet measurement, Sicily, Italy, 2004.

[20] Dorgeuille, F., Varga, P., Betoule, C., Thouenon G., Petitdemange, G., Palacios J.F., "Rationales and scenarios for investigations on next generation of access, backhauling and aggregation networks", CELTIC TIGER2 Technical Report, 2009.

[21] Wooldridge, M., "An Introduction to MultiAgent Systems", John Wiley & Sons Ltd, 2002, ISBN 0-471-49691-X

[22] Lander, S.E., "Issues in multiagent design systems", IEEE Expert, April 1997.

[23] Tessier C., Chaudron L., Mueller, H.J., "Conflicting agents: conflict management in multi-agent systems", Kluwer Academic Publishers, 2001.

[24] Hirayama, K., Yokoo, M., "The distributed breakout algorithms", In Artificial Intelligence, Vol. 161, pp.89–115, 2005.

[25] Gaiti, D., Pujolle, G., Salaun, M., Zimmermann, H., "Autonomous Network Equipments", In LNCS 3854, Springer, 2006.