IPsec と IPv6

IIJ 技術研究所
山本和彦
kazu@iijlab.net

内容

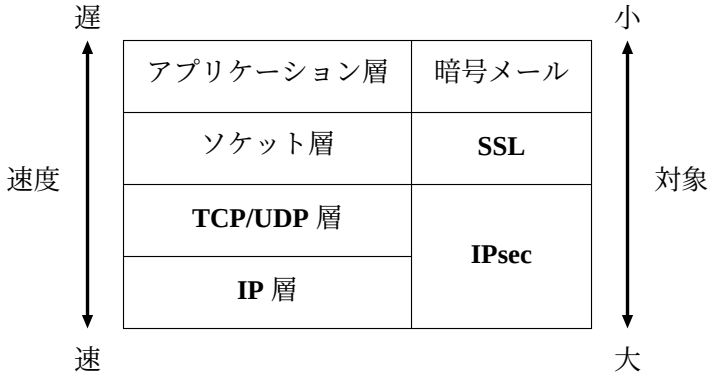
- IPsec
  - **設計思想、認証、暗号、鍵配送**
- IPv6
  - **設計思想、アドレス・アーキテクチャ、近隣探索**
- IPv6 **への移行**
  - 6bone**、トランスレータ**

IPsec

VPN と IPsec

- TCP/IP **レベルのセキュリティ**
  - **暗号ペイロード**(Encapsulating Security Payload)
  - **認証ヘッダ**(Authentication Header)
- VPN(Virtual Private Network)
  - **専用線で各部署を接続するのは高価**
  - **サイトの出口で暗号トンネルを張る**
  - **安価なインターネット上にプライベート・ネットワークを構築**

セキュリティの階層と粒度



IPsec の用途

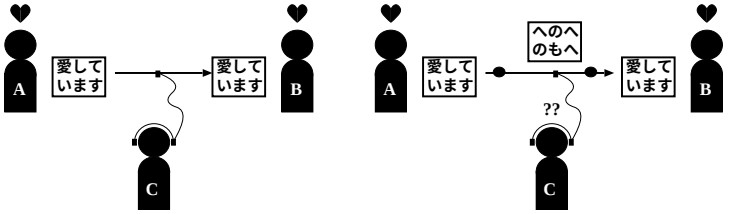
- □ネットワーク単位の通信保護
  - ○VPN の構築
- □ホスト単位の通信保護
- □TCP の保護
  - ○トランザクション指向の TCP の保護
  - ○TCP リセット攻撃の防止
- □UDP の保護
- □IP に直接載っているプロトコルの保護

インターネット・セキュリティの保護機能

□**保護できること**
- ○**機密性**(confidentiality)
- ○**完全性**(integrity)
- ○**認証**(authentication)
- ○**否認防止**(non-repudiation)

□**保護できないこと**
- ○**トラフィック解析の防止**(traffic analysis protection)
- ○**いやがらせの防止**(denial-of-service protection)

機密性

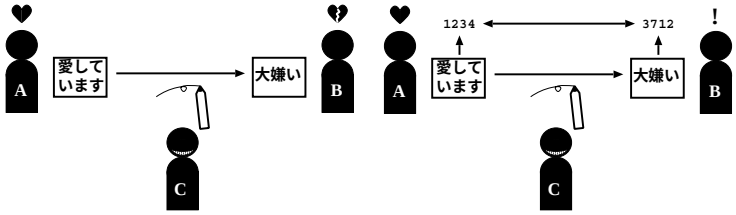


□**意図した相手のみが内容を理解できること**

□**第三者に内容を知られないこと**

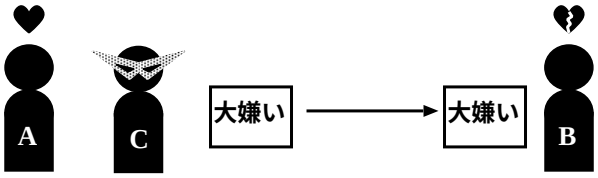
□**暗号で実現する**
- ○**鍵を共有することが本質**

完全性



□送信者が書いた内容がそのまま受信者に伝わること
□第三者による内容の改竄を検知できること
□実現方法
  ○電子署名
  ○MAC(Message Authentication Code)

認証



□ある名前を名乗る人が本当にその人だと確認すること
□その人だけが知っている秘密を確認する
  ○電子署名 - 公開鍵で秘密鍵を確認
□共有している秘密を確認する
  ○MAC(Message Authentication Code)

