ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

Multipath FTP and Stream Transmission Analysis using the MPT Software Environment

Béla Almási[1], Szabolcs Szilágyi[2]

Associate professor, Faculty of Informatics, University of Debrecen, Debrecen, Hungary [1]

Assistant, Faculty of Informatics, University of Debrecen, Debrecen, Hungary[2]

Abstract: The currently used standard Internet communication is restricted to use a single path between the endpoints in a communication session. In the January of 2013 the Multipath TCP technology was established by IETF in order to open the possibility of the multiple path usage. Performance measurements proof that MPTCP efficiently aggregates the paths' throughput. The MPTCP technology works in the Transport layer, and the application must use TCP protocol. In this paper we introduce a new multipath solution technology: the MPT software environment works in the Network layer, so it allows the usage of the UDP protocol in the Transport layer (e.g. for voice or video stream transmission purpose). We created a four path laboratory environment to measure the performance of the MPT tool in the case of file transfer and multimedia content (stream) transmission. The measurement results show that the MPT software library efficiently aggregates the available throughput values of the paths in the case of file transfer and media transfer. For media transfer keeping the jitter at a low level value is also important. Our measurements show, that the jitter value is low in the MPT multipath environment if the aggregated paths' throughput is roughly the same, but it will increase when the speed of the paths are different.

Keywords: Multipath communication, dynamic tunnel, throughput aggregation, UDP tunnel

I. Introduction

The traditional IP communication infrastructure is restricted to a single IP address (and single interface) usage on the communication endpoints. The IP address is used not only to identify the node (or the interface of the node), but it is also used to identify the communication session. Distributing a communication session between different paths is an interesting question, and it is a focused research area today (see e.g. [1], [2]). If the communication session is terminated on a moving node (e.g. on a computer located on a moving car) many request arises for establishing an efficient communication (see e.g. [3], [4], [5]). Opening the possibility of changing the IP address of the end node (with the assumption, that the communication session must continue the work), could open a quite new solution area for these situations: The moving computer could easily change its IP address without losing the communication session's state.

In this paper we investigate the performance of the MPT software library. The MPT software library 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. The applications use the IP address of the tunnel interface for the communication session's identification. 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. 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 Fig. 1.

It must be mentioned, that the RFC 6824 document (see [6], [7]) also introduces a multipath communication technology, but it works in the Transport layer, and is restricted to use the TCP protocol. MPT works in the Network layer and allows the usage of the UDP protocol too for the application.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

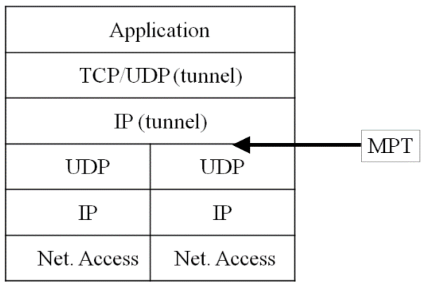

Fig. 1 The Layered structure of the MPT communication environment

In order to investigate the communication efficiency of the MPT environment we created a dedicated measurement laboratory, containing 4 paths between the end nodes. We performed several measurements with different parameters to conclude the results: the MPT software library efficiently aggregates the bandwidth of the four paths even when the paths' bandwidth are different. The bandwidth aggregation feature also holds in the case of media stream transmission, but in this case packet reordering may occur if the bandwidth of the paths are different.

The rest of the paper is organized as follows: Section 2 describes the test laboratory environment used to perform the measurements. The FTP transmission measurement results will be shown in section 3. The media stream transmission measurement results are discussed in section 4, and finally section 5 concludes the paper.

II. THE MEASUREMENT ENVIRONMENT

The measurement 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 OpenSuse Linux 12.0). 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. 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. 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, 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.

III. FTP TRANSFER MEASUREMENT RESULTS

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 I, cases 1-4). The second type of measurement used non-symmetrical paths: the four links between the serial interfaces used the all possible combination between the clock rate values of 1.000.000 and 2.000.000 (see Table I, 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 10MB, and it was 20 MB in the second one. The measurement tests were repeated many 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 2%.

