Direction of Wireless Access Convergence; Can Near Field Communication (NFC) Be a Member of Future Internet?

A. Arutaki[*], D. Cavendish[*], H. Sakai[**], A. Matsumoto[***] and Y. Oie[*]

[*]Network Design Research Center, Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan, {aarutaki, cavendish, oie}@ndrc.kyutech.ac.jp
[**]Dept. of CS and Systems Eng., Kyushu Institute of technology, sakaih@infonet.cse.kyutech.ac.jp
[***]NCOS Laboratories, NEC Communication Systems, Ltd., matsumoto.kr@ncos.nec.co.jp

ABSTRACT

This paper gives bird's-eye view of a variety of wireless access systems with respect to communication protocol structure. As these technologies evolve to higher data-transmission rates, there is more affinity with the Internet protocols. This observation motivates discussions of wireless access convergence in the future. For instance, communication protocols for IC-cards and RFIDs, which are quite different from TCP/IP, may get affinity with Internet protocols, if they gain higher data throughput in the future owing to emerging radio technologies. The authors have picked NFC up as an example to enhance its standardized protocol to let it be more intimate with the Internet. The cross-layer approach is taken to enhance today's standardized NFC protocol, while keeping backward compatibility. The enhanced NFC protocols are proposed and evaluated in terms of performance and complexity. As the result, the authors show the possibility of future convergence in the wireless access arena.

Keywords: Wireless Access, TCP/IP, Near Field Communication (NFC), Protocol, Cross-layer

1 INTRODUCTION

Mobile communication has become one of the most popular and important infrastructures for our life. We cannot live without cellular phones. It is needless to say that the service is supported by constant innovation of wireless access technologies, such as 3GPP/LTE. Data communications, including picture and video, especially benefit from these technologies, and as a natural result, traffic volume of the services has been growing drastically. At the same time, WiFi plays an important role in wireless access for PCs/PDAs and smart-phones. The recent growth of hot-spot deployment has provided another option of wireless communication infrastructure.

On the other hand, alternative wireless access methods have been widely used; RFID tags and contactless IC-cards are typical ones. Both of them were designed for specific services, such as factory automation and electric ticket services, respectively. These specific services do not require sophisticated platforms. Rather, simple and inexpensive devices are preferable, and thus simple protocols are used for their wireless communication.

Although 3GPP/LTE and WiFi make use of Internet protocols in their air interfaces, RFIDs and contactless IC-cards do not. This is because their specific services are isolated from the open Internet. Another reason is that Internet protocols are fairly heavy for their inexpensive devices.

However, recent cellular phones and smart-phones function as contactless IC-cards whose air interface is Near Field Communication (NFC). This trend motivates us to consider future possibility of air-protocol convergence. Namely, NFC may adopt part of Internet protocols, so as to become part of the larger Internet infrastructure.

This paper casts a forward look into NFC as one more integrated communication technology of future Internet.

2 WIRELESS ACCESS VARIETIES

As described in the former section, we are surrounded by a variety of wireless access systems in our busy lives, as illustrated in Figure 1, focusing on radio transmission distance and data rate per channel.

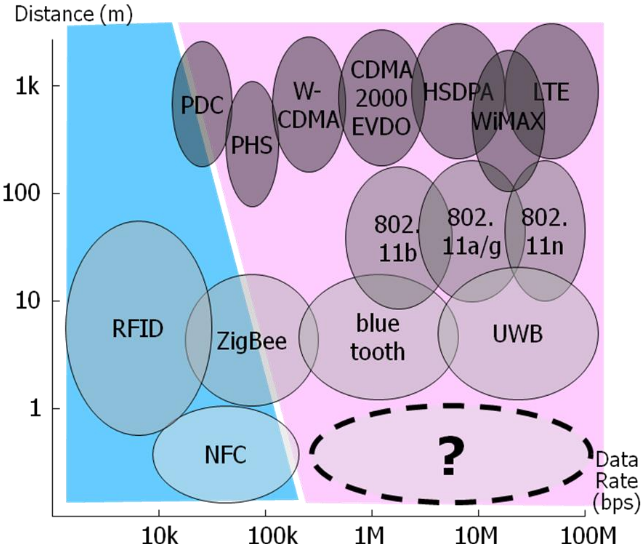


Figure 1: Wireless Access Systems

The wireless access systems plotted in the right upper half of Figure 1 are able to support Internet protocols or similar ones over radio links. On the other hand, the left lower region indicates wireless systems which adopt proprietary protocols. We foresee that, as they evolve to support higher data rates, these systems may also support Internet protocols, becoming part of the plethora of communication technologies embraced by the Internet.