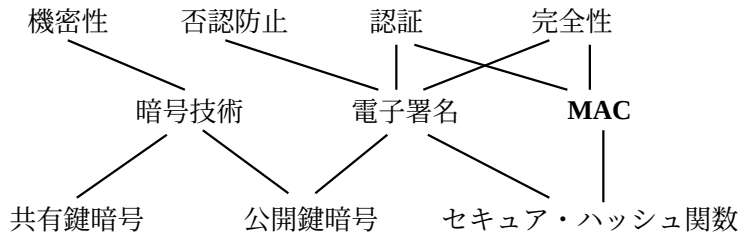
否認防止

□通信内容をあとから否定できないこと
- ○電子商取引には必ず必要

□電子署名で実現する
- ○認証
  - ▹本当にその人が書いた
  - ▹その人のみが書ける(MAC ではダメ)
- ○完全性
  - ▹内容は完全である

□秘密が漏洩する場合を想定しなければならないサービスもある
- ○必ず第三者を通して通信する

セキュリティ保護に求められる機能と技術

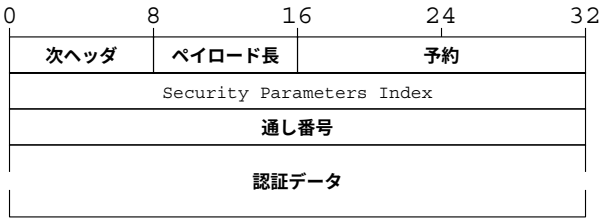


□インターネットでは保護できない項目
- ○トラフィック解析の防止
  - ▹通信のパターンから有益な情報を得ること
- ○いやがらせの防止

IPsec

- □認証ヘッダ
  - ○Authentication Header(AH)
  - ○完全性、認証、否認防止
  - ○暗号の輸出規制のため暗号ペイロードから切り離された
  - ○IP ヘッダも保護できる
- □暗号ペイロード
  - ○Encapsulating Security Payload(ESP)
  - ○機密性、完全性、認証、否認防止
- □変換
  - ○Transform
  - ○認証ヘッダや暗号ペイロードは枠組のみ提供
  - ○具体的な方式は変換で定める

認証ヘッダ



- □SPI
  - ○Security Parameters Index
  - ○方式は隠蔽されている
- □通し番号によるリプレイ攻撃の防止

MAC

- Message Authentication Code
- **認証と完全性の確認**
- **特殊なセキュア・ハッシュ関数を利用**
  - **メッセージとパスワードを連結しハッシュ値を計算する**
  - **パスワードが異なるとハッシュ値が異なる**

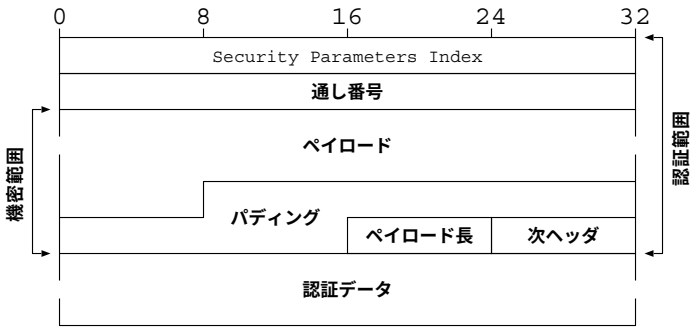
H(Password + Message)

↓

a719abh80

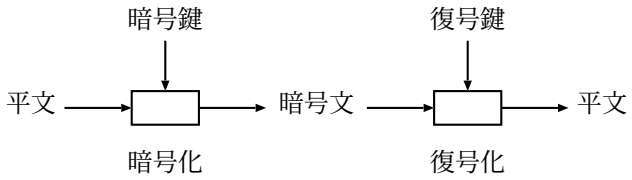
  - **改竄の事実は検出できる**
  - **どこが改竄されたかは分からない**

暗号ペイロード



- SPI **と通し番号は平文**
- **通し番号によるリプレイ攻撃の防止**

暗号

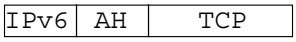


- □ **共有鍵暗号**
  - ○ **暗号鍵と復号鍵が同一**
- □ **公開鍵暗号**
  - ○ **暗号鍵と復号鍵が異なる**

共有鍵暗号　　公開鍵暗号

IPsec のモード

- □ **トランスポート・モード**
  - ○ **始点ホストが** AH **や** ESP **を作る**



- □ **トンネル・モード**
  - ○ **トンネル・ルータが** AH **や** ESP **を作る**
  - ○ VPN (Virtual Private Network)

IPsec アーキテクチャ

- □SPD
  - ○Security Policy Database
  - ○**拡張フィルター**
- □SAD
  - ○Security Association Database
  - ○IPsec **エンジンが参照**
  - ○SA **は暗号や認証方式などの合意**

AH での SPD と SAD の例

- □SPD **出力**
  - ○**アドレス＋ポートで検索**
  - ○AH **を付ける**
- □SAD **出力**
  - ○**アドレス＋ポートで検索**
  - ○AH、HMAC-MD5、**認証鍵**、SPI
- □SAD **入力**
  - ○SPI **で検索**
  - ○AH、HMAC-MD5、**認証鍵**
- □SPD **入力**
  - ○**アドレス＋ポートで検索**
  - ○**認証に失敗したら捨てる**
  - ○AH **が付いていないと捨てる**

