Key Features of the Flow–Based Approach for Analysis of External Channel of a Local Service Provider System

Dmitri G. Korzoun[†], Prof. Victor N. Vasiliev[‡],
Dr. Yury A. Bogoyavlenskiy[§]

[†,‡,§]Department of Computer Science, University of Petrozavodsk
and
[‡,§]Institute for Informatics and Mathematical Modeling
of Technological Processes, Kola Science Center, Apatity

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {dkorzun, ybgv}@cs.karelia.ru,
rector@mainpgu.karelia.ru

Abstract

The quality of the Internet services directly depends on the capacity and functionality of local service providers. An integral element of any local service provider is an external channel. Traditional approaches to process and analyze traffic of an external channel on the packet level are not adequate due to tremendously grown volumes of the transfered data. Last time a new approach is very popular. It is based on data flows instead of individual packets. The approach is useful for a number of actual problems related to the performance evaluation of any external channel. In the paper we consider the foundations of the approach and its theoretical and practical aspects. Our goal is to establish a stable, complete framework for further research.

The research was partially supported by grant N 96–42–36 of the Ministry of Education of the Russian Federation and by grant N 01–07–90259 of the Russian Foundation for the Basic Research.

Contents

1 Introduction

During the last years an intensive growth of the Internet infrastructure has been observed, both with regard to the number of customers and to the services provided. It results in higher loading and utilization of local service providers (LSP)—the closest Internet service providers (ISP) to the customers. The performance of the whole Internet infrastructure directly depends on the capacity and functionality of LSPs because they carry most of the burden due to total Internet growth.

A typical example of LSP is the Federal Petrozavodsk RUNNet Node (FPRN), which is supported by the University of Petrozavodsk. FPRN is a testbed system for various research activities conducted by the LSP Performance Analysis Group of the Department of Computer Science of the University of Petrozavodsk.

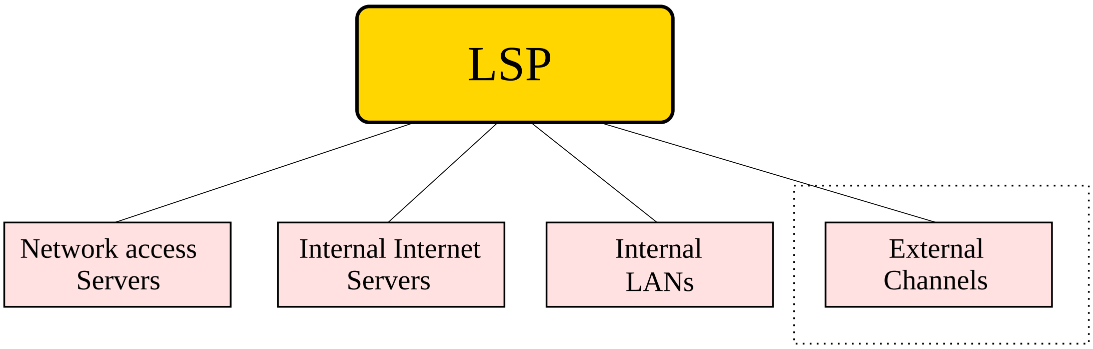

Figure 1: *The decomposition of a LSP on four main subsystems*

As was shown in [1], LSP can change rapidly, both quantitatively and qualitatively, but through all these changes its structure remains essentially the same. More exactly, there are four main subsystems: customer-access servers, internal Internet servers, internal LANs, and external channels (EC) to higher-level service providers.

To illustrate our understanding of LSP notion we first shortly present some theses from [1], where further references can be found.

Figure 1 depicts this decomposition of any LSP system. The stability of this structure is quite natural as the structure elements correspond to the main functions of LSP.

The growth of the Internet infrastructure increases the role of various distributed services located throughout the world. This means that the importance of external channels continues to grow, because they provide the customers with data-transport facilities to access these services.

The performance evaluation and capacity planning problems of the external channel subsystem are an integral part of a more general one—LSP capacity planning. The performance evaluation of EC requires intensive processing of the transfered data and it is not an easy problem since volumes of transfered data are huge, so it seems unpractical to perform complex processing of the traffic data on a very fine granularity level (octets and packets). In this case a new approach—flow-based—appears to be more adequate and helpful. The approach uses data flows (packet trains) as elementary data units instead of individual packets.

Different sides and features of the approach are intensively investigated now by Internet community [4, 5, 6, 7]. We discuss characteristics and advantages of the approach and its potential helpfulness for the EC analysis. Our goal is to systematize the approach as from theoretical so from practical points of view with aim to create stable, complete framework for further research.

The rest of the paper is organized as follows. In section 2 we describe general aspects of the EC subsystem and its different layers of traffic description: the packet layer and the flow layer. Section 3 introduces some theoretical issues of the flow-based approach: a basic definition of a flow and an example of a construction of a more complex object for the formal description of EC on the flow layer. Some important practical aspects of the approach are considered in section 4. In section 5 we demonstrate examples of some actual problems of EC traffic analysis for which the approach can be successfully applied. Section 6 summarizes the key features of the approach.

2 The External Channel Subsystem

In this section we briefly describe what an external channel is, its general properties and two main layers of the data description available for EC: the traditional packet layer and the more aggregated flow layer. The latter is a basic layer for the flow-based approach.

