

## 本資料について

---

- 本資料は下記の文献を基にして作成されたものです。文章の内容の正確さは保証できないため、正確な知識を求める方は原文を参照してください。
- 文献名 IPv4+4
- 著者 Z. Turanyi, A. Valko
- 発表 ICNP2002
  
- 文献名 IPNL: A NAT-Extended Internet Architecture
- 著者 Paul Francis , Ramakrishna Gummadi
- 発表 SIGCOMM 2001

# IPv4+4 , IPNL

名城大学 理工学部 情報科学科 渡邊研究室  
11302j056 後藤 裕司

## はじめに

- インターネットが直面している大きな問題の一つにIPv4アドレスの枯渇
- 1994年にIETF Address Lifetime Expectations (ALE)がIPv4アドレスは2008年に枯渇
- NATが導入
- NATsはアドレス減少の速度を低下
- NATsには欠点がある
- エンドツーエンドで通信を可能にするため、セキュリティの低下、多くのアプリケーションのプロトコルを破ってネットワークを複雑化している。
- IPv4アドレスの枯渇の解決策にIPv6
- NATsの問題を解決するためにIPv4+4を開発

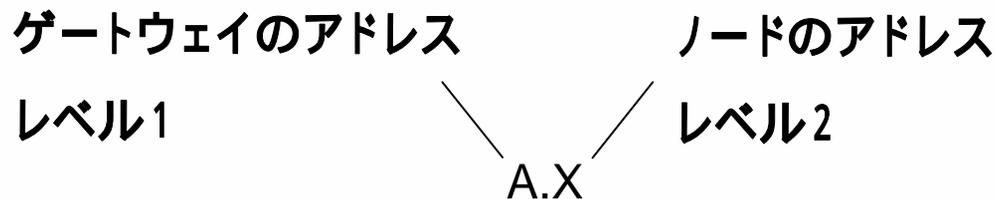
## IPv4+4 特徴

---

- 2つの32ビットのIPv4アドレスを連結
- IPヘッダーをカプセル化する
- IPv4アドレスを入れ替える
- 外側のヘッダはルータに解釈できるアドレス

## IPv4+4 アドレスの指定

- IPv4+4のアドレスはA.Xのように表される
- Aはゲートウェイのアドレス, Xはノード自身のアドレス
- Aはレベル1, Xはレベル2と呼ばれる
- DNSはIPv4と同様, IPv4+4アドレスを格納・読み出しに用いられる
- Aレコードはレベル1, レベル2にノードのアドレスを保存
- ドメイン名を通してアクセスできる
- DNSサーバの改良を必要としない

