2011/04/20 DNSOPS.JP BoF

Simplified DNS Query under IPv4/IPv6 Mixed Environment

IPv4/IPv6 混在通信環境における適切なDNS名前解決方式

○ 北村　浩（ＮＥＣ/電通大）
阿多信吾（阪市大）

発表の概略と目的

- 現在 下記のInternet-Draft を発行し、
  IETF DNSEXT WG で発表審議中です。

Simplified DNS Query under IPv4/IPv6 Mixed Environment
<draft-kitamura-ipv6-simple-dns-query-01.txt>

- この内容を紹介させて頂き、
  参加者の皆様に議論頂きコメントなどを頂きたい。

背景 / 動機

IPv6導入が進むこれからは:

- 一つの通信ノードに対し
  IPv4 及びIPv6 両方のアドレスが設定される
  つまり、種類の異なる複数アドレスが設定される

- ドメイン名（ホスト名）とIPアドレスの対応付けを行っている
  DNSへの情報登録の観点から見てみると
- (ノードを識別する) 一つのドメイン名に対して
  - Aレコード情報（IPv4アドレスに対応）
  - AAAA レコード情報（IPv6アドレスに対応）

  種類の異なる複数の情報が対応つけて登録される

従来のIPv4 1種類だけの時代の技術や機能だけでは適切に対処できない

現状のDNS名前解決処理方式は、
新機能を導入しIPv4/IPv6混在環境に対応しているものの、
その対応は十分ではなく非効率で、多くの問題が発生する方式である

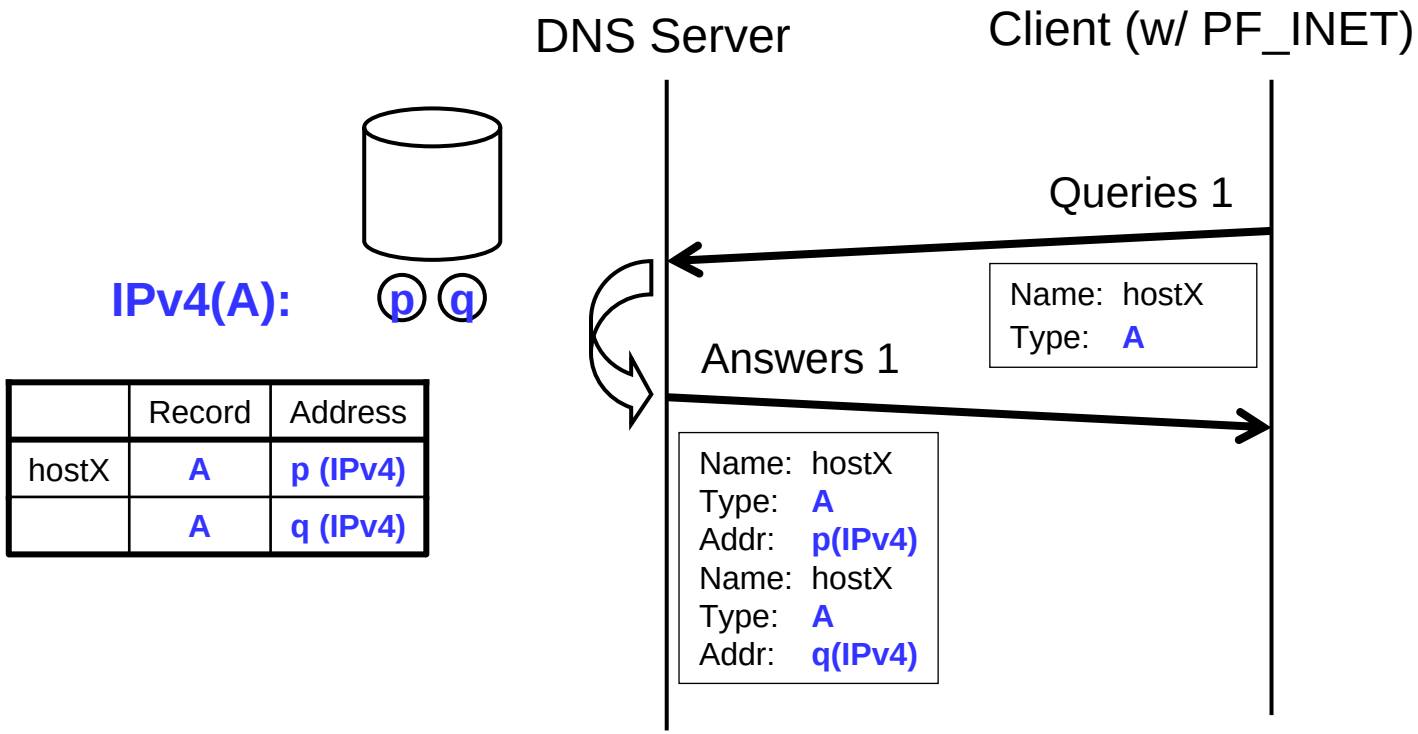
DNS名前解決に期待するもう一つの側面：IPv6の導入環境の改善

- 一般のユーザは ドメイン名を引数情報として、通信相手を指定する
  IPv4及びIPv6アドレスの違いは強く意識する必要がない
- IPv4/IPv6 混在環境にあって
  ドメイン名情報はアドレスの種類の違いに関係のない**唯一の情報**
- ドメイン名情報を介在して異なる種類のアドレスに関連付けられている
  これらの特性はIPv6の円滑な導入に大いに役立つ