2.1 Order of "km" Range

Distance range of the order of kilometers is covered by so-called Cell-phone technologies. As is well-known, cellular phones are widely spread to support our daily life. According to a recent survey by Japanese Government[1], there are more than 120 million subscribers in Japan at the end of 2011 fiscal year. This figure implies that everyone in Japan, including babies and elderly people, owns a cellular phone. Although some proprietary air protocols were used at the beginning of cellular data services, such as WAP1.0[2] for simpler data transmission, today's air protocols are very intimate with TCP/IP. This fact has led cellular and smart phones to play a central role in connecting to the Internet.

2.2 Order of "100m" Range

WiFi is today a ubiquitous access method covering distance range of 100 of meters. It is hard to find a personal computer without WiFi radios store's shelves. In addition, cellular phones with WiFi radio interface have showed up early in this century for in-house VOIP communications, and are widely spread, as cellular/smart phones become the device of choice to Internet access. WiFi access is very similar to the wired broadband ones with respect to Layer 3 and upper layers, as the technology was developed from the beginning as to provide high speed wireless access to the Internet. Hence, WiFi supports Internet access strongly, and will do so in the future, as well.

2.3 Order of "10m" Range

Distance range of tens of meters is typically covered by wireless sensor communication. ZigBee, Bluetooth, and UWB are supposed to function under battery operation, and are ready to support TCP/IP stack, whether they do so or not. In contrast, RFID is driven by inductive power supply. This fact restricts its functionality, and as a natural result, it's simplicity does not allow support of Internet protocols. In the future, depending on RFID development path, it may also become a member of the Internet, in which case considerations similar to NFCs described below may be helpful.

2.4 Order of "1m and less" Range

Contactless IC Cards (PICC; Proximity Inductive Coupling Card) are part of our social infrastructure in many countries around the world. They adopt NFC with inductive power supply so that they can communicate with card readers/writers (PCD; Proximity Coupling Device) within meter distance range. According to a recent survey in Japan[3], more than 80 million PICCs are shipped every year. This figure means that a Japanese, even a baby, gets an PICC every 18 months.

Quick overview of NFC is as follows; the technical specifications are standardized by ISO and other standard bodies[4, 5, 6], and its wireless distance varies from 2mm to 70cm depending on the application. The radio frequency is in the HF band, typically 13.56MHz, and the data transmission rate is in the order of 100kbps and less. An example of NFC packet format[7] is depicted in Figure 2. PICC and PCD exchange this type of packets in a handshake fashion.

Preamble 6 Bytes	SYNC 2 Bytes	Length 1 Byte	Payload Max. 256 Bytes	CRC 2 Bytes

SYNC; Synchronous Pattern, CRC; Cyclic Redundancy Check

Figure 2: Example of NFC Packet Format

The "Payload" information in Figure 2 contains communication identifiers (IDs) such as NFC-ID and/or application commands such as "Read" or "Write" depending on the attributes of the packet. In other words, current NFC protocols aim for simple implementation, not for hierarchical and sophisticated one.

One of the most reasonable purposes for this simple format is to reduce the cost of PICCs. As is described in Table 1, very elementary platform is provided for a PICC, and can be accounted for its wide used.

Table 1: Example of IC-card Specification [8]

CPU	8 bit RISC
Memory size	4,096 bytes
User memory area	2,464 bytes
Communication speed	212 kbps

Although NFC design follows a "Simple is the best" strategy, recent trends motivate the authors to investigate issues such as: Is NFC to stay as they are, or will it support Internet protocols? The fact that recent cellular and smart phones are being furnished with NFC, a similar path to WiFi radio technology, may spell Internet support for this technology in the not so far future. It is expected that NFC-equipped cellular phones will occupy more than 85% out of all the deployed ones in 2015 in the world[9].

Figure 3 shows a system model example of NFC application. In this model, both electronic ticket and NFC-ready cellular phone access the PCD (i.e., electronic ticket gate at a railway station) via NFC.