2.1 General Considerations

An external channel can be defined as a device that transfers data between LSP and higher-level service providers. In general, LSP has several ECs as shown in Figure 2. All nonlocal traffic therefore is separated between the channels and each EC carries only a certain portion of the total traffic.

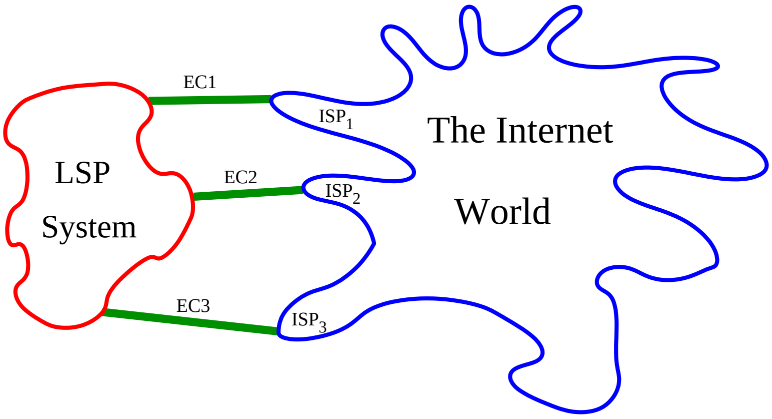

Figure 2: *LSP with 3 external channels EC_1, EC_2 and EC_3 to high-level service providers ISP_1, ISP_2 and ISP_3*

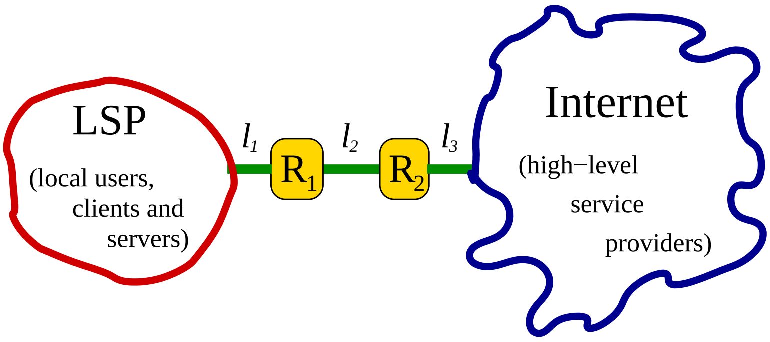
LSP
(local users, clients and servers)
l_1
R_1
l_2
R_2
l_3
Internet
(high−level service providers)

Figure 3: *An external channel as a data-transport device for nonlocal communications of the customers*

Figure 3 shows a location of the EC subsystem in the LSP system. Routers R_1 and R_2 are endpoints of the external channel. Router R_1 belongs to the LSP—it is its peer point, but R_2 is not an element of this LSP—it is a peer of a corresponding high-level service provider.

The routers divide the external channel on three links: l_1, l_2 and l_3. All external traffic must flow through all these links, but it does not mean that the traffic is the same for each link. The routers can restrict traffic or generate auxiliary traffic according to their internal algorithms, customer admission and resource sharing politics.

This fact becomes essential when one chooses a location of a meter to capture the traffic data. Different locations can cause different results and the adequate selection depends on the research goals. For instance, analysis of R_1 performance requires all data of the link l_1, but the traffic of the link l_2 is preferable for the distribution of the external data usage by the local customers.

In the case of the performance evaluation problem EC can be regarded as a network link of a special type. Figure 4 shows this idea. There are a

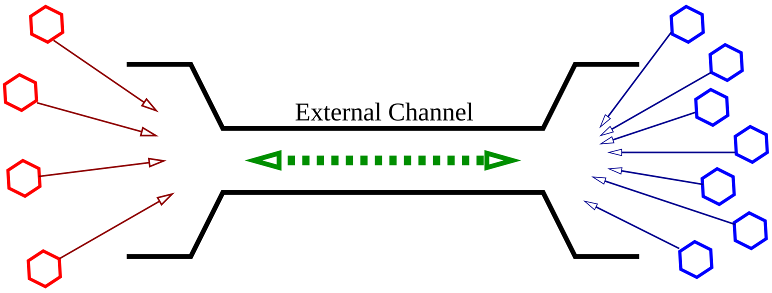

Figure 4: *External channel as a link of a special type*

diversity set of different data sources that are served. These sources are distributed over the Internet as well as located in the LSP system. The link is therefore characterized with a high level of heterogeneity and huge data volumes. The performance evaluation mainly needs information on how data are sent, information on data receivers are less important. For this reason the figure does not contain the data destinations.

2.2 The Packet Layer

Packets are very natural data units for data exchange. Thus, traditional approaches to analyze traffic data use packets as the smallest units. This approach is called *packet-based* and it performs data processing on *the packet layer*.

