2012-08-31　DNS Summer Days 2012

DNSのRFCの歩き方

株式会社ハートビーツ　滝澤隆史
日本Unboundユーザー会

私は誰

- 氏名: 滝澤 隆史 @ttkzw
- 所属: 株式会社ハートビーツ

  - サーバの構築・運用や
    24時間365日の有人監視をやっている会社
  - いわゆるMSP（マネージド サービス プロバイダ）
- DNSとの関わり
  - システム管理者として1997年から2006年までネームサーバの運用
    - BIND4, BIND8, djbdns, BIND9
  - 現在は個人サーバでネームサーバを運用
    - NSD, Unbound
  - 日本Unboundユーザー会
    - Unbound/NSDの文書の翻訳
  - DNSは趣味です(ｷﾘｯ

このセッションの目的

- DNSの仕様を調べるための取っかかりになるような情報を提供すること。

このセッションの概要

- DNSの概念や仕様を定めている
  - RFC 1034 DOMAIN NAMES – CONCEPTS AND FACILITIES（ドメイン名 – 概念と機能）と
  - RFC 1035 DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION（ドメイン名 – 実装と仕様）
- と、後に発行されたRFCで変更・修正された内容および拡張された仕様の一部について説明する。

注意点

- この資料ではRFCの文書を一部翻訳していますが、おおざっぱな訳なので、後で自身で原文を当たってください。
- 話者は英語が得意ではありません。発音は酷いのでご容赦ください。
- 今回は拡張された機能についてはほとんど説明しません。特にDNSSECについては全く説明しません。DNSSECについてのRFCについてはDNSSECジャパンのサイトを訪問してください。

RFCについての復習

RFCとは

- IETF（Internet Engineering Task Force）により発行されている技術文書
  - インターネット標準や情報提供の文書などがある
- RFCは"Request for Comments"の略
- RFCとは何かを理解するためにはRFCを読むのがよい。

RFCの形式

RFCの番号

この番号のRFCを廃止

著者

Network Working Group
Request for Comments: 4033
Obsoletes: 2535, 3008, 3090, 3445, 3655, 3658,
3755, 3757, 3845
Updates: 1034, 1035, 2136, 2181, 2308, 3225,
3007, 3597, 3226
Category: Standards Track

この番号のRFCを更新

分類

R. Arends
Telematica Instituut
R. Austein
ISC
M. Larson
VeriSign
D. Massey
Colorado State University
S. Rose
NIST
March 2005

タイトル

RFCの発行月

本文

DNS Security Introduction and Requirements

Status of This Memo

This document specifies an Internet standards track protocol for the

RFCの番号

- 発行されたRFCの番号は変わらない。
  - 代わりに、致命的な間違いや編集ミスについては"ERATTA"が別途出ることがあるので注意。
  - 新しいRFCにより"obsoletes"されることがある。
- RFCに慣れてくるとRFCのタイトルではなくRFCの番号で会話をするようになってくる。
  - 「RFC 1034ではこう書かれているけどRFC 2181ではこうだ」

RFCの分類

- RFC 1796 "Not All RFCs are Standards"
  すべてのRFCが標準であるわけではない
  - Infomational （情報提供）
  - Experimental （実験的）
  - Standard Track （標準化過程）
    - Proposed Standard （標準への提唱）
    - Draft Standard （標準への草稿）
    - Internet Standard （インターネット標準）
  - Historic （歴史的）

分類: 標準化過程

- Standard Track （標準化過程）
  - Proposed Standard （標準への提唱）
  - Draft Standard （標準への草稿）
    - RFC 6410（2011年10月発行）により"Internet Standard"に統合
  - Internet Standard （インターネット標準）
    - 「STD xxx」のように別途番号付けされる

分類: 非標準化過程、BCP

- Non-Standards Track （非標準化過程）
  - Experimental （実験的）
    - 研究や開発の成果
  - Infomational （情報提供）
    - インターネットコミュニティーのための情報
  - Historic （歴史的）
    - 新しい仕様に置き換えられ、役割が終わったもの
- Best Current Practice （現時点での最良な方法）
  - 運用についての文書
  - 「BCP xxx」のように別途番号付けされる

要求レベルを示すために用いられるキーワード

- RFCの文書中に次のような説明がある

  The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

- RFC 2119 / BCP 14 "Key words for use in RFCs to Indicate Requirement Levels"
  - RFCにおいて要求レベルを示すために用いられるキーワード

要求レベルを示すために用いられるキーワード

大文字の時にキーワードとして意味を持つ

キーワード	邦訳	意味
MUST	しなければならない	要求事項
REQUIRED	要求される	
SHALL	するものとする	
MUST NOT	してはならない	禁止事項
SHALL NOT	しないものとする	
SHOULD	すべきである	推奨事項 （無視する場合は慎重な判断が必要）
RECOMMENDED	推奨される	
SHOULD NOT	すべきでない	非推奨事項 （容認する場合は慎重な判断が必要）
NOT RECOMMEDED	推奨されない	
MAY	してもよい	任意事項
OPTIONAL	任意である	

RFCに関するサイト

- RFC Editor
  - http://www.rfc-editor.org/
- IETF TOOLS
  - http://tools.ietf.org/html/
  - RFCを追いかけるには非常に便利
- JPRS DNS関連技術情報
  - http://jprs.jp/tech/
  - DNS関連のRFCの邦訳がある
- DNSSECジャパン
  - http://dnssec.jp/
  - DNSSECに関連するRFCの邦訳や技術情報がある

DNSの基本仕様

DNSの基本仕様のこの2つのRFC

- RFC 1034
  - DOMAIN NAMES – CONCEPTS AND FACILITIES
  - ドメイン名 – 概念と機能
- RFC 1035
  - DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION
  - ドメイン名 – 実装と仕様

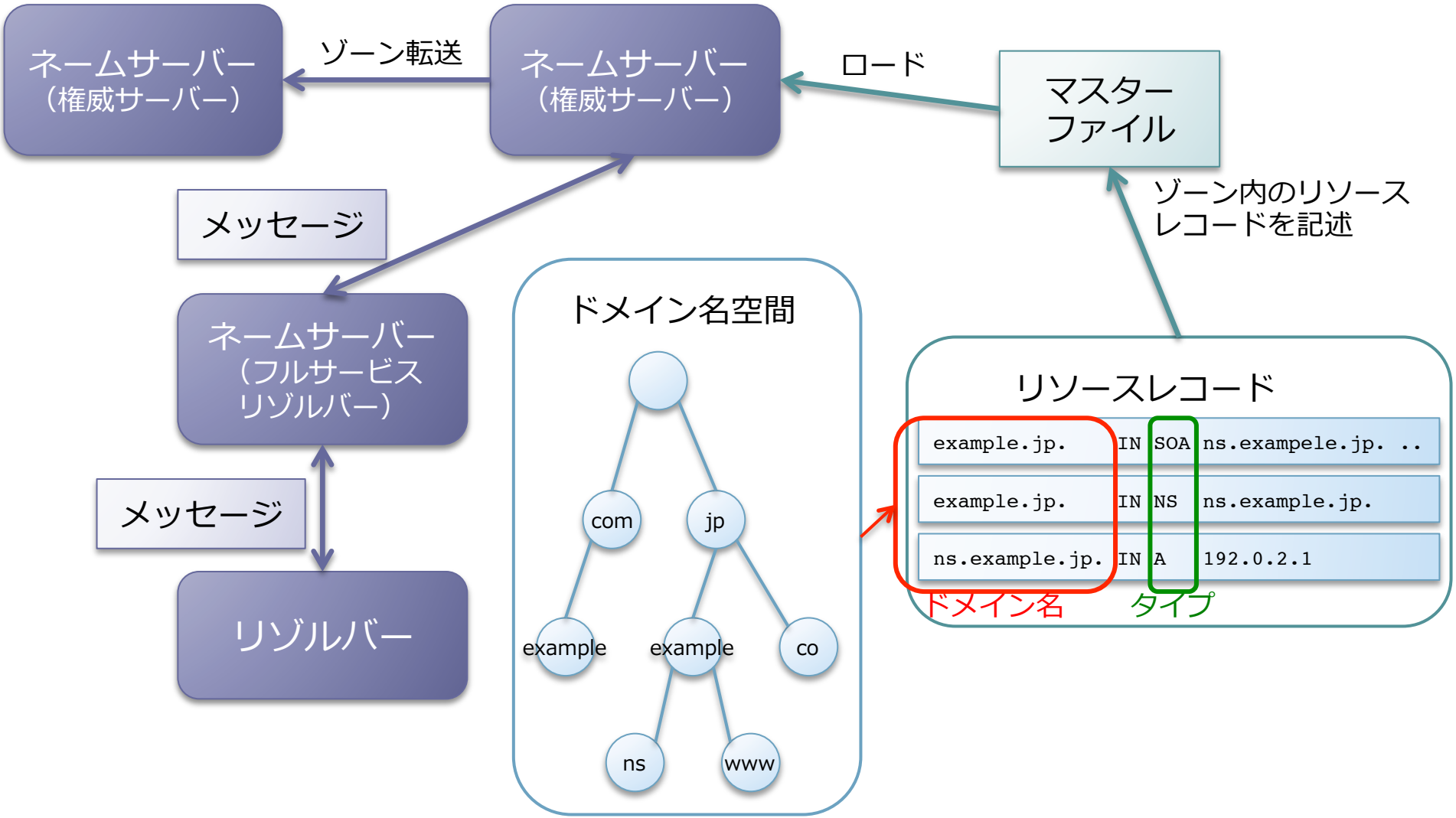
RFC 1034
DOMAIN NAMES - CONCEPTS AND FACILITIES

- タイトル
  - DOMAIN NAMES – CONCEPTS AND FACILITIES
  - ドメイン名 – 概念と機能
- 概要
  - DNSの構成要素の役割や機能についての説明
    - ドメイン名空間とリソースレコード
    - ネームサーバー
    - リゾルバー

RFC 1035
DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

- タイトル
  - DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION
  - ドメイン名 – 実装と仕様
- 概要
  - DNSのプロトコルの仕様
    - ドメイン名空間
    - リソースレコード
    - メッセージ
    - マスターファイル
    - ネームサーバーの実装
    - リゾルバーの実装
    - メールエクスチェンジャ

DNSの構成要素


ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

RFC 1034とRFC 1035への道

- DNSの仕組みをある程度理解していないとRFC 1034とRFC 1035の内容を理解できない。
  - いきなりRFCを読んでもたぶん理解できない。
- 時代背景が異なることを意識すること。
  - DNSが検討開始されたのはARPANETからThe Internetへの過渡期
  - 現在は、IN以外のクラス（CS, CH, HS）は使わない。
    - ただし、CHは本来の用途とは異なり、ネームサーバーの情報の取得に使われている。
      - $ dig TXT CH version.bind.

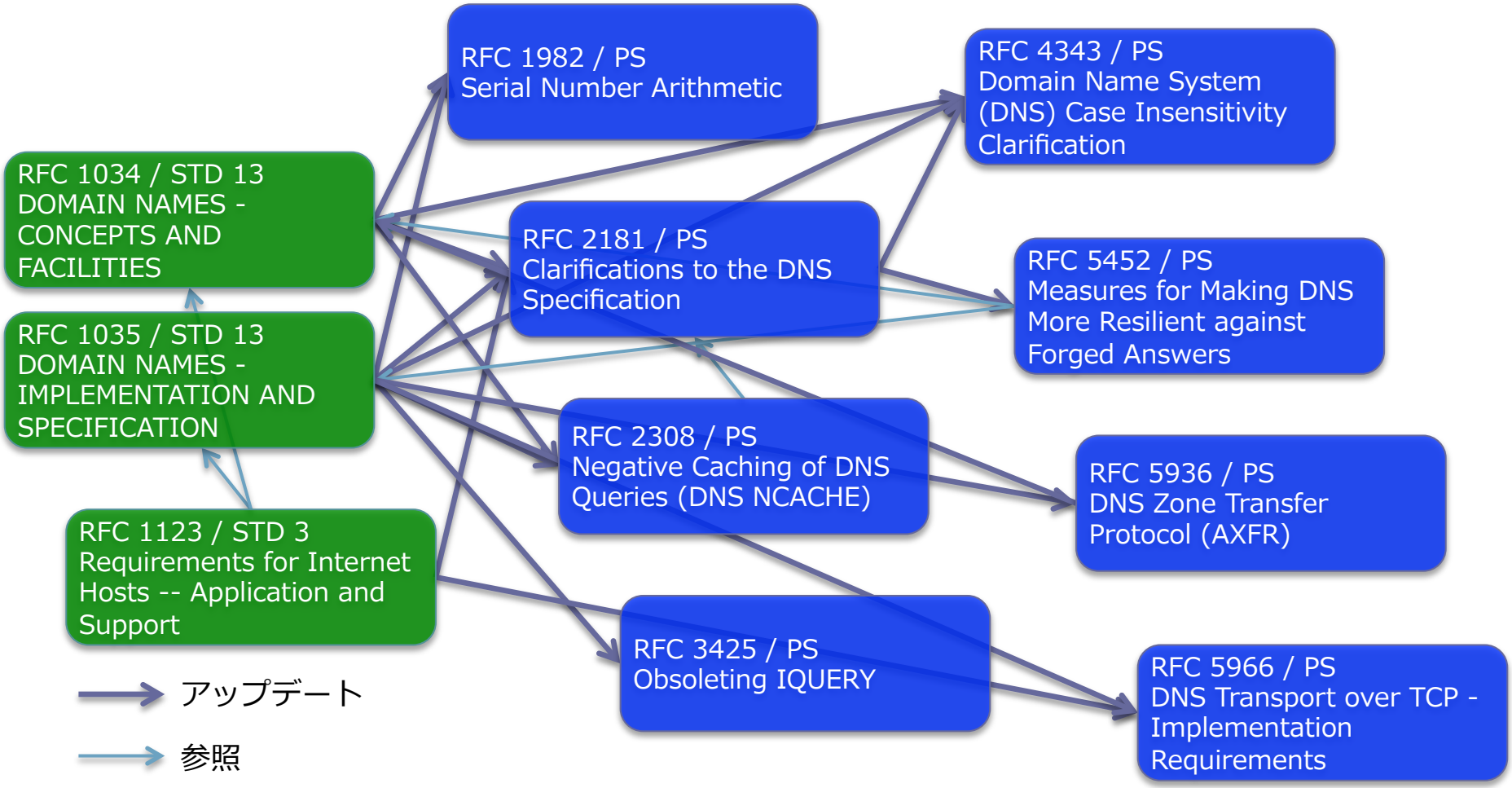
RFC 1034とRFC 1035への道

- 曖昧さや間違いがある。
  - RFC 1034とRFC 1035が基本仕様であるが、
  - 後から公開されたRFCにより、
  - 曖昧な点や間違いが訂正されたり、仕様が変更されたりしているため、
  - RFC 1034とRFC 1035の内容がすべて正しいとは思わないように。
- RFC 1034と1035をアップデートしているRFCも合わせて読みたい。
  - RFC 2181 "Clarifications to the DNS Specification"はDNSの仕様の明確化をしているRFCであるので、読むべき。

RFC 1034と1035のアップデート（拡張機能のRFCは除く）

- RFC 1123
  - Requirements for Internet Hosts -- Application and Support
- RFC 1982
  - Serial Number Arithmetic
- RFC 2181
  - Clarifications to the DNS Specification
- RFC 2308
  - Negative Caching of DNS Queries (DNS NCACHE)
- RFC 3425
  - Obsoleting IQUERY
- RFC 4343
  - Domain Name System (DNS) Case Insensitivity Clarification
- RFC 5452
  - Measures for Making DNS More Resilient against Forged Answers
- RFC 5936
  - DNS Zone Transfer Protocol (AXFR)
- RFC 5966
  - DNS Transport over TCP - Implementation Requirements

DNSの基本仕様およびアップデート



RFC 1034
DOMAIN NAMES – CONCEPTS AND FACILITIES

RFC 1034 ドメイン名 – 概念と機能

1987年11月公開

著者: ポール モカペトリス（Paul Mockapetris）

RFC 1034の目次

- 1. STATUS OF THIS MEMO
  本文書の位置づけ
- 2. INTRODUCTION
  はじめに
- 3. DOMAIN NAME SPACE and RESOURCE RECORDS
  ドメイン名空間とリソースレコード
- 4. NAME SERVERS
  ネームサーバー
- 5. RESOLVERS
  リゾルバー
- 6. A SCENARIO
  シナリオ
- 7. REFERENCES and BIBLIOGRAPHY
  出典および参考文献

RFC 1034の概要

- DNSの構成要素の役割や機能についての説明
  - ドメイン名空間とリソースレコード
  - ネームサーバー
  - リゾルバー

1. STATUS OF THIS MEMO
1. 本文書の位置づけ

- 概要
  - このRFCはDNSについての紹介を行う。
    ・タイトルの通り「概念と機能」について説明
  - 詳細はRFC 1035にて行う。

2. INTRODUCTION
2. はじめに

- 2.1. The history of domain names
  ドメイン名の歴史
- 2.2. DNS design goals
  DNSの設計目標
- 2.3. Assumptions about usage
  利用についての想定
- 2.4. Elements of the DNS
  DNSの要素

