IP fragmentationとそのDNSSECに与える影響

力武 健次[†]　中尾 康二[†◇]　下條 真司[⋆]　野川 裕記[*]

ドメイン名詐称によるセキュリティ攻撃への対策として，DNSの認証プロトコルDNSSECを採用推進する動きが広がりつつある．しかし，DNSSECではサーバ応答のUDPペイロード長が大きくなるため，UDPデータグラムのIPパケット分割による通信の信頼性低下の可能性がある．筆者らは，IPパケット分割がDNSSECの運用に与える影響について，提案済みのDNSSECのIPパケット分割モデルに基づいて運用ネットワーク上でパケット伝送実験を行った．その結果，UDPデータグラムのペイロード長と損失率の間に相関は認められなかった．また，データグラムのバースト損失も確認できた．これらの結果から，筆者らは実運用システムの上でIPパケット分割はDNSSEC全体の信頼性に重大な影響は与えないという結論を得た．

IP fragmentation and the implication in DNSSEC

Kenji RIKITAKE[†], Koji NAKAO[†◇], Shinji SHIMOJO[⋆], and Hiroki NOGAWA[*]

DNSSEC, an authentication protocol for DNS is under major deployment phase as DNS spoofing becomes a popular security attack. DNSSEC increases the UDP payload length of the server response and the IP fragmentation of the UDP datagrams may undermine the reliability of communication. The authors conducted packet transfer experiments over a real-world network applying a proposed IP fragmentation model for DNSSEC, to evaluate how the IP fragmentation affects the DNSSEC operation. The results showed no correlation between the payload length and the loss rate of the transferred UDP datagrams. Burst datagram loss cases were also observed. The authors concluded from the results that the IP fragmentation would not affect the overall reliability of DNSSEC on a real-world network system.

1 Introduction

Domain Name System (DNS) is a mandatory subsystem of the Internet. Traditionally, DNS has no mechanism of identifying who actually put in an Resource Record (RR) to a served zone. This leads into criminal forgery of RRs, a popular form of security attack.

DNSSEC [1, 2, 3] is an DNS extension which is intended to provide cryptographic authentication of RRs, regarding the hierarchy of the delegation of authority from the trust anchor. DNSSEC requires EDNS0 [4], a DNS extension for exchanging larger payloads over UDP datagrams, which has been implemented on a well-known DNS programs such as BIND [5].

DNSSEC and EDNS0 assume the reliable transfer of fragmented IP packets for exchanging the large-payload UDP datagrams. The authors have published in a previous paper that DNSSEC answer messages including additional records exceeding the practical limitation of 1232 bytes, imposed by IPv6 default MTU (Maximum Transmission Unit), became approx. 30% of the samples on a real-world traffic dataset [6]. Ager et al. [7]

[†] 情報通信研究機構 インシデント対策グループ / Network Security Incident Response Group, NICT, Japan

[◇] KDDI 株式会社/ KDDI Corporation

[⋆] 大阪大学 サイバーメディアセンター / Cybermedia Center, Osaka University

[*] 東京医科歯科大学 情報医科学センター / Information Center for Medical Sciences, Tokyo Medical and Dental University

also reported that the size of 72 ~ 77% of the DNSSEC answer messages without errors including the Name Error responses world exceed the IPv4 fragmentation limit of 1472 bytes.

If fragmented packets were not reliably delivered over a wide-area network, the entire DNSSEC-based system would be jeopardized and become impractical. Quantitative estimation of how IP fragmentation affects UDP datagram delivery is needed to assess this underlying reliability issue of DNSSEC.

In this paper, the authors review how IP fragmentation affects UDP applications including DNS, and propose a set of experiment to measure the reliability of delivering IP fragments over a wide-area network. The authors also show the results of conducted experiments, and evaluate the results to find out how DNSSEC traffic is reliable upon a real-world IP networks.

In later sections, the authors first discuss the general IP fragmentation issues on UDP applications in Section 2. They propose a testing method for measuring reliability of large-payload UDP packet transmission for DNSSEC over a wide-area network, and show the test results in Section 3. The author presents the conclusions and future works in Section 4.

2 IP fragmentation issues on UDP applications

IPv4 mandates each node to reassemble fragmented packets (RFC1122 [8] Section 3.3.2) and allows intentional IP packet fragmentation (RFC1122 Section 3.3.3). This means an IPv4 packet could be fragmented at any time in a forwarding router between two end nodes.