The key idea of the packet-based approach is shown in Figure 5. On the packet layer, the external channel is defined as a device that sequentially transmits packets. The channel cannot serve more than one packet at a time. Thus, on any finite time interval there is a sequence of transmitted packets: p_1, p_2, $\ldots$, p_n sorted with time. Each packet p_i is characterized by the following set of attributes: endpoints e_i (sender and

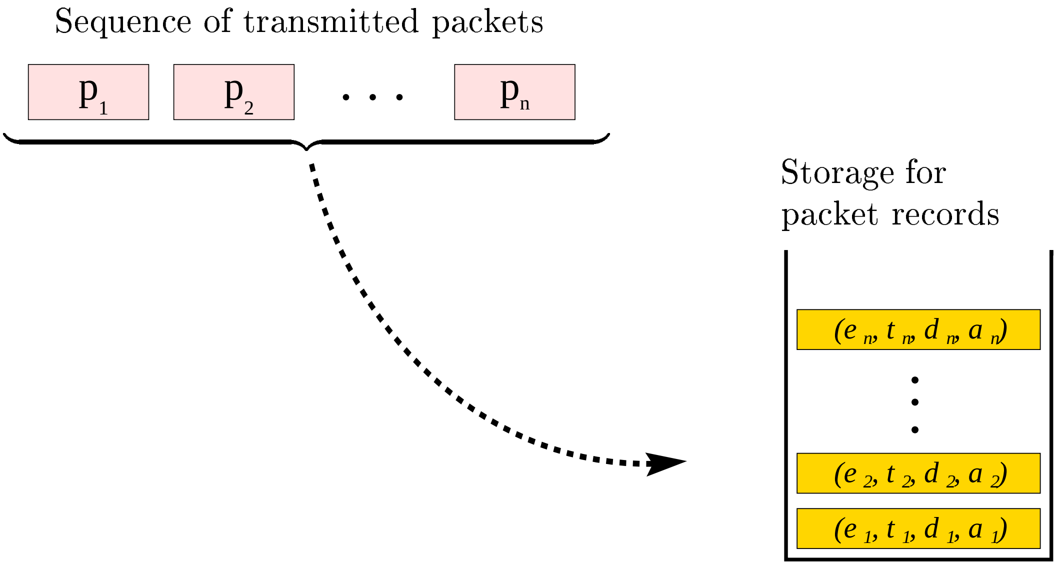

Figure 5: *For each captured packet one must store a record with packet attributes: end-points, timestamp, data volume and auxiliary items*

receiver[1]), timestamp t_i (the moment when a packet was discovered in the external channel), data volume v_i (description of data contents), and possible auxiliary attributes a_i if any (for example, protocols, flags, etc.). Thus, formally we have

Definition 1 *Packet p is a quadruple* (e, t, d, a).

An example of a concrete set of such attributes, which is used in TCPconan system, can be found in our paper on TCPconan system [3].

On the packet layer the collected data contain a record for each packet. As a result, it requires a large amount of memory to save them. Huge data volumes require not only appropriate memory—it makes data processing very time-consuming. This limitation can be a reason to refuse using a lot of well-known data analysis algorithms in practical applications, for example in real-time monitoring systems.

2.3 The Flow Layer

In many cases there is no need to use so fine granular level that is available on the packet layer. One would like to see a general (averaged) picture how the examined system behaves. He/she is not interested in a lot of minor details, caused with a single packet or a small group of them.

Measures of data volume in packets results in extremely lengthy sequence of values (each value characterizes a single packet). It consumes excessive resources to keep and process the data. A modern data processing system requires scalable data units that can be tuned to the aggregation level wanted, accordingly with the aims of research and available resources.

Numerous minor details and outliers make the analysis difficult. Sometimes it might be complicated or even impossible to separate proper events that were induced by real packets, and false events that were a result of measurement errors. This may significantly distort the results of the analysis.

An ordinary packet sequence does not capture dependencies between packets in an explicit form. It requires additional processing to discover them. Standard statistical methods like regression analysis or self-similar methods give some estimate but in many cases it may be difficult to find a good physical explanation to treat them satisfactorily.

[1] In practice, they usually correspond to IP addresses

Packets are a hardware-oriented measure of the performance. They are not adequate units for user-oriented metrics. A single user action does not involve a single packet, but generates a whole group of them—a job, a transaction, a session. Thus, this activity should be measured in more aggregative units that are closer to the user's perception of the network.

There is an alternative to avoid all these disadvantages of the packet layer—to use more aggregated data units which combine several packets in one group. These groups are called *flows.* The flow–based approach uses flows as elementary data units. The corresponding granularity layer is called *the flow layer.*

A formal definition of a flow will be presented in the next section. Nonformally, one can regard a flow as an arbitrary group of packets ordered in time. For example, one can define one flow as "all packets sent by host `alpha.cs.karelia.ru`", and another flow as "those FTP-DATA packets that were observed between 8 p.m. and 9 p.m.".

3 Theoretical Aspects of the Flow–Based Approach

Any theory requires formal definitions and constructions that make a study unambiguous and serve as a base for further research. In this section we give a general definition of a flow and demonstrate a more complex construction for the formal description of the flow–based traffic.

3.1 Definitions

Definition 2 *Define a* ***flow*** ϕ *as a nonempty finite subsequence of a total packet stream.*

A link can serve many flows simultaneously. Thus, a certain system of all observed flows is necessary.

Definition 3 *Define a* ***flow system*** Φ *as a set of flows satisfying the properties: i) any two flows* $\phi_1, \phi_2 \in \Phi$ *have no shared packets, and ii) for any packet* p *there is a flow* $\phi \in \Phi$ *that contains* p.

