INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(5), 704-714, October 2017.

Delay Tolerant Networks over Near Field Communications: The Automatic Multi-packet Communication

R.O. Schoeneich, P. Sadło

Radosław Olgierd Schoeneich*
Institute of Telecommunications
Warsaw University of Technology, Poland
*Corresponding author: rschoeneich@tele.pw.edu.pl

Piotr Sadło
Institute of Informatics
Warsaw University of Technology, Poland
p.sadlo@stud.elka.pw.edu.pl

Abstract: The Near Field Communication (NFC) is designed for sending small data, which size does not exceed the several dozen bytes, for distance not higher than two inches. The NFC can be used in store-carry-forward scenarios which are bases for Disruptive Tolerant Networks (DTN). The aim of this paper is proposal how to use NFC in transmission of large data in DTN. We propose an algorithm of fragmenting, sending and receiving data. We described software which was used in our prototype and discussed about implementation. Finally we showed results of tests performed on our model. Obtained results presents well characteristics of our solution.

Keywords: mobile communication, radio communication, wireless networks, personal communication networks, cooperative communication.

1 Introduction

Nowadays, an attention of scientists working on ad-hoc networks is focused on Delay and Disruptive Tolerant Networks (DTN). The DTN is a specific network in which nodes are mobile, therefore the network topology is often changeable what can be reason of lack of end-to-end path between source and destination nodes. The communication of DTN is done by storing-carrying and forwarding messages.

Applications of DTN are specific: previously it was proposed for interplanetary communication [1], but now it is used in military [14], search and rescue [15], wildlife monitoring [10], body sensing [17] and in many other solutions [21] [13].

The DTN transmission is based on wireless interfaces which are usually long range like radio satellite communications and medium range e.g. WiFi and Bluetooth standards. One of the possible communication interface, which fulfills DTN requirements of wireless medium and easy applicability of store-carry-forward paradigm is a very short range - Near Field Communication (NFC) [24].

NFC is used to transfer small packets on a very short-range. Transmission throughput is very low. This standard is widely used for cashless payment and short tag exchange. It is applicable not only on cards and tags but also on small mobile devices e.g. smartphones. Unlike other wireless communication solutions, NFC is able to setup sessions between devices automatically. The usage of NFC can be effective implemented in the DTN environment. Automatic and secret communication ability is advantage which supports the implementation of DTN objectives.

The aim of this paper is to present solution how to use the NFC to communicate DTN devices using store-carry-forward without data size limitation. For this purpose, we proposed a simple way of communication with the automatic exchange data between devices. We designed and implemented solution how to apply NFC without size limit of the transferred data.

The rest of this paper looks as follows: in section 2 we presented the state of the art: we discuss NFC technology, basics of DTN, and other telecommunication issues. In section 3 we present the algorithm of automatic establishment communication session and a mechanism for transferring big packets of data. In the section 4 we present results of tests performed on our prototype. Chapter 5 contains a conclusions.

2 State of the art

The Near Field Communication is a radio communication standard, based on RFID technology (Radio-Frequency Identification). It allows to wirelessly transfer data on a distance up to 20 cm [2]. In practice, the exchange of information over distances is smaller than 4 cm.

NFC can operate in two modes: active and passive. In the active mode, device alternatively communicate with each other. Both devices need power energy. The device which is at standby state does not emit signals. In passive mode, only one device can initiate the connection. Passive devices cannot read data by themselves; however, they can store it. NFC applications are focused on widely used wireless cashless payment. Contactless payment [16] is done by cards and smartphones. Other area of NFC usage are applications working with NFC tags with web addresses [23], tags for indoor navigation or information and wide aspects of smart homes e.g. office and home secure, body area communications [25]. All of this applications are using very short NFC messages and no one extends the basic communication functionality.

Delay and Disruptive Tolerant Networks firstly was proposed by K. Fall [7] for interplanetary communication. The basic idea is to provide communications in scenarios in which traditional telecommunication approach does not work. The DTN networks evaluate based on wireless ad-hoc and sensor network (MANET and WSN) approach: every node acts as a part of network and can route a message. Based on this for MANET and WSN there was proposed several routing protocols [20] [19] and algorithms which supports data handling [8] [9] [12] [5] [3] and transmission [22]. The DTN adds the new assumption to the Ad-hoc approach: store-carry-forward messages. Storing and forwarding messages is done by nodes called message ferries or mules. As a message ferry can act mobile device associated with human [4], animal or mobile machine [6]. By analogy to MANET, DTN requires specific solutions for routing and data handling [11] [18].