IP fragmentation has many implications in security and reliability. While IP fragmentation allows IP packets to traverse routes of different MTUs, it also causes the reassembly overhead which may lead into an inefficient resource usage [9], and causes a security problem called *Tiny Fragment Attack* (RFC1858 [10] Section 3, RFC3128 [11]) by rewriting TCP headers with multiple IP fragments.

Many protocols implement strategies to avoid IP fragmentation as possible. IPv6 specification [12] only allows end-node fragmentation of IP packets, so that the routers do not have to queue and forward the fragments. Path MTU discovery [13, 14] is another way to discover minimum MTU over the end-to-end path to avoid causing IP fragmentation especially on TCP sessions, which is a default behavior on many servers, such as those running FreeBSD [15].

Delivery of large UDP datagrams exceeding the MTU, however, still assumes the IP fragmentation and reassembly for the proper operation. The end-node fragmentation is mandatory when splitting a UDP datagram into multiple IP packets.

Many wide-area UDP application try to avoid IP fragmentation by restricting the UDP payload size. For example, SIP (RFC3261 [16]), a signaling protocol for Internet telephony, mandates a larger request to be handled over TCP (RFC3261 Section 18). From this viewpoint, DNSSEC is a rather rare example of UDP application depends on the IP fragmentation, since the usage of TCP for DNSSEC is not mandated and is considered as a backup method when the UDP message exchange fails.

Determining the characteristics of fragmented UDP datagrams is important to find out whether DNSSEC has a different characteristics on the transport reliability from the traditional non-authenticated DNS. The difference between the two protocols is solely on the usage of the UDP transport, since the TCP transport usage is not changed at all. If the fragmented and non-fragmented UDP deliveries had little difference, the IP fragmentation issue would be of significant importance on later discussions.

An example of issues which need to be investigated for determining the characteristic difference on UDP transfer caused by the IP fragmentation are:

- the error rate of UDP datagram transfer over wide-area IP networks;
- the difference of the error rate of the fragmented and non-fragmented datagram delivery; and
- how the fragmented packets are reassembled on the actual implementations, especially when the reassembly is incompletely finished, i.e., only some and not all the fragments are properly delivered.

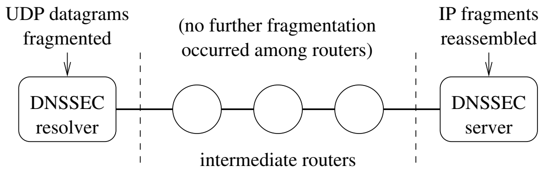

Fig. 1 Test model of UDP delivery for DNSSEC

3 Reliability test of fragmented UDP delivery for DNSSEC

3.1 Test methodologies

The authors decided testing fragmented UDP delivery on the real-world Internet hosts. The characteristics of errors over real-world networks cannot be simply modeled, since multiple entities (routers and links) and factors (bit/packet loss rates, queue length of routers, etc.) involve in the actual transfer.

As far as the authors surveyed, network simulators capable to handle IP fragmentation were not available among open-source software. For example, *ns* version 2 [17] did not handle IP fragments; *MIRAI-SF* [18] did not have tested the IP-fragmentation code; and *yans* [19] did not provide a practical programming interface.

Figure 1 shows the model of UDP delivery between DNSSEC resolvers and servers. In this model, end-node fragmentation is the primary measurement target. The authors assume end-node fragmentation are practically takes major part in most of the cases since:

- most of the Internet backbone networks are made of links with similar MTUs, such as Ethernet;
- the last-hop links on the each endpoint has the minimum MTU values in most cases, such as PPP-over-Ethernet [*1];
- the end-nodes will discover the end-to-end minimum MTU by path MTU discovery; and
- IPv6 requirements will force intermediate router nodes to discard any oversized packets exceeding the MTU.

