Identification of Performance Degradation in IP Networks Using Throughput Statistics

Markus Fiedler[a], Kurt Tutschku[b], Patrik Carlsson[a], and Arne Nilsson[a]

[a]Dept. of Telecomm. and Signal Proc., Blekinge Inst. of Techn., Karlskrona, Sweden.
E-Mail: [markus.fiedler,patrik.carlsson,arne.nilsson]@bth.se

[b]Dept. of Distributed Systems, Inst. of Computer Science, Univ. of Würzburg, Germany.
E-Mail: tutschku@informatik.uni-wuerzburg.de

To be able to satisfy their users, interactive applications like video conferences require a certain Quality-of-Service from heterogeneous networks. This paper proposes the use of throughput histograms as Quality-of-Service indicator. These histograms are built from local, unsynchronized, passive measurements of packet streams from the viewpoint of an application. They can easily be exchanged between sender and receiver, and their comparison provides information about severity and type of a potential bottleneck. We demonstrate the usefulness of these indicators for evaluating the transport quality perceived by a video conferencing application and its users in the presence of a bottleneck.

1. Introduction

Advanced IP network applications, such as IP video conferencing, Voice-over-IP, or on-line games, challenge IP network operation. They generate data streams which are increasingly sensitive to specific delay and throughput requirements. If the requirements are not met, the services may degrade substantially and become of no use.

The provisioning of sufficient throughput for specific applications is a core issue in IP traffic engineering and IP traffic management. This task is highly difficult and complex due to the main principles of the Internet technology itself: packet switching, the end-to-end arguments, as well as the diversity in technology, traffic types, and administrative domains. In order to facilitate Quality-of-Service (QoS) objectives in IP networks, different support and control technologies like IntServ, DiffServ and MPLS have been developed recently, but are not applied in a wide-spread manner. IP performance management in real-world networks is still crippled and reveals the gap between using sophisticated, but complex resource allocation mechanisms and the simple use of over-dimensioning for achieving similar QoS objectives. Moreover, recent considerations show that more simple and focused approaches might succeed over the QoS mechanisms suggested so far [5,9].

The identification of the QoS level which has been received by the application is a basic component of today's IP performance management cycle. This component has to address the basic mechanisms of the IP technology, in particular the heterogeneity of networks, the end-to-end control, and the complexity of applications. However, most monitoring procedures fail here. Many IP network administrators still use simple tools, like `ping` or `traceroute`, to evaluate the influence of the network. This might not be sufficient for advanced applications. Such tools merely provide snapshots of the network state. Furthermore, typical throughput measurements in operating IP networks are averages on intervals in the range of minutes and more. Short-term performance problems caused by "dragonflies", i.e. packet streams with life times up to two seconds [2] that may disturb an ongoing video conference often go unnoticed. Moreover, it has to be mentioned that active measurement methods always impose additional load on the network and also may disturb sensitive applications. Comprehensive overviews on monitoring tools can be found at [3] and in [1,7], for example.

Most users do not care about where in the "black box" network the problems occur. Anyway, their possibilities to monitor the network performance are quite limited. However, they would need some kind of quality feedback when using advanced applications such as video conferencing or on-line gaming. Such indicators also enable users to identify trends, to be aware of upcoming performance problems and to adapt themselves to the ever-changing traffic conditions in today's best-effort Internet. For instance, users of video conferences might want to try a lower bit rate if the application-specific control schemes [8] cannot cope with the network conditions any more. Beyond this, such indicators may assist applications in deriving appropriate control actions. Finally, network operators may profit from indicators revealing severity and type — and perhaps even location — of potential bottlenecks.

On this background, we present a passive measurement method for monitoring and visualizing the network's impact on quality-sensitive packet streams, and thus delivering bottleneck indicators. Section 2 presents the method and some of its appealing properties. Section 3 describes the measurement scenario. Section 4 presents the performance model and describes throughput statistics. The practical use for bottleneck classification of these statistics is demonstrated in Section 5 by evaluating video conferences between the departments of the authors in Sweden and Germany. Section 6 concludes the paper and gives an outlook on future work.

2. The Proposed Method And Its Properties

We suggest a monitoring method which collects throughput statistics on small time scales at the source host and the destination host. The method monitors the throughput on network level just after (before) the traffic has left (entered) the application. The observed throughput histograms are subsequently exchanged for sake of comparison and classification of a potential bottleneck.

