Throughput Performance Analysis of the Multipath Communication Library MPT

Béla Almási, Szabolcs Szilágyi

***Abstract*—Although the currently used mobile communication equipments (laptops, tablets, phones) usually have many network interfaces, the Internet communication technology uses only a single communication path for a connection. In this paper we would like to introduce a new architecture, which gives an easy-to-use extension for the current TCP/IP protocol stack and offers the possibility of using multiple paths for the communication, without rewriting the applications. The new architecture was implemented in Linux by a software tool named MPT. A laboratory environment was established for the throughput performance evaluation of the MPT based multipath communication. The measurement results show that the throughput capacity of the single paths can be efficiently aggregated by MPT.**

Keywords—Multipath communication, throughput, tunnel communication, tunnel software.

I. Introduction

The current Internet communication environment allows only a single path for data transmission in a communication session. The single path assumption is quite acceptable for systems, which use a single connection interface, or a “single exit point” to the Internet. On the other hand, a lot of currently used devices have got factory built-in multiple network interfaces: RJ-45 for the wired network, RF interface for the Wi-Fi wireless network connection, and mobile phone data transfer connection interface (e.g. 3G, Edge or HSDPA).

The single path communication technology is not able to use the advantages of the multiple interfaces. The communication performance (e.g. throughput) could be highly improved if the networking environment would support the usage of multiple paths for a communication session.

The IETF RFC 6824 “TCP Extensions for Multipath Operation with Multiple Addresses” (MPTCP, published in January of 2013, see [1]) is a hot research topic today to extend the current TCP implementations for supporting multiple paths.

In this paper we introduce a new architecture, which gives a simple and easy-to-use extension for the current TCP/IP protocol stack to support multiple paths between the communication endpoints: we are able to establish a highly tuned multipath environment which can be used by the traditional applications to perform multipath communication. The new multipath environment was implemented in a software library (and software tool) named MPT. The modeling background and architecture of MPT is totally different from the MPTCP solution: MPT is able to support applications based not only on the TCP protocol, but also UDP can be used by the applications as the transport layer protocol.

The MPT software was developed in a Linux system to build a laboratory environment for the throughput analysis of the single path and the multipath communications. The measurement results showed us that the throughput capacity of the single paths can be summed in the MPT based multipath environment.

The rest of the paper is organized as follows:

The theoretical working mechanism of MPT is described in Section II. The laboratory measurement environment will be discussed in Section III, and some measurement results will be presented in section IV.

II. The Working Mechanism of MPT

The current IP technology uses the IP address and the TCP or UDP port number as the identification of the communication endpoint (socket). The same IP address is used by the routers for path selection during the packet transmission process: the traditional single path TCP/IP protocol stack is not able to distinguish the endpoint host identification address (used for the sockets) and the address used for the path selection. The Multipath TCP architecture (MPTCP, see [1]) uses a new layered architecture (see Fig. 1). The new layer “MPTCP” gives the connection interface for the applications, while the TCP Subflow layer below the MPTCP layer establishes the multipath communication environment. The usage of the TCP protocol at the subflow layer manages the flow control and provides reliability.

The actual MPTCP document (see [1]) outlines that for the

```
                        +-------------------------------+
                        |          Application          |
+---------------+       +-------------------------------+
|  Application  |       |             MPTCP             |
+---------------+       + - - - - - - - + - - - - - - - +
|      TCP      |       | Subflow (TCP) | Subflow (TCP) |
+---------------+       +-------------------------------+
|      IP       |       |       IP      |      IP       |
+---------------+       +-------------------------------+
```

Fig. 1. The TCP and the MPTCP protocol stack

Manuscript received February 20, 2013.

The work was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

The research was supported by the European Union with the help of the GOP-1.3.2-08-2009-0007 program.

B. Almási is with the Faculty of Informatics, University of Debrecen, Debrecen, Hungary (corresponding author, phone: +36-52-512900 ext. 75012; e-mail: almasi.bela@inf.unideb.hu).

