IPv6移行に伴うDNSペイロード長増加に関する解析と考察

力武　健次[†,††a)]　野川　裕記[†††b)]　田中　俊昭[†c)]　中尾　康二[††††d)]
下條　真司[†††e)]

An Analysis of DNS Payload Length Increase during Transition to IPv6

Kenji RIKITAKE[†,††a)], Hiroki NOGAWA[†††b)], Toshiaki TANAKA[†c)],
Koji NAKAO[††††d)], and Shinji SHIMOJO[†††e)]

あらまし　DNSのサーバとリゾルバ間で交換される情報の量や内容は，IPv6の導入に伴うアドレス長の増加やその他の技術的拡張によって変わりつつある．この変化によってDNSの使用するトランスポート層プロトコルとそのペイロード長に関して根本的な設計変更が必要となっている．本論文では，まずIPv6の技術的要請に伴いDNSの各プロトコルがどのような影響を受けるかについて考察する．次に既存のDNSトラヒックを収集解析し，IPv6への移行に伴い起こる変化をパケット長再計算によるシミュレーションにより評価することで，現在のUDPペイロード長512バイトを超える応答がadditional recordsを含めた場合0.04%から1〜3%へ増えることを示す．その上でこのペイロード長の増加に対応するための手法として，拡張プロトコルEDNS0が有効なことを示す．

キーワード　IPv6，ドメイン名システム，トラヒック解析，トランスポート層プロトコル，ペイロード長

1. まえがき

DNS（ドメイン名システム）はインターネットに不可欠なサブシステムであり，その重要性は来るべきIPv6の時代でも変わらない．DNSの役割は，各ドメイン名に対応するIPアドレスなどの情報であるRR（Resource Record）を蓄積した分散データベースに対して，円滑な問合せと応答処理を提供することにある．

DNSのプロトコル自身は，1987年にMockapetris [1], [2] が定義し後にインターネット標準[3]となって以来大きく変化していない．しかしながら，DNSのプロトコルも利用の変化に応じた見直しが必要になっている．例えばDNSのトランスポート層プロトコルのうち，ほとんどの場合で使われるUDP [4] によるDNSの問合せと応答は，そのペイロード長が512バイトを超えないことを前提にしており，超えた場合はUDPでの交換の内容を廃棄し，再度TCP [5] による交換を行う．

仮にペイロード長が512バイトを超えるDNSの問合せと応答の頻度が増えると，TCPでの再送件数も増える．この際UDPでの交換内容は生かされずまるまる無駄になってしまい，交換するパケット数が増える．また，DNSサーバではUDPの際は必要としないTCPのための状態保持が必要となる．結果としてDNSサーバやそれらが接続しているネットワークの負荷が増える．ルートサーバ（Root Servers）のように頻繁にアクセスされるサーバではその影響は多大となる．

実際にはIPv6の普及などによりDNSのトランスポート層プロトコルのペイロード長は増加傾向にあり，今後512バイトのペイロード長制限に収まらない

† （株）KDDI研究所セキュリティグループ，上福岡市
Security Laboratory, KDDI R&D Laboratories, Inc., 2–1–15 Ohara, Kamifukuoka-shi, 356–8502 Japan
†† 大阪大学大学院情報科学研究科，吹田市
Graduate School of Information Science and Technology, Osaka University, 1–5 Yamadaoka, Suita-shi, 565–0871 Japan
††† 大阪大学サイバーメディアセンター，茨木市
Cybermedia Center, Osaka University, 1–5 Mihogaoka, Ibaraki-shi, 567–0047 Japan
†††† KDDI株式会社情報セキュリティ部，東京都
Information Security Department, KDDI Corporation, 3–10–10, Iidabashi, Chiyoda-ku, Tokyo, 102–8460 Japan
a) E-mail: kenji@kddilabs.jp
b) E-mail: nogawa@cmc.osaka-u.ac.jp
c) E-mail: tl-tanaka@kddi.com
d) E-mail: ko-nakao@kddi.com
e) E-mail: shimojo@cmc.osaka-u.ac.jp

DNS の問合せと応答の頻度は増えると予想しており，ペイロード長の増加に対して何らかの対策を行うことが必要になる．例えば IPv6 の普及に伴い，ドメイン名と 16 バイト長の IPv6 アドレスを対応づけるための AAAA RR がより多く使われる．このため，現在の 4 バイト長の IPv4 アドレスに対応する A RR がペイロードの大きな部分を占める現状から考えると，IPv6 への移行に伴いペイロード長はより大きくなる方向にある．

512 バイトのペイロード長制限は既に運用上の問題を引き起こしている．例えばルートサーバは，この制限により最大数が 13 に抑えられており数を増やせない．TCP での再送による負荷の増加を考えると 512 バイトを超えた応答は現実的ではないため，今後のサーバ数増加を可能にするには何らかの抜本的改善策が必要になる．その他のドメインのサーバについても，IPv6 への移行が進めばこのペイロード長制限の問題はより顕著になる．

本論文ではまず DNS の UDP ペイロード長の制限によって IPv6 移行時に起き得る問題を述べる．そしてそれらの問題がもたらす影響を，DNS の問合せと応答に関する実トラヒックのデータをもとに IPv6 への移行に伴い増えると予想する RR を追加した場合のシミュレーションを行い，定量的に予測して評価する．また，問題の解決法として，DNS の拡張プロトコル EDNS0 [6] などを示し，予想できる効果を比較評価する．

以下，**2.** では DNS のトランスポート層プロトコルとその制限について説明する．**3.** では，IPv6 化などの利用による RR 長とペイロード長の増加とそれによる影響について述べる．**4.** では DNS の実トラヒックに基づく解析と行ったシミュレーションの評価結果を述べる．**5.** ではトランスポート層プロトコルのペイロード長増加への対策手法とそれらがもたらす影響について述べる．最後に **6.** で結論として，IPv6 化によって 512 バイトを超えるペイロード長の応答が 0.04 %から 1~3%に増えること，また EDNS0 が対策法として有効なことを示す．

2. DNS トランスポート層プロトコル

本論文では以降二つの DNS プログラム間で情報をやり取りするためのプロトコルを DNS トランスポート層プロトコルと呼ぶ．

DNS トランスポート層プロトコルの定義は，DNS のアーキテクチャを定義する RFC1034 [1] と実装の詳細を定義する RFC1035 [2] で与えられる．また，インターネットに接続するホストの技術的要件を定めた RFC1123 [3]，及び DNS プロトコル仕様の詳細を補足した RFC2181 [7] においても DNS トランスポート層プロトコルの使い方が記されている．

