Chapter 15
Fuzzy Control for Secure TCP Transfer

Łukasz Apiecionek

Abstract This chapter presents the potential use of fuzzy observance implementation for detecting transmission problems that could appear in the near future. Using quick detection, appropriate action could be taken and the security and reliability of data transfer could be maintained at a high level. As a result the authors present a proposed solution for dividing a data stream between different data links and predicting transmission problems.

Keywords TCP · Multipath TCP · Security · Fuzzy logic · Fuzzy observation

15.1 Introduction

Nowadays many networks require security such as data encryption and reliable transfer to the destination. Health care, rail, and power plant systems are some examples of such systems. When they are used an operator cannot lose a connection between his or her control and management applications and system sensors and actuating equipment. In such a system a lost connection could have tragic consequences. In the case of power plants this could also mean some huge disaster. That is why security regarding reliable transfer is very important. In the case of health systems it could be required for monitoring people but also for execution of medical operations through remote control of medical equipment such as a scalpel. Some problems with rail equipment could cause a train accident. There are also many other systems in which a lost connection would cause real damage. Such systems are called critical infrastructure. For this critical infrastructure more than one connection is usually prepared. For example, those systems can operate using cable and a 3G/4G/LTE connection concurrently. Unfortunately, in most cases, different kinds of used connections are not operated simultaneously. Systems switch between them when the operating connection collapses. When the systems switch connections they can lose some data.

Ł. Apiecionek (✉)
Institute of Technology, Kazimierz Wielki University,
ul. Kopernika 1, 85-074 Bydgoszcz, Poland
e-mail: lapiecionek@ukw.edu.pl

P. Prokopowicz et al. (eds.), *Theory and Applications of Ordered Fuzzy Numbers*,
Studies in Fuzziness and Soft Computing 356, DOI 10.1007/978-3-319-59614-3_15

The loss of data can be very costly. In such systems there is much effort concerning appropriate data encryption. The systems are usually well protected against unauthorized access to the network [3, 10, 25, 40]. In most cases the data are encrypted before being forwarded to the communication transmission layer. The worst situation is when they lose their connection, as mentioned above. In order to ensure that the data reach their destination, transmission control protocols are used, but it could not be enough. When the connection collapses and the system switches to another one, the transmission of the lost packet has to be repeated [12, 13, 15]. In many cases, the system administrator does not know that use of more than one network connection can increase the transfer rate and security level of the data transfer process mentioned as reliability of the data transfer. How can it be achieved? In the situation when only one connection is used, a hacker can sniff all the packets of data in only one place on the network. Thus he or she could collect all the data and then try to decode them. After some time the real information can be recovered. When there is more than one network connection used, a hacker is forced to work on more spots to sniff the packets. More connection used in a parallel way could improve security because in this scenario, a hacker must be familiar with more than one transfer technology [11, 16, 22]. This chapter presents multipath transmission control protocols (MPTCP), which is currently ready-to-use technology. This technology can be used to achieve the aforementioned functionality. One of the huge advantages of MPTCP is the fact that it works at the operating system level. This makes it possible to use the existing application in a simple manner. MPTCP are presented with the three schedulers: the existing one, some secure proposition, and finally the one using Ordered Fuzzy Numbers (OFNs) for fast prediction problems in the transmission [20, 38, 60]. Such a presentation lets us introduce OFNs as a ready-to-use solution that could also be used in connection with other technology and solve some real problems in IP networks. The chapter is focused on OFN usage in already implemented technology such as MPTCP [26, 57, 62].

15.2 Multipath TCP

MPTCP uses the concept of transmission control protocols (TCP) [9, 24, 39, 50]. TCP transmission is used for delivering data between applications running on different machines on the network. TCP can be used to send data in both directions between two hosts using an established connection. A unique identifier is used to describe that connection. That identifier consists of two pairs of values (one for each side of the connection), IP and port number [5, 27, 28]. To achieve complete data and their appropriate order, checksums and sequence numbers are used. The mentioned data are shown in the TCP header presented in Fig. 15.1. When the application intends to establish a TCP connection, it has to exchange appropriate signals. This process is called a three-way handshake and is presented in Fig. 15.2. Host A sends a segment with a set SYN flag, then host B confirms the receipt of the packet and sends back SYN and ACK flags as a response. Finally, host A sends an empty segment, with