All this solutions are working with popular communication standards like WiFi and Bluetooth. As authors know none of these solution is designed for very short range solution like NFC.

3 DTN over NFC - MultiPacket communication

This section describes algorithms for sending and automatic receiving large data. Algorithms are implemented on Android environment. We propose automatic solution that splits large data into elementary packets and avoids "touch to send" restriction that is mandatory in Android NFC API.

3.1 Message fragmentation into NFC packet

Each message before sending must be adopted to the size of NFC packet. It means, the fragmentation process must be taken. Afterwards, NFC session is set up between devices (nodes) and the packets in sequence are sent to the receiver. Sequence of steps is shown in Figure 1.

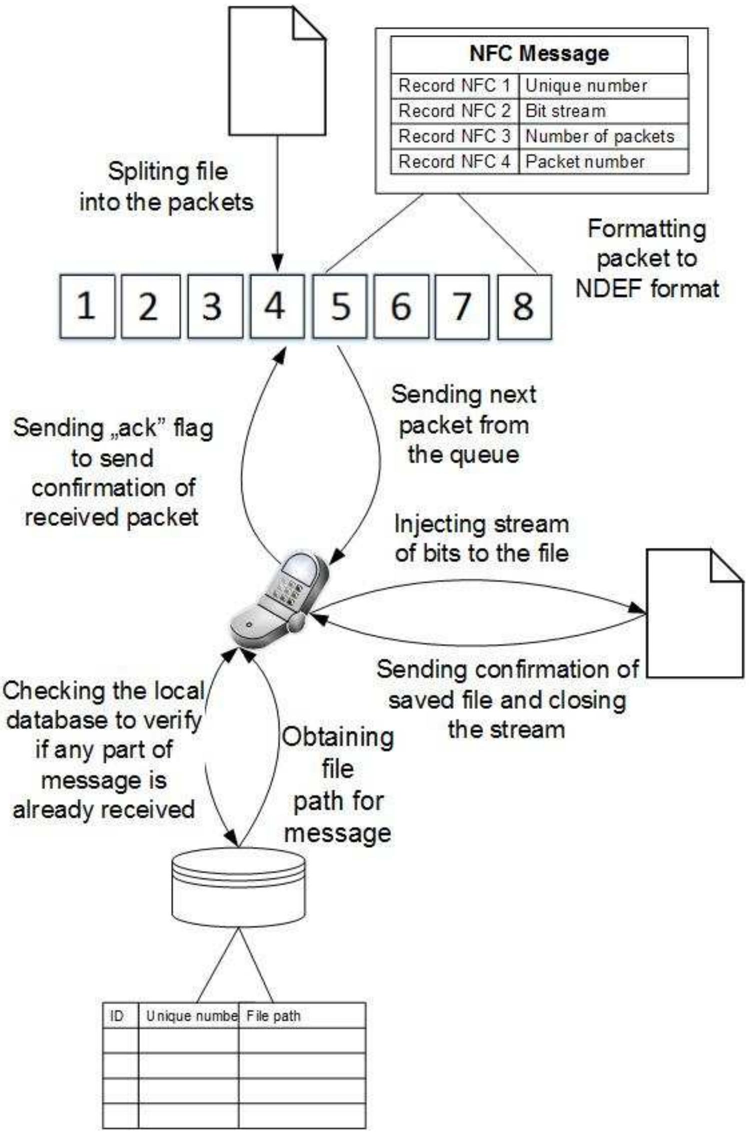
NFC Message
Record NFC 1
Unique number
Record NFC 2
Bit stream
Record NFC 3
Number of packets
Record NFC 4
Packet number
Spliting file into the packets
Formatting packet to NDEF format
1
2
3
4
5
6
7
8
Sending „ack" flag to send confirmation of received packet
Sending next packet from the queue
Injecting stream of bits to the file
Sending confirmation of saved file and closing the stream
Checking the local database to verify if any part of message is already received
Obtaining file path for message
ID
Unique numbe
File path

Figure 1: Sending process

Table 1: The detailed description of each record

Item	*Description*
Unique number	Unique number for the message
Bitstream	Payload
Number of packets	Number of parts generated by splitting the message
Packet number	Succeeding number of part