Sz. Szilágyi is with the Faculty of Electrical Engineering and Information Technology, University of Oradea, Oradea, Romania (e-mail: sszilagyi@uoradea.ro).

optimal operation application support may be necessary, i.e. the rewriting request of the applications may occur. In this paper we introduce a new multipath communication environment: the MPT software library. The working idea of the MPT tool is based on the idea of the UDPTUN software (see [2]). In this mechanism we create a logical (tunnel) interface on the endpoints, which can be used by the applications for the socket identification. The IP packets transmitted to the tunnel interface by the application are encapsulated into a new UDP segment, which will be sent to the physical interface for transmission. The MPT library offers the possibility to map the packets (coming from the tunnel interface) to multiple physical interfaces dynamically, so offering the multipath communication for the application. There is no need to modify the application in this case, as the application software uses only a single logical interface (the tunnel interface) for the communication. Also, the application may use the UDP transport protocol over the tunnel interface, it is not restricted to the TCP protocol (as it is in the case of MPTCP, see [1]). On the other hand, as the MPT library uses the UDP protocol in the encapsulation process, it will not offer retransmission and flow control services below the tunnel interface.

Although in this paper we investigate the throughput performance of the MPT library, it is easy to see, that it can help to reach more efficient communication in many cases (see e.g. [3]).

The layered architecture of the MPT based multipath communication can be seen in Fig. 2.

The MPT software reads the input packet (IPv4 or IPv6 packet) from the tunnel interface at the sender site. This packet is encapsulated into a new UDP segment, and it is sent out to a path chosen from the multiple paths possibilities. At the receiver site the header of the incoming UDP segment is stripped out, and the data (which is the original packet coming from the sender's tunnel interface) is transmitted to the tunnel interface of the receiver host. The (logical) connection between the tunnel interfaces of the peers is a direct, point-to-point connection.

III. Measurement Environment and Settings

Our goal was to create a test environment, which could accurately show the communication between two hosts placed into different networks and separated by the Internet Cloud. The purpose of the measurement environment is to test the throughput efficiency of the MPT software tool.

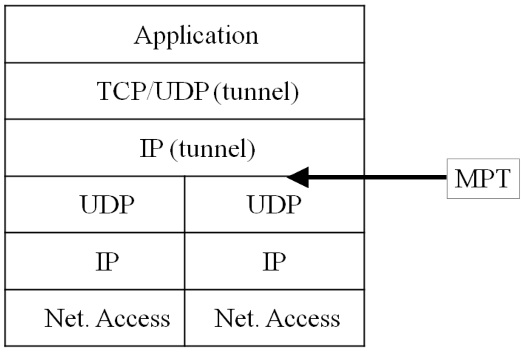

Fig. 2. The layered architecture of MPT

For the measurement we used the network topology, which consists of two hosts, two routers, two Ethernet links from the routers to the hosts and two serial connections between the routers. The serial links between the routers creates two different and independent paths between the hosts. In the measurement we assume that the bandwidth of the serial links is lower than the bandwidth of the Ethernet connection at the endpoints (throughput bottleneck may not occur at the endpoint). The Ethernet link from the host to the router was divided into two logical networks, and a virtual machine environment was used on the endpoint computers to provide two different network interfaces. The virtual host environment was implemented in the Oracle Virtualbox 4.2.6 software. The operating system on the virtual machines was OpenSuse 12.2. The physical computers (used for supplying the real hardware for the virtual machines) contained an Intel Core i5 processor (2.5GHz, 6MB cache), and 8GB memory.

The labor environment was established by using two Cisco 2501 routers with Cisco IOS 12.2. The routers contained an Ethernet and two serial interfaces, which fulfilled the requirements of the test environment.

On both virtual machines two physical (eth1 and eth2) and a logical interface (tun1) had been configured. The default gateways for both machines were set to the Ethernet ports of the respective routers, using two separate IP addresses on the routers' Ethernet interfaces for the two logical networks. The serial links represent the different paths between the endpoints. The bandwidth of the serial links was set independently to each other, by using the appropriate clock rate on the router's DCE interface. On the two routers static routing was implemented (see [4]), which established two different communication paths between the endpoint hosts (see Fig. 3).