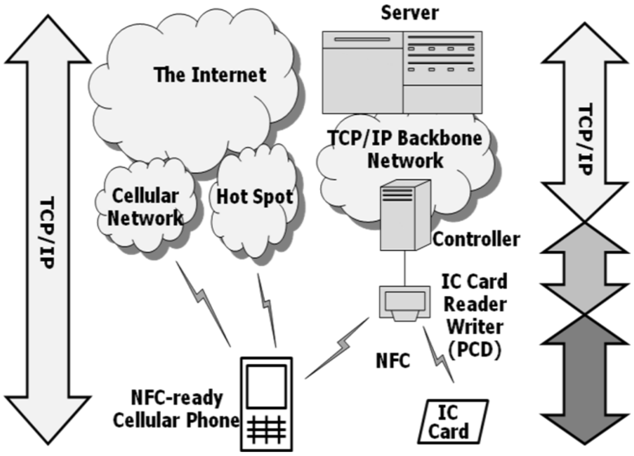


Figure 3: System Model of Electronic Ticket/Prepaid Service

As depicted in Figure 3, the PCD communicates with the IC-card and cellular phone through NFC. On the other hand, the cellular phone has the capability to communicate with the Internet directly via WiFi or public mobile networks. As it is expected that NFC-ready cellular phone becomes dominant, it seems reasonable to expect that Internet protocols, such as TCP, may run over NFC radio links.

However, as with any widely deployed communication technology, evolution of NFC technology must pay attention to backward compatibility. Namely, the PCDs must accept any legacy PICCs with original NFC protocols, in addition to supporting powerful next generation PICCs, which may require more sophisticated communication, being that TCP/IP or other. We address this issue in the next session.

3 CROSS-LAYER APPROACH FOR NFC ENHANCEMENT

This section focuses on issues arising on NFC enhancements. As described in the previous section, NFC may be enhanced to have more affinity with the Internet. However, NFC platforms have less power available than ordinary air edge devices, i.e., cellular and smart phones. Therefore, NFC needs to be carefully examined vis-à-vis Internet protocol support directly into NFC from a power consumption’s perspective.

Another point of consideration is communication delay and session duration. Typical packet level delay, i.e. latency, of the Internet ranges from a few milliseconds to a few hundred milliseconds[10]. Typical round trip delay via public mobile communication systems reaches more than some hundred milliseconds[11] even using mobile-oriented protocols such as Mobile IP or Wireless-profile TCP. Furthermore, at the beginning of communication, there is extra delay incurred by TCP connection set up. This large delay may have serious negative impact on typical NFC applications, at which communication has a very short duration. A cross-layer approach would be to adopt parts of the Internet protocols to NFC, but not all functionalities.

3.1 Functionalities

A. Security and authentication

A future IC-card, as a part of social infrastructure, will be required to have more robust security and authentication. Recent press release[12] makes a case for the need of strong encryption technologies within an IC-card in the near future. Even the stronger encryption is implemented in NFC, it has limitation[13] compared to the Internet. More internet-oriented security technologies may show up in the future.

B. Multi-service interaction

Today's IC-card services are mostly isolated, providing silo application solutions in a fashion of "one card, one application." Multiple services may be performed in a single IC-card, but even in this case, they are supported by server-interaction, not by application collaboration in the card. With further support of multiple application, NFC link performance will need to be increased in the future to support increasing communication between the card and multiple servers sharing a single NFC link.

Moreover, recent penetration of NFC-ready cellular/smart phones may drive collaborative services among their platforms and NFC applications. If this is realized, for example, users may consume more information via NFC links from PCDs, requiring more powerful next generation NFC.

3.2 Performance and Quality

NFC today adopts a conventional handshake protocol with limited performance, as is shown in a later section. This is because today's applications require small data exchanges between a PICC and a PCD. But it is clearly observed that more communication performance with better quality is expected for NFC to download more data and application programs in the future.

A large number of research works has been conducted to improve communication technologies’ performance such as TCP/IP. In particular, improvements in retransmission schemes to handle packet corruption focus on "packet loss" characteristics[14]. Because today's broadband communication systems, such as the Internet, adopt the layered architecture in which case the retransmission function is implemented in Layer 4, i.e., TCP; it can recognize "packet loss", but not "bit error" directly. Hence, past retransmission approaches cannot be directly applied to NFC enhancement for the following reasons.