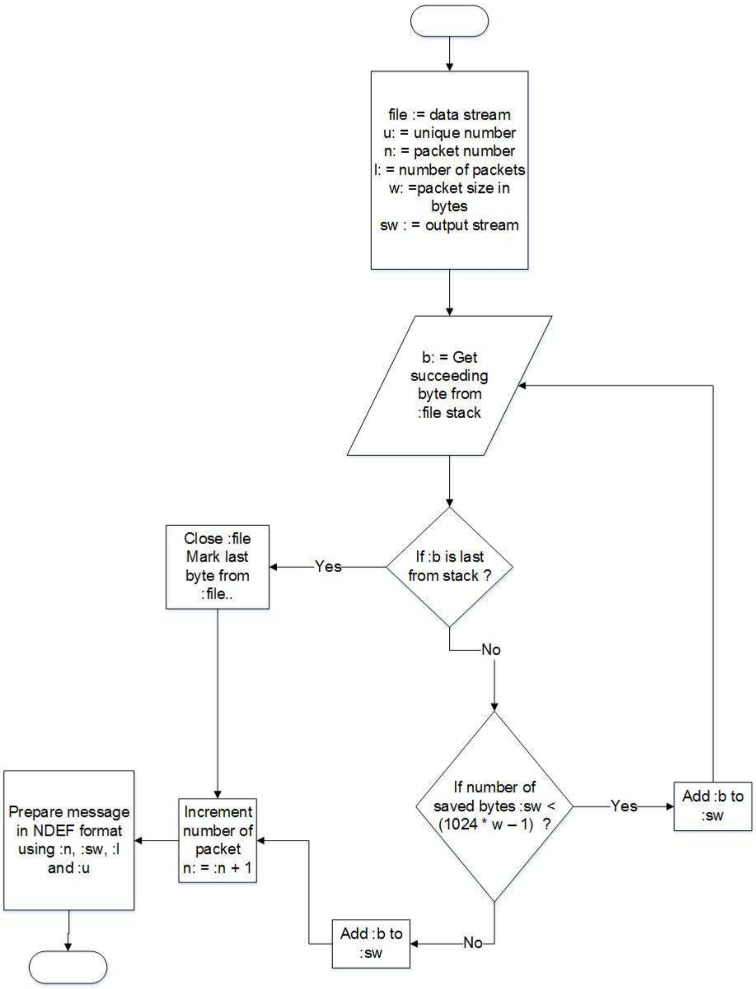
file := data stream
u: = unique number
n: = packet number
l: = number of packets
w: =packet size in bytes
sw : = output stream
b: = Get succeeding byte from :file stack
If :b is last from stack ?
Yes
Close :file Mark last byte from :file..
No
If number of saved bytes :sw < (1024 * w – 1) ?
Yes
Add :b to :sw
No
Add :b to :sw
Increment number of packet n: = :n + 1
Prepare message in NDEF format using :n, :sw, :l and :u

Figure 2: Block diagram of the algorithm responsible for sending the NFC message

In the example the message is divided into 8 fragments. Each fragment is built from four NFC records, which are saved in NFC data exchange format (NDEF). Detailed description of each record is presented in Table 1.

The receiver analyzes each record of the received packet. The unique number is used to check in the local database if exists any received data of the message. The payload is appended to this data in case of existance another data of the message. After this operation the receiver sends the confirmation flag to inform the sender that the packet has been received and processed succesfully. The sender takes the next packet from the stack and sends it to the receiver. This process is repeated until the stack is empty.

3.2 Data sending

Besides specifying a unique part number and payload, the algorithm determines packet number and calculate the total number of packets of the message. The number of packets is calculated by dividing the size of the message by the size of the single packet.

The message is passed to the algorithm responsible for sending data as an input. Additional parameters are: an unique number for transmission which was generated in the previous step, the number of packets to send, the number of the current being sent packet, and the defined packet size. The message is mapped to the byte stream described as the variable file. After reading each byte, algorithm determines if the examined byte is the last in the stream and/or whether the output buffer is not full. The algorithm stops injecting data into the input buffer if: (a) the amount of stored data reaches the maximum size of the packet, or (b) all data has been read from the file buffer. The packet is send after the completion of this process. Algorithm of sending NFC message is presented in Figure 2.