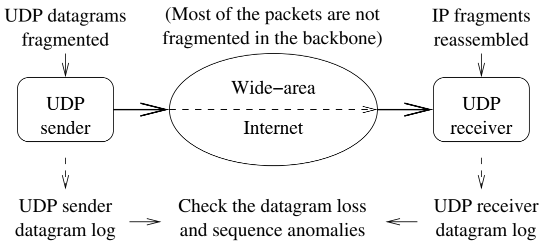
UDP datagrams fragmented
(Most of the packets are not fragmented in the backbone)
IP fragments reassembled
UDP sender
Wide–area Internet
UDP receiver
UDP sender datagram log
Check the datagram loss and sequence anomalies
UDP receiver datagram log

Fig. 2 Simplified model for UDP fragmentation test

Table 1 Specification for the test system hosts

UDP Sender
FreeBSD 5.4-RELEASE Pentium III/1.3GHz PC 4Gbyte memory 100Mbps to a campus network
UDP receiver
FreeBSD 4.11-RELEASE Celeron/1.3GHz PC 512Mbyte memory 50Mbps to an ISP network

3.2 Test environment

The primary purpose of this test was to measure the characteristic difference between the fragmented and non-fragmented UDP datagram delivery. Fig. 2 shows the actual test system configuration and procedure.

The test system consisted of two two hosts shown in Table 1. The hosts had sufficiently high capability to conduct the tests.

The UDP sender and receiver were connected through multiple ISPs with firewalls. The firewalls allowed UDP traffic from the sender to the destination port 53 of the receiver host. The connection between the two systems were stable and no large delay on interactive monitoring to the systems was observed.

The UDP sender sent a stream of packet using an interrupt-timer-driven Perl software of generating arbitrary length of unicast UDP datagram to the receiver. Each datagram was identified with sequential numbers, and the payload length of each datagram could be individually specified. The contents of the datagram was an arbitrary binary string. The authors assumed no content-dependent compres-

[*1] On consumer ADSL and optical-fiber links, a typical value of MTU is 1454 bytes, smaller than the Ethernet MTU of 1500 bytes.

Table 2　Conducted tests

(Date/Time in Japan Standard Time)
Case A1
14-MAR-2007 10:46:50 ~ 14-MAR-2007 13:33:41 1000000 datagrams in 100 datagrams/sec payload length CDF as shown in Figure 3
Case A2
14-MAR-2007 17:03:27 ~ 15-MAR-2007 18:13:52 1000000 datagrams in 10 datagrams/sec payload length CDF as shown in Figure 3
Case B1
17-MAR-2007 18:02:07 ~ 18-MAR-2007 05:09:26 1000000 datagrams ×4 in 100 datagrams/sec fixed payload length of 1200/2400/3600/4800 bytes
Case B2
18-MAR-2007 10:17:20 ~ 18-MAR-2007 21:24:40 1000000 datagrams ×4 in 100 datagrams/sec fixed payload length of 1200/2400/3600/4800 bytes

Table 3　Ping packet loss rate

Case	ping packets		
ID	sent	lost	loss rate
A1	600	0	0
A2	89825	10	1.11×10^{-4}
B1	40000	11	2.75×10^{-4}
B2	40000	4	1.00×10^{-4}

Table 4　Ping RTT statistics

Case	RTT statistics [ms]			
ID	min.	avg.	max.	sd
A1	27.454	28.027	50.627	0.975
A2	26.488	27.498	60.614	0.704
B1	26.647	28.786	61.067	1.109
B2	26.675	28.961	60.941	1.464

(sd: standard deviation)

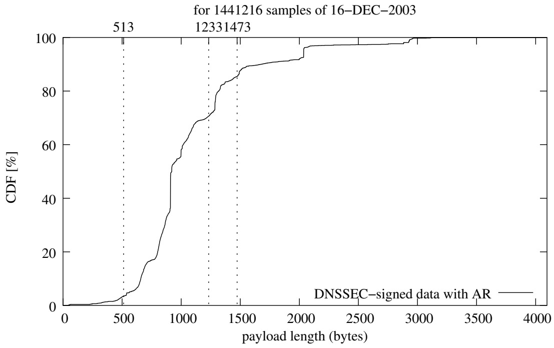

Fig. 3　CDF of DNSSEC answer payload length [6]

Table 5　UDP datagram loss rate for Case A1 and A2

Case	UDP datagrams			
ID	dps	sent	lost	loss rate
A1	100	1000000	63	6.30×10^{-5}
A2	10	906258	6	6.62×10^{-6}

(dps: datagrams per second)

sion was applied between the links.

