Infocommunications Journal

A PUBLICATION OF THE SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS (HTE)

September 2012 **Volume IV** **Number 3** **ISSN 2061-2079**

PAPERS

DESIGN STUDIES

CALL FOR PAPERS

Technically Co-Sponsored by

IEEE COMMUNICATIONS SOCIETY

hte

IEEE HUNGARY SECTION

Indexing information

Infocommunications Journal is covered by INSPEC and Compendex.

The journal is supported by NCA NEMZETI CIVIL ALAPPROGRAM the National Civil Fund.

Infocommunications Journal

Technically co-sponsored by IEEE Communications Society and IEEE Hungary Section

Editorial Office (Subscription and Advertisements)**:**
Scientific Association for Infocommunications
H-1055 Budapest, Kossuth Lajos tér 6-8, Room: 422
Mail Address: 1372 Budapest Pf. 451. Hungary
Phone: +36 1 353 1027, Fax: +36 1 353 0451
E-mail: info@hte.hu
Web: www.hte.hu

Articles can be sent also to the following address:
Budapest University of Technology and Economics
Department of Telecommunications
Tel.: +36 1 463 3261, Fax: +36 1 463 3263
E-mail: szabo@hit.bme.hu

Subscription rates for foreign subscribers:
4 issues 50 USD, single copies 15 USD + postage

Publisher: PÉTER NAGY • Manager: ANDRÁS DANKÓ

HU ISSN 2061-2079 • Layout: MATT DTP Bt. • Printed by: FOM Media

www.infocommunications.hu

Performance evaluation of open-source software for traffic traces manipulation and analysis

German Retamosa and Javier Aracil

Abstract—**The paper presents a performance evaluation of commonly-used open-source software for manipulation and off-line analysis of traffic traces. In traffic analysis, there is a trade-off between either implementing ad-hoc, low-level software that is optimized for performance or using off-the-shelf open-source software written in high-level languages such as Python. Clearly, the former approach has a penalty in development time. While using high-level languages is easier in terms of development, the size of traffic traces is increasing and so is the processing time. We conclude that the use of high-level languages provides similar processing times in comparison with the low-level languages counterpart, provided that a pre-filtering of the traffic can be performed (by means of tcpdump).**

Index Terms—**Traffic analysis, Performance Analysis and Design Aids, Python.**

I. Introduction

NOWADAYS, network infrastructures represent a significant share of the budget in large companies or institutions, thus requiring an accurate capacity planning. Furthermore, the quality of service provided by the network has to be carefully monitored, because an increasing part of customer relationship is carried out through the network. Finally, there are many possible security threads, that can eventually lead to service outages, with severe impact in the organization.

Traffic analysis comes as a fundamental tool for capacity planning, quality of service assessment and reinforcement of security. By traffic analysis we understand the in-depth study of network traffic to extract conclusions about network and application behavior. To this end, we distinguish the following three phases in traffic analysis.

First, there is the traffic capture, which consists of passive traffic sniffing by means of a tap or SPAN port into a traffic probe. Such probe is normally equipped either by specialized hardware for traffic capturing or by commodity hardware. Recently, the use of commodity hardware for traffic sniffing at 10 Gbps has attracted considerable attention, because of the reduced cost and good performance. Actually, the authors in [1] report 10 Gbps packet capturing capability with commodity hardware and no packet loss at minimum packet size. Note that the number of packets per second in the monitored link matter, because the DMA interrupts from the network interface card happen at packet batches. Thus, for the same rate in bits per second, the larger the packet size the smaller the DMA interrupt rate.

Second, there is a preprocessing phase which is a "thinning" of the traffic for storage and subsequent analysis. As the network bandwidth increases to 10 Gbps and beyond it becomes impractical to store raw traffic traces. As an alternative, either the packet payload may be chopped to, say, 100 bytes or the packets may converted into *flows*. The latter is straightforward and it has reduced impact on traffic analysis because the application layer control information (for example the URL) usually appears in the first 100 bytes of the packet. The former requires more computing power in real time because packets are grouped into flows (same source and destination IP address, port and protocol) by means of a hash table in real time. Then, the flow record is dumped to permanent storage. Such flow record contains basic flow data such as flow size, duration, etc, and it may also contain the first bytes of the flow payload in order to drill down in the traffic content. Note that flows can be unidirectional (UDP) or bidirectional (TCP) and the sorting of packets at high speed into flows poses a major performance bottleneck.

Third, there is the processing of the network traffic. While part of the processing can be performed on-line (flow records), specially for alarm reporting, there is a large off-line analysis on the traffic trace. Such off-line analysis of the traffic trace normally involves the in-depth dissection of network protocols and applications in order to assess possible performance bottlenecks. Actually, this falls within the Application aware network performance monitoring (AA-NPM) area, which has been recently identified by Gartner as one of the top strategic areas in information technologies [17].

This paper provides a performance evaluation of commonly-used open-source software for the off-line manipulation of data traffic. Such performance evaluation is interesting for researchers and practitioners in the field because it provides an assessment of development cost versus flexibility and execution time of high-level languages such as *Python* [2]. Thanks to these kind of scripting languages, code prototyping for traffic analysis is a simple task. However, this comes at the expense of a larger execution time, which may deem the traffic analysis programs useless for high-speed environments. As an alternative, ad-hoc traffic analysis programs can be used that make optimized use of the memory and CPU, possibly performing parallelization among the CPU cores. However, this comes at the expense of a higher development cost. This tradeoff is investigated in the paper, which reviews the most popular *DPKT* [4], *Pcapy* [6], *Impacket* [6] approaches for early-prototyping of traffic analysis programs.

Manuscript submitted August 31, revised September 20, 2012. This work was supported by the IEEE.

G. Retamosa is with Naudit High Performance Computing and Networking, Faraday, 7, 28049 Madrid SPAIN (phone: +(34) 91 116 99 40; email: german.retamosa@naudit.es).

J. Aracil is with Universidad Autonoma de Madrid, Francisco Tomas y Valiente, 11, 28049 Madrid SPAIN (phone: +(34) 497 22 72; email: javier.aracil@uam.es).

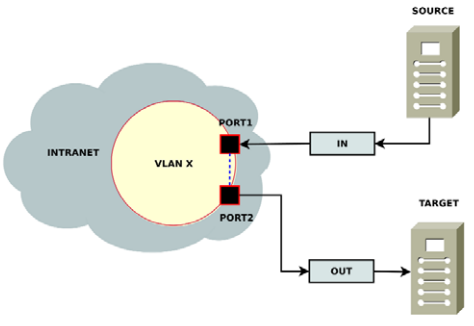


Fig. 1. Traffic duplication with SPAN port over VLAN Architecture

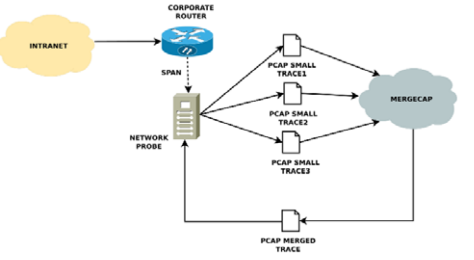


Fig. 2. *Mergecap* Architecture

Section II provides an overview of the off-line traffic analysis process, namely what are the common steps for large traffic trace handling and processing. Section III is devoted to protocol dissectors and, finally, section IV presents a use case, followed by conclusions about the assessment work carried out.

II. Software for traffic traces manipulation

Offline traffic analysis involves the following consecutive steps:

1 Merging: This is the procedure to merge several short traces into a larger one. In most cases, the traffic sniffers produce relatively small traces (1 GB worth of traffic with approximately 2,4 millions of packets), which have to be merged in order not to miss flows that have the starting part in one file and the ending part in another subsequent file.

2 Traffic deduplication: The figure 1 shows an span port on a switch or router VLAN. As it turns out, packets come out of the SPAN twice because the same packet inbound to the VLAN is transmitted eventually outbound of the VLAN. Consequently, there is a packet copy. However, chances are that the packet is not a complete duplicate but the same packet with TTL field decremented by one. This is case when there is an intermediate router in the VLAN under analysis and the packet goes in and out of such intermediate router.

3 Flow analysis: Once the traffic trace has been totally deduplicated the traffic flows have to be identified at different levels: MAC address, IP address, unidirectional flows and bidirectional flows. Normally, UDP flows are unidirection and TCP flows are bidirectional. For both, a flow record is formed that includes statistics such as origin and destination port, number of SYNs being interchanged, TCP segments with the RST flag, etc.

4 Protocol dissection: For the most common services, application performance monitoring is usually performed, such as for instance in HTTP or LDAP services. The procedure involves the extraction of the most common control fields of the protocol, per packet. Then, packets are paired (for instance GET and reply to GET in HTTP) and further statistical treatment is performed, which results in calculation of the service response time, for example (from GET to response to GET).

In what follows we describe and evaluate the software for merging, deduplication and flow analysis.

A. Merging

Merging is the first procedure that all networking software for traffic traces manipulation has to deal with. As we see in 2, traffic merging applications are a very important part in traffic monitoring because they provide a single unified traffic trace to protocol dissectors, from the many different files that are provided by the sniffer.

Traffic sniffers usually dump traffic into small files, say 1 GB worth of traffic with approximately 2,4 millions of packets. That way, the sniffer is continuously dumping traffic from memory to files, instead of keeping the whole traffic capture in memory and dumping it to file afterwards. Note that the latter would be impractical for large traffic capture periods.

In this section we analyze the architecture of merging applications. More specifically, we consider the strengths and drawbacks of the commonly used *mergecap* [11], which combines multiple saved capture files into a single PCAP output.

One of most important problems in merging applications, and specially in applications like *mergecap* [11], is the possible misordering of the packet timestamps. Actually, the packet sequence may be re-ordered when several traffic traces are merged together. Figure 3 left shows a time series of bytes per time interval with this common misordered timestamping problem in the merged traffic trace. As shown, there are noticeable temporal jumps. However, the right graph does not contain this kind of errors.

To cope with this issue it is very important to comply with the following guidelines.

1 Before calling *mergecap*, sort chronologically the PCAP input files in order to append them in chronological order and to reduce the computational load of *mergecap* [11], which employs quadratic algorithms to sort input files.

2 Do not use the exclusive append option, i.e. the *-a* option in *mergecap*, because it disables sorting algorithms in the application.

3 Sort the merged data by timestamp before processing it with traffic dissectors. Our extensive *mergecap* experimentation shows that this final step is necessary, even though the application has already sorted the packets in the merged file.

After the merging has been performed, traffic deduplication follows.

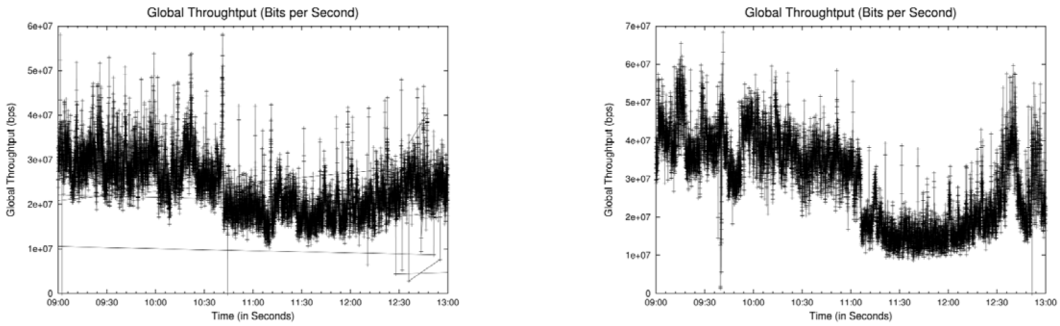


Fig. 3. Comparison of time series (unsorted left, sorted right)

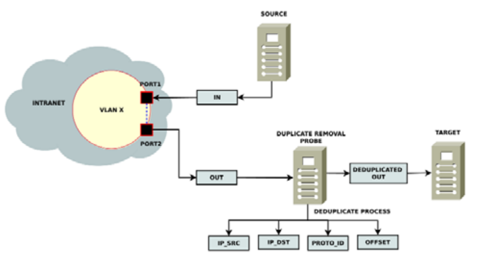


Fig. 4. Custom Duplicate Removal Architecture

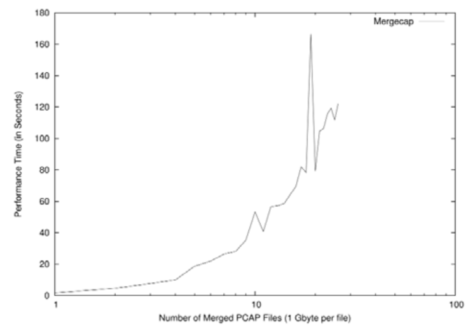


Fig. 5. *Mergecap* Performance

B. Deduplication

Typically, packet duplicates appear when sniffing traffic directly from a VLAN with input and output ports. They can also be due to routing errors. Depending on the architecture, and depending on its elements, duplicates may appear at any TCP/IP stack level.

Using a SPAN port is a very common procedure to capture traffic. The SPAN port is a special port which is configured in the router and provides a copy of the traffic, packet by packet, of either another port or a complete VLAN. For the latter, note that a packet that enters the VLAN from one port will normally leave the VLAN by another different port. Since both ports are part of the monitored VLAN it turns out that the same packet is copied twice to the SPAN.

As a result, we get a duplicate that has to be removed from the traffic trace. Other duplicates may arise when a packet is routed out of a port being monitored, then it goes through another router and then it is injected back into the same port. However, in that case the TTL field is decremented by one. Broadly speaking, we have the following types of duplicates:

1 Duplicates with the same payload but with a slight difference in the header, i.e. sequence number, TTL value and IP/Ethernet values.
2 Duplicates with the same payload and header structure.

First, duplicates with the same payload but with some differences in header fields are mainly due to routing issues or active equipment (firewalls) that duplicate traffic. For this kind of duplicates, the only way to remove them is with a custom application, whose architecture can be shown in figure 4, which evaluates whether the packet payload and IP and TCP/UDP relevant fields (address, port, etc) are indeed the same.

Second, *Editcap* [13] is a popular application to remove duplicates with the same payload and header. It is part of the *Wireshark* [14] framework.

C. Performance evaluation of merging and deduplication

In this section we evaluate the performance of the merging and deduplication procedures.

1) Merging: On the one hand, figure 5 shows the time it takes to merge several 1 Gbyte size files. The X-axis shows the number of files to be merged.

The *Mergecap* execution time increases exponentially with the size of the files and the number of traces to be merged. As it turns out, the packet-reordering process is a potential performance bottleneck, specially when the number of files to be merged is very large. In any case, the execution times are manageable and it is not worthwhile to code ad-hoc applications with performance optimizations.

2) Deduplication: On the other hand, figures 6 and 7 show the execution time of an ad-hoc application and *editcap*. The X-axis shows the size (in GBytes) of the traffic trace to be deduplicated.

The execution time for traffic deduplication, using either ad-hoc applications or editcap, increases exponentially with the number of duplicates that are present in the original PCAP traces. Furthermore, both applications have structural differences because editcap [13] is intended to remove identical packets and our custom application for duplicate removal eliminates packets that have the same payload, source and destination IP address and source and destination TCP/UDP ports.

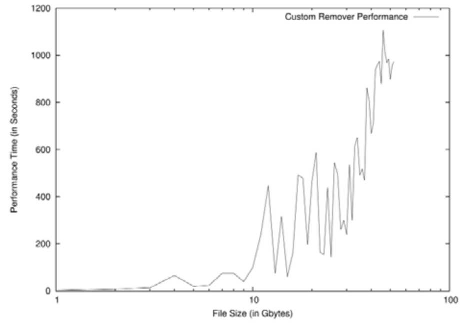


Fig. 6. Custom duplicate removal performance

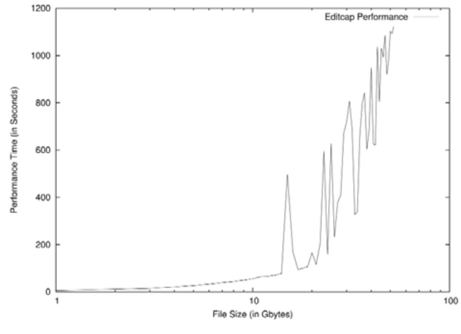


Fig. 7. *Editcap* performance

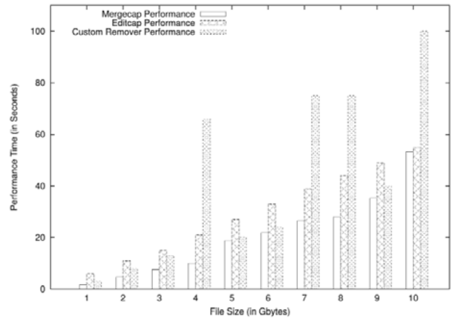


Fig. 8. Performance Comparison

Another potential performance bottleneck is the window size for deduplication, namely the number of packets that are compared with the deduplication process. This means that larger window size involves matching a larger batch of packets in a single pass and thus, impacting in the granularity of the algorithm and a better performance due to the number of packet matches to be carried out during the deduplication phase.

