ユービキタスネットワーキング時代の
モバイルインターネット技術最新動向

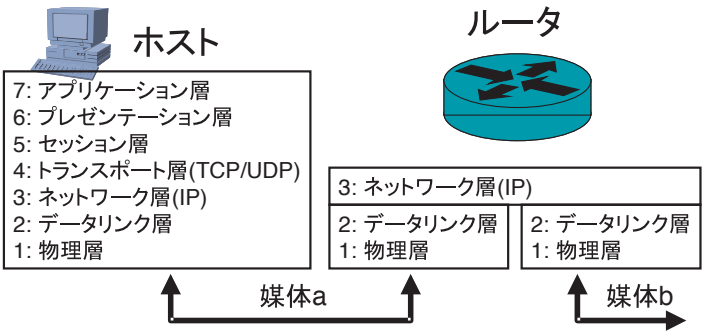
石山 政浩

株式会社東芝 研究開発センター
通信プラットホームラボラトリー
masahiro@isl.rdc.toshiba.co.jp

Mobile IPv4

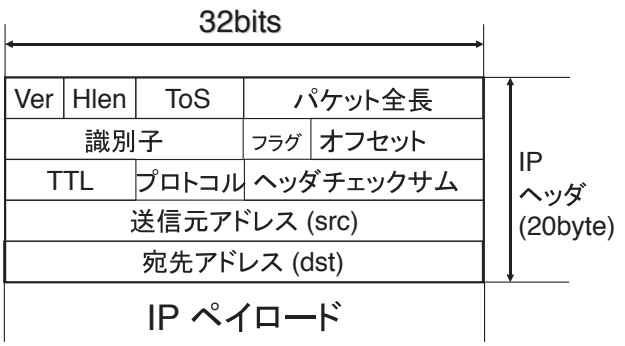
背景:IP によるネットワーク

- OSI 7 階層モデル

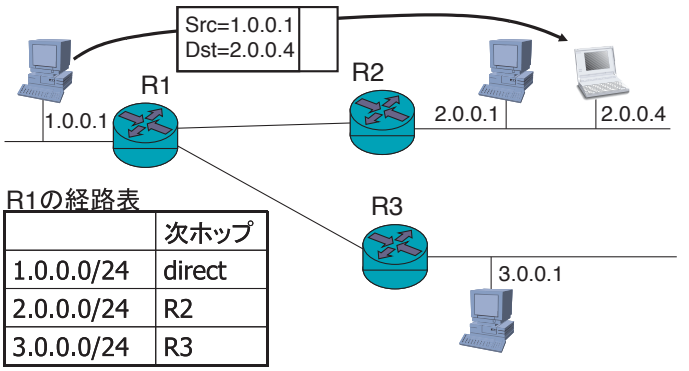


IP パケットの中身

- すべてのデータは「パケット」として運ばれる



IP における経路制御

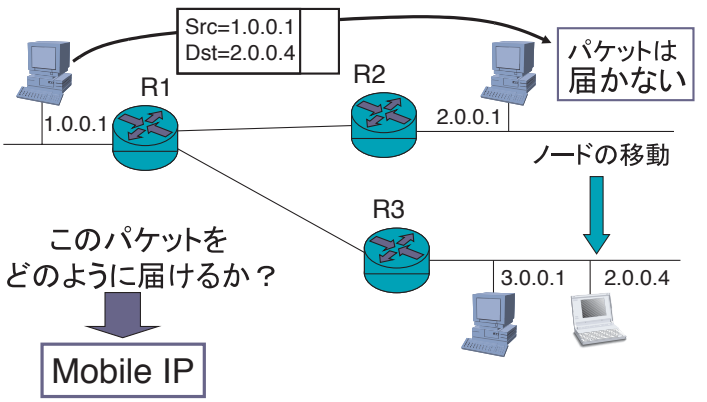


R1の経路表

	次ホップ
1.0.0.0/24	direct
2.0.0.0/24	R2
3.0.0.0/24	R3

/24 =ネットワークプレフィックス長

もしノードが移動したら



アドレスの二重性

- 現在 IP アドレスは二つの意味で使用されている
  - ノードのサブネットへの接続位置 (位置指示子)
  - ノード自体の識別子 (ノード識別子)
- 移動により IP アドレスが変化
- 位置指示子だけでなくノード識別子も変化

ノードの identity の消失

Mobile IP の必要性

- 可搬性：ノードがネットワーク上を移動可能
- DHCP, PPP などを使う

IP アドレスの変更が起きる

- ノードの発見が困難
  - 移動ノードに対して発呼できない (e.g. 電話)
- 移動したらセッションを維持できない
- IP アドレスに依存した制御が不可能

移動透過性

- 移動によるセッション喪失を回避
  - ネットワーク上を移動してもあらゆるセッションを保持できる (e.g. ftp, realaudio, NFS…)
- 移動ノードに対する発呼が可能
  - 移動ノードの現在位置を意識することなく，ある一定の識別子を指定するだけで移動ノードと通信できる
- Mobile IP はこの「移動透過性」をノードが移動しても同じ IP アドレスを使いつづけらるようにして提供する
- 応用例: インターネット携帯電話

Mobile IP

- RFC3344 (obsoletes RFC3220, RFC2002) で規定
- IETF Internet Area:
  - Mobility for IPv4 (mip4) WG で議論
- Chairs:
  - Peter McCann (Lucent)
  - Henrik Levkowetz (ipUnplugged)

IETF, RFC, I-D

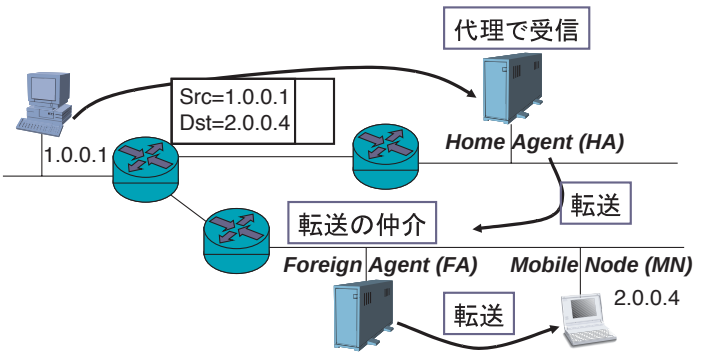
- IETF: Internet Engineering Task Force
  - インターネットで利用されるプロトコルを議論する場
  - 標準化団体 (c.f. International Telcommunication Union)
  - 年に 3 回会合を行なう
- RFC: Request for Comments
  - (簡単に言うと) プロトコル仕様書
- I-D: Internet-Draft
  - RFC にするための議論のたたき台となる文書
- http://www.ietf.org/
- RFC3160 に IETF についての記述があります

Mobile IP とは何か?

- Internet 上でノードの移動透過性を，IP 層で保証するプロトコル
- Macro Mobility: 比較的粒度の荒い移動を提供
  - 対して micro mobility: 高速な移動 (Hand-off) を保証するためのプロトコルの議論もある
- サブネット間の移動をサポート
  - 移動しても IP アドレスは変わらない
  - アプリケーションは移動を気にする必要なく継続可能
  - 移動しても通信 (セッション) は継続可能
  - 通信相手からは「移動していない」ように見える

Mobile IP: 動作概要

- 基本は「転送」



Mobile IP: 用語 (1)

- ホームアドレス (Home Address, HoA)
  - 移動ノードが使いつづけるアドレス
- 気付アドレス (Care-of-Address, CoA)
  - 移動ノードが訪問したサブネットで使用するアドレス
- バインディング
  - HoA と CoA の関係
  - ある HoA をもつ移動ノードが今どこにいるか? (すなわちどの CoA を使用しているか?)
- ホームネットワーク
  - HoA が属するサブネットワーク
  - 移動ノードの本拠地

Mobile IP: 用語 (2)

- Mobile Node (MN)
  - Mobile IP を使用して移動する計算機
- Home Agent (HA)
  - ホームネットワークにいるノード
  - MN がホームネットワークから離れているときに，パケットを代理受信して転送する
- Foreign Agent (FA)
  - 訪問先のネットワークにいるノード
  - MN が移動してきたときに，パケットの転送を支援する