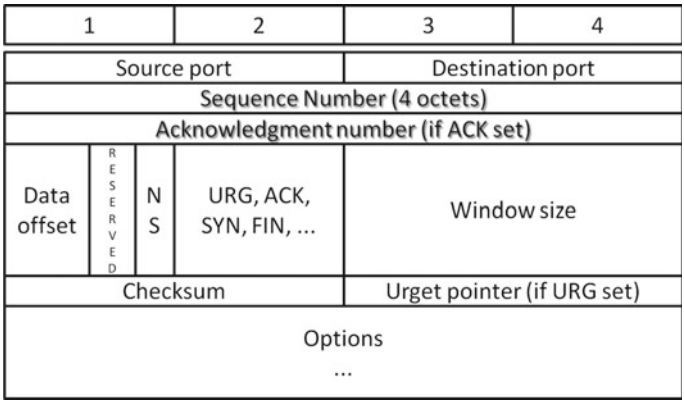

Fig. 15.1 TCP header

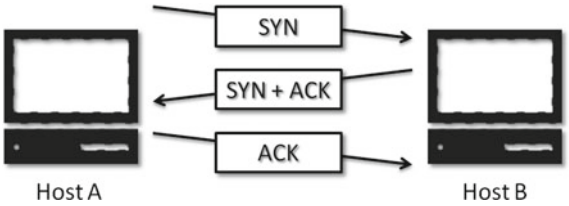

Fig. 15.2 Three-way handshake

only the ACK flag as a response to the previous message [34–36, 50]. One possible problem with TCP is the process of changing network connections which is associated with changing an IP address into another one. When a host switches from an Ethernet cable connection to Wi-Fi, it is assigned a different IP address. This triggers a process of closing the existing TCP connections and resuming them. MPTCP is characterized by a set of extensions to the specification of the existing TCP. These extensions enable the client to establish more than one connection while they each use different network cards, yet they are all used to reach the same destination host. The fact that fault-tolerant and efficient data connections are maintained this way between hosts that are compatible with the already used network infrastructures can be regarded as a big advantage of MPTCP. A possible way of establishing connection using network A and B is presented in Fig. 15.3. Another MPTCP advantage is that it increases the throughput of data transfer. This approach should significantly improve congestion balance between network paths. Simultaneous enabling of MPTCP must not prevent connectivity on a path where regular TCP operates [33, 37, 53, 54]. As already mentioned, MPTCP is located at the transport layer and it is intended to be transparent to other layers: higher and lower ones, as presented in Fig. 15.4. MPTCP can be treated as an additional function of higher layers of the TCP standard.

When a new connection of MPTCP should be established, a three-way handshake algorithm of TCP is used. This is presented in Fig. 15.5. The protocol is enhanced by a new feature that makes the difference as compared to standard TCP.

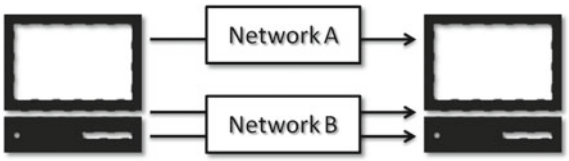

Fig. 15.3 N different TCP connections are represented as a single logical datum

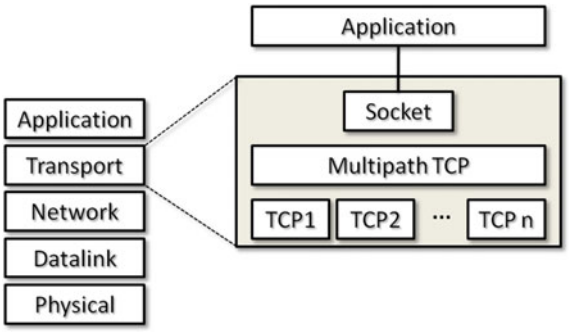

Fig. 15.4 MPTCP in the stack

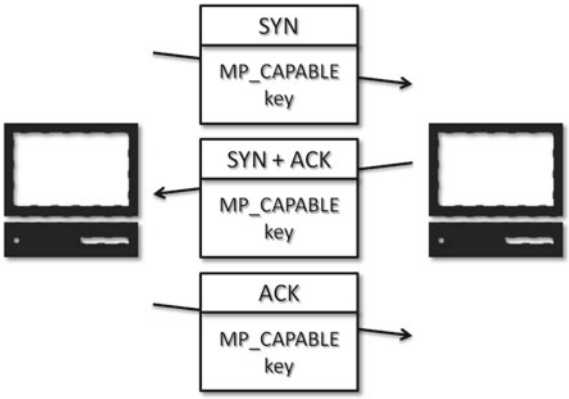

Fig. 15.5 Establishing connection

The MP_CAPABLE option informs both hosts if the MPTCP connection can be established and if the data can be transmitted. The IP networks encompass many routers and switches working as intermediate boxes. Those boxes could complicate the process of establishing connections. For this reason it is insufficient to identify the connection pair (IP address and port number) of the source and destination hosts. MPTCP has extended TCP functionality by adding another option called MP_JOIN. This option is used for generating a new subflow of data. The process of adding a new subflow is presented in Fig. 15.6.

The process of adding a new subflow is done in the following steps.

- In the first step the MP_JOIN option provides a token generated with the key (truncated hash of the key) created during the initial connection.

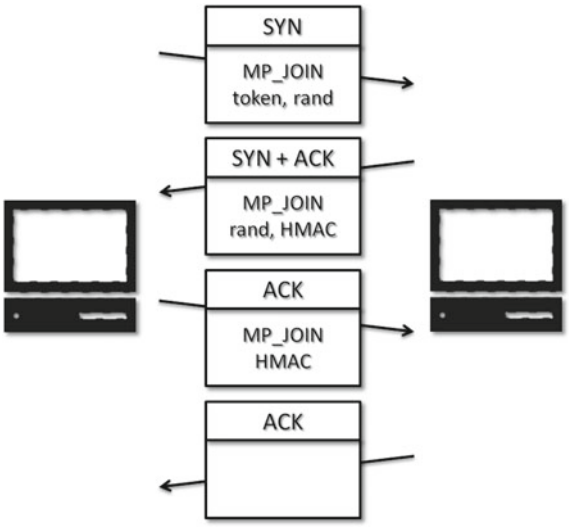

Fig. 15.6 Adding a new subflow into MPTCP

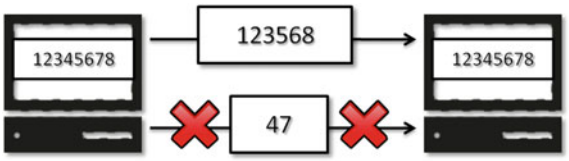

Fig. 15.7 Error control in MPTCP

- In the second step the exchange of HMAC (hash-based message authentication code) takes place.
- In the third step the subflows are established, and MPTCP can use them to exchange data.