DNS トランスポート層プロトコルは，より一般的なインターネットのトランスポート層プロトコルである UDP と TCP 上に実装されており，以下の二つの役割をもつ．

ゾーン転送　特定のドメインに対する RR の集合であるゾーン情報の取得を，そのゾーンの権限をもつサーバより行う．ゾーン転送は DNS サーバ間で情報をもち合うことで，サーバの障害に対する冗長性を高めることを目的として広く用いられている．この役割は専ら TCP 上で実現される．

RR の問合せと応答　特定のドメイン名についてサーバとリゾルバ（DNS の問合せを行うクライアント）間で RR の問合せと応答を行う．アプリケーションソフトウェアはこの機能を使い，ドメイン名の IP アドレスへの変換等を行う．ほとんどの問合せと応答のやり取りは UDP 上で行われるが，TCP 上のやり取りも定義され使われている．

RFC1035 の Section 4.2.1 では，UDP 上の問合せと応答の最大長は 512 バイトに制限されている．これを超えた場合は，サーバからの応答は 512 バイト以内に切り詰められ，応答ヘッダ中にその旨が示される．

TCP での問合せと応答にはメッセージ本体の前に 16 ビット長の正の整数でメッセージ長を示すことになっており，TCP の許容する最大ペイロード長を考えると問合せと応答の最大長は 65533 バイトとなる．

本論文では以降 RR の問合せと応答のやり取りに関する問題のみを扱う．これはゾーン転送は本質的に TCP で行われるファイル転送と同様で転送量の制限もなく，実運用で数時間に一度という転送の発生頻度から見てサーバの処理やネットワークの性能に与える影響は小さいためである．

3. RR 長とペイロード長増加の影響

この章では IPv6 化などによる DNS トランスポート層プロトコルでの RR 長とペイロード長の増加傾向と原因，そしてそれらのもたらす影響について述べる．

表 1　IPv4 から IPv6 への変化に伴う PTR RR の変更
Table 1　Change in PTR RRs from IPv4 to IPv6.

IPv4	111.222.123.234 → 234.123.222.111.in-addr.arpa
IPv6	0123:4567:89ab:cdef:1213:2324:3435:4647 → 7.4.6.4.5.3.4.3.4.2.3.2.3.1.2.1.f.e.d.c.b.a. 9.8.7.6.5.4.3.2.1.0.ip6.arpa

3.1　IPv6 化に伴う RR の種類の変化と長さの増加

IPv4 から IPv6 への移行に伴い，使われる RR の種類と内容また長さの増加について，以下の各点を考慮する必要が生じる．

- IPv6 でのドメイン名から IPv6 アドレスを参照するために，DNS では AAAA RR（RFC1886 [8] Section 2.2）を使用する(注1)．

A RR から AAAA RR への移行に伴い，IP アドレスを示す RR の長さが増加する．IPv4 の A RR（RFC1035 Section 3.4.1）ではドメイン名に対して 32 ビット（4 バイト）のアドレス値を RDATA フィールドに定義するのに対し，AAAA RR ではドメイン名に対して 128 ビット（16 バイト）のアドレス値を定義している．

- IP アドレスから対応するドメイン名を参照（逆引き）するために，IPv4 では in-addr.arpa ドメインを使用していた（RFC1034 Section 5.2.1）が，IPv6 では ip6.arpa ドメイン（RFC3152 [11]）を使用する．

逆引きに使う PTR RR の長さは，IPv4 から IPv6 への移行に伴い長くなる．IPv4 では IP アドレスに対応する逆引き用の名前が 28 バイトであるのに対し，IPv6 では 72 バイトである（表 1）．

RR 長の増加に伴い，DNS トランスポート層プロトコルを使った RR の問合せと応答に際して，以下の変化が発生すると予想する．

- IPv4 から IPv6 への移行に際し，IPv4 から IPv4 と IPv6 の併用，そして全面的な IPv6 化への過程をたどる上で，IP アドレス参照に関する RR の応答長は以下のように変化する．

– IPv4 アドレスの A RR を IPv6 アドレスの AAAA RR に置き換える際は，32 ビットから 128 ビットのアドレス長の変化に伴い，RR 長は 12 バイト増加する．

– あるホストの IPv4 アドレス各々に IPv6 アドレスを一つ追加する場合，最低でも 28 バイト追加することが必要になる．具体的には，RFC1035 Section 4.1.4 のドメイン名圧縮を使用した場合の index 情報が 2 バイト，RR ヘッダが 10 バイト，IPv6 アドレスが 16 バイトの計 28 バイトが増加する．

- 逆引き空間が in-addr.arpa から ip6.arpa に変わることにより，逆引きの問合せを含むペイロード長は問い合わせる名前ごとに最大で 48 バイト増加する可能性がある．しかし RFC1035 のドメイン名圧縮により問合せと応答中の逆引き用ドメイン名は 2 回目の出現からはもとのドメイン名の長さにかかわらず 2 バイトまで減少し，また対応する名前を示す PTR RR の長さは変わらない．そのためこの増加がペイロード長全体に与える影響は少ないと考える．

3.2　IPv6 化以外の DNS ペイロード長増加の要因

IPv6 化の直接の影響以外の DNS ペイロード長の増加要因としては以下の 2 点がある．これらは IPv6 化の直接の結果として起こり得るものではないが，現在進められている DNS の機能強化や利用法の変化に伴い起こりつつあるもので，DNS のトランスポート層プロトコルの今後を考える上で考慮すべき事項である．

- DNSSEC [12] の導入により，各 RR ごとに公開鍵による署名のための RR が追加される．RFC3226 [13] Section 2.1 では，各署名用の RR の長さは 80〜800 バイト，うち大部分は 200 バイト以下であろうと分析している．

- WWW の仮想ドメインサービスのように複数のドメイン名で一つの IP アドレスを共有するサービスなど，一つの問合せに対して多数の応答を返す事例が増加している．例えば IP アドレスの逆引き問合せに対し，対応するすべてのドメイン名を PTR RR として返したり，ドメイン名への問合せに対しサービスを担う複数のホストの IP アドレスを返してランダムに選択させ負荷分散するという手法が一般化し，RR 数の増加を招いている．

3.3　ルートゾーンでのペイロード長制限の影響

512 バイトのペイロード長制限の影響を最も受けているのは，DNS の階層の原点となるルートゾーン（Root Zone）の情報を提供するルートサーバに関する DNS 問合せの応答である．

(注1)：IPv6 でのアドレス空間分割を容易にする目的でビット単位で階層を区切って DNS 空間の権限委任を行う A6 や DNAME といった方式も提案されているが [9]，これらの階層的な方式は experimental とされ，実用段階の普及促進には引き続き AAAA RR を使うことが決まっている（RFC3363 [10] Section 2.a）．

