

Investigating the Performance of the MPT Multipath Communication Library in IPv4 and IPv6

Béla Almási, Szabolcs Szilágyi

Abstract—The currently used mobile devices (laptops, tablets, mobile phones) contain many built-in network cards for communication (e.g. Wi-Fi, 3G, Bluetooth, etc.). A natural request could be combining the resources of the different network connection possibilities in order to increase the throughput of the communication. Unfortunately the standard IP communication technology does not support it: the communication is restricted to one IP address (i.e. to one interface). The Multipath TCP (MPTCP) specification (appeared in January 2013) offers a Transport layer solution of using more than one interface in a TCP communication session. In this paper we investigate a Network layer solution. The MPT multipath communication library opens the multipath communication possibility in the Network layer. Using the MPT library, applications built on the UDP protocol are also able to perform multipath communication. The MPT library was developed by using a full dual-stack technology, which means the MPT based multipath environment can be used both in IPv4 and IPv6. Protocol version change is also possible: an IPv6 based application is able to run in an IPv4 multipath environment, and an IPv4 application can be used in an IPv6 multipath environment. In this paper we give a short overview on the MPT communication library's working mechanism and detailed numerical examples will be shown to present how the MPT library aggregates the paths' throughput in IPv4 and IPv6 environments. The main contribution of the paper is to demonstrate the effective throughput aggregation property of the MPT library in IPv6 and in mixed (i.e. protocol version change) environments.

Keywords—IPv6, Multipath communication, throughput performance, tunnel.

I. INTRODUCTION

The traditional IP communication infrastructure is restricted to use a single interface on the communication endpoints. The IP address of the interface is used not only to identify the node, but it is also used to identify the communication session (socket ID). Distributing a communication session between different paths is an interesting question, and it is a focused research area today (see e.g. [6] – [8]). If the communication session is terminated on a moving node (e.g. on a computer located in a moving car) then it may request for establishing an efficient L3 roaming communication (see e.g. [3]). Opening the possibility of changing the IP address

of the end node (with the assumption that the communication session must continue), could open a quite new conceptual solution idea for well-known handover problems (see e.g. [2]): The moving computer could easily change its IP address without losing the communication session's state by changing a path only.

In this paper we investigate the performance of the MPT software library which was developed at the Faculty of Informatics, University of Debrecen in Hungary. The MPT software library offers a Network layer multipath communication possibility for the computers (typically having multiple network interfaces). The MPT software creates a logical interface (tunnel interface) on the host. The communication of the logical interface is mapped to the physical interfaces dynamically by the MPT software. The applications use the IP address of the tunnel interface for the communication session's identification, so there is no need for modification in the applications' communication software. When an application sends an IP packet, it will use the address of the logical interface (i.e. the address of the tunnel interface) as the sender address. This packet will be encapsulated into a new UDP segment and IP packet by the MPT software. The new IP packet can be dynamically assigned to a physical interface of the host, i.e. the new packet will use the IP address of a physical interface as the sender address. The MPT library covers the task of mapping between the tunnel interface and the physical interfaces. If we change the IP address of the physical interface (or even we change the physical interface itself) the application will not sense this change, it will continue the work over the tunnel interface. The MPT software will recognize the changing and will reorganize the mapping from the tunnel interface to the physical interface, according to the new situation. The layered structure of the MPT networking system can be seen in Figure 1.

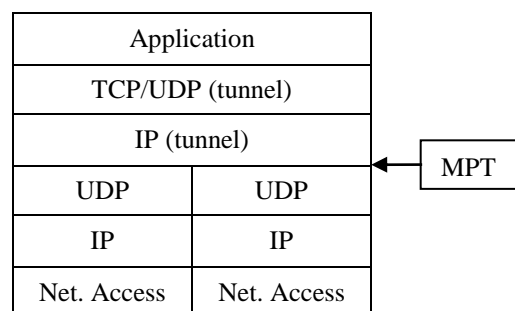


Fig. 1. The layered structure of the MPT multipath environment

Manuscript received November 15, 2015, revised February 25, 2016.

B. Almási, is with the Faculty of Informatics, University of Debrecen, Debrecen, Hungary (e-mail: almasi.bela@inf.unideb.hu).

Sz. Szilágyi is with the Faculty of Informatics, University of Debrecen, Debrecen, Hungary (e-mail: szilagyi.szabolcs@inf.unideb.hu).

doi: 10.11601/ijates.v5i1.148

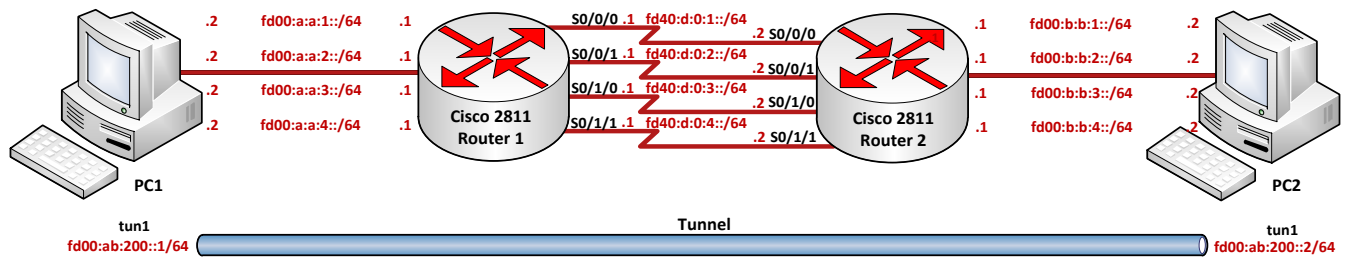


Fig. 2. The four paths measurement laboratory environment

It is easy to see that the tunnel interface and the physical interface may use different IP versions: the application may use the IPv4 address of the tunnel interface while the physical network environment may use IPv6 addresses on the physical interfaces. The protocol change is automatically performed by the MPT software.