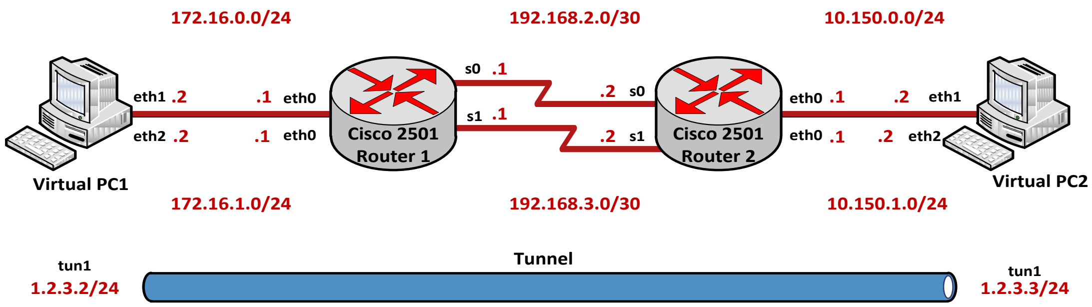

Fig. 3. The measurement environment

For example, the packages coming from PC1's eth1 interface used the destination address of the eth1 interface of PC2, and the packet transmission was performed on the link of the serial 0 interfaces between the routers.

Obviously, the communication between the eth2 interfaces of the two hosts was performed on the other serial link between the routers. The multipath communication performed by the MPT software tool was built on the tun1 logical interface of the hosts. The schema of the labor environment can be seen in Fig. 3. The detailed IP address specification of the labor environment is shown in Table I.

The measurement environment was used to study the networks' throughput using symmetrical and asymmetrical serial connections. In the symmetrical test case both paths used the same bandwidth rate. The clock rate value of 1.300.000 and 2.000.000 cycles per second were chosen, so approximating the speed of the T1 and E1 leased lines. In the real life the bandwidth of the different connection paths can be significantly different (non-symmetrical paths). The case of the non-symmetrical paths was implemented by using non-equal transmission rates on the serial links: the connection of the serial 0 interfaces was set to the clock rate value of 2.000.000 and the link between the serial 1 interfaces was set to the clock rate value of 1.000.000 cycles per second.

The purpose of our research is to investigate the throughput performance of the MPT software tool. Similar research studies for the MPTCP implementation can be found in [5] and [6].

In the next section we present several measurement results, which will proof that the throughput of the multipath environment sums efficiently the throughput of the single paths in the case of symmetrical and asymmetrical paths too.

IV. Measurement Results

In order to test the throughput of the network system three types of measurements were carried out. The first and the second types of the measurements used symmetrical paths with clock rate values of 1.300.000 and 2.000.000 cycles per second respectively. The third type of measurement used non-symmetrical paths: the link between the serial 0 interfaces used the clock rate value of 2.000.000 and the link between the serial 1 interfaces was set to the clock rate value of 1.000.000 cycles per second.

All measurement types contained three cases, which differed in the size of the transmitted data: the size of the file downloaded from the FTP server was 10MB in the first case, 20 MB in the second case and 50 MB in the third one. The measurement tests were repeated many times for each type and each data size. The results were constantly the same: the differences between the measured throughput values were less than 2% for the same type of measurements.

The FTP server was implemented on PC2 by using the built-in FTP server of the operating system. The built-in FTP client was used on PC1 to download the files. The System Monitor application was used to create the measurement reports.

The measurements' results are shown in Fig. 4. – 12. These figures show the interface throughput values for the different test cases.

Concerning the user's point of view, the application layer's throughput is much more interesting than the interface throughput. The values of the throughput measured in the application layer (i.e. dividing the transmitted data size with the transmission time) can be seen in Table II. Of course, the interface throughput values are a little bit bigger than the application layer's one, because of the additional header information appearing on the interfaces.

Easy to see, that the transmission time increases linearly with the data size on the physical interfaces and on the tunnel interface too, i.e. the throughput does not depend on the data size. (The difference is less than 3% in all cases.)

It can be seen from the measurement results that the throughput capacity of the different paths are summed efficiently on the tunnel interface by using the MPT library. This statement holds for the interface throughput and for the application layer's throughput too: the efficiency is better than 90% in all cases.

Also, the figures show that the variance of the interface throughput is a little bit bigger in the non-symmetrical cases. The investigation of this feature is out of the scope of this paper.