IPsec と NAT

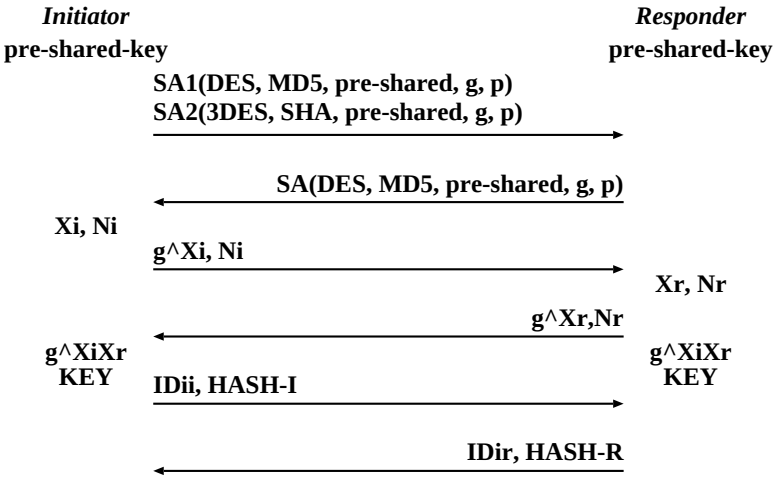
- IPsec と NAT は相性が悪い
  - IP ヘッダを書き換えられる
    - AH は困る
  - サイト内のホストの IP アドレスをポートにマップする
    - ESP ではポートが見えない
  - NAT は状態を持つ
    - リブートすると IP アドレスの対応関係が変わる
- NAT は IPv4 にとっての必要悪
  - IPv6 には NAT は不要

鍵配送

- 手動設定
  - SA のすべてのパラメータを設定するのはめんどう
  - ホストごと、通信ごとに設定するのはめんどう
  - 定期的な更新はめんどう
- IKE
  - Internet Key Exchange
  - SA の自動設定プロトコル
  - IPsec を使用しない
  - UDP を利用 (デーモンとして実装可能)

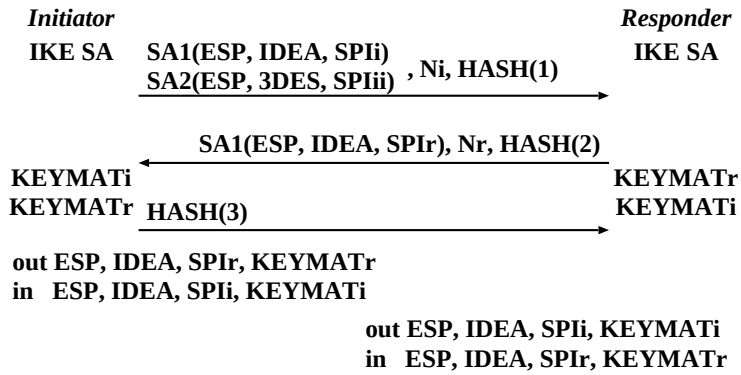
IKE Phase I

□IKE のための SA を確立



IKE Phase II

□IPsec のための SA を確立



IPsec と IKE で利用可能な暗号

- Domain of Interpetation(DOI)
- **ハッシュ関数**
  - MD5、SHA-1 (MUST)
  - DES
- **共有鍵暗号**
  - DES、NULL (MUST)
  - 3DES、CAST、BLOWFISH
  - IDEA、3IDEA、RC4、RC5
- **公開鍵暗号**
  - Diffie-Hellman

IPsec と IKE の実装状況

- **米国では少なくとも** 60 **社が** IPsec **を実装**
- **ルータでの実装が多い**
  - VPN (**トンネル・モード**)
- **国内では数社**
  - **住友電工、東芝、日立**
  - YAMAHA、KAME、**ソリトン**、etc...
- IKE **の相互接続性テストはこれから**

IPv6

次世代 IP の必要性

- 経路表増加の抑制
  - 1組織1クラスBアドレスの割り当て
- クラスBアドレスの枯渇
  - 複数のクラスCアドレスの割り当て
- 経路表の急増
  - CIDR による経路表増加の抑制
- それでもインターネットは成長する
  - IPv4アドレス全体の枯渇
- アドレス空間の大きな IP が必要
  - 次世代 IP or IPng(IP next generation)

IPv6 は NAT のいらない世界

- インターネットの基本は双方向性
  - NAT は IPv4 には必要悪
  - 一方向性はパケット・フィルタリングで実現できる
- 双方向性の必要な環境
  - ホーム・ネットワーク
  - 複数のコネクションを必要とする通信
- グローバル空間におく必要がある端末
  - 自動車
  - 携帯電話

- IPv6 + IPsec + パケット・フィルタリング

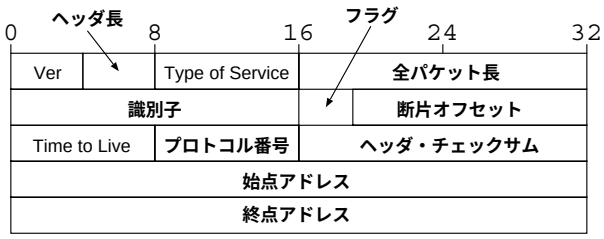
IPv6 の設計思想

- IPv4 よりも大きなアドレス空間を提供
- IPv4 の長所を引き継ぐ
- IPv4 よりも効率をよくする
- 2代目のジレンマを避ける
  - 多機能を追求して仕様を太らせない
- だれでも使えるように
- セキュリティ機能の導入