It must be mentioned, that the RFC 6824 document (see [7], [8]) also introduces a multipath communication technology, but it works in the Transport layer, and the application is restricted to use the TCP protocol. The MPT library works in the Network layer and the application is able to use the UDP protocol too (e.g. for voice or media content transmission).

In order to precisely measure the throughput aggregation performance of the MPT library we established a measurement test-network environment. The measurement laboratory contains 4 paths between the endpoints, and is able to perform communication in IPv4 and IPv6 (offering the possibility of study mixed protocol versions.) Although the aggregation efficiency of the paths' throughputs of the different IP versions are not the same, the results show that the MPT environment successfully aggregates the paths' throughput in IPv4, IPv6 and mixed protocol versions too.

The rest of the paper is organized as follows: Section 2 describes the measurement laboratory network environment. Section 3 shows the measurement results performed in IPv4. Section 4 discusses the measurement results using IPv6 and section 5 will show the measurement results of a mixed protocol environment: in this section the application uses IPv4 (over the tunnel interface) and the real network environment uses IPv6 (on the physical interfaces).

This paper is some kind of continuation of paper [6]: the authors studied the handover capabilities of the MPT software library on the 37th TSP Conference in 2014. In this paper we make a step forward, and investigate the performance of the MPT library in a multiprotocol (IPv4/IPv6) environment.

II. THE MEASUREMENT NETWORK ENVIRONMENT

The measurement test-laboratory contained two PCs (having 8 GB of main memory and Intel Core i5 processor with 2.5GHz, and 6MB cache; using the operating system of Ubuntu 12.04). The PCs were equipped with four interfaces (eth0, eth1, eth2 and eth3). The four interfaces of the PCs were connected to each other by four different paths. The paths were realized by using two Cisco 2811 type routers. The routers were connected to each other by four serial links, thus establishing 4 different paths.

TABLE I
IPv4 AND IPv6 ADDRESSING

Dev	Inter- face	IPv4/IPv6 address/prefix	Default gateway
PC1	eth0	172.16.1.2/24 fd00:a:a:1::2/64	172.16.1.1/24 fd00:a:a:1::1/64
	eth1	172.16.2.2/24 fd00:a:a:2::2/64	172.16.2.1/24 fd00:a:a:2::1/64
	eth2	172.16.3.2/24 fd00:a:a:3::2/64	172.16.3.1/24 fd00:a:a:3::1/64
	eth3	172.16.4.2/24 fd00:a:a:4::2/64	172.16.4.1/24 fd00:a:a:4::1/64
	tun1	1.2.3.2/24 fd00:ab:200::1/64	-
PC2	eth0	10.150.1.2/24 fd00:b:b:1::2/64	10.150.1.1/24 fd00:b:b:1::1/64
	eth1	10.150.2.2/24 fd00:b:b:2::2/64	10.150.2.1/24 fd00:b:b:2::1/64
	eth2	10.150.3.2/24 fd00:b:b:3::2/64	10.150.3.1/24 fd00:b:b:3::1/64
	eth3	10.150.4.2/24 fd00:b:b:4::2/64	10.150.4.1/24 fd00:b:b:4::1/64
	tun1	1.2.3.3/24 fd00:ab:200::1/64	-
R1	FE 0/0.1	172.16.1.1/24 fd00:a:a:1::1/64	-
	FE 0/0.2	172.16.2.1/24 fd00:a:a:2::1/64	-
	FE 0/0.3	172.16.3.1/24 fd00:a:a:3::1/64	-
	FE 0/0.4	172.16.4.1/24 fd00:a:a:4::1/64	-
	serial 0/0/0	192.168.1.1/30 fd40:d:0:1::1/64	-
	serial 0/0/1	192.168.2.1/30 fd40:d:0:2::1/64	-
	serial 0/1/0	192.168.3.1/30 fd40:d:0:3::1/64	-
	serial 0/1/1	192.168.4.1/30 fd40:d:0:4::1/64	-
R2	FE 0/0.1	10.150.1.1/24 fd00:b:b:1::1/64	-
	FE 0/0.2	10.150.2.1/24 fd00:b:b:2::1/64	-
	FE 0/0.3	10.150.3.1/24 fd00:b:b:3::1/64	-
	FE 0/0.4	10.150.4.1/24 fd00:b:b:4::1/64	-
	serial 0/0/0	192.168.1.2/30 fd40:d:0:1::2/64	-
	serial 0/0/1	192.168.2.2/30 fd40:d:0:2::2/64	-
	serial 0/1/0	192.168.3.2/30 fd40:d:0:3::2/64	-
	serial 0/1/1	192.168.4.2/30 fd40:d:0:4::2/64	-

The physical link of the Ethernet connection was common in the paths, but the high speed of this common link (100 Mbps) made it sure that the bottleneck point of the paths was not in the common part of the network. (The maximum clock rate of the serial links was set to 2.000.000 cycles per seconds. The clock rate setting of the DCE serial interfaces was used to tune the bandwidth of the paths independently to each other. The PCs were connected to the routers using 100 Mbps Ethernet links, so the bottleneck point of the communication was realized on the serial links between the routers.). All the routing settings were implemented by static entries both in the routers and in the PCs. The IP addressing scheme of the measurement network can be seen in Table I.