3.3 Data receiving

Algorithm describing the behavior of the system during the process of packets receiving is presented in Figure 3. In first step the algorithm analyzes received records. It is verified whether the payload is not empty and checks the database to determine whether another packets of the same message were not transmitted. Data is appended to the stream in case, when another part of the message is found. If the application determines that the received packet is the last one, it is marked in the database. The algorithm decides whether the packet is the final one by comparing the parameters: the number of the package and the total number of packets.

If the number of packet is equal to the number of message packets, the algorithm assumes that the message was sent. Parts of the message are sent sequentially. The sending node will retransmit the packet, if a confirming flag is not sent by the receiver.

3.4 The automatic NFC multi-packet transfer

The basic mechanism to NFC data communication is Android Beam. As a standard procedure before sending a message to another device, Android Beam enforces to tap mobile device screen. This procedure prevents device from unwanted automatic data send. Automatic data send is mandatory for multi packet data transfer, therefore we propose the method to handle this problem. Automatic tap functionality was implemented using Xposed framework. The framework enables dynamic simulation of the tap action. Xposed modules allow to perform actions immediately before and after triggering a captured method. In order to capture the system methods we re-implemented methods: handleLoadPackage and findAndHookMethod. This methods are called with parameters: the package name, the name of the hooked method and object of an XCMethodHook abstract class. The object have the appropriate method responsible for modifying the behavior of the hooked method.

3.5 Delay tolerant networks support

The NFC multi-packet communication system was proposed as a part of DTN system for a smartphone. The full functionality of DTN device consists of a group of functions responsible for communication and storage. In terms of architecture, this set of functionalities was implemented in an integrated way.

The first component includes a group of functions responsible for autonomous sending and receiving NFC multi-packet messages. The functionality of communications in our implementation consists of the use of communication NFC interfaces. DTN paradigm store-carry-forward is implemented in classical way based on simple Epidemic Routing idea. This means that all messages are replicated and send to every connected device.

4 Results

The goal of our tests was to compare the effectiveness of the NFC multi-packet data transfer process for approaches with manual and automatic handling. Manual handling requires Tap To Send (TTS), the automatic handling is performed without TTS.

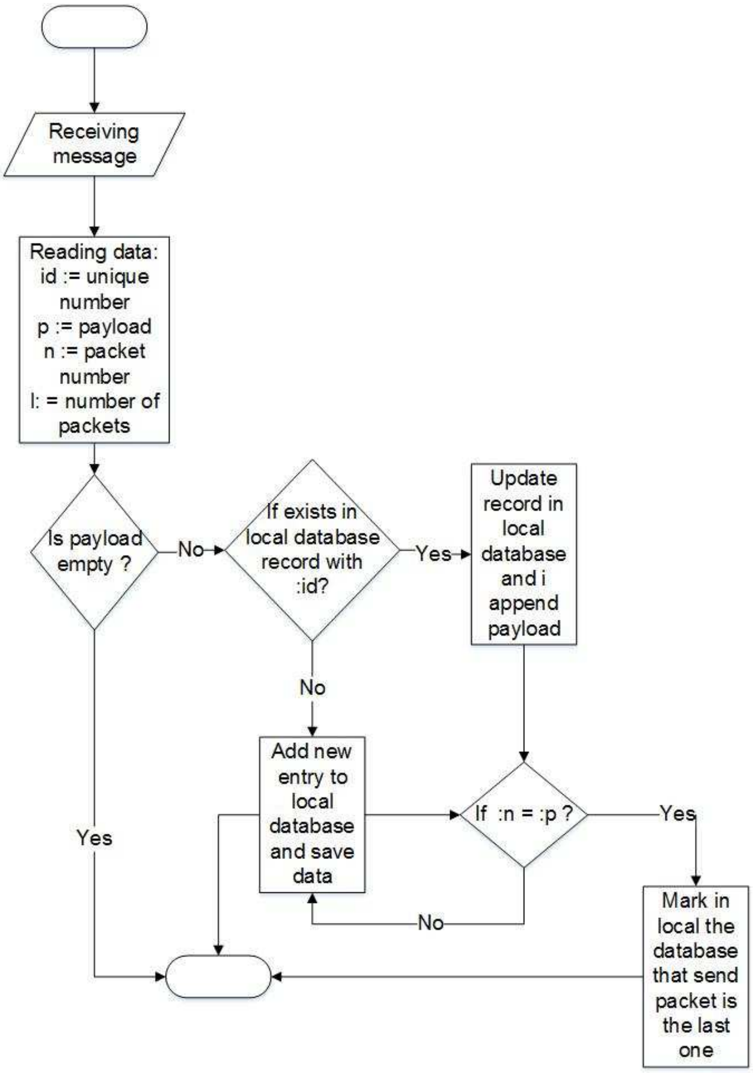
Receiving message
Reading data:
id := unique number
p := payload
n := packet number
l: = number of packets
Is payload empty ?
No
If exists in local database record with :id?
Yes
Update record in local database and i append payload
No
Add new entry to local database and save data
If :n = :p ?
Yes
No
Yes
Mark in local the database that send packet is the last one