IPv6 の特徴

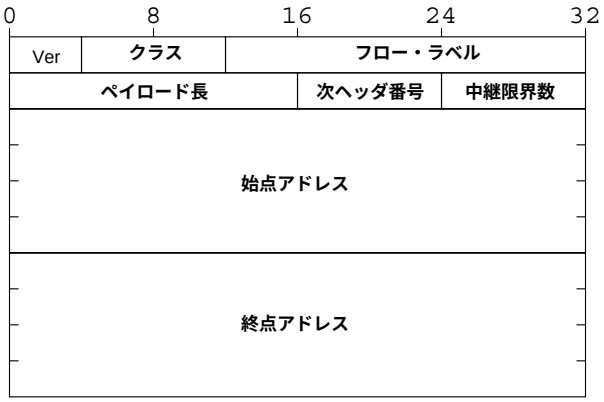
- □アドレスの拡張
  - ○2^32 = 43 億 → 2^128 = 3.4 × 10^38
- □ヘッダの簡略化
  - ○ヘッダ長、TOS、断片オフセットなどの排除
- □数珠つなぎヘッダ
  - ○利用頻度の低い機能を追い出す(断片ヘッダなど)
  - ○汎用的なオプションの定義
- □プラグ＆プレイ
  - ○デフォルト経路、プレフィックスなどの取得
- □IPv4 と変わらない機能
  - ○セキュリティ
    - ▹ただし IPv6 では IPsec が必須
  - ○マルチキャスト、モバイル

IPv4 ヘッダ



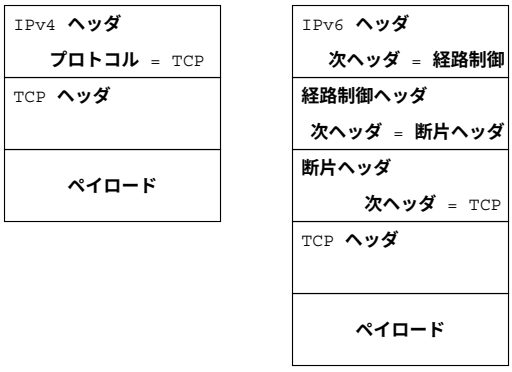
- □除去:
  - ○ヘッダ長、すべてのオプション (ヘッダ長の固定化)
  - ○TOS、ヘッダチェックサム
  - ○識別子、フラグ、断片オフセット (できるだけ断片化しない)
- □名称変更:
  - ○プロトコル番号 → 次ヘッダ番号
  - ○TTL → 中継限界数 (Hop Limit)

IPv6 ヘッダ



- アドレス長は 4 倍、ヘッダ長は 2 倍
- オプションは拡張ヘッダで実現
- 中継限界数は最大 255

拡張ヘッダの数珠つなぎ



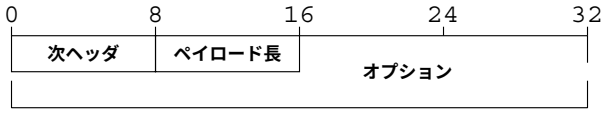
- TCP や UDP を示すプロトコル番号を次ヘッダに抽象化
- 汎用的なオプション
- TCP、UDP、認証ヘッダ、暗号ペイロードの IPv4 との共有

拡張ヘッダとヘッダ番号

0 **中継点オプション・ヘッダ** (Hop-by-Hop Options Header)
1 ICMP
4 IPv4 **ヘッダ**
6 TCP **ヘッダ**
13 UDP **ヘッダ**
41 IPv6 **ヘッダ**
43 **経路制御ヘッダ** (Routing Header)
44 **断片ヘッダ** (Fragment Header)
50 **暗号ペイロード** <IPsec>
51 **認証ヘッダ** <IPsec>
58 ICMPv6
60 **終点オプション・ヘッダ** (Destination Options Header)

ICMP: Internet Control Message Protocol

オプション・ヘッダ



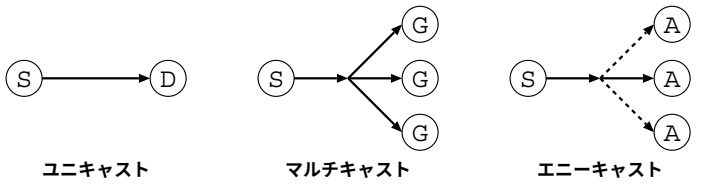
- □ **「オプション番号、長さ、値」 形式**
- □ **8 ビットのオプション番号**
  - ○ICMP **動作ビット**
  - ○change en-route **ビット** (for IPsec)
- □ **中継点オプション・ヘッダ**
  - ○**巨大ペイロード・オプション**
- □ **終点オプション・ヘッダ**

アドレスの表記

- 16 進数 4 桁ごとに ":" で区切る
  - 3ffe:0501:0008:0000:0260:97ff:fe40:efab
  - ff02:0000:0000:0000:0000:0000:0000:0001
- それぞれの先頭の 0 は省略可
  - 3ffe:501:8:0:260:97ff:fe40:efab
  - ff02:0:0:0:0:0:0:1
- 連続する 0 は "::" で表現可
  - 3ffe:501:8::260:97ff:fe40:efab
  - ff02::1
- プレフィックス長は "/" の後に 0 〜 128
  - 3ffe:100::/16

アドレスの種類

- ユニキャスト
  - 特定の 1 ホストと通信
- マルチキャスト
  - ホストのグループと通信
- ブロードキャスト
  - あるリンクに属す全ホストと通信
- エニーキャスト
  - 複数のホストが受け取れるアドレスに送信、受け取るのは 1 ホスト