Once the connection is established each host can send data over any of the subflows. Furthermore, Fig. 15.7 presents the data transmitted over one subflow. If, for example, a packet numbered 4 and 7 is lost it can be retransmitted to another subflow to recover the loss. Finally all the data packets reach the destination. There is a 'subflow sequence number' in standard TCP that supports the reception of a single subflow and ensures detection of any data loss. MPTCP uses "data sequence number" to sort the received data before passing them to the application [23, 51, 52, 55]. The MPTCP header is presented in Fig. 15.8. To inform the destination that the source has no more data to send, the source sends "Data FIN" signals. Its operation is exactly the same as a TCP FIN in standard TCP implementation.

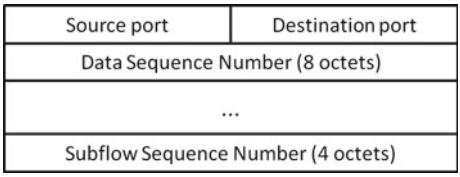

Fig. 15.8 MPTCP header (simplified diagram)

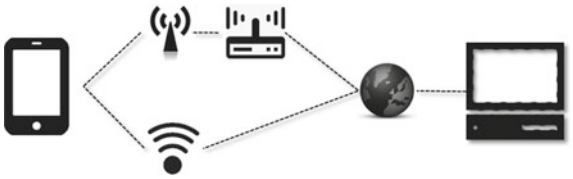

Fig. 15.9 MPTCP on smartphones

15.3 Multipath TCP Schedulers

Three schedulers are presented in the next three subsections: standard, secure, and with OFN usage.

15.3.1 Multipath TCP Standard Scheduler

In general, ordinary users who are connected to the Internet by their smartphones via Wi-Fi or a 3G network do not use these connections concurrently. They use them in series. The MPTCP is able to use both at the same time as shown in Fig. 15.9. If the standard TCP connection fails for some reason, it must be re-established. With MPTCP such a situation can be avoided by dynamic switching to the link. Therefore the user can avoid wasting time re-establishing connections. It enables the optimum data transfer rate selection.

The first mobile system that supports MPTCP [3, 31, 32, 56] is iOS 7. It ensures an uninterrupted transfer in case of failure of one connection or when the connection is aborted. At the moment, MPTCP is used in iOS 7 only for transfer of Siri data. Siri is an intelligent personal assistant for smartphone users. Such a system of scheduling connections was originally proposed by MTCP authors.

15.3.2 Multipath TCP Secure Scheduler

Another possible MPTCP scheduler is a secure scheduler. As follows from the literature, MPTCP is able to increase the security level of the transmitted data by application of many different links to reach the destination. This solution is contrary to the present methodology, which is based only on network protection and [1, 2,

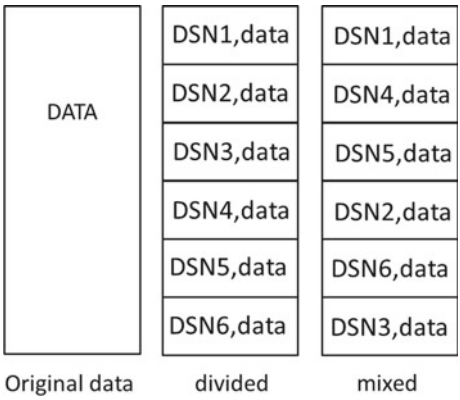

Fig. 15.10 Mixing data process

14, 17, 18, 61] on network access control [7, 8, 29, 30, 58]. This scheduler treats the transmitted IP packets as raw binary data, which can be divided into blocks and then passed to the transmission layer. As regards data protection from being sniffed by a hacker, the scheduling algorithm consists of the steps:

- Step 1. Data are divided into blocks.
- Step 2. Data are assigned a special sequence number, data sequence number (DSN).
- Step 3. Blocks are collected in a random sequence.
- Step 4. Data are encoded.
- Step 5. Blocks of data are passed to the MPTCP socket, which will transmit them to their destination.
- Step 6. Receiver side collects the blocks of data.
- Step 7. Data are decrypted.
- Step 8. Receiver side connects the blocks of data in an appropriate order.

The process of dividing the data into blocks, assigning a special DSN (data sequence number) to it, and putting it in a random sequence (Steps 2 and 3), is shown in Fig. 15.10. Step 5 of the proposed algorithm is presented in Fig. 15.11. The data passed to the MPTCP socket is transmitted using different data connections in a parallel way. In the vulnerable spots, where the data can be sniffed, a hacker is able to get only a portion of transmitted data. These data do not carry any clue as to what part of the original data they are [4, 6, 41].