現在ルートゾーンのサーバのドメイン名は a.root-servers.net というように a から m までの 13 個の名前を使っている．512 バイト長制限下では，これ以上サーバを増やすことはできない．これはルートサーバに対して TCP の問合せを行うと負荷が増えるため，極力 UDP で問合せを行うことが推奨されているからである．この制限により，新規ルートサーバの追加だけでなく IPv6 アドレスの追加も事実上できない状態になっている．

5.7 で，この問題を IP アドレスの選択処理によって対処する方法を述べる．

4. DNS の実トラヒック解析と IPv6 移行時のシミュレーション

この章では IPv4 から IPv6 への移行過程及び移行後に UDP 上の DNS ペイロード長の傾向がどのように変わるかを，実トラヒックに対するパケット長再計算によるシミュレーションによって予想する．

4.1 DNS トラヒック収集の方法

IPv6 への移行による AAAA RR の増加がどのようにペイロード長に影響するかを調べるため，図 1 に示すシステムで DNS 実トラヒックの収集を行った．このシステムでは，大阪大学のキャンパスネットワークで学内–学外間トラヒックを扱う二つのコアスイッチのうちの一つからミラーリングによってパケットを複製し，1000BASE-SX で接続した FreeBSD [14] の動作する観測用ホストでネットワーク監視用プログラム snort [15] を使い複製結果を収集した．UDP の fragment offset が 0 でないパケットは無視している．

収集作業は日本時間の 2003 年 11 月 28 日朝と，2003 年 12 月 16 日深夜から，それぞれ 12 時間連続で収集作業を行った．収集結果の解析には tcpdump [16] に DNS パケット分析機能を追加したものを使用した．

4.2 解析対象の選定

本研究では収集したパケットのうち解析するものを IPv4 上の UDP のみとし，更に DNS に割り当てられているポート 53 番を発信元か送信先どちらかに指定しているものに限定した．また，解析対象は DNS サーバの応答のみとし，問合せは除外した．この理由としては以下の 4 点がある．

- 実運用では DNS のトラヒックのほとんどが IPv4 上で流れており，IPv6 のみを使った DNS のやり取りは少ない．例えばルートサーバには IPv6 のアドレスは割り当てられていない．同様に IPv6 の逆引き（ip6.arpa）ゾーンの四つの authorized server のうち二つは，IPv4 でしか問合せができない．また，IPv4 と IPv6 それぞれからアクセスできる DNS 空間の分断を防ぐため，すべての DNS サーバは IPv4 上でゾーン情報を提供しなければならないことを推奨する文書もある [17].
- TCP による RR の問合せと応答が発生する頻度は UDP のそれに比べ頻度としては 0.12〜0.16%程度である（表 2）．また，TCP による応答の 9 割程度が DNSSEC の TSIG 認証 [18] での秘密鍵の交換に使われる TKEY RR [19] である（表 3）．その他の応答総数は 2 回の収集作業どちらも 500〜600 の間であり，これらは UDP による DNS 応答総数の 0.01〜0.02 %程度でしかなく，本研究のシミュレーションでは除いても大勢に影響ないと考える．
- DNS の名前は 255 バイト以内に制限されてお

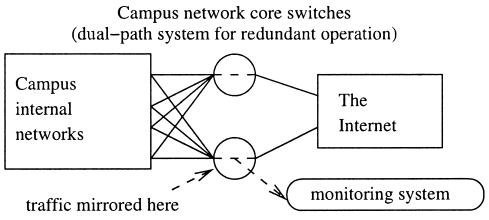

図 1　DNS トラヒック収集用システム構成図
Fig. 1　System diagram for collecting DNS traffic.

表 2　解析用に収集した DNS 応答の総数
Table 2　Numbers of DNS answers collected for the analysis.

starting and ending times of each measurement (JST: Japan Standard Time)	numbers of answers		TCP/UDP (%)
	TCP	UDP	
28-NOV-2003 0836〜2035JST	7387	6249736	0.118
16-DEC-2003 0047〜1246JST	4581	2997881	0.153
Total	11968	9247617	0.129

表 3　収集した TCP による DNS 応答の特徴別数と比率
Table 3　Numbers and percentage of characteristics in collected DNS answers with TCP.

Characteristics	28-NOV-2003	16-DEC-2003
Single TKEY RR	6838 (92.6%)	4018 (87.7%)
Non-TKEY RRs of length >512	51 (0.7%)	61 (1.3%)
Non-TKEY RRs of 1 ≤ length ≤ 512	341 (4.6%)	334 (7.3%)
Others	157 (2.1%)	168 (3.7%)

(length values are in Bytes)

表 4 収集した UDP による DNS 応答中の RR の種類別比率
Table 4 Percentage of RRs in collected UDP DNS answers.

RR Type	28-NOV-2003	16-DEC-2003
A	40.17%	42.69%
AAAA	0.87%	1.95%
CNAME	0.94%	0.59%
MX	1.26%	1.73%
NS	35.64%	39.22%
OPT	16.32%	9.12%
PTR	0.84%	0.98%
SOA	3.96%	3.73%
Others	$< 0.01\%$	$< 0.01\%$
Numbers of total RRs	22642277	14460833

り（RFC1035 Section 2.3.4），DNS で問い合わせる名前で使う QNAME 形式にすると更に 2 バイト増え，最大で 257 バイトとなる．各問合せ当りのペイロード長を考えた場合，これとヘッダ 12 バイト，また問合せの種類を決めるのに必要な 4 バイトを加えても最大 273 バイトであり 512 バイトには達しない．そのため 512 バイトを超えるかどうかが問題になる本論文での評価からは除外しても問題ないと考える．

表 4 に収集した DNS 応答中の RR の種類と数を示す．A RR と AAAA RR を比べた場合，A RR が全体の 4 割以上を占めているのに対し，AAAA RR は 0.8%から 2%弱を占めているに過ぎない．これはアドレスを示す RR が DNS の応答の主な部分を占めており，また IPv6 への移行がまだ進んでいないことを示しているといえる．

解析方法として，DNS の応答のそれぞれについて，全体のペイロード長と含まれる RR の数を RR の種類ごとに集計した．これによって，**3. 1** で述べた AAAA RR の追加や置換えによるペイロード長の変化をシミュレートできる．

4. 3 シミュレーションの方法

シミュレーションは以下の 2 通りの場合を想定して行った．

IPv4+IPv6 併用 移行期において，IPv4 のアドレスに対応する A RR 一つに対し AAAA RR を追加することを想定し，A RR 一つにつき 28 バイトずつペイロード長を増やす．