The UDP sender and receiver logged the transmission and arrival time of each datagram, payload length, and the identification number. The datagram loss and sequence anomalies could be detected by reviewing the logs. The difference of internal clocks of the two hosts was periodically measured by using SNTP [20] to the same reference time server. During the experimentation period, the authors managed the difference between the two hosts to approx. 5ms in average.

3.3 Test cases and results

Table 2 shows the test cases conducted. Case A1 and A2 were datagram error rate tests based on the payload-length distribution on Fig. 3, a model given by the authors [6]. The payload length values were pre-calculated with a set of random numbers with the distribution represented by the CDF.

Case B1 and B2 were datagram error rate tests for 4 fixed-length datagrams, of 1200, 2400, 3600, 4800 bytes, which had 1, 2, 3, 4 fragments per datagram, respectively. Each size of datagram was sent 1000000 times. The datagram rate consumed ≤ 5Mbps, which was small enough to prevent occupying the available bandwidth and the congestion problem.

During those tests the RTT measurement of the link was also conducted, by sending 56-byte ICMP echo (ping) from the UDP sender to the receiver. Table 3 shows the ping loss rate during the tests, and Tab. 4 shows the Round-Trip Time (RTT). The results show the link RTT was stable around ≃ 28ms, and the ICMP packet loss rate was approx. $(1 \sim 3) \times 10^{-4}$.

Table 5 shows the datagram loss rate for the Case A1 and A2. The two cases had different data-

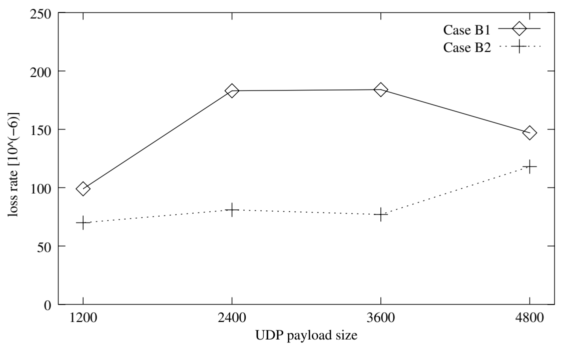

Fig. 4 Datagram loss rate of Case B1 and B2 for each fixed size value

Table 6 Datagram burst loss rate for Case B1 and B2 for each fixed size value

Case	length	lost datagrams		
ID	[bytes]	all	burst	single
B1	1200	99	61	38
B1	2400	183	77	106
B1	3600	184	49	135
B1	4800	147	143	4
B2	1200	70	67	3
B2	2400	81	38	43
B2	3600	77	0	77
B2	4800	86	32	86

(1000000 datagrams for each length)

gram rate, and the loss rate of A1 was about 10 times higher than that of A2. The authors learned that a continuous multiple (burst) datagram loss of 61 datagrams (for approx. 610ms) occurred in the Case A1, and none was observed in the Case A1 result. The results indicate a burst data loss might largely affect the overall datagram loss rates, and that a burst data loss might not be notified under a relatively low datagram transfer rate.

Figure 4 shows the datagram loss rate for the Case B1 and B2. In these cases datagram loss rates for different payload length and fragment pieces were measured. The loss rate was approx $(0.6 \sim 2) \times 10^{-4}$, which was within the same order of magnitude as the ICMP loss rate and that for the Case A1 and A2.

The line curve in Fig. 4 for the Case B1 shows that the loss rate did not increase as the number of IP fragments increased, while that of the Case B2 shows the increasing tendency. Theoretically the datagram loss rate is expected to increase proportionally to the number of IP fragments per UDP datagram[*2], but that characteristics were not found in either case.

Table 6 shows the numbers of detected burst errors during the tests. For example, the authors learned burst datagram losses occurred during the test cases, maximum for 67 datagrams (for approx. 670ms) during the Case B1. In the 4800byte/datagram sequence of Case B2, 143 of 147 losses were caused by 3 burst losses, of 56, 67, 17 datagrams, respectively.

During the whole cases no datagram sequence number reversal was observed. Datagrams were either lost or arrived in correct sequence. IP reassembly procedure will not forward the contents to the upper transport layer (Section 10.5 of Gary and Stevens [21]), so this shows very few UDP sequence reversal may occur in the real-world Internet, if any.