The process of mixing the original data blocks uses random sequence and is performed on the sender's side, whereas the information about the appropriate sequence is passed to the receiver's side using the DSN.

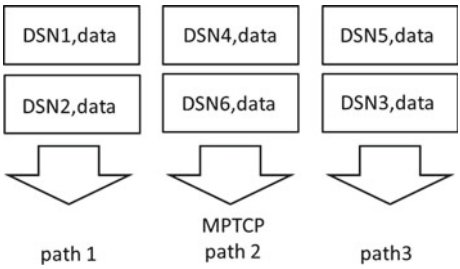

Fig. 15.11 Transmission process

15.3.3 Multipath TCP Scheduler with OFN Usage

As already mentioned above, MPTCP can increase network security regarding such parameters as a destination reachability and network reliability [42, 43, 48]. For any mentioned scheduler, a transmission error can occur at the used channel. The error can cause a need for data retransmission over the same channel, or if the number of errors grows, the channel can be closed and another connection used. Use of OFNs can increase the time of the data transmission link change or can decrease the number of retransmissions. OFNs can be used for predicting data loss in the used channel and may accelerate the decision on some changes such as quicker retransmission of packets or use of a different channel [45–47].

15.3.4 OFN for Problem Detection

An algorithm has been proposed for OFN use for detecting future problems in the used connection [19, 21, 44, 49, 59]. For this purpose the algorithm should measure a TCP retransmission in all used channels during the transmission as a percentage value of transmitted packets (during the given timeslot). This measurement should be continuous and statistics should be taken for specific timeslots. Four timeslots of a continuous measurement can be defined as follows.

$$t_i, t_{(i-1)}, t_{(i-2)}, t_{(i-3)} \tag{15.1}$$

where t_i is a current timeslot.
All four measurements together make up a fuzzy number in OFN notation where

- $f_A(0)$ corresponds to $t_{(i-3)}$.
- $f_A(1)$ corresponds to $t_{(i-2)}$.
- $g_A(1)$ corresponds to $t_{(i-1)}$.
- $g_A(0)$ corresponds to t_i.

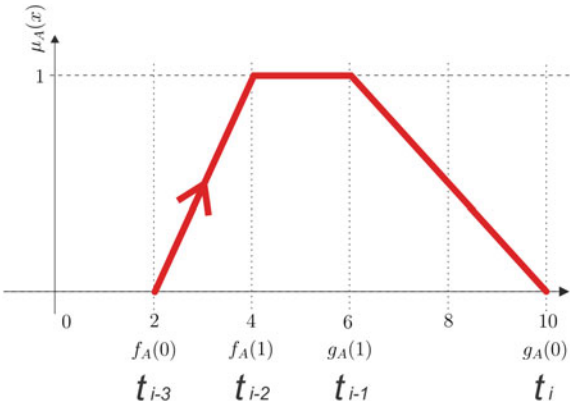

Fig. 15.12 Fuzzy number in OFN notation

That fuzzy number in OFN notation is presented in Fig. 15.12. This is a definition of a fuzzy observance of a connection.

Definition 1 Fuzzy observance of C router in time t_i is a set

$$C/t_i = \{f_C(0)/t_{i-3}, f_C(1)/t_{i-2}, g_C(1)/t_{i-1}, g_C(0))/t_i\} \tag{15.2}$$

where

$$t_i > t_{i-1} > t_{i-2} > t_{3-1}$$
$$|t_i - t_{i-1}| = |t_{i-1} - t_{i-2}| = |t_{i-2} - t_{i-3}| = t_n, \text{timeslot of the measurement}$$
$$f_C(0) \leq f_C(1) \leq g_C(1) \leq g_C(0)$$

This provides Lemma 1.

Lemma 1

$$C_{positive} = \begin{cases} f_C(0) < f_C(1) < g_C(1) \\ or \\ f_C(1) < g_C(1) < g_C(0) \end{cases} \tag{15.3}$$

In other situations $C_{negative}$.

According to this definition, during observance of connections the counters should give:

- Positive order of OFN when the packet retransmission count **increases**
- Negative order of OFN when the packet retransmission count **decreases**

The interpretations of those orders are presented in Fig. 15.13. Then the statistics collected at each connection provide results for fuzzy number preparation. Fuzzy observance of the MPTCP connections can also be defined. Fuzzy observance of the MPTCP connections is defined as follows.

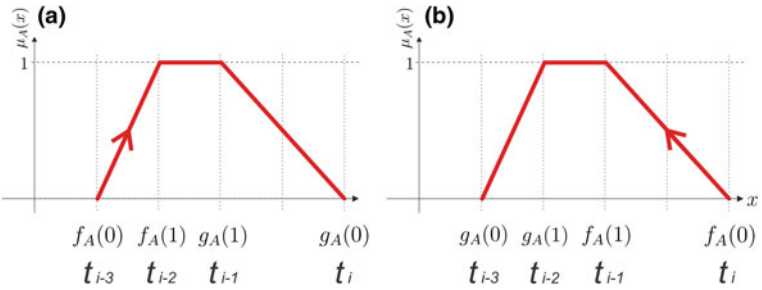

Fig. 15.13 Order interpretation in OFN notation

Definition 2 Fuzzy observance of the MPTCP connections is defined by the formula:

$$S_m = \sum_{i=1}^{n} \left\{ \begin{matrix} R_{positive} | R_{negative} \\ R_i * w_i | - R_i * w_i \end{matrix} \right\}. \tag{15.4}$$

where $w_i \in \{w_i, ..., w_n\}$ describes an impact on all connections.