III. MEASUREMENT RESULTS – IPV4 ONLY NETWORK

In order to test the throughput of the network system two types of measurements have been carried out. The first type of the measurement used symmetrical paths with clock rate values of 1.000.000 and 2.000.000 cycles per second respectively (see Table II, cases 1-4). The second type of measurement used non-symmetrical paths: the four links between the serial interfaces used all the possible combination of the clock rate values of 1.000.000 and 2.000.000 (see Table II, cases 5-10). All measurement types contained two cases, which differed only in the size of the transmitted file: in the first case the transmitted file's size was 10 MB, and it was 20 MB in the second one. For all cases (assumed also in the following sections) the efficiency in the tunnel has been evaluated as:

$$\text{Tunnel efficiency [\%]} = \frac{\text{throughput value measured on tunnel interface}}{\text{sum of the throughput values measured on physical interfaces}} * 100 \quad (1)$$

The measurement tests were always repeated ten times for each type of measurement and for each data size. The results were constantly the same: the differences between the measured throughput values were less than 3%. The host PC1 was used as the FTP server starting the built in FTP server daemon of the operating system (vsftpd). The built-in FTP client was used on PC2 to download the files. The System Monitor application was used to create the measurement reports. The detailed measurements' results are shown in Table II. and Figures 3. – 10. The figures show the interface throughput values for the test cases using 10 and 20 MB file sizes. The variance of the interface throughput increases a little bit at the aggregation. The results show the same performance for both file sizes: it is easy to see that the transmission time increases linearly with the data size on the physical interfaces (on the tunnel interface too), i.e. the throughput aggregation does not depend on the data size. (The difference is less than 3% in all 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, measured in seconds) can be seen in Table II (the transmitted data size is measured in 1024 bytes unit). Of course, the interface throughput values (in Figures 3. – 10.) are a little bit higher than the Application layer's one, because of the additional header information appearing on

the interfaces.

TABLE II
FTP FILE TRANSMISSION RESULTS – IPV4 ONLY NETWORK

Case	Interface	Time (sec)	Throughput (KB/s)	Efficiency (%)
Case 1. 10 MB 1 M / 1 M 1 M / 1 M	tun1	22	465,5	99,9
	eth0-3	88	116,4	
Case 2. 20 MB 1 M / 1 M 1 M / 1 M	tun1	45	455,1	98,3
	eth0-3	177	115,7	
Case 3. 10 MB 2 M / 2 M 2 M / 2 M	tun1	11	930,9	99,9
	eth0-3	44	232,8	
Case 4. 20 MB 2 M / 2 M 2 M / 2 M	tun1	22	930,9	99,9
	eth0-3	88	232,8	
Case 5. 10 MB 2 M / 1 M 1 M / 1 M	tun1	18	568,9	97,8
	eth0	44	232,8	
	eth1-3	88	116,4	
Case 6. 20 MB 2 M / 1 M 1 M / 1 M	tun1	36	568,9	98,1
	eth0	88	232,8	
	eth1-3	177	115,7	
Case 7. 10 MB 2 M / 2 M 1 M / 1 M	tun1	15	682,7	97,8
	eth0-1	44	232,8	
	eth2-3	88	116,4	
Case 8. 20 MB 2 M / 2 M 1 M / 1 M	tun1	30	682,7	98,0
	eth0-1	88	232,8	
	eth2-3	177	115,7	
Case 9. 10 MB 2 M / 2 M 2 M / 1 M	tun1	13	787,7	96,7
	eth0-2	44	232,8	
	eth3	88	116,4	
Case 10. 20 MB 2 M / 2 M 2 M / 1 M	tun1	26	787,7	96,8
	eth0-2	88	232,8	
	eth3	177	115,7	

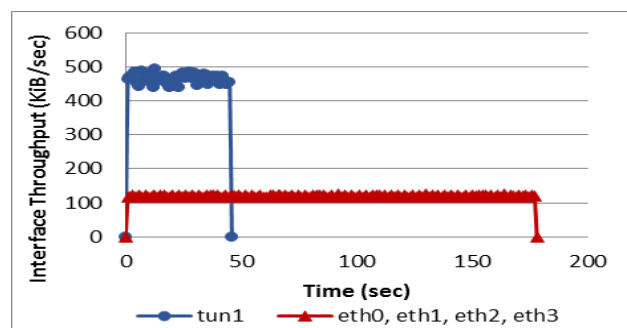


Fig. 3. IPv4 Test case 2: Data size: 20 MB, Clock rates: 1.000.000 / 1.000.000 / 1.000.000

It can be seen from the measurement results that the throughput capacity of the four paths are summed efficiently on the tunnel interface by using the MPT library. This statement holds both for the interface throughput and for the Application layer's throughput: the efficiency is better than 96% in all cases (see Table II).

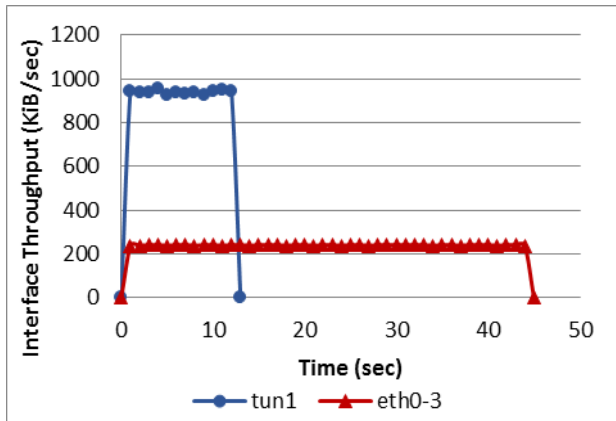


Fig. 4. IPv4 Test case 3: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000

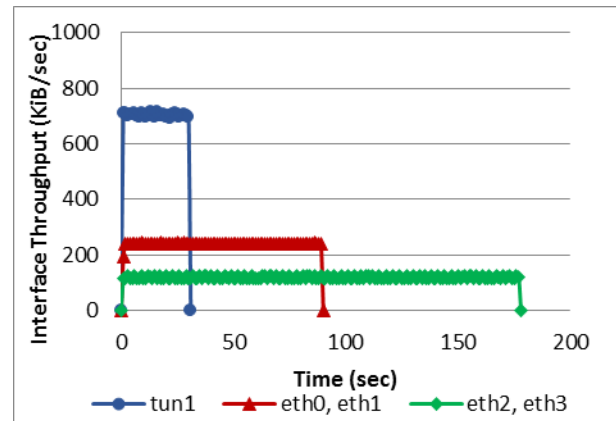


Fig. 8. IPv4 Test case 8: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 1.000.000 / 1.000.000