ユニキャスト　マルチキャスト　エニーキャスト

アドレスのおおまかな分類

3 ビットのプレフィックス (2進数)

000 特殊なアドレス
001 経路集約型アドレス
010 未割り当て (was プロバイダ型アドレス)
011 未割り当て (was 地域型アドレス)
100 未割り当て
101 未割り当て
110 未割り当て
111 リンクローカル、サイトローカル、マルチキャスト

(注) ブロードキャストは無くなった

特殊なアドレス (ユニキャスト)

□ループバック・アドレス
- ○0000:0000:0000:0000:0000:0000:0000:0001 or ::1

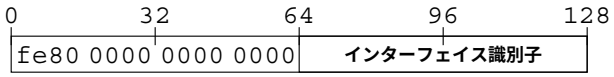
□未指定アドレス
- ○0000:0000:0000:0000:0000:0000:0000:0000 or ::
- ○重複アドレス検知に利用

□IPv4 互換アドレス
- ○0000:0000:0000:0000:0000:0000:xxxx:xxxx
- ○(例) ::163.221.202.11
- ○自動トンネルに利用

□IPv4 射影アドレス
- ○0000:0000:0000:0000:0000:ffff:xxxx:xxxx
- ○(例) ::ffff:163.221.202.11
- ○カーネルの実装に利用

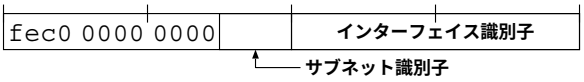
ローカルアドレス

□ **リンクローカル**



○(**例**) fe80::260:97ff:fe40:efab

□ **サイトローカル**



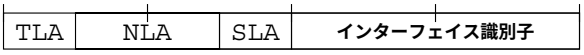
○(**例**) fec0::1234:260:97ff:fe40:efab

グローバル・アドレス

□ **経路集約型アドレス**

○**位置情報と識別子の分離**

○**パブリックとサイトの分離**



□ TLA (Top Level Aggregator)

○8,192 **個**

□ NLA (Next Level Aggregator)

○NLA1, NLA2,...

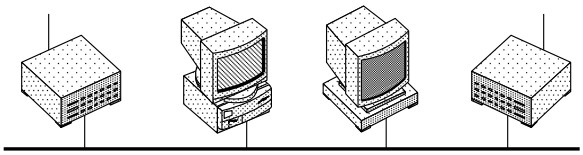
□ SLA (Site Level Aggregator)

○**サイトローカルとサブネット番号を共有**

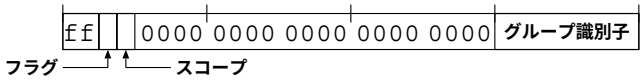
3ffe:501:8:1234:260:97ff:fe40:efab

エニーキャスト

- □サービス探索に利用
- □みかけ上はユニキャストと区別が付かない
- □あまり経験がない
  - ○定義されているのはサブネット・ルータ・エニーキャストのみ
- □だれが受け取るのか？
  - ○サブネット外からは経路制御で決まる
  - ○サブネット内からは近隣探索で決まる

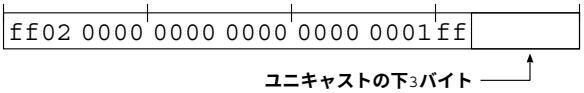


マルチキャスト



- □4ビットのスコープ
  - ○1 ノードローカル・スコープ
  - ○2 リンクローカル・スコープ
  - ○5 サイトローカル・スコープ
  - ○8 組織ローカル・スコープ
  - ○e グローバル・スコープ
- □32 ビットのグループ識別子
  - ○ff01::1 (ノードローカル全ノード)
  - ○ff02::1 (リンクローカル全ノード)
  - ○ff02::2 (リンクローカル全ルータ)

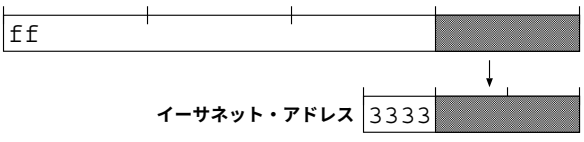
要請マルチキャスト・アドレス



- アドレス解決(いわゆる ARP)に使う
  - ブロードキャストはない
  - リンクローカル・全ノード・アドレスは大きすぎる
  - もう少し小さいマルチキャスト・アドレスが必要
  - (例) fe80::2056:01ff:fe12:3456 → ff02::1:ff12:3456
  - 通常のマルチキャストでは ff00:0000 〜 ffff:ffff を使わない

イーサネット・アドレスと IPv6 アドレス

- リンクローカル
  - fe80::0260:97ff:fe40:efab ← 00:60:97:40:ef:ab
- 経路集約型アドレス (グローバル)
  - 3ffe:501:808::0260:97ff:fe40:efab ← 00:60:97:40:ef:ab
- マルチキャスト