As a conclusion, figure 8 and specially with file sizes of 4, 7, 8 and 10 Gbytes, shows that when the number of duplicates, either identical packets or with some differences in header fields, is very high the execution time of deduplicating PCAP traces is higher than merging them due to the internal processing of PCAP packets with DPI libraries. In all other cases, their execution times and performance are close related between them.

Finally, as we see in figure 8, we conclude that the execution times of merging PCAP traces with *mergecap* is lower than deduplicating them with custom applications or *editcap* in all cases evaluated.

III. Software for traffic dissection

Developing a software application for traffic dissection is one of the most important issues in Application aware network performance monitoring (AA-NPM) but it is not a simple task. On the one hand, there is a set of communication protocols that must be selected according to user needs and studied to obtain the most important parameters. On the other hand, we have to take into account the prototyping time, analysis depth and code maintenance among others, because they are relevant aspects for the choice of the programming language and also for the overall performance. In this section, we evaluate some of the main traffic dissection programming techniques in terms of performance and timing.

A. Python *Libraries*

Python is a general-purpose and high-level programming language whose design philosophy emphasizes features such as code readability, fast prototyping and simple maintenance and as a result, It is becoming increasingly popular in recent times. Furthermore, due to its strong developer community, it is well supported and there are new versions and patches that are continuously posted.

In terms of networking and third-party libraries for traffic dissection, the most important ones are *DPKT* [4], *PCAPY* [6] and *SCAPY* [9]. *Python DPKT* [4] is a third-party library hosted into *Google Project Hosting* [3], which is maintained periodically. The following code snippet is a HTTP traffic dissection software written in *Python* [2] and *DPKT* [4] library and it is divided into three tasks: load libraries, parse PCAP files and analyze network traffic.

```
import dpkt

f = open('test.pcap')
if f:
    pcap = dpkt.pcap.Reader(f)

    for ts, buf in pcap:
        eth = dpkt.ethernet.Ethernet(buf)

    ip = eth.data
    tcp = ip.data

    if tcp.dport == 80 and len(tcp.data) > 0:
        http = dpkt.http.Request(tcp.data)
        print http.uri

    f.close()
```

Code Snippet 1. *Python DPKT* sample code

First of all, all *Python* [2] scripts must define or import all third-party dependencies (*DPKT* [4] in this case). Thus, the *Python* [2] application can use all methods inherited from those libraries.

Once the *DPKT* [4] module is loaded, the application opens the offline PCAP file (after the merging) and parses it with *DPKT* routines for networking traffic dissection.

```
f = open('test.pcap')
if f:
    pcap = dpkt.pcap.Reader(f)
```

Code Snippet 2. *Python DPKT* file reading

Finally, code snippet 3 shows the analysis process for each flow record. As shown, it is quite simple with this library and supports a large number of protocols such as LDAP, DNS or HTTP.

```
for ts, buf in pcap:
    eth = dpkt.ethernet.Ethernet(buf)

    ip = eth.data
    tcp = ip.data

        if tcp.dport == 80 and len(tcp.data) > 0:
        http = dpkt.http.Request(tcp.data)
        print http.uri
```

Code Snippet 3. *Python DPKT* file parsing

From the above code examples, it turns out that prototypes can be coded in a very short time and that maintenance is good but unfortunately, performance and analysis depth are the most important drawbacks for this kind of libraries.

The second networking library to assess is *Scapy* [9], a powerful interactive packet manipulation program written in *Python* in August 2007. As in the previous library, we show a sample code of a HTTP traffic dissection software written in *Python* and *Scapy* library, which also shows the same three steps: loading of the libraries, parsing the PCAP files and analyzing packets.

```
from scapy.all import *

pkts = rdpcap('test.pcap')
for pkt in pkts:
    if pkt.haslayer(TCP):
        if pkt.getlayer(TCP).dport == 80:
            if pkt.haslayer(Raw):
                print pkt
```

Code Snippet 4. *Python Scapy* sample code

Code snippet 4 shows a source code even shorter than other alternatives but slightly different. First of all and like the *DPKT* sample, third-party dependencies are needed to instantiate the *Scapy* methods.

Once the libraries have been loaded, code snippet 5 shows how *Scapy* [9], with *rdpcap* instruction, tries to read, parse and store in-memory at the same time. We will see later the multiple problems that this entails.

```
pkts = rdpcap('test.pcap')
```

Code Snippet 5. *Python Scapy* PCAP reading

Finally and as we see in code snippet 6, packets are filtered to the particular protocol being analysed. The method that *Scapy* [9] used to filter packets according to a specific protocol, like HTTP, is by checking protocol (TCP) and port (80).

```
if pkt.haslayer(TCP):
    if pkt.getlayer(TCP).dport == 80:
        if pkt.haslayer(Raw):
            print pkt
```

Code Snippet 6. *Python Scapy* PCAP filtering

The third library that we assess in this section is the combination of two, *Pcapy* [6] and *Impacket* [6]. *Pcapy* is a module extension that interfaces with the *libpcap* packet capture library and *Impacket* is a collection of *Python* classes focused on providing access to network packets.

As we see in Code Snippet 7, the mixed solution with *Pcapy* and *Impacket* has a code very similar to that of the *DPKT* library with the difference that *Pcapy* has an approach based into n-ary trees, where each child contains the information of each TCP/IP stack layer.

```
from pcapy import open_offline
from impacket.ImpactDecoder import EthDecoder
from impacket.ImpactPacket import IP, TCP, UDP, ICMP

decoder = EthDecoder()

def callback(jdr, data):
    packet = decoder.decode(data)
    child = packet.child()
    if isinstance(child, IP):
        child = child.child()
    if isinstance(child, TCP):
        if child.get_th_dport() == 80:
            print 'HTTP'

pcap = open_offline('test.pcap')
pcap.loop(0, callback)
```

Code Snippet 7. *Python Pcapy+Impacket* sample code

Also, the structure of the code is very similar to *DPKT* starting with reading and parsing the PCAP file (code snippet 8)

```
pcap = open_offline('test.pcap')
pcap.loop(0, callback)
```

Code Snippet 8. *Python Pcapy* PCAP reading

and then filtering and analyzing the network protocol (code snippet 9):

```
decoder = EthDecoder()

def callback(jdr, data):
    packet = decoder.decode(data)
    child = packet.child()
    if isinstance(child, IP):
        child = child.child()
    if isinstance(child, TCP):
        if child.get_th_dport() == 80:
            print 'HTTP'
```

Code Snippet 9. *Python Pcapy* PCAP filtering

As we discussed at the beginning of the section, *Python* is one of the most promising languages of the moment due to its easy integration with web interfaces, fast deployment and prototyping. Furthermore, the growing availability of new

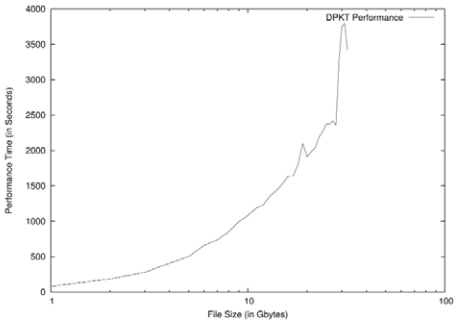


Fig. 9. *Python DPKT* Performance

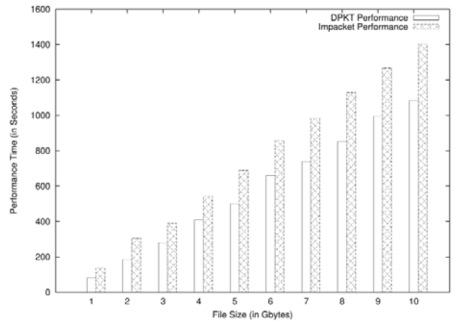


Fig. 11. Performance Comparison

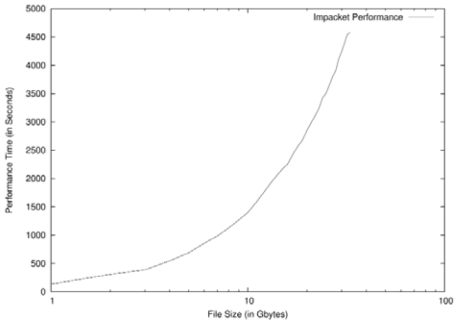


Fig. 10. *Impacket* + *Pcapy* Performance

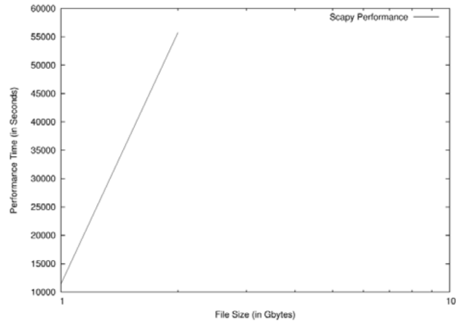


Fig. 12. Scapy Performance

third-party libraries such as *pylibpcap* [7] and *pycap* [8] represents a significant advantage, albeit there is very few documentation.

B. Performance evaluation of Python libraries

Once the most relevant Python libraries for traffic analysis have been presented in the previous section, we will assess them in order to analyze their performance and potential bottlenecks.

This assessment process consisted of measuring the execution time of a trace containing one-day worth of traffic with approximately 300 Gbytes (600 million packets) size with protocols as varied as DNS, HTTP, LDAP, TNS and so on. Furthermore, the testing architecture is composed with the following features: Intel Xeon CPU E5645 with 24 cores of 2.40GHz, 28 Terabytes of HDD, 24 GBytes of RAM and 1 interface Gigabit Ethernet.

As we can see in figures 9 and 10, the execution time increases exponentially with the PCAP file size, represented on X-axis. Furthermore, both libraries are significantly slower than the low-level application counterpart and, as we see in figure 11, their differences are slightly nonexistent.

However, the figure 12 show much worse performance figures. This is due to the internal logic of the application that individually evaluates each packet on a larger protocol list, thus impacting on search times.

As a conclusion, the main drawback of these libraries is the performance compared to low-level applications written in C with *libpcap* [10] packet capture library. Thus, deploying mixed applications combining *Tcpdump* [10] and *Python* [2] appear as the preferred solution in terms of performance.

C. Tcpdump

As we have seen in the previous subsection, *Python* libraries are a good solution for rapid prototyping of traffic dissectors and without an advanced networking and programming skills needed thanks to the abstraction layers that they provide. However, the assessment procedure carried out over these libraries shows that the bottleneck is the execution time against other solutions written in low-level languages like C.

At this point, we propose the best compromise solution between complexity and execution time is the usage of *Tcpdump* [10] in combination with *Python*. As we will show in this section, this technique serves to accelerate the *Python* performance.

Thanks to the profiling and auto-inspection methods that some *Python* packages provides, we have concluded that the *Python* application waste a significant amount of time dissecting each packet. Therefore, a compromised solution is to filter the original trace and to extract the packets subject of analysis (for instance DNS) and, then, using the *Python* application. The filtering procedure is performed with *Tcpdump* as follows

```
tcpdump -r 'input.pcap' -w 'output.pcap' "port 53"
```

Code Snippet 10. *Tcpdump* command example

which results in the extraction of the UDP DNS packets (port 53) from the original trace file *input.pcap* to the output file *output.pcap*. Figure 13 shows a performance comparison between a ad-hoc Python application and a combined solution with Tcpdump and Python. Note that the excution time decreases significantly thanks to the *tcpdump* pre-filtering.

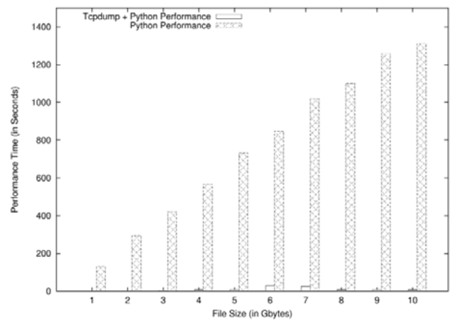


Fig. 13. *Tcpdump* Performance

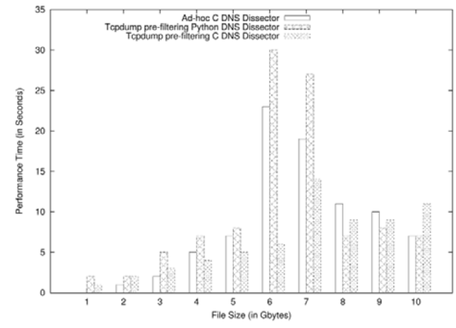


Fig. 14. DNS Traffic Dissector Performance Comparison.

The following section shows, by a simple use case, a practical and consolidated view of the previous assumption.

IV. USE CASE

In this section we show a use case for the DNS protocol. This protocol, explained with more detail into RFC 1034 [15] and RFC 1035 [16], is one of the most well-known communication protocol which provides a hierarchical distributed naming system for computers, services or any other resource connected to the Internet.

By creating a simple DNS dissector with *Python* and *Tcpdump*, it is possible obtain valuable information such as for example IP clients and servers with a high number of connections, IP clients and servers with a high number of errors or temporal series of response times and identification of clients and servers.

Before using the dissector, it is necessary to obtain UDP packets from the original PCAP trace file and filter according with the selected protocol, i.e. 53 port for DNS. As we discussed into the previous section, the usage of tools like *Tcpdump* aims to improve the performance of *Python* dissectors and reduce the workload of Python libraries on the filtered trace.

```
tcpdump -r original.pcap -w filtered.pcap "port 53"
```

Code Snippet 11. *Tcpdump* [10] DNS filtering

Secondly, we will use the *DPKT* library because after evaluating all *Python* libraries, *DPKT* has demonstrated to be the fastest and most complete library for analyzing PCAP traces in *Python*.

As we see in 12, the abstraction layer created by *DPKT* library provides a high level abstraction layer to parse PCAP files with little effort. Furthermore, *Python* [2] provides a useful implementation of lists, dictionaries and tuples that help to reduce the development time significantly.

```
import socket
import dpkt
import sys
import numpy as np

pcapReader = dpkt.pcap.Reader(open('input.pcap'))

pkts_no_ethernet = 0
pkts_IP = 0
pkts_lost = 0

for ts, data in pcapReader:
  try:
    ether = dpkt.ethernet.Ethernet(data)
    if ether.type == dpkt.ethernet.ETH_TYPE_IP:
      ip = ether.data
    if ip.p == 17:
      udp = ip.data
      pkts_IP += 1
      if (udp.dport == 53) and ip.off & 0x00FF == 0:
        dns = dpkt.dns.DNS(udp.data)
      elif (udp.sport == 53) and ip.off & 0x00FF == 0:
        dns = dpkt.dns.DNS(udp.data)
      else:
        pkts_lost+=1
    else:
      pkts_no_ethernet+=1

  except Exception as ex:
    continue
```

Code Snippet 12. *Tcpdump* DNS filtering

As conclusion of the use case, figure 14 shows the results of a performance assessment that compares a traffic dissector written in C language with *libpcap* [10] library, *tcpdump* pre-filtering with *Python DPKT* [4] script and *tcpdump* pre-filtering with the same ad-hoc C language traffic dissector.

