課程博士公聴会資料

A Study of DNS Transport Protocol for Improving the Reliability
（信頼性向上のための DNSトランスポートプロトコルの研究）

力武 健次

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 応用メディア工学講座

平成16年11月17日

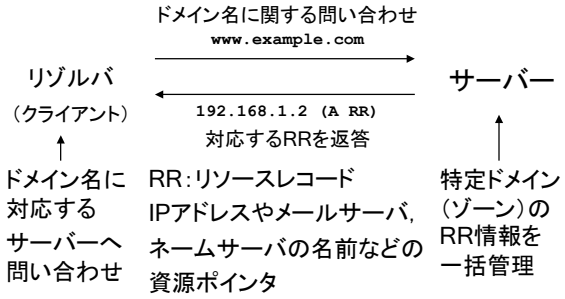
論文内容

第1章

序論

DNSの果たして来た役割

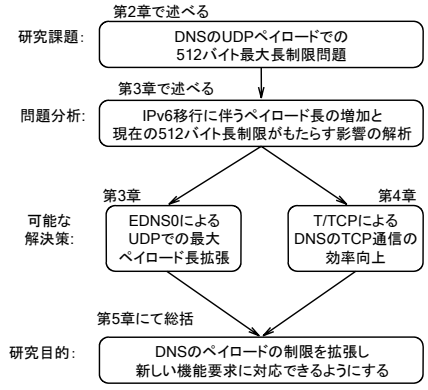
- ドメイン名とリソースレコード（RR）を結合



DNSが抱える現在の課題

- 新しい機能要求でペイロード長が増加
  - IPv4からIPv6への移行
  - DNSSECによるなりすまし防止の認証
  - DNS UPDATEによる動的内容更新
- ペイロード長増加への対応が急務
  - 新機能への対応には512バイトでは不足
    - 現状: UDPペイロードは最大512バイト
    - あまり大きいとUDP fragmentationが問題となる
  - TCP伝送の効率化でサーバ負荷軽減が必要
    - 現状: 512バイト超ではTCPで再度交信を行う

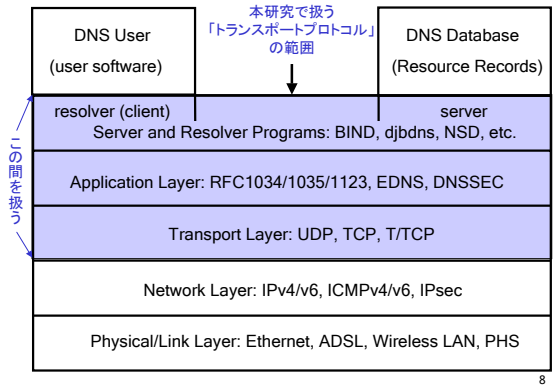
本論文の研究内容



第2章

DNSアーキテクチャと トランスポートプロトコル

DNSの階層図と「トランスポート」の範囲



「DNSトランスポートプロトコル」の役割と課題

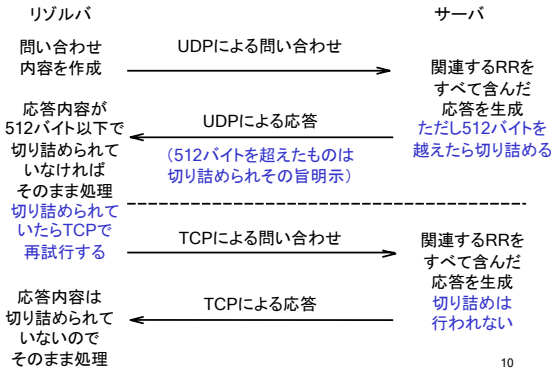
- データベース問い合わせと応答が主
- 回答RRは1つのペイロードにまとめて送る
- 短いペイロードはUDPの1パケットに収める
- (512バイトを超える)長いものはTCPへ
  - 現実の実装では一度UDPで試行後フォールバック
  - そのためTCPでの通信は2度手間になる