DNS名前解決処理は:
（IPアドレスの種類に依存しない）ドメイン名情報⇒IPアドレスを取得
IPv6の導入を容易に行えるかどうかを左右する重要な処理

ねらい：
IPv4/IPv6 混在通信環境での、現状のDNS名前解決処理の課題の明確化
IPv6の導入環境を改善することができる、
効率的な新しいDNS名前解決処理方式の提案

IPv4単独環境におけるDNS名前解決方式

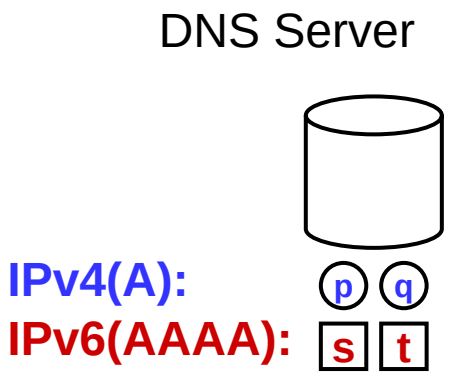


	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)

一つのドメイン名 hostX に対して、
二つの IPv4アドレス(p,q) が 二つのAレコードのエントリとして登録
典型的な状態でのDNS名前解決処理

極めて自然な方法: 最小のセットであるQuery/Answerの
一対メッセージだけで二つのエントリ情報が同時に一度で取得される

IPv4/IPv6 混在環境での DNSへの情報登録の典型的な状態



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

一つのドメイン名 **hostX** に対して:

- 二つの **IPv4** アドレス (**p, q**) が
  二つの **A** record エントリとして登録

加えて

- 二つの **IPv6** アドレス (**s, t**) が
  二つの **AAAA** record エントリとして登録

Aレコード（IPv4に対応）と**AAAAレコード**（IPv6に対応）
異なる**2種類**のレコードが存在する
一種類のIPv4だけの場合と比べると複雑な処理になる

IPv6導入の黎明期に考えられた
現状の IPv4/IPv6混在環境での
DNS名前解決処理の仕様を決定付けた要件

1. 既に実装があり動作している
   IPv4だけのDNS名前解決処理に影響を与えないこと
   （IPv4用のDNS名前解決処理パケットとは
   独立したパケットを用いて動作すること）
2. DNS Server 側が IPv6のことを知らず、
   IPv6に対応していない環境でも正しく動作すること
3. 一つのドメイン名に対して、
   AレコードとAAAAレコード両方が取得できる場合は、
   AAAAレコード情報が優先されて使われること
   （でないと、後発のIPv6が選ばれて通信することがないため)

要件を満たし、現状 使われている IPv4/IPv6 混在環境でのDNS名前解決処理方式

- **2対**の独立した Query/Answerメッセージ
  1. **A**レコードの情報取得用
  2. **AAAA**レコード情報取得用

  によって実現される

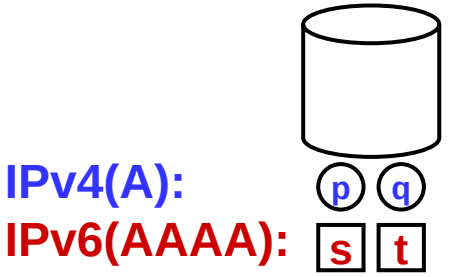
クライアント アプリケーションが名前解決のために呼び出す関数は**1つ**（getaddrinfo(PF_UNSPECを指定)）**だが**、**2対**の独立した Query/Answer**メッセージ**が発行される

4種類の名前解決メッセージ発行の型

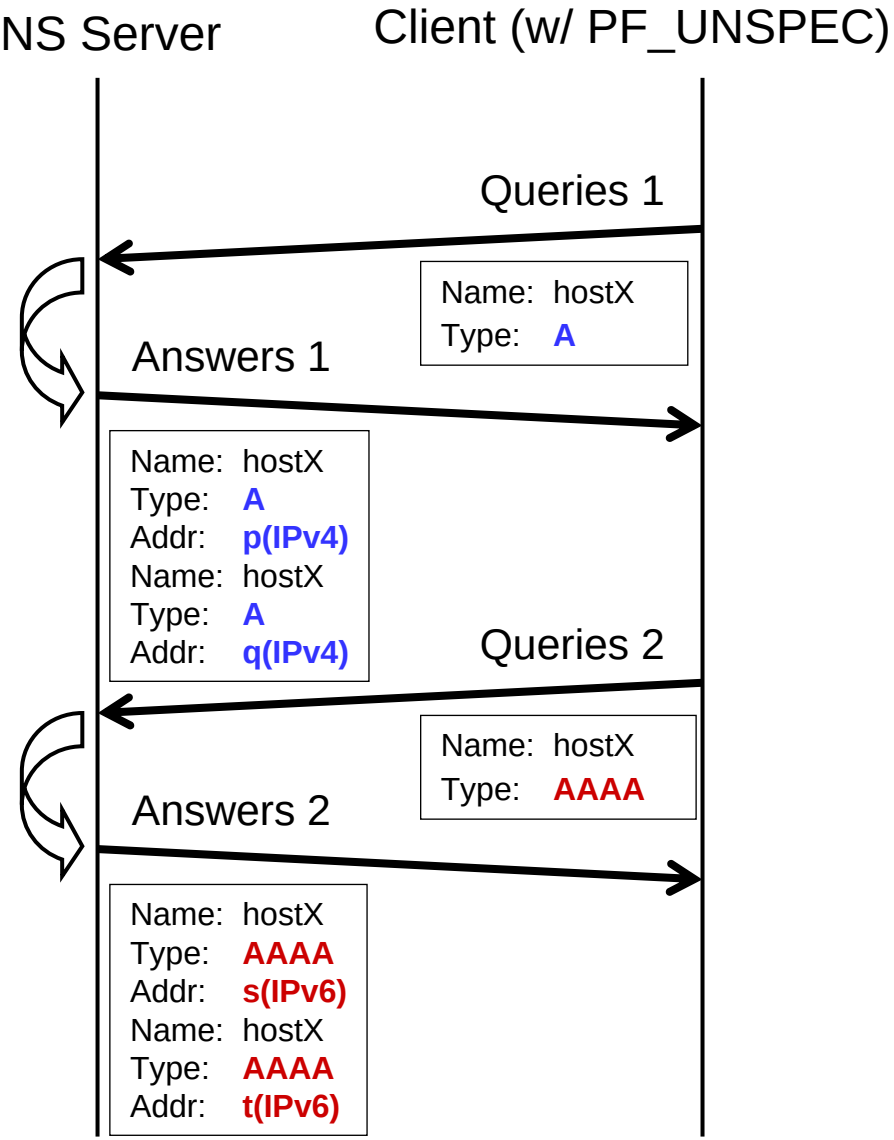
2対のメッセージなので、
発行の順番、発行方法（直列or並列）で4種類の型がある