The first reason is backward compatibility. As is mentioned above, NFC has already been standardized and is widely used. This technology is quite different from TCP.

Secondly, we should not completely abandon NFC design principle of “the simple is the best.” NFC utilizes bit error detection via CRC-16, not on a packet basis. We should better take advantage of this "physical layer" power to improve the upper layer communication performance and quality, in a sound Cross-layer approach[15]. To our best knowledge, there have not been past works which handle bit error characteristics and higher layer retransmission performance including contents level verification.

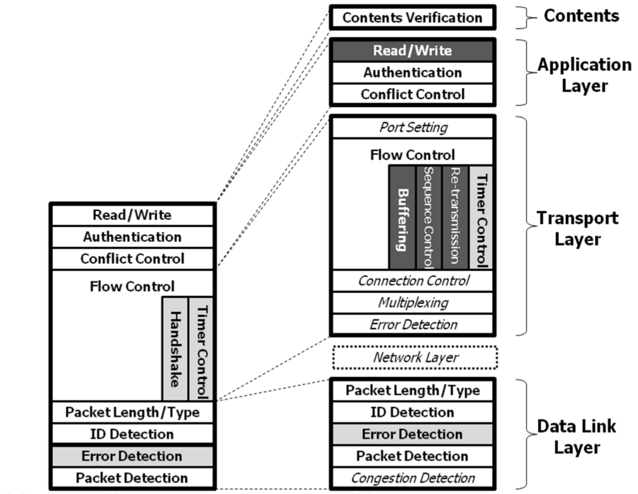
Another reason to adopt a cross-layer approach is time duration difference between NFC and Internet protocols. As is mentioned at the beginning of this section, it is known that a typical TCP session lasts at least one order of magnitude more than typical NFC applications: 10s to 100s milliseconds round trip time. One of the most popular applications of contactless IC-cards in Japan, SUICA[16], allows only 0.2 seconds to complete information exchange via NFC. Hence, a complete NFC data exchange does not last long enough to match a single TCP round trip time.

In what follows, the authors introduce a cross-layer approach to improve the NFC performance while taking into account NFC current characteristics and maintaining backward compatibility with legacy NFC interfaces.

3.3 Function Allocation

As discussed so far, NFC has been used in a variety of applications. Some standards have already fixed NFC protocols which are widely used. As described in NFC packet format illustrated in Figure 2, NFC protocols do not follow a hierarchical and sophisticated concept.

Figure 4 depicts NFC protocol function structure vis-à-vis typical Internet communication.



(a) Near Field Communication **(b) The Internet**

Figure 4: Protocol Function Allocation

Figure 4(a) depicts the protocol function structure of NFC. As PCD and PICC involved in NFC can easily identify each other because of no others' involvement in the communication link, Network Layer is shrunk; no IP (Internet Protocol) exists. Transmission errors can be only detected by CRC-16 appended at the end of each packet. Integrity of packet transmission in NFC is achieved by the combination of error detecting function and ACK (Acknowledgement) /NACK (Negative Acknowledgment) handshaking.

Figure 4(b) shows a typical Internet protocol stack. This is far more sophisticated than NFC's one. However, if NFC borrows some functionalities from TCP, its performance and reliability of data transmission can be improved. From this viewpoint, the authors pick some functionality up from the Internet protocol suites, namely: Buffering, Sequence Control, Re-transmission, and Read/Write; they are emphasized by the dark boxes in Fig. 4(b). More specifically, using a cross-layer strategy, the authors propose the following enhancements to NFC protocol performance.

A. *Introduction of Window*

As is explained earlier, NFC standards adopt a simple "handshake" protocol. If the protocol borrows the TCP nature of "flow control by window", data transmission performance can be improved by eliminating redundant handshaking. Some readers may wonder whether this feature introduces TCP complexity into NFC protocols. The authors show that the complexity does not increase by this "flow control by window", if the window size is fixed. The error detection by the original CRC-16 of NFC can substitute the TCP's error detecting function with less complexity.

B. *Introduction of Sequence Control*

The next step to improve the protocol performance is to introduce packet sequence numbering, which resembles TCP specifications, but not the sliding window mechanism. We believe that a fixed size window with sequence number control can improve NFC performance with small additional complexity, while maintaining backward compatibility.