At first glance, we note that for some file sizes the performance of tcpdump pre-filtering and Python is worse compared to the ad-hoc. As it turns out, the performance depends on the number of DNS packets rather than the file size. Furthermore, the obtained results are close in most cases. However, in those cases where the number of DNS packets is too high, namely with file sizes of 6 and 7 Gbytes, the performance of python pre-filtered solution is significantly worse than its homologous in C and even worse than the ad-hoc C dissector.

The main reason for this loss of performance is due to, as we discussed in III-B, all *Python* libraries evaluated waste large amounts of time with respect to C applications, to classify each packet individually within their internal lists of protocols. Therefore, the number of DNS packets into PCAP traces will affect the number of packet classifications and search times within protocol lists and thus, the performance variations respect to C dissectors.

V. Conclusions

In conclusion, the usage of third-party libraries written in *Python* have some advantages in terms of readability and fast prototyping time with respect to low-level solutions written in C and *libpcap*. However, the main drawback for these solutions is performance and the best choice are pre-filtered solutions with *Tcpdump* or similar pre-filtering tools to obtain good performance results. As a future work, we plan to extend the assessment procedure to newer libraries and follow the evolution that the well-established libraries which have been studied in this paper.

References

[1] S. Han, K. Jang, K. Park, and S. Moon. Packetshader: a gpu-accelerated software router. In *ACM SIGCOMM, 2010*.
[2] Python Website. http://python.org
[3] Google Project Hostings. http://code.google.com
[4] DPKT Website. http://code.google.com/p/dpkt/
[5] PCAPY Website. http://oss.coresecurity.com/projects/pcapy.html
[6] Impacket Website. http://oss.coresecurity.com/projects/impacket.html
[7] pylibpcap Website. http://pylibpcap.sourceforge.net/
[8] pycap Website. http://pycap.sourceforge.net/
[9] SCAPY Website. http://www.secdev.org/projects/scapy/
[10] Tcpdump and libpcap Website. http://www.tcpdump.org
[11] Mergecap Website. http://www.ethereal.com/docs/man-pages/mergecap.1.html
[12] Ethereal Website. http://www.ethereal.com
[13] Editcap Website. http://www.wireshark.org/docs/man-pages/editcap.html
[14] Wireshark Website. http://www.wireshark.org
[15] P. Mockapetris. RFC 1034. http://www.ietf.org/rfc/rfc1034.txt
[16] P. Mockapetris. RFC 1035. http://www.ietf.org/rfc/rfc1035.txt
[17] J. Kowall, D. Curtis. Vendor Landscape for Application-Aware Network Performance Monitoring and Network Packet Brokers. http://www.gartner.com/resId=1987715

Javier Aracil received the M.Sc. and Ph.D. degrees (Honors) from Technical University of Madrid in 1993 and 1995, both in Telecommunications Engineering. In 1995 he was awarded with a Fulbright scholarship and was appointed as a Postdoctoral Researcher of the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. In 1998 he was a research scholar at the Center for Advanced Telecommunications, Systems and Services of The University of Texas at Dallas. He has been an associate professor for University of Cantabria and Public University of Navarra and he is currently a full professor at Universidad Autnoma de Madrid, Madrid, Spain. His research interest are in optical networks and performance evaluation of communication networks. He has authored more than 100 papers in international conferences and journals.

German Retamosa received the M.Sc. from Universidad Autonoma of Madrid in 2009 in Computer Science and Telecommunications Engineering. As a serial entrepreneur, he combined his computing science and telecommunications studies with R+D jobs at IBM Global Services, Movistar and as a cofounder and CEO of his own startup, Tink Security. He is currently R+D senior developer at Naudit High Performance Computing and a Ph.D. student at Universidad Autnoma de Madrid, Madrid, Spain. His research interest are in high performance computing systems on communication networks and information security threads evaluation.

Crosstalk Compensation in Thermal Transient Measurements

Péter G. Szabó, Vladimír Székely

Abstract—Nowadays as the application area of thermal transient testing is advancing further than just being a JEDEC (Joint Electron Device Engineering Council) standard to investigate the thermal behavior of electronic devices, packages etc., the importance of avoiding any disturbances in the measured signal is more relevant. A possible effect that can corrupt the signal is the electrical crosstalk between different parts of the system or between the ambient and the whole system. But if the characteristic behavior of the crosstalk is known then special procedures can be designed in order to clean the corrupted signal from the disturbances. In our paper we present two methods to eliminate the disturbing effects. The first approach is based on polarity inversion while the second utilizes multiple level excitations to manage signal separation. To verify the feasibility of these methods, the procedure is demonstrated on electro-thermal MEMS (microelectromechnical system) devices.

***Index Terms*—Crosstalk compensation; Polarity change; Regression; Thermal transient testing;**

I. INTRODUCTION

Nowadays the importance of thermal transient measurement increases as it became an industry standard to describe the thermal behavior of electronic devices and packagings [1][2]. As its application area is being pushed further to its limits [3][4][5] it has become evident to investigate the applicability, precision and noise sensitivity for measuring other devices, such as electro-thermal microsystems. From the transient temperature response of such a device, its Cauer R-C ladder and other important parameters such as the speed of thermal functional circuits can be derived (Fig. 1). However these microdevices are more sensitive against the disturbances caused by the parasitic elements than the macro scale devices because of their small geometric sizes. This kind of disturbances can be originated from the device itself or from the evaluating electronics. Depending on their location they produce an undesired signal with small or large amplitude in the transient response which could undermine our attempt to create the thermal equivalent circuit of the device. A typical disturbance is originated from the parasitic electrical impedance of the system which time constant is in the range of ms-μs.

Manuscript received May 27, revised September 20, 2012. This work was partially supported by the JEMSiP_3D ENIAC JU project (No 120016) of NKTH (No OMFB-00166/2010).

P. G. Szabó is with the Department of Electron Devices, Budapest University of Technology and Economics, Budapest, Hungary (e-mail: szabop@eet.bme.hu).

V. Székely is professor emeritus at the Department of Electron Devices, Budapest University of Technology and Economics, Budapest, Hungary (e-mail: szekely@eet.bme.hu).

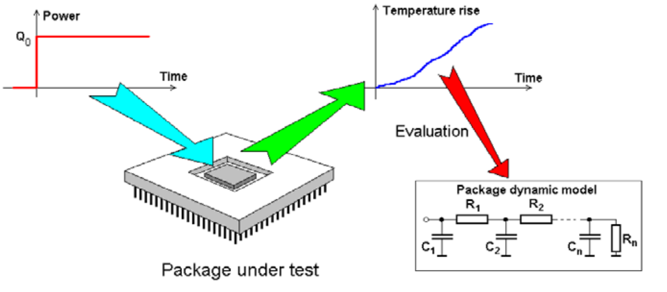


Fig. 1 Concept of thermal transient measurement

By using some special mathematical methods such as linear or square rooted curve extrapolation on the response function the unwanted signal can be filtered, but if the parasitic time constants are in the same range as that of the device then the result of the filtering will be inaccurate.

It is more straightforward to develop a special technique where the separation of the parasitic and the useful signal can be achieved.

In this paper we present two methods to separate the unwanted electric components from the thermal parts in the thermal transient measurements. In the early region of the thermal transient measurement results peaks and other non-linearities may appear in the response function due to the electric coupling between the input and the output nodes which may influence the precision of the measurement. There are a few heuristic solutions where a linear or square function is fitted to the inflexion or minimum point of the temperature rise curves. However these solutions are not adequate when structures with small time constants – for example MEMS – are measured [6][7]. These papers mention a compensation method where the thermal response is separated from the parasitic electric signal at electrostatic resonators. In the measurement setup two independent current sources were used which were utilized in common and reverse mode. As a consequence by measuring the electric resistance of the resonator the end of the electric signal and the beginning of the thermal response have been managed to pinpoint.

Another interesting method developed to filter out the electrical part in thermal transient measurements of HEMT devices was introduced recently where the basic voltage driven setup was replaced by a mixed voltage-current driven setup [8]. The basic idea behind this concept was to prevent the so called drain-lag [9]. Another often used method to separate the electrical and thermal part in the transient response of semiconductor devices is to measure them with higher input amplitudes. This can result in a faster current buildup which carries faster electrical transient that can become separable from the thermal transient.

However these methods cannot be used when the response signal is distorted by parasitic crosstalk and neither can it be generalized to be used on other structures. One of the main advantages of our newly developed characterization methods are that they can be utilized when the systems have different layout and work at other multiphysical domains as well. In addition they give the opportunity to minimize or filter out the errors which come from electrical crosstalk by processing the response function recorded at different levels of excitations.

II. INVESTIGATED SYSTEM

Examples of such systems which have small thermal time constants are called Quadratic Transfer Characteristics microsystems [10] or thermal converters [11]. The schematic of the MEMS is shown in Fig. 2. This electro-thermal device contains a dissipating resistance (R) which heats up a region H of the substrate while a thermocouple array senses the temperature difference between the H hot region and the C cold side. Since the dissipated heat is related to the square of the V_0 voltage, the T_H-T_C temperature difference and thus the output voltage will be proportional to the square of V_0. Thanks to its operation in multiphysical domains the structure is suitable to be used to identify its thermal and electrical parameters as well.

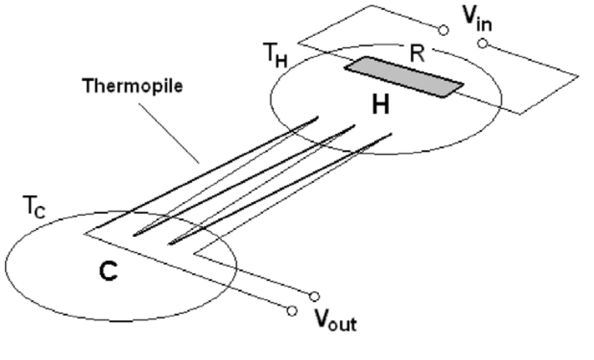


Fig. 2 Principle of the QTC element

The chips which we use were manufactured at TIMA Laboratory [12] by a combination of front-side bulk micromachining and CMOS technology. The previously described circuits which are shown in Fig. 3 were realized on cantilevers for better thermal isolation.

According to [10] [13][14] the time-constants of the cantilever can be calculated as:

$$|\sigma| = \frac{1}{\tau} = (2 \cdot n - 1)^2 \cdot \frac{\pi^2}{4 \cdot R_{th} \cdot C_{th}} + \frac{G_{th}}{C_{th}} \qquad (1)$$

where R_{th} is the thermal resistance, C_{th} is the thermal capacitance, the G_{th} is the parallel conductance of the surrounding air and n is an integer giving the n^{th} time constant.

It should be noticed that the system has symmetrical layout and is thermally insensible to polarity change.

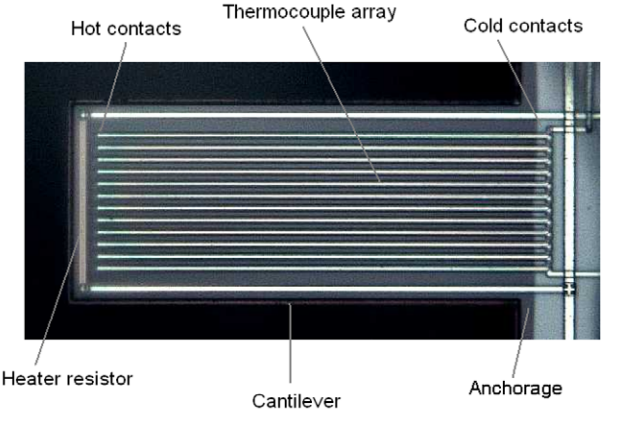


Fig. 3 The investigated microsystem

III. POLARITY CHANGE METHOD

Our first suggested solution for filtering the crosstalk is based on polarity inversion. If we assume capacitive coupling between the input and the output, then the charge movements at dynamic measurements induce a current and with it, a voltage glitch in the output signal. This scenario happens for example at a thermal transient measurement [15]. However the direction of this glitch depends on the polarity of the driving voltage. If the input is driven with positive voltage (compared to ground) then there will be a positive peak in the response function of the thermal transient measurements (Fig. 4). If the input is driven with a negative voltage then a negative peak will arise in the response function (Fig. 5). When evaluating the measurement with the standard method described in section I, these peaks will cause an additional time-constant and will also shift the time constants of the system which are in the same range.

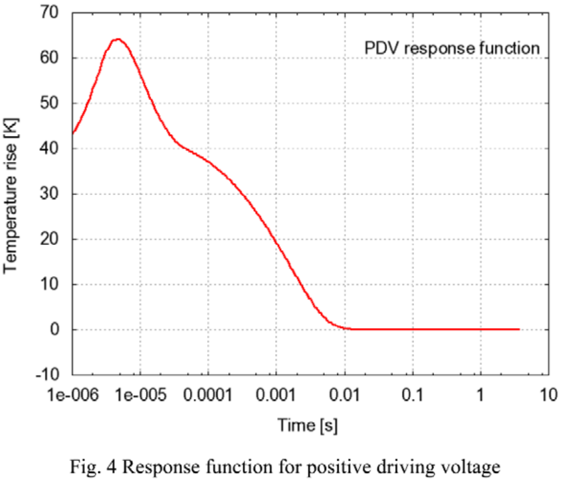


Fig. 4 Response function for positive driving voltage

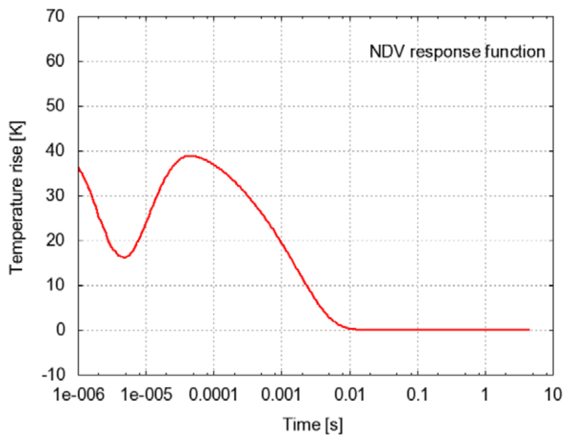


Fig. 5 Response function for negative driving voltage

The effect of polarity change against the direction of the peaks can be used when the results are processed. In theory the peaks have the same amplitude, so if the average of the two response functions is composed along the timeline, then the crosstalk free signal can be obtained. Based on this assumption, if we subtract the signals from each other and divide them by two, then the coupling signal will remain. It is practical to make these modifications in the early steps of the processing, because after this step the evaluation can be continued with the standard thermal transient evaluation methods. A processed but pre-evaluated response function can be seen in Fig. 6 where the unwanted crosstalk has been eliminated and only a small distortion remained in the early part of the response function.

It must be stated that this method can only be used if the investigated device is not sensitive to polarity inversion, but the noise is. For example this method cannot be applied to a diode or to a transistor because these devices behave in an absolutely different manner when their input is driven by positive or negative voltages.

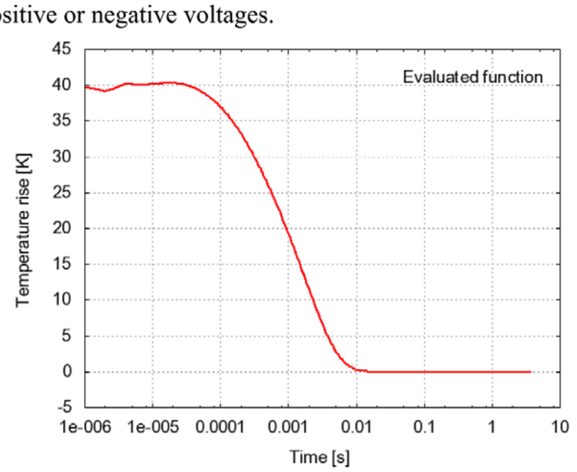


Fig. 6 Processed response function

IV. Regression Based Signal Separation

As the major part of semiconductor devices show some polarity dependency another method has to be developed which is polarity independent. For this purpose the responses of the devices against different excitations have to be investigated. Since the behavior of the devices, the ambient noises and crosstalks show specific characteristics, they can be separated analytically if the response functions are investigated properly.