2. INTRODUCTION
2. はじめに

- 概要
  - DNSを導入するに至った経緯やDNSがどういうものかを紹介している。

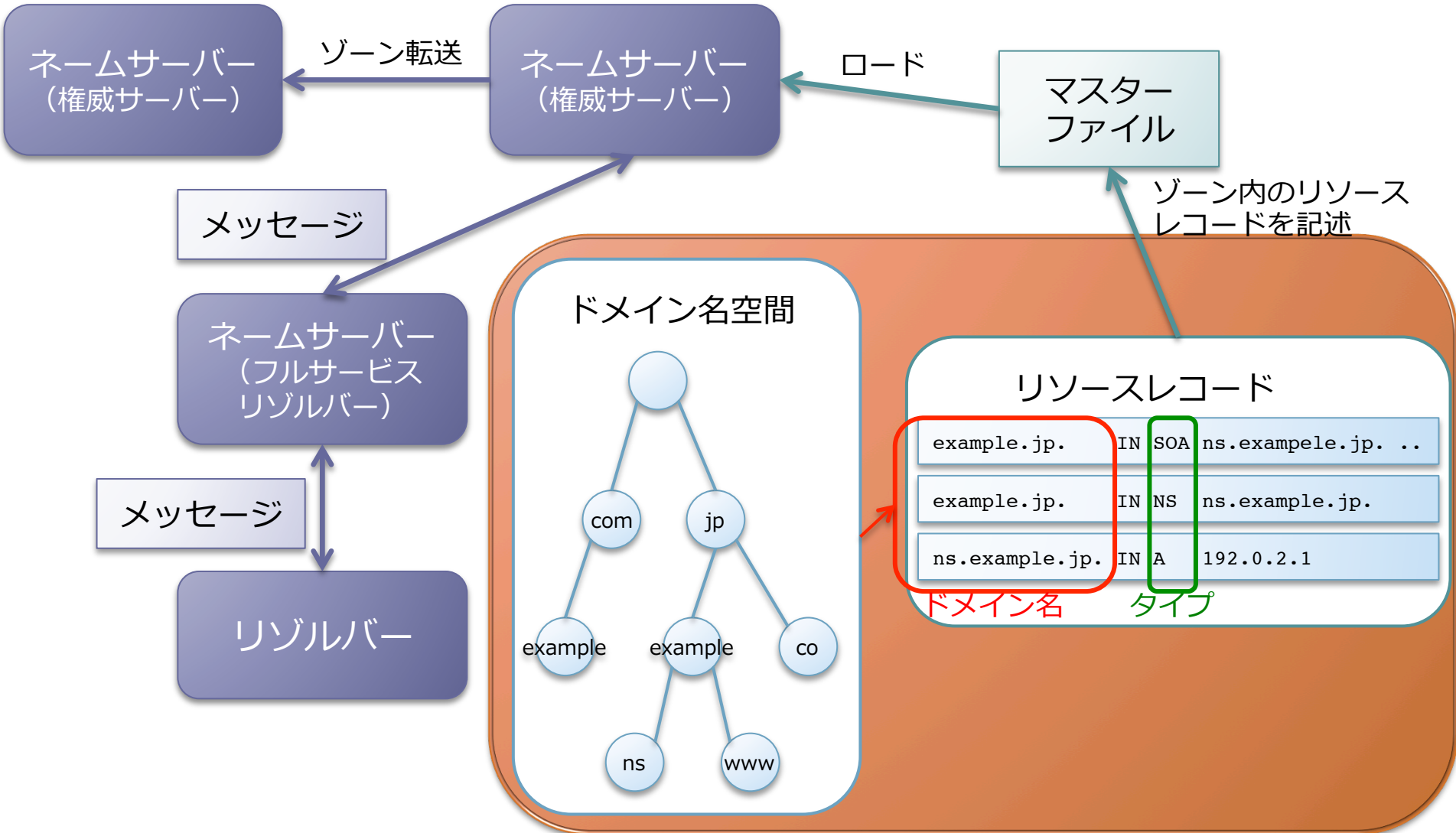
3. DOMAIN NAME SPACE and RESOURCE RECORDS
3. ドメイン名空間とリソースレコード

- 3.1. Name space specifications and terminology
  名前空間の仕様と用語
- 3.2. Administrative guidelines on use
  利用上の管理ガイドライン
- 3.3. Technical guidelines on use
  利用上の技術ガイドライン
- 3.4. Example name space
  名前空間の例
- 3.5. Preferred name syntax
  名前の構文

3. DOMAIN NAME SPACE and RESOURCE RECORDS
3. ドメイン名空間とリソースレコード

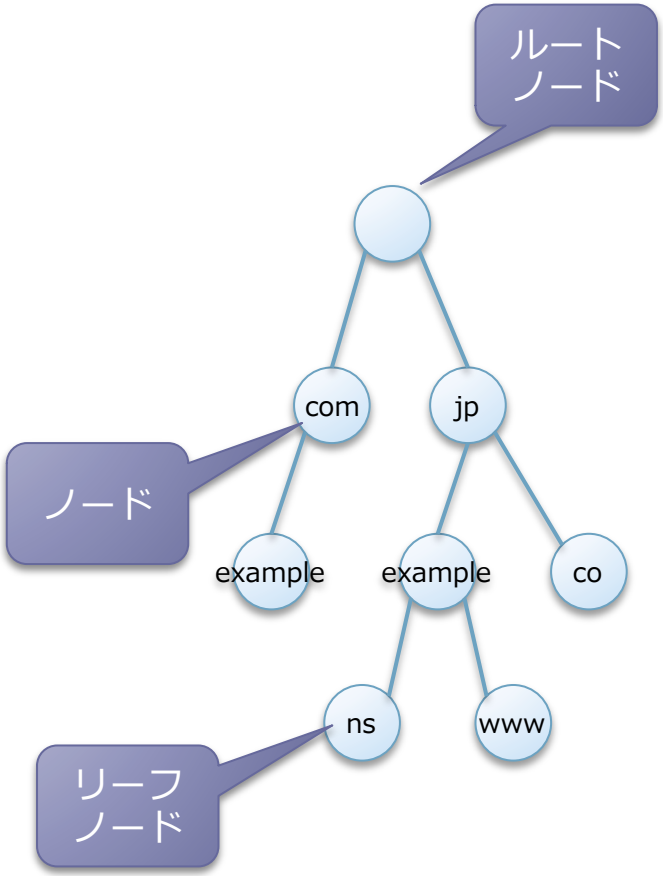
- 3.6. Resource Records
  リソースレコード
- 3.7. Queries
  問い合わせ
- 3.8. Status queries (Experimental)
  状態問い合わせ（実験機能）
- 3.9. Completion queries (Obsolete)
  補完問い合わせ（廃止機能）

3章の位置づけ


ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

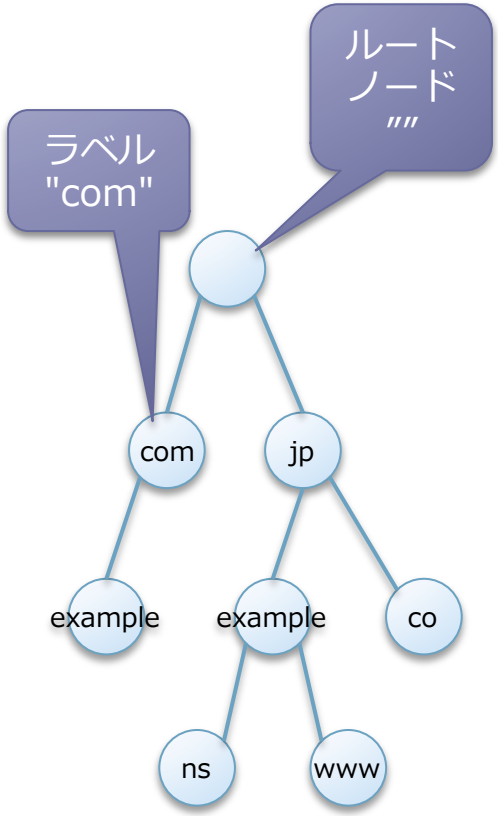
3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- ツリー構造とノード
  - ドメイン名空間はツリー構造
  - 各ノードとリーフはリソースの集まりに対応している
  - 内部ノードとリーフノードを区別しない。両方とも「ノード」と呼ぶ。



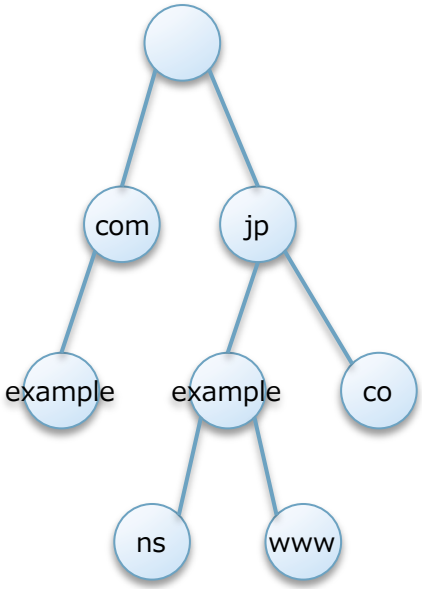
3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- ラベル
  - 各ノードは「ラベル」を持つ。
  - ラベルの長さは0オクテットから63オクテットまで
  - 兄弟ノードは同じラベルを持たない。
  - 兄弟でないノードは同じラベルを持てる。
  - ルートのためにnullラベル（長さ0）が予約されている。



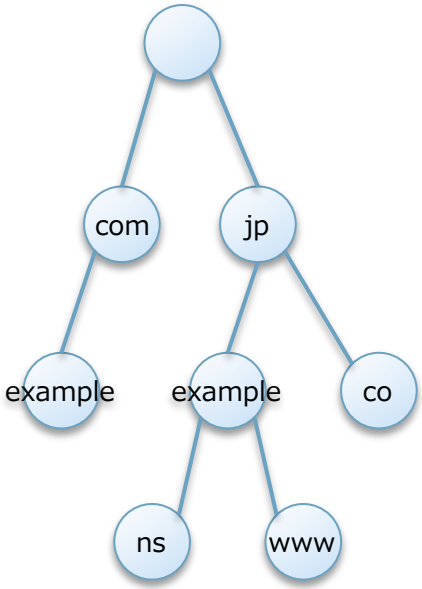
3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- ドメイン名
  - ノードのドメイン名はそのノードからルートノードまでのパス上のラベルのリスト
    - 例) "www", "example", "jp", ""



3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- ドメイン名の内部表現
  - ドメイン名はラベルをつなげたもの
  - ラベルはオクテットの長さと文字列で表される。
    - "www"の内部表現を16進数で表すと"3 77 77 77"となる。
  - すべてのドメイン名はルートで終わり、ルートのラベルはnull文字であるため、内部表現はドメイン名の終わりに0バイトの長さを使う。
    - www.example.jp.の内部表現 3 77 77 77 8 65 78 61 6d 70 6c 65 2 6a 70 0
- ドメイン名の長さ
  - ドメイン名のオクテット数は255まで

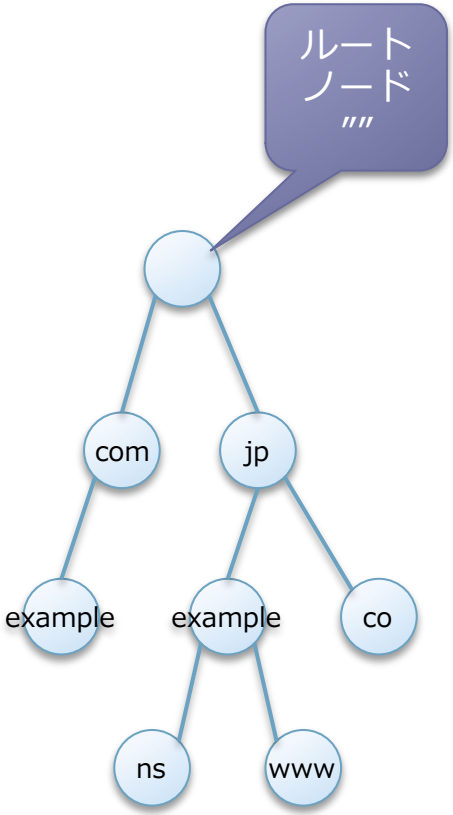


3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- 大文字小文字
  - ドメイン名の比較の際には大文字小文字を区別しない。
  - ドメイン名を受け取ったときには大文字小文字を維持すべき
- RFC 4343 "Domain Name System (DNS) Case Insensitivity Clarification"

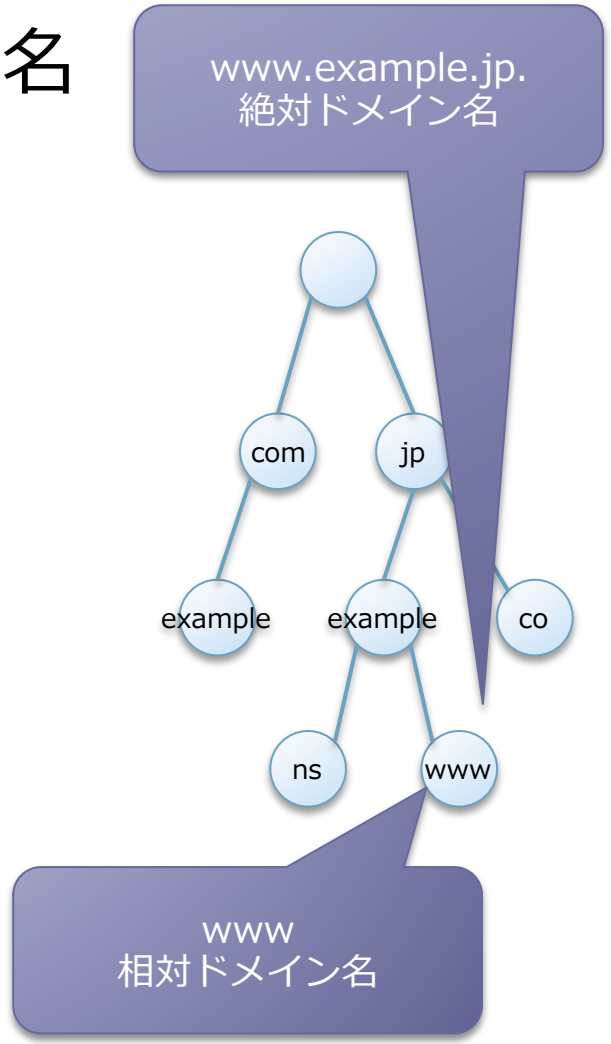
3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- ドメイン名の入力は表示上の表現
  - ラベルの長さを省き、ラベルを"."で分ける。
    - 例）www.example.jp.
  - ドメイン名はルート（空の）ラベルで終わるため、ドットで終わる形式になる。
    - 例）www.example.jp."空のラベル（非表示）"



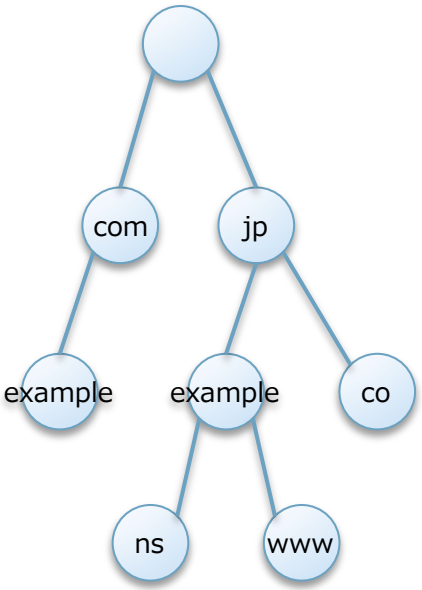
3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- 絶対ドメイン名と相対ドメイン名



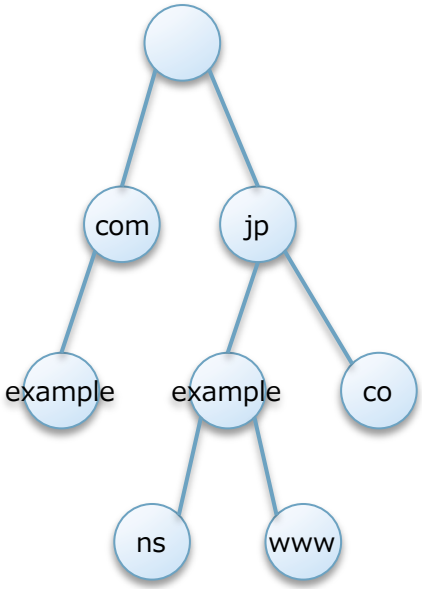
3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- 相対ドメイン名とオリジンや検索リスト
  - 相対ドメイン名はオリジン（注：マスターファイルのオリジン）や検索リスト（注：リゾルバーの検索リスト）に対して相対。
  - オリジンや検索リストのメンバーとしてルート"."が解釈される。入力を省略するために最後の"."がよく省かれる。
    - 例）www.example.jp
    - →FQDN（Fully Qualified Domain Name、完全修飾ドメイン名）？



3.1. Name space specifications and terminology
3.1. 名前空間の仕様と用語

- FQDN（Fully Qualified Domain Name、完全修飾ドメイン名）
  - トップレベルドメインまでを含んだドメイン名
    - 例: www.example.jp
  - この文法を示した明確な定義はない？
  - RFC 1594 FYI on Questions and Answers - Answers to Commonly asked "New Internet User" Questions
    - 5.2 What is a Fully Qualified Domain Name?
  - RFC 1983 Internet Users' Glossary
    - Fully Qualified Domain Name (FQDN)



3.2. Administrative guidelines on use
3.2. 利用上の管理ガイドライン

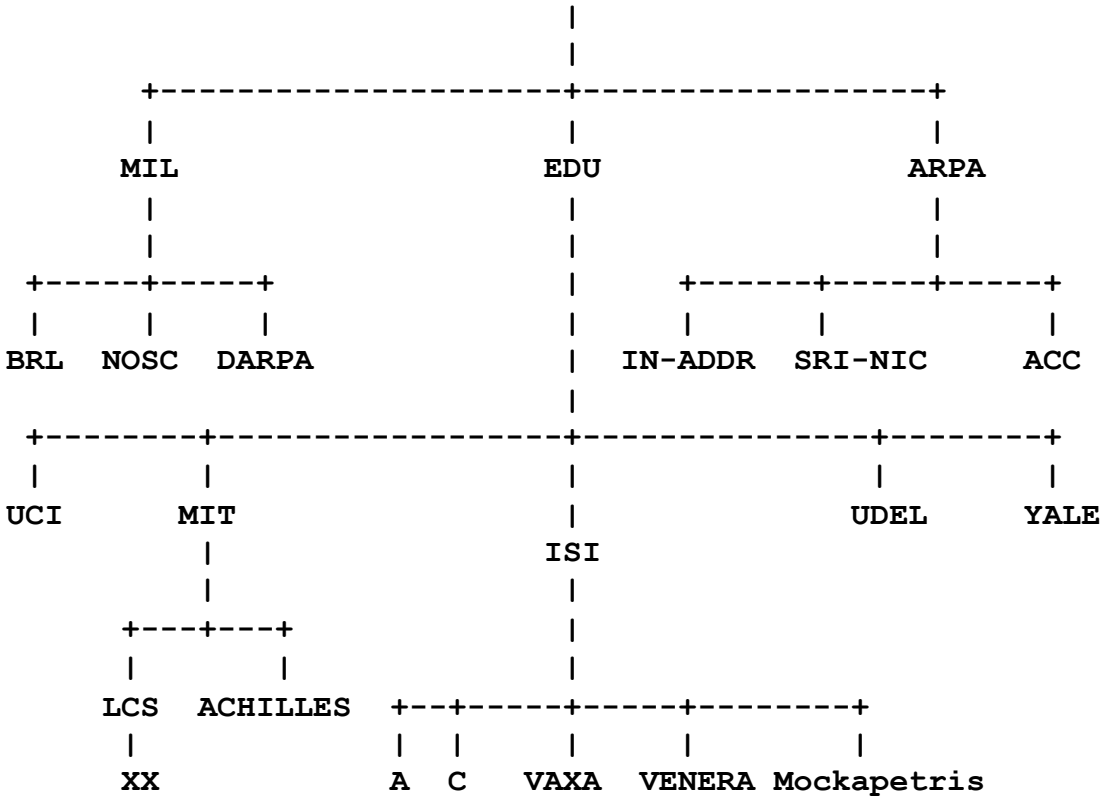
- 概要
  - DNSの技術仕様としては特定のツリー構造やラベルの選択規則を強要しないという方針について記述されている。