A flow unites all packets with the same characteristics (satisfying the same criteria). Grouping packets into flows transforms the one-dimensional sequence of transmitted packets to a new two-dimensional

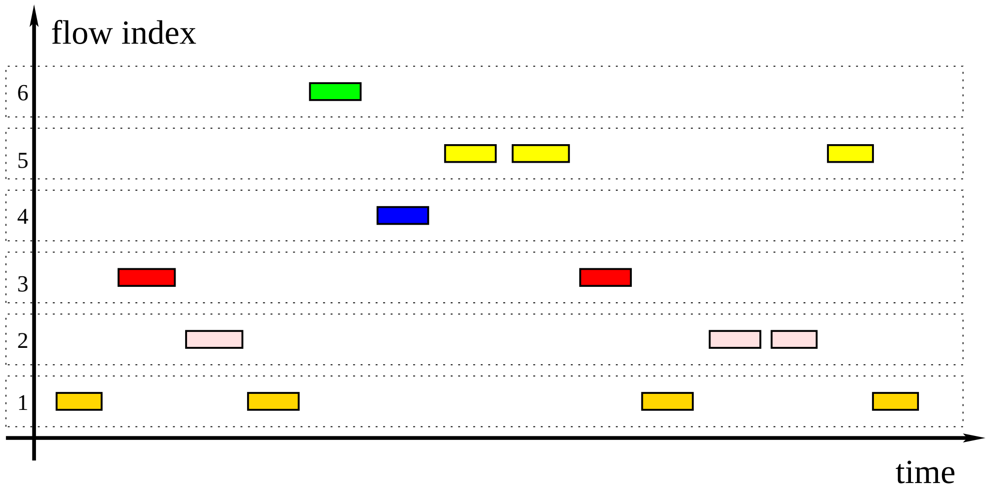

Figure 6: *A flow system as 2D structure of traffic data. The X-axis corresponds to packet timestamps. The Y-axis enumerates flows* $1, 2, \ldots, 6$.

structure—flow system Φ. Figure 6 visualizes this two-dimensional character of a flow system.

There are two main problems of the flow structure construction:

1. A way to form flows from packets.
2. A way to transform individual properties of the packets to a limited set of flow attributes.

The first problem is a definition of certain rules or criteria of the packet classification. This is a problem of the flow identification. The second problem is a definition of what attributes a flow has and what algorithms to use for their calculation. Practical aspects of both problems will be discussed later in section 4.

3.2 A Matrix of the Flow System

The definitions of a flow and a flow system can be used to define a more complex construction for the formal description of the traffic. Here we consider one example of such a construction.

On the packet layer traffic is described by a sequence of transmitted packets $\pi = (p_1, p_2, \ldots, p_n)$. The flow system is a more complex structure.

Definition 4 *Let* $\Phi = \{\phi_1, \phi_2, \ldots, \phi_m\}$ *be a flow system. Define a boolean matrix* S *as*

$$s_{ij} = \begin{cases} 1\,, & \text{if } p_j \in \phi_i \\ 0\,, & \text{otherwise} \end{cases}$$

and call it as ***a matrix of a flow system*** *(MFS).*

Each row of the matrix corresponds to some flow and vice versa.

MFS S unambiguously describes the corresponding flow system. It allows to present standard operations on packets and flows in a compact form. Some of its properties are listed below.

1. MFS has dimension $m \times n$, where m and n are the numbers of flows and packets respectively.
2. Any column of MFS has all 0s except the only 1: $\sum\limits_{i=1}^{m} s_{ij} = 1$. This is a corollary of property (i) from definition 3.
3. The number of 1s in any row of MFS is equal to the number of packets of the corresponding flow: $\sum\limits_{j=1}^{n} s_{ij} = |\phi_i|$. This is a corollary of property (ii) from definition 3.
4. There are exactly m^n different MFSs for a fixed sequence of packets.
5. Let D be a $(m \times k)$-matrix whose rows are vectors describing data volumes of corresponding packets:

$$D = \begin{pmatrix} d_1(p_1) & d_2(p_1) & \cdots & d_k(p_1) \\ d_1(p_2) & d_2(p_2) & \cdots & d_k(p_2) \\ \vdots & \vdots & \ddots & \vdots \\ d_1(p_m) & d_2(p_m) & \cdots & d_k(p_m) \end{pmatrix},$$

   where k is the number of different data types transfered through the link, and $d_l(p_j)$ is a volume of l-type data in the packet p_j. Then matrix $B = S \cdot D$ determines the data volumes of each flow.
6. Flows are scalable units: any set of flows $\{\phi_{i_1}, \phi_{i_2}, \ldots, \phi_{i_r}\}$ can be aggregated to composite flow ϕ that contains all packets of the united flows. Using MFS the unit operation reduced to a sum of the corresponding rows of the matrix.

4 Practical Aspects of the Flow–Based Approach

The definition of a flow as an arbitrary group of packets introduced in section 3 is very general. In practice one should have a more concrete notion what a flow is in the study. In this section we consider the most popular practical aspects related to the flow–based approach: the flow identification—rules to group packets into flows, the flow attributes—values that should be kept for each flow for posterior processing and analysis, and the flow–based workload characterization—a tunable description of the workload that allows inference on resource utilization.

4.1 The Flow Identification Problem

There are three aspects of a flow that are most important at its identification.

- Endpoints (a spatial aspect of a flow)
- Establishment and termination (a temporal aspect of a flow)
- Directionality (an orientational aspect of a flow)

These aspects are traditionally used for the flow identification [3, 4, 5, 6] and they determine the level of data aggregation in three dimensions: space, time and orientation.