4 PROTOCOL MODEL AND PERFORMANCE ANALYSIS OF NFC

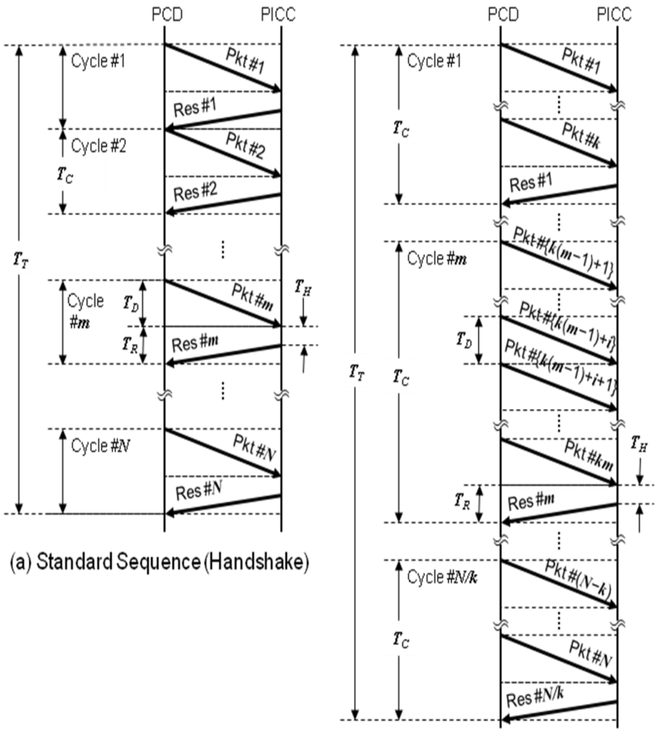
As explained earlier, NFC protocol adopts a handshaking mechanism for data transmission. The PICC receiver sends ACK/NACK back to the sender, when the CRC-16 appended in the received packet is correct/incorrect, respectively. Retransmission of data is conducted when the sender recognized a NACK from the receiver. This mechanism is effective, when the sending data is not large in size, and when the communication link is not stable such as the case between an automatic ticket gate and an IC-card ticket carried by a walking passenger. But in case of download of large data, in which an IC-card can be located stably on a card reader/writer, more efficient methods of data transmission are anticipated. From this point of view, the authors introduce a continuous k packet transmission without ACK/NACK reception; Fixed Window Transmission Protocol (FWTP-k).

Figure 5(a) depicts the standard handshake protocol. When PICC receives the m-th data packet (Pkt#m) from PCD, PICC sends a response packet (Res#m) back to PCD. This process of data transmission repeats N times, until all the packets are conveyed to PICC.

Figure 5(b) depicts the protocol sequence of the proposed FWTP-k. N data packets are divided into N/k groups (Cycles), i.e., k packets in a Cycle, assuming that N is a multiple of k. In Cycle#m, the PCD sends k packets, Pkt# $\{k(m-1)+1\}$ through Pkt#km to PICC, without pausing to receive any ACK. Once PICC receives its k-th packet, it sends a response packet, i.e., Res#m, back to PCD.

This cyclic manner repeats N/k times so as to send all N packets. Note that the standard handshake protocol is a subset of this FWTP-k, namely k=1.

As the data packet length is L_D bits, kL_D bits of data packets are transmitted in Cycle#m. If PICC detects a single bit error or more in the kL_D bits, it sends a NACK back to the PCD. Then the PCD retransmits the Cycle again from the beginning, i.e., Pkt# $\{k(m-1)+1\}$ through Pkt#km.



(a) Standard Sequence (Handshake)

(b) Fixed Window Transmission Protocol (FWTP-k)

Figure 5: Protocol Sequences

Assuming that the random bit error rate is r and that each packet transmission is independent of the others, the expected value Ex(T_T) of the total transmission time T_T can be derived as follows. Paying attention to the independency of the error occurrence in each Cycle, Ex(T_T) is the sum of Ex(T_C), which is the expected value of the consuming time T_C of each Cycle. Assuming that the Cycle consumes time of iT_C with (i−1) retransmission, its probability is

$$\sum_{i=1}^{\infty} p^{(i-1)}(1-r)^{kL_D}\, i(kT_D+T_R) = \frac{(1-r)^{kL_D}(kT_D+T_R)}{p}\sum_{i=1}^{\infty} ip^i, \qquad \text{where } p = 1-(1-r)^{kL_D}.$$

where T_D is the transmission time of a packet, and T_R is the transmission time of an ACK/NACK packet.