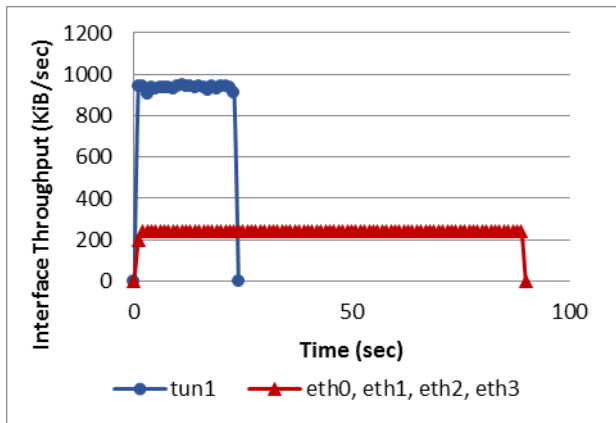


Fig. 5. IPv4 Test case 4: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000

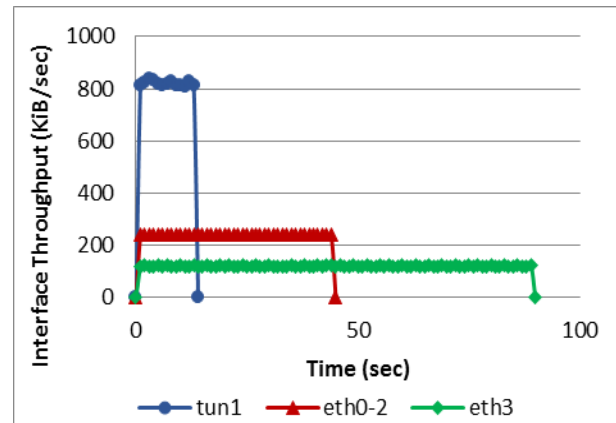


Fig. 9. IPv4 Test case 9: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

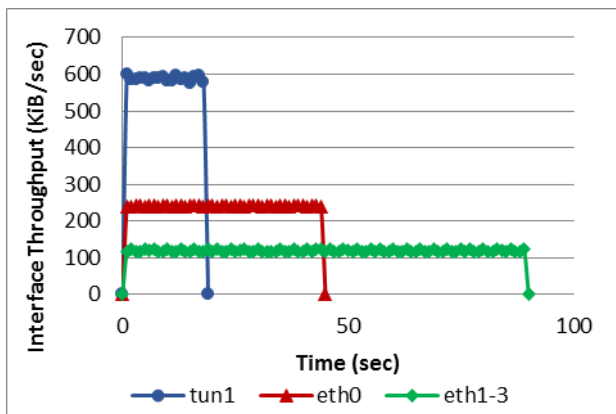


Fig. 6. IPv4 Test case 5: Data size: 10 MB, Clock rates: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000

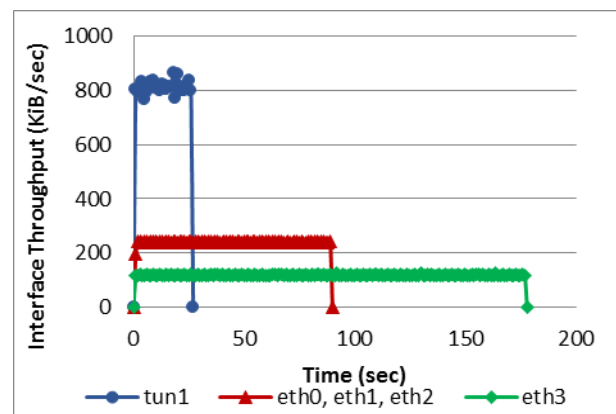


Fig. 10. IPv4 Test case 10: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

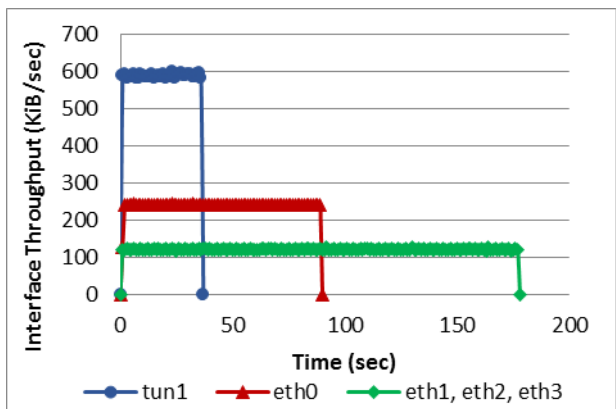


Fig. 7. IPv4 Test case 6: Data size: 20 MB, Clock rates: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000

Also, the figures show that the variance of the interface throughput is a little bit higher in the non-symmetrical cases. Further IPv4 measurement results can be found in [1].

IV. MEASUREMENT RESULTS – IPV6 ONLY NETWORK

Using IPv6 both over the tunnel and under the tunnel changes the measurement situation: the additional IP encapsulation will use a longer IPv6 header, so the efficiency decrease can be expected in this case.

The Physical and Data-link layer properties of the measurement environment remained the same as it was in the previous chapter (i.e. clock rates, file sizes). Only the Network layer protocol was changed from IPv4 to IPv6.

The measurements' results of the interfaces' throughput

can be seen in Figure 11. – 18. The numerical values of the application's layer throughput can be seen in Table III. (i.e. the user's point of view).