Indeed, there are other parameters that can be used additionally or instead of the ones listed above. For example, packets can be classified according to their size: a flow for small packets only, a flow for medium ones, and a flow for large. Another example is an approach that uses the cluster analysis to discover packet condensations according to some distance metrics. However, such classifications are less popular in practice and these three parameters are enough for many performance evaluation problems.

Endpoints

An endpoint is a description of a network communication entity that acts as a data source and/or destination (someone who or something which is responsible for some activity on the network). Each flow has two endpoints A and B, and a logical data link can be seen between them. For example, an endpoint can correspond to an application (e.g. TCP port), to a host (e.g. IP or Ethernet address), to a network (e.g. address prefix, domain name or arbitrary group of hosts), to a path of the network (e.g. interface number) or to some combination of previous ones.

Endpoints are often supposed to be the most important criteria for the flow identification and it is confirmed with the following definition.

> *A burst of packets arriving from the same source and heading to the same destination.* (According to Jain as quoted by Claffi [4])

In the extreme case one endpoint can be considered as all possible sources and/or destinations—flows are aggregated for one endpoint. An example is all traffic sent from a local host (e.g. `proxy.karelia.ru`) to all destinations.

Endpoints characterize a level of the granularity of network communication entities. Different levels result in more or less aggregated elements. Internet service providers as a rule are interested in coarser-grained flows corresponding to a subnet or a host group. A more detailed granularity would be useful for analysis of the activity of some selected network components like certain applications or hosts.

Establishment and termination

The establishment and termination aspect is intended to define start and stop points of a flow—the flow time bounds. The start time is equal to a timestamp of the first observed packet belonging to the flow, and the stop time is accordingly a time stamp of the last packet. Between these time points the flow is *active*. The length of the activity period is a *flow lifetime*.

This aspect plays an important role in the flow identification, for example a flow can be defined as

> *A portion of traffic, delimited by a start and stop time, that was generated by a particular accountable entity.* (According to Brownlee [6])

Various rules of the flow activity period could be defined, depending on the type of data traffic and the demands of the analysis.

- Protocol paradigm.
  Useful if the most part of the examined traffic transferred with a connection–oriented protocol like TCP. In this case, the first packet of a flow (TCP connection) has a special flag (SYN), and the last packet has another flag (FIN or RST). This allows to identify the start and stop times explicitely, testing these flags in each observed packet.
- Timeout on the idle period.
  There is a fixed threshold T. The first observed packet satisfying all criteria of an inactive flow is the first packet of this flow. This event determines the start time and the flow becomes active. If a flow is idle more than time T (no packet has been observed) then the flow is terminated, a timestamp of its last packet is the flow's stop time, and the flow becomes inactive.
- A fixed flow lifetime.
  There is a fixed value L. The examined time period is partitioned into equal periods with length L: t_0, $t_1 = t_0 + L$, $t_2 = t_1 + L$, $\ldots$, $t_n = t_{n-1} + L$. All packets are considered as belonging to a certain flow if all flow criteria have been satisfied and all timestamps of these packets are inside $[t_{i-1}, t_i)$ for current $i \in \{1, 2, \ldots, n\}$. The first and last packets of this group determine the start and stop times.
- A limited flow lifetime.
  There is a fixed value L as in the fixed lifetime method. The first observed packet satisfying all criteria of an inactive flow is the first packet of this flow. The flow is not active more than time L. In contrast to the fixed flow time method, a flow can be terminated earlier even if its lifetime has not exceeded L. For the termination of a flow one can use for instance the protocol paradigm method or timeout on the idle period.

These rules can be used in combination. For example, for connection–oriented protocols the first method is preferable but in the case of a SYN or FIN packet loss, the correct processing requires additional rules like limiting of flow idle periods and/or flow lifetimes. Another example is a combination of timeout on idle periods and limited lifetime to systematically collect data for long-lived flows which can be active nearly always.

Directionality

There are two types of flows: uni- and bidirectional. In the unidirectional case the data are sent only in one direction—'forward', i.e. either from A to B (A is a data source, B is a destination), or from B to A (B is a data source, A is a destination), but not in both. In the bidirectional case the data can be transfered in both directions.

Defining flows as unidirectional means that traffic between A and B is separated into two distinct flows: $A \to B$ and $B \to A$. Thus, the correlation between the traffic orientations is lost, but it may be important for some types of analysis.

Connection–oriented traffic generally has bidirectionality as its intrinsic property: a data stream in one direction also generates a data stream in the reverse direction. An example is the TCP acknowledgment mechanism that induces ack-packets to acknowledge the successful data delivery. On the other hand, applications such as non-interactive audio or video conference do not require acknowledgments from the receiving end and therefore generates unidirectional or near-unidirectional flows.

Unidirectional flows are easier to monitor and collect, which is a reason for the less popularity of the bidirectional strategy. For example, in the widely spreaded CISCO NetFlow Technology [5] a flow is defined as

> *A unidirectional sequence of packets between given source and destination.*

4.2 Flow Attributes

When a flow becomes active (the first packet has been observed), a corresponding *flow entry* is created and it contains a certain set of flow attributes (such as counters for packets and bytes). When a flow has been terminated its entry is saved on a disk as a record.