IPv6 移行後 A RR がすべて AAAA RR へ移行したことを想定し，A RR 一つにつき 12 バイトずつペイロード長を増やす．

上記の処理は answer section と additional section 双方のすべての A RR に対して行う．

この方法では既に A RR と AAAA RR が割り当てられているドメイン名のことを考慮していないが，実際は全体の A RR の数が AAAA RR に比べ 20 倍以上も多く（表 4），A RR のみを考慮したシミュレーションでも DNS 応答全体の変化の予測としては支障ないと考える．

また，RFC2181 の Section 9 では，DNS 応答中の RR の内 additional section に含まれるなど必ずしも一度に送られる必要のないものについて，その RR が原因でペイロード長制限を超える場合は，同一のドメイン名に対応した同じ種別の RR の集合（RRset）全体を応答から削除し，応答全体をペイロード長に収めることが推奨されている．

本論文のシミュレーションではこの処理を想定し，AR（Additional Records，additional section に含まれる RR）について，AR を含む場合と AR 中の A RR を削除した場合の 2 通りを想定した．

実際の運用では，AR の削除を徹底した場合問題が発生する場合もある [20]．また AR が含むアドレスを示す RR は，同一応答内の他のサーバへの参照を示す NS RR に対応したアドレスであることが一般的であり，サーバへの問合せの回数を減らすために不可欠である．AR の削除あるいは部分的選択はあくまで 512 バイトのペイロード長制限下での運用上の問題と位置づけるべきと考える．以後の分析では，主に AR を含む場合について取り扱う．

4. 4 解析とシミュレーション結果の分析

表 5 に収集した各応答と A RR の AAAA RR 追加と置換えのシミュレーション結果の統計を示す．

AR の削除あるいは AR を残したいずれの場合でも，AAAA RR の追加や置換えによって平均（μ）や標準偏差（σ）が増加している．AR を残した場合では 512 バイトを超えるペイロード長の割合が原データの 0.04 %以下からシミュレーション後は全体の 1~3%へ達している．一方，AR を削除した場合では 512 バイトを超えるペイロード長の割合は 0.001~0.002%へと削除しない場合の 1/10 ~ 1/20 に小さくなるが，この場合もシミュレーション後の割合は全体の 0.06~0.14 %へと増える．これらの結果より，AR の削除あるいは AR を残した双方の場合で，シミュレーション後の 512 バイトを超えるペイロード長の割合はシミュレーション前のものに比べ 20~100 倍程度に増えていることが分かる．

表 5　シミュレーション結果の統計
Table 5　Statistics for the simulation results.

for 6249736 samples of 28-NOV-2003				
	μ	σ	max	>512
raw data w/o AR	81.80	52.73	1149	0.001
raw data with AR	108.26	79.68	1192	0.023
AAAA+A w/o AR	89.14	66.66	3025	0.136
AAAA+A with AR	149.01	142.28	3124	2.117
AAAA→A w/o AR	84.95	57.74	1953	0.075
AAAA→A with AR	125.72	105.98	2020	1.124
for 2997881 samples of 16-DEC-2003				
	μ	σ	max	>512
raw data w/o AR	102.28	53.57	944	0.002
raw data with AR	137.90	87.16	1112	0.035
AAAA+A w/o AR	110.58	66.02	1485	0.128
AAAA+A with AR	195.55	155.43	2285	2.772
AAAA→A w/o AR	105.84	57.82	973	0.064
AAAA→A with AR	162.60	115.83	1533	1.656

μ: mean value (Bytes)
σ: unbiased standard deviation (Bytes)
max: maximum payload length (Bytes)
>512: % of payloads longer than 512 Bytes
w/o AR: without Additional Records

表 6　収集した UDP による DNS 応答の特徴別比率
Table 6　Percentage of characteristics of RRs in collected UDP DNS answers.

Characteristics	28-NOV-2003	16-DEC-2003
Answer with authority (1)	1730603 (27.7%)	1070553 (19.5%)
Server errors (2)	1217000 (19.5%)	168369 (5.6%)
Referring to other servers (3)	1763301 (28.2%)	1164987 (38.9%)
Others	1538432 (24.6%)	593972 (19.8%)

(1) ancount>0 or RCODEs including NXDOMAIN and DNS UPDATE messages
(2) SERVFAIL, FORMERR, NOTIMP
(3) nscount>0 but ancount=0

収集した DNS 応答の特徴のうち，問合せを受けたサーバ自体の回答が得られたもの(注2)か，他のサーバを参照している(注3)のか，プロトコルエラーなのかについて表 6 に分類した．回答が得られたものは，全体の 20∼28%である．また他のサーバを参照しているものが全体の 28∼39%あり，これが AR の有無によるペイロード長の違いにつながると考える．

収集した DNS 応答中の QNAME 長の統計を表 7 に示す．2003 年 12 月 16 日の分のサンプル数は表 5 に比べて 14 個少ない．この 14 個はすべて同一のホストから返された不正なパケットであり，QNAME 部を含んでいないため統計の対象とはしなかった．

表 7　QNAME 長の統計
Table 7　Statistics for the QNAME length.

	μ	σ	max
28-NOV-2003 (6249736 valid samples)	22.49	5.65	193
16-DEC-2003 (2997867 valid samples)	23.59	6.18	94

μ: mean value (Bytes)
σ: unbiased standard deviation (Bytes)
max: maximum QNAME length (Bytes)

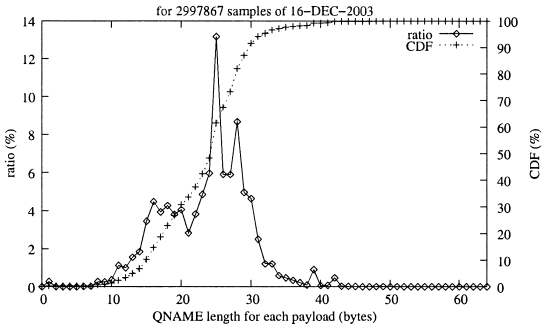

図 2　2003 年 12 月 16 日に収集した各 DNS 応答の QNAME 長とその累積分布関数
Fig. 2　QNAME length and the CDF for each DNS answer collected on 16-DEC-2003.