Type 名	1st Query	2nd Query	Serial / Parallel	知られている実装など
4-**6** Serial	for **A** record	for **AAAA** record	Serial	Windows Vista/7 FreeBSD
6-**4** Serial	for **AAAA** record	for **A** record	Serial	Windows XP RFC4472 shows:
4-**6** Parallel	for **A** record	for **AAAA** record	Parallel	Some Linux
6-**4** Parallel	for **AAAA** record	for **A** record	Parallel	

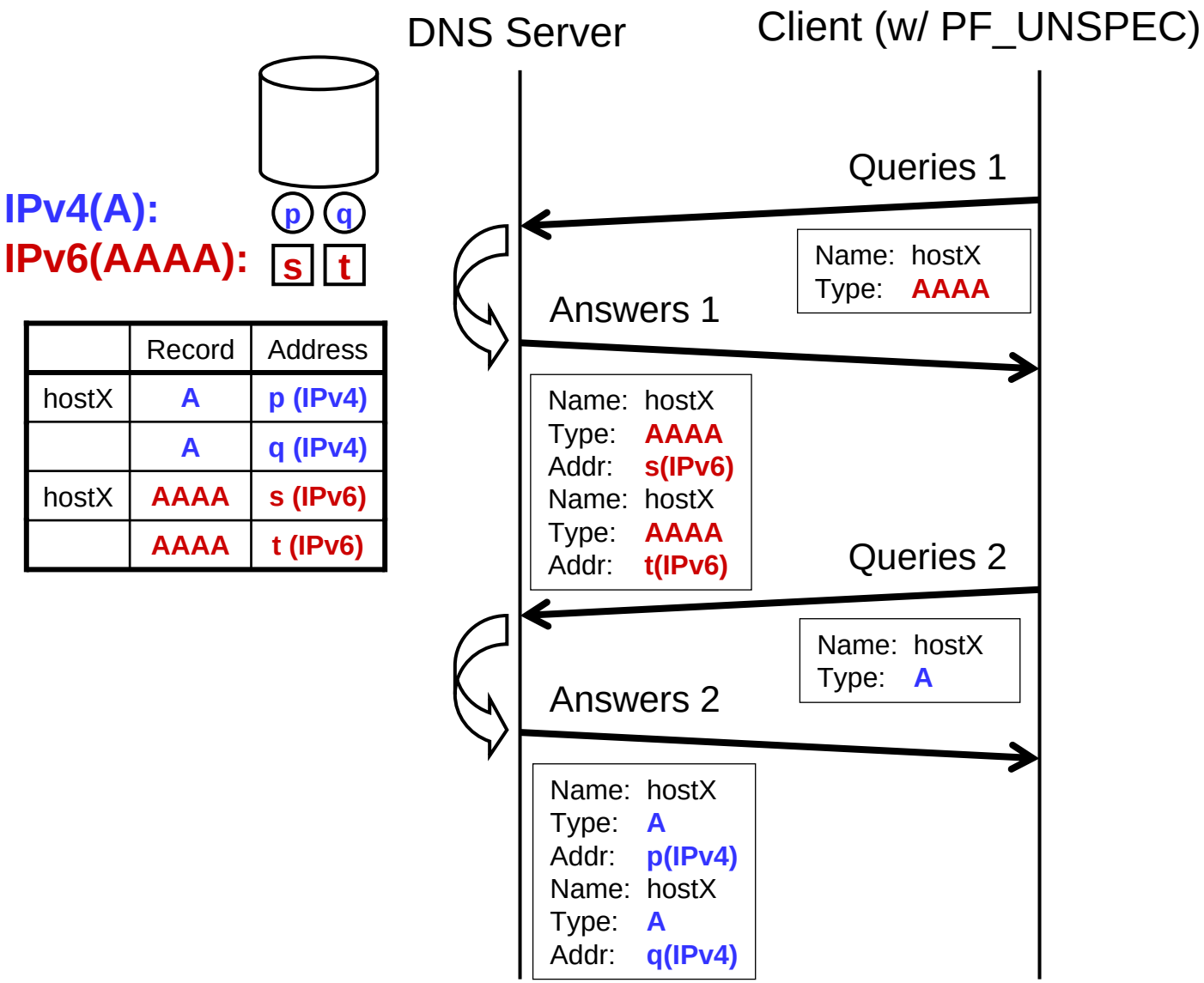
4-6 (A first) Serial Type



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

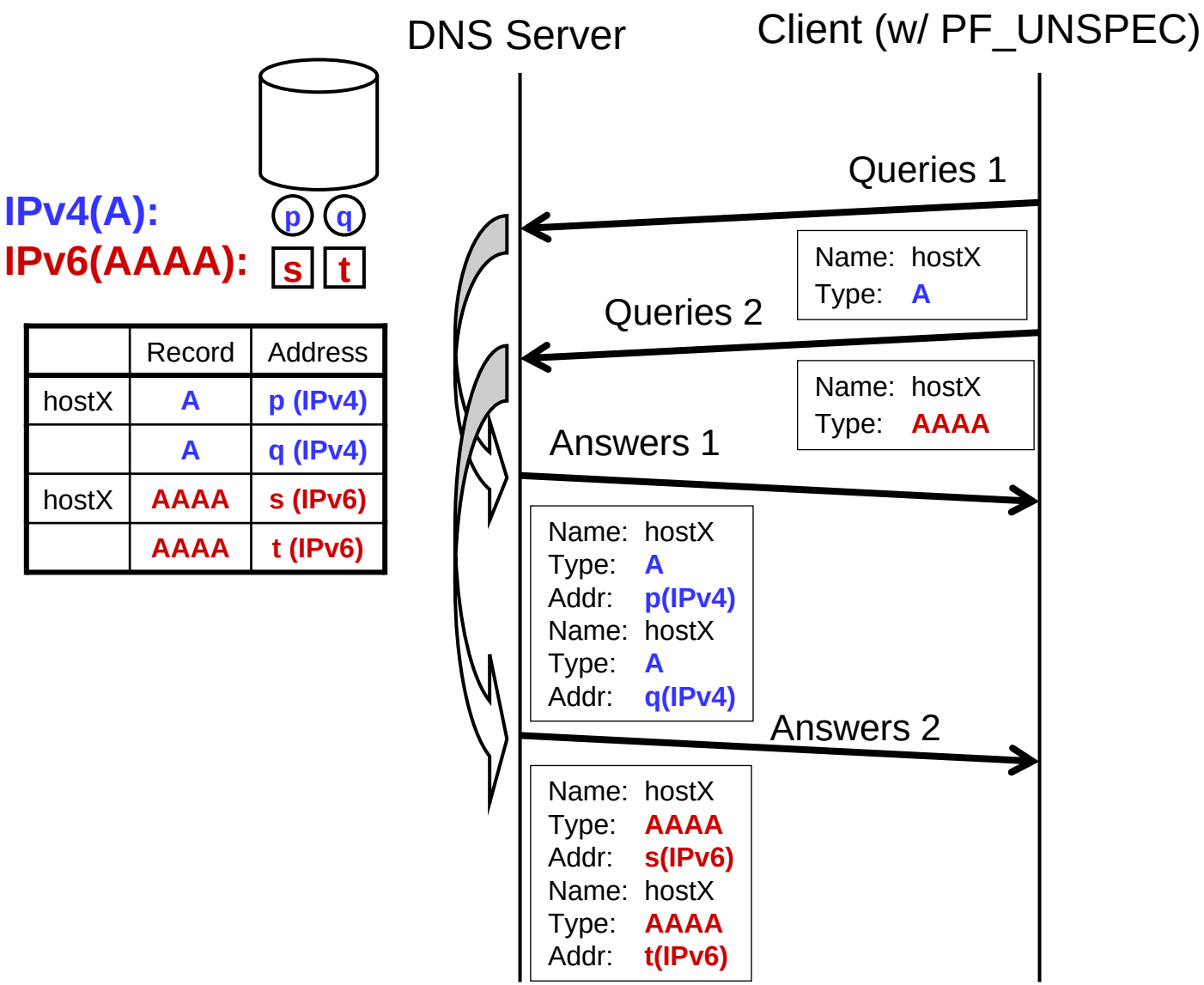


6-4 (AAAA first) Serial Type



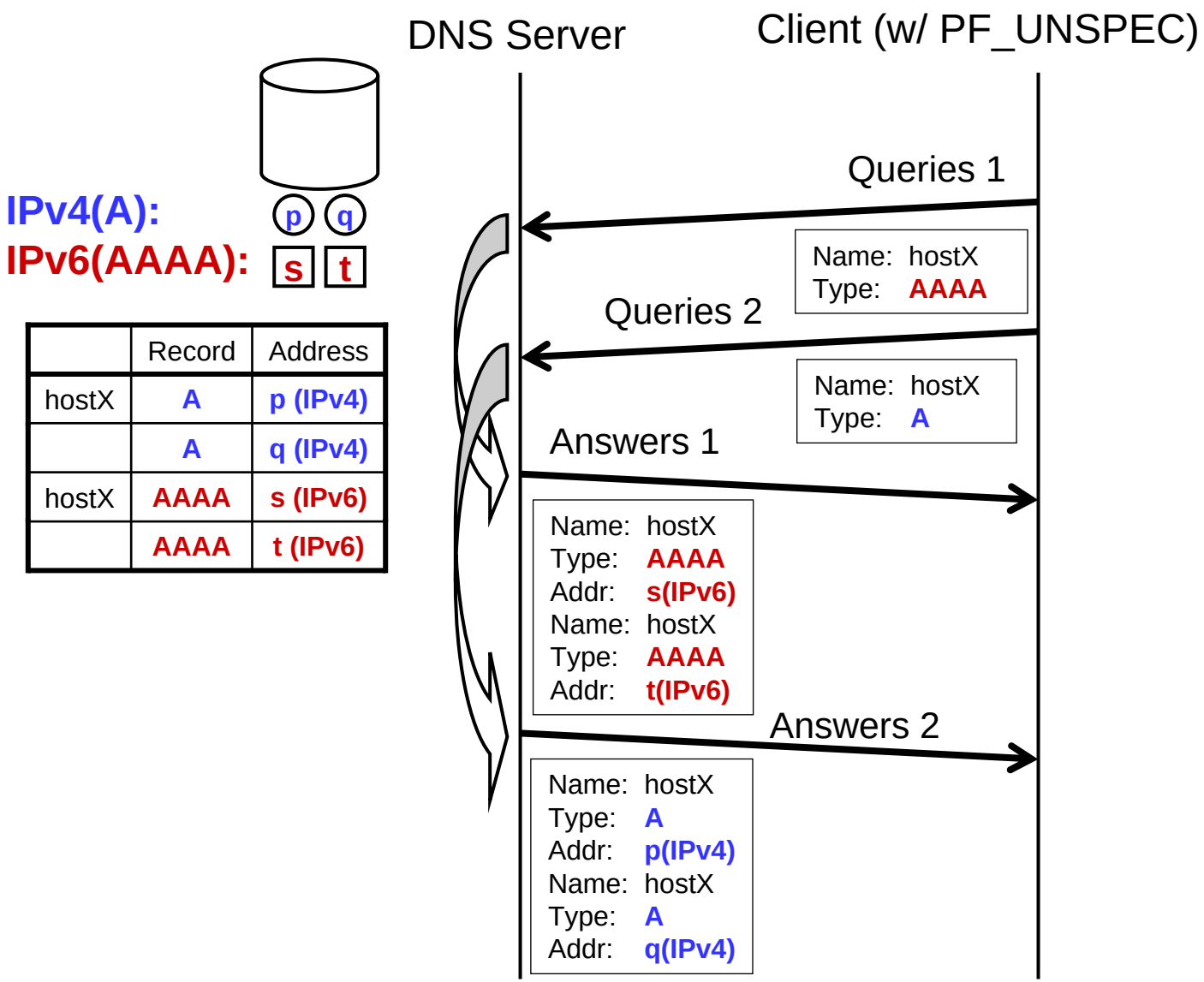
	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