Contents of a flow entry and a corresponding record are called *flow attributes*. These attributes depend on a method of the flow identification (they store a level of the aggregation) and a type of data collected for a flow (they describe events which take place in the duration of a flow).

Attributes related to the flow identification include endpoints, start and stop times, and directionality descriptions in some form. These attributes may be very complicated and consist of a complex criterion. For example, let

$$\begin{aligned} A &= \texttt{cs.karelia.ru} \\ B &= \texttt{TELNET} \\ C &= \text{any nonlocal host} \\ D &= \texttt{beta.cs.karelia.ru} \\ E &= \texttt{zeta.cs.karelia.ru} \end{aligned}$$

One can construct a complex flow criterion $\mathfrak{C}$ as follows

$$\begin{aligned} \mathfrak{C} = \ & \mathsf{NOT}\Big(\mathrm{APPS}(B)\Big) \ \& \ \Big(\mathrm{SRC}(D) \mid \mathrm{SRC}(E)\Big) \ \mid \\ & \mid \ \mathrm{DST}(C) \ \& \ \mathrm{SRC}(A) \ \& \ \mathrm{APPS}(B) \end{aligned}$$

This criterion corresponds to all non-TELNET traffic generated by {beta, zeta}.cs.karelia.ru and all TELNET traffic of cs.karelia.ru to any nonlocal destination.

If such criteria are used, one should implement a special format to store them. We call this format *an aggregation scheme*. There are two main requirements for the scheme: it should be i) economical (a few memory consumptions to store the scheme and time-efficient processing), and ii) comprehensive (its power is enough to implement all necessary aggregations of the given research problem).

Attributes for data description in the simplest case include counts for components of the flow traffic. They are measured in bytes and/or packets. If a flow is bidirectional then there must be two counters for each component: ‘forward’ and ‘backward’. The flow traffic can be multi-typed, for example, separate counters for proper data, auxiliary data and retransmitted data.

Data counters give a very coarse flow traffic summary. A functional form of the traffic dynamics can be used for more detailed data description. Let t_1 and t_2 be start and stop times of a flow. The traffic dynamics can be described with function $d(t)$, where $t \in [t_1, t_2)$. The value $d(t)$ for a fixed t determines a volume of data transfered by the flow during period $[t_1, t)$. In the simplest case $d(t)$ is linear:

$$d(t) = \frac{t - t_1}{t_2 - t_1} V \ ,$$

where V is the total volume of data on $[t_1, t_2)$ and this case is equivalent to the counter approach (V is a counter). Fixing some intermediate points

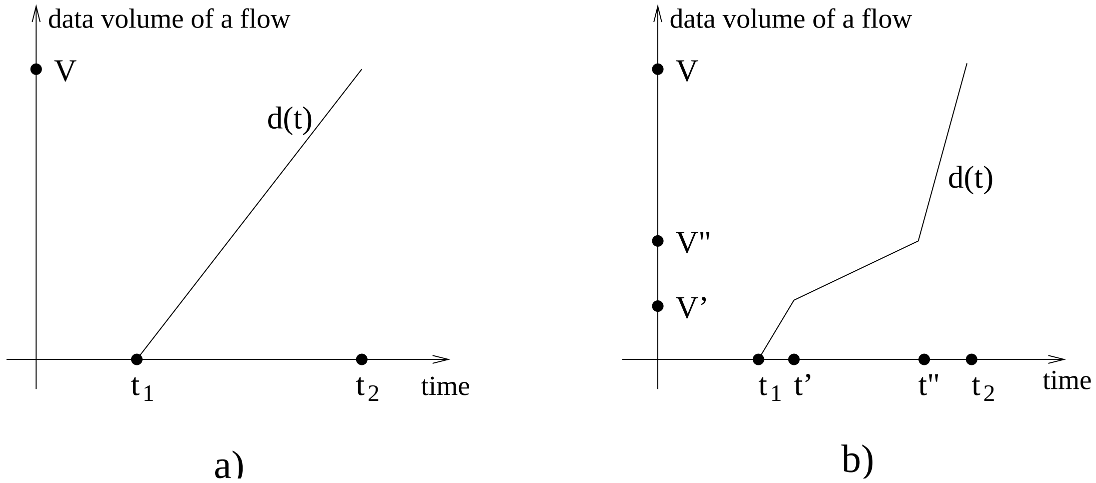

Figure 7: *The less and more detailed description of the flow dynamics. a) no intermediate points ($d(t)$ is linear). In this case the flow dynamics is uniform for all $t \in [t_1, t_2)$. b) two intermediate points t' and t'' and two additional data counters V' and V''. In this case period $[t_1, t_2)$ is divided into three different parts: $[t_1, t')$—high throughput, $[t', t'')$—low throughput, $[t'', t_2)$—extremely high throughput.*

in $[t_1, t_2)$ one can capture more detailed properties of the flow traffic dynamics (see Figure 7).

4.3 Flow–Based Workload Characterization

It is well-known that network traffic has a high level of heterogeneity. Traffic of an external channel is not an exception. Moreover, EC heterogeneity is higher than for local links, because EC serves a lot of data sources distributed over the whole Internet as well as in the local LSP system. These sources have very different characteristics and properties. It means that the traffic should be classified into several classes or types.