Mobile IP: 動作シナリオ

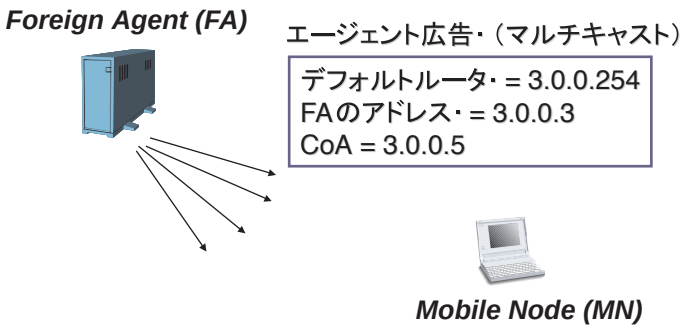
- エージェントの発見
- CoA の決定
- HA への登録
- パケットの送信
- パケットの受信
  - HA による代理受信
  - FA へのトンネルを使った転送
  - FA から移動ノードへの配送
- 移動の検出

エージェントの発見 (1)

- 移動ノードは，移動したらエージェントを探索する
- エージェント広告 (Agent Advertisement, AA)
  - 定期的にサブネットにマルチキャスト
- エージェント要請 (Agent Solicitation, AS)
  - エージェント広告を要求するマルチキャスト

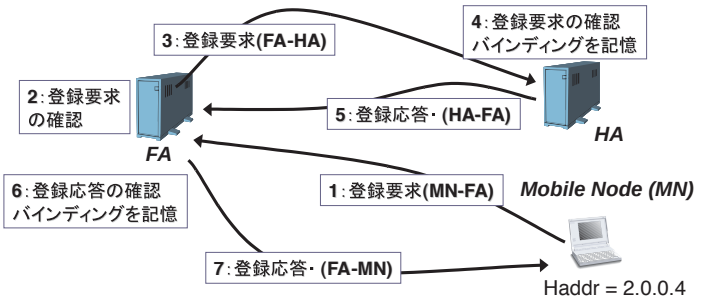
エージェントの発見 (2)

- エージェントからさまざまな情報を取得できる



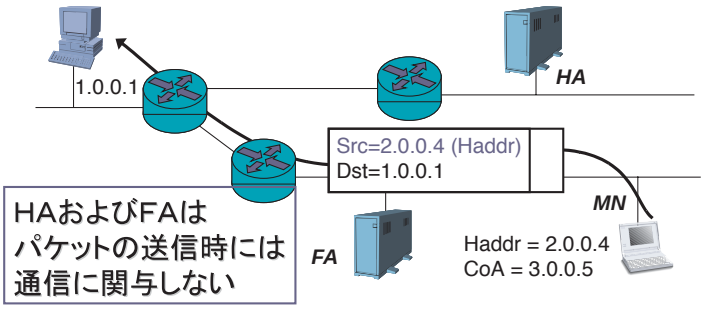
HA への登録

- FA を経由して，HA に現在位置を登録する
- UDP (Port 434) を利用



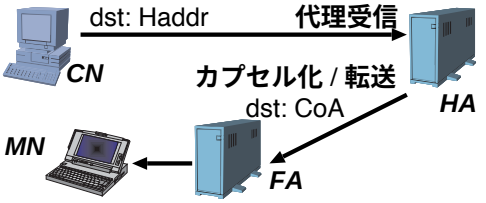
移動ノードからのパケットの送信

- 通常とまったく同じに送信してよい



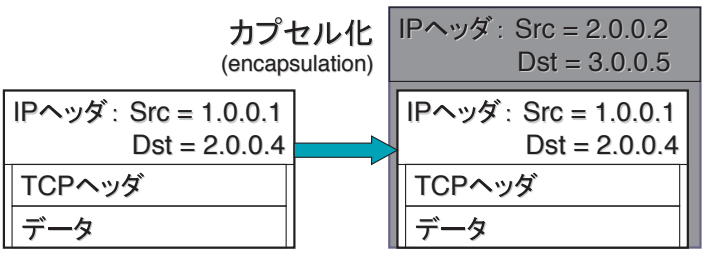
移動ノードのパケットの受信: HA からの転送

- パケットは HoA 宛て:ホームネットワークに届く
- HA は HoA 宛てのパケットを代理受信バインディングを確認して,「カプセル化」を行い，現在の CoA に向けてパケットを転送
- FA は,「カプセル開放」を行って，移動ノードにパケットを転送



IP-in-IP カプセル化

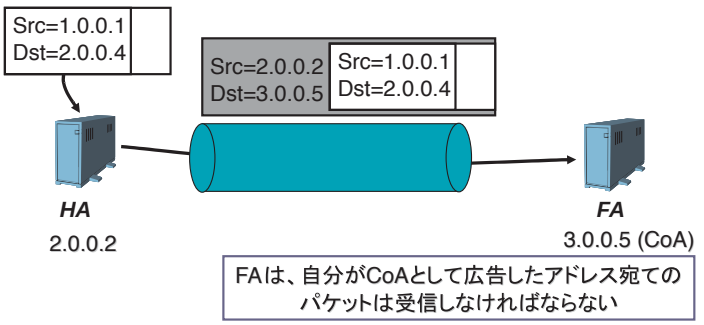
- IP パケットに，さらに IP ヘッダを加える



- 他のメカニズムも使用可能
  - Minimal Encapsulation
  - Generic Routing Encapsulation (GRE)

FA へのトンネルを使った配送

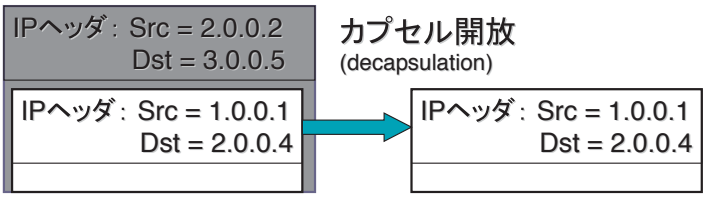
■ トンネル： カプセル化されたパケットの通る経路



FAは、自分がCoAとして広告したアドレス宛ての
パケットは受信しなければならない

FA から移動ノードへの配送

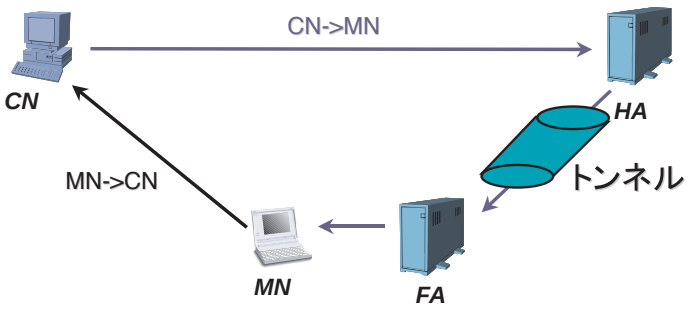
■ トンネルからきたパケットをカプセル開放



FAは、カプセル開放して得られた
パケットをMNへ転送する

三角経路

- 最終的に，Mobile IP を使った通信は HA, FA を経由した三角形の経路を通る



FA がない場合はどうなるのか？

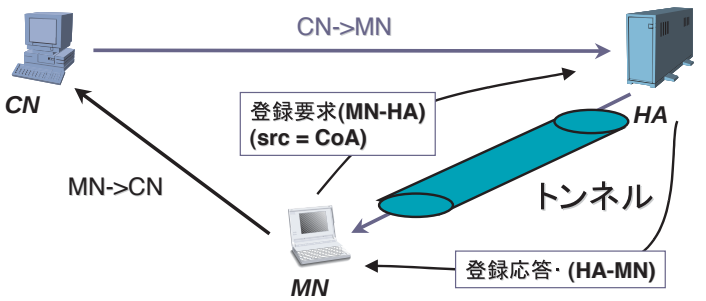
- 移動ノードは，自分で CoA を取得してもよい (たとえば DHCP)