Figure 3: Block diagram of the algorithm responsible for receiving the NFC message

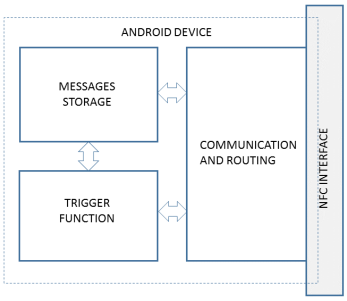

Figure 4: Modular architecture for DTN via NFC using Android device

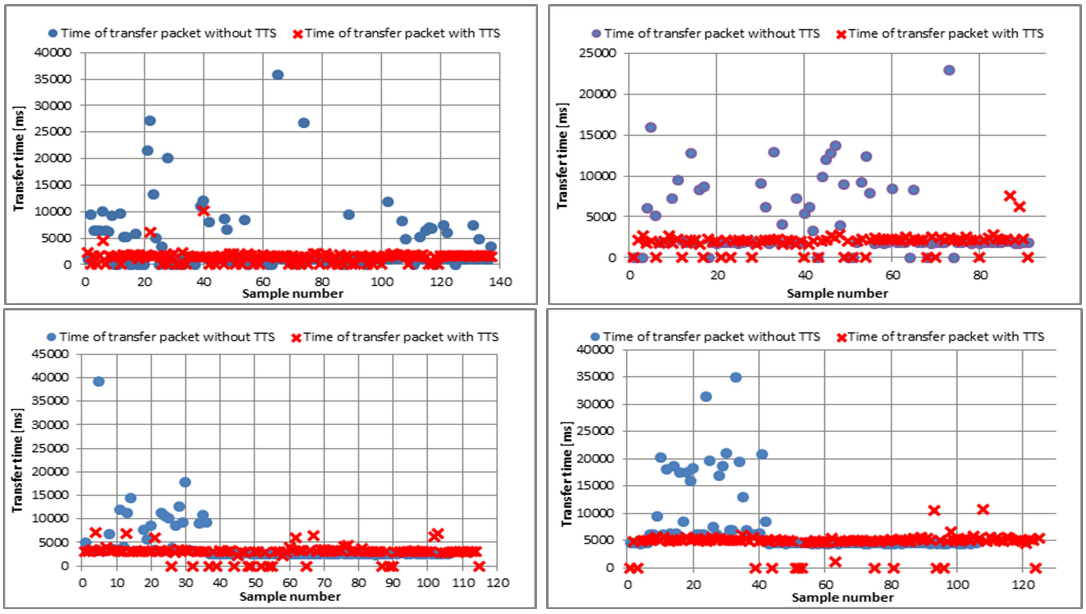

Figure 5: The total mean transfer time for automatic and manual transfer for packets: a) 2KB, b) 4KB, c) 8KB and d) 16K

For tests we proposed packets with sizes: 2 KB, 4KB, 8 KB, 16 KB, 32 KB and 64 KB. For each message more than 100 measurements were done. For each measurement the transfer execution time and the total mean throughput was reported. The total transfer time was defined as a sum of: (a) preparation of the NFC message, (b) user interaction (during tests with TTS), (c) data transfer. For TTS we manually perform screen tapping as soon as it is possible.

Tests were done on Samsung Galaxy S3 and LG L65 mobile phones. Both devices have got built-in NFC module. Samsung Galaxy S3 was working on Android 4.0 operation system, LG L65 was working on 4.4 Android operation system. Tests were done in 12 square meters room.

At first we perform measurements of packet transfer time (see Figure 5). The time increases with the size of the packet as expected for TTS and without TTS. Some transfer attempts were failed during tests. Sessions were interrupted by the Logical Link Control Protocol (LLCP). LLCP is a second layer protocol responsible for setting up session between two NFC enabled devices in peer-to-peer communication. LLCP was returning errors during setting up session (marked in Figure 5 as 0 ms transfer time sample) or setting up session longer than expected. The second circumstance had big influence on the transfer time, which was getting longer.