TABLE III
FTP FILE TRANSMISSION RESULTS – IPV6 ONLY NETWORK

Case	Interface	Time (sec)	Through-put (KB/s)	Efficiency (%)
Case 1. 10 MB 1M / 1M 1M / 1M	tun1	23	445,2	97,8
	eth0-3	90	113,8	
Case 2. 20 MB 1M / 1M 1M / 1M	tun1	47	435,7	95,7
	eth0-3	180	113,8	
Case 3. 10 MB 2M / 2M 2M / 2M	tun1	12	853,3	93,7
	eth0-3	45	227,6	
Case 4. 20 MB 2M / 2M 2M / 2M	tun1	23	890,4	97,8
	eth0-3	90	227,6	
Case 5. 10 MB 2M / 1M 1M / 1M	tun1	19	538,9	94,7
	eth0	45	227,6	
	eth1-3	90	113,8	
Case 6. 20 MB 2M / 1M 1M / 1M	tun1	38	538,9	94,7
	eth0	90	227,6	
	eth1-3	180	113,8	
Case 7. 10 MB 2M / 2M 1M / 1M	tun1	16	640,0	93,7
	eth0-1	45	227,6	
	eth2-3	90	113,8	
Case 8. 20 MB 2M / 2M 1M / 1M	tun1	32	640,0	93,7
	eth0-1	90	227,6	
	eth2-3	180	113,8	
Case 9. 10 MB 2M / 2M 2M / 1M	tun1	14	731,4	91,8
	eth0-2	45	227,6	
	eth3	90	113,8	
Case 10. 20 MB 2M / 2M 2M / 1M	tun1	27	758,5	95,2
	eth0-2	90	227,6	
	eth3	180	113,8	

It is easy to see that the efficiency of the IPv6 throughput aggregation is a little bit lower than it was in the case of IPv4. All the efficiency values are smaller, the difference is maximum 5 percent in each case. As it can be seen in Table III. all the efficiency measurement numbers are greater than 90 percent in the IPv6 cases, so the aggregation performance is good also in the case of IPv6 only network environment.

Similarly to the IPv4 results the interface throughput is a little bit greater than the application's layer throughput in the case of IPv6 protocol too.

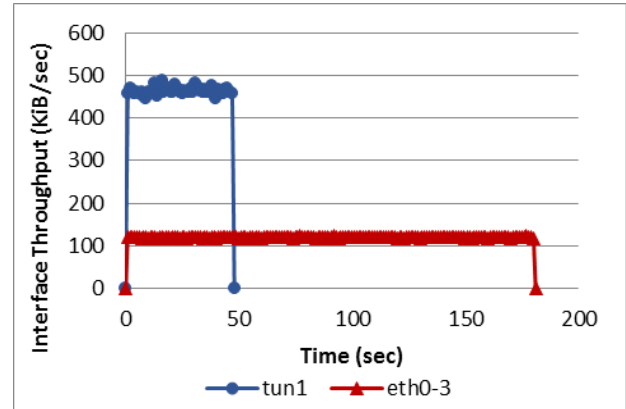


Fig. 11. IPv6 Test case 2: Data size: 20 MB, Clock rates: 1.000.000 / 1.000.000 / 1.000.000 / 1.000.000

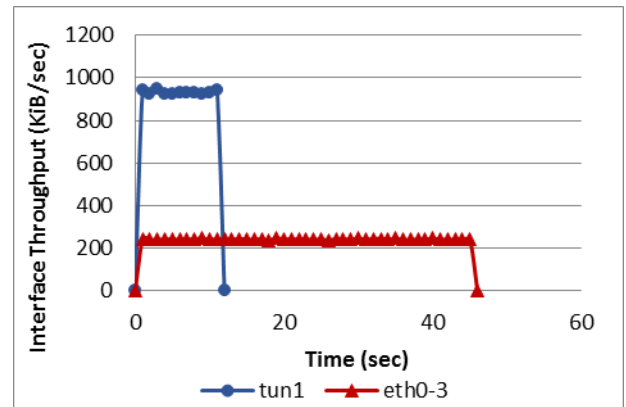


Fig. 12. IPv6 Test case 3: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000

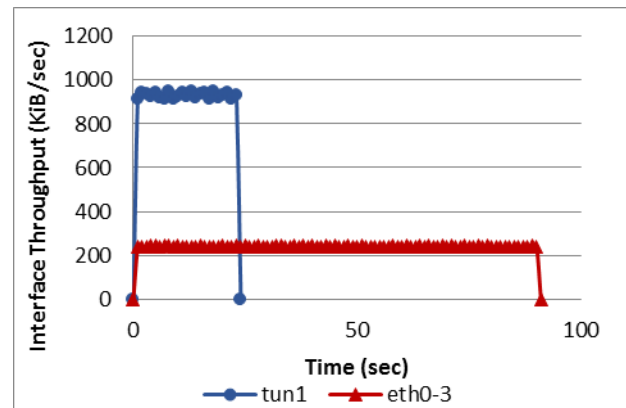


Fig. 13. IPv6 Test case 4: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000

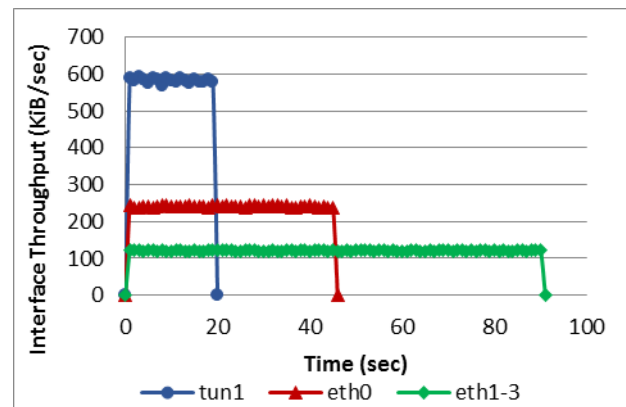


Fig. 14. IPv6 Test case 5: Data size: 10 MB, Clock rates: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000