TABLE I
IP Addressing Table

Device	Interface	IP address/prefix	Default gateway
PC1	eth1	172.16.0.2/24	172.16.0.1
	eth2	172.16.1.2/24	172.16.1.1
	tun1	1.2.3.2/24	-
PC2	eth1	10.150.0.2/24	10.150.0.1
	eth2	10.150.1.2/24	10.150.1.1
	tun1	1.2.3.3/24	-
Router1	eth0	172.16.0.1/24	-
	eth0	172.16.1.1/24	-
	serial0	192.168.2.1/30	-
	serial1	192.168.3.1/30	-
Router2	eth0	10.150.0.1/24	-
	eth0	10.150.1.1/24	-
	serial0	192.168.2.2/30	-
	serial1	192.168.3.2/30	-

TABLE II
Measurement results

Case	Int.	Time (seconds)	Application layer's throughput (KiB/s)
Case 1: 10MB S0, S1:1.300.000	tun1	37	283.4
	eth1, eth2	72	145.6
Case 2: 20MB S0, S1:1.300.000	tun1	74	283.4
	eth1, eth2	144	145.6
Case 3: 50MB S0, S1:1.300.000	tun1	185	283.4
	eth1, eth2	362	144.8
Case 4: 10MB S0, S1:2.000.000	tun1	23	455.9
	eth1, eth2	46	228.0

Case 5: 20MB S0, S1:2.000.000	tun1	47	446.2
	eth1, eth2	92	228.0
Case 6: 50MB S0, S1:2.000.000	tun1	118	444.3
	eth1, eth2	231	227.0
Case 7: 10MB S0:2.000.000 S1:1.000.000	tun1	31	338.3
	eth1	46	228.0
	eth2	89	117.8
Case 8: 20MB S0:2.000.000 S1:1.000.000	tun1	62	338.3
	eth1	92	228.0
	eth2	179	117.2
Case 9: 50MB S0:2.000.000 S1:1.000.000	tun1	157	333.9
	eth1	231	227.0
	eth2	449	116.8