The effectiveness defined as transmission success ratio of both solutions was presented in Figure 6. The effectiveness is increasing with the size of the packet.

Success ratio was higher than 80 % almost in each test which we have done. In most cases approach without TTS was more efficient. Furth more there was 100% correctly sent samples for 8 KB and 16 KB packets.

Although, the efficacy of transferring data without the TTS was higher, there were more samples with extreme values. Transferring time reached tens of seconds.

Xposed is not officially supported by Google. This framework can be unstable. Release of the next Android version cause that Xposed authors has to adopted framework to the new version. We suspect that Xposed instability has got influence on measurements without TTS tests.

The variation and standard deviation were much larger for transmission with bypassed TTS. Because of that we decided to focus on median value to calculate throughput for tests. Figure 7 presents the median time of transferring packets. Results presents the better transfer time for

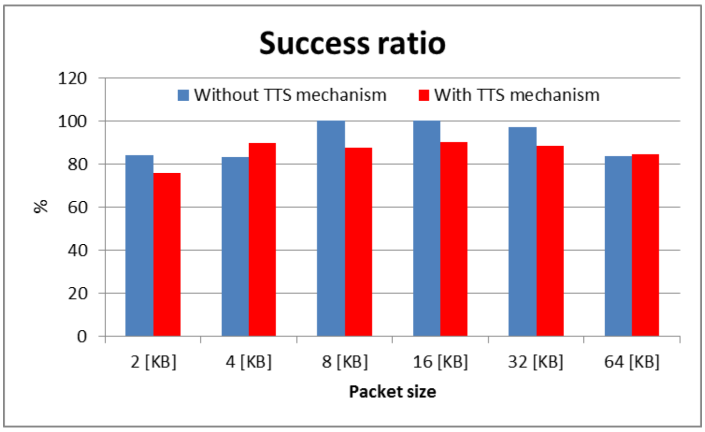

Figure 6: The success ratio of multi-packet transfer

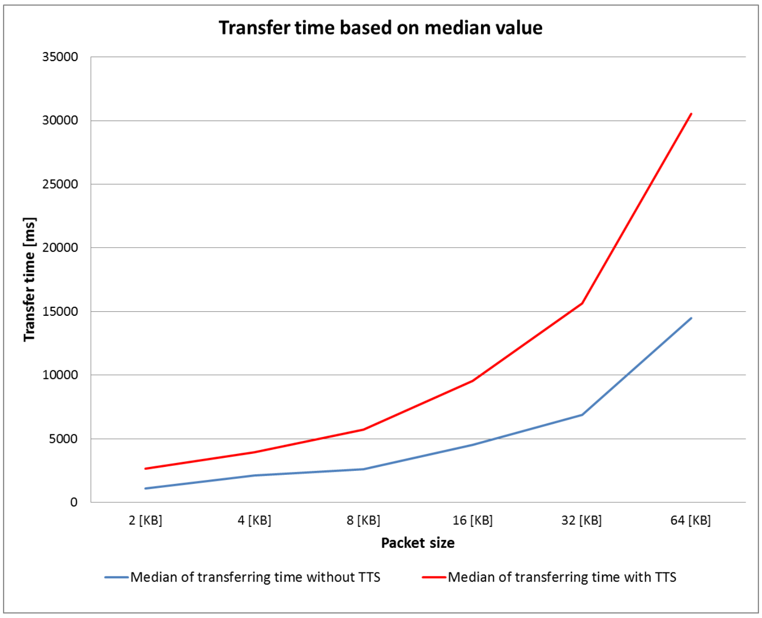

Figure 7: The median of mean transfer time

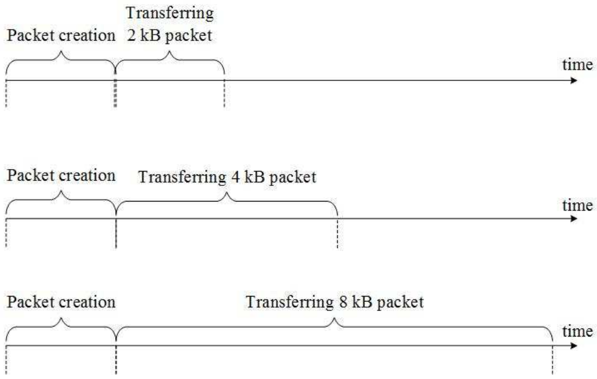

Figure 8: Analysis of time needed to perform the operation while transferring the file