共存気付アドレス (co-located care-of-address)

- 利点
  - 訪問先のネットに FA がいなくてもよい
- 欠点
  - 移動の検出が難しい
  - 訪問先ネットのアドレスを消費する

共存気付アドレスの場合の通信

- **登録：** HA **へ直接送信する**
- **通信：** MN **が直接カプセル開放を行う**



Mobile IPv4 の問題点とその対策

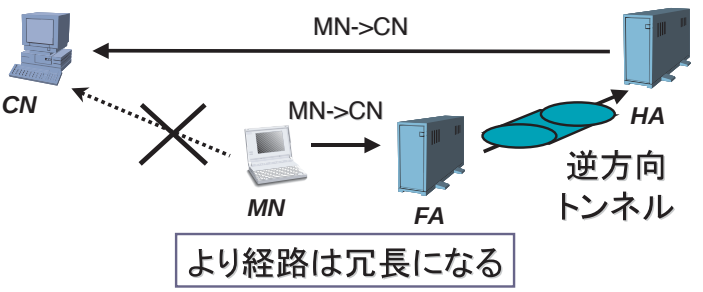
- Mobile IPv4 **の問題点**
  - **送信元アドレス詐称攻撃** (source address spoofing attack) **との誤認**
  - **三角経路による遅延増加および通信効率の低下**
  - NAT **との共存**
  - **高速な移動** (Fast Handoff)
  - **セキュリティとスケーラビリティ**

送信元アドレス詐称攻撃 (source address spoofing attack)

- 悪意を持つノードが，自分の送信元アドレスを偽ってパケットを送信する
  - IP アドレスに基づいたアクセス制御の無効化
  - 攻撃の逆探査を逃れる
- 入口フィルタリング (ingress filtering)
  - ネットワークの構造的に正しくない送信元アドレスを通過させない
  - 主にサービスプロバイダ，大学，企業などの出口ルータで行われる
- 送信元アドレスがホームアドレス
  - 入口フィルタリングを通過できない可能性

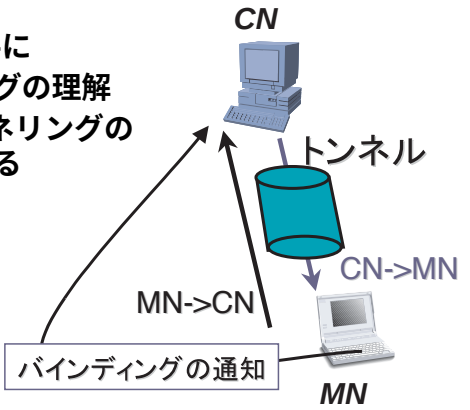
逆方向トンネリング

- MN が送信するパケットも，トンネルを使って一度 HA に戻して，そこから発信したように見せる
- RFC3024



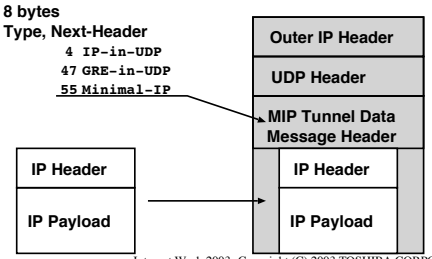
経路最適化

- 自分の通信相手にバインディングを記憶してもらう
- しかし，通信相手に
  - バインディングの理解
  - IP-in-IP トンネリングの実現を強制する



NAT との共存

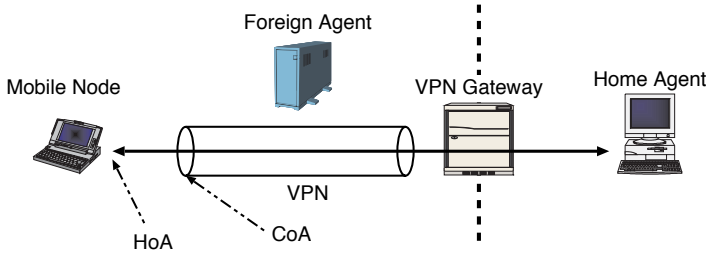
- 規定の tunneling 方式では，NAT の裏側に移動した場合に利用できない
- UDP (Port 434) を利用して Tunneling を行なう
  - 逆方向トンネリングは必須
- RFC3519



VPN との共存

- VPN を利用しながら Mobile IP を行なうための問題

draft-ietf-mip4-vpn-problem-statement



Fast Handoff

- Mobile IP を使って非常に煩雑に移動することを考えると，いくつかの課題が見える
- HA が遠い場合，登録処理の時間が長く，パケットロスを多く発生する
- コントロールパケットが多くなり，ネットワークおよび HA に大きな負荷を与える

- より高速な移動処理への要求
- 最近の Mobile IP WG の主要トピックのひとつ
- 多くの提案が draft として提出されていた

Fast Handoff (cont'd)

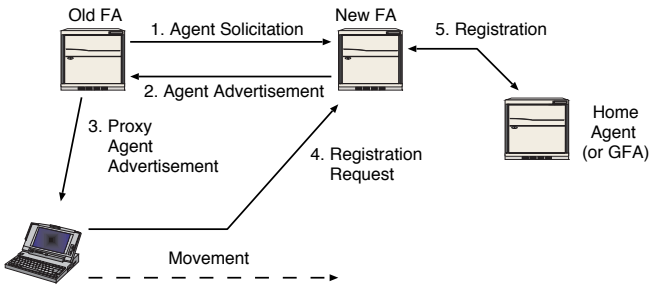
- Fast Handoff **には大きく二種類のアプローチ**
- IP **的に移動した後の支援**
  - **登録までの時間の短縮**
  - e.g. Regional Registration
  - **移動のサポート方法**
- IP **的な移動を隠蔽** (Micro Mobility)
  - **網を囲い込み，ホストルートで配送など**
  - e.g. CellularIP

Fast Handoff

- draft-ietf-mobileip-lowlatency-handoffs
- **いくつかの方法が示されている**:
  - Pre-Registration Handoff
    - Network-Initiated Handoff
      - Source Trigger, Target Trigger
    - Mobile-Initiated Handoff
  - Post-Registration Handoff
    - Two Party Handoff
    - Three Party Handoff

Fast Handoff (例)

- Pre-Registration Handoff
- L3 Handoff が完了した後 L2 Handoff

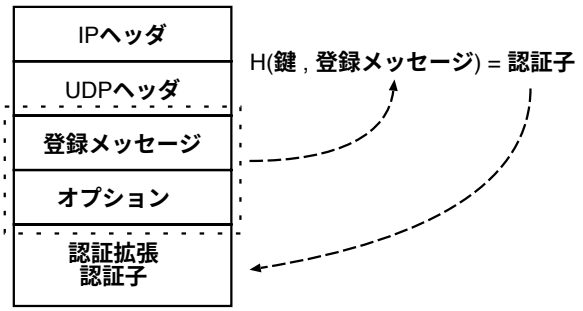


セキュリティ：認証とは

- 自分が「本人」であることを証明するための手続き
- 認証がなければ，簡単に悪意を持つ人間があなたの移動ノードに「なりすます」ことができる
- Mobile IP では，ホームエージェントと移動ノードの間の認証は必須である
- FA と MN，FA と HA との間も認証してもよい

認証メカニズム

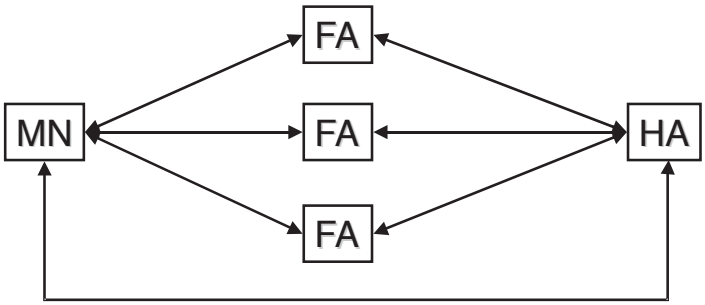
- MN と HA(そして FA) 間で鍵を共有
- 認証子をハッシュ関数を使用して計算