QNAME 長の平均は約 22∼23 バイトであり，ペイロード長全体の平均値の 17∼21%である．これより QNAME 以外の応答内容が DNS ペイロードの大部分を占めていることが分かる．

2003 年 12 月 16 日に収集した DNS 応答中の QNAME 長の分布を図 2 に示す．CDF（累積分布関数）を見る限り，この分布中では QNAME 長が 43 バイト以上の DNS ペイロードは 0.2%以下とまれにしか発生していない．

図 3 に 2003 年 12 月 16 日に収集した DNS 応答ごとの A RR 数の分布を示す(注4)．RR 数が 13 のところのデータが目立っているが，これはルートサーバや `.com`，`.net` の gTLD（generic Top Level Domain）サーバなど頻繁にアクセスされるゾーンの RR 数が

(注2)：ヘッダ中の ancount>0，あるいは RCODE 中に NXDOMAIN や DNS UPDATE [21] 関連応答などサーバ自身がゾーン中の内容を判断して存在の有無を応答したもの．
(注3)：ヘッダ中の nscount>0 であり，かつ ancount=0 であるもの．
(注4)：RR 数が 75 のデータがあるが，これは実際にはペイロード長が MTU（Maximum Tranmission Unit）を超えるためパケットが分断されていた．該当する問合せを後日再度 TCP で行ったところ，分断されていない完全な応答での A RR の数は 150 に達し，ペイロード長は 2665 バイトであった．

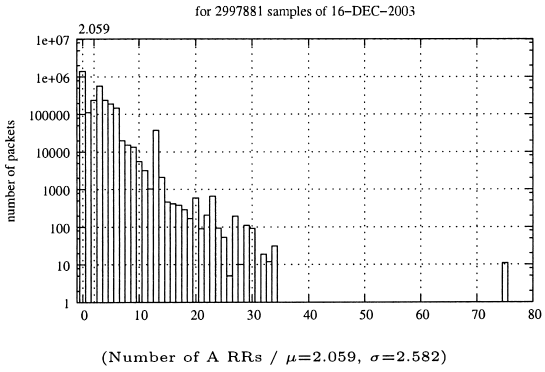

図 3　2003 年 12 月 16 日に収集した各 DNS 応答の A RR 数
Fig. 3　Numbers of A RRs for each DNS answer collected on 16-DEC-2003.

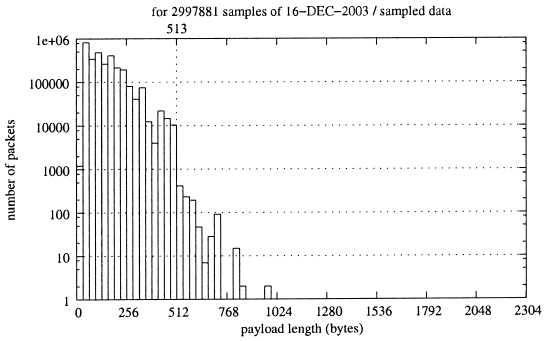

図 4　2003 年 12 月 16 日に収集した DNS 応答
Fig. 4　DNS answers collected on 16-DEC-2003.

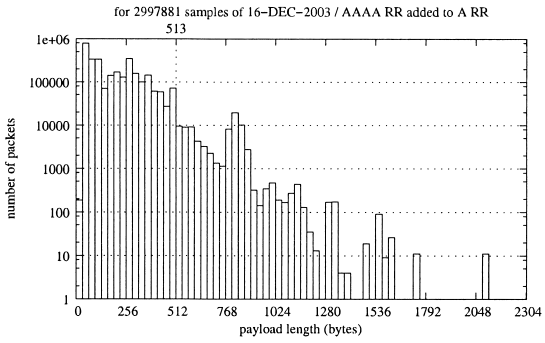

図 5　2003 年 12 月 16 日に収集した DNS 応答の A RR に AAAA RR を加えた場合のシミュレーション結果
Fig. 5　Result of a simulation adding an AAAA RR to each A RR for the DNS answers collected on 16-DEC-2003.

13 であり，これらからの応答と推定する．

収集した DNS 応答とそのシミュレーション結果の一例として，2003 年 12 月 16 日に収集した DNS 応答のヒストグラムを図 4 に，A RR ごとに AAAA RR を加えた場合のシミュレーション結果を図 5，また AAAA RR に A RR を置き換えた場合のシミュレーション結果を図 6 に示す．AAAA RR の追加や置換えの場合，表 5 同様，512 バイトを超えるパケットの割合が増えていることが分かる．

512 バイトを超えるパケットの割合の比較のため，図 7 に，収集した DNS 応答とシミュレーション結果のペイロード長 512 バイト前後で変化の大きな部分の CDF を示した．ペイロード長 512 バイトを超える割合は収集したデータでは <0.1% なのに対し，シミュレーション結果での AAAA RR の追加の場合は約 2.77%，AAAA RR の置換えの場合は約 1.66% まで増加している．

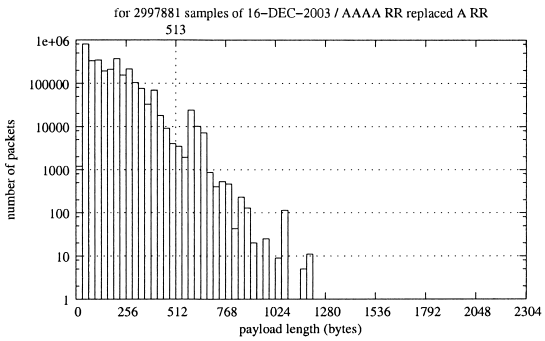

図 6　2003 年 12 月 16 日に収集した DNS 応答の A RR を AAAA RR に置き換えた場合のシミュレーション結果
Fig. 6　Result of a simulation replacing each A RR to an AAAA RR for the DNS answers collected on 16-DEC-2003.

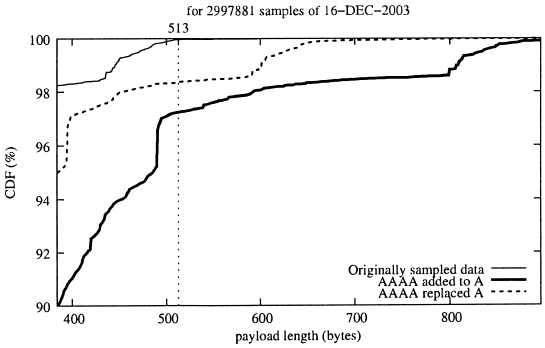

図 7　2003 年 12 月 16 日に収集した DNS 応答とシミュレーション結果の累積分布関数
Fig. 7　The CDF of DNS answers collected on 16-DEC-2003 and the simulation results.