Flows are a certain classification of EC traffic by definition. Each flow corresponds to one type of traffic. Tuning the flow identification criteria a researcher gets several data types needed for his/her study. Flows allows to characterize the workload in terms essentially more close to a problem under study in comparison with packets. Flows can be fruitfully used on a level of the user-oriented workload [7]. In this case they are related to the number of users, jobs, sessions, etc.

The workload characterization may include (according to Lam and Chan [7]):

1. The number of hosts or applications simultaneously sending data to the channel. Using the flow–based approach this problem is reduced to the number of simultaneously active flows.
2. Mean user think time. On the flow layer it corresponds to mean duration of the inactive flow period.
3. Resource requirements for a single user activity. This characterization can be described with some statistical estimations of the flow lifetime and/or volumes of the flow data.

Summarizing all mentioned above we conclude in this section that the flow–based approach is a very flexible and tunable scheme to collect data needed. It can be applied for almost any problem of the EC performance evaluation presenting measured data in the most adequate form.

5 External Channel on the Flow Layer

In this section we give a review of problems for EC performance evaluation. Then we demonstrate applications of the flow–based description for two important problems: defining of metrics for EC performance estimation and EC traffic dynamics.

5.1 Problems of EC Performance Evaluation

There are several problems of EC that are important in the context of the performance evaluation.

- Traffic structure
- Throughput
- Utilization
- User–oriented metrics
- Resource scheduling
- Stability / burstness

The problem of the traffic structure identification can be treated as most important. It is the base for a successful solution of all other problems. The knowledge of the traffic structure allows us not to consider EC as a "black-box", but as a system with a known internal composite organization. In the simplest case, the data sources, defined in some proper way, can be considered as good candidates for the role of elements for the traffic structure.

Throughput is a measure of the EC performance and it is generally defined as $T = \frac{V}{L}$, where V is a volume of data transferred during period L. According to the flow–based approach throughput is measured in flows per second. Tuning the flow aggregation level one obtains an adequate measure for his/her study. Throughput evaluation traditionally include two parts: i) throughput dynamics analysis and ii) throughput prediction. The main aim of the throughput dynamics is to discover throughput trends, cycles and some standard patterns like daily, weekly and monthly summaries.

EC utilization is a measure of the EC usage and generally defined as $U = \frac{T}{B}$, where T is the measured throughput and B is the maximum bandwidth of the channel. Bandwidth characterizes an upper limit of EC resources. Bandwidth is a technical parameter and traditionally is available only in bps (bits per second) units. In the flow–based approach the throughput is measured in more aggregated units and, therefore, a more precise formula is $U = a\frac{T}{B}$, where a is a transmission coefficient (flows to bits)[2] which depends on the chosen flow identification method.

Flows are excellent data units as input parameters for user-oriented metrics. Generally, the user-oriented metrics include metrics of resource consumption for an average user's activity. Flows can be defined in such a way that they would be a reflection of this activity: each completed user action corresponds to a flow. An example of such activity is a user session that can be characterized with total data volume, available throughput, time spent, etc. In these conditions the total number of nondegenerate flows is another important characteristic of EC that estimates the number of users that can be served simultaneously on a certain satisfactory level.

EC simultaneously serves many users (flows) that compete for the EC resource—its bandwidth. The analysis of the resource scheduling between

[2] This coefficient is available only in some averaged form, for example, mean value of flow data volume.

these flows is very important, because a user is interested in fair sharing. Existence of dominated users that consume the most part of the bandwidth may disappoint and antagonize other ones.

The ratio between stability and burstness is an important factor of the EC performance. The presence of traffic bursts means that there can exist periods when EC is extremely either overloaded or underloaded. A high level of EC stability means that some key parameters of the link have a low variation (fluctuate on a certain level) even if in the presence of highly variable data traffic. Closely related to the stability/burstness problem is a problem of optimal EC reaction on traffic changes, because the optimal strategy generally requires to react with a minimum change (stability preservation). As a rule, the significant changes of EC behavior is a result of EC incapability to adopt to transit traffic.

5.2 EC Flow–Based Metrics

There are three general classes of metrics.

1. Current state description.
2. Virtual metrics.
3. Combine metrics

Current state metrics are generally evaluated based on real measurement data. Their aim is to estimate how EC manages with the current traffic. An example is the averaged throughput $U(t)$ that is available for a user in time point t, or the response time $\tau(t, v, e)$ that has been required for transmission v units of data starting in time t to endpoint e.

Virtual metrics are necessary for the so-called *what-if-strategy*. They estimate virtual events that have not happened really, but they could or can be: what would be if the traffic is changed in some way. An example is the predicted total throughput $\widetilde{U}(t)$ that is expected in time t. The second example is expected time consumption $\widetilde{\tau}(t, v, e)$ to transfer v units of data starting at t to endpoint e. Some estimations of EC variability also belong to the virtual metrics. For example changes of the EC utilization on adding flows to or removing them from the total traffic.

Combine metrics are some combinations of the previous two. They allow estimating a total state of EC. For instance, consider the response time τ_ϕ of flow ϕ as the time required to transfer a unit of data with this flow. One can evaluate this value τ_ϕ for real flow ϕ, and τ_{ϕ_v} for virtual flow ϕ_v. A response time coefficient can be defined as

$$K = \frac{\tau_\phi}{\tau_\phi + \tau_{\phi_v}}$$

If K is close to 1 then EC has less free resource for additional flows. If K is significantly less than 1 then it means that EC has available bandwidth and it is capable to work with a larger amount of traffic.