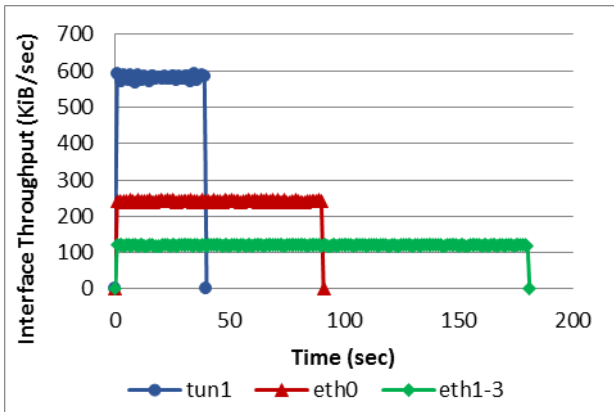


Fig. 15. IPv6 Test case 6: Data size: 20 MB, Clock rates: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000

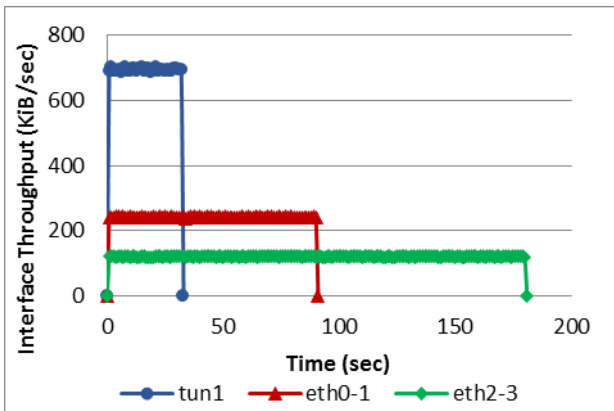


Fig. 16. IPv6 Test case 8: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 1.000.000 / 1.000.000

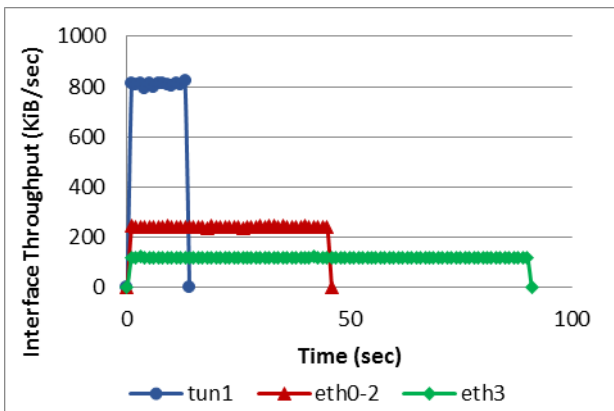


Fig. 17. IPv6 Test case 9: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

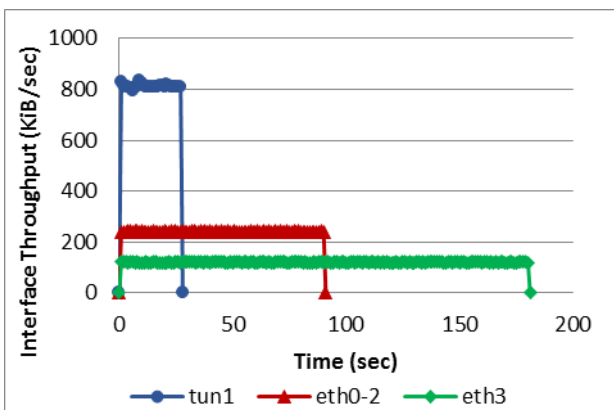


Fig. 18. IPv6 Test case 10: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

V. MEASUREMENT RESULTS – MIXED NETWORK (IPv4 OVER IPv6)

The IPv4 address space has been exhausted in 2012 (i.e. all the IPv4 network id is assigned to ISPs, see [4],[9]). This fact may accelerate the IPv6 usage on the backbone and in the end user area too. In the future the need of special applications, which are not upgraded to the IPv6 protocol, may occur in special networking environments, where only IPv6 network is available. One solution for these special situations can be the usage of MPT: The MPT environment offers the possibility to use the IPv4 protocol on the logical tunnel interfaces (i.e. the applications will use IPv4) while the physical interfaces are connected to an IPv6 only network. This chapter evaluates the performance of the MPT library in a mixed protocol environment. The Physical and Data link layer parameters remain the same as it was in the previous chapters.

TABLE IV
MEASUREMENT RESULTS – MIXED NETWORK (IPv4 OVER IPv6)

Case	Interface	Time (sec)	Throughput (KB/s)	Efficiency (%)
Case 1: 10MB 1M / 1M 1M / 1M	tun1	23	445,2	97,8
	eth0-3	90	113,8	
Case 2: 20MB 1M / 1M 1M / 1M	tun1	46	445,2	97,8
	eth0-3	180	113,8	
Case 3: 10MB 2M / 2M 2M / 2M	tun1	12	853,3	93,7
	eth0-3	45	227,6	
Case 4: 20MB 2M / 2M 2M / 2M	tun1	23	890,4	97,8
	eth0-3	90	227,6	
Case 5: 10MB 2M / 1M 1M / 1M	tun1	19	538,9	94,7
	eth0	45	227,6	
	eth1-3	90	113,8	
Case 6: 20MB 2M / 1M 1M / 1M	tun1	38	538,9	94,7
	eth0	90	227,6	
	eth1-3	180	113,8	
Case 7: 10MB 2M / 2M 1M / 1M	tun1	16	640,0	93,7
	eth0-1	45	227,6	
	eth2-3	90	113,8	
Case 8: 20MB 2M / 2M 1M / 1M	tun1	31	660,6	96,7
	eth0-1	90	227,6	
	eth2-3	180	113,8	
Case 9: 10MB 2M / 2M 2M / 1M	tun1	13	787,7	98,8
	eth0-2	45	227,6	
	eth3	90	113,8	
Case 10: 20MB 2M / 2M 2M / 1M	tun1	27	758,5	95,2
	eth0-2	90	227,6	
	eth3	180	113,8	