Applying the binomial theorem[17]

$$\sum_{i=1}^{\infty} ix^i = x/(1-x)^2$$

to the above, we obtain the following equation,

$$\mathrm{Ex}(T_T) = \frac{N}{k(1-r)^{kL_D}}(kT_D+T_R). \tag{1}$$

This equation indicates the transmission performance characteristics of the FWTP-k. The notation in the equations above and throughout the paper is as follows;

r ; random bit error rate
L_D ; number of bits in a data packet
L_R ; number of bits in a response packet
N ; total number of data packets to download
k ; number of data packets in a Cycle
T_D ; time duration to send a data packet
T_R ; time duration to send a response packet
T_C ; time duration of internal processes in PICC
T_C ; time duration of a Cycle
T_T ; time duration to download all data packets
Ex(T_T) ; expected value of T_T

Also note that the authors assume "no bit errors" in a Res packet which is much shorter than a data packet.

5 SIMULATION AND NUMERICAL EVALUATION

For the purpose of the simulation, two entities, i.e., PCD and PICC are separately described in the program.

The data of length (L_D−2)bytes in PCD is randomly assigned, and 2 bytes of CRC-16 are calculated over this data and appended. This process of a single packet generation repeats k times.

When PICC receives a packet, it intentionally inserts random bit errors for the simulation purposes. Then PICC checks the packet CRC, and sends ACK or NACK according to the result. If an error is detected, "Corruption Counter" is incremented. This process continues k times, then PICC sends ACK/NACK back to the PCD depending on the value of "Corrupt."

When PCD receives NACK from PICC, the simulation process goes back to the starting point of the Cycle. Namely, PCD retransmits all the packets of the current Cycle. Otherwise, the simulation process goes on to the next Cycle.

According to the algorithm above, the total transmission-time duration is simulated and the results are depicted in Figure 6. The results of the theoretical calculation based on Equation(1) in the previous section are also depicted in the figure, and are exactly identical to the simulation.

Note that the constants and variables for both the simulation and theoretical calculation are shown in Table 2.

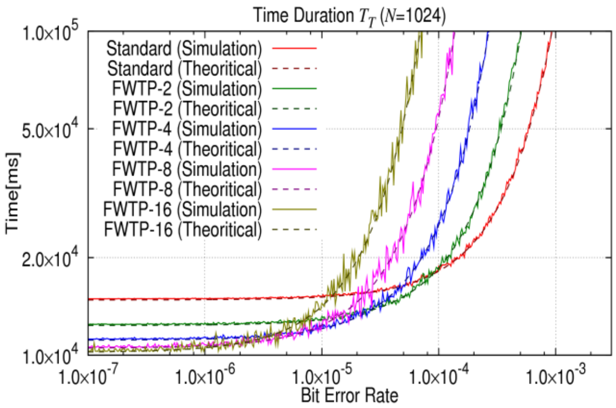


Figure 6: Expected Value of Time Duration

Table 2: Simulation and Theoretical Calculation Parameters

Bit Rate	212 kbps
L_D	2048 bits
T_D	9.660 ms
L_R	8 bits
T_R	0.038 ms
T_H	4.8 ms
r	1.0×10^{-7} to 3.0×10^{-3}
N	1024
k	1(Standard), 2, 4, 8, 16

The protocol performance is evaluated by observing the transmission time-durations; the less the duration, the better the performance (the higher throughput). It is clear that the transmission performance of FWTP-*k* is better than the standard handshake protocol, when the bit error rate is less than the order of 10^{-5} or 10^{-4}, depending on the value of *k*. This is because the retransmission redundancy increases when the bit error rate is high. To resolve this issue, retransmission mechanism can be modified to reduce the redundancy, and better performance is obtained[18].

Figure 7 shows the simulation results of distribution of the transmission-time durations. In this figure, the bit error rate is fixed at 1.0×10^{-5}, while the other parameters are identical to Table 3. The simulation is conducted 2000 times under the fixed parameter values, and the distribution is presented by the number of occurrences.

Figure 7 also indicates that the mean values of time durations of FWTP-*k* seem better than the standard one. It is interesting that, when *k* is large, the occurrence distribution has a large deviation. This also suggests that more stable protocol is expected for large *k* values[18].