This makes it possible to define the MPTCP scheduler with OFNs.

15.4 OFN Scheduler Algorithm

An algorithm proposed as OFNs used for transmission error anticipation consists of the following steps.
Step 1. Administrator declares w_i and L_i for all used connections, where w_i describes an impact on all connections, and L_i describes the load of all data that should be sent by those connections when the transmission starts. L_i should be provided as a percentage value.
Step 2. The amount of packets P_i that will be transferred over each connection for each timeslot is calculated using the formula:

$$P_i = \frac{L_i}{\sum_{i=1}^{n}} * Data \tag{15.5}$$

Step 3. During the transmission C_i is calculated for each connection according to data retransmissions and S_i is calculated according to the given definition.
Step 4. When the calculated S_i is positive and exceeds the acceptance level AL, there is an error increase detected on this connection. In this situation L_i for a given connection will be changed according to the formula:

$$L_i = \frac{L_i}{ErrorCorector} \tag{15.6}$$

When the calculated S_i is negative, there is an error decrease detected on this connection. In this situation L_i for a given connection will be changed according to the formula:

$$L_i = L_i * ErrorCorector \tag{15.7}$$

The *ErrorCorector* is a value that describes how quickly the system should stop using a given connection in which the amount of errors has increased. This value should also be provided by the network administrator.

15.5 Simulation Test Results

To check a MPTCP scheduler with OFNs, some simulations were made. The system has got two connection links. Connection 1, labeled C1, was a Wi-Fi connection with maximum rate of 11 Mbit/s. The second connection used, labeled C2, was an LTE connection with the maximum rate of 5 Mbit/s. The parameters of the algorithm were:

- Corrector for the links $ErrorCorector = 2$.
- Acceptance level $AL = 3$.
- Load balance at the start for the connection $C1$ was $L_1 = 66$.
- Load balance at the start for the connection $C2$ was $L_2 = 34$.
- 60-second timeslots were used.

The results obtained by the applied algorithm according to load balancing between connections during data transfer are presented in Table 15.1. There were errors on measured data links and the OFN was calculated according to the presented algorithm. The number of packets transferred over each link was modified according to the level of errors and OFN order. When the percentage of errors increased, the number of packets passed to the link with problems (C2) was decreased. Obviously, the OFN was calculated after four timeslots.

Table 15.2 shows the number of packets passed to the connections and the number of packets that had to be retransmitted due to an error on the link when the MPTCP with and without the OFN algorithm was used. Note that the percentage of errors on the C2 link decreased. That is why there were fewer packets transferred during the network problems. The number of errors on the C1 connection increased because there were more packets transferred through this link. The most important column is a sum of errors in both links. When the algorithm decreased the number of packets passed through the C2 link, the sum of errors decreased even if the number of errors on the C2 link increased. The final results prove that the number of errors in the transmission can be decreased using OFNs in the MPTCP scheduler.

Table 15.1 Normalized packets count on routers during test

Time slots	L - load balance		% Error on connection		S			
	L1	L2	C1	C2	S1	Order	S2	Order
1	66	34	2	3				Order
2	66	34	2	3				
3	66	34	2	5				
4	66	34	1	8	[2,1,2,1]	Positive	[3,4,5,8]	Positive
5	66	17	1	12	[1,2,1,1]	Negative	[4,5,8,12]	Positive
6	66	8.5	2	11	[2,1,1,2]	Positive	[5,8,12,11]	Positive
7	66	4.25	2	10	[1,1,2,2]	Positive	[8,12,11,10]	Positive
8	66	2.125	1	11	[1,2,2,1]	Positive	[12,11,10,11]	Negative
9	66	4.25	2	9	[2,2,1,2]	Negative	[11,10,11,9]	Negative
10	66	8.5	1	8	[2,1,2,1]	Positive	[10,11,9,8]	Negative
11	66	17	1	7	[1,2,1,1]	Positive	[11,9,8,7]	Negative
12	66	34	1	7	[2,1,1,1]	Positive	[9,8,7,3]	Negative
13	66	34	3	2	[1,1,1,2]	Positive	[8,7,3,2]	Negative
14	66	34	3	3	[1,1,2,2]	Positive	[7,3,2,3]	Negative
15	66	34	3	2	[1,2,2,2]	Positive	[3,2,3,2]	Negative

Table 15.2 Number of packets and errors during the transmission

Time slots	Packet count		Error count					
			MPTCP with KFC			MPTCP without KFC		
	C1	C2	C1	C2	C1+C2	C1	C2	C1+C2
1	52800	27200	1056	816	1872	1056	816	1872
2	52800	27200	528	1088	1616	528	1088	1616
3	52800	27200	1056	1360	2416	1056	1360	2416
4	52800	27200	528	2176	2704	528	2176	2704
5	63614	16386	636	1966	2602	528	3264	3792
6	70870	9128	1417	1004	2421	1056	2992	4048
7	75160	4840	1503	484	1987	1056	2720	3776
8	77505	2495	755	275	1050	528	2992	3520
9	75160	4840	1503	436	1939	1056	2448	3504
10	70872	9128	709	730	1439	528	2176	2704
11	63614	16386	636	1147	1783	528	1904	2432
12	52800	27200	528	816	1344	528	816	1344
13	52800	27200	1056	544	1600	1056	544	1600
14	52800	27200	1056	816	1872	1056	816	1872
15	52800	27200	1056	544	1600	1056	544	1600