5. ペイロード長増加への対策法と考察

この章では DNS のペイロード長増加への対策として現在取られている方法とその効果について考察する．

5.1 シミュレーション結果からの予想

4.3 のシミュレーションの結果として，IPv4 から IPv6 への移行過程また移行後には 512 バイトを超える UDP ペイロードの割合は全体の 0.04%以下から 1〜3%に増加し，その分だけ TCP での再送が発生する．これはすべての DNS サーバで問題になるため，何らかの具体的な対策が必要になる．

5.2 EDNS0 によるペイロード長拡張とその効果

512 バイトのペイロード長制限に対処する方法の一つとして，EDNS0 [6] という DNS トランスポート層プロトコルの拡張手順が提案されている．EDNS0 ではプロトコル拡張のための OPT RR を定義し，サーバとリゾルバそれぞれが送受信可能な最大ペイロード長をやり取りすることで，RFC1035 の 512 バイトの制限を超えた伝送を可能にする．

具体的な EDNS0 の処理手順は以下のとおりである．

- まず EDNS0 に対応しているリゾルバが問合せの際 OPT RR を使い EDNS0 に対応していることと許容最大ペイロード長を伝える．
- 問合せを受信したサーバが EDNS0 に対応していれば OPT RR を返して受信した最大ペイロード長を考慮した処理を行う．対応していなければ OPT RR を無視する．
- リゾルバは受信した応答中の OPT RR の存在を調べればサーバが EDNS0 に対応しているかどうかが分かる．

表 8 に，代表的な DNS サーバでの EDNS0 の対応状況を示した．BIND [22]，NSD [23]，Windows 2003 Server [24](注5) は EDNS0 に対応している．対応する各サーバの許容最大ペイロード長は，コンパイル時または実行時に変えることができる．一方，djbdns [25] は対応していない．

収集データからは表 9 に示すように，応答全体のほぼ半数以上で EDNS0 が使われている．また許容最大ペイロード長 UDPsize はほぼ 3 分の 2 以上の応答で 4096 バイトが指定されている．表 9 の結果から見て，EDNS0 は今後の普及の可能性が高いことが予想できる．

図 4 でペイロード長が 512 バイトを超えているパケットが少量ながら存在するのは，実際に EDNS0 の拡張に対応したやり取りが行われているのが原因だと考える．

表 8 代表的 DNS サーバでの EDNS0 対応状況
Table 8 EDNS0 support of popular DNS servers.

Server name and the versions	EDNS0-capable? (yes/no)	maximum length (bytes)
BIND 8 and 9 (since 8.3.0)	yes	4096
djbdns-1.05	no	N/A
NSD 1 and 2	yes	4096
DNS Server of Windows Server 2003	yes	1280

(maximum payload length values shown are the default values, configurable in the source-code or by a runtime parameter)

表 9 EDNS0 とその指定ペイロード長の内容詳細
Table 9 Usage details of EDNS0 and the specified payload length.

for 3694918 answers of 28-NOV-2003 with OPT RRs (43.97% of 6249736 answers)				
UDPsize	512	1280	2048	4096
numbers	13	1399	706497	2987009
%	< 0.01	0.038	19.12	80.84
for 1318187 answers of 16-DEC-2003 with OPT RRs (59.12% of 2997881 answers)				
UDPsize	512	1280	2048	4096
numbers	1	1126	434632	882428
%	< 0.01	0.085	32.97	66.94

5.3 EDNS0 によるオーバヘッド

EDNS0 で UDP ペイロード長伝送のために使われる OPT RR は，他に拡張データが付かない場合で 11 バイトを消費する．もっとも，OPT RR で指定するペイロード長が十分長ければ OPT RR の影響は軽微と考える．実際には表 5 の測定結果から見てペイロードの最大長が 4096 バイトを超えるのは極めてまれであり，仮に EDNS0 ですべての許容最大 UDP ペイロード長を 4096 バイトと指定できれば，IPv6 への移行中また移行完了時でも TCP での再送をほとんど抑止することができると考える．

EDNS0 の実装では，最大ペイロード長が大きくなる分サーバやリゾルバのペイロード用バッファに使う記憶領域は増えるが，これらは処理後すぐに解放できるので大きなオーバヘッドにはならないと考える．例えば BIND 9.2.3 では，EDNS0 情報受信の有無にかかわらず，応答処理のための送受信バッファ領域長

(注5)：Windows は Microsoft Corporation の登録商標である．

は一定である．また，EDNS0 の処理に関わる内部記憶構造(注6)の大きさの合計は i386 アーキテクチャでは 60 バイトである．この場合，仮に同時に 10 万件の問合せの状態を保持しておく必要があった場合でも，必要な記憶領域は約 6 メガバイトとなり，現在のハードウェアでは問題はないと考える．

また，EDNS0 によるペイロード長の増加で，IP の MTU 値の制限により，UDP パケットの複数の IP パケットへの分割（fragmentation）が起こる可能性がある．IPv6 では，MTU の最小値を 1280 に定めている（RFC2460 [26] Section 5）．IPv6 の基本ヘッダ（40 バイト，RFC2460 Section 3）と UDP ヘッダ（8 バイト）だけの最小限の構成を考えると，分割されない UDP パケットのペイロード長は，$1280 - 48 = 1232$ バイトとなる．

本論文での最もペイロード長が長くなる A RR に AAAA RR を加えた場合のシミュレーション結果（AR を含む場合）では，ペイロード長が 1232 バイトを超える割合は 2003 年 11 月 28 日に収集したトラヒックをもとにした場合全体の約 0.006%で，2003 年 12 月 16 日の場合は約 0.018%である．応答全体の 3%弱のペイロード長が 512 バイトを超えることを考えると，これらの 512 バイトを超えたパケットのうち，約 99% ではパケット分割が発生しないと予想できる．

5.4 TCP のオーバヘッドと T/TCP による改善

DNS トランスポート層プロトコルで UDP から TCP へ切換を行う際は，TCP の接続と切断時の手順により，1 回のやり取りに最低五つのパケットが発生する．一方，UDP の場合は 1 回当りのパケットは誤り等がなければ二つで済む．

実際には TCP の場合でも，T/TCP [27] を使い同じリゾルバからの 2 回目以降の TCP 接続については 1 回のやり取りのパケットを 3 個に減らすことができる [28]．この手法により，EDNS0 が使えない場合でも TCP 再送の際の負荷を下げることができる．また，T/TCP の利用により，TCP の切断時のタイムアウト時間を約 1/8 に減らすことができる [27]．