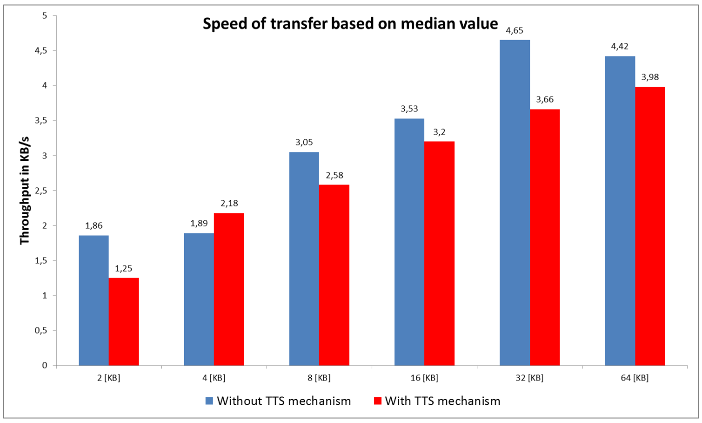

Figure 9: Throughput in function packet size

automatic solution - without TTS.

The second parameter measured was throughput - the total speed of data transfer. We determine the throughput as a message size divided by total transfer time. The total transfer time is the sum of preparation message time, user interaction time and data transfer time. Data transfer time increases with the size of the packet. Creation time is constant for all samples in the packet size group and has huge impact on the results. The throughput increases with the size of the packet. Packet creation time is more negligible with larger packet size. This phenomenon is illustrated on Figure 8. Moreover the automatic mechanism is more effective through the TTS mechanism elimination. Results was presented in Figure 9.

5 Conclusions

In this paper we have presented the idea of NFC multi packet communication for DTN networks. We decided to create this work due to the lack of data about NFC usage in DTN networks especially for larger message transmission which requires fragmentation. We described in detail algorithms for sending, receiving and automatic handling of fragmented message. The algorithms was implemented on Android device and extensively tested. We examined our automatic approach comparing with human-involved standard Android Tap-To-Send method. All test was conducted with different packet sizes. Obtained results confirm the good performance of our solution. The automatic approach has lower total transfer time and higher throughput then manual one.

Bibliography

[1] Burleigh S., Hooke A., Torgerson L., Fall K., Cerf V., Durst B., Scott K., Weiss H. (2003); Delay-tolerant networking: an approach to interplanetary internet. *IEEE Commun. Mag.*, 128-136, 2003. doi:10.1109/MCOM.2003.1204759

[2] Curran K., Millar A., Mc Garvey C. (2012); Near field communication, *International Journal of Electrical and Computer Engineering*, 2(3), 2012.

[3] Domaszewicz J., Koziuk M., Schoeneich R.O. (2008); Context-Addressable Messaging Service with Ontology-Driven Addresses, *in Proc. On the Move to Meaningful Internet Systems: OTM 2008*, 1471-1481, 2008. doi:10.1007/978-3-540-88873-4_37

[4] Dziekonski A.M., Schoeneich R.O. (2016); DTN Routing Algorithm for Networks with Nodes Social Behavior, *in Proc. International Journal of Computers Communications and Control*, 11(4), 457-471, 2016. doi:10.15837/ijccc.2016.4.1454

[5] Golanski M., Schoeneich R.O., Siwko M. (2010); The algorithm for distribution of large-size data in the Wireless Ad-hoc Sensor Network, *in proc. Concepts and Implementation for Innovative Military Communications and Information Technologies, Military University of Technology*, 577-584, 2010.