3.3. Technical guidelines on use
3.3. 利用上の技術ガイドライン

- 概要
  - DNSで扱うオブジェクトの要件について述べている。
    - ・オブジェクト名とドメイン名のマッピング
    - ・RRタイプとオブジェクトを記述するデータ形式

3.4. Example name space
3.4. 名前空間の例

- 当時の例。現在とは異なるので注意。



3.5. Preferred name syntax
3.5. 好ましい名前の構文

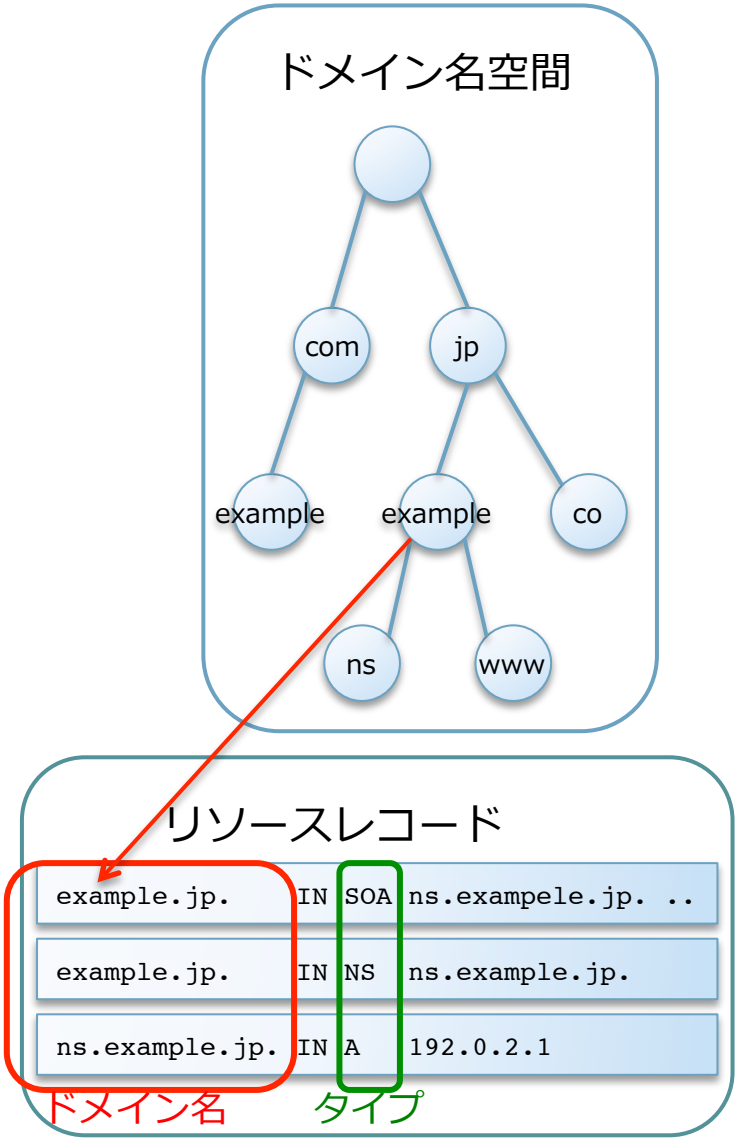
- 概要
  - ラベルの構文について
- ラベルとホスト名
  - ラベルはARPANETホスト名の規則に従う。
    - RFC 952 DOD INTERNET HOST TABLE SPECIFICATION
    - RFC 1123 Requirements for Internet Hosts -- Application and Support によりホスト名の仕様が変更された
- ホスト名
  - 英文字で始まる
    → 英文字あるいは数字で始まる（RFC 1123）
  - 英文字あるいは数字で終わる
  - 間の文字は英文字、数字、ハイフンが使える
- ラベル
  - ラベルは63文字未満

3.6. Resource Records
3.6. リソースレコード

- 概要
  - リソースレコードについての説明を記述する。

3.6. Resource Records
3.6. リソースレコード

- リソースレコード
  - ドメイン名はノードを識別する。
  - 各ノードはリソース情報の集まりを持つ。空でもよい。
  - 特定の名前に関連づけられたリソース情報の集まりは別々のリソースレコード（RRs）から構成される。
  - 集まりの中のRRsの順番は指定できないし、維持される必要も無い。



3.6. Resource Records
3.6. リソースレコード

- リソースレコードの用語
  - owner（オーナー）
    - そのRRがあるドメイン名
  - type（タイプ）
    - このリソースレコードのリソースのタイプを識別する符号化された16ビットの値。タイプは抽象的なリソースを参照する。
    - A, CNAME, HINFO, MX, NS, PTR, SOA
  - class
    - プロトコルファミリーを識別する符号化された16ビットの数
    - IN(the Internet system), CH(the Chaos system)

3.6. Resource Records
3.6. リソースレコード

- リソースレコードの用語
  - TTL
    - RRの生存期間。このフィールドは秒単位の32ビット整数。リゾルバーがキャッシュするときに使う。TTLはRRが破棄されるまでにキャッシュしてよい期間を示す。
    - RFC 2181 Clarifications to the DNS Specification "8. Time to Live (TTL)"
      - 符号無し整数
      - 最小値: 0
      - 最大値: 2147483647 (2^31 - 1)
      - 最上位ビットが1であるときにはTTLを0と扱うべき
  - RDATA
    - タイプとクラスに依存するデータ。

3.6.1. Textual expression of RRs
3.6.1. RRsのテキスト表現

- RRのテキスト表現の形式
  - RRは1行で示される。複数行になる場合には括弧を使う。

```
example.com. 172800 IN NS a.iana-servers.net.
example.com.   3600 IN SOA dns1.icann.org. (
                hostmaster.icann.org.
                2012080872 7200 3600 1209600 3600 )
```

  - 行の先頭はRRのオーナー。

```
example.com. 172800 IN NS a.iana-servers.net.
```

  - 空白で始まる行は、オーナーが前のRRと同じと想定される。

```
example.com. 172800 IN NS a.iana-servers.net.
             172800 IN NS b.iana-servers.net.
```

3.6.1. Textual expression of RRs
3.6.1. RRsのテキスト表現

- RRのテキスト表現の形式 – TTL,タイプ,クラス
- オーナーの次は、TTLとタイプとクラス。
  - クラスとタイプはニーモニックを使う。
  - TTLは整数を使う。
  - タイプは必ず最後である。
  - INクラスとTTLはわかりやすさのために例からよく省略される。
- RRのテキスト表現の形式 – RDATA
  - リソースデータあるいはRRのRDATAセクションは典型的表現の知識を使って与えられる。

3.6.2. Aliases and canonical names
3.6.2. 別名と正式名

- 概要
  - CNAMEについての説明
- CNAME
  - CNAMEは別名に対するオーナー名を識別する。RRのRDATAセクションの対応する正式名を示す。
  - CNAME RRがノードに存在したら、他のデータは存在すべきではない。これは、正式名とその別名に違いがでないを保証する。
  - この規則はキャッシュされたCNAMEは権威サーバーに他のRRタイプを確認することなしに使われることも保証する。

3.6.2. Aliases and canonical names
3.6.2. 別名と正式名

- RFC 2181 Clarifications to the DNS Specification
  "10.1. CNAME resource records"
  - CNAMEの意味を明確化
    - CNAME ("canonical name")「正式名」は"alias name"「別名」と関連づけるために使う
    - CNAMEは「別名」を示すのでなく、「別名」に対する「正式名」を示す。
      - 別名 IN CNAME 正式名

3.7. Queries
3.7. 問い合わせ

- 概要
  - 問い合わせについて
- 問い合わせ
  - DNS問い合わせと応答は標準メッセージフォーマットで運ばれる。
  - メッセージフォーマットは常に存在するいくつかの固定フィールドと問い合わせパラメータとRRを運ぶ4つのセクション
  - ヘッダーにある最も重要なフィールドは異なる問い合わせを分けるopcodeと呼ばれる4ビットのフィールドである。

3.7. Queries
3.7. 問い合わせ

- 4つのセクション
  - Question
    - 問い合わせ名と他の問い合わせパラメータ
  - Answer
    - 回答のRR
  - Authority
    - 他の権威サーバーを示すRR。answerセクションの権威データのSOA RRでもよい。
  - Additional
    - 他のセクションのRRを使う際に役に立つかもしれないRR

3.7.1. Standard queries
3.7.1. 標準の問い合わせ

- 標準の問い合わせは
  - ・ターゲットドメイン名（QNAME）、
  - ・問い合わせタイプ（QTYPE）、
  - ・問い合わせクラス（QCLASS）
  - ・を示し、一致するRRを尋ねる。

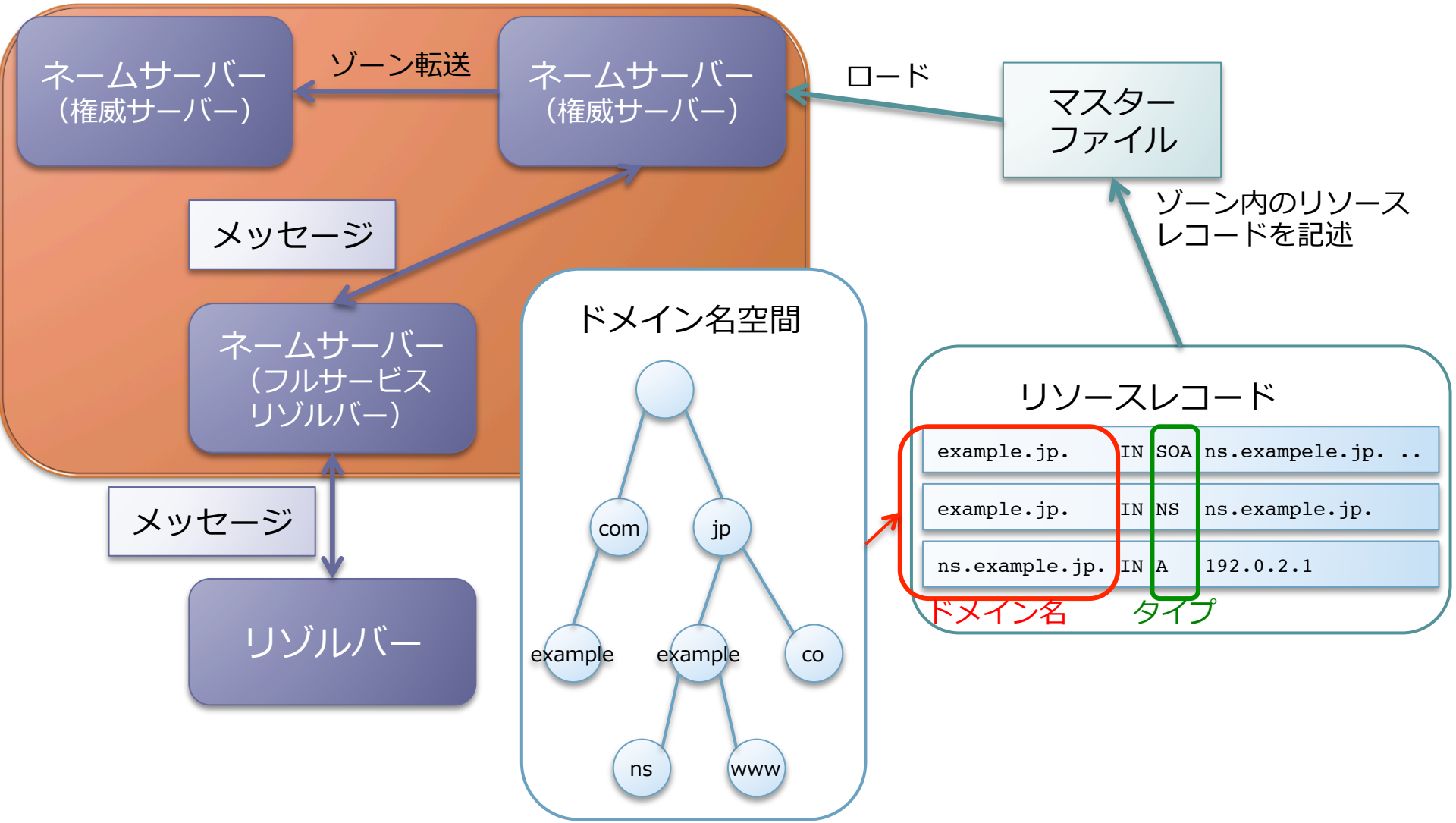
3.7.2. Inverse queries (Optional)
3.7.2. 逆問い合わせ (付加機能)

- RFC 3425 Obsoleting IQUERYにより廃止

4. NAME SERVERS
4. ネームサーバー

- 4.1. Introduction
  はじめに
- 4.2. How the database is divided into zones
  データベースのゾーンの分割方法
- 4.3. Name server internals
  ネームサーバーの内部

4章の位置づけ


ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

4.1. Introduction
4.1. はじめに

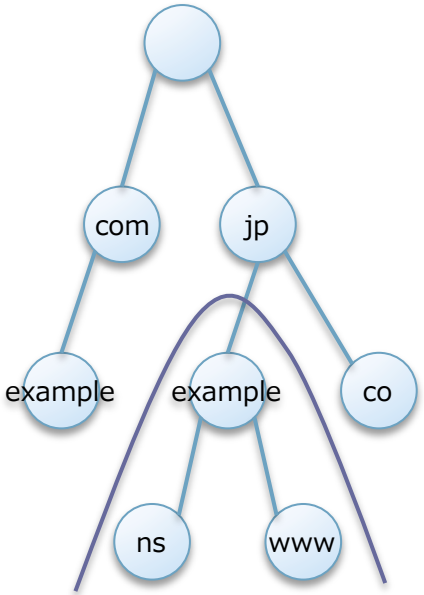
- 概要
  - ネームサーバーとゾーンについて記述している。
- ネームサーバーとゾーン
  - ネームサーバーはドメインのデータベースのリポジトリーである。
  - データベースはゾーンと呼ばれる部分に分割され、ネームサーバー間に分散されている。
  - ネームサーバーの基本的なタスクはゾーン内のデータへの問い合わせに回答することである。

4.2. How the database is divided into zones
4.2. データベースをゾーンに分割する方法

- 概要
  - データベースをゾーンに分割する方法について説明する。

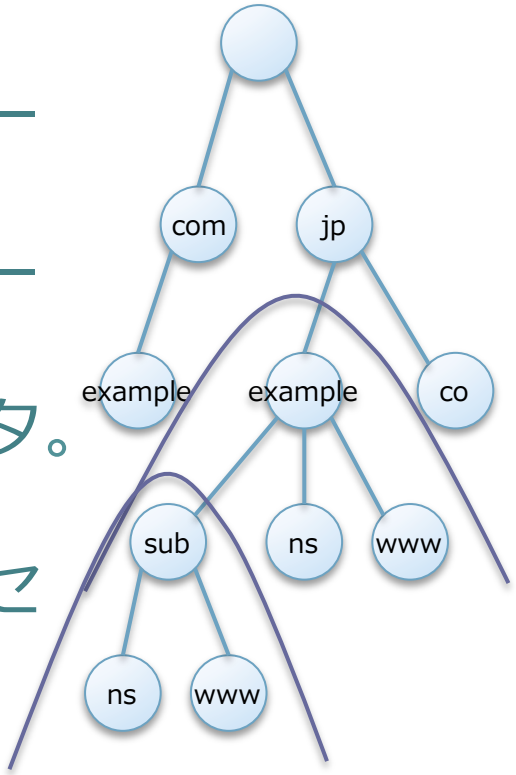
4.2. How the database is divided into zones
4.2. データベースをゾーンに分割する方法

- 名前空間の分割は2つの隣接したノードの間で行われる。
  - 左記の例ではexampleとjpの間
- 接続された名前空間の各グループは別々のゾーンとなる。
- ゾーンは領域内のすべての名前の権威となる。
- RFC 2181 Clarifications to the DNS Specification "6. Zone Cuts"に詳細に説明あり



4.2.1. Technical considerations
4.2.1. 技術的な考慮

- ゾーンを記述するデータは4つの部分を持つ
  - ゾーン内のすべてのノードの権威データ
  - ゾーンのトップノードを定義するデータ
  - 委任したサブゾーンを記述するデータ。すなわち、ゾーンの最下部の切断。
  - サブゾーンのネームサーバーにアクセスをさせるデータ（グルー"glue"データと呼ばれる。）



4.2.2. Administrative considerations
4.2.2. 管理上の考慮

- 組織が自身のドメインを管理したいときの最初の一歩は、正しい親ゾーンを識別し、親ゾーンの所有者に管理を委任してもらうように同意を得ることである。
- 新しいサブゾーンに適した名前が選ばれたら、新しい所有者は複数のネームサーバーを用意することを示すべきである。
- 最後の導入の段階では、委任が効果を持つのに必要なNS RRsとグルーRRsを親ゾーンに追加してもらう。両方のゾーンの管理者はゾーンカットの両側で同じNSとグルーRRsを持ち続けるようにする。

4.3. Name server internals
4.3. ネームサーバーの内部

4.3.1. Queries and responses
4.3.1. 問い合わせと応答

- ネームサーバの主要な活動は標準問い合わせに回答することである。
- 問い合わせと応答はRFC 1035のメッセージフォーマットで運ばれる。
- 問い合わせはQTYPE, QCLASS, QNAMEを含む。

4.3.1. Queries and responses
4.3.1. 問い合わせと応答

- 非再帰検索モード
  - サーバーとして最も単純なモードは非再帰である。
  - ローカル情報のみを使って回答する。
  - 応答はエラー、回答、回答に最も近い他のサーバへの参照のいずれかを含む。
  - すべてのネームサーバーは非再帰問い合わせを実装しなければならない。

4.3.1. Queries and responses
4.3.1. 問い合わせと応答

- 再帰検索モード
  - クライアントとして最も単純なモードは再帰検索モードである。
  - このモードでは、ネームサーバーはリゾルバーの役割として動作し、エラーあるいは回答を返す。しかし、参照は返さない。
  - このサービスはネームサーバーでは付加機能である。ネームサーバーは再帰検索モードを使うクライアントを制限してもよい。