  - ff02::1 → 33:33:00:00:00:01
  - ff02::1:ff40:efab → 33:33:ff:40:ef:ab

DNS

- □正引き
  - ○AAAA レコード
  - ○(例)

  mine.v6.org. IN AAAA 3ffe:501:808:1:200:f8ff:fe01:6317
- □逆引き
  - ○PTR レコード
  - ○(例)
  - ○$ORIGIN 1.0.0.0.8.0.8.0.1.0.5.0.e.f.f.3.IP6.INT.
  - ○7.1.3.6.1.0.e.f.f.f.8.f.0.0.2.0 IN PTR mine.v6.org.
- □BIND 4.9.4 以降でサポート
  - ○要求にはまだ IPv4 を使う
  - ○Newbie では IPv6 でも要求できる

DNS API

- □ホスト名からは IPv4 か IPv6 かを判定できない
  - ○A レコード
  - ○AAAA レコード
- □抽象化された DNS ライブラリ
  - ○getaddrinfo + AF_UNSPEC
    - ▹IPv4 でも IPv6 でも OK
  - ○ユーザは IPv4 か IPv6 かを気にしなくてよい

ICMPv6

1 **終点到達不能** (Destination Unreachable)
2 **パケット過大** (Packet Too Big)
3 **時間超過** (Time Exceeded)
4 **パラメータ問題** (Parameter Problem)
128 **エコー要求** (Echo Request)
129 **エコー返答** (Echo Reply)
130 **マルチキャスト受信者探索**(Multicast Listener Query)
131 **マルチキャスト受信者通知**(Multicast Listener Report)
132 **マルチキャスト受信者探索終了**(Multicast Listener Done)
133 **ルータ要請** (Router Solicitation)
134 **ルータ通知** (Router Advertisement)
135 **近隣ホスト要請** (Neighbor Solicitation)
136 **近隣ホスト通知** (Neighbor Advertisement)
137 **向け直し** (Redirect)
138 **ルータ再設定** (Router Renumbering)

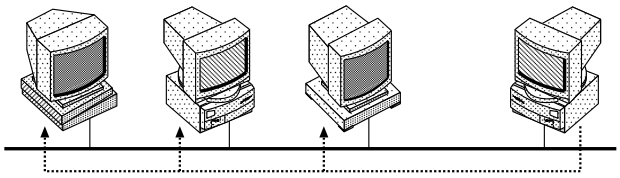
近隣探索 (プラグ＆プレイ)

- **アドレスの自動生成**
- **重複アドレス検知** (Duplicate Address Detection)
- **デフォルト経路の取得**
- **プレフィックスの取得**
- **アドレス解決** (Address Resolution)
- **到達不能検知**
  (Neighbor Unreachability Detection)
- **向け直し** (Redirect)

アドレスの自動生成

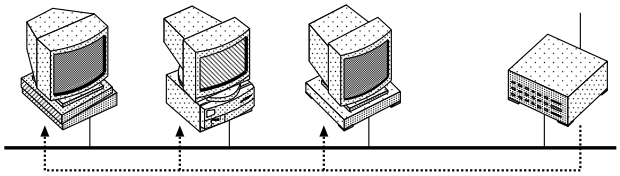
- □イーサネット (IEEE 802 アドレス)
  - ○00:60:97:40:ef:ab
- □インターフェイス識別子 (EUI 64 アドレス) の生成
  - ○260:97ff:fe40:efab
- □リンクローカルを仮に割り当てる
  - ○fe80::260:97ff:fe40:efab
- □マルチキャスト・アドレスへの参加
  - ○リンクローカル・全ノード・マルチキャスト・アドレス
    ff02::1
  - ○要請マルチキャスト・アドレス
    fe80::260:97ff:fe40:efab → ff02::1:ff40:efab

重複アドレス検知



- □近隣要請を出す
  - ○終点アドレスは、要請マルチキャスト・アドレス
  - ○始点アドレスは、未指定アドレス (::)
  - ○対象アドレスは、自分の仮のアドレス
- □対象アドレスが重複した場合
  - ○近隣通知で重複を知らせる
  - ○終点アドレスはリンクローカル全ノード・マルチキャスト

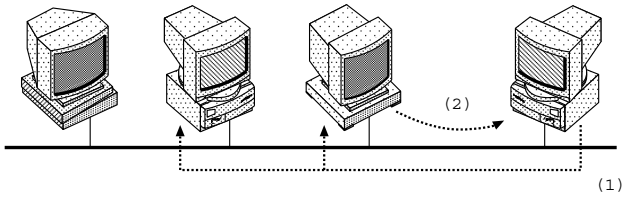
デフォルト経路とプレフィックスの取得



□**ルータ通知**
- ○**定期的なアナウンス** (to ff02::1)
- ○**ルータ要請**(to ff02::2) **への反応**
- ○**デフォルト経路、プレフィックス、**etc...