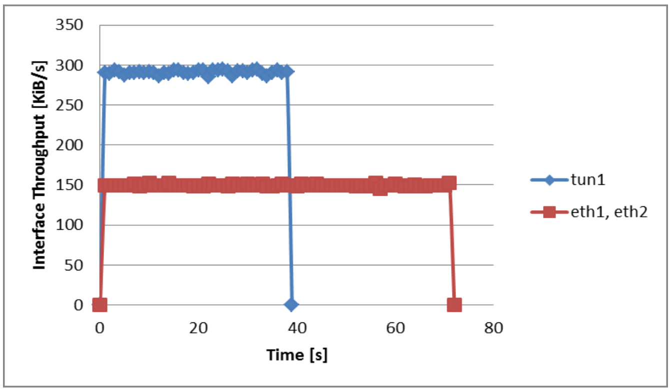

Fig. 4. . Test case 1: Data size: 10 MB, Clock rates: 1.300.000 / 1.300.000

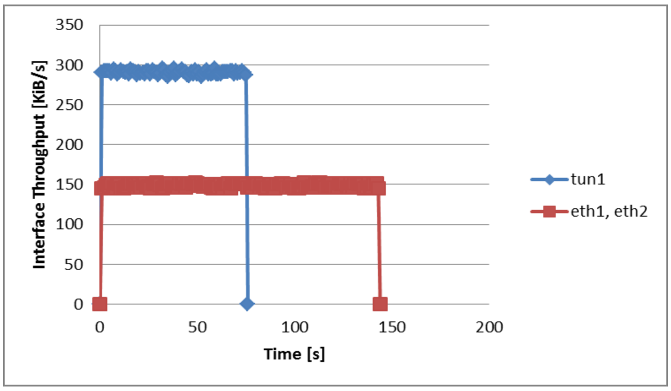

Fig. 5. Test case 2: Data size: 20 MB, Clock rates: 1.300.000 / 1.300.000

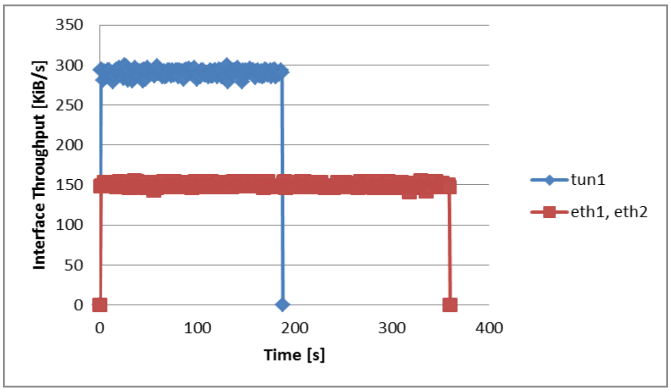

Fig. 6. Test case 3: Data size: 50 MB, Clock rates: 1.300.000 / 1.300.000

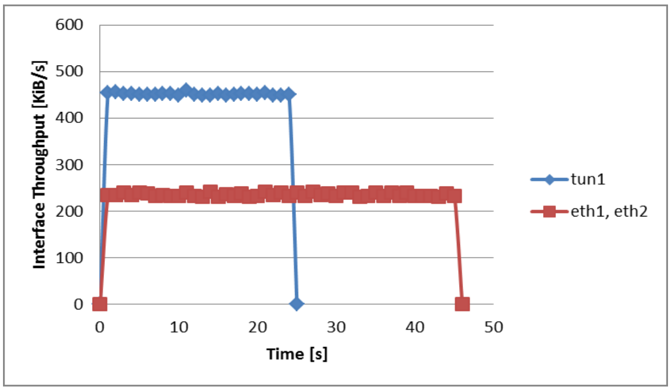

Fig. 7. Test case 4: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000

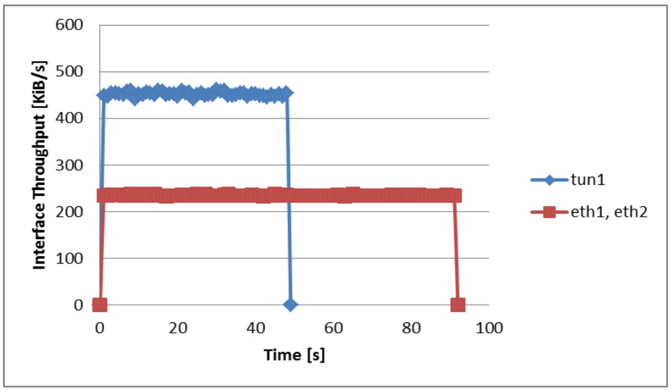

Fig. 8. Test case 5: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000

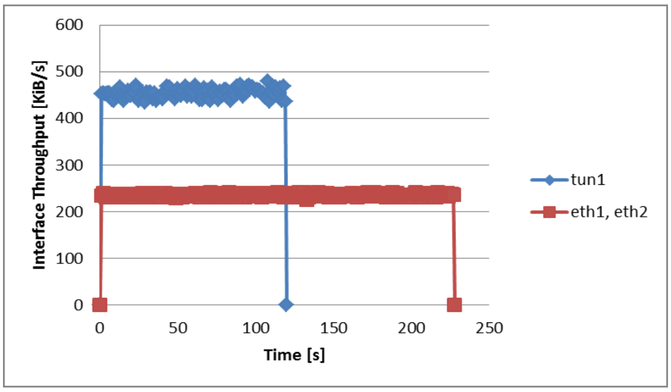

Fig. 9. Test case 6: Data size: 50 MB, Clock rates: 2.000.000 / 2.000.000

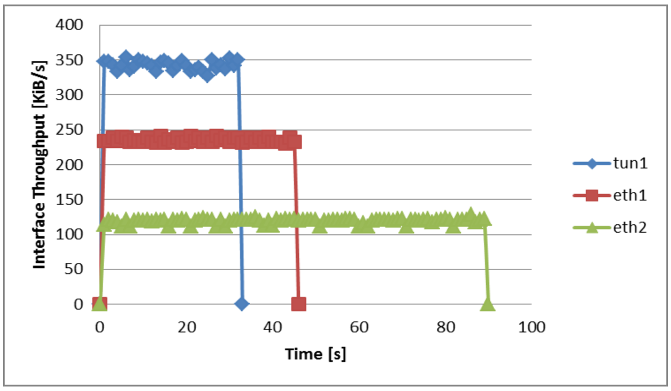

Fig. 10. Test case 7: Data size: 10 MB, Clock rates: 2.000.000 / 1.000.000

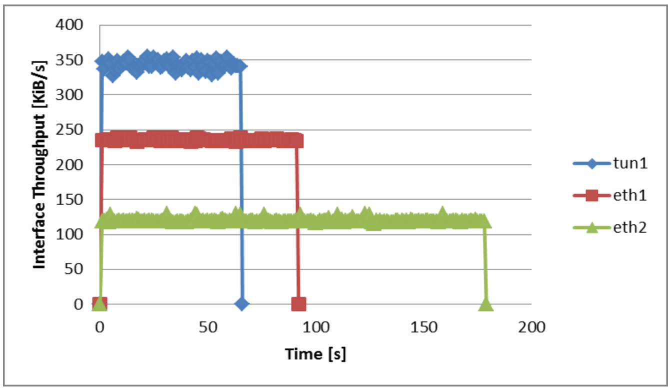

Fig. 11. Test case 8: Data size: 20 MB, Clock rates: 2.000.000 / 1.000.000

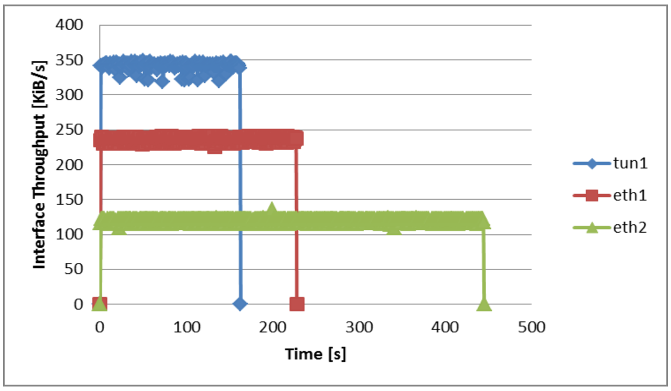

Fig. 12. Test case 9: Data size: 50 MB, Clock rates: 2.000.000 / 1.000.000

V. SUMMARY

In this paper we introduced a software library, named MPT, which can be used to perform multipath communication between the end hosts. The applications over the MPT tool may use any kind of transport layer protocol: the usage of both TCP and UDP is allowed. The focused purpose of the paper was to investigate the throughput performance of the MPT tool. We established a measurement environment which provided two independent connection paths for the communicating hosts. The throughput performance was analyzed using symmetrical and non-symmetrical bandwidth rates, and