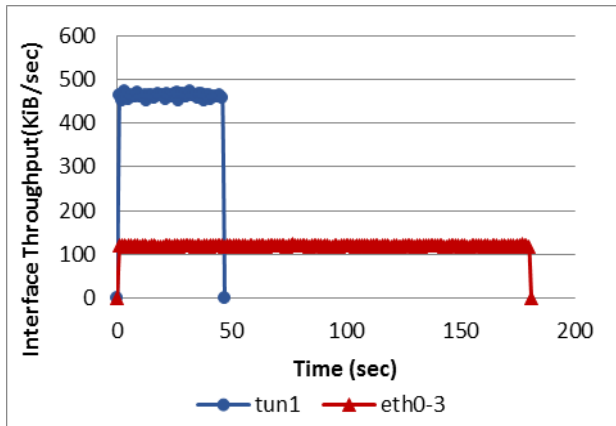


Fig. 19. Mixed Test case 2: Data size: 20 MB, Clock rates: 1.000.000 / 1.000.000 / 1.000.000

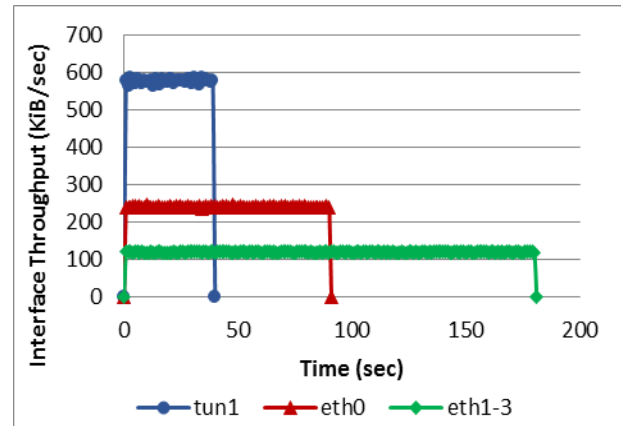


Fig. 23. Mixed Test case 6: Data size: 20 MB, Clock rates: 2.000.000 / 1.000.000 / 1.000.000

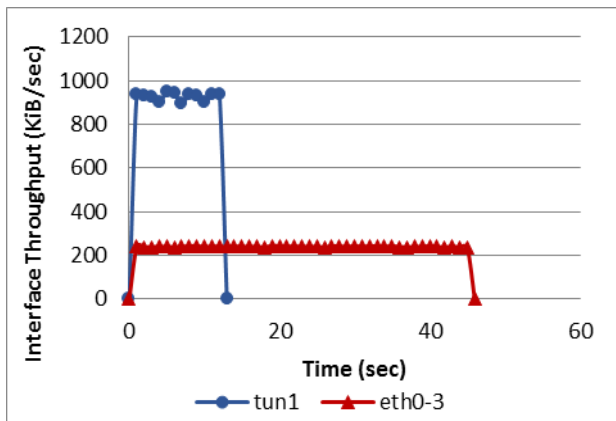


Fig. 20. Mixed Test case 3: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000

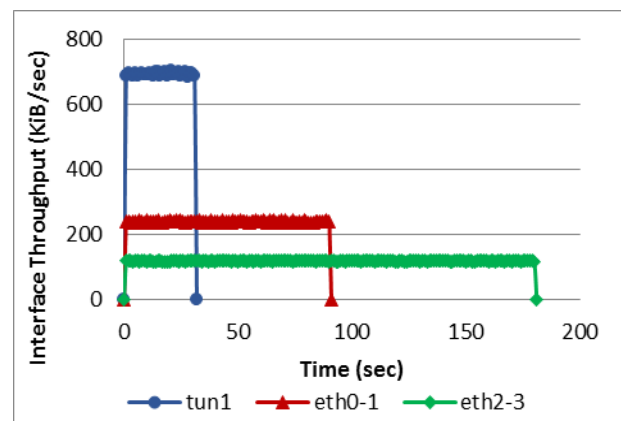


Fig. 24. Mixed Test case 8: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 1.000.000

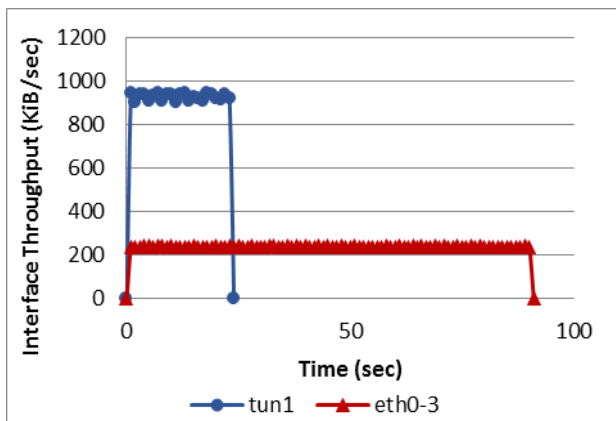


Fig. 21. Mixed Test case 4: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 2.000.000

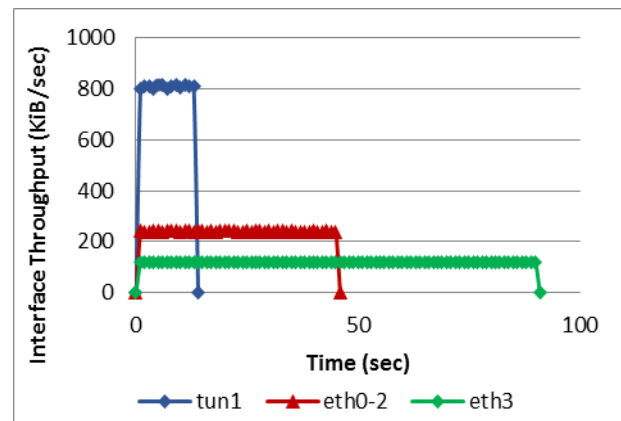


Fig. 25. Mixed Test case 9: Data size: 10 MB, Clock rates: 2.000.000 / 2.000.000 / 1.000.000