4.3.2. Algorithm
4.3.2. アルゴリズム

- 概要
  - ネームサーバーの問い合わせに対するアルゴリズムを説明している。

4.3.3. Wildcards
4.3.3. ワイルドカード

- 概要
  - ラベル"*"で始まる所有者名を持つRRsは特別な扱いを行う。
  - このようなRRをワイルドカードと呼ぶ。

4.3.4. Negative response caching (Optional)
4.3.4. 比例応答のキャッシュ (付加機能)

- 否定応答をキャッシュすることについての説明
- 誤りあり
  - The method is that a name server may add an SOA RR to the additional section of a response when that response is authoritative.
- RFC 2181 Clarifications to the DNS Specification "7.1. Placement of SOA RRs in authoritative answers"により訂正
  - リソースが存在しないときにレスポンスに含めるSOAレコードはadditionalセクションではなくauthorityセクションに置く。

4.3.4. Negative response caching (Optional)
4.3.4. 比例応答のキャッシュ (付加機能)

- RFC 2308 Negative Caching of DNS Queries (DNS NCACHE)で詳細な説明と再定義
  - RFC 1034からの変更点（8 - Changes from RFC 1034）
    - ネガティブキャッシュはoptionalであったが、RFC 2308ではキャッシュする場合は、ネガティブキャッシュもしなければならないよう（must）になった。
    - AuthorityセクションのSOAレコードはキャッシュしなければならない（MUST）。
    - キャッシュしたSOAレコードは応答に加えなければならない（MUST）。

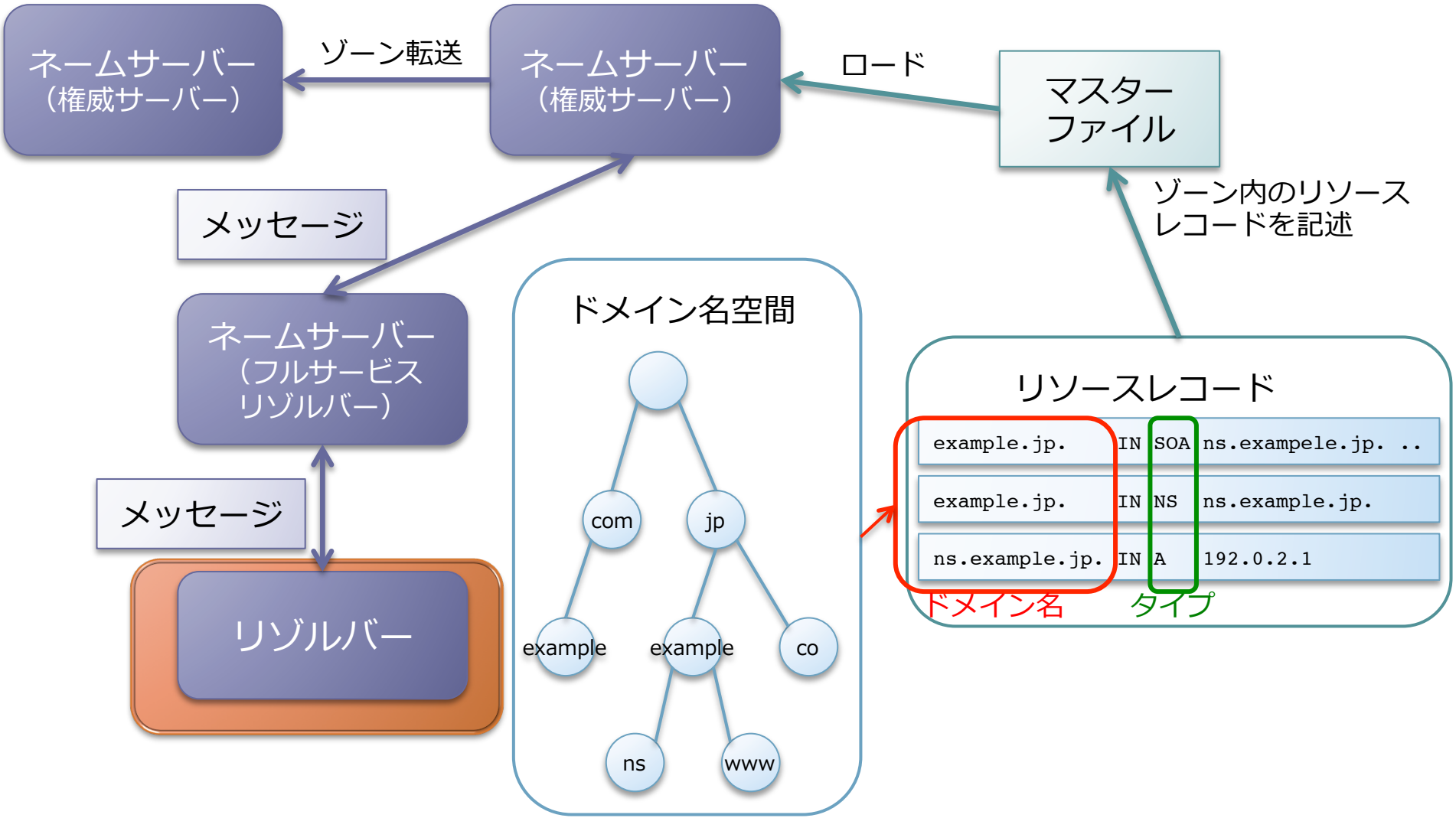
4.3.5. Zone maintenance and transfers
4.3.5. ゾーンの保守と転送

- 概要
  - ゾーン転送についての説明
- RFC 5936 DNS Zone Transfer Protocol (AXFR)でAXFRのすべてがアップデートされている。

5. RESOLVERS
5. リゾルバー

- 5.1. Introduction
  はじめに
- 5.2. Client-resolver interface
  クライアント-リゾルバー インターフェイス
- 5.3. Resolver internals
  リゾルバーの内部

5章の位置づけ


ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

5.1. Introduction
5.1. はじめに

- 概要
  - リゾルバーはユーザープログラムとドメインネームサーバーへのインターフェイスのプログラムである。
  - リゾルバーはユーザープログラムからサブルーティンコールあるいはシステムコールにより要求を受け付け、ローカルのホストのデータ形式と互換性のある形式で欲しい情報を返す。

5.2. Client-resolver interface
5.2. クライアント-リゾルバーのインターフェイス

- 概要
  - 典型的な機能
    - ホスト名からホストアドレスへの変換
    - ホストアドレスからホスト名への変換
    - 一般的な検索機能

5.3. Resolver internals
5.3. リゾルバーの内部

- 概要
  - リゾルバーの実装についての説明
  - スタブリゾルバー
  - リソース
  - アルゴリズム

6. A SCENARIO
6. シナリオ

- 省略

7. REFERENCES and BIBLIOGRAPHY
7. 出典と参考文献

- 省略

RFC 1035
DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

RFC 1035 ドメイン名 – 実装と仕様

1987年11月発行

著者: ポール モカペトリス（Paul Mockapetris）

RFC 1035の目次

RFC 1035の概要

- DNSのプロトコルの定義
  - ドメイン名空間
  - リソースレコード
  - メッセージ
  - マスターファイル
  - ネームサーバーの実装
  - リゾルバーの実装
  - メールエクスチェンジャ

1. STATUS OF THIS MEMO

1. 本文書の位置づけ

- 概要
  - この文書はドメイン名システムとそのプロトコルについての詳細を記述するものである。

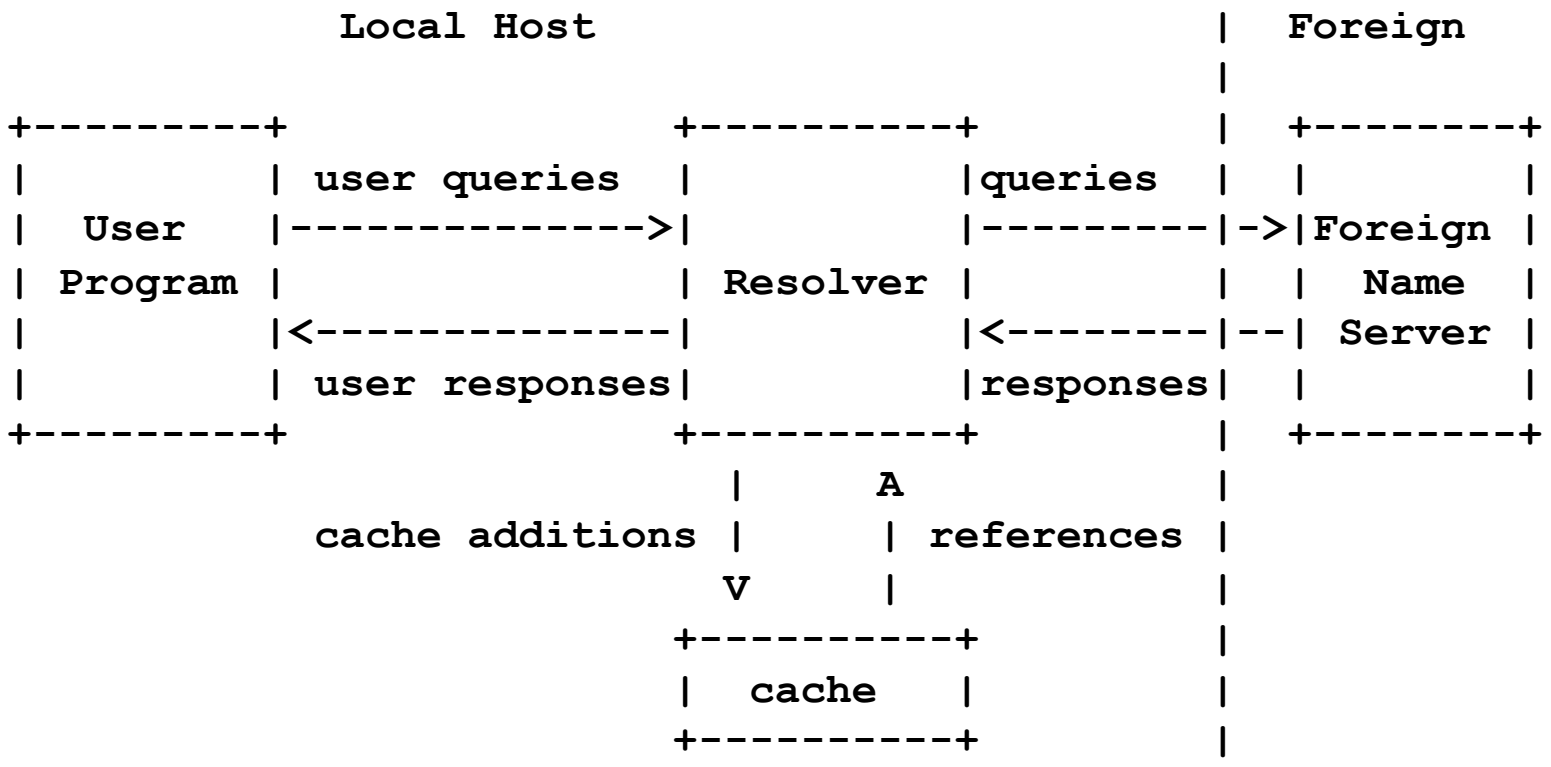
2. INTRODUCTION
2. はじめに

- 2.1. Overview
  概説
- 2.2. Common configuration
  共通の構成
- 2.3. Conventions
  取り決め