T/TCP は，FreeBSD [14] では実行時設定だけで利用できる．また，Linux 上でも利用できる実装が存在する [29]．

T/TCP と通常の TCP を比べた場合，FreeBSD などでは T/TCP のための状態保持用記憶領域が T/TCP を使わない場合も確保されており，T/TCP による記憶領域のオーバヘッドは増えないと考える．また，T/TCP は UDP と違い，パケット分割による再送のオーバヘッドを考慮する必要がなく，かつ伝送内容は信頼できる．**4.2** での TKEY RR のやり取りが TCP を使っているのは，DNSSEC の事前鍵交換には信頼性が要求されるからと考える．

T/TCP は，今後 IPv6 上での DNSSEC のパケットをやり取りした場合，その有用性が高まると考える．EDNS0 だけでは，分割は避けられない可能性が高い．RFC3226 の Section 3 では DNSSEC に対応するサーバとリゾルバには最低 1220 バイト，推奨値として 4000 バイト以上のペイロード長に対応することが要求されている．この場合最低値を 13 バイト超えれば UDP パケットは分割されてしまうため，単純な IPv6 化による増加に比べるとその確率はずっと高くなることが予想できる．

例えばルートゾーンの場合を考えると，RFC2535 Section 5.4 の例では，RR 一つごとに 640 ビット（80 バイト）の署名を含んだ SIG RR が付く．SIG RR のヘッダは，署名者の名前（signer's name）にドメイン名圧縮を使った場合でも最低で 20 バイトとなり，SIG RR の全体長は一つ当り 100 バイトとなる．仮にルートゾーンの 13 のサーバを示す NS RR にそれぞれ SIG RR を付けた場合，SIG RR 全体だけで 1300 バイト増えるため，署名だけで既に IPv6 で分割されない UDP パケットの最大値である 1232 バイトを超えてしまう．

5.5 キャッシュによる EDNS0 非対応サーバやリゾルバの隠ぺい

5.2 で述べた方法に直接対応していないリゾルバやサーバでも，他の対応するサーバやキャッシュによって通信を中継することで，運用上ペイロード長制限の問題を起こさないようにすることができる．

5.6 IP アドレスの選択処理

プロトコル拡張をせずにペイロード長の増加を防ぐために，DNS サーバが多数の RR を応答に含める際，応答の内容によって処理を変える場合がある．例えば同じ名前に対して IPv4 アドレスを示す多数の A RR が付いていた場合，tinydns では上限を設け最大 8 個の A RR をランダムに選んで返す．A RR や AAAA RR の場合，どの IP アドレスへアクセスしても動作が同じことが前提であるため選択処理をしても問題は起きない．ペイロード長を減らす上でこのような処理は

(注6)：`ns_client` 構造体中の関連メンバ `udpsize`，`opt`，及び OPT RR を収める `dns_rdataset` 構造体．

有効である．**4.3** で述べた RFC2181 の推奨する AR の省略等も，この選択処理と同様に，省略可能な RR を送らないことによってペイロード長を減らすための手法といえる．

その一方で，TXT RR など省略できない内容については，内容を間引く方法は使えない．また NS RR に対する additional record として，二つ以上のアドレスを含む RR を返すことが運用上推奨されており [20]，無制限に IP アドレスに関する RR の選択処理を行うことはできない．よってこの方法はあくまで運用上の工夫の域を越えないものと考える．

5.7 ルートゾーンに対する IP アドレス選択処理の適用

5.6 での選択処理は，ルートゾーンを含むすべてのゾーンに対して適用可能である．一例としてルートゾーンへの問合せに対していくつかのサーバに関する RR を選択して返す場合の制限について考える．ここでは各サーバに対応する A RR と AAAA RR が最大で 1 個までで，応答では NS，A，AAAA の各 RR を過不足なく返すものとする．

ルートゾーンへの SOA RR の問合せに対するペイロード長は，まずヘッダに 12 バイト，Question section（問合せの内容）に 5 バイト必要となる．次に authority section として，SOA RR に 75 バイト(注7)，最初の NS RR に 13 バイト(注8)，二つ目からの NS RR はそれぞれ 15 バイト(注9)必要となる．更に，additional section として，応答中の既出のドメイン名を参照して圧縮した上で，各々の A RR に 16 バイト，AAAA RR に 28 バイト必要になる．

これらを総合すると，サーバの個数 n に対して必要なペイロード長の値と 512 バイトの制限を超えない n の最大値は

- A RR のみ：$90 + 31n$, 13
- A RR と AAAA RR: $90 + 59n$, 7
- AAAA RR のみ：$90 + 43n$, 9

となる．この最大値の範囲内で，取捨選択することになる．

このような選択処理を行った場合，選択の仕方に偏りがあれば，特定のサーバへ負荷が集中することになる．均等に負荷を分散させるにはランダムな選択を行うことが必要である．

また，BIND や djbdns などのサーバの実装では，ルートサーバ 13 個の IPv4 アドレスを別途固定値として実行時に参照することが一般的だが，これらの固定値と参照応答のどちらを信用して動作するかという問題が生じる．一つの解として，ルートゾーンの情報がキャッシュにないときには固定値を参照し，あるときにはキャッシュの中の情報を優先するというやり方がある．こうすることで固定値の情報が古くても，ルートサーバから得た新しい情報を使って参照できる．一例として，BIND ではこの解に示す挙動をしている．

なお，ルートゾーンの場合は QNAME 長が 1 バイトと最短だが，一般のゾーンの場合は QNAME 長は最大で 257 バイトであるため，QNAME 長を考慮する必要がある．JPRS の試算 [20] では，`jp` ドメインで同様の計算をした場合，QNAME 長を考慮すると A と AAAA 双方の RR を返した場合のサーバの個数は 3 である．このことより，最大ペイロード長の制限は各ゾーンで応答する必要のある名前の長さにも影響を受けることが分かる．

6. むすび

本論文では，大規模キャンパスネットワークから収集した DNS の実トラヒックに基づいて，IPv4 から IPv6 へ移行する際の RR の変化に伴うペイロード長の増加をシミュレーションによって予測し，現在の制限値 512 バイトを超える UDP ペイロードの割合が AR を含む場合 0.04%以下から 1〜3%へと大きく増えることを示した．また，ペイロード長の増加に際して拡張方式の一つ EDNS0 が有効であり，EDNS0 で最大ペイロード長を 4096 バイトに設定することで IPv6 への移行の際でも DNS トランスポート層プロトコルでの TCP での再送がほとんど行われないようにできることを示した．

DNS ペイロード長の増加は，IPv6 への移行のほかにも DNSSEC や応答に含まれる RR 数そのものの増加など，単なる運用上の工夫だけでは抑えるのが難しい要因を含んでいる．不必要な RR を DNS データベース上に定義するのは運用上は極力控えるべきだが，今後のペイロード長増加傾向を考えると，以下の 3 点の対策を早急にインターネット全域で実施し，DNS トランスポート層プロトコルでの TCP での再送を極力抑止してサーバの負荷を減らすことが必要である．