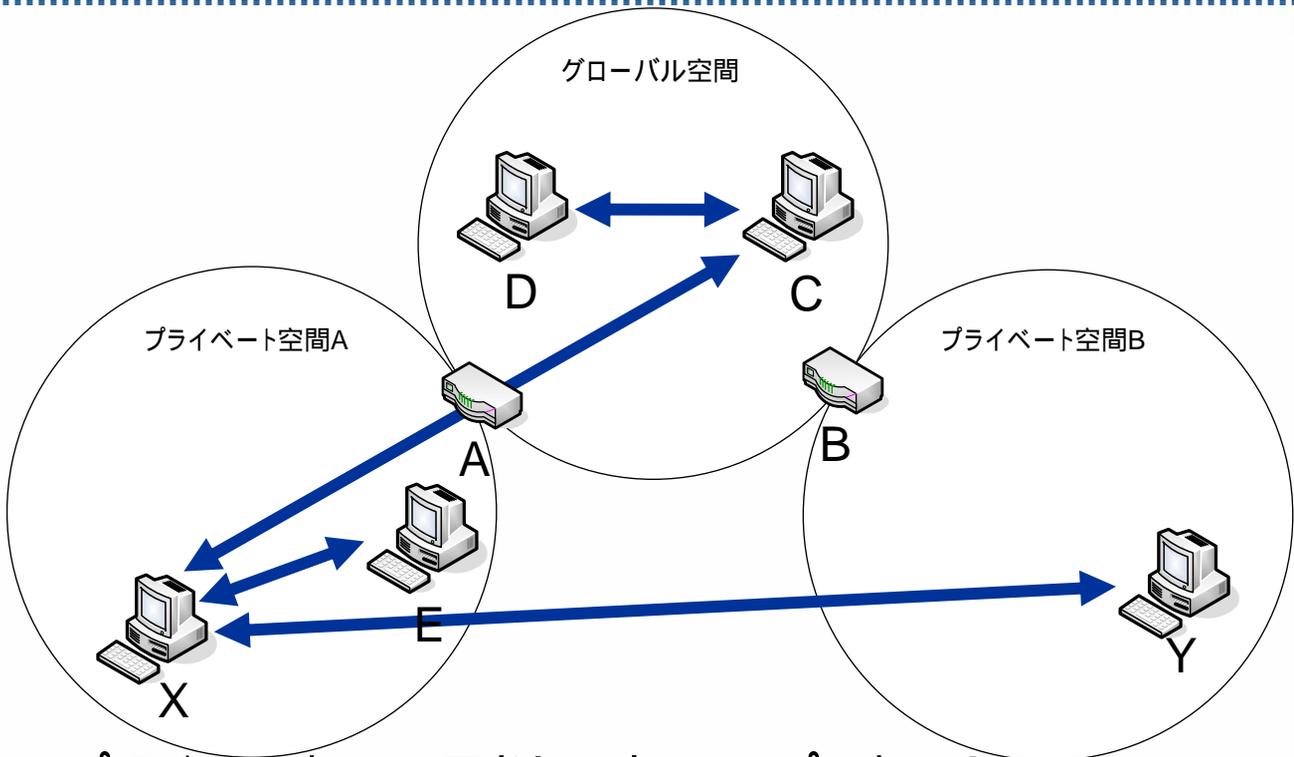


## IPv4+4 ヘッダー構成

|                       |      |            |                   |                 |
|-----------------------|------|------------|-------------------|-----------------|
| Ver                   | Hlen | DS byte    | Total Length      |                 |
| Identification        |      |            | Flg               | Fragment offset |
| TTL                   |      | Protocol 1 | Header Checksum 1 |                 |
| Source Address 1      |      |            |                   |                 |
| Destination Address 1 |      |            |                   |                 |
| Source Address 2      |      |            |                   |                 |
| Destination Address 2 |      |            |                   |                 |
| Protocol 2            | SPos | DPos       | Header Checksum 2 |                 |

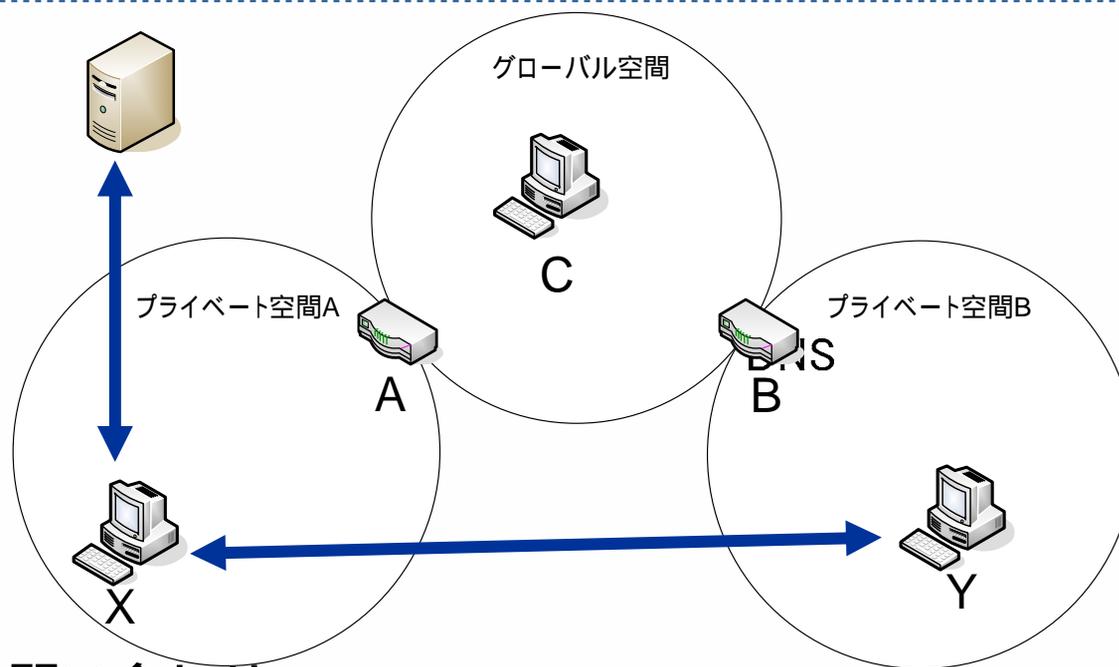
SposとDposはIPv4+4アドレスがどのように交換されるか示す  
 値が0であれば交換しない、1であれば交換