As we have seen previously in section II, the effective output signal is proportional to the square of V_0 and as a result of the charge movements the unwanted (capacitive) coupling is in linear relationship with V_0. Let us denote the output signal with S and use a and b to as the intensities of the proportional input components. Throughout the analysis the minimum square approach is used. According to it, the signal which appears at the input of the transient tester is:

$$S = aV_0 + bV_0^2 \tag{2}$$

To estimate the variables a and b, a series of n measurements with different V_0 values are performed (Fig. 7).

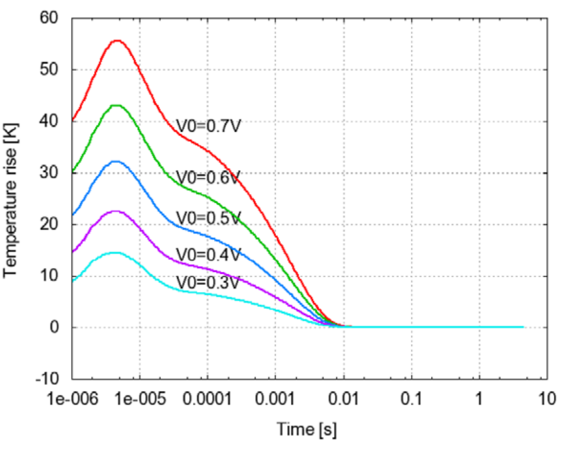


Fig. 7 Measured response functions for different V_0 values

Let us assign an e matching error for our calculations:

$$e = \sum_i^n \left(a \cdot V_{0i} + b \cdot V_{0i}^2 - S_i\right)^2 \tag{3}$$

where all the summations are over the available recordings for different V_0 values. To find the minimum of this error, (3) has to be derived with respect to variables a and b and the result should be equal to zero:

$$\frac{\partial e}{\partial a} = 2 \cdot \sum_i^n \left(a \cdot V_{0i} + b \cdot V_{0i}^2 - S_i\right) \cdot V_{0i} = 0$$

$$a \cdot \sum_i^n V_{0i}^2 + b \cdot \sum_i^n V_{0i}^3 - \sum_i^n S_i \cdot V_{0i} = 0 \tag{4}$$

and

$$\frac{\partial e}{\partial b} = 2 \cdot \sum_i^n \left(a \cdot V_{0i} + b \cdot V_{0i}^2 - S_i\right) \cdot V_{0i}^2 = 0$$

$$a \cdot \sum_i^n V_{0i}^3 + b \cdot \sum_i^n V_{0i}^4 - \sum_i^n S_i \cdot V_{0i}^2 = 0 \tag{5}$$

These equations constitute the following linear system of equations for a and b variables:

$$\sum V_{0i}S_i = a\sum V_{0i}^2 + b\sum V_{0i}^3$$
$$\sum V_{0i}^2 S_i = a\sum V_{0i}^3 + b\sum V_{0i}^4 \qquad (6)$$

This system of equations has to be solved for *a* and *b*. The amplitude of the component depending linearly on V_0 is given by *a* while *b* gives the amplitude of the component proportional to the second power of V_0. After solving the equations a vector is obtained for *a* and *b*. If V_0 and V_0^2 are multiplied by them accordingly, then the data in Fig. 8 is obtained. It can be seen that the glitch and the useful signal are detached effectively in the range above 5 μs.

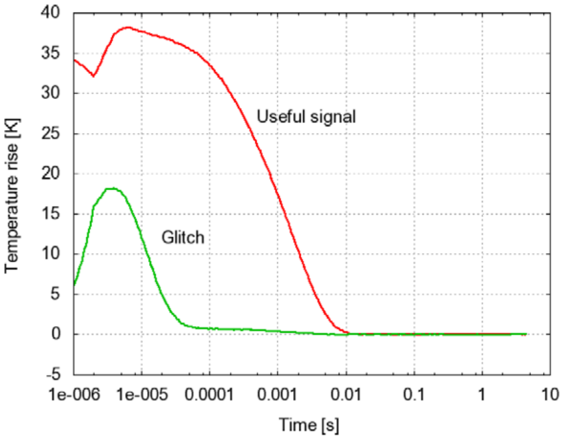


Fig. 8 Separated components

As this method is based on the differences of the signal components on different signal levels, it can be implemented when the system can only be driven by unipolar signals. As a consequence it can be used to analyze the components of devices with pn junctions. Since for example diodes have such junctions and parasitic elements such as serial resistors, their influence can be calculated by analyzing the response signals of the device under different excitations.

V. Effect of Additional Unknown Components

The primary condition of the adaptability of the regression based signal separation technique is to know the major characteristic components of the analyzed system and those of the parasitics. However it is nearly impossible to know all the components of an element and it's still worth investigating which components can be neglected because of their relatively low coefficient. For example the examined microsystems have temperature dependent components which result in V_0^4, V_0^6, V_0^8… elements [10]. The evident question emerges: which component can be neglected? Beside the influence of external noises other undesired effects such as temperature dependent material properties can also affect the separation.

For our analysis a general case is investigated when it is not known which component is neglected. Therefore a random noise or component is added to a generated response function and we investigate its effect on the separation. In order to examine the rate of the degradation in the accuracy, several sets of virtual measurement data are generated in the first step.

In each set a glitch which is proportional to the V_0 parameter is inserted, next to the useful signal which is proportional to the square of V_0. The amplitude of the glitch is either positive or negative. These virtual data functions are presented in Fig. 9 for a few V_0 values.

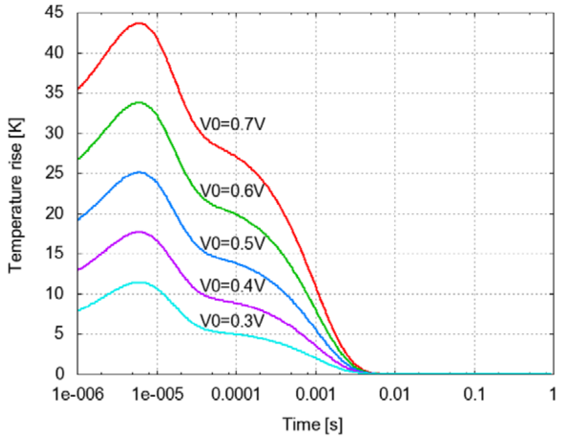


Fig. 9 Set of noiseless virtual data

In the generated functions the glitches corresponding to the crosstalk are described with

$$V_{glitch} = V_0\left((1-\exp(-t/\tau_1)) - (1-\exp(-t/\tau_2))\right) \qquad (7)$$

where $\tau_1 = 4\cdot 10^{-6}$ s and $\tau_2 = 10^{-5}$ s while the useful signal is generated by using

$$S_{useful} = V_0\left(1-\exp(-t/\tau_u)\right) \qquad (8)$$

with $\tau_u = 10^{-3}$ s. It should be noticed that exponential functions are used to describe the transient behaviors, because of their capacitive characteristics. At the glitches there are rising and falling exponential functions with different time-constants to model the charge buildup and the useful signal is described with a single stage RC ladder which yields an exponential function with one time-constant. To model the unknown components a *psr [0,1)* pseudo-random noise value multiplied by the virtual signals was added to the original signal

$$Noise = \left(S_{useful} + V_{glitch}\right)\cdot psr \cdot nl \qquad (9)$$

where *nl* is the noise level. It can be noticed that this noise is proportional to the virtual signal. This means that it has a maximum value when the corresponding signal is at its highest point and its value is zero when the signal reaches its minimum. According to (7)(8)(9) Fig. 10 show the generated signals where *nl=5 %.*

After utilizing the regression based separation technique the resulting signals can be observed in Fig. 11. It is clearly visible that for *nl=5 %* reduced effectiveness is found. In order to gain a deeper insight the results are investigated in the time-constant spectrum. The noisy useful signal in Fig. 12 contains additional time constants compared to the pure useful signal which has only one time constant. These extra time constants make difficult to distinguish the signal from some noisy components.

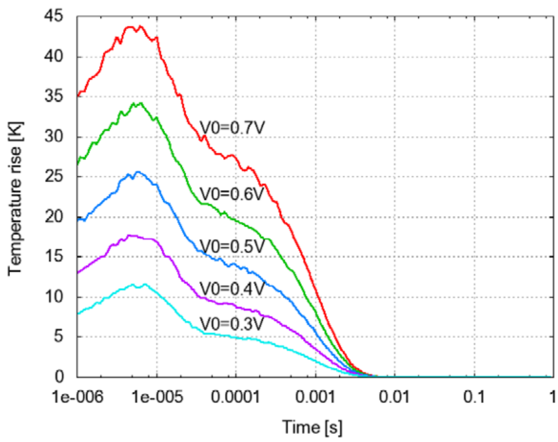


Fig. 10 Noisy signals with nl=5%

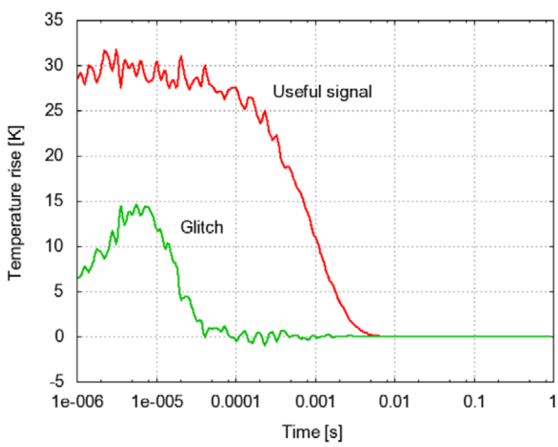


Fig. 11 Separated noisy signals with nl=5%

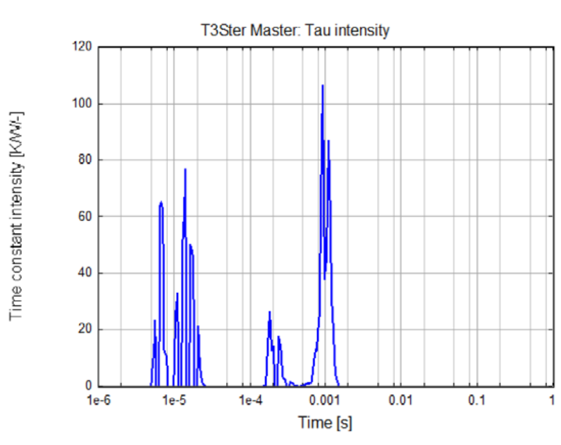


Fig. 12 Time-constant spectrum of the useful signal with nl=5%

However, when the *nl* - noise level- is decreased below 1 % then the separation could still be effective (Fig. 13). The magnitudes of the additional time-constants which can be observed in Fig. 14 are decreased as well.

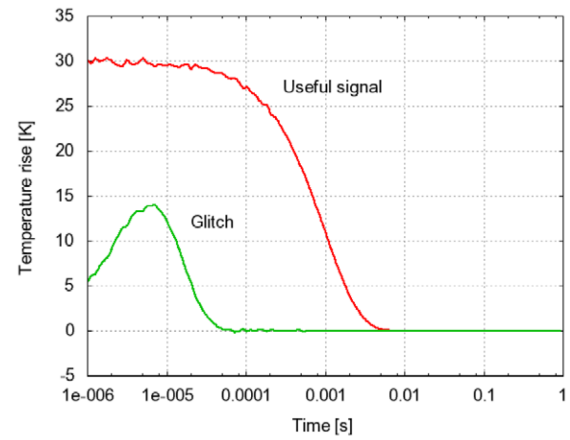


Fig. 13 Separated noisy signals with nl=1%

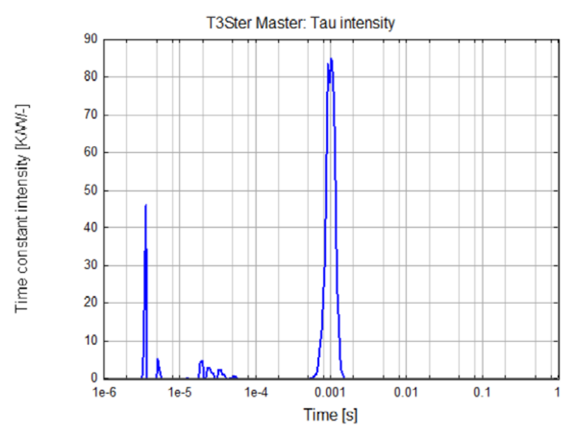


Fig. 14 Time-constant spectrum of the useful signal with nl=1%

This means that if the effects of the excess components of the response are below 1 % then they can be neglected in the separation process, but if their ratios exceed a certain level then the validation of the regression based separation becomes questionable.

VI. Conclusions

The presented case study shows new compensation techniques to improve the efficiency of thermal transient measurements in special microsystems. These methods are based on the analysis of the response signal either if the polarity of the excitation is changed or using a set of response signals recorded with different levels of driving voltage. Both methods are suitable to separate the useful and spurious signal components if their dependences on the excitation levels are known. Besides the use of similar type of microsystems, further fields of applications can be examined. For example it can be tested on semiconductor devices and microsystems which utilize other types of physical crosseffects.

REFERENCES

[1] Electronic Industries Association, "Transient Dual Interface Test Method for the Measurement of the Thermal Resistance Junction-to-Case of Semiconductor Devices with Heat Flow Through a Single Path", EIA / JEDEC Standard, JESD51-14, 2010 [www.jedec.org]
[2] Dirk Schweitzer, Heinz Pape, Liu Chen, Rudolf Kutscherauer, and Martin Walder "Transient Dual Interface Measurement – A New JEDEC Standard for the Measurement of the Junction-to-Case Thermal Resistance", Proceedings of the 27th IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM'11), pp- 222-229, 20-24 March 2011, San Jose, USA
[3] A. Vass-Várnai, B. Plesz, Z. Sárkány, A. Malek, M. Rencz "Application of Thermal Transient Testing for Solar Cell Characterization", Proceedings of the 28th IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM'12), pp. 162-168, 18-22 March 2012, San Jose, USA
[4] A. Vass-Várnai, Z. Sárkány, M. Rencz "Characterization method for thermal interface materials imitating an in-situ environment", MICROELECTRONICS JOURNAL In Press: 8 p., 2011
[5] A. Vass-Várnai, V. Székely, Z. Sárkány, M. Rencz "New level of accuracy in TIM measurements", Proceedings of the 27th IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM'11), pp. 317-324, 20-24 March 2011, San Jose, USA
[6] P. Szabó, B. Németh, M. Rencz, B. Courtois, "Characterisation of the etching quality in micro-electro-mechanical systems by thermal transient methodology", *Collection of Papers Presented at the DTIP of MEMS &MOEMS 2006*, pp. 322–327, 26–28 April 2006, Stresa, Lago Maggiore, Italy
[7] P. Szabó, B. Németh, M. Rencz, "Thermal transient characterisation of the etching quality of micro electro mechanical systems" Microelectronics Journal, 40(7):1042–1047, 2009.
[8] Z. Sárkány, G. Farkas, M. Rencz, "Thermal transient characterization of pHEMT devices", Proceedings of the 18th International Workshop on THERMal INvestigation of ICs and Systems (THERMINIC'12), pp. 225-228, 25-27 September 2012, Budapest, Hungary
[9] K. Kunihiro and Y. Ohno, "A large-signal equivalent circuit model for substrate-induced drain-lag phenomena in HJFETs", IEEE Trans. Electron Devices, Vol. 43, No. 9, pp. 1336–1342, September, 1996
[10] P. G. Szabó, V. Székely, "Characterization and modeling of electro-thermal MEMS structures", *Microsystem Technologies*, Vol. 15. No. 8, pp. 1293-1301, August, 2009
[11] H. Baltes, O. Brand, G. K. Fedder, C. Hierold, J. G. Korvink, O. Tabata "CMOS-MEMS: Advanced Micro and Nanosystems", Vol. 2. WILEY-VCH Verlag GmbH and Co., 2005.
[12] B. Charlot, S. Mir, E. F. Cota, M. Lubaszewski, B. Courtois, "Fault simulation of MEMS using HDLs" *Proceedings of SPIE*, Vol. 3680, pp. 70-77, 1999
[13] V. Székely, "New type of thermal-function IC: The 4-quadrant multiplier", *Electronics Letters*, Vol. 12, No. 15, pp. 372-373., 22nd July 1976
[14] P. G. Szabó, V. Székely, "Investigation of Parallel Heat-flow Path in Electro-thermal Microsystems", MICROSYSTEM TECHNOLOGIES Vol. 17, No. 4, pp. 533-541, 2011
[15] V. Székely, M. Rencz, B. Courtois, "Thermal Transient Testing", Microelectronics International, Vol. 14, No. 2, pp. 8-10, 1997 DOI: 10.1108/13565369710800529

ACKNOWLEDGEMENT

The authors acknowledge the help and contribution of their colleagues Albin Szalai, Ferenc Ender and Gergely Nagy for their hints during the programming. The work was partially supported by the JEMSiP_3D ENIAC JU project (No 120016) of NKTH (No OMFB-00166/2010).