From those test cases and results, the authors observed:

- the absolute values of UDP datagram loss rate with the CDF of DNSSEC payload length distribution were approx. 10^{-5};
- the absolute values of UDP datagram loss up to 4 IP fragments per UDP datagram were approx. $10^{-5} \sim 10^{-4}$;
- no proportional relevance of the UDP datagram loss rate to IP fragment numbers per UDP datagram was observed;
- burst datagram loss contributed in a significant and sometimes major portion of overall datagram errors; and
- no UDP datagram reversal on the arrival was observed during the test.

4 Conclusions and future works

The authors conducted a preliminary set of UDP datagram loss rate tests on a real-world wide-area Internet, and observed that the number of IP fragments per each UDP datagram did not contribute to proportionally increase the datagram loss rate. The authors also found out burst errors up to a few hundred milliseconds contributed to the increase

[*2] $(1 + p)^n \simeq 1 + np$, if p is very small.

of UDP datagram loss rate. From these observations, the authors concluded that IP fragmentation details would not affect the UDP loss rate, and the overall reliability of DNSSEC on a real-world network system. The conclusions suggest that a network simulator which handles each UDP datagram as an atomic (undividable) event are useful for simulating DNSSEC traffics.

The future works include the following issues:

- more detailed analysis on the very-large-scale DNS and DNSSEC traffic;
- large-scale DNS and DNSSEC traffic simulation based on network simulators;
- optimization of UDP payload length with a more firewall-friendly method, such as Packetization Layer Path MTU Discovery [22], combined with the DNSSEC-level optimization of the message length; and
- development of DNS transport protocols resilient to burst errors.

Acknowledgements

The authors thank ODINS (Osaka Daigaku Information Network System) staff for their support.

This work was supported by NICT Incentive Research Fund for FY2006.

References

[1] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: DNS Security Introduction and Requirements (2005). RFC4033.

[2] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: Resource Records for the DNS Security Extensions (2005). RFC4034.

[3] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: Protocol Modifications for the DNS Security Extensions (2005). RFC4035.

[4] Vixie, P.: Extension Mechanisms for DNS (EDNS0) (1999). RFC2671.

[5] Internet Software Consortium: BIND. http://www.isc.org/bind/.

[6] Rikitake, K., Nogawa, H., Tanaka, T., Nakao, K. and Shimojo, S.: An Analysis of DNSSEC Transport Overhead Increase, *IPSJ SIG Technical Reports 2005-CSEC-28*, Vol. 2005, No. 33, pp. 345–350 (2005). ISSN 0919-6072.

[7] Ager, B., Dreger, H. and Feldmann, A.: Exploring the Overhead of DNSSEC. April 2005, http://www.net.informatik.tu-muenchen.de/~anja/feldmann/papers/dnssec05.pdf.

[8] R. Braden (Editor): Requirements for Internet Hosts – Communication Layers (1989). RFC1122.

[9] Kent, C. A. and Mogul, J. C.: Fragmentation Considered Harmful (1987). Digital Equipment Corporation Western Research Laboratory Technical Report 87.3, http://research.compaq.com/wrl/techreports/abstracts/87.3.html.

[10] Ziemba, G., Reed, D. and Traina, P.: Security Considerations for IP Fragment Filtering (1995). RFC1858.

[11] Miller, I.: Protection Against a Variant of the Tiny Fragment Attack (2001). RFC3128.

[12] Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification (1998). RFC2460.

[13] Mogul, J. and Deering, S.: Path MTU Discovery (1990). RFC1191.

[14] McCann, J., Deering, S. and Mogul, J.: Path MTU Discovery for IP version 6 (1996). RFC1981.

[15] The FreeBSD Project: FreeBSD. http://www.freebsd.org/.

[16] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol (2002). RFC3261.

[17] USC/ISI: The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[18] NICT: MIRAI-SF: MIRAI-Simulation Framework. http://mirai-sf.nict.go.jp/overview_e.html.

[19] INRIA: yans: Yen Another Network Simulator. http://yans.inria.fr/.

[20] Mills, D.: Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI (1996). RFC2030.

[21] Wright, G. R. and Stevens, W. R.: *TCP/IP Illustrated, Volume 2*, Addison–Wesley (1995).

[22] Mathis, M. and Heffner, J.: Packetization Layer Path MTU Discovery (2007). RFC4821.