# IPv4+4 ルーティング1



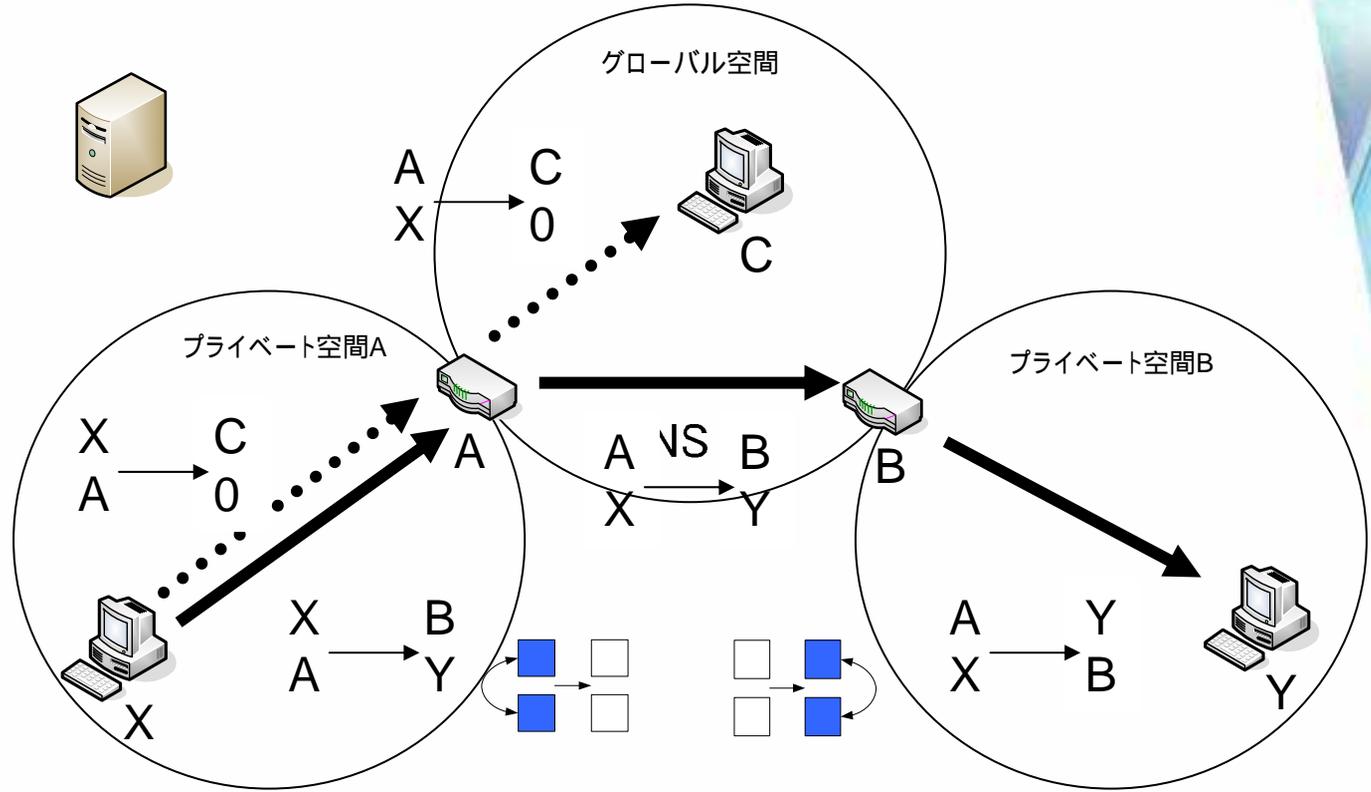
プライベートIPv4アドレスとIPV4プロトコル  
グローバルIPv4アドレスとIPv4プロトコル  
NATを使用  
IPv4+4を使用しないと不可能

## IPv4+4 ルーティング2



- DNSに問い合わせ
- レベルのいずれかにアドレス部と一致するものかどうかを調べる
- 一致すれば同じ空間内なのでIPv4を使用
- 一致しなければIPv4+4を使用
- ノードXはレベル1, 2を選択し送信元アドレスX.A宛先アドレスB.Yとする