An advantage of the suggested method is that it neither requires any specific monitoring station nor the modification of network equipment as proposed in [1,7]. The collection and preprocessing of the statistics can be performed on the computers that run the observed application. Of course, it has to verified that both the application and the monitoring process don't suffer from each other.

The suggested method is robust with regards to the characteristics of the data traffic. Since the traffic streams are compared at the input and at the output, the volatility of network traffic, e.g. due to TCP's control behavior, plays a minor role. In general, the proposed method does not require any assumptions about the traffic patterns being generated. Even though the application may initiate packets at a constant rate, these packets may leave the host already with jitter. If this fact is not taken into account, the network would be blamed for something that it is not responsible for. Since the vizualization method is flexible in terms of resolution in time and bandwidth, it can easily be adapted to the application being monitored.

The suggested monitoring method is passive. It introduces only a small additional overhead in the network when intermediate measurement results are transferred for comparison. In case of a video conference, an existing control connection could be used to exchange the histograms.

Finally, the method is supported by analytical performance investigations of bottlenecks [6], which will be explained in Section 4. Throughput statistics have been successfully applied to the evaluation of QoS degradation, cf. [4]. Our contribution, however, consists in the comparison and evaluation of throughput histograms of the same IP stream at input and output of the network.

3. Measurement Methodology

In order to describe the suggested measurement method in greater detail, we outline a prototype implementation of the measurement architecture. The method performs passive measurements on links that connect hosts to the network, see Figure 1. The measurement points (MP) are placed as close as possible to the hosts. An MP consists of a monitoring machine that is connected to one or more wiretaps. The MP runs the well-known packet capture software TCPDUMP [10] on each monitored interface. In the case of full-duplex links, wiretaps with two separate output links are needed. Consequently, the MP needs two network interface cards (NIC) and two TCPDUMP processes to monitor a full-duplex link. The complete measurement setup is shown in Figure 1. Here, hosts A and host B are the communicating entities and MP A and MP B are their corresponding measurement points. Host C and D in Figure 1 are used to generate interfering traffic, which will be described in Section 5.

The simultaneously operated measurement points MP A and MP B generate two data sets, one for each direction of the tapped link. The data sets were analyzed to identify the UDP ports used by the voice and video streams. The corresponding streams were then extracted into separate files. As a result, the measurements generate four data sets at each location with the attributes "voice/video" and "sent/received". For each packet p in these data sets, we extract the time T_p when that particular packet was received by the kernel in the MP computer, and the size L_p of its UDP payload. We anticipate that in the future, the measurements will be performed inside the host.

4. Fluid Flow Model and Throughput Histograms

Now we turn the focus on how to obtain and interpret throughput statistics for the observed voice and video streams. From the application point of view, the network is treated as single *equivalent bottleneck*, which is modeled by a *time-discrete fluid flow model*. The feasibility of this model for bottleneck identification and characterization is shown in [6]; the work presented here extends those theoretical results to real environments. Before