- 認証子は鍵を知らないと正しく生成できない

より安全な登録には

- すべてのエージェント間と移動ノード間で鍵を共有する必要がある



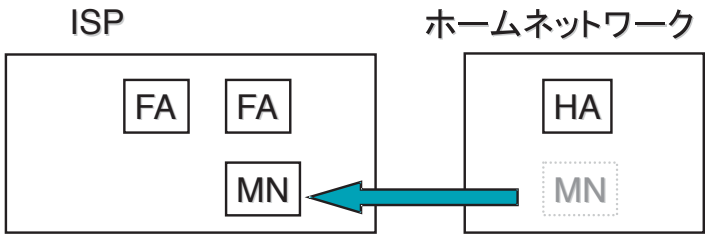
サービスプロバイダの懸念

- エージェント間で鍵を保持しあうため，スケール性に疑問
- 課金メカニズムがない

AAA: Authentication (認証)
Authorization（許可）
Accounting（課金）
の導入

例：ドメインを超えた Mobile IP

- ISP: はたしてこの MN の課金先はホームネットワークで本当にいいのか?
- ホームネットワーク：はたして本当に自分の管理下の MN が ISP に接続されたのか?
- すべてのエージェント間で鍵を共有するのは困難



Mobile IP における AAA

- DIAMETER Mobile IP 拡張
- AAA との連携
  - 異なるドメイン (たとえば ISP) 間での認証と許可 (ローミング)
  - 大規模 ISP でもスケールする鍵配布
- DIAMETER 認証メカニズム
  - RADIUS の拡張
- 課金する相手の明確化
  - Network Access Identifier (NAI) の導入

NAI: Network Access Identifier

- 従来の Mobile IP はホームアドレスに依存した認証
  - 個人の特定が困難
  - 動的な Home Address 割り当てが難しい
- Network Access Identifier (NAI)
  - メールアドレスに似た形式
  - e.g. foo@bar.com
- RFC 2794

Summary: MobileIPv4

- ノードの移動透過性を IP 層で保証するプロトコル
  - Home Address と呼ばれるアドレスを使い続ける
  - 通信相手からは「移動していない」ように見える
- Home Agent(HA) を利用した転送方式
  - 3 角経路
- Macro Mobility
  - 粒度の荒い移動
  - 電話のようなアプリケーションにはサポートメカニズムが必要
    - Fast Handoff

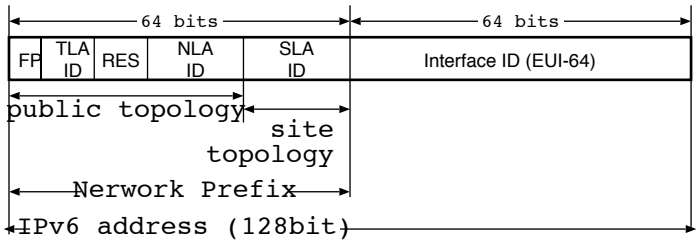
Mobile IPv6

IPv4 と IPv6 の比較

- アドレス空間の拡大
  - IPv4：32bit, IPv6 128bit
  - ほぼ 50 億倍の 50 億倍の 50 億倍
- 各ノードによる自律的なアドレス設定
- アドレスの構造化
- QoS，セキュリティ，マルチキャストを考慮した設計

アドレスの構造化

- 代表的な形式：集約型アドレス (Aggregatable Global Unicast Address)
- 上位 64bit がネットワークプレフィクス
- 下位 64bit がインターフェイス識別子
  - 主に MAC アドレス等から導出される (EUI-64)



アドレスの読み方

- **十六進数で** 16bit **ごとに「：」で区切る**

3ffe:0501:100f:1048:0260:97ff:fe47:9bd2
ff02:0000:0000:0000:0000:0000:0000:0001

- **先頭の「**0**」は省略可能**

3ffe:501:100f:1048:260:97ff:fe47:9bd2
ff02:0:0:0:0:0:0:1

- **連続する「**0**」は一度だけ「**::**」で省略可能**

ff02::1

IPv6 ヘッダ

拡張ヘッダ (Extension Header)

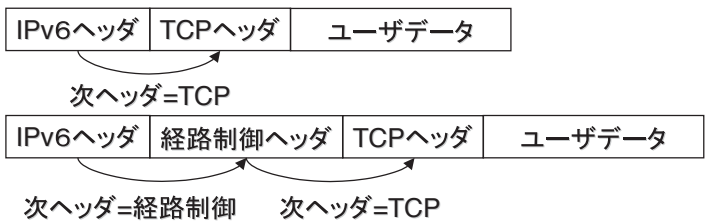
- IPv6 では，必須以外の情報はすべて拡張ヘッダで取り扱われるようになった
- 次ヘッダフィールドに記入される
  - TCP/UDP/ICMP もここに記入される

値	ヘッダーの内容
0	Hop-by-Hop option
6	TCP
17	UDP
43	経路制御ヘッダ (Routing Header)
50	IPsec ESP
51	IPsec AH
60	終点オプションヘッダ (Destination Option)

（抜粋）

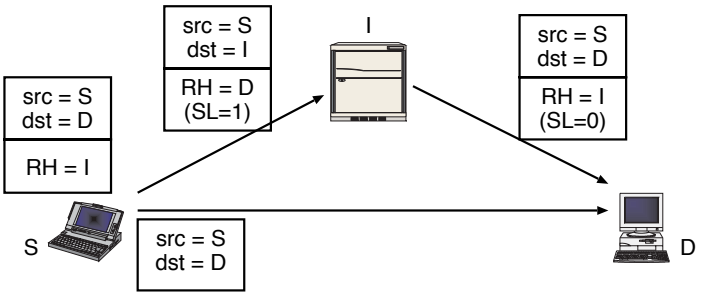
拡張ヘッダの付与方法

- 「次がどのヘッダか」が記入される



Routing Header (経路制御ヘッダ)

- 発信者が『経由点』を指定する



IP security (IPsec)

- IP 層でのセキュリティ保証
- Authentication Header (AH)
  - 完全性，認証の提供
  - IP ヘッダを含む
- Encapsulating Security Payload (ESP)
  - 秘匿性，完全性，認証の提供
  - IP ヘッダを含まない
  - ESP ヘッダ以降は暗号化されている

Mobile IPv6

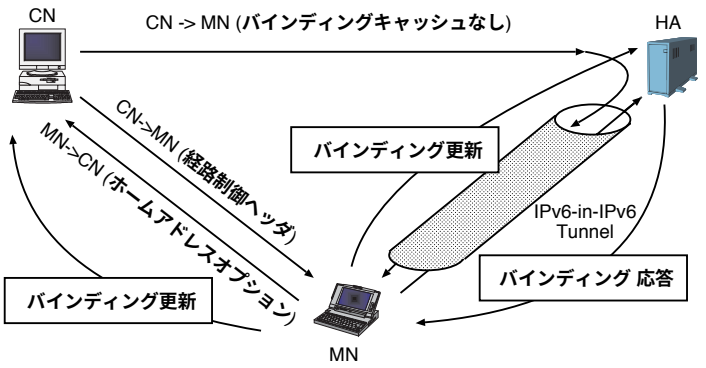
- 現段階ではまだドラフト
  - draft-ietf-mobileip-ipv6
- Mobility for IPv6 (mip6) WG で議論
  - Chairs:
    - Basavaraj Patil (Nokia)
    - Goral Dommety (Cisco)

Mobile IPv4 と Mobile IPv6 の主な違い

- 経路最適化が最初から考慮されている
  - IPv6 Extention Header の利用
- 発信元アドレスが CoA になった
  - Ingress Filtering 対応
  - ホームアドレスオプションの導入
- Foreign Agent の廃止

Mobile IPv6 動作概要

- 大枠は Mobile IPv4 と類似



IPv6 終点オプションヘッダ

- 受信者 (Destination) だけが処理すべきオプションの集まり
- Mobile IPv6 では，新たに 1 つのオプションを定義した
  - ホームアドレスオプション