15.6 Conclusions

The new concept of an MPTCP scheduler using OFNs presented herein was tested during a data transfer simulation. As shown in the previous section, with the proposed algorithm it is possible to decrease the retransmission count. This could be achieved because there were fewer packets transferred over the connection link where some problems were detected. This is a potential use of OFNs in a simple way intended to improve existing solutions such as MPTCP without complicated algorithms that require a great deal of processor capacity. The other advantages of using an OFN scheduler are that it could be connected with the presented secure scheduler and coexist on the transmissions. Such solutions present the huge potential of OFNs.

References

1. Apiecionek, L., Czerniak, J.: Qos solution for network resource protection. In: Proceedings of International Scientific Conference INFORMATICS 2013, 5-7 November 2013, Spisská Nová Ves, Slovakia (2013)
2. Apiecionek, L., Czerniak, J., Dobrosielski, W.: Quality of services method as a DDoS protection tool. In: Advances in Intelligent Systems and Computing. Springer, Berlin (2015)
3. Apiecionek, L., Romantowski, M.: Secure ip network model. Comput. Method Sci. Technol. **19**(4), 209–213 (2013)
4. Apiecionek, L., Czerniak, J.M.: Qos solution for network resource protection. In: Informatics 2013: Proceedings of the Twelfth International Conference on Informatics, pp. 73–76 (2013)
5. Apiecionek, L., Czerniak, J.M., Dobrosielski, W.T.: Quality of services method as a DDoS protection tool. In: Intelligent Systems'2014, vol. 2: Tools, Architectures, Systems, Applications, vol. 323, pp. 225–234 (2015)
6. Apiecionek, L., Czerniak, J.M., Zarzycki, H.: Protection tool for distributed denial of services attack. In: Beyond Databases, Architectures and Structures, BDAS 2014, vol. 424, pp. 405–414 (2014)
7. Chapman, B., Zwicky, E.: Building Internet Firewalls. O'Reilly & Associates, Inc. (1995)
8. Cheswick, W., Bellovin, S.: Firewalls and Internet Security: Repelling the Wily Hacker. Addison-Wesley Publishing Company (1994)
9. Czabanski, R., Jezewski, J., Horoba, K., Jezewski, M.: Fetal state assessment using fuzzy analysis of fetal heart rate signals – agreement with the neonatal outcome. Biocybern. Biomed. Eng. **33**(3), 145–155 (2013)
10. Czabanski, R., Jezewski, M., Horoba, K., Jezewski, J., Leski, J.: Fuzzy analysis of delivery outcome attributes for improving the automated fetal state assessment. Appl. Artif. Intell. **30**(6), 556–571 (2016)
11. Czerniak, J.M., Dobrosielski, W., Zarzycki, H., Apiecionek, L.: A proposal of the new owlant method for determining the distance between terms in ontology. In: Intelligent Systems'2014, vol. 2: Tools, Architectures, Systems, Applications, vol. 323, pp. 235–246 (2015)
12. Czerniak, J.M., Dobrosielski, W.T., Apiecionek, Ł., Ewald, D., Paprzycki, M.: Practical Application of OFN Arithmetics in a Crisis Control Center Monitoring, pp. 51–64. Springer International Publishing, Cham (2016)
13. Czerniak, J.M., Ewald, D.: A new mglaber approach as an example of novel artificial acari optimization, pp. 545–557 (2016)
14. Czerniak, J., Apiecionek, Ł., Zarzycki, H., Ewald, D.: Proposed caeva simulation method for evacuation of people from a buildings on fire. Adv. Intell. Syst. Comput. **401**, 315–326 (2016)