BIOGRAPHIES

Péter Gábor Szabó - PhD in EE, Assistant professor - was born in Budapest, Hungary in 1983. He received his M.Sc. degree in Electrical Engineering at the Budapest University of Technology and Economics in June 2007. At 2011 he became an assistant professor at the Department of Electron Devices and one year later he obtained his PhD in Electrical Engineering. His main research areas are the modelling, design and characterization of MEMS devices and electro-thermal systems. His other research fields include thermal management and signal integrity.

Dr. Vladimír Székely - Professor emeritus, member of Hungarian Academy of Sciences - received the electrical engineering degree from the Technical University of Budapest, Hungary, in 1964. He joined the Department of Electron Devices in that year. He was Head of Department between 1990 and 2005. Currently he is professor emeritus of Budapest University of Technology and Economics. His first research area was the theory of Gunn devices. His later research interests are mainly in the area of computer aided design of integrated circuits, with a particular emphasis on circuit simulation, thermal simulation, and device modeling. He conducted the development of several CAD programs in the field of integrated circuit design and simulation. Another area of his technical interest is computer-graphics and image-processing. He has been engaged in the investigation of thermal properties of semiconductor devices and integrated circuits for the last 30 years. This resulted in the development of novel thermal based IC elements and thermal IC simulator programs. Vladimír Székely has published his theoretical and practical results in more than 380 technical papers and 12 books or book-chapters.

Cambridge Correlator in Driver Assistance Systems

[1]T. Harasthy, [2]Ľ. Ovseník and [3]Ján Turán

***Abstract*—This paper discusses the very actual and important issue of current time, the safety of road traffic. Many car designers are presently concentrating on proposing cars with modern design, but driver and crew safety is also necessary. Traffic Signs Recognition Systems are presently implemented in many cars to obtain better safety of the driver and comfortable driving. Traffic Sign Recognition system presented in this paper as a part of Driver Assistance System uses Cambridge Correlator as image comparator. Two images (reference traffic sign and input traffic sign) are compared according to two criteria, similarity and relative position.**

***Index Terms*— Color filtering, Joint Power Spectrum, Optical Correlator, Traffic Sign Recognition**

I. INTRODUCTION

THE Driver Assistance System (DAS) is nowadays usual and required accessory in many types of cars. There are many forms of DAS. The major role of DAS could be various, because each DAS can be specialized to different target [1,2,3,4]. There are systems as The Park Assistance System, Traffic Sign Recognition System, etc., which are situated in cars to protect the driver and the passengers of the vehicle [1,5,6,7,8]. These individual parts of DAS are connected with each other to obtain safe and comfortable drive. Each of these individual parts of DAS has its requirements to captured input information. The paper concentrates on Traffic Sign Recognition System based on Optical Correlator. This part of DAS requires captured color video scene which can include Traffic Sign. Video scene is captured from color CMOS sensor situated in car. These captured video scenes are preprocessed with proposed algorithm and then ROI is compared using Optical Processor (Correlator). As Optical Correlator (OC), the Cambridge Correlator (CC) is used [9,10,11].

Cambridge Correlator is a device which can compare two images or two-dimensional (2D) data sets at very high speed. There are two types of OCs. The first one is based on Vanderlugt filter called also Matched Filter Correlator (MFC). Cambridge Correlator belongs to second group of OCs, based on Joint Transform and is called Joint Transform Correlator (JTC) [12,13,14,15]. Both types of OCs compare images according to two criteria: similarity and relative position. JTCs require reference image (database of traffic signs) and preprocessed input image (ROI from input scene) aligned each other in input scene of correlation process. Output of correlation is pair of correlation peaks per match. Intensity and position of correlation peaks give information about which traffic sign from database was recognized [14,15].

Due to rapid progress in key elements technology (SLM, CMOS, LD and Fourier optics), the optical correlators are now commercially available and widely used for various applications in military target detection, navigation, security (face and fingerprint detection), medical and industrial image analysis [12,13,14,15].

The goal of this paper is to present experiments with DAS based on Cambridge Correlator used in recognition stage of the proposed new traffic sign recognition system [9,10,11].

The paper is organized as follows. Chapter II is devoted to basic color models. Block diagram of the whole proposed Traffic Sign Recognition System is described in Chapter III. Main blocks and their functions are discussed. Chapter III also explains the details of pre-processing of input scene and detection of ROI. Also it is devoted to describing applications of optical correlation with using Cambridge Correlator and explains identification traffic signs from correlations peaks. Experiments and results with proposed system are described in Chapter IV. Conclusions are covered in Chapter V.

II. BASIC COLOR MODELS

Color images can be modeled as three-band monochrome image data where each band of data corresponds to a different color. The actual information stored in the digital image is the brightness information in each spectral band. When the image is displayed, the corresponding brightness information is displayed on the screen by picture elements that emit light energy corresponding to that particular color.

Typical color images are represented as red, green and blue, or RGB images. Using 8-bit monochrome standard as a model, the corresponding color image would have 24 bits/pixel – 8 bits for each of the three color bands. RGB color model is shown in Fig. 1 [16,17].

The final color depends on combination of these colors. As it's known, every byte can encode values in range from 0 to 255, so we can create 16777216 various colors by combination of three basic color values [17].

Manuscript received July 6, 2012, revised September 26, 2012.

F. T. Harasthy. Technical University in Košice, Department of Electronics and Multimedia Communications, Košice, Slovakia (corresponding author phone: 055 602 4277; e-mail: tomas.harasthy@ tuke.sk).

S. Ľ. Ovseník, Technical University in Košice, Department of Electronics and Multimedia Communications, Košice, Slovakia.

T. J. Turán Technical University in Košice, Department of Electronics and Multimedia Communications, Košice, Slovakia.

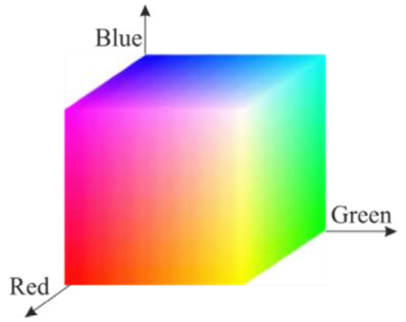


Fig. 1. RGB color model.

In many applications, RGB color information is transformed into mathematical space that decouples the brightness information from the color information. After this is done, the image information consists of one-dimensional brightness, or luminance, space and a two-dimensional color space. Now the two-dimensional color space does not contain any brightness information, but it typically contains information regarding the relative amounts of the different colors. An additional benefit of modeling the color information in this manner is that it creates a more people-oriented way of describing the colors.

For example, the Hue/Saturation/Value (HSV) color transform allows the description of colors in terms that we can more readily understand (Fig. 2) [16,17].

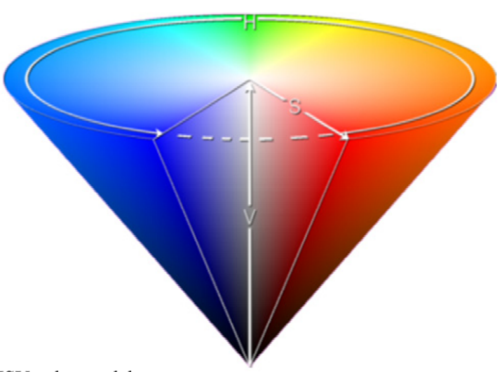


Fig. 2. HSV color model.

The Value represents quantity of white light or specifies the rate of light reflectivity. The Hue is what we normally think of as "color" (e.g. blue, purple, red, yellow etc.). It can be in range from 0 to 360 degrees. The value of Saturation labels quantity of grey color in proportion to basic hue. It can be in range from 0 to 100 % [16,17,18].

III. Traffic Sign Recognition System

Traffic Sign Recognition is one of most important part of DAS [1,5,6,9,10]. These systems provide information of the road signs on the way and guide the driver. DAS helps the driver to recognize the traffic signs earlier and accurately. In ideal conditions this system could avoid false recognition (or misleading recognition) of traffic signs caused by various human factors (inattention, tiredness, sleepiness or micro-sleep). Thus, the traffic sign recognition system makes the driving safer and easier. The outdoor traffic scene, due to illumination changes (day, night, haze, snow, fog, rain, etc.), shadows, partial occlusions, sign rotation and damage becomes a very challenging and difficult task [7,9,10].

Fast preprocessing time is very important to operate the recognition system in real time environment. The systems consist of four main components (Fig. 3).

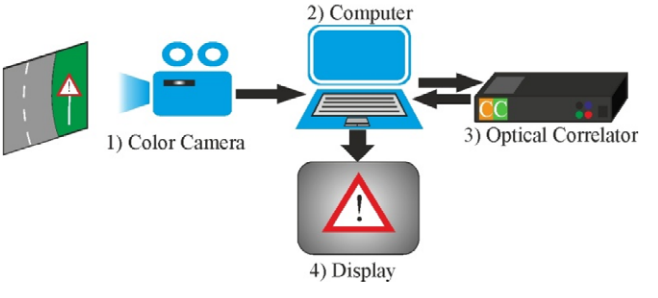


Fig. 3. Hardware scheme of developed system.

1) Color Camera, traffic scene has to be captured by the color camera (CMOS is mostly used). Color interpretation of input traffic scene is required due to remove background of traffic sign (color filtering).
2) Computer, pre-processing of input scene is necessary due to minimize of DC noise in correlation process. ROI is defined in input scene using color filtering and background removing.
3) Optical Correlator (Cambridge correlator) is used to compare ROI with database of traffic signs. Output of correlation consist of two correlations peaks per match, situated in exactly defined positions. Intensity of correlations peaks is measure of similarity between compared objects.
4) Display serves to caution of driver.

These four main components are used in the whole system.

Block diagram of developed Traffic Sign Recognition System is shown in Fig. 4 [9,11]. It consists of three major functional blocks:

1) Preprocessing,
2) Optical Correlation
3) Traffic Sign Identification

Traffic scene is captured with the CMOS color camera. From traffic video stream the key frame is extracted and then used in next processing. As key frames are referred only PCM coded reference frames from coded video sequence, which usually define also the starting point of any smooth transition in the scene. So key frames can be simply extracted from coded video and processed using PC.

Next processing consists of converting key frame from RGB to HSV color space. The RGB color representation is far from human color concept. Image processing in RGB color representation has several disadvantages: the RGB components depends on light intensity, colors, which for a human vision system might be perceptually close, do not have to be close (in Euclidean distance) to each other in RGB space and smoothly shaded surfaces might correspond to several cluster in a RGB color space [16,17].

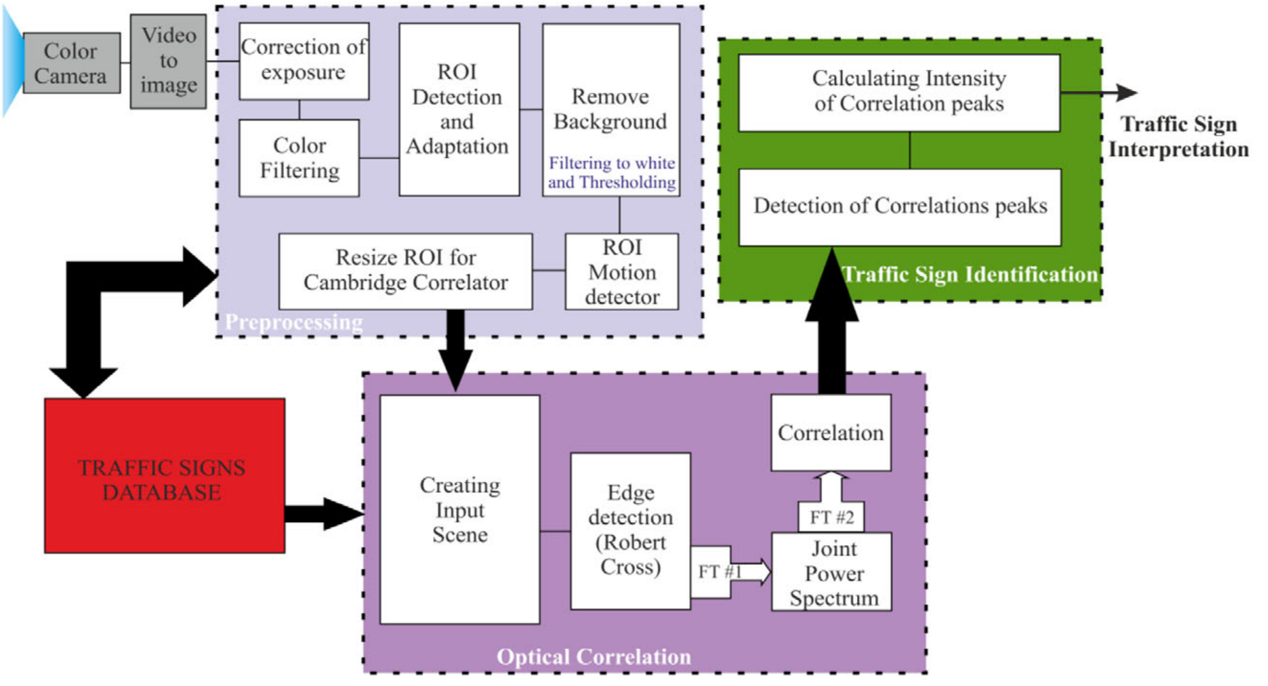


Fig. 4. Block Diagram of Traffic Sign Recognition System with Cambridge Correlator.

This indicates that color filtering in RGB space is very affected by light condition changes in the processed traffic scene. The used HSV color space is much less sensitive in light condition changes than the RGB color space [16,17].

A. *Preprocessing of input scene*

Traffic scenes used in our experiments were obtained from a real road (Fig. 5a). The correction of exposure is required due to different light conditions when the traffic scenes were grabbed [9].

Next step of preprocessing is Color Filtering. In this step the irrelevant color clusters are removed from input traffic scene. Colors filters are designed to remove all colors expect Red, Blue and Yellow. As is mentioned above all input traffic scenes are converted in HSV color space. The color segmentation is processed. Every traffic sign has its dominant color. On the Slovak roads most often Yellow, Red and Blue color are used. This means we need to create three binary maps, one for each of these colors. By analyzing hue component (H), we can identify Blue, Yellow and Red regions in our processed image.

For each image pixel is possible to use hue-based detection for Blue, Red and Yellow color. For each color the following equations are used

$$Y = e^{\frac{-(x-42)^2}{30^2}} \tag{1}$$

$$R = e^{\frac{-x^2}{20^2}} + e^{\frac{-(x-255)^2}{20^2}} \tag{2}$$

$$B = e^{\frac{-(x-170)^2}{30^2}} \tag{3}$$

Equations (1) gives values for Y close to 1 for Yellow regions, Eq. (2) gives values for R close to 1 for Red regions and Eq. (3) gives values for B close to 1 for Blue regions. In this equations we can see, that H can be in the range of 0-255 [9,10,16]. Yellow can be detected near value 42, Red near values 0 and 255 and Blue value is 170. These equations can be tuned for every other needed color.

Saturation detection value is described by the following equation

$$S = e^{\frac{-(x-1255)^2}{115^2}} \tag{4}$$

From equations (1), (2) and (3) we got 3 values. Every value is multiplied by S value. This value will be D_n, which means normalized. Values D_n close to zero will be discarded, others are processed with following equation

$$D_n = \begin{cases} 0, if \det < 0,3 \\ \left(\dfrac{\det - 0,3}{0,7 - 0,3}\right)^2 \\ 1, if \det \geq 0,7 \end{cases} \tag{5}$$

After threshold for detected colors three binary maps (Red, Blue and Yellow) are created.