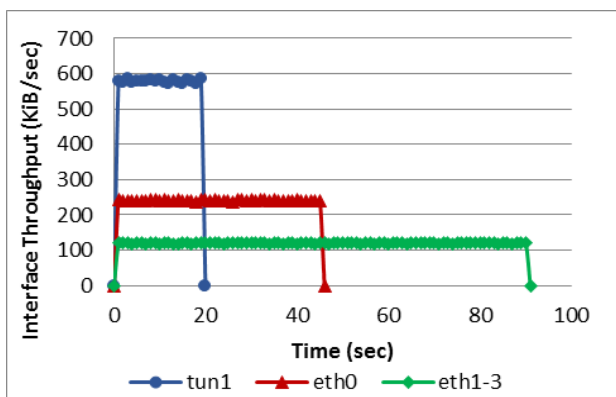


Fig. 22. Mixed Test case 5: Data size: 10 MB, Clock rates: 2.000.000 / 1.000.000 / 1.000.000

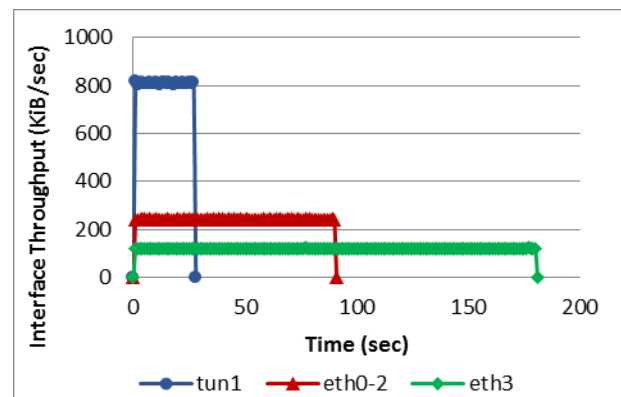


Fig. 26. Mixed Test case 10: Data size: 20 MB, Clock rates: 2.000.000 / 2.000.000 / 1.000.000

The measurements' results are shown in Figure 19. – 26. These figures show the interface throughput values for the test cases using 10 and 20 MB file sizes.

Table IV shows the Application layer throughput in the mixed protocol environment: The results are quite similar to the results of the IPv4 and IPv6 only networks measurement: in some cases the results are similar to the IPv4 environment (see cases 8, 9), other test produced numerical results like it was in the IPv6 only measurement (see cases 3 – 7). It means, that the mixed network environment produced numerical results between the IPv4 and IPv6 only measurements (as it could be expected).

VI. CONCLUSION

In this paper we investigated the throughput aggregation property of the multipath a software library MPT. 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 using IPv4, IPv6 and mixed network environments. We established a measurement laboratory environment which provided four independent connection paths for the communicating hosts. The throughput performance was analyzed using symmetrical and non-symmetrical bandwidth rates, and different transmitted data sizes. Although the results showed some difference in the different protocol versions, the test measurements showed that the MPT multipath environment successfully aggregates the physical paths throughput capacity.

REFERENCES

- [1] B. Almási, Sz. Szilágyi, "Multipath FTP and Stream Transmission Analysis using the MPT Software Environment", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 4267-4272, Nov. 2013.
- [2] N. C. Wang, Y. K. Chiang, S. M. Wei, "RSVP Extensions for Seamless Handoff in Heterogeneous WLAN/WiMAX Networks", *Journal of Applied Research and Technology*, vol. 11, no. 4, pp. 540-548, 2013.

- [3] O. H. Sabrie, H. S. Hasan, R. Salleh, "Fast Handoff for 802.11 Wireless Network", *Communications and Network*, vol. 3, pp. 250-256, 2011.
- [4] G. Goth, "The End of IPv4 is Nearly Here", *IEEE J. Internet Computing*, vol. 16, no. 2, pp. 7-11, 2012.
- [5] B. Almási, "Multipath Communication – a new basis for the Future Internet Cognitive Infocommunication", *Proceedings of the CogInfoCom 2013 Conference*, Budapest, Hungary, 2013, pp. 201-204.
- [6] B. Almási, "A Solution for Changing the Communication Interfaces between WiFi and 3G without Packet Loss", *Proceedings of the 37th International Conference on Telecommunications and Signal Processing*, Berlin, Germany, 2014, pp. 73-77.
- [7] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. (2013). TCP Extensions for Multipath Operation with Multiple Addresses, IETF RFC-6824. Available: <http://tools.ietf.org/html/rfc6824>
- [8] M. Handley, O. Bonaventure. (2014, January). TCP Extensions for Multipath Operation with Multiple Addresses draft-ietf-mptcp-rfc6824bis-02. IETF Internet-Draft. Available: <https://datatracker.ietf.org/doc/draft-ietf-mptcp-rfc6824bis/>
- [9] Global Policy for Post Exhaustion IPv4 Allocation Mechanisms by the IANA. (Ratified 6 May 2012). Available: <https://www.icann.org/en/resources/policy/global-addressing/allocation-ipv4-post-exhaustion>



Béla Almási received MSc and PhD in mathematics and computer science from University of Debrecen, Hungary in 1989 and 1996 respectively. He works as an associate professor at the Department of Informatics Systems and Networks, Faculty of Informatics at the University of Debrecen, Hungary. His research interests include the establishment and performance evaluation of stochastic queueing models for communication systems and investigating the multipath communication technologies. He is the head of the Multipath Communication Research Group at the Faculty of Informatics, University of Debrecen.



Szabolcs Szilágyi received his MSc in electrical engineering and computer science at the University of Oradea in 2009, and his PhD in 2015.

He has been working for the Department of Informatics Systems and Networks, Faculty of Informatics, University of Debrecen since 2013. He teaches Computer networks, and Cisco CCNA Routing and Switching. Now, he is a Senior Lecturer. The area of his research includes performance analysis of infocommunication networks.