ホームアドレスオプション

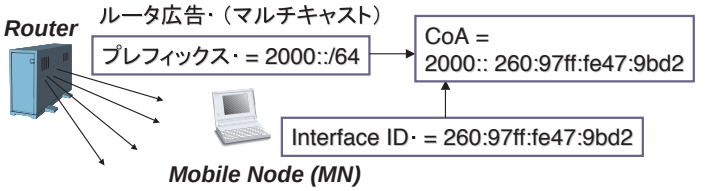
- Mobile IPv6 では送信元アドレスに CoA を使う
- 通信相手に自分のホームアドレスを伝えるためのオプション
- 受信者は，このオプションがついていた場合，送信元アドレスに書いてあるアドレスではなく，このオプションに記入されているホームアドレスから送信されたものとして扱わなければならない

IPv6 Mobility Header

- Mobile IPv6 が定義した「新しい」Extention Header
- Mobile Node の情報交換は主にこのヘッダを利用して行なわれる
  - Binding Refresh Request (BRR)
  - Home Test Init (HoTI)
  - Care-of Test Init (HoTI)
  - Home Test (HoT)
  - Care-of Test (CoT)
  - Binding Update (BU)
  - Binding Acknowledgement (BA)
  - Binding Error (BE)
  - + Mobility Option

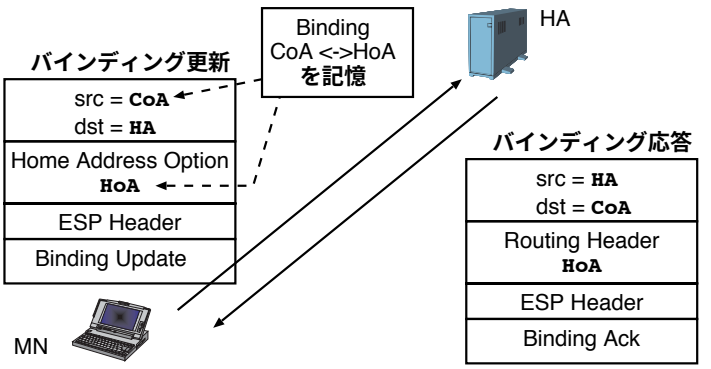
CoA の取得

- Stateless Address Autoconfiguration
- **ルータは定期的にネットワークプレフィックスをサブネットにマルチキャストする (ルータ広告)**
- **自分のインターフェイス ID を連結するとアドレスになる**
  - **DHCP 等を使用してもよい**



HA にバインディングを通知

- Binding Update message **を使用**
- **典型的には** IPv6 **ヘッダのみのパケットとなる**

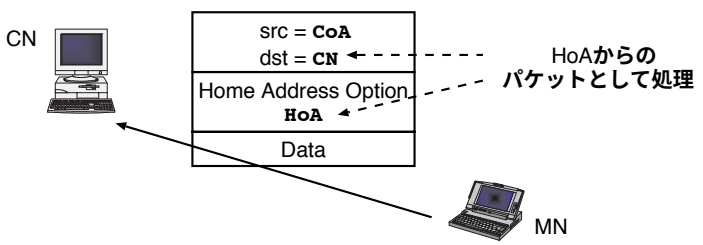


バインディングキャッシュ

- バインディングを覚えておく領域
- 通常の通信相手もバインディングキャッシュを持ってもよい
  - まったく持たなくてもよい
- キャッシュなので任意の時点で廃棄してもよい
  - たとえば多くの移動ノードと通信をしていて領域が足りなくなった場合など
- ただし HA は有効期限までは廃棄してはいけない
  - MN は有効期限が切れる前に更新する

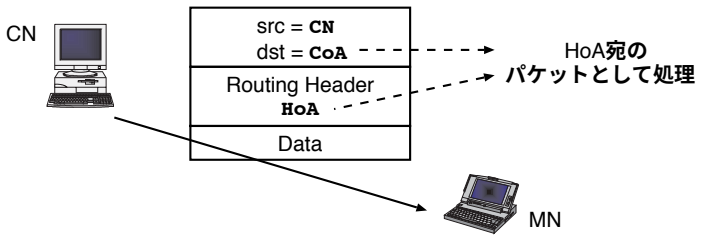
通信相手へのパケットの送信

- 送信元アドレスは CoA
- ホームアドレスオプションを付与
- 受信者はホームアドレスからきたパケットであると解釈



通信相手からのパケット (バインディングキャッシュがある場合)

- 経路制御ヘッダを付与して HA を介さない
- (IPv6-in-IPv6 ではない)



セキュリティ

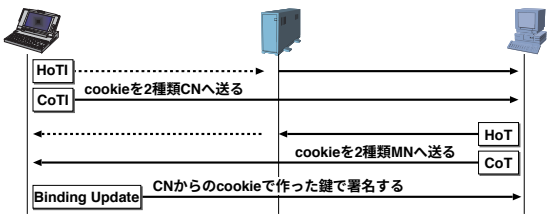
- 「なりすまし」がより容易になった
  - バインディングの通知が通常の通信相手にも可能
  - セキュリティ機能（特に認証）は必須
- IP Security (IPsec) の利用を前提

しかし...

- 2001/03/19，IESG から一通のメールが mobile-ip の ML に届く： IESG Security Concerns with MIPv6
- "The IESG strongly recommends that the WG find an alternate approach that is not tied to IPsec/AH"
  - SA 確立までの overhead の問題
  - IPsec 処理ルールの複雑さ
- HA-MN は信頼関係の前提をおくことが可能
  - 鍵の手設定もありえる: IPsec を前提とできる
- MN-CN は不特定多数との通信
  - IPsec 以外のセキュリティメカニズムを考える必要

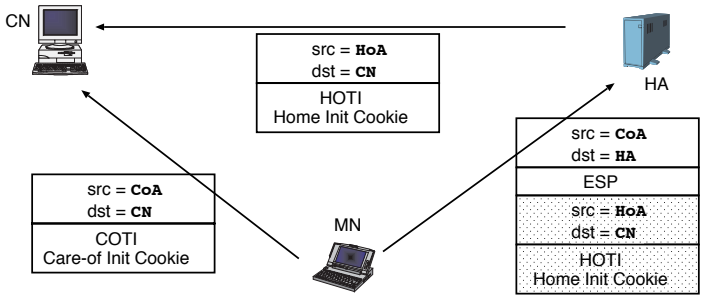
Return Routability(RR)

- MN-CN で利用
- 簡易な認証
  - HA-CN, MN-CN の二点を盗聴すれば破れる
- しかし PKI のような前提が必要ないため導入は容易
- インターネット上の無制限なりすましからの防衛



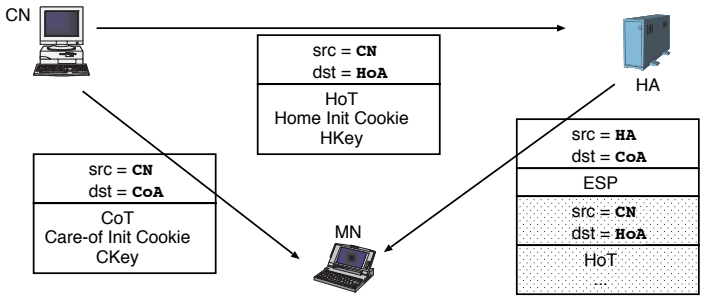
Return Routability(RR) (1/3)

- CN に RR の開始を要求する



Return Routability(RR) (2/3)