→UDPの長さの制限とTCPの効率の悪さが今後の機能拡張への対応のボトルネック

→平均ペイロード長の増加傾向が顕著

→ルートゾーン運用では現実的制約になっている

UDPからTCPへのフォールバック手順



データベース問い合わせと応答のペイロード

QNAME (問い合わせるドメイン名)	問い合わせる 名前と種類 (応答ペイロードにも この内容は含まれる)
QTYPE and QCLASS (問い合わせるRRの型とクラス)	
answer section RRs (サーバが情報を持っていて応答できるもの)	問い合わせへの 応答となるRR群 (回答できる場合は 必要不可欠)
authority section RRs (他の該当情報をもつサーバの参照情報)	
additional section RRs (サーバのアドレスなど上記参照用補助情報)	問い合わせに対する 補助的情報のRR群 (運用上は必須)

ルートゾーン運用での現状の問題点

- ルートゾーンはDNSの根幹
  - .comなど最上位ドメインのサーバ情報を提供
- ルートゾーンはIPv4アドレスで手一杯
  - NS RRを含むペイロードは512バイトが最大
  - 現在は13個のNS RRとA RR(IPv4)のみ
  - ルートサーバの個数はもう増やせない
  - AAAA RR(IPv6)を増やすこともできない

→UDP最大ペイロード長の増加措置が必要

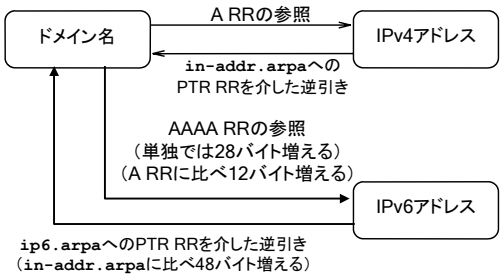
- 増やした場合の問題の定量的評価も必要

第3章

IPv6移行に伴う DNSペイロード長増加に関する 解析と考察

関連論文業績
[1-2] IPv6移行に伴うDNSペイロード長増加に関する解析と考察
[2-1] DNS Transport Size Issues in IPv6 Environment

IPv4からIPv6への移行での変化



本研究での評価手順

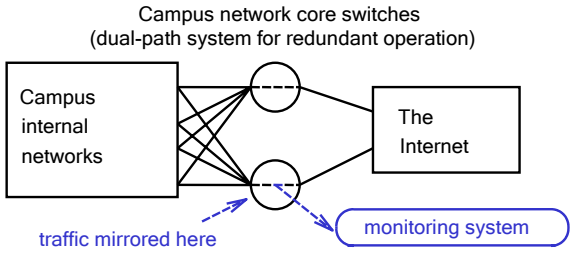
- ODINSの実トラフィックよりDNSのUDPペイロードを抽出
- 以下の仮定をもとにシミュレーションを行った

IPv4のみ 現状 →	IPv4+IPv6 併行期 →	IPv6 移行後
A RR	A RR	
	AAAA RR	AAAA RR
A RR毎の 追加ペイロード長	+28バイト	+12バイト