□**グローバル・アドレスの生成**
- ○**プレフィックス＋インターフェイス識別子**
- ○(**例**)3ffe:0501:0808::260:97ff:fe40:efab

アドレス解決

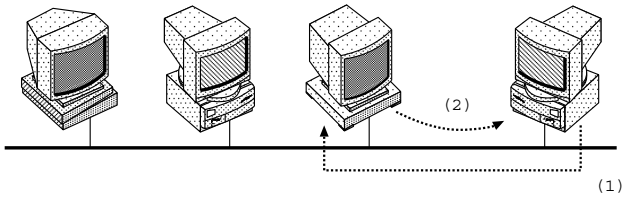


□ARP **の抽象化**
- ○IPv4 **と違いデータリンクごとに** ARP **を定める必要はない**

□**近隣要請**
- ○**近隣要請マルチキャスト・アドレスに近隣要請**

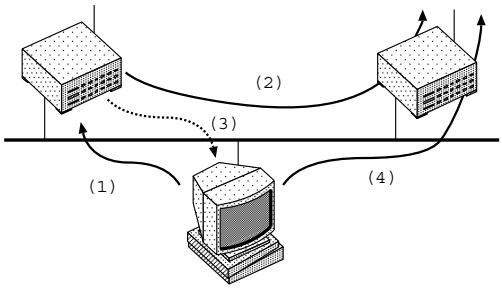
□**近隣通知**
- ○**受信ホストは対象アドレスを検査**
- ○**対象アドレスが一致すると** MAC **アドレスを応答**

到達不能検知



- □近隣キャッシュ(ARP表)は状態を持つ
- □ユニキャストの近隣要請
  - ○時間がたって「あやしく」なったとき

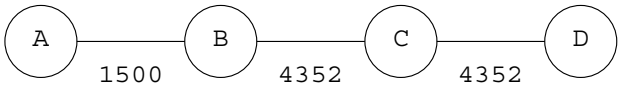
向け直し



- ○ホストは経路制御に参加しない
- ○ホストははじめデフォルト経路のみを持つ
- ○間違っている経路は向け直してもらい、学習する
- ○次からは正しいルータに転送できる

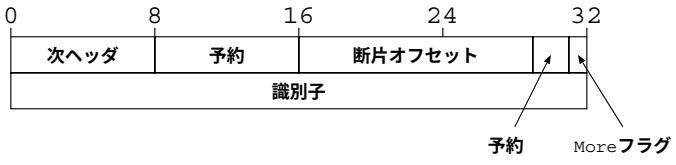
経路 MTU 探索

- □ローカル MTU が最小の場合が多い
  - ○TCP は MSS(Maximum Segment Size)を交換
  - ○1500 が大多数
- □経路上で MTU が小さくなるのはせいぜい 1 回
  - ○経路 MTU は一方向
- □経路 MTU 探索
  - ○ローカル MTU で送信
  - ○ルータからのパケット過大メッセージにより補正
- □IPv6 最小 MTU
  - ○1280 = 1024 + 256
  - ○(was 576)



分割＆再構成

- □分割は回避した方がよい
  - ○TCP の通信単位(セグメント)にあわせる
  - ○UDP や IP では分割は避けられない
- □ルータでは分割しない
  - ○パケットは破棄
  - ○パケット過大メッセージを返す
- □始点ホストのみで分割
- □断片ヘッダ



IPv6 の現状

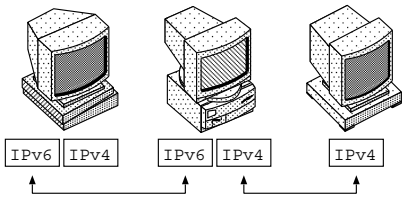
- IETF の IPng 分科会
  - http://playground.sun.com/pub/ipng/html/ipng-main.html
- ルータ
  - Bay Networks、Cisco、3Com
- ホスト & ルータ
  - 日立、東芝、富士通、
  - WIDE プロジェクト、INRIA、
  - NRL、UNH、etc...

IPv6 への移行

移行のストーリー

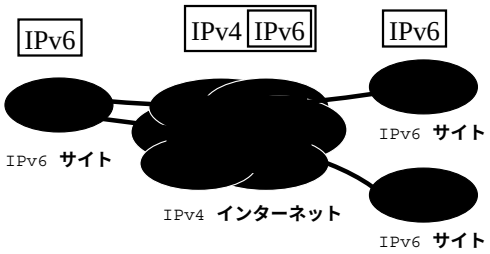
- 初期
  - IPv4 が多数、IPv6 が小数
  - デュアル・スタック
  - IPv6 in IPv4 トンネル
  - トランスレータ
- 後期
  - IPv6 が多数、IPv4 が小数
  - IPv4 in IPv6 トンネル
  - トランスレータ

デュアル・スタック



- デュアル・スタック・ホスト
  - 初期の IPv6 ホストは IPv4 もしゃべる必要がある
  - IPv4 ホストとは IPv4 で
  - デュアル・スタック・ホストとはどちらでも可
  - DNS は IPv4 で検索
- BITS (Bump In The Stack)
  - OS の入れ換えなしにデュアル・スタック機能を持たせる
    - IPv4 ホストのドライバを入れ換える

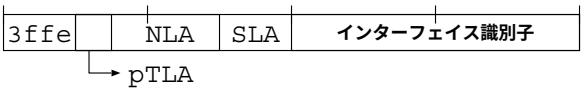
IPv6 in IPv4 トンネル



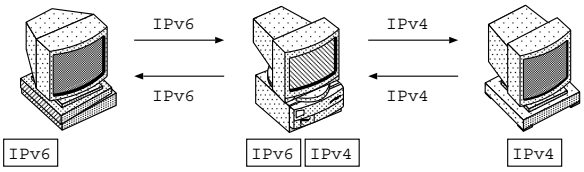
- IPv6 サイトは IPv4 の海に浮かぶ島
- IPv6 島を接続
  - IPv4 インターネットをデータリンクだと思う
  - IPv6 パケットを IPv4 パケットにカプセル化