15. Czerniak, J., Dobrosielski, W., Apiecionek, L.: Representation of a trend in ofn during fuzzy observance of the water level from the crisis control center. In: Proceedings of the Federated Conference on Computer Science and Information Systems, IEEE Digital Library, ACSIS **5**, 443–447 (2015)
16. Czerniak, J., Ewald, D., Macko, M., Smigielski, G., Tyszczuk, K.: Approach to the monitoring of energy consumption in eco-grinder based on abc optimization. In: Beyond Databases, Architectures and Structures, BDAS 2015, vol. 521, pp. 516–529 (2015)
17. Czerniak, J., Macko, M., Ewald, D.: The cutmag as a new hybrid method for multi-edge grinder design optimization. Adv. Intell. Syst. Comput. **401**, 327–337 (2016)
18. Czerniak, J., Smigielski, G., Ewald, D., Paprzycki, M.: New proposed implementation of abc method to optimization of water capsule flight. In: Proceedings of the Federated Conference on Computer Science and Information Systems, IEEE Digital Library, ACSIS **5**(489-493) (2015)
19. Dobrosielski, W.T., Szczepanski, J., Zarzycki, H.: A proposal for a method of defuzzification based on the golden ratio - gr. In: Novel Developments in Uncertainty Representation and Processing, pp. 75–84. Springer International Publishing (2016)
20. Dyczkowski, K.: A less cumulative algorithm of mining linguistic browsing patterns in the world wide web (2007)
21. Dyczkowski, K., Wygralak, M.: On triangular norm-based generalized cardinals and singular fuzzy sets. Fuzzy Sets Syst. **133**(2), 211–226 (2003)
22. Ewald, D., Czerniak, J.M., Zarzycki, H.: Approach to solve a criteria problem of the abc algorithm used to the wbdp multicriteria optimization. In: Intelligent Systems'2014, vol. 1: Mathematical Foundations, Theory, Analyses, vol. 322, pp. 129–137 (2015)
23. Ford, A., Raiciu, C., Handley, M., Bonaventure, O.: TCP extensions for multipath operation with multiple addresses. Technical report (2013)
24. Jezewski, M., Czabanski, R., Horoba, K., Leski, J.: Clustering with pairs of prototypes to support automated assessment of the fetal state. Appl. Artif. Intell. **30**(6), 572–589 (2016)
25. Jezewski, M., Leski, J.M., Czabanski, R.: Classification Based on Incremental Fuzzy $(1 + p)$-Means Clustering, pp. 563–572. Springer International Publishing, Cham (2016)
26. Kacprzak, D., Kosiński, W.: Optimizing firm inventory costs as a fuzzy problem. Stud. Log. Gramm. Rhetor. **37**, 89–105 (2014)
27. Kacprzak, D., Kosiński, W., Kosiński, W.K.: Financial stock data and ordered fuzzy numbers. In: Artificial Intelligence and Soft Computing: 12th International Conference, ICAISC'2013, pp. 259–270. IEEE (2013)
28. Kacprzak, D., Kosiński, W., Prokopowicz, P.: Fuzziness - representation of dynamic changes by ordered fuzzy numbers. Stud. Fuzziness Soft Comput. **243**, 485–508 (2009)
29. Kacprzak, M., Kosiński, W.: On lattice structure and implications on ordered fuzzy numbers. In: Proceedings of EUSFLAT. Artificial Intelligence and Soft Computing. LNCS, vol. 7267, pp. 247–255 (2011)
30. Kacprzak, M., Starosta, B., Węgrzyn-Wolska, K.: Metasets and opinion mining in new decision support system. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) Artificial Intelligence and Soft Computing, Part II. Lecture Notes in Artificial Intelligence, vol. 9120, pp. 625–636. Springer International Publishing (2015)
31. Kacprzak, M., Starosta, B., Węgrzyn-Wolska, K.: New approach to decision making. In: Abraham, A., Wegrzyn-Wolska, K., Hassanien, A.E., Snasel, V., Alimi, A.M. (eds.) Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, Advances in Intelligent Systems and Computing, vol. 427, pp. 397–407. Springer International Publishing (2015)
32. Kosinski, W., Chwastyk, A.: Ordered fuzzy numbers in financial stock and accounting problems. In: Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 546–551 (2013)
33. Kosiński, W., Prokopowicz, P., Rosa, A.: Defuzzification functionals of ordered fuzzy numbers. IEEE Trans. Fuzzy Syst. **21**(6), 1163–1169 (2013). doi:10.1109/TFUZZ.2013.2243456
34. Kosinski, W., Prokopowicz, P., Slezak, D.: Calculus with fuzzy numbers. In: Bolc, L., Michalewicz, Z., Nishida, T. (ed.) Intelligent Media Technology for Communicative Intelligence. Lecture Notes in Artificial Intelligence, vol. 3490, pp. 21–28, 2nd Intenational Workshop