2.1. Overview
2.1. 概説

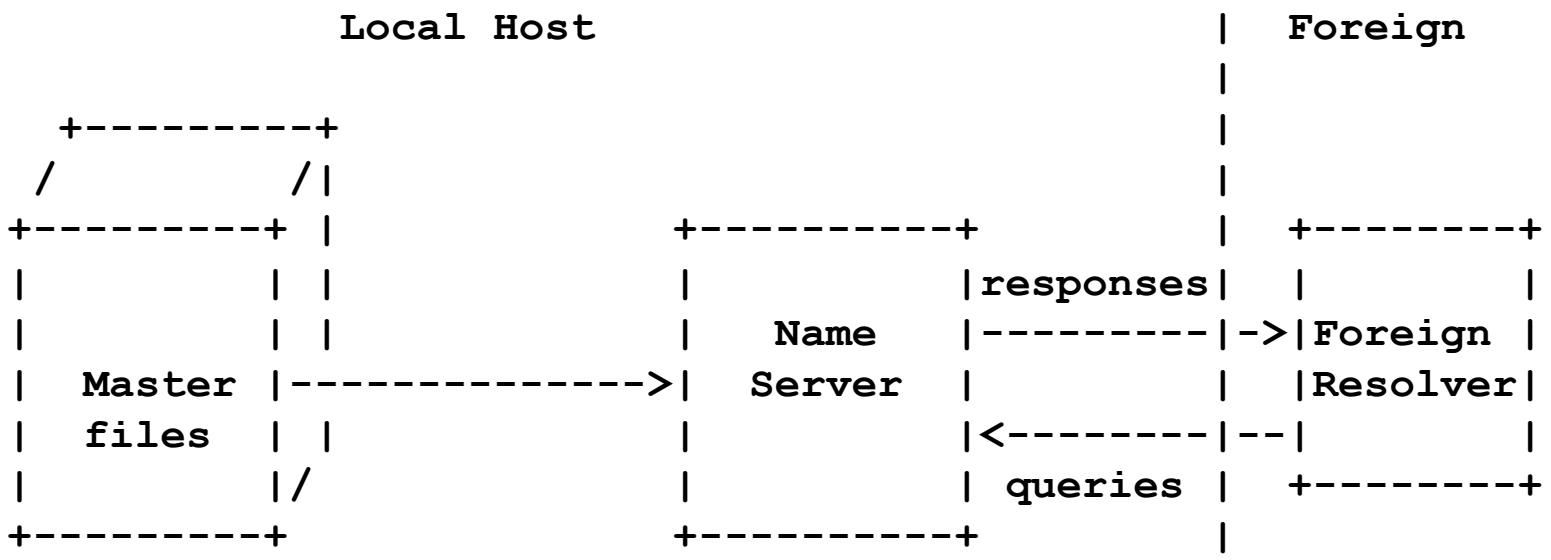
2.2. Common configurations
2.2. 共通の構成

- リゾルバーの最も単純で典型的な構成



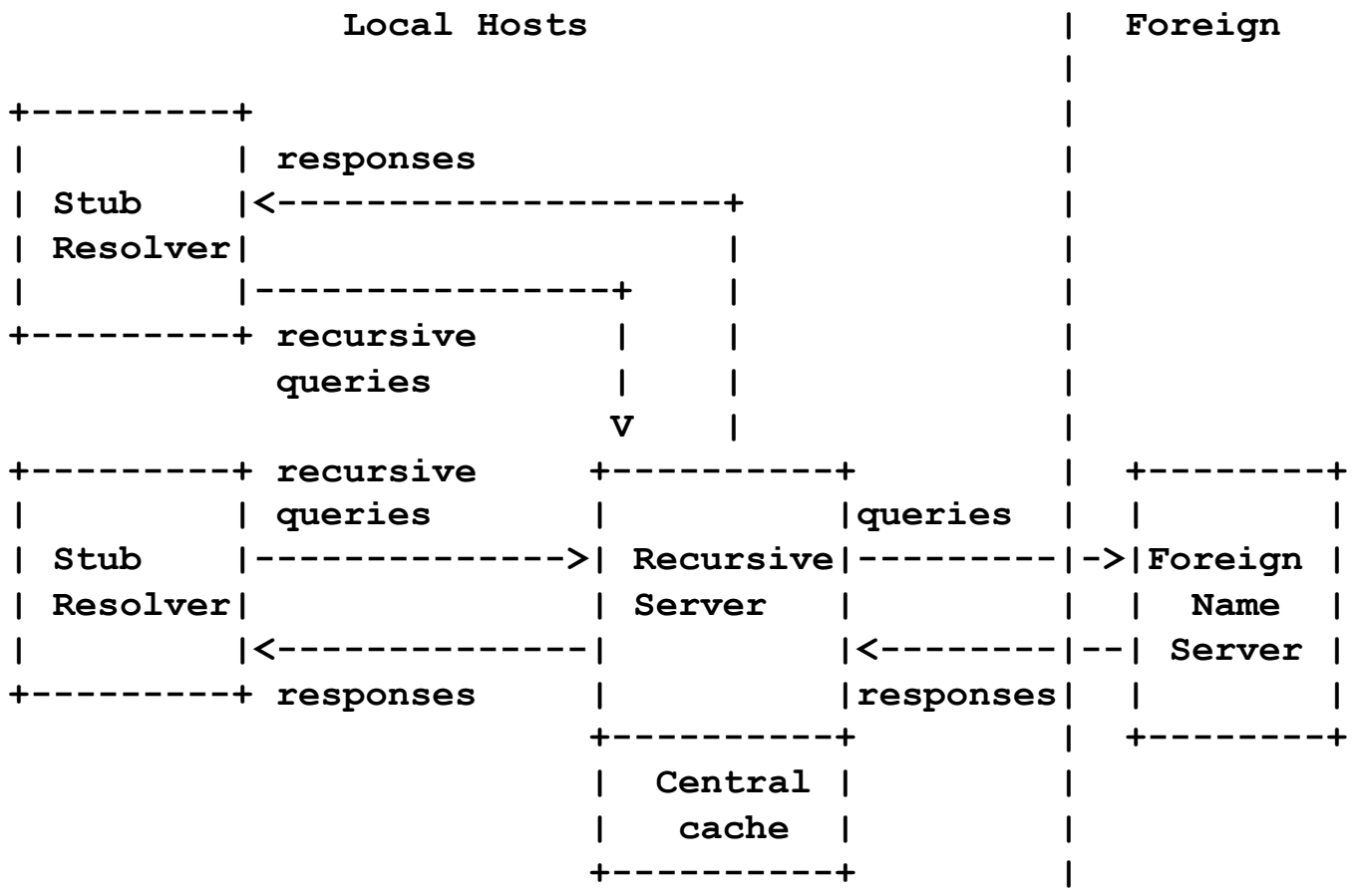
2.2. Common configurations
2.2. 共通の構成

- ネームサーバーの単純な構成



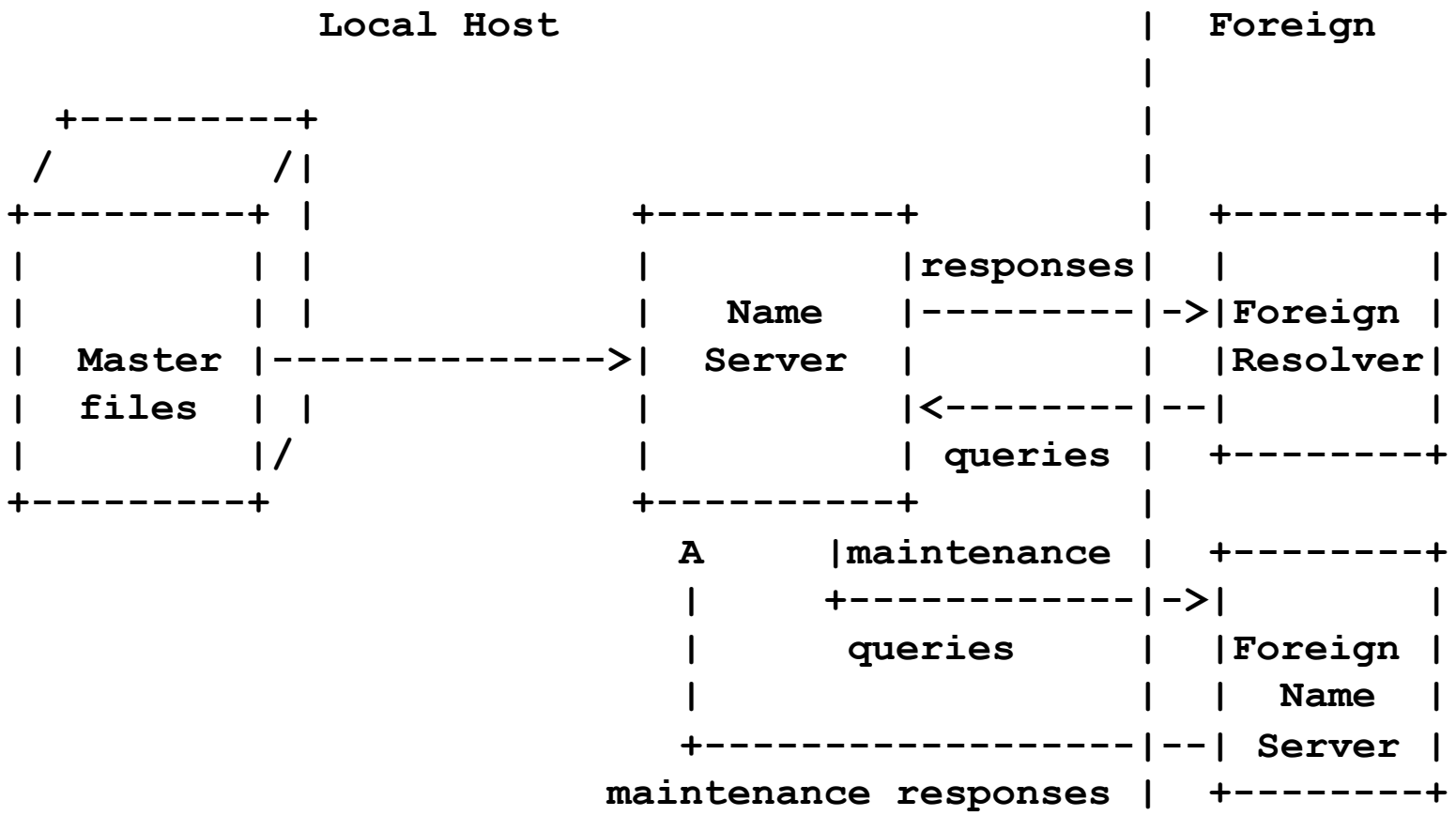
2.2. Common configurations
2.2. 共通の構成

- フルサービスリゾルバーのある構成



2.2. Common configurations
2.2. 共通の構成

- ゾーン転送を使う構成



2.3. Conventions
2.3. 取り決め

2.3.1. Preferred name syntax
2.3.1. 名前の構文

- 概要
  - ドメイン名の構文
  - ラベルの構文
- ドメイン名の構文
  - <domain> ::= <subdomain> | " "
  - <subdomain> ::= <label> | <subdomain> "." <label>
  - <label> ::= <letter> [[<ldh-str>] <let-dig>]
  - <ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>
  - <let-dig-hyp> ::= <let-dig> | "-"
  - <let-dig> ::= <letter> | <digit>
  - <letter> ::= any one of the 52 alphabetic characters A through Z in
  - upper case and a through z in lower case
  - <digit> ::= any one of the ten digits 0 through 9

2.3.1. Preferred name syntax
2.3.1. 名前の構文

- ラベル
  - ラベルはホスト名の規則に従う。
    - 英文字で始まる → 英文字あるいは数字で始まる（RFC 1123）
    - 英文字あるいは数字で終わる
    - 間の文字は英文字、数字、ハイフンが使える
  - 当時のホスト名の仕様
    - RFC 952 DOD INTERNET HOST TABLE SPECIFICATION
  - RFC 1123 Requirements for Internet Hosts -- Application and Support によりホスト名の仕様が変更された
  - ラベルは63文字未満

2.3.2. Data Transmission Order
2.3.2. データ転送の順番

- 概要
  - データのオクテットレベルでの順番を説明している。

2.3.3. Character Case
2.3.3. 大文字小文字

- 概要
  - 大文字小文字の取り扱いについて
- 大文字小文字の区別
  - ラベルやドメイン名などの比較の際には大文字小文字を区別しない。
- 大文字小文字の維持
  - 可能な限り大文字小文字を維持する。
- RFC 4343 "Domain Name System (DNS) Case Insensitivity Clarification"

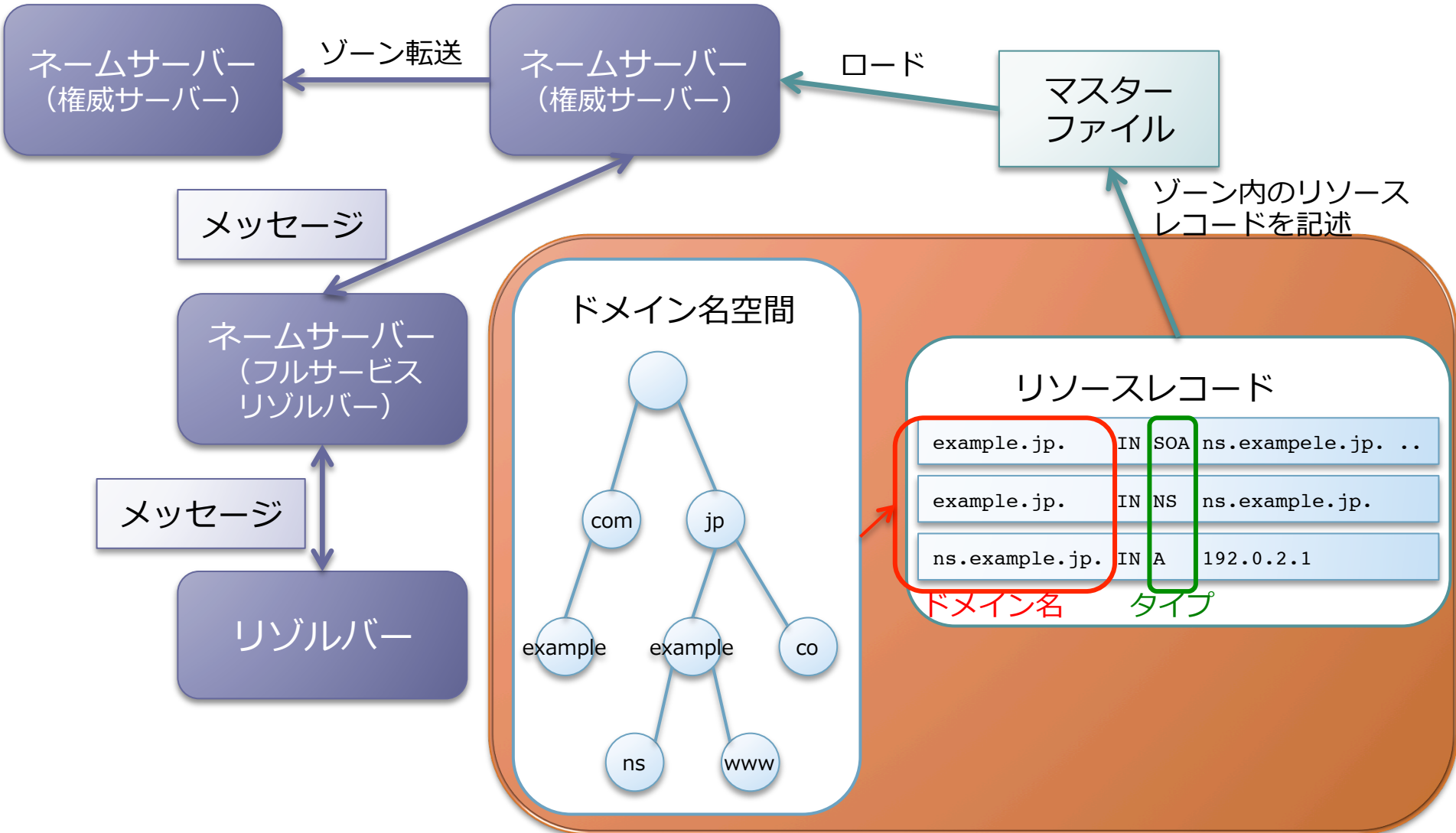
2.3.4. Size limits
2.3.4. サイズの制限

- 概要
  - サイズの制限
- サイズの制限
  - ラベル
    - 63オクテット以下
  - 名前
    - 255オクテット以下
  - TTL
    - 符号付き32ビット整数の正の数
  - UDPメッセージ
    - 512オクテット以下

3. DOMAIN NAME SPACE AND RR DEFINITIONS
3. ドメイン名空間とRRの定義

- 3.1. Name space definitions
  名前空間の定義
- 3.2. RR definitions
  RRの定義
- 3.3. Standard RRs
  標準のRRs
- 3.4. ARPA Internet specific RRs
  Internet特有のRRs
- 3.5. IN-ADDR.ARPA domain
  IN-ADDR.ARPAドメイン
- 3.6. Defining new types, classes, and special namespaces
  新しいタイプ、クラス、特別な名前空間の定義方法

3章の位置づけ

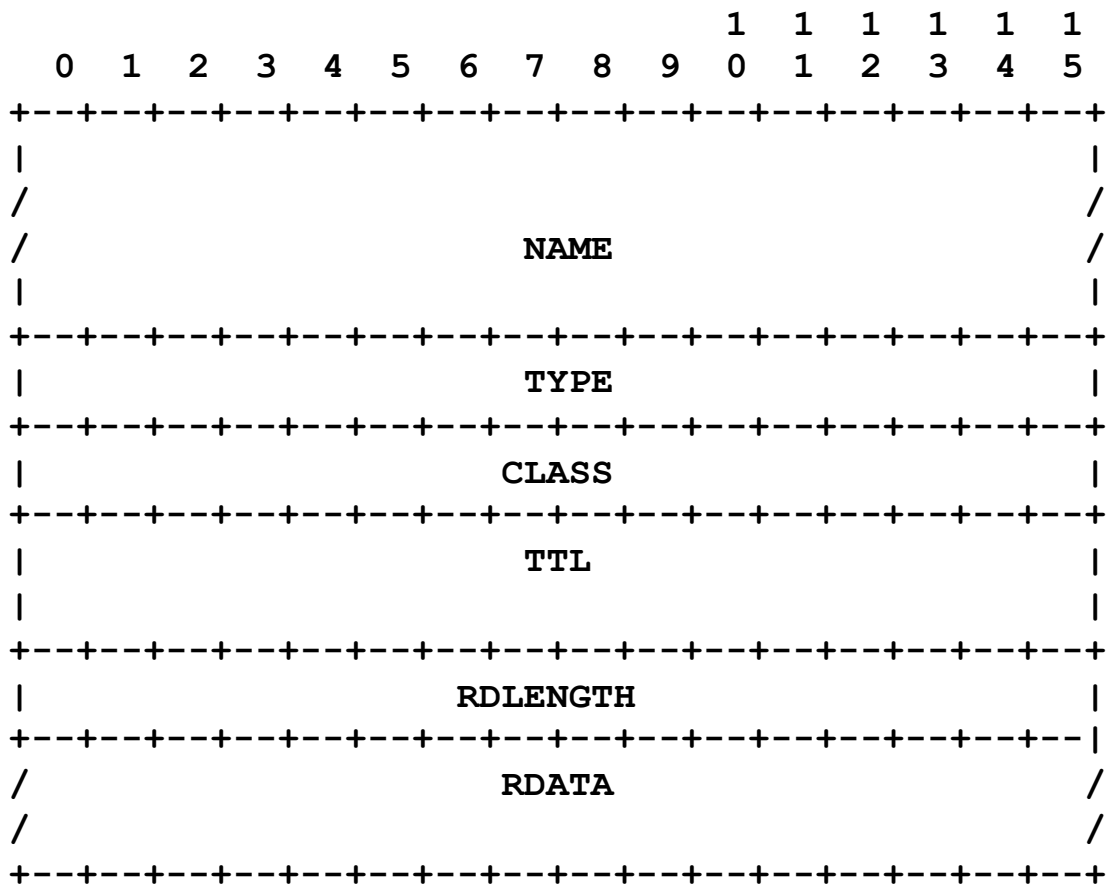

ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

3.1. Name space definitions
3.1. 名前空間の定義

- 概要
  - ドメイン名
  - ラベル
- ドメイン名とラベルの定義
  - メッセージ内のドメイン名は一続きのラベルで説明される。
  - 各ラベルは一つのオクテットの長さとオクテットの数で表現される。
  - 各ドメイン名はルートのnullラベルで終わる。ドメイン名は0の長さのバイトで終わる。
  - 各長さの上位2ビットは0になり、長さのオクテットの残りの6ビットはラベルを63オクテット以下に制限する。
  - ドメイン名の長さは255オクテット以下。

3.2. RR definitions
3.2. RRの定義

3.2.1. Format
3.2.1. フォーマット



3.2.1. Format
3.2.1. フォーマット

- NAME
  - 所有者名。このリソースレコードに関するノードの名前
- TYPE
  - RR TYPEコードの一つを含む2オクテット
- CLASS
  - RR CLASSコードの一つを含む2オクテット

3.2.1. Format
3.2.1. フォーマット

- TTL
  - 符号付き32ビット整数。リソースレコードがキャッシュされてもよい時間間隔。
- RDLENGTH
  - RDATAフィールドのオクテット長を示す符号無し16ビット整数。
- RDATA
  - リソースを示す可変長の文字。

3.2.1. Format
3.2.1. フォーマット

- 間違い
  - "SOA records are always distributed with a zero TTL to prohibit caching."
- RFC 2181 Clarifications to the DNS Specification "7.2. TTLs on SOA RRs"
  - どこにも言及されていないし、そのような実装は行われていない。実装はTTLが0であることを想定すべきではないし、0のTTLを持つSOAレコードを送信することを要求してもいけない。

3.2.2. TYPE values
3.2.2. TYPEの値

TYPE	value	meaning
A	1	a host address
NS	2	a host address
CNAME	5	the canonical name for an alias
SOA	6	marks the start of a zone of authority
WKS	11	a well known service description
PTR	12	a domain name pointer
MX	15	mail exchange
TXT	16	text strings

3.2.3. QTYPE values
3.2.3. QTYPEの値

- QTYPEは問い合わせのquestionパートで出てくる。
- QTYPEはTYPEのスーパーセット。
- QTYPEの"MAILB"と"MAILA"は使われていない。

QTYPE	value	meaning
AXFR	252	A request for a transfer of an entire zone
*	255	A request for all records

3.2.4. CLASS values
3.2.4. CLASSの値

CLASS mnemonics	value	meaning
IN	1	the Internet
CS	2	the CSNET class (Obsolete)
CH	3	the CHAOS class
HS	4	Hesiod

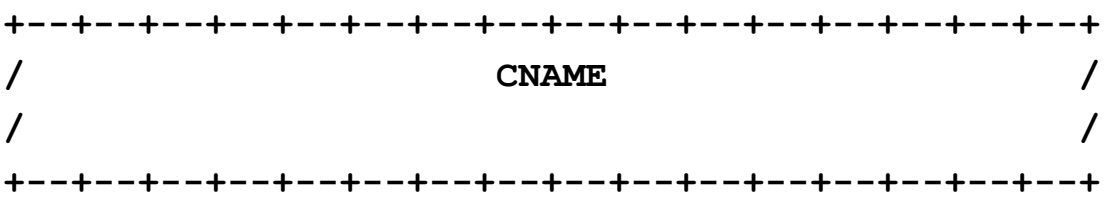
3.2.5. QCLASS values
3.2.5. QCLASSの値

- QCLASSは問い合わせのquestionセクションで出てくる。
- QCLASSはCLASSのスーパーセット

QCLASS mnemonics	value	meaning
*	255	any class

3.3. Standard RRs
3.3. 標準のRRs

3.3.1. CNAME RDATA format

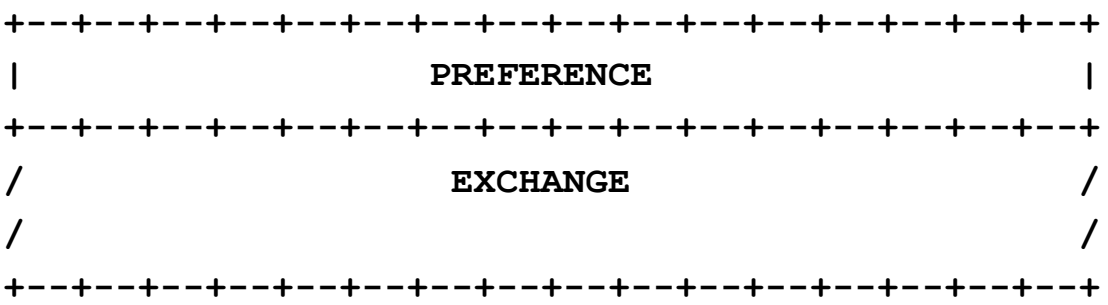


where:

CNAME	A <domain-name> which specifies the canonical or primary name for the owner. The owner name is an alias.

CNAME RRs cause no additional section processing, but name servers may choose to restart the query at the canonical name in certain cases. See the description of name server logic in [RFC-1034] for details.

3.3.9. MX RDATA format

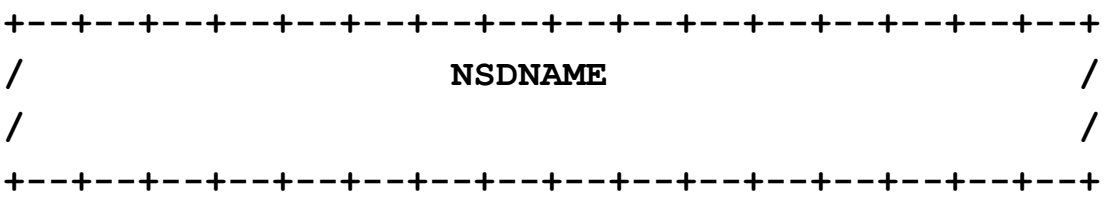


where:

PREFERENCE	A 16 bit integer which specifies the preference given to this RR among others at the same owner. Lower values are preferred.
EXCHANGE	A <domain-name> which specifies a host willing to act as a mail exchange for the owner name.

MX records cause type A additional section processing for the host specified by EXCHANGE. The use of MX RRs is explained in detail in [RFC-974].