4-6 (A first) Parallel Type



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

6-4 (AAAA first) Parallel Type



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

現状の2対のメッセージ方式のよる DNS名前解決処理方式の課題

DNS名前解決のためにユーザ アプリケーションが呼び出す関数は1つなのに、
実体は 2対の独立したQuery/Answer のメッセージを発行し、
1つのドメイン名に対応する情報を2回に分けて取得している
そのため、以下に示すような課題がある

1. **複雑である　単純な処理手順になっていない**
   仮に一方のメッセージが途中失われた場合、
   その失われ方と回復をどう行うかなどは非常に多様で複雑である
2. **非効率である**
   1対のメッセージ処理の場合が、単純で最大効率な方法あることは明確で
   2対のメッセージ処理方式を採用すれば、当然非効率になる
3. **処理を完了し結果を得るまでに時間がかかる**
   一つ目のメッセージ処理結果を待って、二つ目のメッセージの処理を行うため、1対のメッセージの場合に比べ長い時間を要する
4. **2倍の量のトランザクションになる**
   1対のメッセージに比べれば2倍の量のトランザクションになる

DNS名前解決処理方式の観点から見る IPv6導入の黎明期当時と現在の状況との違い

1. **DNSサーバ側のIPv6対応が進んだ**
   既に多くのDNSサーバはIPv6に対応しており、
   　　非対応のDNSサーバの数はわずかであると考えてよい
   IPv6非対応 DNSサーバのことを重視して対応する必要性が下がった。
2. **DNS名前解決処理を発行するクライアント側のIPv6対応も進んだ**
   メジャーなクライアントOSのIPv6化が進み、ほとんどのクライアントが
   IPv6に対応したDNS名前解決処理を行うことができる。
3. **非効率で問題を起こす名前解決処理は無視できなくなってきている**
   IPv6対応ノードの数が少ない頃は、影響の少ないマイナーなことであると
   軽視されていたが、現在では無視できなくなっている

   更に、こういった状況を放置することは、IPv6の導入促進あるいは、今後
   のインターネットの健全は発展を阻害する要因になるかもしれないと思わ
   れている

提案する解決策:
1対のQuery/Answer のメッセージだけで
DNS名前解決処理を行う新しい方式

DNS名前解決処理を（2対メッセージで行うのではなく）
1対のQuery/Answer のメッセージだけで行う
1つのドメイン名に対応付けられている
全て(IPv4及びIPv6)のアドレス情報（A及びAAAAレコード）を
1回で取得できる方法を用いる