- 他のRRは出現回数が少ないため無視できる

DNSトラフィック収集用システム構成図

ODINSからのトラフィック収集データを解析
2つのコアスイッチのうち片方だけからミラー

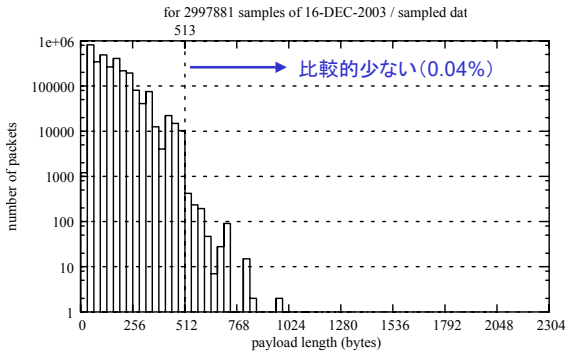


12時間サンプリングの結果

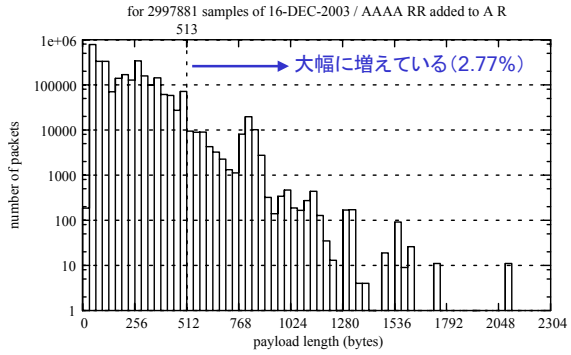
0.04%→2.8%まで大きく増加

6249736 サンプル / 28-NOV-2003 0836JST～				
（単位: バイト）	平均	標準偏差	最大	>512
生データ(AR含む)	108.26	79.68	1192	0.023
AAAA を A に追加	149.01	142.28	3124	2.117
AAAA で A を 置換	125.72	105.98	2020	1.124
2997881 サンプル / 16-DEC-2003 0047JST～				
（単位: バイト）	平均	標準偏差	最大	>512
生データ(AR含む)	137.90	87.16	1112	0.035
AAAA を A に 追加	195.55	155.43	2285	2.772
AAAA で A を 置換	162.60	115.83	1533	1.656

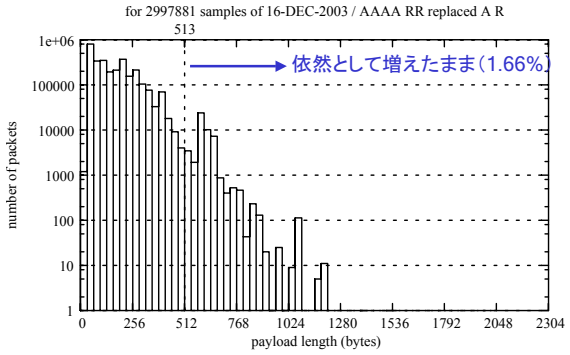
DNS応答のペイロード長別個数分布



AAAA RR を A RR に加えた場合の結果



AAAA RR で A RR を置換した場合の結果



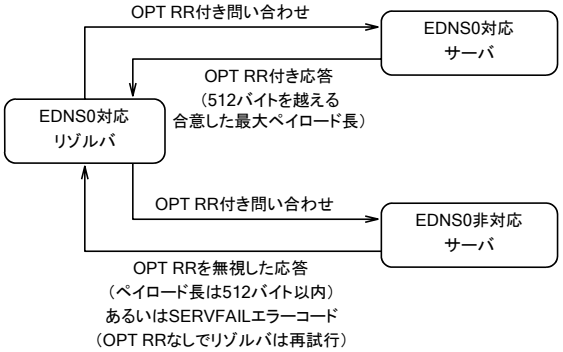
512バイト超ペイロード増加の影響

- TCPペイロードの急増(数十倍に増加)
  - 現在TCPによるDNSパケットの比率は0.07%以下
    (加藤, 関谷「ISPのDNSサーバのDNSトラヒックの解析」, 信学論(B), Vol. J78-B, No. 3, pp. 327-335(2004))
  - 故にTCPペイロードは全体の3%近くに増加する
- 必要な計算資源とネットワーク資源が増加
  - コネクション記憶用領域の増加
  - ネットワークパケット数の増加
- DNSサーバの計算資源を圧迫
  - 特に同時処理数の多いサーバは大きく性能低下
  - インターネット全域の問題だけに影響は甚大

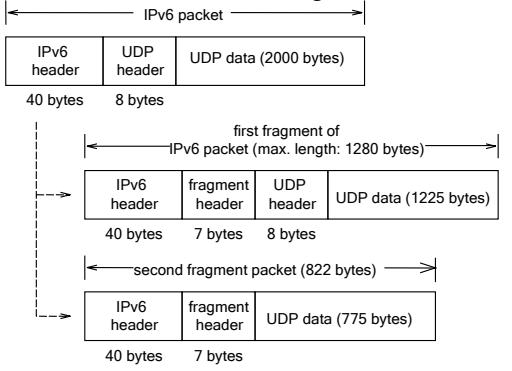
EDNS0によるUDP最大ペイロード長の拡張

- EDNS0(RFC2671)による拡張
  - UDP最大ペイロード長をOPT RRで指定する
  - BINDやNSDでは4096バイトを指定
  - 対応していないサーバは無視して返答またはエラーを返して再試行を促す
- 処理のオーバヘッドには対応可能
  - 必要記憶領域は同時10万件で約6Mバイト
- UDP自身のfragmentationへの考慮
  - IPv6の場合最小構成では最大1232バイト
  - しかし512バイト超応答の99%には対応可