on Intelligent Media Technology for Communicative Intelligence, Warsaw, POLAND, 13–14 September 2004 (2004)
35. Kosinski, W., Prokopowicz, P.: Fuzziness - representation of dynamic changes? In: Stepnicka, M., Novak, V., Bodenhofer, U. (ed.) New Dimensions in Fuzzy Logic and Related Technologies, vol. I, Proceedings, pp. 449–456. European Society for Fuzzy Logic & Technology, University Ostrava, Ostravska University & Ostrave, Dvorakova 7, Ostrava 1, 701 03, Czech Republic, 5th Conference of the European-Society-for-Fuzzy-Logic-and-Technology, Ostrava, Czech Republic, 11–14 September 2007 (2007)
36. Kosiński, W., Prokopowicz, P., Ślęzak, D.: On algebraic operations on fuzzy reals. In: Neural Networks and Soft Computing: Proceedings of the Sixth International Conference on Neural Networks and Soft Computing, Zakopane, Poland, June 11–15, 2002, pp. 54–61. Physica-Verlag HD, Heidelberg (2003). doi:10.1007/978-3-7908-1902-1_8
37. Krieger, U.: Evolution of transport protocols in high-speed networks. Lectures materials Multimedia-Kommunikation in Hochgeschwindigkeitsnetzen (KTR-MMK-M) (2014)
38. Lebiediewa, S., Zarzycki, H., Dobrosielski, W.: A new approach to the equivalence of relational and object-oriented databases. In: Novel Developments in Uncertainty Representation and Processing, pp. 85–93. Springer International Publishing (2016)
39. Leski, J.M.: Neuro-fuzzy system with learning tolerant to imprecision. Fuzzy Sets Syst. **138**(2), 427–439 (2003)
40. Leski, J.M.: On support vector regression machines with linguistic interpretation of the kernel matrix. Fuzzy Sets Syst. **157**(8), 1092–1113 (2006)
41. Marszalek, A., Burczyński, T.: Modeling and forecasting financial time series with ordered fuzzy candlesticks. Inf. Sci. **273**, 144–155 (2014). http://www.sciencedirect.com/science/article/pii/S0020025514003107
42. Marszalek, A., Burczyński, T.: Financial fuzzy time series models based on ordered fuzzy numbers. In: Pedrycz, W., Chen, S.M. (eds.) Time Series Analysis, Modeling and Applications: A Computational Intelligence Perspective, pp. 77–95. Springer, Berlin (2013)
43. Marszalek, A., Burczyński, T.: Modelling financial high frequency data using ordered fuzzy numbers. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) Artificial Intelligence and Soft Computing: 12th International Conference, ICAISC 2013, Zakopane, Poland, June 9–13, 2013, Proceedings, Part I, pp. 345–352. Springer, Berlin (2013)
44. Mikolajewska, E., Mikolajewski, D.: Wheelchair development from perspective of physical therapists and biomedical engineers. Adv. Clin. Exp. Med **19**(6), 771–776 (2010)
45. Mikolajewska, E., Mikolajewski, D.: Neuroprostheses for increasing disabled patients' mobility and control. Adv. Clin. Exp. Med. **21**(2), 263–272 (2012)
46. Mikolajewska, E., Mikolajewski, D.: Ethical considerations in wider use of brain-computer interfaces in patients with neurological deficits. Central Eur. J. Med. **8**(6), 720–724 (2013)
47. Mikolajewska, E., Mikolajewski, D.: Integrated it environment of disabled people - a new concept. Central Eur. J. Med. **9**(1), 177–182 (2014)
48. Mikolajewska, E., Mikolajewski, D.: The prospects of brain-computer application in children. Central Eur. J. Med. **9**(1), 74–79 (2014)
49. Mikolajewski, D., Mikolajewska, E.: Exoskeletons in neurological diseases - current and potential future applications. Adv. Clin. Exp. Med. **20**(2), 227–233 (2011)
50. Postel, J.: Transmission control protocol (1981)
51. Prokopowicz, P.: Methods based on ordered fuzzy numbers used in fuzzy control. In: Proceedings of the Fifth International Workshop on Robot Motion and Control, 2005. RoMoCo '05, pp. 349–354, June 2005. doi:10.1109/ROMOCO.2005.201448
52. Prokopowicz, P.: Adaptation of rules in the fuzzy control system using the arithmetic of ordered fuzzy numbers. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds.) Artificial Intelligence and Soft Computing - ICAISC 2008. Lecture Notes in Computer Science, vol. 5097, pp. 306–316. Springer, Berlin (2008). doi:10.1007/978-3-540-69731-2_30
53. Prokopowicz, P.: Flexible and simple methods of calculations on fuzzy numbers with the ordered fuzzy numbers model. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz,

R., Zadeh, L., Zurada, J. (eds.) Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science, vol. 7894, pp. 365–375. Springer, Berlin (2013). doi:10.1007/978-3-642-38658-9_33
54. Prokopowicz, P.: Analysis of the changes in processes using the kosinski's fuzzy numbers. In: Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.) Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, Annals of Computer Science and Information Systems, vol. 8, pp. 121–128. IEEE (2016). doi:10.15439/2016F140
55. Prokopowicz, P.: The directed inference for the kosinski's fuzzy number model. In: Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, pp. 493–503. Springer International Publishing, Cham (2016). doi:10.1007/978-3-319-29504-6_46
56. Schneider, K.M., Mast, K., Krieger, U.R.: Evolution of transport protocols in high-speed networks. Lectures materials Multimedia-Kommunikation in Hochgeschwindigkeitsnetzen (KTR-MMK-M) (2014)
57. Smigielski, G., Dygdała, R., Zarzycki, H., Lewandowski, D.: Real-time system of delivering water-capsule for firefighting. In: Advances in Intelligent Systems and Computing, vol. 534, pp. 102–111. Springer International Publishing (2016)
58. Sobol, I., Kacprzak, D., Kosiński, W.: Optimizing of a company's cost under fuzzy data and optimal orders under dynamic conditions. Optimum. Studia Ekonomiczne **5**, 172–187 (2014)
59. Stachowiak, A., Dyczkowski, K.: A similarity measure with uncertainty for incompletely known fuzzy sets. In: Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 390–394 (2013)
60. Stachowiak, A., Dyczkowski, K., Wojtowicz, A., Zywica, P., Wygralak, M.: A bipolar view on medical diagnosis in ovaexpert system (2016)
61. Vokorokos, L., Ennert, M., Hartinger, M., Radušovskỳ, J.: A survey of parallel intrusion detection on graphical processors. In: Proceedings of International Scientific Conference INFORMATICS 2013, 5-7 November 2013, Spisská Nová Ves, Slovakia (2013)
62. Zarzycki, H., Czerniak, J., Lakomski, D., Kardasz, P.: Performance comparison of CRM systems dedicated to reporting failures to it department. In: Software Engineering: Challenges and Solutions, Advances in Intelligent Systems and Computing, vol. 504, pp. 133–146. Springer International Publishing (2016)