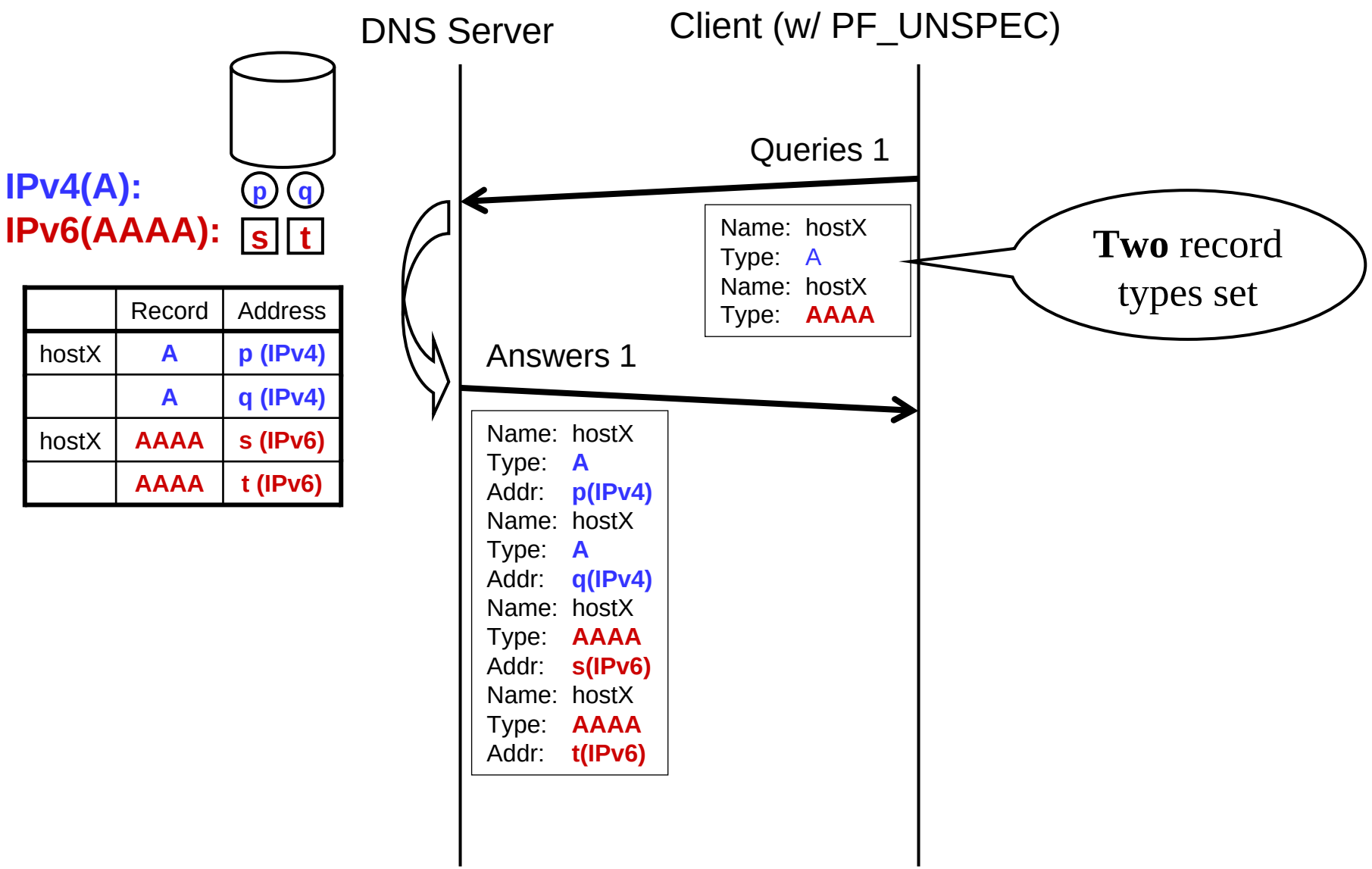
3種類の実現方法が考えられる

1.Two Existing Records Combined 型

2.One New Special Record 型

3.One Existing Record w/ Mapped Transformation 型

Two Existing Records Combined 型 (1/2)