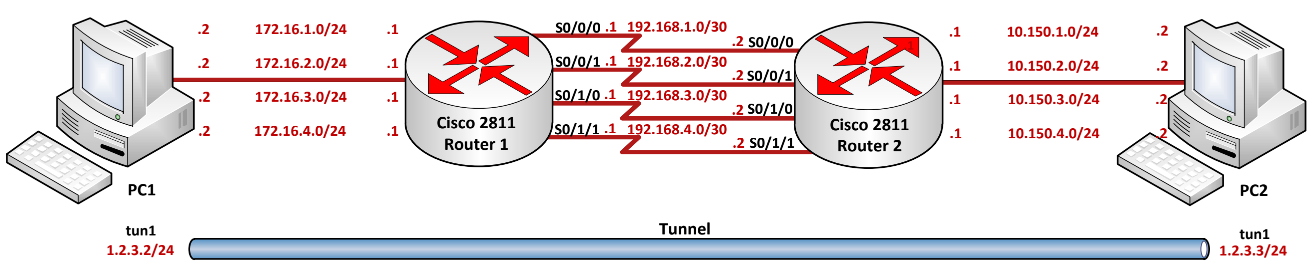

Fig. 2 The measurement environment

The host PC1 was used as the FTP server (starting the built in FTP server of the operating system). 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 measurements' results are shown in Fig. 3. – 7. These figures show the interface throughput values for the test cases using 20 MB file size. The same results were produced in the cases of using 10 MB file size (see Table I.).

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 I. 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.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

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 2% in all cases.)

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 the interface throughput and for the application layer's throughput: the efficiency is better than 96% in all cases. (see Table I)

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.

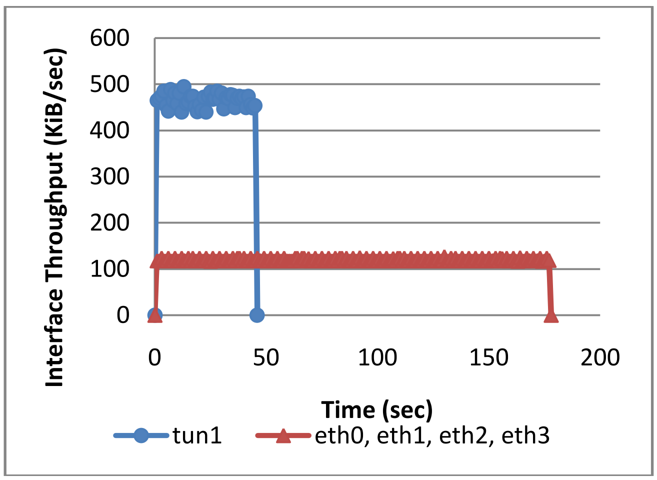

Fig. 3. Case 2. Data size: 20MB, Clock rates: 1M/1M/1M/1M

TABLE I.
FTP TRANSMISSION MEASUREMENT RESULTS

Case	Inter-face	Time (sec)	Throughput (KiB/s)	Efficiency (%)
Case 1. 10 MB 1 M / 1 M 1 M /1 M	tun1	22	476,6	99,9
	eth0-3	88	119,2	
Case 2. 20 MB 1 M / 1 M 1 M / 1 M	tun1	45	466,0	98,3
	eth0-3	177	118,5	
Case 3. 10 MB 2 M / 2 M 2 M / 2 M	tun1	11	953,3	99,9
	eth0-3	44	238,3	
Case 4. 20 MB 2 M / 2 M 2 M / 2 M	tun1	22	953,3	99,9
	eth0-3	88	238,3	
Case 5. 10 MB 2 M / 1 M 1 M / 1 M	tun1	18	582,5	97,7
	eth0	44	238,3	
	eth1-3	88	119,2	
Case 6. 20 MB 2 M / 1 M 1 M / 1 M	tun1	36	582,5	98,1
	eth0	88	238,3	
	eth1-3	177	118,5	
Case 7. 10 MB 2 M / 2 M 1 M / 1 M	tun1	15	699,1	97,7
	eth0-1	44	238,3	
	eth2-3	88	119,2	
Case 8. 20 MB 2 M / 2 M 1 M /1 M	tun1	30	699,1	98,0
	eth0-1	88	238,3	
	eth2-3	177	118,5	
Case 9. 10 MB 2 M / 2 M 2 M / 1 M	tun1	13	806,6	96,7
	eth0-2	44	238,3	
	eth3	88	119,2	
Case 10. 20 MB 2 M / 2 M 2 M / 1 M	tun1	26	806,6	96,7
	eth0-2	88	238,3	
	eth3	177	118,5	