3.3.11. NS RDATA format

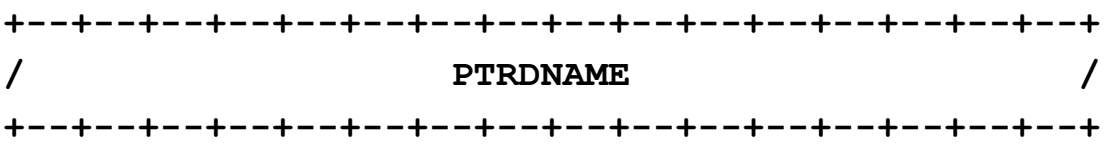


where:

NSDNAME A <domain-name> which specifies a host which should be authoritative for the specified class and domain.

NS records cause both the usual additional section processing to locate a type A record, and, when used in a referral, a special search of the zone in which they reside for glue information.

3.3.12. PTR RDATA format

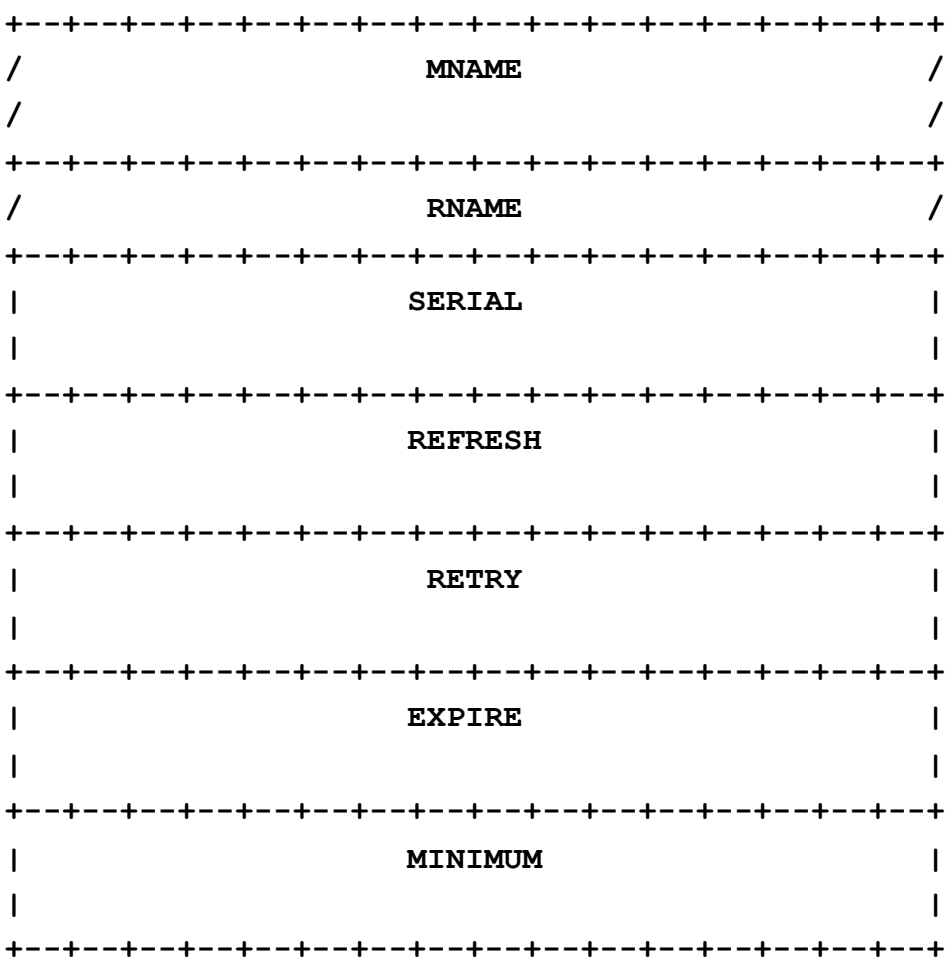


where:

PTRDNAME A <domain-name> which points to some location in the domain name space.

PTR records cause no additional section processing. These RRs are used in special domains to point to some other location in the domain space. These records are simple data, and don't imply any special processing similar to that performed by CNAME, which identifies aliases. See the description of the IN-ADDR.ARPA domain for an example.

3.3.13. SOA RDATA format



3.3.13. SOA RDATA format

- MNAME
  - このゾーンのデータのオリジナルあるいはプライマリであるネームサーバーのドメイン名。
  - RFC 2181 Clarifications to the DNS Specification "7.3. The SOA.MNAME field"
    - ・SOAレコードのMNAMEフィールドはゾーンのマスターサーバの名前を設定する。
    - ・ゾーン自体の名前を書くべきではない。

3.3.13. SOA RDATA format

- RNAME
  - このゾーンの責任者のメールアドレス。
  - 訳注）メールアドレスの"@"を"."に置き換える。
    ・例）"foo@example.com"は"foo.example.com"に。
- SERIAL
  - ゾーンのオリジナルコピーの符号無し32ビットバージョン番号。ゾーン転送はこの値を維持する。この値は周回し、sequence space arithmeticを使って比較する。
  - RFC 1982 Serial Number Arithmeticで比較について詳細な説明がある。

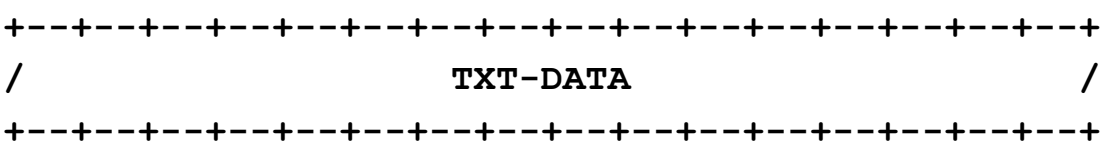
3.3.13. SOA RDATA format

- REFRESH
  - ゾーンをリフレッシュするまでの32ビット時間間隔。
- RETRY
  - 失敗したリフレッシュを再試行するまでの32ビット時間間隔。
- EXPIRE
  - ゾーンが権威をなくすまでの時間の上限を示す32ビットの時間の値。

3.3.13. SOA RDATA format

- MINIMUM
  - このゾーンのRRを送信するときに指定されるTTLの最小値。符号無し32ビット。
  - RFC 2308 Negative Caching of DNS Queries (DNS NCACHE)により否定応答のキャッシュのTTLとして再定義された。

3.3.14. TXT RDATA format

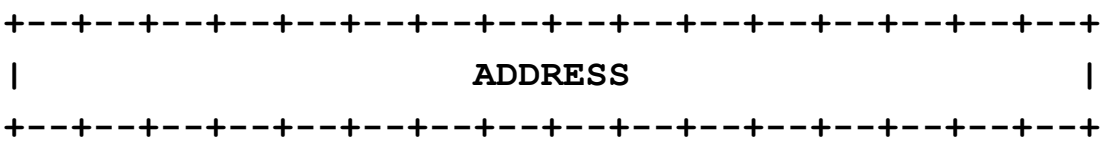


where:

TXT-DATA One or more <character-string>s.

TXT RRs are used to hold descriptive text. The semantics of the text depends on the domain where it is found.

3.4. Internet specific RRs
3.4.1. A RDATA format



where:

ADDRESS A 32 bit Internet address.

Hosts that have multiple Internet addresses will have multiple A records.

A records cause no additional section processing. The RDATA section of an A line in a master file is an Internet address expressed as four decimal numbers separated by dots without any imbedded spaces (e.g., "10.2.0.52" or "192.0.5.6").

3.5. IN-ADDR.ARPA domain

- 概要
  - IN-ADDR.ARPAドメインは逆引き（IPアドレスからホスト名へのマッピング）のために用いられる。

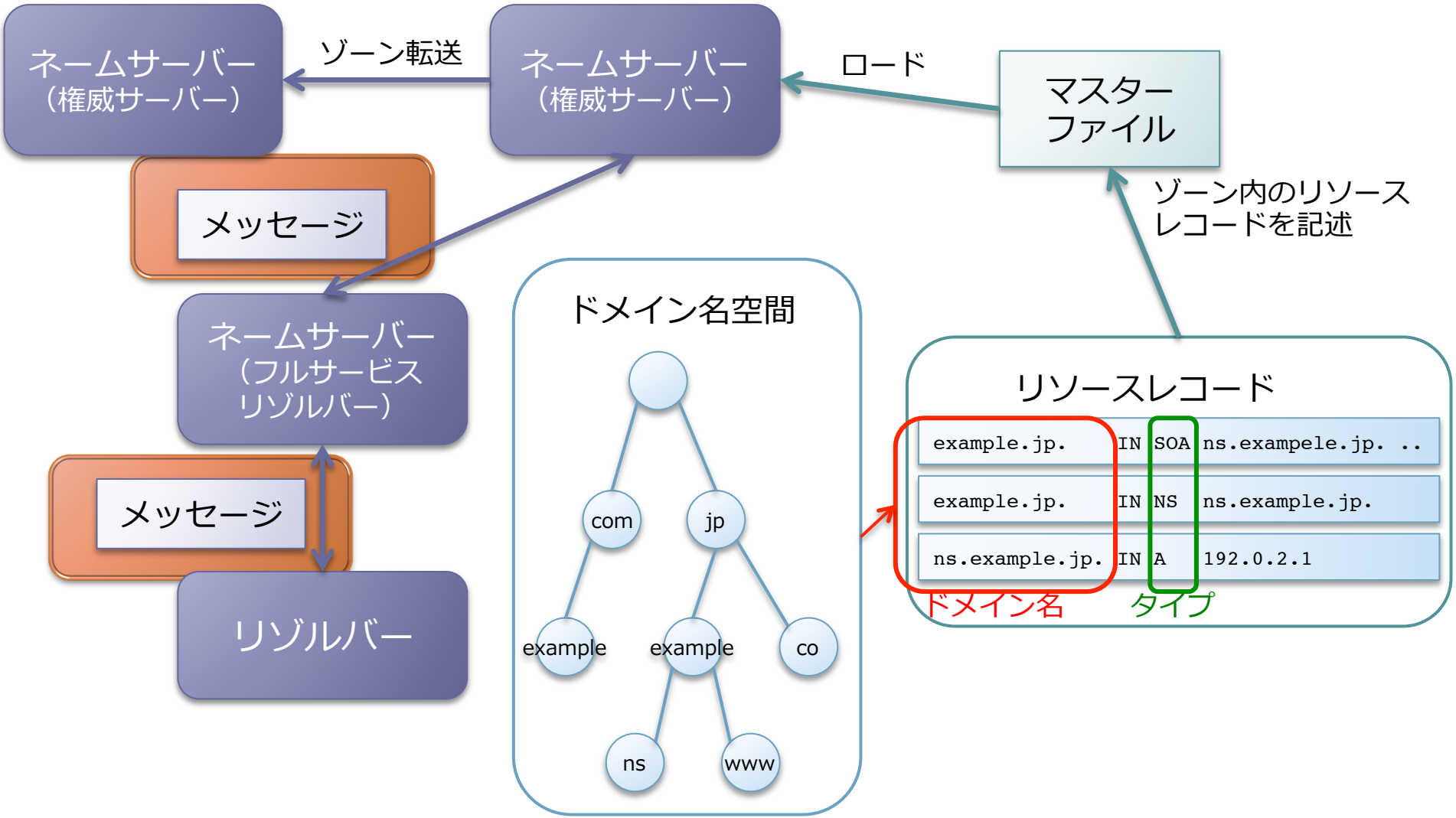
3.6. Defining new types, classes, and special namespaces

- 概要
  - 新しいタイプやクラスや特別な名前空間の定義の仕方を説明。

4. MESSAGES
4. メッセージ

- 4.1. Format
  フォーマット
- 4.2. Transport
  転送

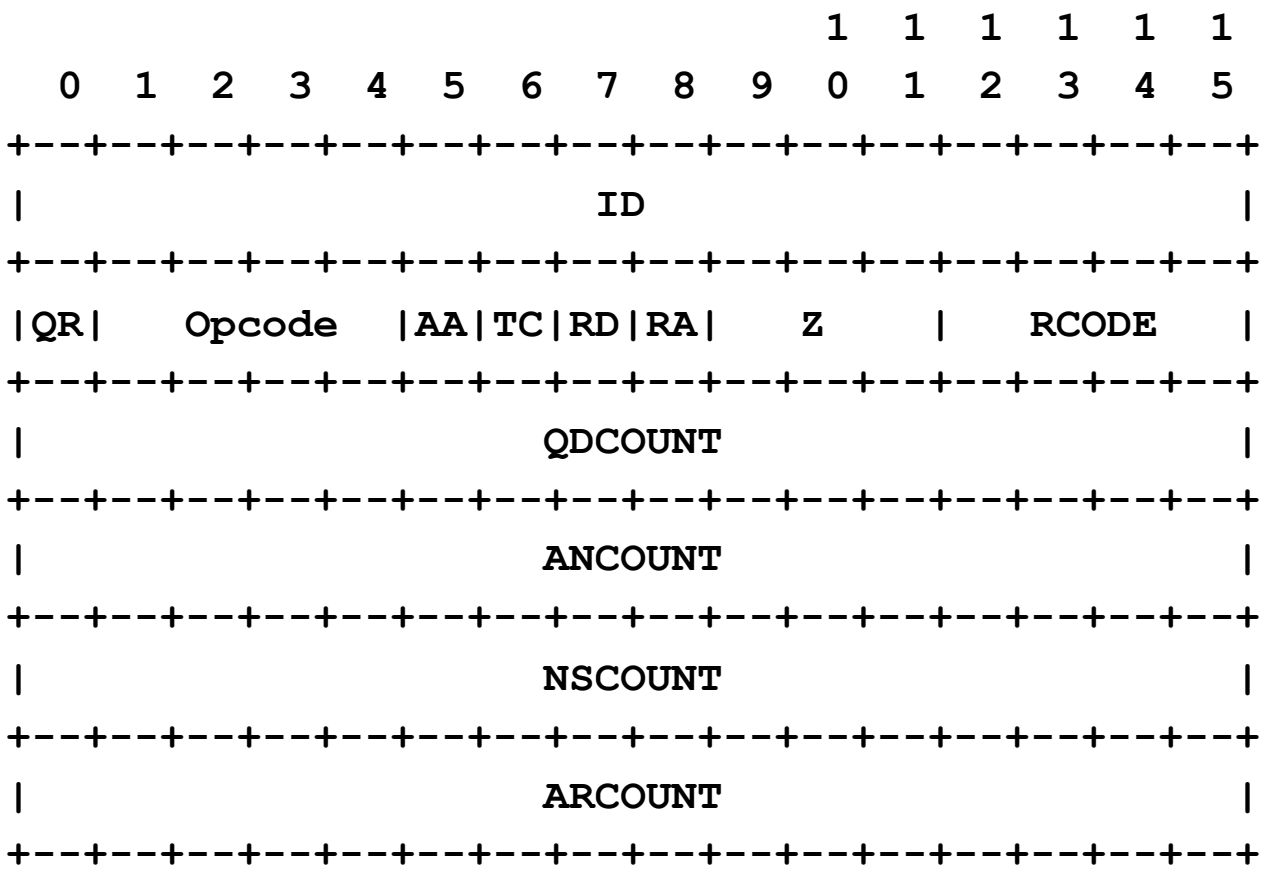
4章の位置づけ


ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

4.1. Format
4.1. フォーマット

```
+---------------------+
|        Header       |
+---------------------+
|       Question      | the question for the name server
+---------------------+
|        Answer       | RRs answering the question
+---------------------+
|      Authority      | RRs pointing toward an authority
+---------------------+
|      Additional     | RRs holding additional information
+---------------------+
```

4.1.1. Header section format
4.1.1. Headerセクションフォーマット



4.1.1. Header section format
4.1.1. Headerセクションフォーマット

- ID
  - 16ビットの識別子。
- QR
  - Query(0)かResponse(1)かを示す1ビット
- OPCODE
  - 問い合わせの種類を示す4ビット。
- AA
  - Authoritative Answer（権威ある回答）であるかを示す1ビット。
  - 対応したネームサーバーがquestionセクションのドメイン名に対する権威を持っているかを示す。

4.1.1. Header section format
4.1.1. Headerセクションフォーマット

- TC
  - TrunCation – メッセージが大きくて切り詰められたことを示す。
- RD
  - Recursion Desired – ネームサーバーに再帰問い合わせをすることを指示する。
- RA
  - Recursion Available – ネームサーバーが再帰問い合わせをサポートしているかを示す。
- Z
  - 将来のための予約。0にする。

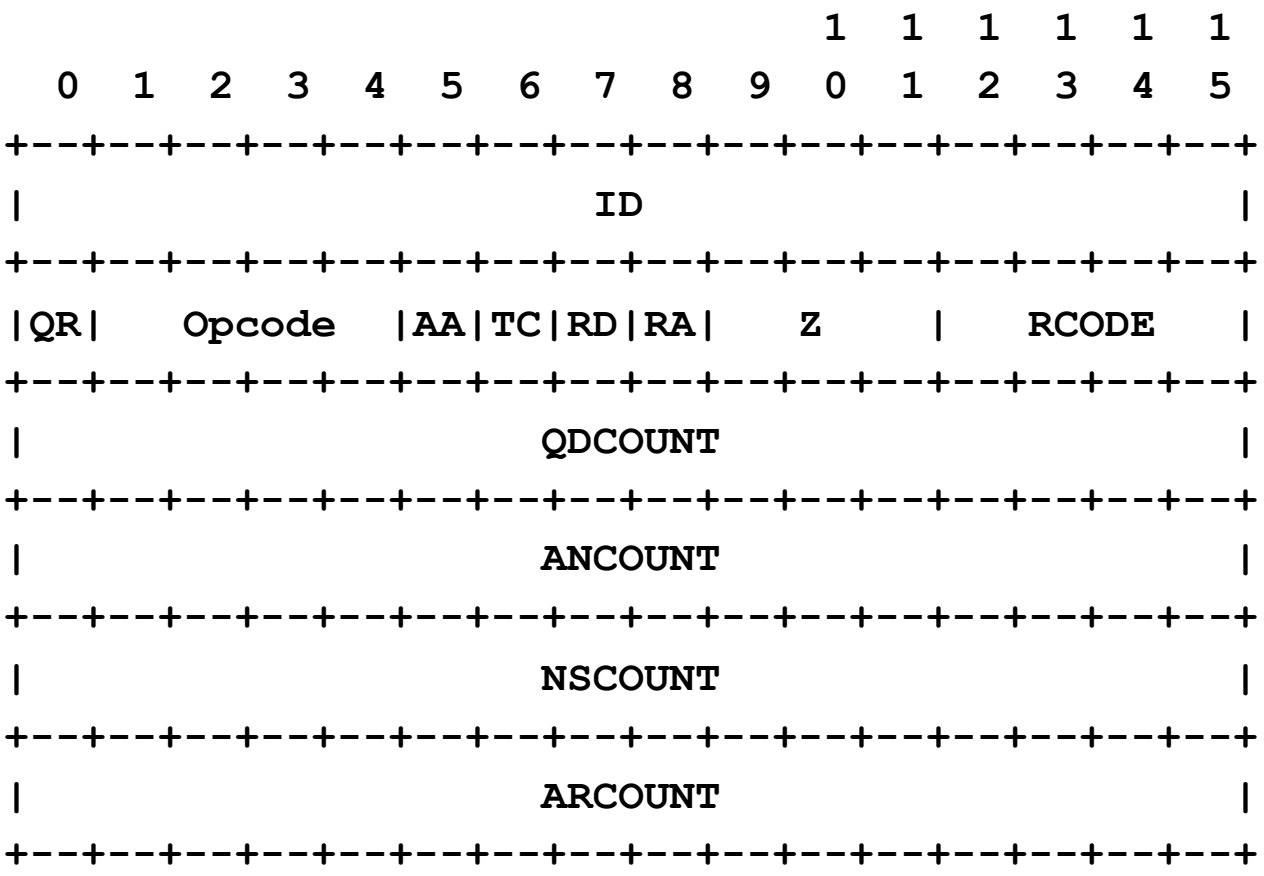
4.1.1. Header section format
4.1.1. Headerセクションフォーマット

- RCODE
  - Response code – 応答コード
    - 0: No error condition
    - 1: Format error
    - 2: Server failure
    - 3: Name Error
    - 4: Not Implemented
    - 5: Refused
    - 6-15: Reserved for future use.