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

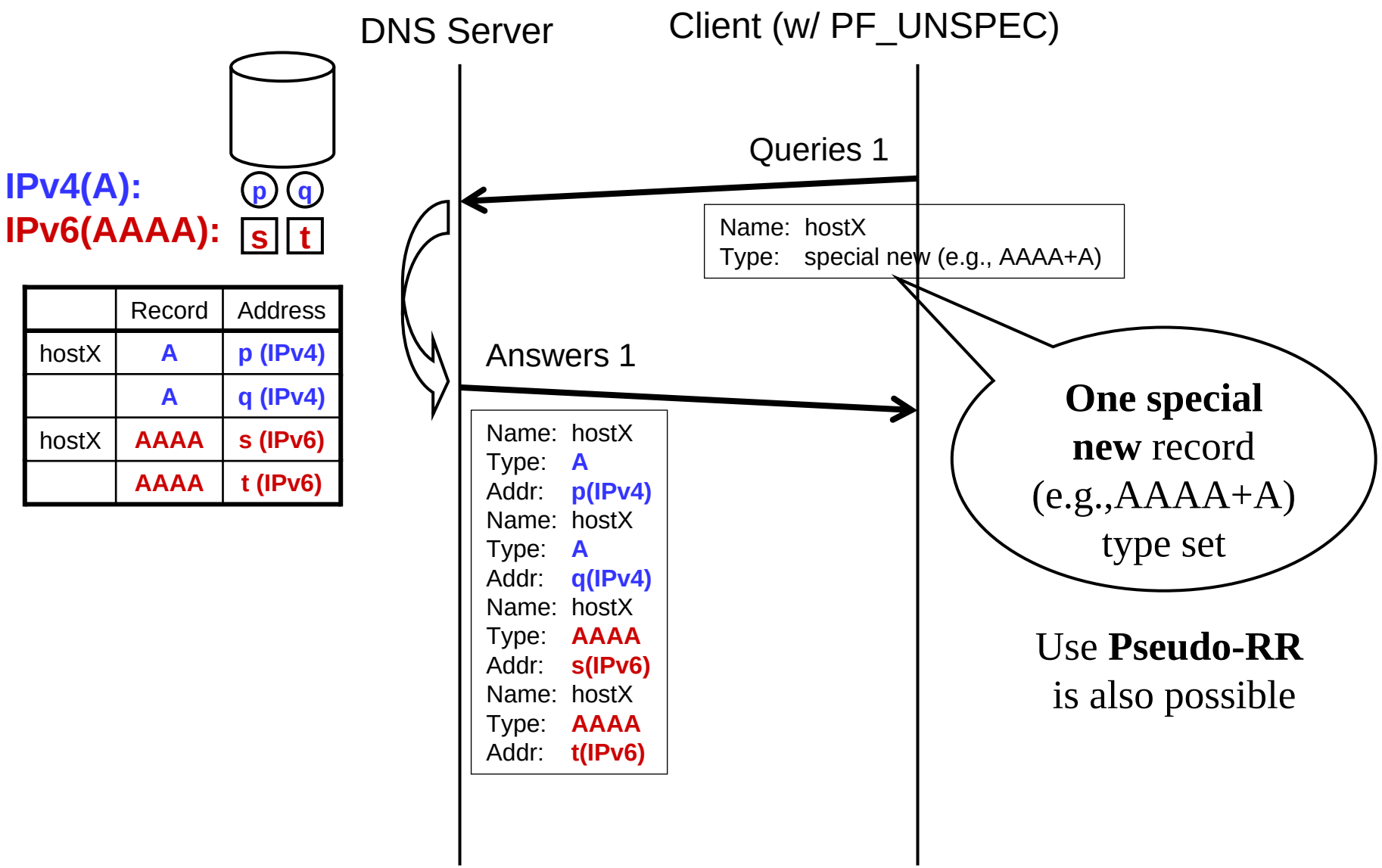
Two Existing Records Combined 型 (2/2)

既存の2つレコード型（AとAAAA）を**一緒にまとめて 1つのQueryに設定**する

- **長所:**
  - 既存レコードタイプ用いるので、新しいレコードタイプといった機能拡張が不要
  - DNS 名前解決のパケットフォーマットとしても、
    設計段階で汎用的な対応である複数のレコードを一緒にまとめて設定するということが可能なフォーマットを採用している
- **短所:**
  - クライアント側の名前解決機能の実装をこの新しい方式に対応したものに更新する必要がある
  - クライアント側の更新であり、膨大な数でもあり、実現は容易なことではない
  - パケットフォーマットの設計思想として、複数のレコードを一緒にまとめて設定することが可能になっているといえども、
    現状 1 Query に複数のレコードが設定されているような例は知られておらず、
    1 Queryには1レコードしか設定しないというのが実質的な決まりごとになっており、論理的に可能だけど実績となる前例はないという状態

すぐにではなく、長期的な視点で物事を見たらということになるが、
理想的にはどうすべきかを求めた方式であるともいえる。

One New Special Record 型 (1/2)