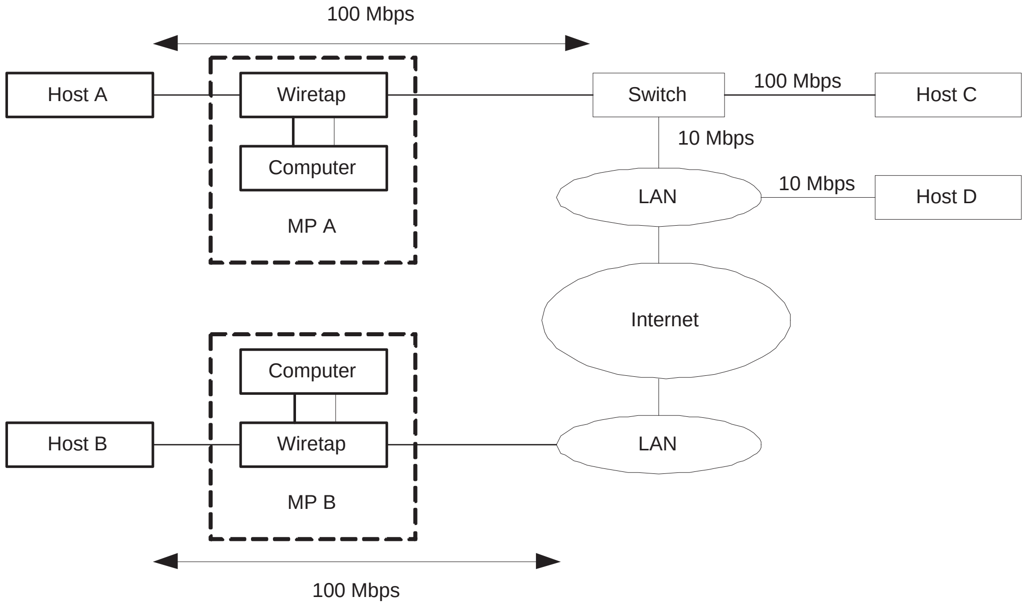


Figure 1. Measurement setup and architecture.

describing the extension, we elaborate on some general characteristics of the applied model and on the related statistics.

The considered fluid flow model works on *throughput values*, where R_s denotes the average bit rate observed during the interval $](s-1)\Delta T, s\Delta T]$. These values are obtained from the measurements of packet arrival time process $\{T_p\}_{p=1}^{k}$ and payload length process $\{L_p\}_{p=1}^{k}$. During a time window W, a *time resolution* ΔT gives $n = \lfloor W/\Delta T \rfloor$ throughput values as described in the following.

Sampling Process: Given a link capacity of C_{Link}, the time at which the payload of packet p begins is obtained as $T'_p = T_p - L_p/C_{\text{Link}}$. The arrival time of the payload of the first observed packet T'_1 is defined to be the time from which the sampling of the throughput is started. This is a natural triggering point especially from the viewpoint of the receiving application that begins to act upon reception of this payload. In order to simplify the notation, time is re-scaled:

$$t_{(\cdot)} = \frac{T_{(\cdot)} - T'_1}{\Delta T}\,. \tag{1}$$

A single sampling interval s may include several complete packets as well as parts of packets. Upon initializing $R_s = 0$, $s = 1 \ldots n$, the contributions of payload number p to the time series $\{R_s\}_{s=1}^{n}$ are calculated according to the following algorithm:

1. Calculate start interval $s' = \lceil t'_p \rceil$ and end interval $s^* = \lceil t_p \rceil$.

2. If $s' = s^*$, then all the bits of payload p belong into one sampling interval:

$$R_{s^*} = R_{s^*} + \frac{L_p}{\Delta T} \tag{2}$$

3. If $s' < s^*$, payload p covers two or more intervals:

$$\begin{aligned} R_{s'} &= R_{s'} + (s' - t'_p)\, C_{\text{Link}} \\ R_s &= C_{\text{Link}} \quad \forall s:\, s' < s < s^* \\ R_{s^*} &= R_{s^*} + (t_p - s^* + 1)\, C_{\text{Link}} \end{aligned} \tag{3}$$

Equivalent Fluid Flow Bottleneck: We are now going to look at the fluid model of the equivalent bottleneck "network". The time series $\left\{R_s^{\text{in}}\right\}_{s=1}^n$ respectively $\{R_s^{\text{out}}\}_{s=1}^n$ describes the packet stream entering respectively leaving this potential bottleneck in terms of throughput, observed by or close to the sender respectively the receiver. We denote the amount of traffic in the network at the end of interval s by X_s and define $X_0 = 0$ due to the fact that sender and receiver use the same event, the beginning of the first payload, as a starting point. Based on the observations of $\left\{R_s^{\text{in}}\right\}_{s=1}^n$ and $\{R_s^{\text{out}}\}_{s=1}^n$, the amounts of traffic X_s are determined by

$$X_s = X_{s-1} + (R_s^{\text{in}} - R_s^{\text{out}})\Delta T\,. \tag{4}$$

If the input to the network matches the output ($\left\{R_s^{\text{in}}\right\}_{s=1}^n = \{R_s^{\text{out}}\}_{s=1}^n$), the network is transparent besides a constant transmission time and does not introduce any loss. In this case, the equivalent bottleneck remains empty. However, the reality looks different in a packet-switched best-effort network without bandwidth and delivery guarantees. In such a network, different packets can experience significantly different delays and even get lost due to temporary resource shortage. In the fluid flow model, the delays are leading to variations in the throughput of streams induced by the network are reflected in the variation of the values X_s, while sudden, irreversible jumps of X_s indicate the amount of traffic lost in the network.