After color filtering, just relevant colors are available. If the "RED" filter is used, bitmap looks as in Fig. 5b. Images after color filtering also contains color clusters, which are irrelevant for detecting traffic signs. In the next step, the irrelevant color clusters are deleted (Fig. 5b,c). Too small regions and too big regions are discarded. In the first step white point in image is found. Searching is done by rows. After finding the first white point, a method called seed-fill is used. With this method we are finding regions. If the region's size is lower than 400 pixels, then it is discarded. If the region has more than 16000 pixels then it is also discarded [9,10,11,16].

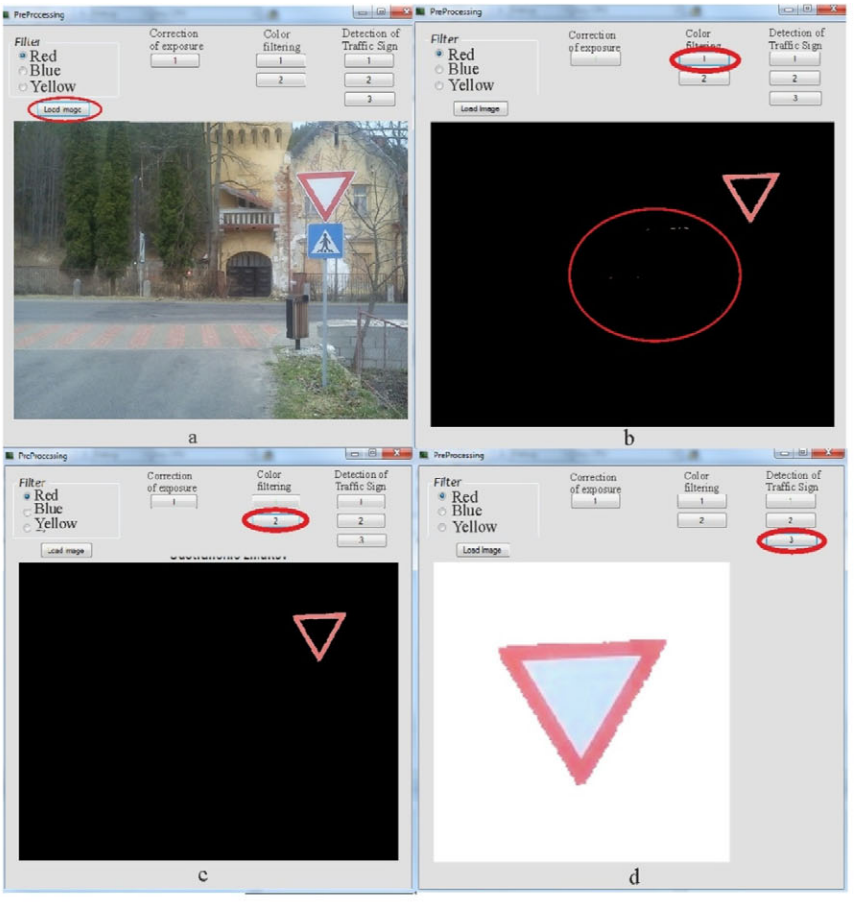


Fig. 5. Red Color Filter action.

After color filtering and irrelevant cluster removing, the potential Region of Interest (ROI) is extracted (Fig. 5b,c,d). The potential ROI is defined at the input image.

ROI crop input image to chosen traffic sign. The ROI Motion Detector uses moving vectors to discard moving features from traffic sign on moving object, i.e. on the vehicle.

The background is removed from input image and then the traffic sign without background is resized to correct size for input to the Optical Correlation (Fig. 5d).

B. Optical Correlation

Optical correlation between input Traffic Sign and chosen reference Traffic Signs is done with using Cambridge Correlator. Due to principle of JTC, the input Traffic Sign and reference sings are aligned to each other in input correlation plane. Joint Transform Correlation consists of three main steps [12,13,14].

First of all, the input correlation plane has to be made as described in Fig. 6a. Input scene is Fourier-transformed to produce Joint Power Spectrum - JPS (Fig. 6b) and then this JPS is pre-processed (mostly threshold is used – Fig. 6c). Pre-pre-processed JPS is Fourier-transformed again to produce optical correlation (Fig. 6d).

Optical correlation consists of pairs of correlation peaks per match. Position and intensity of correlations peaks give information about which Traffic Sign from input scene was recognized [14,15].

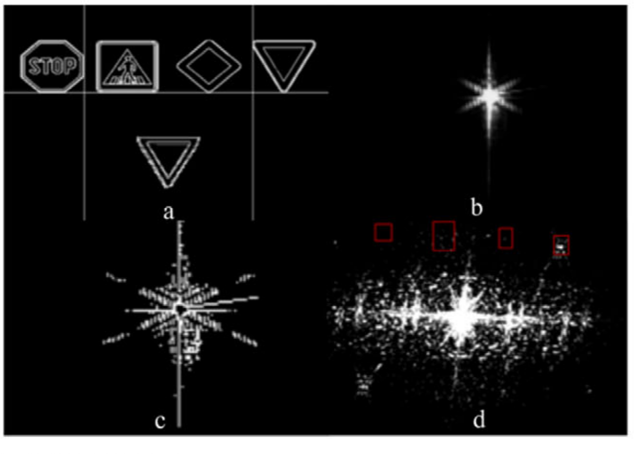


Fig. 6. Correlation process.

C. Traffic Sign Identification

Correlation peaks are monitored in exactly defined areas. If correlation peaks are situated in a defined area, traffic sign assigned to this area was recognized. In the case of Fig. 6, four types of Traffic Signs were inserted in input plane, so four ROIs were monitored in correlation plane.

As shown in Fig. 6, the highest intensity was detected in ROI#4, so in this position was recognized Traffic Sign (Yield).

IV. EXPERIMENTS AND RESULTS

A. Experiments

In our experiments, we used Traffic Sign Recognition System to warning traffic signs. Four types of warning Traffic Signs were chosen. All Traffic scenes were captured in real traffic on Slovakia roads. Tested signs were "Bend to left", "Bent to right" "Double Bend First to Right" and "Double Bend First to Left". Traffic Signs used in experiment are shown in Fig. 7.

Fig. 7. Tested Traffic Signs.

After preprocessing described above, information about color is irrelevant so after background removal we use edge detection to minimize energy of DC noise in correlation process. For edge detection, the Robert Cross operator were used.

Using edge detection, just relevant information for correlation process left and correlation peaks in correlation plane are easily detectable. In this case the four types of traffic signs are tested so monitored areas in correlation plane are situated similarly, as shown in Fig. 6.

As we can see, all tested traffic signs belong to same group of traffic signs (Warning Traffic Signs).

Fig. 8. Symbols of chosen Traffic Signs.

Each traffic sign contains red triangle which is necessary in detecting the traffic sign by using color filtering. However this triangle is adverse in correlation process because match was detected between each of traffic signs.

This triangle was in remove background process marked as background. As input image to correlation process was marked just symbol inside triangle (Fig. 8). This method can be used in many other cases, when are tested Traffic Signs from same category, e.g. Speed Limit traffic signs.

Traffic Signs were captured with ideal and low light conditions, mostly fog. Weak light conditions influence on detection and recognition part of system. In traffic scenes captured with lower visibility were correctly recognized less traffic signs, than in traffic scenes captured with ideal light conditions (ideal visibility).

Example of traffic scene with low visibility is shown in Fig. 9. If traffic scene is captured with low visibility, the two cases can occur. Traffic scene captured with weak visibility has less saturation and red (or other) color is not exactly defined.

Fig. 9. Traffic Scene with ideal (a,c) and low (b,d) visibility.

If color will not detect in traffic scene, ROI will not be defined and Traffic Sign recognized (no correlation peak in correlation process). If color will be detected just partially, ROI can be defined imprecisely. In this case there will be correlation peak in correlation plane with lower intensity than with traffic scene with ideal visibility.

Intensity of peaks obtained by correlation is represented in 8 bit depth grayscale, so every peak can take values from 0 to 255. Correct setting decision level of intensity of peak is necessary.

In the next experiment, the Speed limit traffic signs were used. Three traffic signs were chosen and tested. All speed limits traffic signs contains red circle, so in preprocessing the "RED" filter was used. Every red circle on the traffic scene was defined as a ROI. After defining ROI, the background was removed. The red circle of the traffic sign was defined as a background, because every speed limit traffic sign contains this circle and correlation is too high between any two signs. Background was deleted first to red color, and then to black color. After this preprocessing, just black symbols inside the red circle were left. As an input to the correlator only numbers 40, 60 and 80 were used. Fig. 10a illustrates the input scene of the correlation process. Output of correlation is illustrated in Fig. 10b. Fig.10c describes the output of the correlation process in more detail. Since the input scene consists of symbols containing number "0", correlation peaks were detected between any two compared symbols. Correlation between the same symbols was much higher, so as a match between symbols "40" was defined.

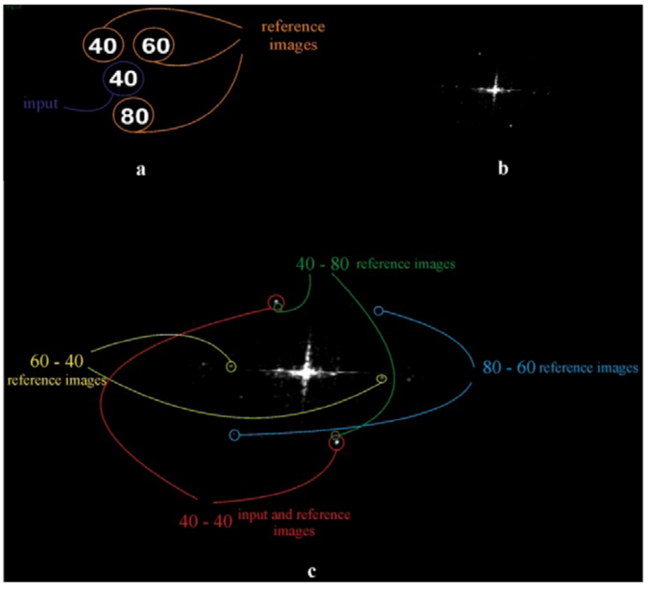


Fig. 10. Compare of similar Traffic signs.

B. Results

The developed Traffic Sign Recognition system was tested on real traffic scenes captured by color CMOS sensors and preprocessed with the algorithm described above. Traffic scenes were captured with ideal and low visibility. Traffic scenes with lower visibility were not recognized because of strong destruction of color. If the color of the traffic sign cannot be determined, no ROI can be defined and the output of the correlation does not contain any correlation peaks.

Results of experiments are recorded in Table I.

TABLE I
RESULTS OF EXPERIMENTS

Traffic Sign	Average Recognition Success [%]		
	Ideal visibility	Low visibility	Entire Recognition Success
Bend to left (35)	92	83	87
Bend to right (23)	94	86	90
Double bend first to right (32)	91	86	88
Double bend first to left (30)	90	83	86
Speed Limit “40” (10)	92	88	90
Speed Limit “60” (12)	90	88	89
Speed Limit “80” (10)	93	89	91

V. CONCLUSIONS

Nowadays the Driver Assistance System is a necessary part of almost each type of automobiles. DAS can consist of many parts which can actively or passively assist the driver when driving a car. In this paper one part of assistance system, the Traffic Sign Recognition System was presented based on Optical Correlator.

The system recognizes Traffic Signs from captured traffic scenes by color CMOS camera. The major information about basic color model was described. Block diagram of the whole Traffic Sign Recognition System was shown. The block diagram consists of three main parts, which can be independently modified, supplemented and improved.

Color filters were designed and used to detect relevant colors of traffic signs. In our experiments we used red color filter to detect red traffic signs. In these experiments the warning traffic signs were tested. ROIs were detected using these filters and other processing like irrelevant cluster removing etc. (Fig. 4). Experiments were done with four types of warning traffic signs. In experiments real traffic scenes were used captured at low and ideal light conditions (foggy and sunny days). The results of experiments demonstrate that the main advantage of this system is the usage of the optical correlator. Very fast processing time can save a lot of time and the driver can be earlier warned. Recognition of traffic signs using optical correlator is much faster than other pure software implementations running on PC. There are many Traffic Sign Recognition Systems mostly based in shape detection and color segmentation such as Automatic Detection and Classification of Traffic Signs [5], Real Time Road Signs [2] and Traffic Sign Recognition System [3,6]. All of these systems are based on pure software realization, so using hardware implementation can speed up process of recognition. The proposed system uses detector of traffic signs similarly as in other systems, but the process of recognition is done using optical processor so it is performed in real time. Processing time of Driver Assistance System using optical correlator is limited just with detection part, which can be improved using FPGA modules.

Using Optical Correlator in Driver Assistance System is not limited just to Traffic Signs Recognition System. Using optical processing can be very helpful in systems for detection of road lane markings as well. Thanks to fast processing time of optical processors the application of Optical Correlator in many recognition systems is very attractive and requested.

ACKNOWLEDGMENT

We support research activities in Slovakia / Project is co-financed from EU funds. This paper was developed within the Project "Centrum excelentnosti integrovaného výskumu a využitia progresívnych materiálov a technológií v oblasti automobilovej elektroniky", ITMS 26220120055.

REFERENCES

[1] J. C. Noyer, P. Lanvin and M. Benjelloun, “Correlation-based particle filter for 3D object tracking,” *Integrated Computer-Aided Engineering*, 16:2, pp. 165-177, 2009.

[2] A. Broggi, P. Cerri, P. Medici and P.P. Porta, “Real Time Road Signs Recognition,” *Intelligent Vehicles Symposium*, 2007 IEEE Volume, Issue, 13-15, 981 – 986, june 2007.

[3] M. Fifik, J. Turán and Ľ. Ovseník, “Traffic Signs Recognition Experiments with Transform based Traffic Sign Recognition System,” *17th Symposium IMEKO TC 4, 3 rd Symposium IMEKO TC 19 and 15 th IWADC Workshop*, Instrumentation for the ICT Era, September 8 – 10, 2010, Kosice, Slovakia.

[4] L. Luo and X. Li "A method to search for color segmentation threshold in traffic sign detection," Fifth International Conference on Image and Graphics, 2009.

[5] F.P. Paulo, L.P. Correia, "Automatic detection and classification of traffic signs", *Image Analysis for Multimedia Interactive Services,* WIAMIS, 6-8 June 2007.

[6] M. Fifík, J. Turán, Ľ. Ovseník and K. Fazekas, "Experiments with a Transform based Traffic Sign Recognition System"*, Proc. Of 17th International Conference on System, Signals and Image Processing IWSSIP 2010, Rio de Janeiro, Brazil,* June 17-19, 2010, 227-230.

[7] K. Kastner, T. Michalke, T. Burbach, J. Fritsch and Ch. Goerick, "Attention – based traffic sign recognition with an array of weak classifiers,", IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA, 2012.

[8] P. Paclík, J. Novovičová, P. Pudil and P. Somol "Road Sign Classification using Laplace Kernel Classifier," Czech Technical University, Prague, Czech Republic.

[9] T. Harasthy, J. Turan, Ľ. Ovsenik, K. Fazekas, "Optical Correlator Based Traffic Signs Recognition"*, Proc. IWSSIP 2012, Vienna, Austria,*11-12 April 2012, 142-145.ISBN: 978-3-200-02328-4, http://webcache.googleusercontent.com/search?q=cache:YS8e5KzC8fIJ:edas.info/web/iwssip2012/program.html+optical+correlator+based+traffic+signs+recognition+ovsen%C3%ADk&cd=2&hl=sk&ct=clnk&gl=sk

[10] T. Harasthy, J. Turan, Ľ. Ovsenik, K. Fazekas, "Traffic Signs Recognition with Using Optical Correlator"*, Proc. IWSSIP 2011, Sarajevo, Bosnia and Herzegovina,*16-18 June 2011, 239-242.ISBN: 978-9958-9966-1-0,IEEE Catalog number: CFP1155E-PRT.

[11] Ľ. Ovseník, T. Harasthy and J. Turán, "Video Driver Assistance Sytem Using Optical Correlator," 35th Jubilee International Convention on Information and Communication Technology. Electronics and Microelectronics, 21-25 May 2012, Opatija/Abbazia, Adriatic Coast, Croatia.