（1） まず EDNS0 の普及を進めて UDP の最大許容ペイロード長を実用上十分長くする．

(注7)：この中に `a.root-servers.net` という名前が含まれる．
(注8)：SOA RR 中のドメイン名を参照して圧縮．
(注9)：`root-servers.net` という名前を参照して圧縮．

（2） 次に EDNS0 に対応していないサーバやリゾルバに対しては，対応しているサーバやキャッシュによって隠ぺいすることで TCP での再送の可能性を抑える．

（3） 最後に TCP での再送そのものの効率を上げるために T/TCP を DNS に導入する．

今後は，ルートサーバや大規模プロバイダ等での DNS トラヒック測定とそれに基づく追評価 [30]，また EDNS0 や T/TCP の計算量的オーバヘッドの評価などが課題になる．

謝辞 常日ごろ御指導頂く（株）KDDI 研究所の浅見徹代表取締役所長に感謝します．また，本研究に必要な運用ネットワーク上のトラヒック収集解析に際して御協力頂いた ODINS（大阪大学総合情報通信システム）運用本部の皆様に感謝します．

文　献

[1] P.V. Mockapetris, “Domain names — Concepts and facilities,” RFC1034 (also STD13), 1987.
[2] P.V. Mockapetris, “Domain names — Implementation and specification,” RFC1035 (also STD13), 1987.
[3] R. Braden (Ed.), “Requirements for Internet hosts — Application and support,” RFC1123, 1989.
[4] J. Postel, “User datagram protocol,” RFC768 (also STD6), 1980.
[5] J. Postel, “Transmission control protocol,” RFC793 (also STD7), 1981.
[6] P. Vixie, “Extension mechanisms for DNS (EDNS0),” RFC2671, 1999.
[7] R. Elz and R. Bush, “Clarification to the DNS specification,” RFC2181, 1997.
[8] S. Thomson and C. Huitema, “DNS extensions to support IP version 6,” RFC1886, 1995.
[9] M. Crawford and C. Huitema, “DNS extensions to support IPv6 address aggregation and renumbering,” RFC2874, 2000.
[10] R. Austein, “Tradeoffs in domain name system (DNS) support for Internet protocol version 6 (IPv6),” RFC3364, 2002.
[11] R. Bush, “Delegation of IP6.ARPA,” RFC3152, 2000.
[12] D. Eastlake, “Domain name system security extensions,” RFC2535, 1999.
[13] O. Gudmundsson, “DNSSEC and IPv6 A6 aware server/resolver message size requirements,” RFC3226, 2001.
[14] The FreeBSD Project, “FreeBSD.” http://www.freebsd.org/
[15] M. Roesch et al., “Snort.” http://www.snort.org/
[16] TCPDUMP Public Repository, “tcpdump.” http://www.tcpdump.org/
[17] A. Durand and J. Ihren, “DNS IPv6 transport operational guidelines,” 2003. INTERNET-DRAFT draft-ietf-dnsop-ipv6-transport-guidelines-01.txt
[18] P. Vixie, O. Gudmundsson, D. Eastlake, and B. Wellington, “Secure key trancsation authentication for DNS (TSIG),” RFC2845, 2000.
[19] D. Eastlake, “Secret key establishment for DNS (TKEY RR),” RFC2930, 2000.
[20] 日本レジストリサービス，“DNS サーバの最大数について，” 2003. http://jprs.jp/tech/jp-dns-info/2003-07-10-max-number-of-dns-server.html
[21] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, “Dynamic updates in the domain name system (DNS UPDATE),” RFC2136, 1997.
[22] Internet Software Consortium, “BIND.” http://www.isc.org/bind/
[23] NLnet Labs, “Name Server Daemon (NSD).” http://www.nlnetlabs.nl/nsd/
[24] Microsoft Corporation, “Using extension mechanisms for DNS (EDNS0).” http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/sag_DNS_imp_EDNSsupport.asp
[25] D.J. Bernstein, “djbdns.” http://cr.yp.to/djbdns.html
[26] S. Deering and R. Hinden, “Internet protocol, version 6 (IPv6) specification,” RFC2460, 1998.
[27] R. Braden, “T/TCP – TCP extensions for transactions functional specification,” RFC1644, 1994.
[28] K. Rikitake, K. Nakao, H. Nogawa, and S. Shimojo, “T/TCP for DNS: A performance and security analysis,” J. IPSJ, vol.44, no.8, pp.2060–2071, 2003.
[29] L. Ren and J. Zhang, “T/TCP for Linux,” http://ttcplinux.sourceforge.net/
[30] 加藤　朗，関谷勇司，“ISP の DNS サーバの DNS トラヒックの解析，” 信学論（B），vol.J87-B, no.3, pp.327–335, March 2004.

（平成 16 年 1 月 9 日受付，5 月 6 日再受付）

力武　健次　（正員）

技術士（情報工学部門）．1990 東大大学院修士課程了．現在（株）KDDI 研究所セキュリティグループ主任研究員．大阪大学大学院情報科学研究科マルチメディア工学専攻博士後期課程在学中．DNS，インターネットセキュリティ，テレワークの研究に従事．著書に「プロフェッショナルインターネット」（1998 年，オーム社）ほか．第 63 回情報処理学会全国大会大会優秀賞受賞．情報処理学会，ACM，日本テレワーク学会，Internet Society 各会員．

野川　裕記　（正員）

1990 阪大・医卒．医博．2000 より大阪大学サイバーメディアセンター講師．コンピュータセキュリティ，セキュリティマネジメント，社会セキュリティの研究に従事．情報処理学会，日本セキュリティマネジメント学会各会員．

田中　俊昭　（正員）

1986 阪大大学院工学研究科通信工学専攻博士前期課程了．現在（株）KDDI 研究所セキュリティグループリーダー．暗号プロトコル，著作権保護，モバイルセキュリティ，次世代 IDS の研究に従事．情報処理学会会員．

中尾　康二　（正員）

1979 早大・教育・数学卒．現在 KDDI（株）情報セキュリティ部長．早稲田大学，電気通信大学の非常勤講師を兼務．ネットワーク技術，情報セキュリティ技術の研究開発に従事．1987 年度情報処理学会研究賞受賞．情報処理学会会員．

下條　真司　（正員）

1986 阪大大学院基礎工学研究科博士後期課程了．工博．1998 より大阪大学サイバーメディアセンター教授．分散オペレーティングシステムの研究に従事．情報処理学会，IEEE，ACM 各会員．