We now consider a condensed representation of $\{R_s\}_{s=1}^n$ in form of a summary statistics. We define *throughput histograms* $\mathcal{H}\left(\{R_s\}_{s=1}^n, \Delta R\right)$ with

$$h_i = \frac{\text{number of } R_s \in](i-1)\Delta R, i\Delta R] \text{ in window } W}{n} \quad \forall i\,, \tag{5}$$

where ΔR defines the *bandwidth resolution.* As demonstrated in [6], the comparison of throughput histograms for individual streams at input and output of a fluid flow bottleneck provides information on nature and severity of that bottleneck. We extend those theoretical findings to our real-world scenario and show that information on the quality of the intermediate network is obtained from comparing the histogram at the receiver $\mathcal{H}\left(\{R_s^{\text{out}}\}_{s=1}^n, \Delta R\right)$ with the corresponding histogram from the sender $\mathcal{H}\left(\left\{R_s^{\text{in}}\right\}_{s=1}^n, \Delta R\right)$. At the end of time window W, the receiver transfers its histogram $\mathcal{H}\left(\{R_s^{\text{out}}\}_{s=1}^n, \Delta R\right)$ to the sender. The number of bins in the histogram should be kept in reasonable limits in order to minimize transfer overhead and to permit efficient comparison between the histograms. This features contributes to the applicability of the method in real-world scenarios.

Histogram Difference Plots: From the throughput histograms at input and output, *throughput histogram differences* $\Delta\mathcal{H}\left(\{R_s^{\text{out}}\}_{s=1}^n, \left\{R_s^{\text{in}}\right\}_{s=1}^n, \Delta R\right)$ are calculated by subtracting the corresponding input histogram values from output histogram values:

$$\Delta h_i = h_i^{\text{out}} - h_i^{\text{in}}\,. \tag{6}$$

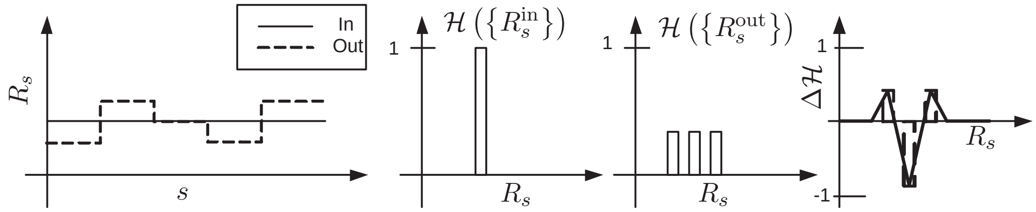


Figure 2. Antcipated time plot, throughput histograms at input and output and histogram difference plot (from left to right) in case of a shared bottleneck.

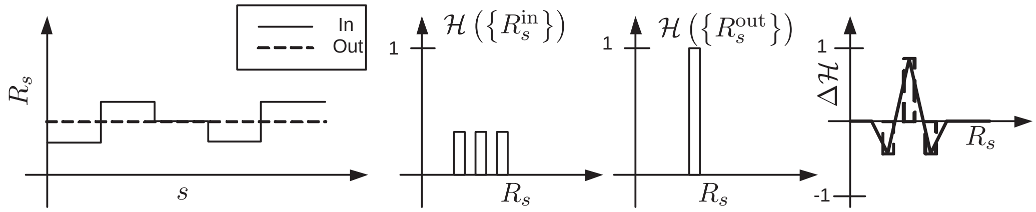


Figure 3. Antcipated time plot, throughput histograms at input and output and histogram difference plot (from left to right) in case of a shaping bottleneck.

Histogram difference plots are obtained by interconnecting the Δh_i values when plotting them versus throughput. These plots visualize the network impact on the throughput histograms and thus on the throughput itself. They are characterized by

- *width* $= \Delta R\,(\max\{\,i\,|\,\Delta h_i \neq 0\} - \min\{\,i\,|\,\Delta h_i \neq 0\})$;
- *peak-to-peak* value $= \max\{h_i\} + |\min\{h_i\}| \in [0,2]$;
- *grade of deviation* $= \sum_i \frac{|\Delta h_i|}{2} \in [0,1]$.