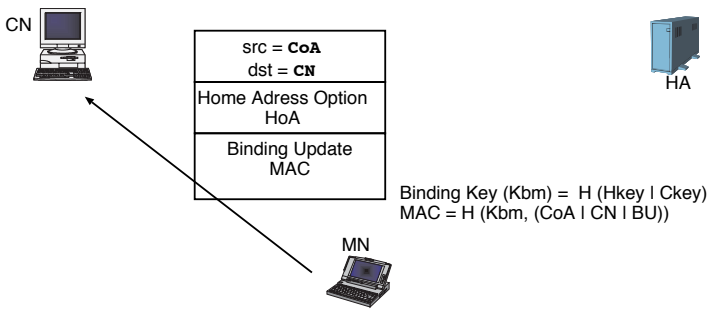
- CN から BU に署名するための鍵の元を受け取る



home keygen token (HKey) = H(Kcn, (home address | nonce | 0))
care-of keygen token (CKey) = H (Kcn, (care-of address | nonce | 1))
Kcn: CNが持つ秘密鍵

Return Routability(RR) (3/3)

- 鍵 (Kbm) を元にハッシュによる署名をつけて BU を CN に送信する

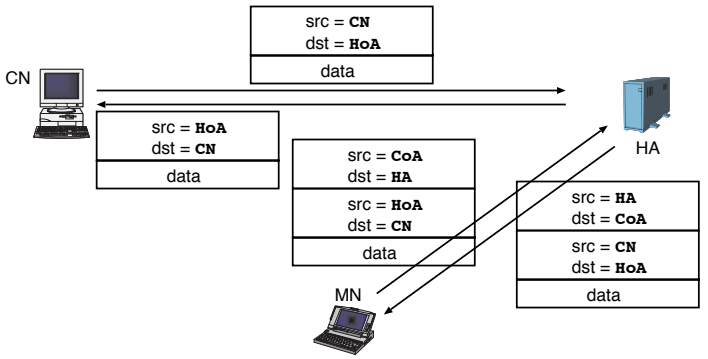


すべてのノードが Mobile IPv6 をサポートしなければならないか?

- 現在のコンセンサスは
  - All nodes MUST (no requirements)
  - All nodes SHOULD be able to participate in Route Optimization
  - All nodes MAY be a mobile node
  - All routers MAY be a home agent
- なので Home Address Option などを理解しなくてもよい
- Binding Cache を持たないもの同じ処理

Binding Cache を持たない場合

- すべて HA を経由した通信となる
  - HA の tunnel には IPsec を適用しても良い



Mobile IPv6 の課題

- 経路の overhead は無くなっていない
  - 相手はバインディングキャッシュを持ってくれないかもしれない
  - Mobile Node への発呼時には必ず HA を経由する
- パケットのオーバヘッドが大きい
  - 移動ノード同士が通信すると，ホームアドレスオプションと経路制御ヘッダで最低でも 44byte のオプションが必要

Mobile IPv6 の課題 (cont'd)

- IPv4 では，FA を使ったさまざまな拡張が行われている
  - AAA を使ったローミング
  - 高速なハンドオフ
- これらの拡張は Mobile IPv6 では再度議論する必要がある

Summary: Mobile IPv6

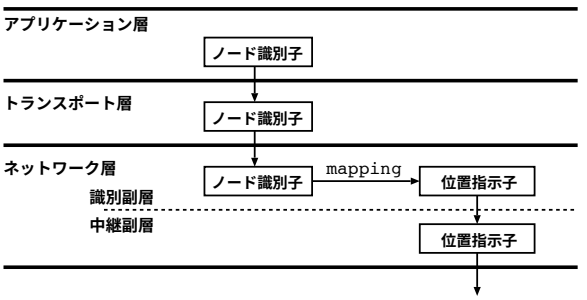
- まだ議論中なので仕様は固まっていない
- 基本的には Mobile IPv4 の方式を踏襲
  - FA はない
  - 経路最適化が盛り込まれている
    - Binding Cache
- IPv6 Extension Header を利用した signaling
  - HA への登録
- セキュリティへの課題
  - MN-HN: IPsec
  - MN-CN: Return Routability(RR)

LIN6: Location Independent Networking for IPv6

LINA (Location Independent Network Architecture) の基本概念

- 位置指示子とノード識別子の概念を分離
  - ネットワーク層を二つの副層に分割
- 縮退アドレスモデルの導入
  - ヘッダオーバーヘッドの回避
- End-to-End の通信
  - Mapping Agent の導入

位置指示子とノード識別子の概念を分離

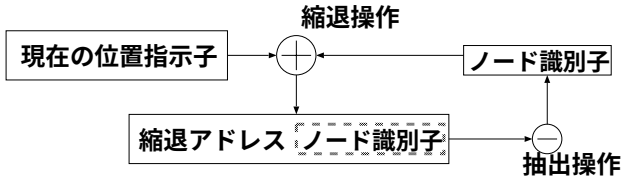


- 識別副層
  - ノード識別子を利用し，現在の位置指示子を得る
- 中継副層
  - 位置指示子を用い，パケットを中継する

縮退アドレスモデル

位置指示子の中にノード識別子を埋め込む

- 縮退アドレス



概念的に２層に分離されたヘッダを１つのヘッダに統合することが可能

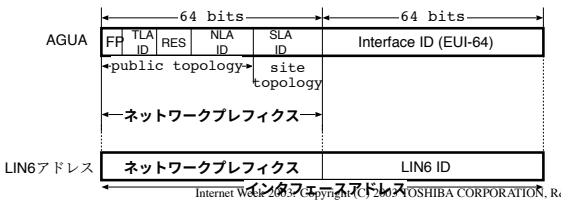
ノード識別子と位置指示子の対応づけ

- **対応関係** = Mapping
- Mapping Agent(MA) **を導入**
  - MA **の動作**
    - **位置指示子とノード識別子の情報を保持**
    - **ノード識別子に対する位置指示子を通知**
  - **移動ノードの動作**
    - **通信したいノードの位置指示子を要求**
    - **移動の際に現在の位置指示子を登録**

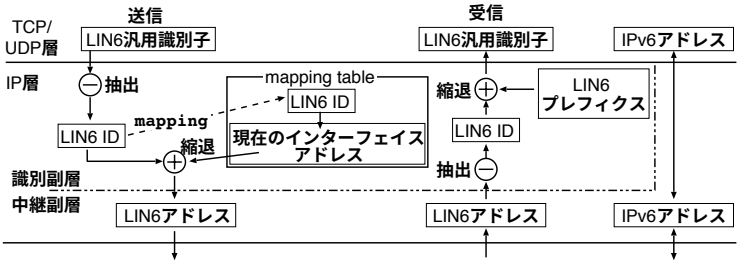
LIN6: LINA を IPv6 に適用

アドレス構造: **縮退アドレスの実現**

- IPv6 **アドレス**
  - **上位** 64bit: **ネットワークプレフィクス**
  - **下位** 64bit: **インタフェース識別子**
- LIN6 **アドレス**
  - 128bit **全体**: **位置指示子**
  - **位置指示子の中の** 64bit: **ノード識別子** (LIN6 ID)