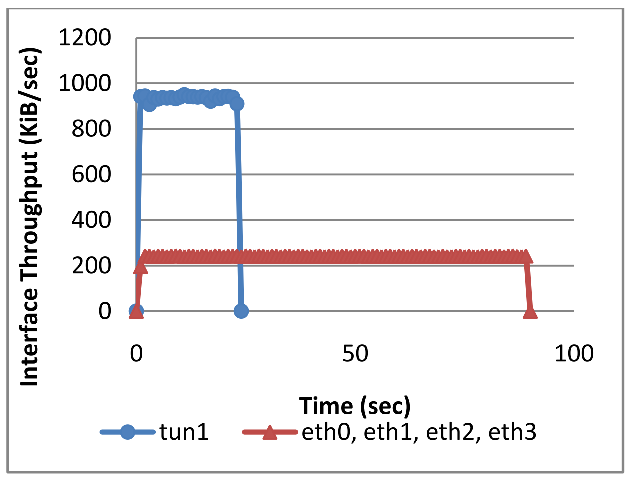

Fig. 4. Case 4. Data size: 20 MB, Clock rates: 2M/2M/2M/2M

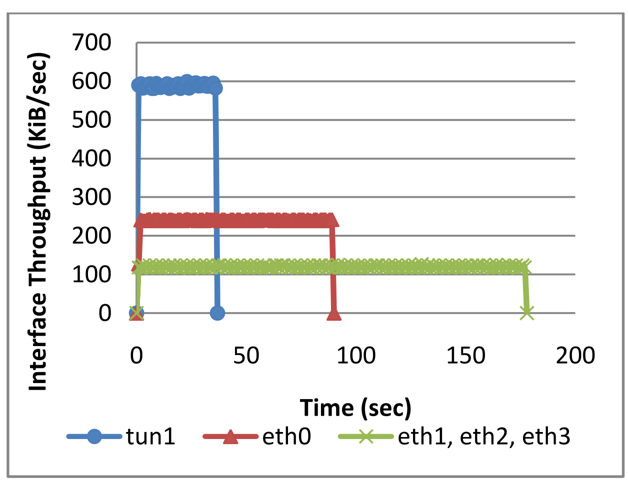

Fig. 5. Case 6. Data size: 20 MB, Clock rates: 2M/1M/1M/1M

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

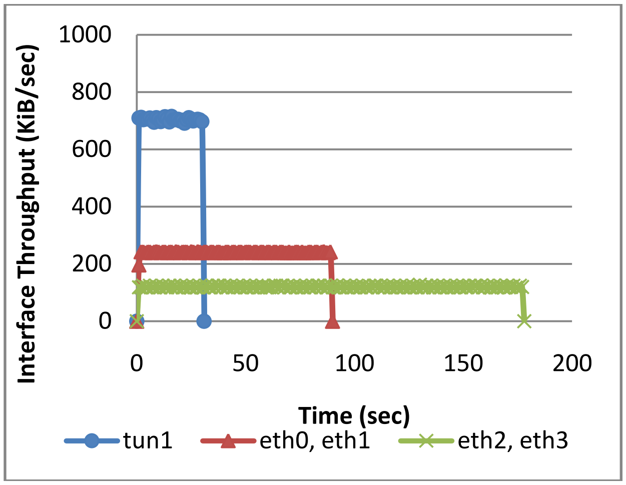

Fig. 6. Case 8. Data size: 20 MB, Clock rates: 2M/2M/1M/1M

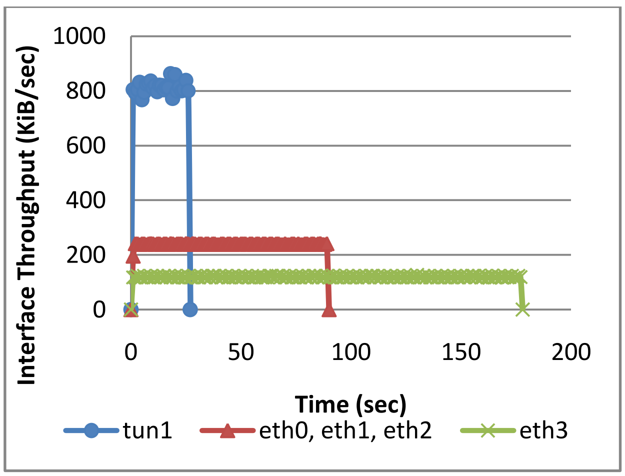

Fig. 7. Case 10. Data size: 20 MB, Clock rates: 2M/2M/2M/1M

IV. Media Stream Transfer Measurement Results

The previous sections confirmed that the MPT software library efficiently aggregates the throughput of four different paths in the case of TCP based file transmission. The results also showed, that the variance of the one second's throughput increased (comparing to the single path case). The increasing variance may result packet reordering in the multipath environment. If the packet realignment occurs inside a small window during the FTP transmission, then TCP will correct the problem, and the application will not sense it. Using UDP protocol in the transport layer (e.g. in the case of voice or video stream transmission) will produce a quite different behaviour: the reordering is not corrected by UDP, the application will sense the realignment, and must find a solution (e.g. discarding some packets).