The "shape" of a histogram difference plot contains information about the nature of the bottleneck. This is illustrated by the following simplified examples. Figure 2 refers to a stream whose initially constant throughput is changed in a *shared bottleneck*. As soon as the demand for resources of all streams exceeds the capacity, queuing occurs and the throughput decreases. As soon as the demand falls below the capacity, the queue is relaxed, which implies higher throughput at the output. Altogether, the variability – or the burstiness – of the traffic grows. The resulting difference plot has the shape of an "M" with negative values close to the original speed and positive values at both lower and higher speeds.

Figure 3 illustrates the effect of a *shaping bottleneck* on a stream. In this case, the throughput variations are smoothed, i.e. the burstiness of the traffic decreases. The difference plot has now the shape of a "W" with positive values close to the shaper's throughput and negative values at lower and higher speeds. In the sequel, we are going

Table 1
Histogram difference parameters for voice from Karlskrona to Würzburg at different levels of disturbance.

Interfering	Difference plot			Type of
Traffic [Mbps]	Width [Mbps]	Peak-to-peak	Grade of dev.	bottleneck
0	0	0	0	Not present
2	0.080	0.0100	0.7 %	Shared
4	0.080	0.027	2.0 %	Shared
6	0.046	0.007	0.5 %	Unspecified
8	0.080	0.047	3.3 %	Shaper

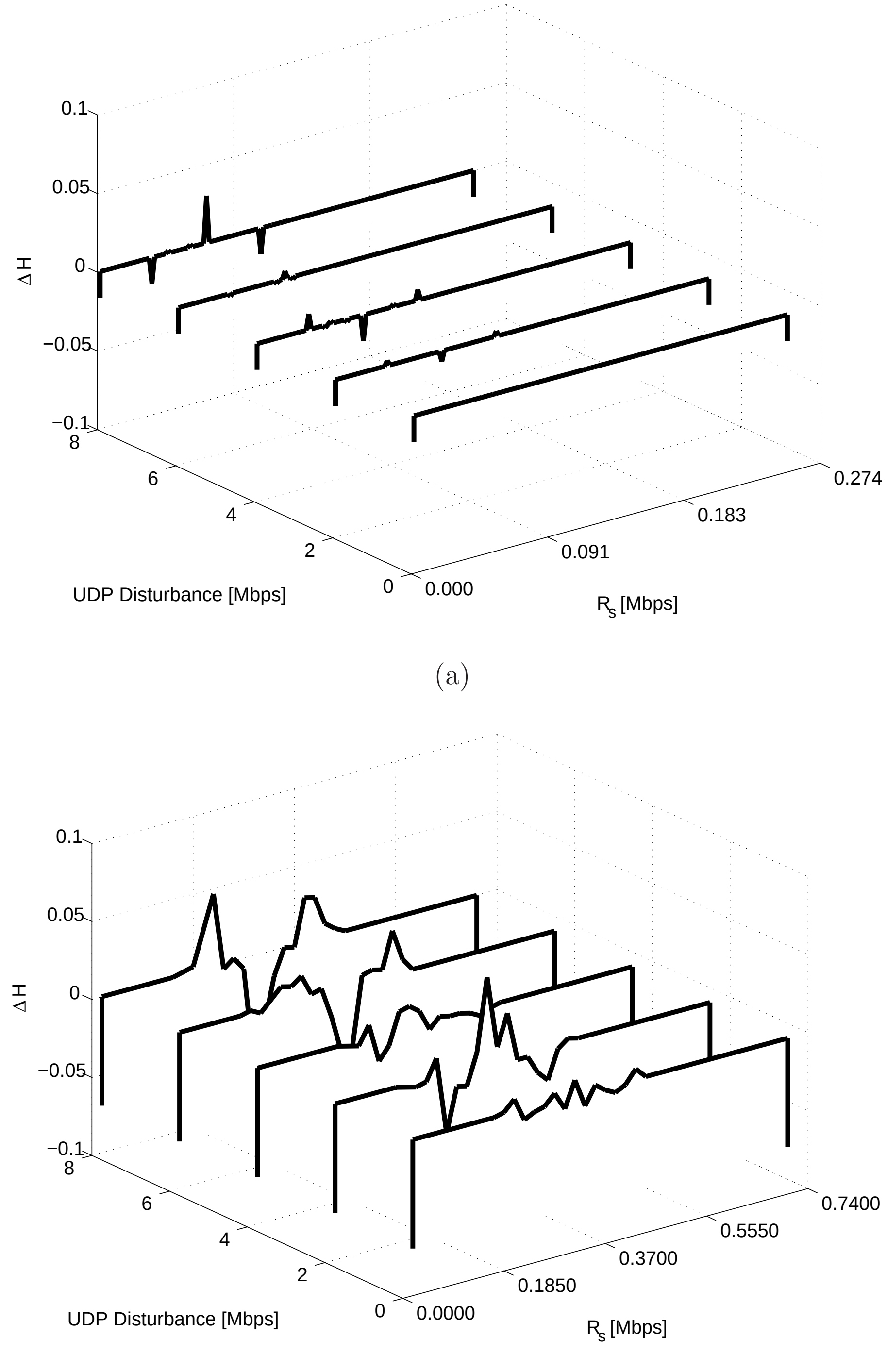
to describe such behaviors by considering real-world voice and video streams interacting with a bottleneck.