[6] Ha P., Yamamoto H., Yamazaki K. (2013); Using Autonomous Air Vehicle in DTN Sensor Network for Environmental Observation, *in Proc. Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual. IEEE*, 447-450, 2013. Doi:10.1109/COMPSAC.2013.74

[7] Fall K. (2003); A delay-tolerant network architecture for chalenged internets, *In Proc. of The 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03)*, 27-34, 2003. doi:10.1145/863955.863960

[8] Fan C.-S. (2016); HIGH: A Hexagon-based Intelligent Grouping Approach in Wireless Sensor Networks, *Advances in Electrical and Computer Engineering*, 16(1), 41-46, 2016. doi:10.4316/AECE.2016.01006

[9] Fang W., Li S., Liang X., Li Z. (2012); Cluster-based Data Gathering in Long-Strip Wireless Sensor Networks, *Advances in Electrical and Computer Engineering*, 12(1), 3-8, 2012. doi:10.4316/AECE.2012.01001

[10] Juang P., Oki H., Wang Y., Martonosi M., Peh L.S., Rubenstein D. (2002); Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, *SIGOPS Oper. Syst. Rev.*, 36(5) (October 2002), 96-107. doi:10.1145/635508.605408

[11] Kawecki M., Schoeneich R.O. (2016); Mobility-based routing algorithm in delay tolerant networks, *in Proc. EURASIP Journal on Wireless Communications and Networking*, 81, 1-9, 2016. doi:10.1007/978-3-642-32518-2_23

[12] Koziuk M., Domaszewicz J., Schoeneich R.O., Jablonowski M., Boetzel P. (2008); Mobile Context-Addressable Messaging with DL-Lite Domain Model, *in Proc. Smart Sensing and Context, 5279*, 168-181, 2008. doi 10.1007/978-3-540-88793-5_13

[13] Lindgren A., Doria A., Schelen O. (2003); Probabilistic routing in intermittently connected networks, *ACM SIGMOBILE Mobile Computing and Communications Review*, 19-20, 2003. doi:10.1145/961268.961272

[14] Lu Z., Fan J. (2010); Delay/disruption tolerant network and its application in military communications, *Proc. of computer design and applications*, 2010. doi: 10.1109/ICCDA.2010.5541302

[15] Matsuzaki R., Ebara H., Muranaka N. (2015); Rescue Support System with DTN for Earthquake Disasters, *IEICE Trans. Commun.*, 98(9), 1832-1847, 2015, doi:10.1587/transcom.E98.B.1832

[16] Ondrus J., Pigneur Y. (2007); An Assessment of NFC for Future Mobile Payment Systems, *Proceedings of the International Conference on the Management of Mobile Business (ICMB '07). IEEE Computer Society*, 2007, doi:10.1109/ICMB.2007.9

[17] Quwaider M., Biswas S. (2010); DTN routing in body sensor networks with dynamic postural partitioning, *Ad Hoc Networks*, 8(8), 824-841, 2010, http://dx.doi.org/10.1016/j.adhoc.2010.03.002

[18] Palka P., Schoeneich R.O. (2013); Multi-commodity Trade Application to the Routing Algorithm for the Delay and Disruptive Tolerant Networks, *in Proc. New Trends in Databases and Information Systems*, 185, 241-250, 2013. doi:10.1007/978-3-642-32518-2_23

[19] Schoeneich R.O., Domaszewicz J., Koziuk M. (2009); Concept Based Routing in Ad-Hoc Networks, *Lecture Notes In Computer Science*, 5408, 43-48, 2009. doi:10.1007/978-3-540-92295-7_8

[20] Schoeneich R.O., Golanski M. (2007); Mesh Cluster Based Routing Protocol: Enhancing Multi-hop Internet Access using Cluster paradigm, *Proc. EUROCON, 2007. The International Conference on Computer as a Tool*, 962-965, 2007. doi:10.1109/EURCON.2007.4400318

[21] Spyropoulos T., Psounis K., Raghavendra C.S. (2005); Spray and wait: an efficient routing scheme for intermittently connected mobile networks, *Proc. of The 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 252-259, 2005. doi:10.1145/1080139.1080143

[22] Vladuta A.-V., Pura M.L., Bica I. (2016); MAC Protocol for Data Gathering in Wireless Sensor Networks with the Aid of Unmanned Aerial Vehicles, *Advances in Electrical and Computer Engineering*, 16(2), 51-56, 2016. doi:10.4316/AECE.2016.02007

[23] Want R. (2006); An introduction to RFID technology, *IEEE Pervasive Computing* 5(1), 25-33, 2006. doi:10.1109/MPRV.2006.2

[24] Want R. (2011); Near field communication, *IEEE Pervasive Computing*, 3(10), 4-7, 2011. doi:10.1109/MPRV.2011.55

[25] Zimmerman T.G. (1996); Personal area networks: near-field intrabody communication, *IBM Syst. J.*, 35(3-4), 609-617, 1996. doi:10.1147/sj.353.0609