different transmitted data sizes. All the test measurement showed that the MPT multipath environment aggregates the physical paths throughput capacity very efficiently. Further research work of the MPT tool can focus on the NAT compatibility of MPT, or on questions related to the QoS area.

REFERENCES

[1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresse*", IETF RFC-6824*, 2013. Available: http://tools.ietf.org/html/rfc6824.

[2] B. Almási, "UDPTUN – Direct TCP connection between 'NAT behind' hosts", *8th International Conference on Applied Informatics*, Eger, Hungary, pp. 325-332, 2010.

[3] B. Almási, "A simple solution for wireless network layer roaming problems," *Carpathian Journal of Electronic and Computer Engineering,* vol. 5, no. 1, pp. 5-8, 2012.

[4] W. Odom, "CCNA ICND2 Official Exam Certification Guide*",* (Book style)*,* 3rd ed., *Cisco Press*, Indianapolis, USA, pp. 180–183, 2011.

[5] M. Scharf, T.R. Banniza, P. Schefczik, A. Singh, and A. Timm-Giel, "Evaluation and Prototyping of Multipath Protocol Mechanisms", *10th Würzburg Workshop on IP: Joint ITG, ITC and Euro-NF Workshop „Visions of Future Generation Networks"*, Würzburg, Germany, Aug., 2010. Available: http://www.euroview2010.com/data/poster/19_Scharf_ Multipath.pdf.

[6] A. Singh, C. Görg, A. Timm-Giel, M. Scharf, and T.R. Banniza, "Performance Evaluation of Multipath TCP Linux Implementations", *11th Würzburg Workshop on IP: Joint ITG, ITC and Euro-NF Workshop „Visions of Future Generation Networks"*, Würzburg, Germany, Aug., 2011. Available: http://www.euroview2011.com/fileadmin/content/euroview2011/abstracts/abstract_singh.pdf.