In this section we investigate the performance of the MPT multipath environment in the case of stream transmission. We used the "iperf" traffic generator to start a stream transmission. The most interesting question is the behaviour of the multipath environment in the case of congested (or almost congested) situation. In order to present a congestion situation the clock rates of the serial links between the routers were decreased to 64000 and 128000 cycles per second.

The stream bitrate was chosen to 128, 256 and 512 kbps. The length of the stream was set to 60 seconds. The summary of the stream transmission measurement results can be seen in Table II.

In the single path transmission case choosing the clock rate of 128000 the system successfully transmitted a stream with 128 kbps bitrate (no packet loss, no reordering; see case 11).

TABLE II
STREAM TRANSMISSION MEASUREMENT RESULTS

Test No.	No. of paths	Clock rates (1000 cycles/second)	Stream bitrate (kbps)	Avg. throughput measured (kbps)	Avg. jitter (ms)	Lost ratio (%)	Realignment ratio (%)
Case 11.	1	128	128	123,49	2,33	0	0
Case 12.	1	128	256	123,49	13,84	48,26	0
Case 13.	4	128-128-128-128	128	124,67	0,42	0	0
Case 14.	4	128-128-128-128	256	249,67	0,42	0	0
Case 15.	4	128-128-128-128	512	486,26	8,24	0	0
Case 16.	4	64-128-64-128	128	124,98	61,22	0	32,81
Case 17.	4	64-128-64-128	256	249,45	60,73	0	33,36

Increasing the bitrate of the stream to 256 kbps produced a highly congested environment: after the 10th second (when the router's buffer got full, and the congestion was created) 50 percent of the packets was lost (see case 12).