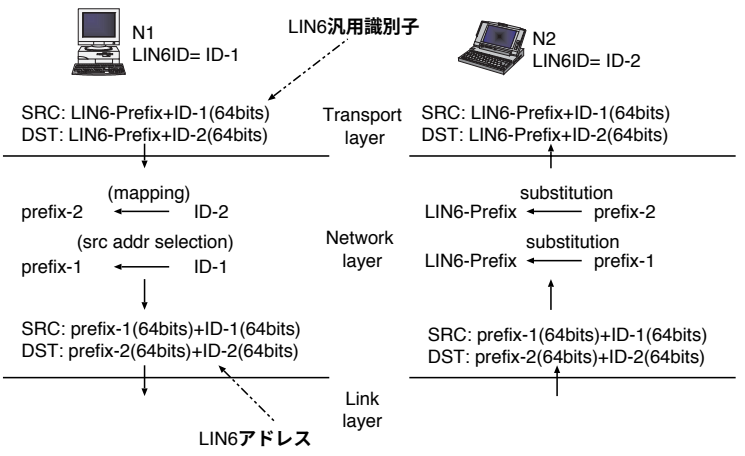


LIN6 の通信モデル

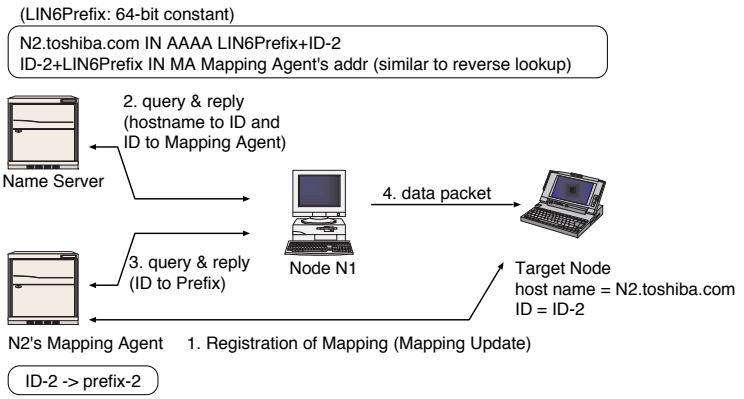


- LIN6 汎用識別子
  - ネットワーク層より上位で用いられる識別子
- LIN6 プレフィクス
  - 汎用識別子を生成するための固定値

LIN6 による通信手続き



Mapping の解決



ノードの移動: マッピングの更新

- 移動したノードからの通知
  - Mapping Update
  - Prefix Refresh Request
- 自律的な update
  - Mapping の lifetime expire
  - ICMP unreach

Mobile IPv6 に対する利点

- MA は HA のようにホームアドレスの位置に依存しない
  - 導入が容易
  - 耐障害性を高くできる
- ヘッダ長増大によるオーバーヘッドが無い
  - Mobile IPv6 で移動ノード間で通信すると Routing Header+Home address option で 48byte のヘッダオーバヘッド
- 三角経路が発生しない
  - つねに最適な経路での通信が可能

Summary: LIN6

- 位置指示子とノード識別子の概念を分離
- MA による高い対障害性
- 縮退アドレスによる効率的な移動透過性
- 現状
  - IETF draft: draft-teraoka-ipng-lin6-02.txt
  - Source code is available at http://www.lin6.net/
    - Running on
      - BSDs (NetBSD, BSDi, OpenBSD, FreeBSD)
      - Linux
      - Win2k

トランスポート層への支援: Performance Enhancing Proxy

Performance Enhancing Proxy (PEP)

- 無線リンクにおける TCP のための性能向上手段がいま大きな議論となっている
- リンク特性に依存して性能劣化が起きるものについて，どのように支援するか?
  - Transport Layer (TCP)
  - Application Layer (Web, Mail...)
- PEP による性能向上

TCP と無線リンク

- TCP は，パケットロスを輻輳としてとらえる
- TCP は，パケットロスをタイムアウトと重複 ACK で検出する
- 無線リンクはエラー率が高い
  - 輻輳がおきていないにもかかわらずパケットが落ちる
- 無線リンクは再送機能を持つ場合がある
  - TCP の再送タイマーと独立なので，無駄な再送が発生する可能性がある
  - RTT の揺らぎが大きく，再送タイムアウトの妥当性を失う

TCP のパフォーマンスを低下させる

Congestion Window (cwnd)

- 送信者が管理する window
  - いわゆる Window Size は受信側が管理する
- 送信者は，提示された window size と自分の持つ cwnd のどちらか小さい方まで送信可能
  - cwnd が大きくなるにつれてスループットが上昇する (一般的には)
- cwnd は，新しい ack を受けるたびに増加
- 輻輳が起きた場合には，送信者は送信量を減らす必要がある
  - cwnd を小さくする → congestion avoidance

TCP のタイムアウト

- TCP の送信者は再送タイマーをパケットに対してセットする
- そのパケットに対する ack が RTO (Retransmission TimeOut) 以内に来なかった場合，パケットは lost したと判断される．
- RTO は動的に計測される
- パケットロスは輻輳として扱われる
- タイムアウトが起きた場合，cwnd は 1MSS に戻り，slowstart する
- スループットは下がる

Fast Retransmit/Fast Recovery

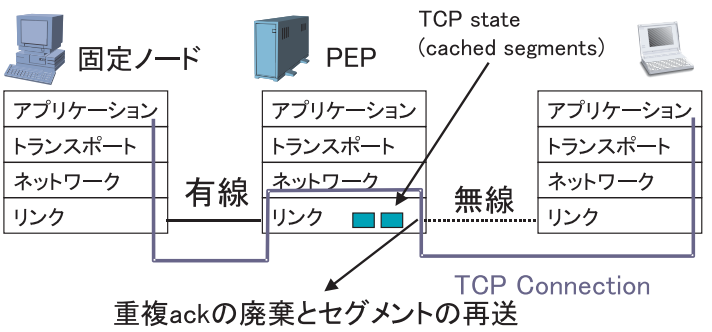
- タイムアウトには時間がかかる: より早く再送するためには
- 重複 ack:
  - パケットの喪失
  - パケットの乱順到着
- 3 つの重複 ack が届いたら，パケットの喪失が起きたと判断する
- cwnd は半分になる
- スループットは下がる

Random errorがTCPに与えるパフォーマンスの影響

- パケットがエラーで落ちる → Fast Retransmit の機能が働くが...
- 喪失したパケットの再送
- cwnd の縮小 → スループットの低下
- しかしエラーが原因なら (輻輳ではないので) cwnd の縮小は必要はない
- いかに送信者に対してエラーによるパケットロスを隠蔽するか

TCP-Aware Link Layer: snoop

- 再送を PEP のリンク層上での監視で行う



snoop

- PEP の Link 層で各 TCP セグメントをキャッシュ
- 重複 ack の監視: 観測された重複 ack は破棄され，キャッシュから適切なセグメントが再送される

snoop の特徴

- コネクションを分断せずにローカルリカバリーを実現
  - 無線リンク上でのパケットロスは固定ホストに伝わらない
- end-to-end セマンティクスの保存
  - 固定ホストに ack が返れば，移動ノードがそこまでのセグメントを受信したことを保証する
- ソフトステート
  - 通信中に異なる PEP に移動することは容易
  - 通信中に PEP がステートを失っても (パフォーマンスは低下するが)TCP セッションは継続可能

snoop の問題点

- 無線区間の遅延を隠蔽することはできない
  - 無線リンクの再送メカニズムによる遅延の揺らぎ
  - 再送タイムアウトの値が適切ではなくなる (無駄に長くなる) 可能性
  - 有線区間のパフォーマンスの低下
- 同様に，無線区間の一時的な disconnect も隠蔽できない (disconnect handling)
  - 送信者のタイムアウト再送が発生する

PEP の持つ問題点

- IPsec が使用できない
  - snoop は ESP されてなければ使用可能
- 非対称な経路での使用は難しい
- end-to-end セマンティクスを喪失する PEP もある
- single-point of failure の発生
- 障害時の問題発見の複雑化

Summary: 無線リンクにおける TCP の PEP

- 無線 link 上の TCP が持つ問題
  - TCP はパケットロスを輻輳とみなし，輻輳制御を行なう
  - 無線のエラーから輻輳制御等が働き性能が劣化
- 手法
  - パケットロスの隠蔽
  - 早い PEP からの再送
- 問題もある
  - e.g. single-point of failure
- PEP ではなく，SACK(RFC2883) の利用や帯域測定を用いた無線 link での TCP の性能向上方式も盛んに研究されている

Robust Header Compression (ROHC)

Robust Header Compression (ROHC)