5. Test Description and Results

We investigate and visualize the performance of a video conference via European research networks with the general setup depicted in Figure 1. Host A is located at Blekinge Institute of Technology, Karlskrona, Sweden and host B is located at University of Würzburg, Germany. Both hosts run an off-the-shelf video conferencing application [8], based on H.323 on top of UDP/IP. The application offers video communications up to 384 kbps, i.e. 320 kbps for video and 64 kbps for voice. For the video stream, Dynamic Bandwidth Allocation [8] is applied, which results in variable throughput already at the sender. According to packet traces, on average $16\frac{2}{3}$ voice packets are sent per second. At the sender, their inter-packet times are roughly multiples of 16 ms, which means that the sending application already introduces a significant amount of jitter. As voice is more sensitive to delay and jitter, it is prioritized by the application, cf. [8].

Our experience is that video conferences between Karlskrona and Würzburg usually work very well, which is mainly the result of the corresponding research networks being over-dimensioned. To compromise this quality, host C, cf. Figure 1, was used to send UDP packet streams to host D, thus turning the 10 Mbps link between the switch and the LAN into a bottleneck. With 6 Mbps of disturbing UDP traffic, the users still perceived a sufficient quality. However, at 8 Mbps disturbance, glitches in the video occurred, whereas voice was not affected. Overloading the bottleneck by adding 10 Mbps corrupted the transmission and caused a tear down of the video conference. In the following, we concentrate on the streams from Karlskrona to Würzburg passing the bottleneck in the same direction as the disturbing UDP stream. The time window was chosen as $W =$ 1 minute and the time resolution as $\Delta T = 100$ ms, while the throughput resolutions for voice and video are set to 2 kbps and 20 kbps.

Figure 4(a) shows histogram difference plots for video and rising levels of UDP disturbance, and Table 1 contains some related parameters as defined in the previous section. In the undisturbed case, the voice stream does not experience any change in its bit rate statistics. For disturbances of 2 Mbps and 4 Mbps, the network influence is that of a shared bottleneck. For disturbances of 6 Mbps and especially of 8 Mbps, the voice stream



(a)

(b)

Figure 4. Histogram difference plots for (a) voice and (b) video from Karlskrona to Würzburg for different levels of disturbance.

Table 2
Histogram difference parameters for video from Karlskrona to Würzburg at different levels of disturbance.

Interfering	Difference plot			Type of
Traffic [Mbps]	Width [Mbps]	Peak-to-peak	Grade of dev.	bottleneck
0	0.260	0.018	3.5 %	Unspecified
2	0.320	0.093	11.3 %	Shaper
4	0.280	0.030	4.3 %	Unspecified
6	0.300	0.070	10.2 %	Shared
8	0.300	0.117	14.5 %	Shared

experiences some shaping: More intervals of 100 ms containing two packets ($\simeq$ 80 kbps) were observed at the receiver than at the sender, while the relative frequency of intervals containing one or three packets decreased.