EDNS0のクライアントーサーバ間手順



IPv6におけるUDP fragmentation



DNSSECのためのTCPでのやりとりの効率化

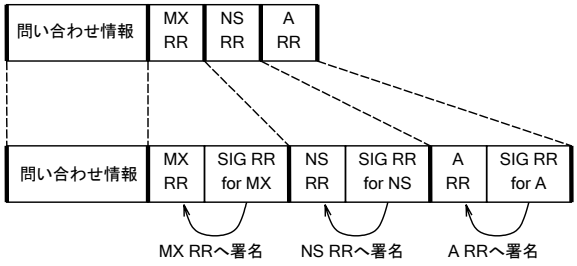
- IPv6とDNSSECの併用にはUDPでは不十分
  - fragmentationが不可避（推奨4000バイト以上）
- DNSSECではTCPをすでに多用
  - TSIG認証のための鍵交換TKEY RRに使用
- TCP自身の効率化にT/TCPが有効（第4章）

→IPv6だけならEDNS0が有効

→DNSSECも併用するならT/TCPの利用がペイロード長増加の対処には有効

DNSSECでのペイロード長の増加

DNSSECでは応答中の各RRにSIG RRの署名が付くため
ペイロード長が増える



第3章の結論

- 実トラフィックのデータに基づく予測により，IPv4からIPv6移行時にDNSの512バイトを超えるUDPペイロード長の割合が0.04%から3%程度へと大きく増えることを示した
- この増加に対処するにはEDNS0が有効
- DNSSECと併用する場合はT/TCPが有効
- 今後の実運用では以下の移行措置が必要
  - EDNS0の早期普及の推進
  - EDNS0非対応サーバやリゾルバの隠蔽
  - T/TCPのDNSへの導入

第4章

T/TCPのDNS応用時における性能とセキュリティに関する解析

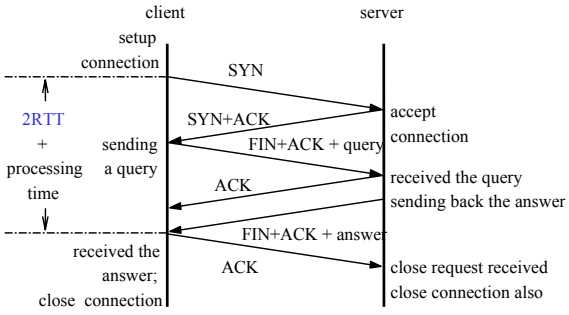
関連論文業績
[1-1] T/TCP for DNS: A Performance and Security Analysis
[3-12] Securing Public DNS Communication

T/TCPの概要

- TCPのプロトコル拡張（RFC1644）
- 送受1往復の通信（transaction）に特化
  - DNSデータベースの問い合わせと応答に適応
- コネクション開始時にデータを相乗り
  - 同一クライアントーサーバ組間のパケット数を削減
- CC (Connection Count) オプションの採用
  - 特定ホスト組間でのSYNの重複を防ぐ
  - CCの単調増加によりDoS攻撃に対する耐性を確保
- TIME_WAIT時の待ち時間短縮
  - 60秒から約8秒 (2MSL → 8RTO)

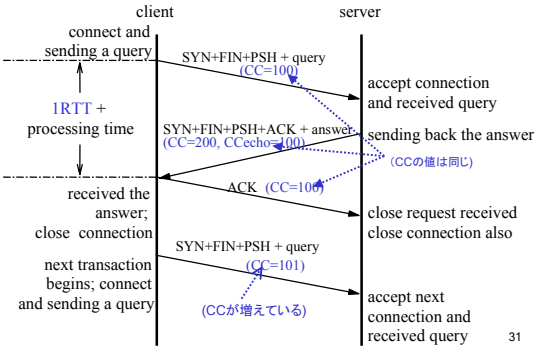
従来のTCPでのやり取り

データの1回のやり取りに2RTT以上の時間を必要とした



T/TCPでのやり取り

SYNとACKパケットにデータを相乗りさせることで
1RTTでの処理を可能にした



DNSへのT/TCP導入とその副次的効果

- UNIX機で簡単に実現できる
  - FreeBSDは設定だけで使用可
  - Linuxには実装が存在
- プログラミング変更は簡単（以下の2点のみ）
  - `setsockopt()`に`TCP_NOPUSH`オプション
  - `sendto()`に`MSG_EOF`フラグをつけて送信
- TCPだけにできればファイヤウォール設定が楽
  - DNSがUDPをやめれば運用上はTCPだけでOK
- 他のTCPコネクションの立ち上がりも速くなる