Using four paths with the same clock rate value of 128000 on the paths the MPT software environment produced a very effective communicating environment: The stream with bitrate of 128 kbps and 256 kbps was successfully transmitted (no packet loss, no reordering). Even the jitter value was very small, less than 1 millisecond in these cases (see case 13, case 14). Choosing the stream bitrate to 512 kbps reached the limiting border of the MPT multipath environment (containing 4 paths with 128000 cycles per second clock rate at the bottleneck point): the packet transmission was successful (no packet loss, no reordering), but the value of jitter increased to 10 milliseconds due to some congestion, which appeared after 30 seconds, see case 15).

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

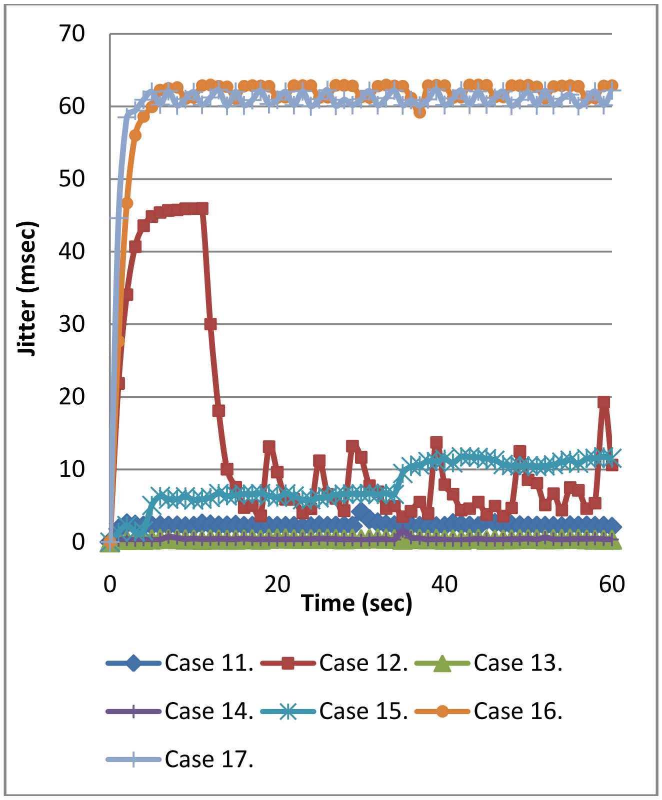

Fig. 8. Case 16. Media Stream Jitter Values

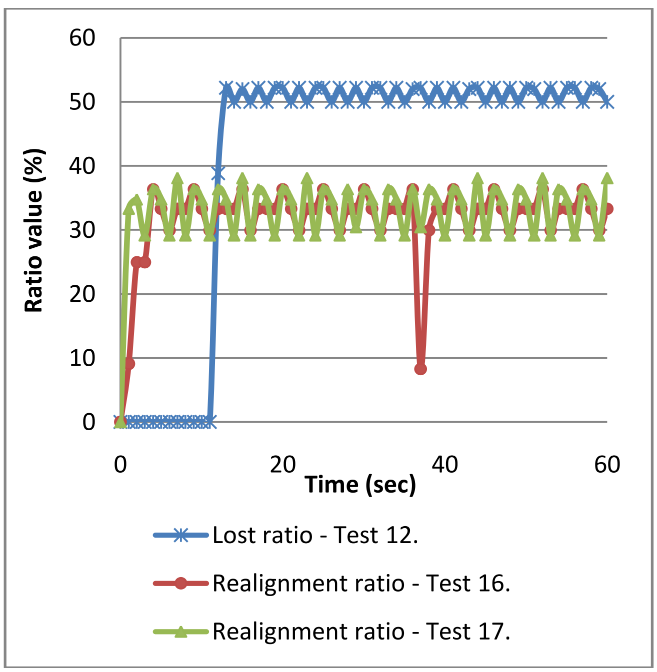

Fig. 9. Media Stream – Lost and Realignment ratio

In order to investigate a "non-equal paths" environment (i.e. 4 paths, with different throughput values at the bottleneck point) we set the clock rate to 64000 on two paths and to 128000 on the remaining two paths. Our measurement results confirm the results published in [8] for single thread multipath communication software: the jitter and the variance increases in the non-equal multipath environment. The most important problem related to the stream transmission in the "non-equal multipath" environment, that the source station starts a packet on the slower speed path (having clock rate of 64000), and then immediately a new packet is started on the faster speed path (having clock rate of 128000) the later started packet will arrive earlier to the destination than the previous packet: the realignment is sure in this case.