[12] J. W. Goodman, *Introduction to Fourier Optics*, Colorado: Roberts & Company, Publishers, 186 s. ISBN 0-9747077-2-4, 2005.

[13] T. Naughton, M. Klima, J.Toth, *Improved Joint Transform Correlation Performance Through Spectral Domain Thresholding* [online]. Maynooth, Ireland, 1998,

[14] Cambridge Correlators, http://www.cambridgecorrelators.com, June 2012.

[15] Cambridge Correlators, *Fourier Optics Experimenter*, User Guide.

[16] S. E. Umbauch, *Computer Vision and Image Processing and A Practical Approach Using CVIPtools,* Prentice Hall PTR, Upper Saddle River, NJ 07458, June 1999.

[17] Q. Ye, W. Gao and W. Zeng "Color Image Segmentation Using Density-Based Clustering," ICASSP 2003, China, 2003

[18] L. Luo, X. Li "A method to search for color segmentation threshold in traffic sign detection," Fifth International Conference on Image and Graphics, 2009.

BIOGRAPHIES

T. Harasthy (Ing.) received Ing. (MSc.) degree in 2010 at Department of Electronics and Multimedia Telecommunications, Faculty of Electrical Engineering and Informatics of Technical University of Košice. Since September 2010 he has been at University of Technology, Košice as PhD. student. His research interests include object recognition systems.

Ľ. Ovseník (doc., Ing., PhD.) received Ing. (MSc.) degree in radioelectronics from the University of Technology, Košice, in 1990. He received PhD. degree in electronics from University of Technology, Košice, Slovakia, in 2002. Since February 1997, he has been at the University of Technology, Košice as Associate Professor for electronics and information technology. His general research interests include optoelectronic, digital signal processing, photonics, fiber optic communications and fiber optic sensors.

J. Turán (Dr.h.c., prof., RNDr., Ing. DrSc.) received Ing. (MSc.) degree in physical engineering with honours from the Czech Technical University, Prague, Czech Republic, in 1974, and RNDr. (MSc.) degree in experimental physics with honours from Charles University, Prague, Czech Republic, in 1980. He received a CSc. (PhD.) and DrSc. degrees in radioelectronics from University of Technology, Košice, Slovakia, in 1983, and 1992, respectively. Since March 1979, he has been at the University of Technology, Košice as Professor for electronics and information technology. His research interests include digital signal processing and fiber optics, communication and sensing.

Software Application for QoS Characteristics Calculation

Vladimír Hottmar and Bohumil Adamec

Abstract — **This paper deals with the algorithmization of a software application intended for QoS statistical parameters and characteristics quantification. Empirical probability distribution of packet delays and packet jitters, average values of delays and jitters will be determined. Practical software implementation of QoS calculation methodology in Delphi environment is shown. Subsequently, in the context of the currently used QoS mechanisms, the practical use of obtained results is described. Created software application was tested on a real network segment with RTP voice, RTP video, SMTP and FTP traffic streams. The operating parameters (queue length, proportions of allocate bandwidth) of implemented QoS mechanisms on this network can be optimized based on the results obtained by application.**

Index Terms — **Packet delay, packet jitter, empirical probability distribution of packet delay, empirical probability distribution of packet jitter, Quality of Service – QoS, Voice over IP - VoIP**.

I. Introduction

CURRENTLY, there are many different emulators of network traffic, but not all of them provide detailed statistical characteristics of the transported packet delays and jitters. However, for a detailed analysis of conditions in a network or for a precise qualitative analysis of operation of the potentially implemented QoS tools, it is necessary to know the empirical probability distribution of packet delays, packet jitters and other additional statistical parameters and characteristics [1], [2].

Because modern traffic emulators that have been used for the emulation purposes do not provide detailed statistical characteristics and information about transported data traffic it was necessary to create additional software application. Most professional traffic emulators provide either globally statistical characteristics or incomplete statistical parameters. Results of modern emulators usually do not include required statistical parameters and characteristics that relate to the individual data streams. Therefore, it was necessary to create auxiliary application that allows obtaining detailed statistical information about individual traffic streams. Created software application is closely related to a research project that deals with the examination of the QoS tools and optimal implementation of these tools to the modern converged packet networks.

For a detailed verification of the results from the research process it was necessary to create described software application. Therefore, described software application is a specific software tool that is not normally described in scientific articles. However, the application can also be used in practice for the purpose of optimal QoS implementation according to the specific customer requirements – this is actually a novelty. A software application in object-oriented programming language Delphi 7.0 was created for purposes of calculation and graphical interpretation of the empirical distributions of either packet delays or packet jitters. Mentioned application is described in this article. Created software application uses two data files containing captured data traffic in two different network nodes, which are usually located on the edge of the analyzed network (Fig. 1).

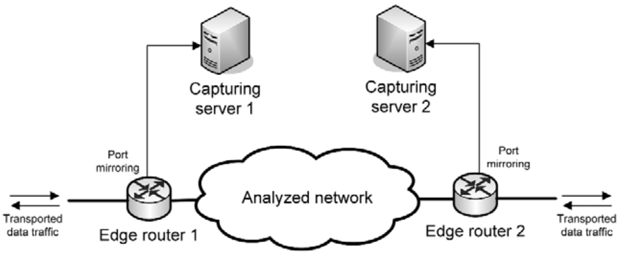


Fig. 1. Located of the network nodes for data traffic analysis.

Obtained statistical characteristics determine on what qualitative level analyzed network segment is able to transport the data traffic. From captured data files is possible to calculate both basic statistical parameters (e. g. average packet delay, average packet loss, average packet jitter) and additional statistical characteristics (e.g. empirical distribution function of delays, maximum delay, minimum delay and so on). All of these statistical parameters and characteristics may relate to packets belonging to a specific data flow that belonging to a specific application (e. g. voice traffic, video traffic and so on) or may relate to the transported packets generally. In the case when all of these characteristics are related

Manuscript received June 18, 2012, revised September7. This project was supported by the VEGA grant no. 1/0375/08 of Ministry of Education of the Slovak Republic.

V. Hottmar is with the Department of Telecommunications and Multimedia, Faculty of Electrical Engineering, University of Žilina. Univerzitná 8215/1, 010 26 Žilina, Slovakia (e-mail: vladimir.hottmar@fel.uniza.sk).

B. Adamec is with the Department of Telecommunications and Multimedia, Faculty of Electrical Engineering, University of Žilina. Univerzitná 8215/1, 010 26 Žilina, Slovakia (e-mail: bohumil.adamec@fel.uniza.sk).

to the transported data traffic irrespective of the specific data flows of which it consists, we denote these quantities as global characteristics. A more detailed overview about conditions in the network reflects specific characteristics which are related to the specific data flow. On the basis of the results obtained from the created software application it is possible to examine the qualitative level of traffic processing by analyzed network segment and the overall transported data traffic composition.

Qualitative level of data traffic processing can be examined on the basis of the set of statistical parameters and characteristics. These statistical coefficients and characteristics that are related to a particular application indicate the number of discarded packets, maximum, minimum and average packet delay, packet jitter and other basic characteristics. Furthermore, for a detailed data traffic processing review, it is possible to determine the number of packet delays or jitters falling within a certain time interval that is located in the time range between minimum and maximum observed packet delay or packet jitter. The number of these time intervals is a user-defined parameter and the user can adjust it according to his/her needs. In this way the user is able to vary the density and hence the accuracy of the empirical probability distribution.

Another part of created software application allows for determining the composition of the transported data traffic. Based on the transported data traffic composition results it is possible to determine what kind of packets and in what percentage proportions are transported by the network. For instance, according to these results, suitable QoS tool set can be implemented. Individual QoS traffic classes based on of data traffic composition results can be defined and created by the user. Classes are designed to integrate different data streams with identical QoS requirements. The operating parameters for each class can be set according to data traffic composition results and subsequently, these parameters can be corrected according to full statistical results obtained by created software application.

II. Description of Created Software Application

Inputs of the created software application are two data files containing information about the packets captured at two different network nodes. All significant QoS characteristics of network traffic are calculated for interactive RTP voice and video traffic and non-interactive FTP and SMTP data traffic. Packet identification is based on a unique identifier of each packet. Identification of a packet passing through different network nodes is realized using the following parameters:

- Protocol Info – type of packet that is transported in the specific frame in application layer,
- Time – capture moment of the frame corresponding to a specific network node,
- Identification – packet number derived from network layer,
- Protocol Mark – this is Payload Type and Sequence Number derived from application layer in the case of RTP packet or Header Checksum derived from network layer in the case of SMTP or FTP packet.

The described software application creates two groups of arrays. The first array group relates to the server side – this array group is labeled as server. The second array group relates to the client side – this array group is labeled as client. Both array groups consist of four specific sub-arrays that contain the needed parameters and information for the traffic analysis. The first sub-array is labeled as Protocol and specifies the packet type. Based on this information the application decides whether the specific packet is processed or not. The second array is labeled as Time and contains time moments of packet transmission and packet reception. Time characteristics of transported data traffic such as delay, jitter, empirical probability distribution of delay and jitter are calculated based on these parameters. The third array is labeled as ID – Identification and contains identification numbers of all the analyzed packets. These numbers are used for packet identification purposes and to determine concrete packet that passes through different network nodes. The fourth array is labeled as Mark and contains other additional identifiers to uniquely identify transported packet passing through different network nodes.

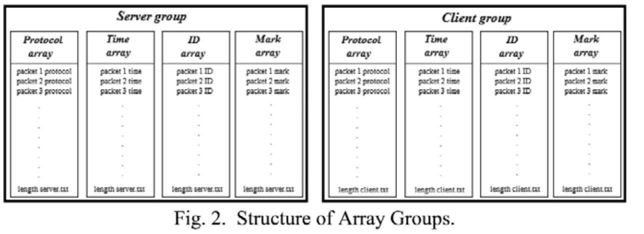


Fig. 2. Structure of Array Groups.

Fig. 2 shows the arrays concept described above.

The size of individual arrays is given by the size of application input data files. Application input data files size depends on the volume of captured data traffic to be analyzed. Specific arrays in the server array group may not have the same number of components as the corresponding specific array in the client array group. This phenomenon is caused mainly by the packet loss in overload network nodes.

However, it is necessary to ensure the same dimensions of all arrays within the individual array group. The same array sizes within individual array group ensure that the comparison and parameter calculation will be done with the data of same packet that was captured in various network nodes.

The created software application provides an overview of the analyzed protocol packets (RTP, FTP, and SMTP) as well as an overview of the complete transported data traffic packets. The software application subprogram result is the total number of transported data packets regardless of their type and number of data packets that are not analyzed such as ARP, EIGRP. Calculation of the parameters mentioned above realizes subprogram labeled as parameter parsing server and subprogram labeled as parameter parsing client (see Fig. 3).

The basic QoS parameter is the average delay of transported data packets. Packet delay is defined as the difference between the time moment of packet reception and

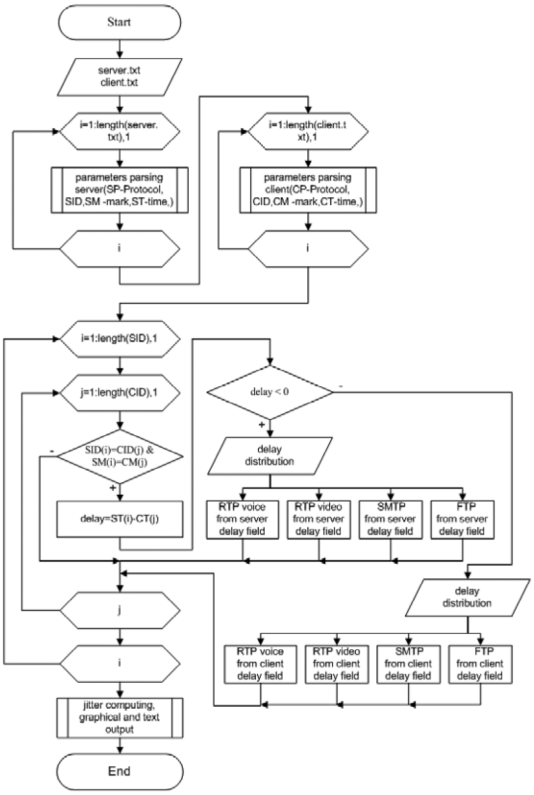


Fig. 3. Software Application Flow Diagram.

time moment of packet transmission. Time moments needed to calculate the delay values are stored in the Month/Day/Year/Hour/Minute/Second/millisecond form.

Subsequently, the values of time moments are converted to integer values that represent these time moments in milliseconds. Next, the packet delays related to particular packets are calculated. Calculated delays are stored in the specific array based on the protocol type. The additional traffic characteristics (average delay, minimum and maximum delay) are calculated based on these parameters. For the calculation of all the statistical parameters and characteristics conventional procedures and formulas were used that are not the main subject of this article. The computational methodology used is described in detail in [2], [5].

The structure of packet delay arrays is show in Fig. 4. Consequently, the delay jitter is calculated. Delay jitter is defined and calculated as the average value of the differences between packet delays. All calculated values are displayed in a numerical and graphical form. Graphical design of result form is show in Fig.5.

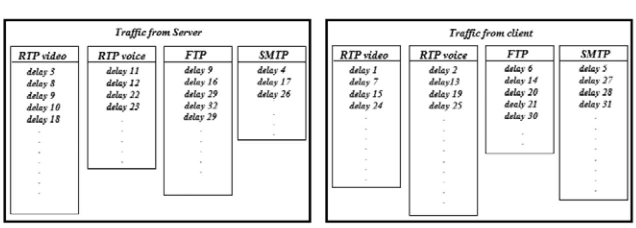


Fig. 4. Packet Delay Arrays Organization.

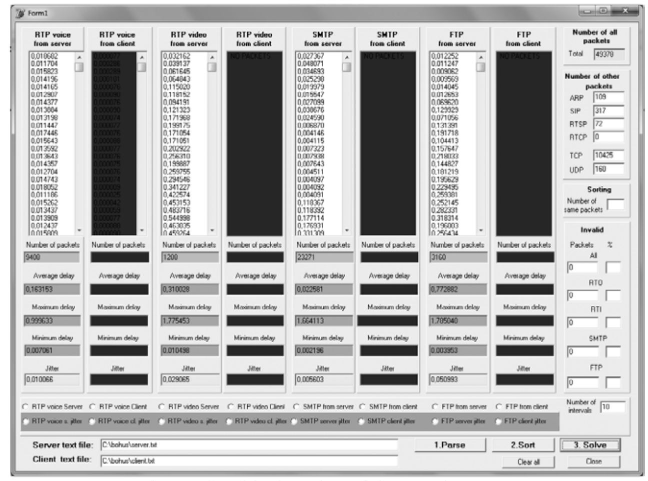


Fig. 5. Graphical Design of the Results Form.

III. Practical Application of the Software Tool

Before the practical implementation of QoS in a real network segment it is necessary to analyze the transported data traffic [3], [4]. In particular it is necessary to determine which types of packets are transported. Next it is necessary to determine percentage proportion of specific packets in the overall transported traffic [5], [6]. This analysis can be realized by the created software application. Some results of concrete traffic analysis are shown in Fig. 6.

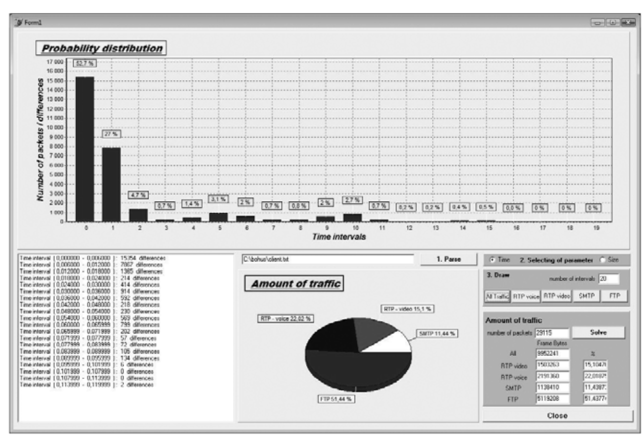


Fig. 6. Results of Traffic Analysis.