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

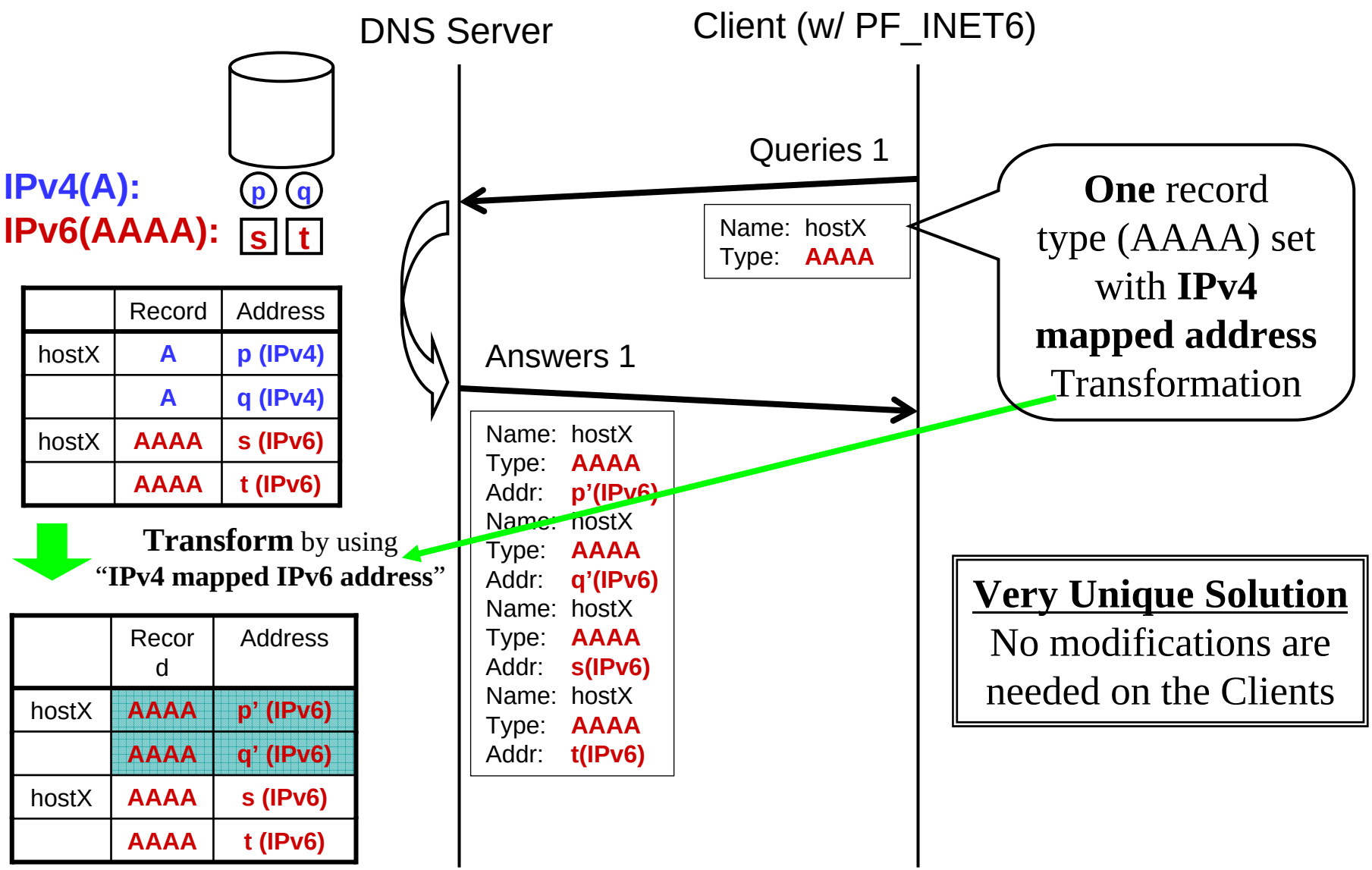
One New Special Record 型 (2/2)

既存にはない**新しい特別な1つのレコード型**
（例えば、AAAA+A）を導入し **1つのQueryに設定**する

- **長所:**
  - 1Queryには1レコードしか設定しないという実質的な決まりごとに従っており、前述のTwo Existing Records Combined 型より改善されている
- **短所:**
  - 現状の仕様にない新しいレコード型を導入しなければならないこと
  - Two Existing Records Combined 型と同様、クライアント側の名前解決機能の実装をこの新しい方式に対応したものに更新する必要がある
  - これは実用的な普及を考えた場合に越えなくてはならない大きな問題である

この方式も、長期的な視点で物事を見たらということになるが、
理想的にはどうすべきかを求めた方式であるともいえる。

One Existing Record w/ Mapped Trans.型 (1/2)



	Record	Address
hostX	A	p (IPv4)
	A	q (IPv4)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

	Record	Address
hostX	AAAA	p' (IPv6)
	AAAA	q' (IPv6)
hostX	AAAA	s (IPv6)
	AAAA	t (IPv6)

One Existing Record w/ Mapped Trans.型 (2/2)

既存の1つのレコード型(AAAA)を **1つのQueryに設定**する
更に **IPv4 mapped IPv6 address** の**特性をうまく利用**した**変換を加える**
今までの考え方を大きく変える、**独創的な方式**
IPv4アドレスも IPv4 mapped IPv6 アドレスのAAAAレコードとして扱う
全てのアドレスはIPv6 アドレスであり AAAAレコードだけで事足りると考える
（アプリケーションの中でも IPv4アドレスを Mapped IPv6 アドレスとして扱う）

- **長所:**
  - 既存のレコード型 AAAA を使うため、新しいレコード型を導入する必要がない。
  - 1 Queryには1レコードしか設定しないという実質的な決まりごとにも従っている。
  - 最大の長所は上述した二つの方式と異なり、クライアント側の名前解決機能の実装は既存のままでよく、新しい方式に対応したものに更新する必要がないこと
- **短所:**
  - いささかトリッキーかなと思われる点を除けば**基本的には問題はない**
  - IPv4 mapped IPv6 アドレスという機能の実装に依っているのでこの機能が有効でないクライアントではうまく動作しない
  - クライアント側の機能を更新しなくて良いため、サーバ側でやるべきことがある（これは短所といえなくもないが、新しい機能を導入するためにどこかの修正が必要であり、膨大で多様な実装があるクライアント側でなくサーバ側で実現できることは、短所と呼ぶより長所なのかもしれない）

1対のQuery/Answer のメッセージだけで実現する 提案する 新しいDNS名前解決処理方式のまとめ

- **長期的な視点、理想的なDNS名前解決処理方式を選ぶとすれば、**
  - **Two Existing Records Combined 型**
  - **One New Special Record 型**

  いずれもその有力な候補になる良い方式になる

  しかし、
  クライアントにおける実装を新しい方式へ更新するという
  実用的な普及を考えた場合に越えなくてはならない大きな問題がある

- **実用的な普及を重視するとすれば、**
  One Existing Record with Mapped Transformation 型は、クライアント実装の変更が不要であり、独創的な概念を理解するには時間がかかるかもしれないが、致命的に不都合になることはなく、有力な方式になる

皆さんのご意見をお聞かせ下さい。