4.1.1. Header section format
4.1.1. Headerセクションフォーマット

- QDCOUNT
  - questionセクションのエントリーの数を示す符号無し16ビット整数
- ANCOUNT
  - answerセクションのエントリーの数を示す符号無し16ビット整数
- NSCOUNT
  - authorityレコードセクションのリソースレコードの数を示す符号無し16ビット整数
- ARCOUNT
  - additionalレコードセクションのリソースレコードの数を示す符号無し16ビット整数

4.1.1. Header section format
4.1.1. Headerセクションフォーマット



4.1.1. Header section format
4.1.1. Headerセクションフォーマット

```
$ dig @127.0.0.1 emaillab.jp. NS
略
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33981
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;emaillab.jp.                   IN      NS

;; ANSWER SECTION:
emaillab.jp.            86400   IN      NS      ns.emaillab.jp.
emaillab.jp.            86400   IN      NS      ns2.emaillab.jp.

;; ADDITIONAL SECTION:
ns.emaillab.jp.         86400   IN      A       49.212.17.32
ns2.emaillab.jp.        86400   IN      A       49.212.24.158
```

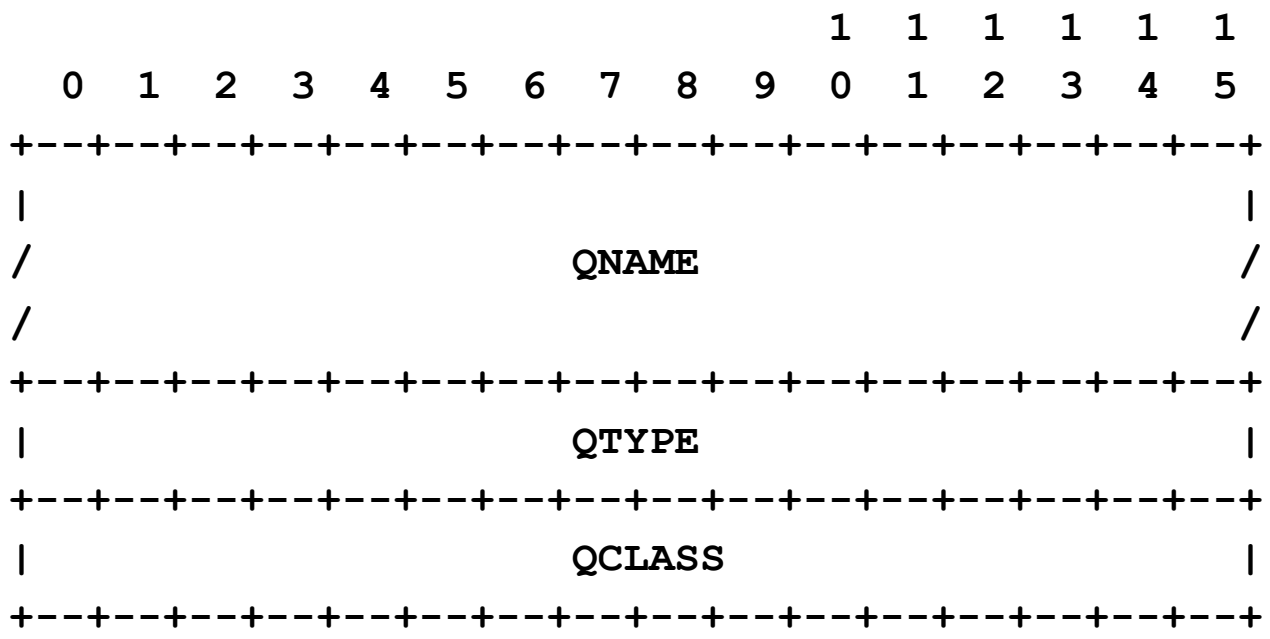
4.1.1. Header section format
4.1.1. Headerセクションフォーマット

```
$ dig +norec @ns.emaillab.jp. emaillab.jp. NS
略
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24693
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;emaillab.jp.                   IN      NS

;; ANSWER SECTION:
emaillab.jp.            86400   IN      NS      ns.emaillab.jp.
emaillab.jp.            86400   IN      NS      ns2.emaillab.jp.

;; ADDITIONAL SECTION:
ns.emaillab.jp.         86400   IN      A       49.212.17.32
ns2.emaillab.jp.        86400   IN      A       49.212.24.158
```

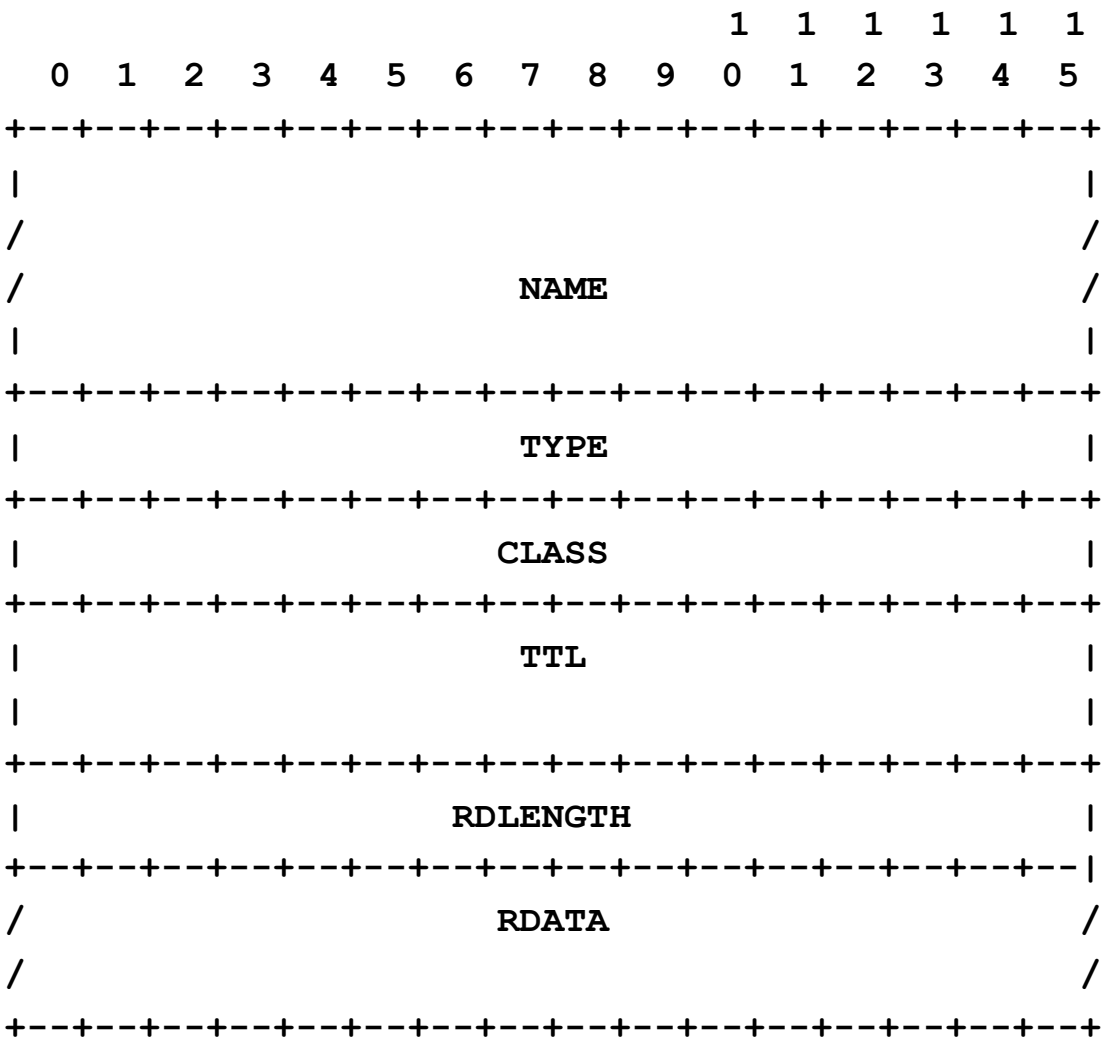
4.1.1. Header section format
4.1.1. Headerセクションフォーマット

```
$ dig +norec @a.dns.jp. emaillab.jp. NS

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35302
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;emaillab.jp.                   IN      NS

;; AUTHORITY SECTION:
emaillab.jp.            86400   IN      NS      ns.emaillab.jp.
emaillab.jp.            86400   IN      NS      ns2.emaillab.jp.

;; ADDITIONAL SECTION:
ns.emaillab.jp.         86400   IN      A       49.212.17.32
ns2.emaillab.jp.        86400   IN      A       49.212.24.158
```

4.1.2. Question section format
4.1.2. Questionセクションフォーマット



4.1.3. Resource record format
4.1.3. リソースレコードフォーマット



4.2. Transport
4.2. 転送

- 概要
  - DNSのメッセージがUDPあるいはTCPで転送されることを説明している。

4.2.1. UDP usage

- メッセージは512バイトに制限される。
- 512バイトより大きいメッセージは切り詰められ、ヘッダーにTCビットを立てる。
- RFC 2181 Clarifications to the DNS Specification "9 The TC (truncated) header bit"にTCビットがどういうときにセットされるかについての説明がある。
- RFC 2671 Extension Mechanisms for DNS (EDNS0) により、UDPで512バイトより大きいメッセージを送ることが可能である。

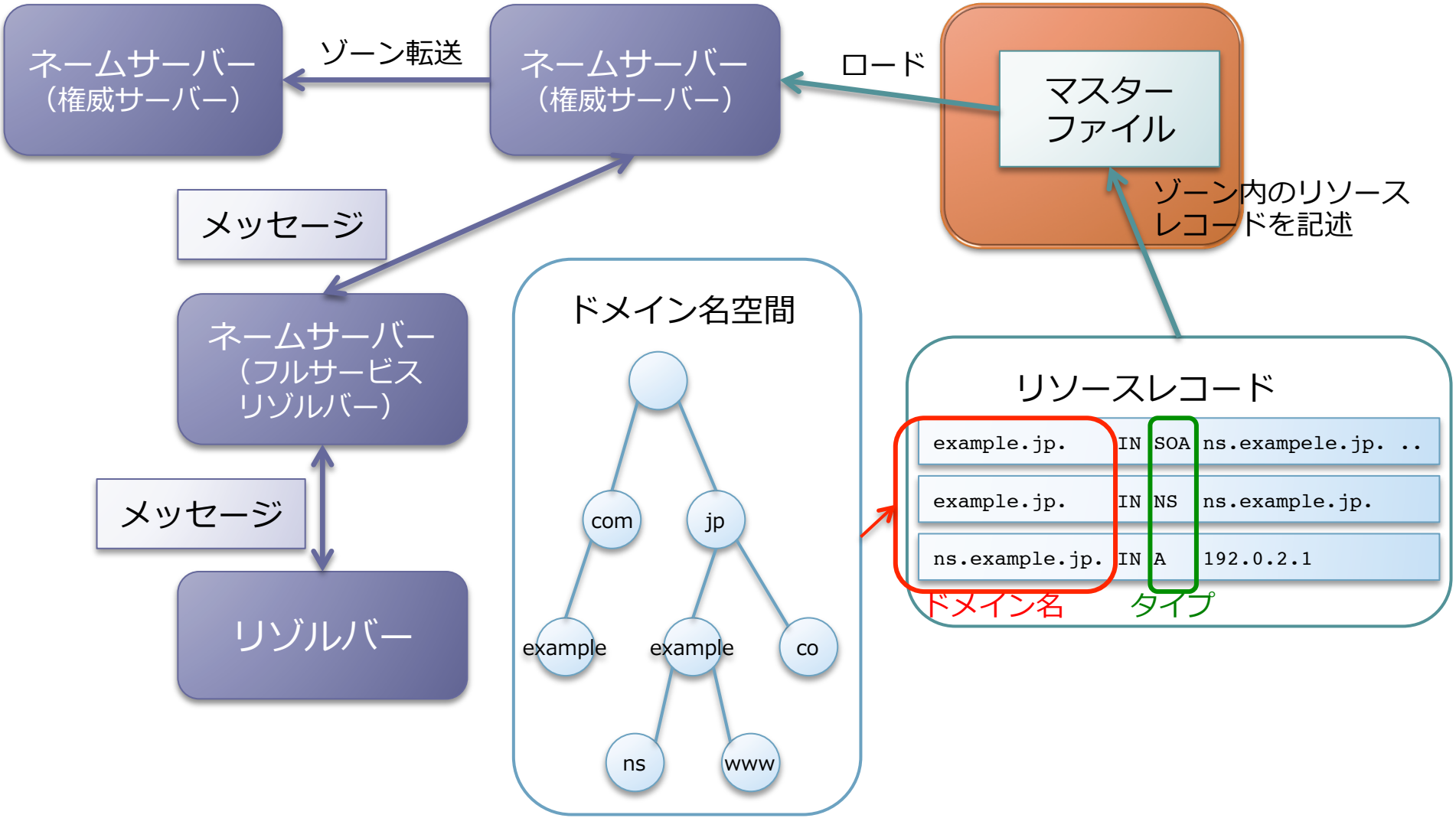
4.2.2. TCP usage

- RFC 5966 DNS Transport over TCP - Implementation RequirementsによりDNS over TCPの仕様が更新されている。

5. MASTER FILES
5. マスターファイル

- 5.1. Format
- 5.2. Use of master files to define zones
- 5.3. Master file example

5章の位置づけ


ネームサーバー
（権威サーバー）
ゾーン転送
ネームサーバー
（権威サーバー）
ロード
マスター
ファイル
ゾーン内のリソース
レコードを記述
メッセージ
ネームサーバー
（フルサービス
リゾルバー）
メッセージ
リゾルバー
ドメイン名空間
com
jp
example
example
co
ns
www
リソースレコード
example.jp. IN SOA ns.exampele.jp. ..
example.jp. IN NS ns.example.jp.
ns.example.jp. IN A 192.0.2.1
ドメイン名
タイプ

5.1. Format

- エントリーは行毎に。
- アイテムが行を跨がる場合には括弧（）で囲む。
- アイテム間のデリミターとしてスペースとタブおよびその組み合わせが使える。
- 行の終わりはコメントで終わることができる。
- コメントは";"で始まる。
- @はオリジンを元に戻す。

5.1. Format

- エントリーの形式
  - <blank>[<comment>]
  - $ORIGIN <domain-name> [<comment>]
  - $INCLUDE <file-name> [<domain-name>] [<comment>]
  - <domain-name><rr> [<comment>]
  - <blank><rr> [<comment>]
- <rr>の定義
  - [<TTL>] [<class>] <type> <RDATA>
  - [<class>] [<TTL>] <type> <RDATA>

6. NAME SERVER IMPLEMENTATION
6. ネームサーバーの実装

- 6.1. Architecture
- 6.2. Standard query processing
- 6.3. Zone refresh and reload processing
- 6.4. Inverse queries (Optional)
- 6.5. Completion queries and responses

- 本章はネームサーバーの開発者向けの内容であり、チュートリアルには向かない内容なので、説明は省略する。

7. RESOLVER IMPLEMENTATION
7. リゾルバーの実装

- 7.1. Transforming a user request into a query
- 7.2. Sending the queries
- 7.3. Processing responses
- 7.4. Using the cache

- 本章はネームサーバーの開発者向けの内容であり、チュートリアルには向かない内容なので、説明は省略する。

8. MAIL SUPPORT
8. メールサポート

- 省略

9. REFERENCES and BIBLIOGRAPHY
9. 出典と参考文献

- 省略

DNSの基本仕様のアップデート

RFC 1123

Requirements for Internet Hosts -- Application and Support

- タイトル
  - Requirements for Internet Hosts -- Application and Support
  - インターネットホストの要求事項 – アプリケーションとサポート
- 概要
  - インターネットに接続するホストが満たすべき要求事項を定めている。
  - もちろん、ホスト名やDNSのサポートについても

RFC 1123

Requirements for Internet Hosts -- Application and Support

- 2.1 Host Names and Numbers
  ホスト名と数
  - ホスト名の文法の改訂（RFC 952の改訂）
    - ホスト名の最初の文字は英文字のみであったが、数字も使えるようになった。
    - ホスト名の長さは24文字までであったが、下記のようになった。
  - ホスト名
    - 英文字あるいは数字で始まる
    - 英文字あるいは数字で終わる
    - 間の文字は英文字、数字、ハイフンが使える
  - ホストソフトウェアは
    - 63文字までのホスト名を扱わなければならない（MUST）
    - 255文字までのホスト名を扱うべきである（SHOULD）

RFC 1982
Serial Number Arithmetic

- タイトル
  - Serial Number Arithmetic
  - シリアル番号演算
- 概要
  - RFC 1034とRFC 1035ではゾーン転送におけるSOAレコードのシリアル番号の比較に"sequence space arithmetic"を使うと記述されている。
  - しかし、"sequence space arithmetic"そのものの定義がない。
  - RFC 1982ではその定義を行う。

RFC 1982
Serial Number Arithmetic

- おおざっぱにまとめると
  - シリアル番号の扱い
    - 32ビットの符号無し整数
    - 範囲は[0 .. 4294967295]（2^32 - 1）
    - 最大値から0に周回する。つまり、4294967295+1は0
    - 増加の範囲は[0 .. 2147483647]（2^31 - 1）
  - シリアル番号の増分が増加の最大値以下であれば「シリアル番号の増加(increment)」と判断する。
    - 例）「4294967295 → 0」は増分が1なので、「増加」と判断できる。

RFC 1982
Serial Number Arithmetic

- シリアル番号の保守
  - RFC 2182 "Selection and Operation of Secondary DNS Servers"
    - "7. Serial Number Maintenance"
      - 間違って大きすぎる番号を付けてしまった場合に、小さい番号に付け直す方法が書いてある。

RFC 2181
Clarifications to the DNS Specification

- タイトル
  - Clarifications to the DNS Specification
  - DNSの仕様の明確化
- 概要
  - この文書はDNSの仕様の問題として確認されている分野について考察し、確認された欠陥の改善を提案する。

RFC 2181
Clarifications to the DNS Specification

- 4. Server Reply Source Address Selection
  - マルチホームのサーバにおける応答の送信元アドレスの選択方法
- 4.1. UDP Source Address Selection
  - 問い合わせの宛先アドレスを応答の送信元アドレスとしてセットする。
- 4.2. Port Number Selection
  - 問い合わせの送信元ポート番号を応答の宛先ポート番号として使う。
  - 応答は指示されたポート番号から送信される。
    - DNSのウェルノウンポート、53

RFC 2181
Clarifications to the DNS Specification

- 5. Resource Record Sets
  - 同じラベル、クラス、タイプのリソースレコードの集まりを「リソースレコードセット」（RRset）と定義する。
  - 問い合わせに対して、関連するRRが複数あるときには、RRsetのすべてのRRを返す。
  - RRsetのすべてのRRは同じTTLを使う。

RFC 2181

Clarifications to the DNS Specification

- 5.4.1. Ranking data
  - データの信頼性の順位について論じている。下に行くほど低い。
    - Data from a primary zone file, other than glue data,
    - Data from a zone transfer, other than glue,
    - The authoritative data included in the answer section of an authoritative reply.
    - Data from the authority section of an authoritative answer,
    - Glue from a primary zone, or glue from a zone transfer,
    - Data from the answer section of a non-authoritative answer, and non-authoritative data from the answer section of authoritative answers,
    - Additional information from an authoritative answer,
      Data from the authority section of a non-authoritative answer,
      Additional information from non-authoritative answers.