This figure shows empirical probability distribution of time differences between time moments of capture of two immediate packets and percentages of specific transported packets belonging to the individual applications. Based on these results it is possible to decide which packets should be processed as a priority [4], [5]. Priority processing of transported traffic can be realized by the different QoS tools (WFQ, CBWFQ and LLQ). Implementation of these mechanisms requires setting their operational parameters – queue length and bandwidth proportion belonging to the specific traffic class [2], [3], [4].

Quantification of operational parameters can be realized based on a mathematical model. Application of the mathematical model requires specific statistical parameters (average packet size, load coefficient and so on) that can be obtained from the results of described software tool. The implementation of the mathematical model and the created software tool has been tested and successfully implemented on a real network segment.

IV. Conclusion

The created software application has been used in the real network segment in the context of optimal application of QoS mechanisms. Based on the results obtained from the created software tool, the appropriate QoS mechanisms have been implemented. Implementation of QoS mechanisms has been performed on the basis of the mathematical model and statistical parameters of transported data traffic obtained from the software tool. In this way, it is possible to apply the QoS mechanisms by the deterministic method and not by the trial and error way. Based on the results of the analytical model and created software application the QoS tools implementtation can be optimized [5]. In order to meet the predefined user requirements the operating parameters of applied QoS tools can be optimally adjusted – this fact can be considered as the main benefit.

Compared with commercially available applications, the created software tool provides additionally the detailed statistical analysis of transported data traffic. It provides the basic statistical parameters (total number of transported packets and bytes, total number of transported packets and bytes related to a specific protocol, percentage of transported packets and bytes related to a specific protocol) and additional parameters and relations (empirical probability distribution of packet size, empirical probability distribution of packet size related to a specific protocol, empirical probability distribution of time differences between time moments of capture of two immediate packets and so on).

These statistical parameters and characteristics are very important mainly for optimal application of QoS mechanisms. Furthermore, in addition to average and peak values of packet delays and jitters application also provides the empirical probability distribution of these characteristics. The described application could be extended to the other protocols and applications.

Acknowledgment

This project was supported by the VEGA grant no. 1/0375/08 of Ministry of Education of the Slovak Republic. The results presented in this article are part of the solutions of this VEGA grant. The results of this VEGA grant have also been incorporated in the dissertation thesis of one of the authors.

References

[1] Liu, X., Ravindran, K., Loguinov, D., "A queueing-theoretic foundation of available bandwidth estimation: single-hop analysis," IEEE/ACM Transactions on Networking, vol. 15, no. 4, pp. 918–931, 2007.
[2] Hock, N. Ch., Hee, S. B., "Queuing Modelling Fundamentals with Applications in Communication Networks," New York: J. Wiley and Sons, Inc., 2008, ISBN: 978-0-470-51957-8.
[3] Odom, W., Cavanaugh, M., J., "Cisco QoS – Exam Certification Guide", Indianapolis: Cisco Press, 2005, ISBN: 1-58720-124-0.
[4] Chaskar, H. M., Madhow, U., "Fair scheduling with tunable latency: a Round Robin approach," in Proceedings of the IEEE Global Telecommunication Conference (GLOBECOM '99), pp. 1328–1333, December 1999.
[5] Adamec, B., "Design of optimization model for data communication management in Campus LAN", PhD. dissertation thesis, Department of Telecommunications and Multimedia, Faculty of Electrical Engineering, University of Zilina, Zilina, Slovakia, 2010 (in Slovak).
[6] C. W. Chan, Performance Analysis of Telecommunications and Local Area Networks, Kluwer Academic, Norwell, Mass, USA, 2000.

Vladimír Hottmar graduated from the University of Zilina in 1975 with the Master's degree in Transport and Communications. At the beginning of his career he was employed at the Research Institute of Computers Techniques in Zilina, where he worked as a researcher in the Department of Microcomputers and Development Systems for 22 years, later being responsible for running the Research Department. Currently he is employed at the University of Zilina in the Department of Telecommunications and Multimedia in the position of an associate professor and is also involved in some research work. He is a project leader of the project VEGA. In 1999 he accomplished his PhD studies at the University of Zilina. Since May 2006 he has worked on the post of an associate professor at the University of Zilina.

Bohumil Adamec graduated from the University of Zilina in 2007 with the Master's degree in Telecommunications. Currently he is employed at the University of Zilina in the Department of Telecommunications and Multimedia in the position of an associate professor and is also involved in some research work. His range of interest is home networking especially simulation and mathematical modeling of modern multimedia home networks.

DRCN 9th International Conference on Design of Reliable Communication Networks | March 4-7, 2013 Budapest, Hungary

www.drcn2013.org

CALL FOR PAPERS

DRCN is a well established biennial forum for scientists, engineers, designers and planners from industry, government and academia who have interests in the reliability and availability of communication networks and services. The conference covers topics from equipment and technology for survivability to network management and public policy, through theory and techniques for survivable and robust networks and application design. The aim of the conference is to bring together people from industry, government and academia in those disciplines in a lively forum.

To guarantee the high visibility of the conference, the proceedings will be available through IEEE Xplore. The best papers will be invited to leading related Journals: IEEE Transactions on Reliability (ToR) and Elsevier Optical Switching and Networking (OSN).

The 9th DRCN will be held in the attractive city of Budapest, Hungary. The conference will be enriched by a set of tutorials and invited talks. Additionally, two IFIP best paper awards will be granted. We seek papers that address theoretical, experimental, systems-related and regulatory issues in the area of dependability and survivability of communication networks, end-systems and infrastructure. Topics of interest include, but are not limited to the following areas:

- Operational aspects:
  - Fault management, monitoring, and control
  - Methodologies, equipment and technology for network survivability
  - Survivability of optical and multi-layer networks
  - Reliability of wireless access and mesh networking
  - Resilient wired access networks
  - Dependability of cellular/mobile networks including horizontal handover
  - Resilience of multi-domain connections in the Internet
  - Reliability of emerging technologies (e.g. network virtualization, thin client architectures)
  - Survivability in grid and distributed computing
  - Network dependability in cloud computing
  - Management of survivable networks
- Theory and modelling:
  - Network reliability analysis
  - Methods and theory for survivable network and systems design, analysis and operation (including scalability and complexity)
  - Planning and optimization of reliable networks, systems, and services
  - Simulation techniques for network resilience
- Services:
  - Reliability requirements and metrics for individual users, businesses, and the society
  - Restoration of services under various types of failures
  - Service differentiation based on recovery methods
  - Dependability of networked applications
  - Recovery of overlay and peer-to-peer networks
  - Application and service-specific survivability techniques
  - Survivability of multimedia networks including voice over IP, IPTV, and content delivery
  - Reliability and resilience of data centre networks
  - Robustness of compound services
- Broad context:
  - Telecommunication networks as an element of critical national infrastructures
  - Public policy issues for survivability and resilience
  - Standardization of network resilience and reliability
  - Network resilience combined with economics and commercial issues
  - Quality of experience and network survivability
  - Security issues in networks and their relation to survivability
  - Dependability and energy consumption trade-offs
  - Risk and reliability in the Internet and enterprise networks
  - New and emerging threats

DRCN 9th International Conference on Design of Reliable Communication Networks

March 4-7, 2013
Budapest, Hungary

www.drcn2013.org

IMPORTANT DATES

Paper registration and abstract:	November 16, 2012
Submission of papers for review:	**November 23, 2012**
Acceptance notifications:	January 18, 2013
Camera-ready due:	February 8, 2013
Conference:	March 4-7, 2013

AUTHOR GUIDELINES

All submissions should be written in English with a maximum paper length of eight (8) double-column printed pages (10-point font) including figures. All papers must be formatted according to the IEEE manuscript template for conference proceedings. Only PDF files will be accepted for the review process and all submissions must be done through EDAS.

STEERING COMMITTEE

Piet Demeester, *(Steering Committee Chair), Ghent University - IBBT -IMEC, Belgium*

Prosper Chemouil, *Orange Labs, France*

Tibor Cinkler, *Budapest University of Technology and Economics, Hungary*

Roberto Clemente, *Telecom Italia, Italy*

Robert Doverspike, *AT&T Labs, USA*

Wayne D. Grover, *TRLabs, University of Alberta, Canada*

Deep Medhi, *University of Missouri-Kansas City, USA*

Ken-ichi Sato, *Nagoya University, Japan*

Dominic Schupke, *Nokia Siemens Networks, Germany*

David Tipper, *University of Pittsburgh, USA*

TECHNICAL CO-SPONSORS

IFIP *(pending)*
HTE: *Hungarian Scientific Association for Infocommunications (Sister Society of IEEE)*
BME: *Budapest University of Technology and Economics*
University of Waterloo
EC FP7 IP COMBO *(pending)*

TPC CHAIRS

Pin-Han Ho, *University of Waterloo, Canada*
János Tapolcai, *Budapest University of Technology and Economics, Hungary*

GENERAL CHAIR

Tibor Cinkler, *Budapest University of Technology and Economics, Hungary*

Guidelines for our Authors

Format of the manuscripts

Original manuscripts and final versions of papers should be submitted in IEEE format according to the formatting instructions available on

http://www.ieee.org/publications_standards/publications/authors/authors_journals.html#sect2,

"Template and Instructions on How to Create Your Paper".

Length of the manuscripts

The length of papers in the aforementioned format should be 6-8 journal pages.

Wherever appropriate, include 1-2 figures or tables per journal page.

Paper structure

Papers should follow the standard structure, consisting of *Introduction* (the part of paper numbered by "1"), and *Conclusion* (the last numbered part) and several *Sections* in between.

The Introduction should introduce the topic, tell why the subject of the paper is important, summarize the state of the art with references to existing works and underline the main innovative results of the paper. The Introduction should conclude with outlining the structure of the paper.

Accompanying parts

Papers should be accompanied by an *Abstract* and a few *index terms (Keywords)*. For the final version of accepted papers, please send the *short cvs* and *photos* of the authors as well.

Authors

In the title of the paper, authors are listed in the order given in the submitted manuscript. Their full affiliations and e-mail addresses will be given in a footnote on the first page as shown in the template. No degrees or other titles of the authors are given. Memberships of IEEE, HTE and other professional societies will be indicated so please supply this information. When submitting the manuscript, one of the authors should be indicated as corresponding author providing his/her postal address, fax number and telephone number for eventual correspondence and communication with the Editorial Board.

References

References should be listed at the end of the paper in the IEEE format, see below:

a) Last name of author or authors and first name or initials, or name of organization
b) Title of article in quotation marks
c) Title of periodical in full and set in italics
d) Volume, number, and, if available, part
e) First and last pages of article
f) Date of issue

[11] Boggs, S.A. and Fujimoto, N., "Techniques and instrumentation for measurement of transients in gas-insulated switchgear," IEEE Transactions on Electrical Installation, vol. ET-19, no. 2, pp.87–92, April 1984.

Format of a book reference:

[26] Peck, R.B., Hanson, W.E., and Thornburn, T.H., Foundation Engineering, 2nd ed. New York: McGraw-Hill, 1972, pp.230–292.

All references should be referred by the corresponding numbers in the text.

Figures

Figures should be black-and-white, clear, and drawn by the authors. Do not use figures or pictures downloaded from the Internet. Figures and pictures should be submitted also as separate files. Captions are obligatory. Within the text, references should be made by figure numbers, e.g. "see Fig. 2."

When using figures from other printed materials, exact references and note on copyright should be included. Obtaining the copyright is the responsibility of authors.

Contact address

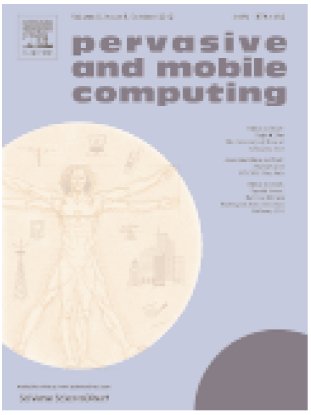
Authors are requested to send their manuscripts via electronic mail or on an electronic medium such as a CD by mail to the Editor-in-Chief:

Csaba A. Szabo
Dept. of Telecommunications
Budapest University of Technology and Economics
2 Magyar Tudosok krt.
Budapest 1117 Hungary
szabo@hit.bme.hu

IEEE PerCom in Budapest in 2014!

The **IEEE Pervasive Computing and Communication** (PerCom) conference is the worldwide premier scholarly venue in the areas of pervasive computing and communications. Since 2003, the conference has grown significantly in terms of quality and variety of the technical programs – it is recognized as a top tier conference by most universities and organizations across the world.

PerCom provides a high profile, leading edge forum for researchers, engineers, and practitioners to present state-of-the-art research in the respective fields of pervasive computing and communications. The conference features a diverse mixture of presentation forums including core technical sessions, keynote talks, panel discussions from worldwide experts, demonstrations, a PhD forum, and work in progress posters. The conference also hosts a number of workshops that have themselves become well recognized in the community as forums for specialized topics within the field.



The conference also provides formats to honor excellence in the field. The Mark Weiser Best Paper award, sponsored by Elsevier, is given to authors of the PerCom's best paper. In addition, the highest quality papers from the conference are published in a special issue of the Pervasive and Mobile Computing Journal.

The IEEE PerCom Steering Committee has recently decided to accept the joint application of the Budapest University of Technology and Economics (BME), Department of Telecommunications and the Scientific Association for Infocommunications (HTE) to organize PerCom 2014 in Hungary. Thus, after the 2013 edition in San Diego, Budapest will host this prestigious conference in 2014. It will be a 5 day-event, with associated workshops, and will hopefully attract several hundreds of participants. More information can be found on the conference's website:

www.percom.org

SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS

Who we are

Founded in 1949, the Scientific Association for Infocommunications (formerly known as Scientific Society for Telecommunications) is a voluntary and autonomous professional society of engineers and economists, researchers and businessmen, managers and educational, regulatory and other professionals working in the fields of telecommunications, broadcasting, electronics, information and media technologies in Hungary.

Besides its more than 1300 individual members, the Scientific Association for Infocommunications (in Hungarian: Hírközlési és Informatikai Tudományos Egyesület, HTE) has more than 60 corporate members as well. Among them there are large companies and small-and-medium enterprises with industrial, trade, service-providing, research and development activities, as well as educational institutions and research centers.

HTE is a Sister Society of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) and the IEEE Communications Society. HTE is corporate member of International Telecommunications Society (ITS).

What we do

HTE has a broad range of activities that aim to promote the convergence of information and communication technologies and the deployment of synergic applications and services, to broaden the knowledge and skills of our members, to facilitate the exchange of ideas and experiences, as well as to integrate and harmonize the professional opinions and standpoints derived from various group interests and market dynamics.

To achieve these goals, we…

- contribute to the analysis of technical, economic, and social questions related to our field of competence, and forward the synthesized opinion of our experts to scientific, legislative, industrial and educational organizations and institutions;
- follow the national and international trends and results related to our field of competence, foster the professional and business relations between foreign and Hungarian companies and institutes;
- organize an extensive range of lectures, seminars, debates, conferences, exhibitions, company presentations, and club events in order to transfer and deploy scientific, technical and economic knowledge and skills;
- promote professional secondary and higher education and take active part in the development of professional education, teaching and training;
- establish and maintain relations with other domestic and foreign fellow associations, IEEE sister societies;
- award prizes for outstanding scientific, educational, managerial, commercial and/or societal activities and achievements in the fields of infocommunication.

Contact information

President: **DR. GÁBOR HUSZTY** • *ghuszty@entel.hu*

Secretary-General: **DR. ISTVÁN BARTOLITS** • *bartolits@nmhh.hu*

Managing Director, Deputy Secretary-General: **PÉTER NAGY** • *nagy.peter@hte.hu*

International Affairs: **ROLLAND VIDA, PhD** • *vida@tmit.bme.hu*

Addresses

Office: H-1055 Budapest, V. Kossuth Lajos square 6-8, Room: 422.

Mail Address: 1372 Budapest, Pf. 451., Hungary

Phone: +36 1 353 1027, Fax: +36 1 353 0451

E-mail: *info@hte.hu*, Web: *www.hte.hu*