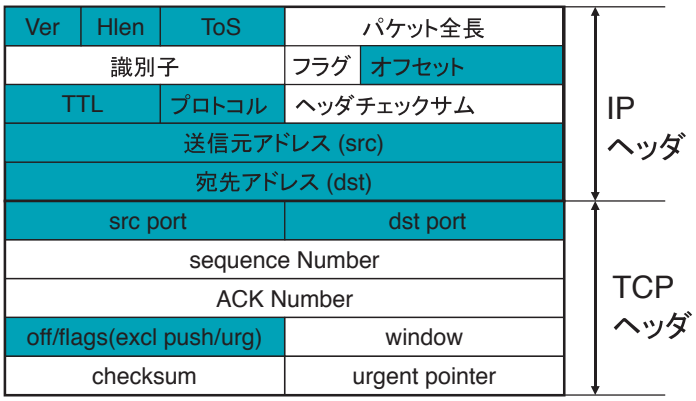
- 高い遅延，狭いバンド幅だけでなく，無線のような高いエラーレートのリンクに対応したヘッダ圧縮についての議論
  - rfc1144, rfc2508 では性能が出ない
  - SACK[rfc2018] 等の TCP option も圧縮対象
- IP/TCP/UDP/RTP がターゲット
- 3GPP/3GPP2 との連携
- IETF WG: http://www.ietf.org/html.charters/rohc-charter.html

過去のヘッダ圧縮: e.g. RFC1144

- aka VJ Compression or Compressed TCP(CTCP)
- TCP Connection が対象
  - c.f. RFC2508 (CRTP): Compresses IP/UDP/RTP
  - RFC1144 を拡張したもの
- 変化しないヘッダフィールドは送らない
- 変化するフィールドも，sequential なものは差分にして送る

IP/TCP ヘッダ

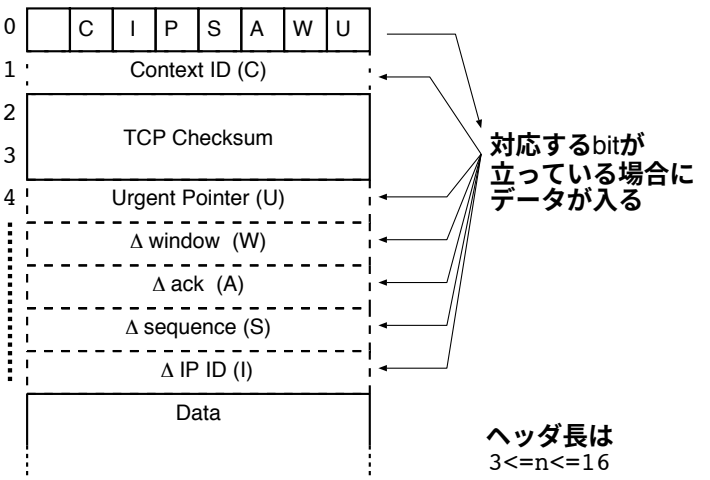
- 色がついている部分が (通常) 変化しない



状態 (context) の保存

- ヘッダの保存
  - connection ごとに，(再現された)TCP/IP ヘッダを保存しておく
- 保存されたヘッダは Context ID (CID) で識別
  - 圧縮されたパケットは，CID と変化した部分の差分となる

ヘッダ概略



パケットの差別化

- パケットを３種類に分ける
  - TYPE-IP: TCP ではないパケット (圧縮できない)
  - COMPRESSED-TCP: 圧縮されたパケット
  - UNCOMPRESSED-TCP: 圧縮されていないパケット
- Link Layer のヘッダを使ってこの情報を相手に伝える

パケットの喪失による状態の不一致 (out-of-sync) からの回復

- VJ Compression では，TCP の再送をトリガーに回復
  - Compressor が
    - sequence 番号の差分が 0 か負
    - duplicate ACK
- を検出した場合，UNCOMPRESSED-TCP を送信して状態を refresh する
- CRTP では明示的な signaling を使用する

ROHC の必要性：従来の方式ではなぜいけないのか

- Long-Thin Pipe の出現
  - 高遅延 (Long Delay)
  - 狭帯域 (Low Bandwidth)
  - 高いエラーレート (High Error Rate)
- 多くの無線ネットワークは Long-Thin Pipe

Long-Thin Pipeでの従来方式の問題

- **高エラーレート**: **高いパケットの喪失率**
  - out-of-sync **となりやすい**
- **高遅延**: out-of-sync **からの復帰が遅い**
  - **バーストロスの可能性**
- **挟帯域**: out-of-sync **がさらに帯域を消費**
  - context **の回復にはかならず全ヘッダを送信する**
- Long-Thin Pipe **では性能は出ない**

ROHC (RTP)

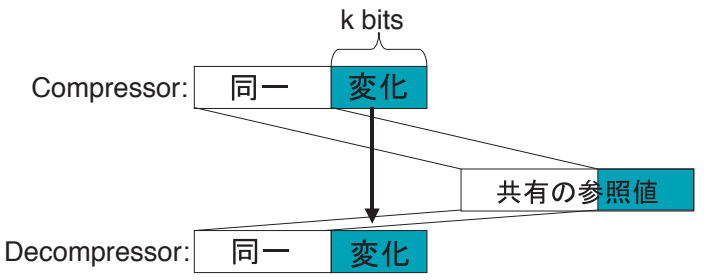
- RFC3095 (draft-ietf-rohc-rtp-09.txt)
- **対象となるプロトコル群**
  - RTP/UDP/IP
  - UDP/IP
  - ESP[rfc2406]/IP
- **変化しない部分は送らないという方針は同じ**
- **パケットの喪失・エラーに対して頑強** (Robust) **な ヘッダ圧縮**
- TCP **は別方式**

ROHC (RTP) (cont'd)

- Robustness の確保: パケットの喪失に強い Encoding
  - Window-based LSB Encoding
  - Scaled RTP Timestamp Encoding
  - Offset IP-ID encoding
- 属性に応じた encoding を使用
- CRC による Header reconstruct の正しさの検出
- 圧縮されるプロトコルは profile で識別

LSB encoding

- ある参照値に対して，LSB k bit の部分だけ送信する



- Decompressor は，最後に復号に成功した値を参照値として使用する

Window-Based LSB encoding

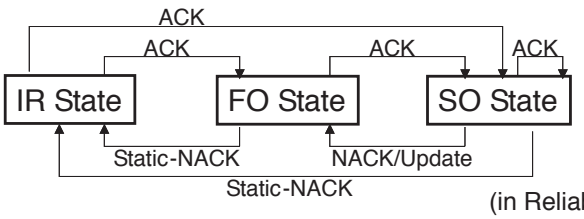
- Compressor は参照値を複数個持つ (Sliding Window)
- 送信した値は，参照値として保存される。
- $k = max(g(v, v_{min}), g(v, v_{max}))$
  - ただし v_{min}, v_{max} は参照値の最大・最小値。
  - $g()$ は参照値に対して最小の k が得られる関数
- Compressor は，Decompressor が参照値として使わなくなった値 (e.g. 最後に ack された値より古い値) は，参照値からはずす

ROHC (RTP) (cont'd)

- Link の状態に応じた 3 つのモード
- Unidirectional mode (U-mode)
  - C → D のみの path でも使用可能
- Bidirectional Optimistic mode (O-mode)
  - 基本的に C → D のみだが，error recovery など重要なものは D から feedback
  - feedback channel を sparse に保つ
- Bidirectional Reliable mode (R-mode)
  - 積極的に D → C を使う (e.g. context update ack)
  - Robustness を最大化

ROHC (RTP) (cont'd)

- **効率の向上：3 つの状態に分ける**
- e.g. Compressor
  - Initialization and Refresh (IR)
  - First Order (FO)
  - Second Order (SO)



(in Reliable Mode)

その他の議論

- Signaling Compression
  - SIP **等での** signaling packet **の圧縮方式**
- Universal Decompressor Virtual Machine (UDVM)
  - **圧縮のための** virtual machine **提案**
- TCP
  - RFC3095 **ベース**
  - **短いコネクションも効率的に圧縮**
    - e.g. Context replication
  - TCP options **も対称**: e.g. SACK
- Lower Layer Guidelines for ROHC
  - **二層の設計への要求．**Cellular **についての項目もある**

Summary: ROHC

- 狭帯域の通信においては，ヘッダが占める overhead は多い
  - 特に payload の小さいデータを送る voice streaming 等
- Long-Thin Pipe では，従来のヘッダ圧縮では性能が出ない
- パケットロスに対する耐性 (Robust)
  - e.g. LSB encoding
- フィードバック方式の選択性
  - 回線の有効利用