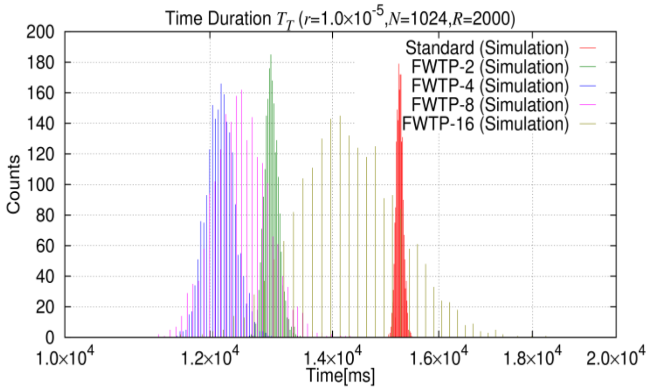


Figure 7: Distribution of Time Duration

6 COMPARISON OF COMPLEXITY AND PLATFORM

As mentioned earlier, NFC is designed for simple and inexpensive platforms such as shown in Table 1. Though it is clear that the enhanced protocols, i.e., FWTP- *k*, have better performance, it does not necessarily mean that they are better than the original one. This is because they may have more complexity to be implemented into today's contactless IC-cards. Furthermore, it is meaningful to compare complexity with the entire Internet protocol stacks which may have a chance to be buried into IC-cards in the far future. We select uIP[19] for this comparison candidate, which is widely used for embedded systems.

The authors assume that memory size required to implement the protocols fairly represents the complexity. Another index, for example, how much the protocol induces programming bugs, may have strong relation with complexity. However, especially in case of an IC-card, program and data-buffer sizes are very keen issues in terms of implementation into such a poor platform.

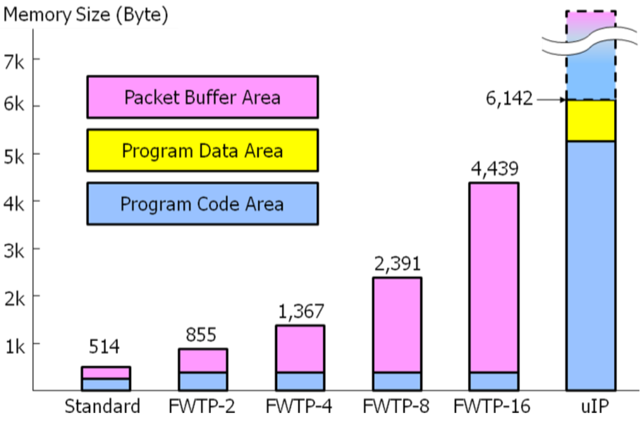


Figure 8: Memory Size of Protocols

Figure 8 illustrates typical memory sizes mentioned above. The “Program Code Area” colored in blue indicates the ROM size required to contain the protocol-operation program object. The “Program Data Area” in yellow and “Packet Buffer Area” in red show the RAM sizes for protocol operations and for buffering packets, respectively.

The bar named as “Standard” indicates the complexity of the original standardized NFC protocol of hand-shaking. Also, “FWPT- *k*” for the enhanced NFC protocols and “uIP” for uIP protocol. Note that the shown memory sizes in Figure 8 indicate required volumes just for communication, not for application operations, such as security and card-ID verification ones.

It is clear that the proposed protocol has similar complexity to the standard handshake protocol, because the consumed program-code-area requires less than 0.1k bytes increase. It is reasonable that the packet-buffer-area size to be proportional to the retransmission buffer size. On the other hand, uIP requires 10 times larger program-code-area, even though it uses a simple handshaking sequence without retransmission function. Note that the retransmission function and packet buffer memories need to be provided by the applications anyway, when using uIP.

The authors believe that the proposed NFC protocol enhancements will be suitable for future NFC. Especially for FeliCa[8] with a few kilo byte RAM, FWTP-2 will improve the performance with less implementing impacts. More wider window version may be taken advantage of for more larger platforms such as Type B[4] NFC platforms[20].

7 CONCLUSION

The authors provided bird’s-eye view of a variety of wireless access systems with respect to communication protocol structure. In addition, some NFC protocol improvement methods were proposed in a cross-layer design approach. It is shown that the proposed protocol enhancements are more efficient than the standard handshake sequence, and have more affinity with the Internet, which will contribute to wireless access convergence.