The transmitted packets are distributed between the faster and the slower paths according to the clock rate values of the paths (i.e. 66 percent of the packets on the faster paths and 33 percent of the packets on the slower paths in our test case). It comes out, that the packets started on the slower paths will be out of order. This statement holds for the stream transmission having of 128 kbps bitrate and having of 256 kbps bitrate too (see case 16, 17 and Fig 9.). We are able to successfully aggregate the throughput of different paths even in the case of stream transmission, but in the case of "non-equal paths" the reordering and the jitter's increasing are also involved behaviours.

V. CONCLUSION

In this paper we investigated the performance of a new multipath communication environment, the MPT library, considering the file and media stream transmission features. A dedicated measurement environment was created in order to perform the measurements.

The measurement results showed that the MPT environment is capable to efficiently aggregate the available throughput of four paths, even in the case of having quite different throughput values on the paths. This statement holds both for the file transmission and for the media transmission. Concerning the jitter value at the media stream transmission, we experienced that the jitter value is quite small if the aggregated paths speed is roughly the same, but it will increase as the speed of the paths becomes different. The MPT multipath software environment can be get from the authors of the paper. Analyzing the reliability of the communication sessions a hot research area too (see e.g. [9]), so further research work can be done to investigate the reliability of the multipath environment.

ACKNOWLEDGMENT

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.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

REFERENCES

[1] SangeethaPriya Mohan, Saradha Sekar, Devipriya Chinnasamy, Lavanya Krishnasamy, "Multipath Routing Using Reinforcement Learning Algorithm", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 9, pp 3404-3410, 2013.

[2] B. Almási, Sz. Szilágyi, "Throughput Performance Analysis of the Multipath Communication Library MPT", Proceedings of the 36th International Conference on Telecommunications and Signal Processing (TSP, ISBN:978-1-4799-0403-7), pp 86-90, Rome, Italy, 2013.

[3] Tejpreet Singh, Balpreet Kaur, Sandeep Kaur Dhanda, "Performance Evaluation of Routing Protocols in VANETs" International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 9, pp 3590-3594, 2013.

[4] Rongali Avataram, Dr.B.Prabhakara Rao, B.A.S.Roopa Devi, "A Hybrid Routing Mechanism for Fast and Secure Data Transmission in MANET", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 8, pp 3058-3063, 2013.

[5] Z. Gál, Gy. Terdik, "On the Statistical Analysis of Wireless Sensor vs. Wired Data Network Traffics", Carpathian Journal of Electronic and Computer Engineering, Vol. 4, No. 1, pp. 41-47, 2011.

[6] 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.

[7] M. Handley, O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses draft-ietf-mptcp-rfc6824bis-00", IETF Internet-Draft, October 20. 2013., Available https://datatracker.ietf.org/doc/draft-ietf-mptcp-rfc6824bis/

[8] B. Almási, "Multipath Communication – a new basis for the Future Internet Cognitive Infocommunication", Proceedings of the CogInfoCom 2013 Conference, Budapest, December 2013 (to appear).

[9] G. Lencse and I. Derka, "Investigation of the Fault Tolerance of the PIM-SM IP Multicast Routing Protocol for IPTV Purposes", Infocommunications Journal, Vol. V, No. 1. pp. 21-28, March, 2013.

BIOGRAPHY

Dr Béla Almási is an Associate Professor at the Faculty of Informatics, University of Debrecen in Debrecen, Hungary. His favorite research area is network communication, next generation networks, performance tuning and stochastic modeling of computer networks.

Szabolcs Szilágyi is an Assistant at the Faculty of Informatics, University of Debrecen in Debrecen, Hungary. His favorite research area is computer networks.