In the case of the video stream, cf. Figure 4(b) and Table 2, the deviations are generally larger and appear more frequently. For a disturbance of 2 Mbps, we observe the typical "W" shape belonging to a shaping bottleneck, while disturbances of 6 Mbps and 8 Mbps show the typical "M" shape of a shared bottleneck increasing the burstiness of the traffic. In the other cases, there is no clear indication about the type of the bottleneck. If the bottleneck is undisturbed, the whole network between Karlskrona and Würzburg merely introduces some throughput "noise". On the other hand, and obviously, the deviations rise heavily from 4 Mbps to 8 Mbps disturbance.

According to the users' experience, the QoS seems to fall below a critical threshold when increasing the disturbance from 6 to 8 Mbps. The latter observation is supported by experiments where the UDP disturbance was replaced by downloads on top of TCP that were carried out simultaneously between both locations. The aggregate download throughput for both directions reached 7.3 Mbps, while no loss in video quality was perceived, probably thanks to TCPs congestion control. Interestingly enough, the throughput histogram difference plots indicate a shared bottleneck already *before* the users get to feel it, and although the video application controls the actual throughput [8]. This underlines the capability of the proposed indicator "throughput histogram difference plots" of signalling upcoming performance performance problems to the user.

6. Conclusions and Outlook

We have presented a method for identifying and visualizing performance problems perceived by streams of IP packets. It is based on passive monitoring of throughput statistics and exchanging these between sender and receiver. The comparison of throughput histograms reveals existence and type of a bottleneck. The method was tested on data streams stemming from an off-the-shelf video conference application in a real-world environment. It can be applied to any kind of IP data stream since it doesn't make any assumptions about the underlying traffic characteristics. Moreover, it is not limited to end-to-end investigations, but can also be used across single network elements. Thus, it serves both users and operators for their specific purposes.

Due to its universality, the suggested method may be applied in many contexts such as load control on virtual links, cf. [9], or as flow control on or on top the transport layer. Of course, both users and applications can make use of the method to adapt their data rate to an optimal level. Operators, on the other hand, may use the proposed indicator for finding and eliminating bottlenecks.

Future research will be necessary to improve the generality of the method. In particular, further investigations on the optimal sampling and quantization parameters with respect to the observed traffic are needed. In addition, we demonstrated so far only the qualitative capabilities of the method. Additional research is needed for obtaining quantitative results like thresholds for sending signals to users or applications.

REFERENCES

1. J. Bolliger and T. Gross. Bandwidth monitoring for network-aware applications. In *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01), 7–9 August, San Francisco, USA*, 2001.
2. N. Brownlee and KC Claffy. Understanding Internet traffic streams: Dragonflies and turtoises. *IEEE Communications Magazine*, 40(10):110–117, October 2002.
3. CAIDA. Performance Measurement Tools Taxonomy. http://www.caida.org/tools/taxonomy/performance.xml.
4. J. Charzinski. Fun Factor Dimensioning for Elastic Traffic. In *Proceedings of the ITC Specialist Seminar on Internet Traffic Measurement, Modeling and Management, 18–20 September, Monterey, USA*, 2000.
5. D. Clark, K. Sollins, J. Wroclawski, and R. Braden. Tussle in cyberspace: Defining tomorrow's Internet. In *Proceedings of the ACM SIGCOMM Conference 2002, Pittsburgh, 19–23 August, 2002.*
6. M. Fiedler and K. Tutschku. Application of the stochastic fluid flow model for bottleneck identification and classification. In *Proceedings of the SCS Conference on Design, Analysis, and Simulation of Distributed Systems 2003, 30 March – 3 April, Orlando, USA*, 2003.
7. T. Lindh. A new approach to performance montoring in IP networks – combining active and passive methods. In *Proceedings of the Passive and Active Measurement Workshop 2002, 24–26 March, Fort Collins, USA*, 2002.
8. T. O'Neil. Network-based quality of service for IP video conferencing. White paper available at http://www.polycom.com/, Polycom Inc., 2002.
9. L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. OverQoS: Offering QoS using overlays. In *Proceedings of the First Workshop on Hop Topics in Networks (HotNets-I), Princeton, USA*, 2002.
10. TCPDUMP Public Repository. http://www.tcpdump.org.