現在の world-wide 6bone

- ホームページ
  - http://www.6bone.net/
- RIPng から BGP4+ への移行
- 経路集約型アドレスへの移行

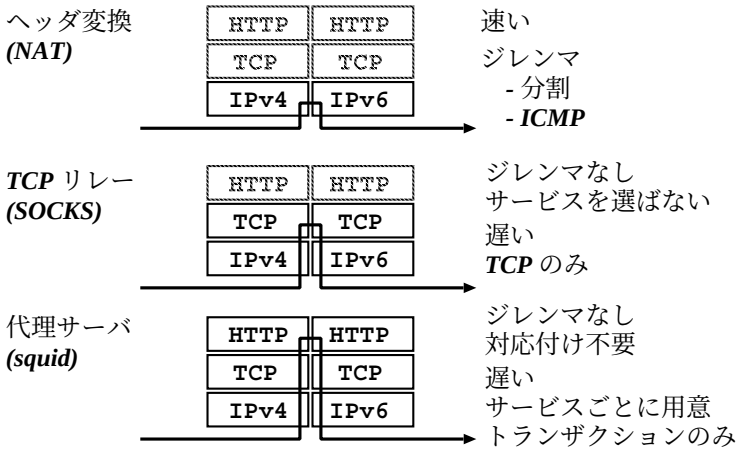


トランスレータ

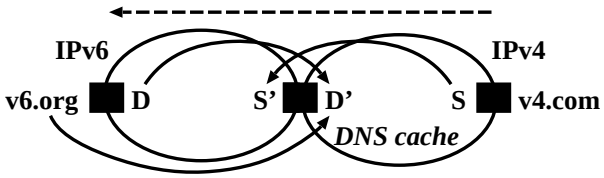


- 移行初期から IPv6 ホストが出現
- 移行後期も IPv4 ホストは残る
  - IPv4 アドレスはいずれ枯渇する
- IPv4 ホストと IPv6 ホストが混在する
  - 通訳が必要
- トランスレータ
  - プロトコル変換
  - アドレスの対応付け

プロトコル変換



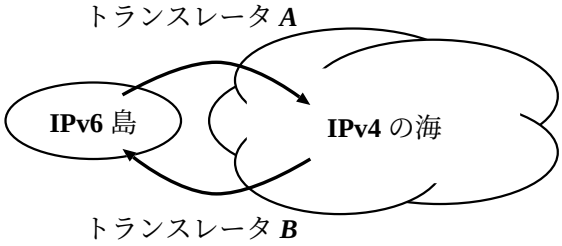
アドレスの対応付け



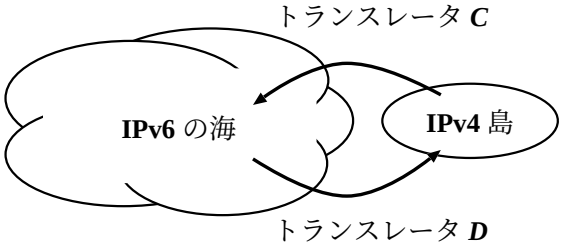
- v4.com が v6.org に通信を始める
- トランスレータは v6.org に IPv4 アドレスを割り当てる
- 終点アドレスの対応付けが本質
  - 始点アドレスはトランスレータのアドレスで代用可
- IPv4 から IPv6 への対応付けは容易 (静的)
- IPv6 から IPv4 への対応付けは困難 (動的)
  - IPv4 アドレス・プールは小さい
  - DNS のキャッシュ問題

トランスレータの分類

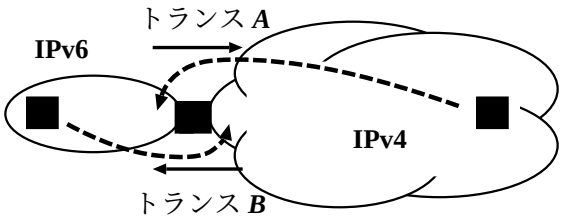
移行初期



移行後期

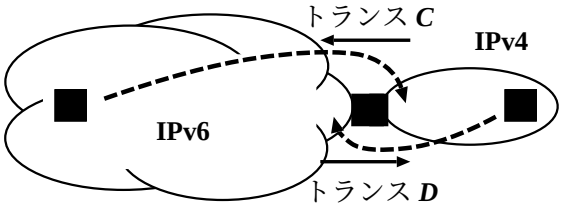


トランスレータ A と B



- □ トランスレータ A
  - ○グローバル IPv4 → グローバル IPv6 (静的)
  - ○実装は容易
- □ トランスレータ B
  - ○グローバル IPv6 → グローバル IPv4 (動的)
  - ○グローバル IPv4 アドレス・プールは小さい
  - ○DNS キャッシュが IPv4 の海に拡散する
  - ○実装不可能に近い

トランスレータ C と D



- □ トランスレータ C
  - ○グローバル IPv6 → プライベート IPv4 (動的)
  - ○プライベート IPv4 アドレスを利用可能
  - ○DNS キャッシュは IPv4 島に閉じる
  - ○実装可能
- □ トランスレータ D
  - ○グローバル IPv4 → グローバル IPv6 (静的)
  - ○実装は容易