RFC 2181
Clarifications to the DNS Specification

- 6. Zone Cuts
  - zone
  - zone cut
  - "child" zone
  - "parent" zone
  - zone's "origin"
- 6.1. Zone authority
  - ゾーンの権威について

RFC 2181
Clarifications to the DNS Specification

- 7. SOA RRs
  - SOAレコードに関する記述を3つ修正

RFC 2181

Clarifications to the DNS Specification

- 7.1. Placement of SOA RRs in authoritative answers
  - RFC 1034 "4.3.4 Negative response caching (Optional)"の内容が間違っている。
    - The method is that a name server may add an SOA RR to the additional section of a response when that response is authoritative.
  - ネガティブキャッシュのときにもSOAレコードをauthorityセクションにおいて回答する。
    - SOA records, if added, are to be placed in the authority section.

RFC 2181

Clarifications to the DNS Specification

- 7.2. TTLs on SOA RRs
  - RFC 1035 "3.2.1. Format"の内容が間違っている。
    - SOA records are always distributed with a zero TTL to prohibit caching.
- 7.2. TTLs on SOA RRs
  - どこにも言及されていないし、そのような実装は行われていない。実装はTTLが0であることを想定すべきではないし、0のTTLを持つSOAレコードを送信することを要求してもいけない。
    - This is mentioned nowhere else, and has not generally been implemented. Implementations should not assume that SOA records will have a TTL of zero, nor are they required to send SOA records with a TTL of zero.

RFC 2181
Clarifications to the DNS Specification

- 7.3. The SOA.MNAME field
  - 仕様では明確になっているが広く無視されている。
  - SOAレコードのMNAMEフィールドはゾーンのマスターサーバの名前を設定する。
  - ゾーン自体の名前を書くべきではない。

RFC 2181

Clarifications to the DNS Specification

- 8. Time to Live (TTL)
  - RFC 1035におけるTTLの定義
    - 2.3.4. Size limits
      - positive values of a signed 32 bit number.
    - 3.2.1. Format
      - a 32 bit signed integer that specifies the time interval
    - 4.1.3. Resource record format
      - a 32 bit unsigned integer that specifies the time interval
  - 明確にする
    - 符号無し整数
    - 最小値: 0
    - 最大値: 2147483647 (2^31 - 1)
    - 最上位ビットが1であるときにはTTLを0と扱うべき

RFC 2181
Clarifications to the DNS Specification

- 9. The TC (truncated) header bit
  - TCビットについての説明

RFC 2181
Clarifications to the DNS Specification

- 10. Naming issues

RFC 2181
Clarifications to the DNS Specification

- 10.1. CNAME resource records
  - CNAMEの意味を明確化
    - CNAME ("canonical name")「正式名」は"alias name"「別名」と関連づけるために使う
    - CNAMEは「別名」を示すのでなく、「別名」に対する「正式名」を示す。
      - 別名 IN CNAME 正式名

RFC 2181
Clarifications to the DNS Specification

- 10.2. PTR records
  - PTRレコードは一つだけ持つというのは誤り。
  - PTRレコードの値には（CNAMEで定義される）別名であってならない。

RFC 2181
Clarifications to the DNS Specification

- 10.3. MX and NS records
  - MXレコードとNSレコードの値は（CNAMEで定義される）別名であってはならない。

RFC 2181
Clarifications to the DNS Specification

- 11. Name syntax
  - DNSはホスト名からデータへ、IPアドレスからホスト名へとマッピングするためだけのものではない。
  - DNSは汎用的な階層型データベースであり、様々なデータを様々な目的で格納できる。

RFC 2308
Negative Caching of DNS Queries (DNS NCACHE)

- タイトル
  - Negative Caching of DNS Queries (DNS NCACHE)
  - DNS問い合わせのネガティブキャッシュ
- 概要
  - ネガティブキャッシュについての詳細な説明と再定義を行っている。

RFC 2308

Negative Caching of DNS Queries (DNS NCACHE)

- RFC 1034からの変更点（8 - Changes from RFC 1034）
  - ネガティブキャッシュはoptionalであったが、RFC 2308ではキャッシュする場合は、ネガティブキャッシュもしなければならないよう（must）になった。
  - AuthorityセクションのSOAレコードはキャッシュしなければならない（MUST）。
  - キャッシュしたSOAレコードは応答に加えなければならない（MUST）。
- SOA Minimumフィールド
  - 否定応答のTTLとして使われる。
  - 元々別の意味で定義されていたが再定義された。

RFC 4343
Domain Name System (DNS) Case Insensitivity Clarification

- タイトル
  - Domain Name System (DNS) Case Insensitivity Clarification
  - DNSの大文字小文字を区別しないことの明確化
- 概要
  - ドメイン名の大文字小文字を区別しない。
    - ASCII文字が対象
  - この文書はその意味を説明し、仕様を明確にする。

RFC 4343

Domain Name System (DNS) Case Insensitivity Clarification

- 問い合わせにおいては大文字小文字を区別すべきでない。
  - 次の2つのドメイン名の問い合わせは同じ結果となる。
    - foo.example.net.
    - Foo.ExamplE.net.
  - 実装
    - 比較前に0x61-0x7Aの範囲の文字（小文字）であれば0x20を引く。
- 大文字小文字の維持
  - 可能な限りオリジナルの大文字小文字を維持すべきである。

RFC 4343
Domain Name System (DNS) Case Insensitivity Clarification

- ラベルのエスケープ方法
  - DNSのプロトコル上は8ビット文字も使える。
  - 文字コードが0x00-0x20,0x2E(.),0x7F-0xFFである文字をバックスラッシュでエスケープする記法
    - バックスラッシュ(¥) + 文字コードの3桁の十進数
    - バックスラッシュ(¥) + 文字（ASCII印字可能文字の場合）
  - 例
    - バックスラッシュ"\" → "\092" or "\\"
    - ピリオド"." → "\046" or "\."
    - スペース" " → "\032"

RFC 4343
Domain Name System (DNS) Case Insensitivity Clarification

- Use of Bit 0x20 in DNS Labels to Improve Transaction Identity
  - というInternet Draftがかつて提案されていた。
  - 通称"dns 0x20"
    - 大文字のコードに対して0x20ビットを立てると小文字のコードになる。
      - 0x41 (A) + 0x20 = 0x61 (a)
  - キャッシュポイズニングへの耐性を向上させることを目的とする。
  - 問い合わせとその回答において大文字小文字を維持する仕様を利用する。

RFC 4343
Domain Name System (DNS) Case Insensitivity Clarification

- Use of Bit 0x20 in DNS Labels to Improve Transaction Identity
  - リゾルバーは問い合わせに大文字小文字をランダムに設定する。
    - FoO.ExAmplE.nEt.
  - 権威サーバーは問い合わせの名前を大文字小文字を維持したままコピーして回答する。
    - FoO.ExAmplE.nEt.
  - リゾルバーは問い合わせと回答の文字列を比較して同じであれば信用する。異なれば偽の回答と判断する。

RFC 5452

Measures for Making DNS More Resilient against Forged Answers

- タイトル
  - Measures for Making DNS More Resilient against Forged Answers
  - DNSを偽の回答に対する耐性の強化方法
- 概要
  - DNSを不正な応答を受け付けることに対して耐性を強化する方法について説明する。

RFC 5452

Measures for Making DNS More Resilient against Forged Answers

- 9.1. Query Matching Rules
  - RFC 2181 "5.4.1. Ranking data"のDNSの信頼性規則を適応する前に
  - リゾルバーの実装は次のような問い合わせの属性のすべてに、応答が一致していなければならない（MUST）。
    - 問い合わせの宛先アドレスに対する送信元アドレス
    - 問い合わせの送信元アドレスに対する宛先アドレス
    - 問い合わせの送信元ポートに対する宛先ポート
    - 問い合わせID
    - 問い合わせの名前
    - 問い合わせのクラスとタイプ
  - 一致しない応答は不正と判断しなければならない（MUST）。

RFC 5452

Measures for Making DNS More Resilient against Forged Answers

- 9.2. Extending the Q-ID Space by Using Ports and Addresses
  - リゾルバーの実装（MUST）：
    - 外部への問い合わせには、実際に利用できて可能な限り大きい範囲のポート番号から予測不能な送信元ポート番号を使う
    - 同時に複数の問い合わせが起こる場合には、複数の異なる送信元ポート番号を使う
    - 外部への問い合わせには、全範囲（0-65535）を利用して、予測不能な問い合わせIDを使う。

RFC 5452

Measures for Making DNS More Resilient against Forged Answers

- 9.3. Spoof Detection and Countermeasure
  - リゾルバは詐称を検出したら、UDP問い合わせを破棄し、TCPで再発行させてもよい（MAY）
    - ・TCPは詐称に対する耐性が強い

RFC 5936
DNS Zone Transfer Protocol (AXFR)

- タイトル
  - DNS Zone Transfer Protocol (AXFR)
  - DNSのゾーン転送プロトコル (AXFR)
- 概要
  - RFC 1034とRFC 1035におけるAXFRの定義が不十分であり、仮定して実装しなければならないところがあり、相互運用性を阻害している。
  - この文書はAXFRの新しい定義である。
  - 「新しい」は相互運用できるAXFR機能の正確な定義を記録する的な意味合いで。

RFC 5966

DNS Transport over TCP - Implementation Requirements

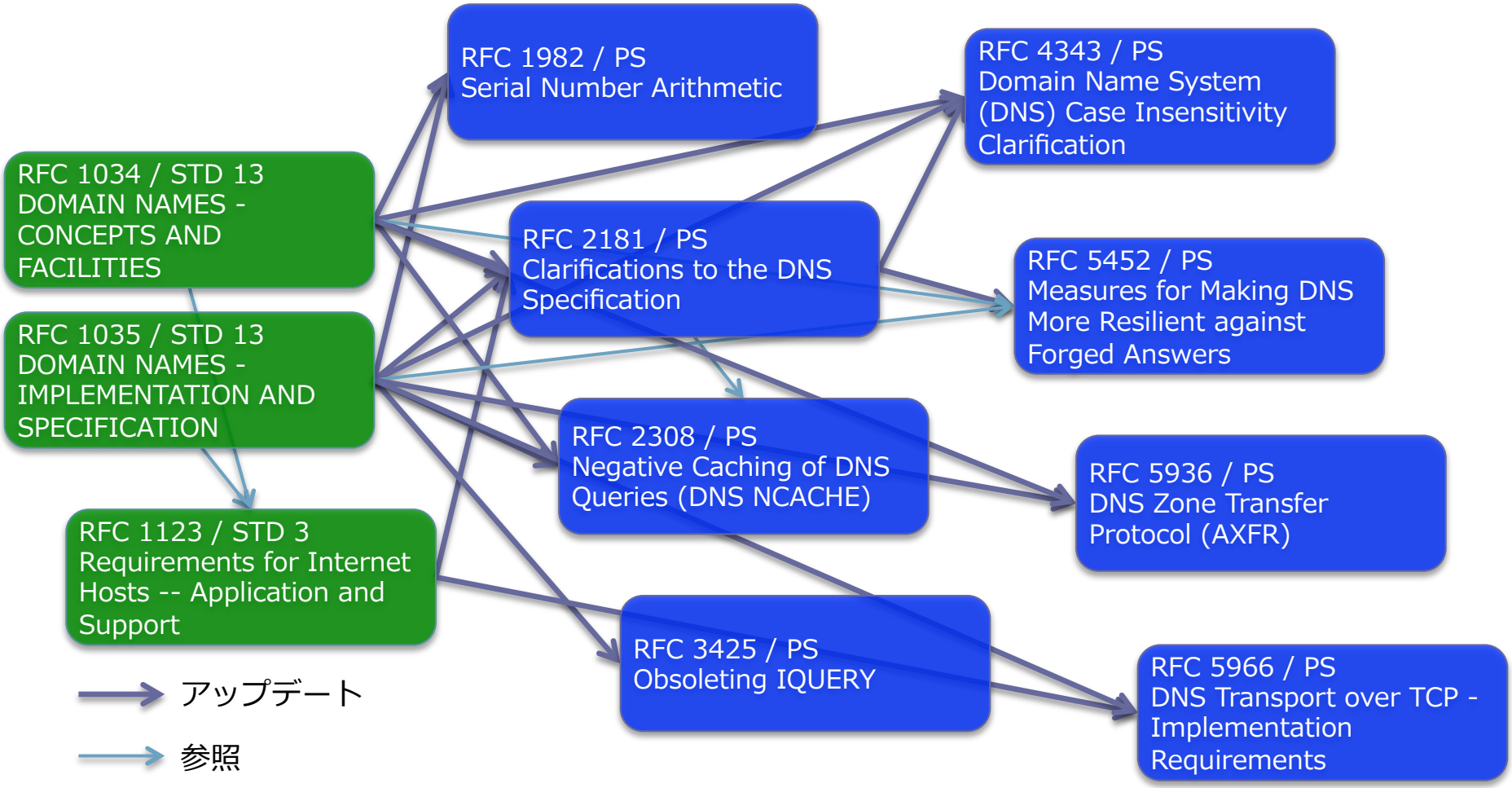
- タイトル
  - DNS Transport over TCP - Implementation Requirements
  - DNSのTCP転送 – 実装と要求事項
- 概要
  - この文書はDNSのTCPでの転送の要求事項を更新する。
  - TCPサポートがSHOULDであったのをMUSTにする。

RFC 5966

DNS Transport over TCP - Implementation Requirements

- 4. Transport Protocol Selection
  - すべての通常の目的のDNSの実装はUDPとTCPの両方をサポートしなければならない（MUST）
    - 権威サーバーの実装は、応答のサイズを一つのUDPパケットに収めるサイズに制限しないようにTCPをサポートしなければならない（MUST）。
    - 再帰検索サーバー（あるいはフォワーダー）の実装は、TCPを利用できるサーバーからの大きなサイズの応答をTCPが利用できるクライアントに達するのを妨げないように、TCPをサポートしなければならない（MUST）。
    - スタブリゾルバーの実装は、自身のクライアントと上流のサーバーとの相互運用性を制限しないように応答サイズを、TCPをサポートしなければならない（MUST）。

DNSの基本仕様およびアップデート（再掲）



DNSの拡張仕様（広く使われているもの）

RFC 2671
Extension Mechanisms for DNS (EDNS0)

- タイトル
  - Extension Mechanisms for DNS (EDNS0)
  - DNSの拡張機構（EDNS0）
- 概要
  - DNSのプロトコルを機能する。
  - これにより、UDPで512オクテットを超えるメッセージを扱うことができる。

RFC 1995
Incremental Zone Transfer in DNS

- タイトル
  - Incremental Zone Transfer in DNS
  - DNSの差分ゾーン転送
- 概要
  - DNSのプロトコルにIXFR（差分ゾーン転送）機能を提供するための拡張を提案する。

RFC 1996

A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)

- タイトル
  - A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)
  - ゾーン変更の通知機能（DNS NOTIFY）
- 概要
  - マスターサーバーがスレーブサーバーにゾーンの変更を通知するNOTIFY opcodeを提案する。

RFC 2845

Secret Key Transaction Authentication for DNS (TSIG)

- タイトル
  - Secret Key Transaction Authentication for DNS (TSIG)
  - DNSの秘密鍵によるトランザクション認証（TSIG）
- 概要
  - 共有秘密鍵を一方向ハッシュを使ったトランザクションレベルの認証を提案する。

RFC 3645

Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)

- タイトル
  - Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)
  - DNSの秘密鍵トランザクション認証のためのGSS-API（GSS-TSIG）
- 概要
  - TSIGでGSS-APIを使う拡張を提案する。

RFC 4635
HMAC SHA TSIG Algorithm Identifiers

- タイトル
  - HMAC SHA TSIG Algorithm Identifiers
  - HMAC SHA TSIGアルゴリズム識別子
- 概要
  - TSIGでHMAC SHAを扱う方法を提案する。

RFC 2136
Dynamic Updates in the Domain Name System (DNS UPDATE)

- タイトル
  - Dynamic Updates in the Domain Name System (DNS UPDATE)
  - DNSにおける動的更新（DNS UPDATE）
- 概要
  - UPDATE opcodeを使って、指定したゾーンのRRやRRsetの追加や削除を行う方法を提案する。

RFC 3007
Secure Domain Name System (DNS) Dynamic Update

- タイトル
  - Secure Domain Name System (DNS) Dynamic Update
  - セキュアなDNS動的更新
- 概要
  - DNS動的更新に認証機能を追加する。

おつかれさまでした。