Finally, the authors understand that the memory consumption is just part of the protocol complexity evaluation. Other comparisons, such as processing time in PICC, will be studied in the future. Also in this paper, we ignore the residual error of CRC-16. According to past work[21], CRC-16 is not perfect under some conditions. As a matter of fact, the authors observed that residual errors of CRC-16 are not zero on conducting the simulation, if the bit error rate is high.

ACKNOWLEDGMENT

The authors express their appreciation to the members of Kyushu Institute of Technology and NEC Communication Systems Limited for fruitful discussions and valuable comments.

REFERENCES

[1] Information and Communications Policy Bureau, “WHITE PAPER; Information and Communications in Japan,” Ministry of Internal Affairs and Communications, Japan, August 2011, http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2011/chapter-4.pdf#page=5

[2] For example, K. Hendry, “Web engineering for mobile devices,” M.Sc Thesis, Dpt. of CS, Univ. of Helsinki, Finland, February 28, 2005, pp. 10-16.

[3] IT Media Enterprise, Japan, January 20, 2010, http://www.itmedia.co.jp/enterprise/articles/1001/20/news001.html (in Japanese).

[4] International Organization for Standardization, “Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 4: Transmission protocol,” ISO/IEC 14443-4:2008(E), July 2008.

[5] Japanese Industrial Standards Committee, "Specification of implementation for integrated circuit (s) cards－Part 4: High speed proximity cards," JIS X 6319-4:2010(J), October 2010 (in Japanese).

[6] NFC Forum, "NFC Digital Protocol Technical Specification," NFCForum-TS-DigitalProtocol-1.0, November 2010.

[7] Ecma International, "Near Field Communication - Interface and Protocol (NFCIP-1)," 2nd Draft Standard ECMA-340, February 2008

[8] Sony, "Contactless IC Card RC-S860," Brochure, January 2010, http://www.sony.net/Products/felica/business/data/RC-S860_E.pdf

[9] H.I.Business Partners, “Oversea Market Trend and Scale Forecast of NFC,” Japan, February 25, 2011, http://www.hibizpartners.com/Study/NFC/nfc.html (in Japanese)

[10] K. Yoshida, et. al., “A Trend Analysis of Delay and Packet Loss in Broadband Internet Environment through End Customers' View,” IEICE Trans. on Communications (Japanese Ed.), Vol. J91-B, No. 10, October 2008, pp. 1182-1192.

[11] O. Takahashi, et. al., “Implementation and Evaluation of Stealth Gateway System for 3G Mobile Network,” Trans. IPS Japan, Vol.44, No. 3, March 2003, pp. 637-646 (in Japanese).

[12] Sony, “Sony to launch next generation FeliCa contactless IC chip,” News Releases, June 2011, http://www.sony.net/SonyInfo/News/Press/201106/11-066E/

[13] Ecma International, “NFCIP-1 Security Services and Protocol Cryptography Standard Using ECDH and AES,” ECMA/TC47/2008/089, December 9, 2008.

[14] T. Lakshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-delay Products and Random Loss," IEEE/ACM Trans. on Networking, Vol. 5, No. 3, June 1997, pp. 336-350.

[15] V. Srivastava and M. Motani, "Cross-Layer Design: A Survey and the Road Ahead," IEEE Network Magazine, December 2005, pp. 112-119.

[16] A. Shiibashi, "Development of Suica Autonomous Decentralized IC Card Ticket System," Japan Railway & Transport Review, No. 50, September 2008, pp. 6-15.

[17] For example, REA, "Handbook of Mathematical, Science, and Engineering Formulas, Tables, Functions,

Graphs, Transforms," ISBN-0-87891-521-4, New Jersey, U.S.A., 1988, p.305

[18] H. Sakai and A. Arutaki, “Protocol Enhancement for Near Field Communication (NFC); Future Direction and Cross-Layer Approach,” IEEE INCoS2011, November 2011, Fukuoka, Japan.

[19] http://www.sics.se/~adam/old-uip/

[20] DNP, “IC Card Product Information,” Brochure, http://www.dnp.co.jp/bf/ic_card/products/2way.html (in Japanese).

[21] P. Koopman and T. Chakravarty, "Cyclic Redundancy Code Polynomial Selection For Embedded Networks," IEEE DSN2004, June 2004, Florence, Italy.