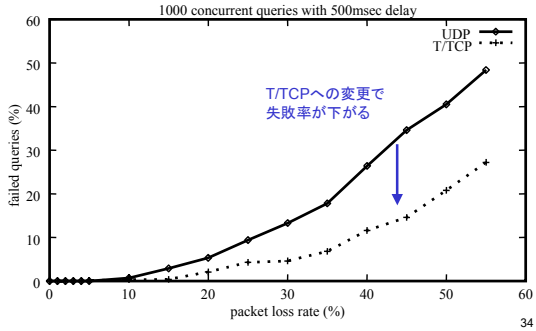
T/TCP / UDP / 従来のTCPの処理時間比較

1000回の連続問い合わせによる比較
（総経過時間，単位: sec，`djbdns`に実装）

	local	Ether	ADSL
RTT(ms)	約0.04	約0.4	60～70
UDP	0.22	2.40	67.77
T/TCP	0.52	8.70	74.70
TCP	0.53	8.93	138.80

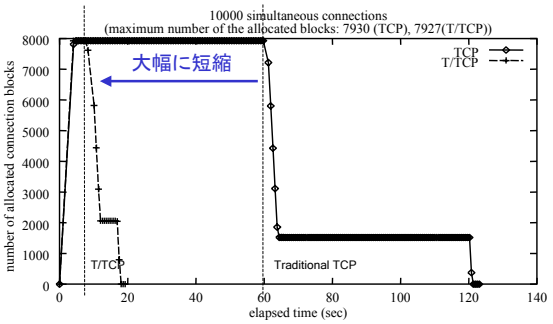
高パケットロス時のUDPとT/TCPの比較

パケットロス率が非常に高い環境においては
問い合わせの成功率でT/TCPがTCPと同じ特性を示しUDPより有利



T/TCPによるTIME_WAIT時間の短縮

従来のTCPに比べT/TCPでは
TIME_WAIT時間が短くなることで記憶領域の占有時間が短くなる



T/TCPのUDPや従来のTCPとの比較

- 低遅延回線ではパケット数の差が効いてくる
  - UDP : T/TCP (TCP) = 2 : 5
- 遅延が大きくなるとRTTが効いてくる
  - T/TCP: UDPの10%増程度
  - 従来のTCP: UDPの2倍かかる
  - 故にT/TCP: 従来のTCPの1/2強で済む

→高遅延回線でT/TCPはUDP並に速い

- T/TCPは再送処理のため高パケットロスに強い
- TIME_WAITの短縮で記憶領域の解放も速い

第4章の結論

- T/TCPをDNSトランスポートに使うための検討と**djbdns**上の実装を通じて性能試験を行った
- 以下の点で従来のTCPに比べ性能向上を確認
  - 従来のTCPに比べ問合せあたりの処理時間の減少
  - 高パケットロス時での問合せ成功率の向上
- 今後応用可能な分野
  - モバイル機器からのDNS参照
  - UDPを通さないファイアウォールでのDNS参照
  - TCPによるDNS参照の全面置き換え

第5章

結論

本研究の成果

- DNSトランスポートプロトコルの実装技術について今後の拡張方向とその有効性を明確にした
- IPv6移行の際現状の実害とその解決法を示した
  - 単純なIPv6化ではEDNS0が有効
  - DNSSEC併用の場合はT/TCPによる効率化が必要
- T/TCPのDNSトランスポートへの有効性を示した
  - 遅延の大きい環境でもUDPに対して性能で遜色はなく，従来のTCPより処理時間を短縮することに成功
  - パケットロスの大きい環境ではUDPよりも通信の成功率を高く維持できることを示した

今後の課題

- 大規模サーバでの提案技術の実証実験
  - ルートサーバや大規模レンタルサーバ等での検証
  - プロトコルシミュレータによる負荷予測とその検証
- より攻撃に強いプロトコルの応用を検討
  - SCTP（通信開始前のcookie認証が有効）
  - TCPの認証強化（MD5チェックサムなど）
- DNSSECによるセキュリティ強化への対応
  - ペイロード長増加の定量的評価
  - 増大したペイロード伝送に耐えるプロトコルの策定

［以上］