5.3 EC Traffic Dynamics

In the case of EC dynamics there exist some general characteristic properties that can be formulated on the flow–layer.

Suppose that conditions of the EC environment are fixed:

- New flows do not appear in the external channel.
- Existing flows are not terminated.
- Individual behavior of flow traffic does not change for each flow.
- Network elements are also stable.

In these assumptions all existing flows should reach a stable regime with the maximum available throughput. These throughputs are limited by finiteness of EC resources and their sharing between all concurrent flows. The stable regime can be considered as some optimal state of EC, because moving out from this state decreases the flow throughputs. Indeed, this only concerns those ECs that try to adopt to users' requirements as much as possible. ECs that do not have this property should be considered as not correct.

Unfortunately, these assumptions on the stability do not fit the reality. Components of the EC environment are subject to various alternations. These alternations are discrete in time because of the discrete character of the Internet.

When an alternation has had a place flows try to adopt to new circumstances and to reach the optimal stable regime. EC attempts to redistribute its resources between flows, but this attempt also conducts a new alternation. Thus, one alternation is the reason for another and so on—there is a sequence of the correlated alternations. Figure 8 demonstrates two types of such alternations for the throughput.

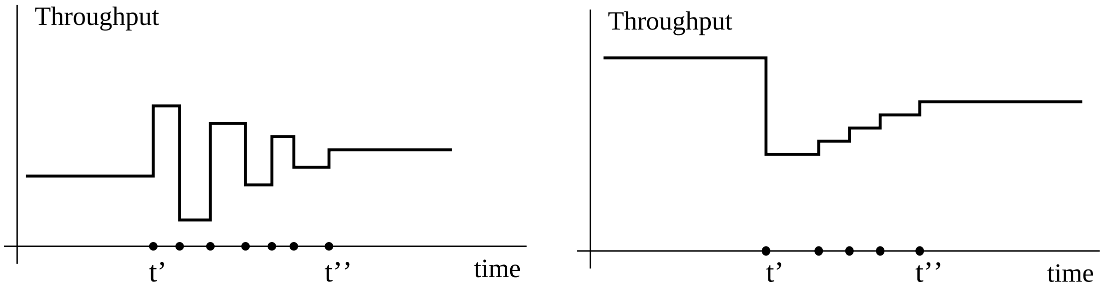

Figure 8: *Correlated alternations of the throughput. a) The first alternation (e.g. termination of a flow) at t' resulted in several subsequent jumps of the throughput that either overestimate or underestimate the optimal value, but each following is closer to the optimum, which is reached at t''. b) The first alternation (e.g. appearance of a flow) resulted in a very reduced throughput, but EC had available resources and subsequently improved the throughput and the stable regime is reached at t''.*

We define EC as *naturally working* if it always tries to move to the optimal stable regime while the environment conditions are the same (fixed). Such ECs do not degrade the flow performance.

In the general case, different ECs reach the stable regime with different speeds. This allows comparing ECs: if one EC consumes less time than another to move to the optimal state, then the first EC *works more efficiently* than the second one.

6 Summary

In this paper we considered the flow–based approach, its theoretical and practical aspects and discussed its possible applications to a problem of the EC performance evaluation.

The following list summarizes the key advantages of the approach.

- Significant reduction of data volumes (flow data can be efficiently collected and processed).
- Data aggregation to the appropriate level (removing minor details and outliers).
- Scalable data units (several small flows can be aggregated into a new complex flow).

- Natural way to keep some data correlations (a flow is a group of correlated packets).
- Different levels of the workload characterization (the most important case is the user-oriented one)
- More useful terms to describe various traffic properties (for example, natural EC behavior).

This approach has a great potential for successful traffic processing and analysis and it is sure to be put in practice by the modern research.

References

[1] V. N. Vasiliev, N. S. Ruzanova, T. Alanko, and I. A. Bogoiavlenski[3], *An Approach to Capacity Planning of a Local Service Provider as an Element of Internet Infrastructure.* Proceedings of FDPW'97-98, vol. 1, University of Petrozavodsk, 1998. pp. 21–379.

[2] D. G. Korzoun, and I. A. Bogoiavlenski, *TCPconal: A Prototype of Specification Language for TCP Connections Data Processing.* Proceedings of FDPW'97-98, vol. 1, University of Petrozavodsk, 1998. pp. 212–237.

[3] D. G. Korzoun, and I. A. Bogoiavlenski, *TCPconan: A System with Flexible Management of TCP Connections Data Processing.* Proceedings of FDPW'99, vol. 2, University of Petrozavodsk, 1999. pp. 77–94.

[4] K. Claffy, *Internet Traffic Characterization.* PhD Thesis, University of California, 1994.

[5] *NetFlow Services and Applications.* Cisco White Paper. 1998.

[6] N. Brownlee, C. Mills, and G. Ruth, *Traffic Flow Measurement: Architecture.* RFC 2063, 1997.

[7] S. F. Lam, K. H. Chan, *Computer Capacity Planning: Theory and Practice.* Academic Press, Inc. 1987.

[3] The name "Yury A. Bogoyavlenskiy" was spelled as "Iouri A. Bogoiavlenski" in accordance with the old passport for traveling abroad, which was renewed at 2000.