# IPv4+4 ルーティング3

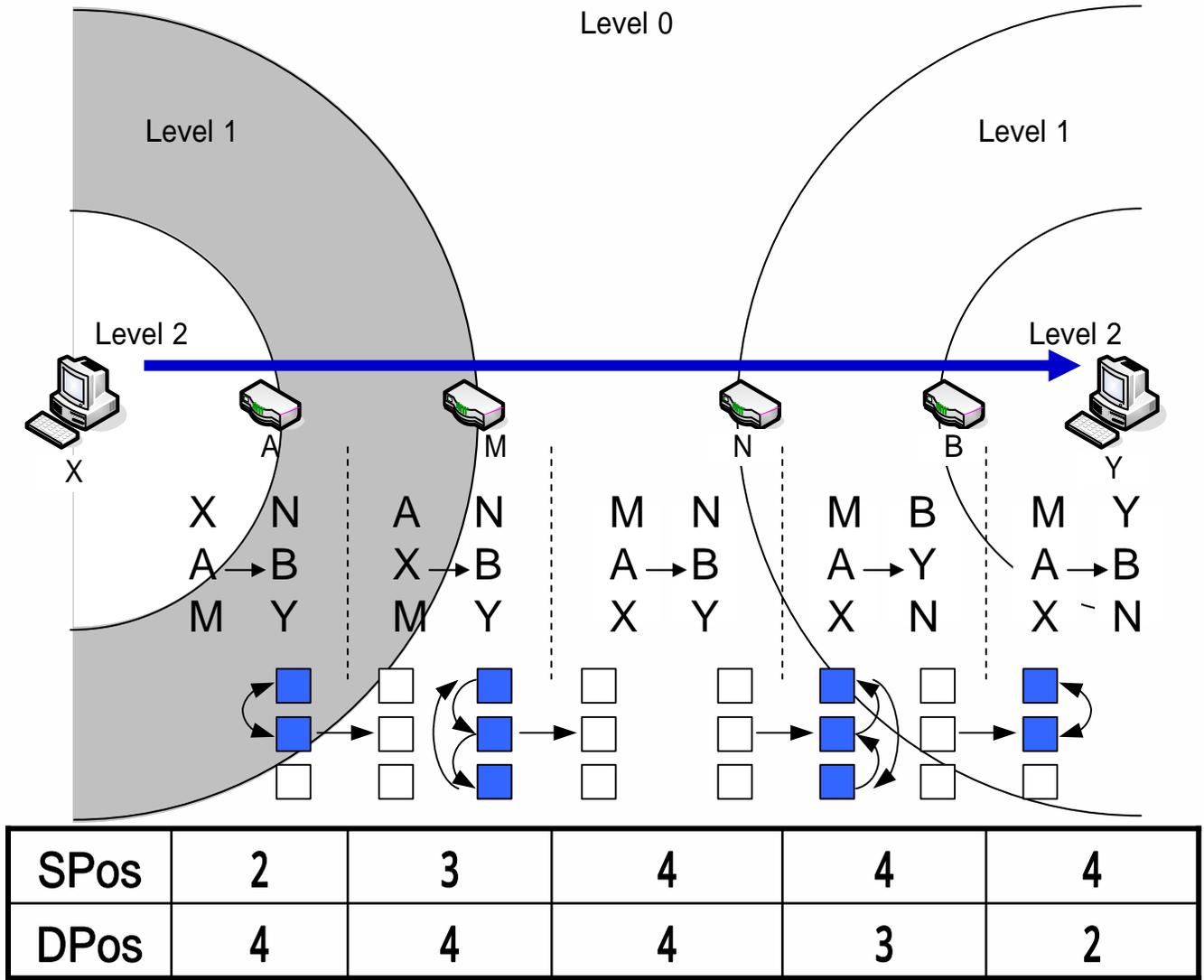


レベル1とレベル2を入れ替える

両端末がIPv4+4が可能ならばIPv4+4ヘッダーを使用することが可能

ノードCのIPv4+4アドレスはC.0

# IPv4+4 アドレス空間のさらなる拡張



## IPv4+4 評価

---

- Linux2.4.18でIPv4+4の初期バージョンを作成
- 2000行のCのコード
- メモリ256MB, CPU1GHzのPentium
- アドレス交換のオーバーヘッドは1.8マイクロ秒

## IPv4+4 まとめ

---

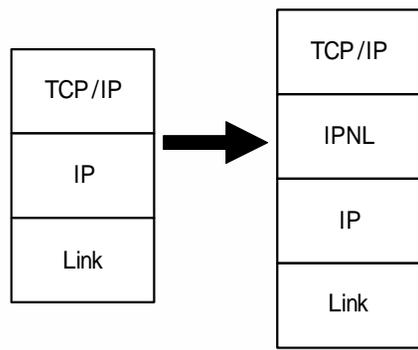
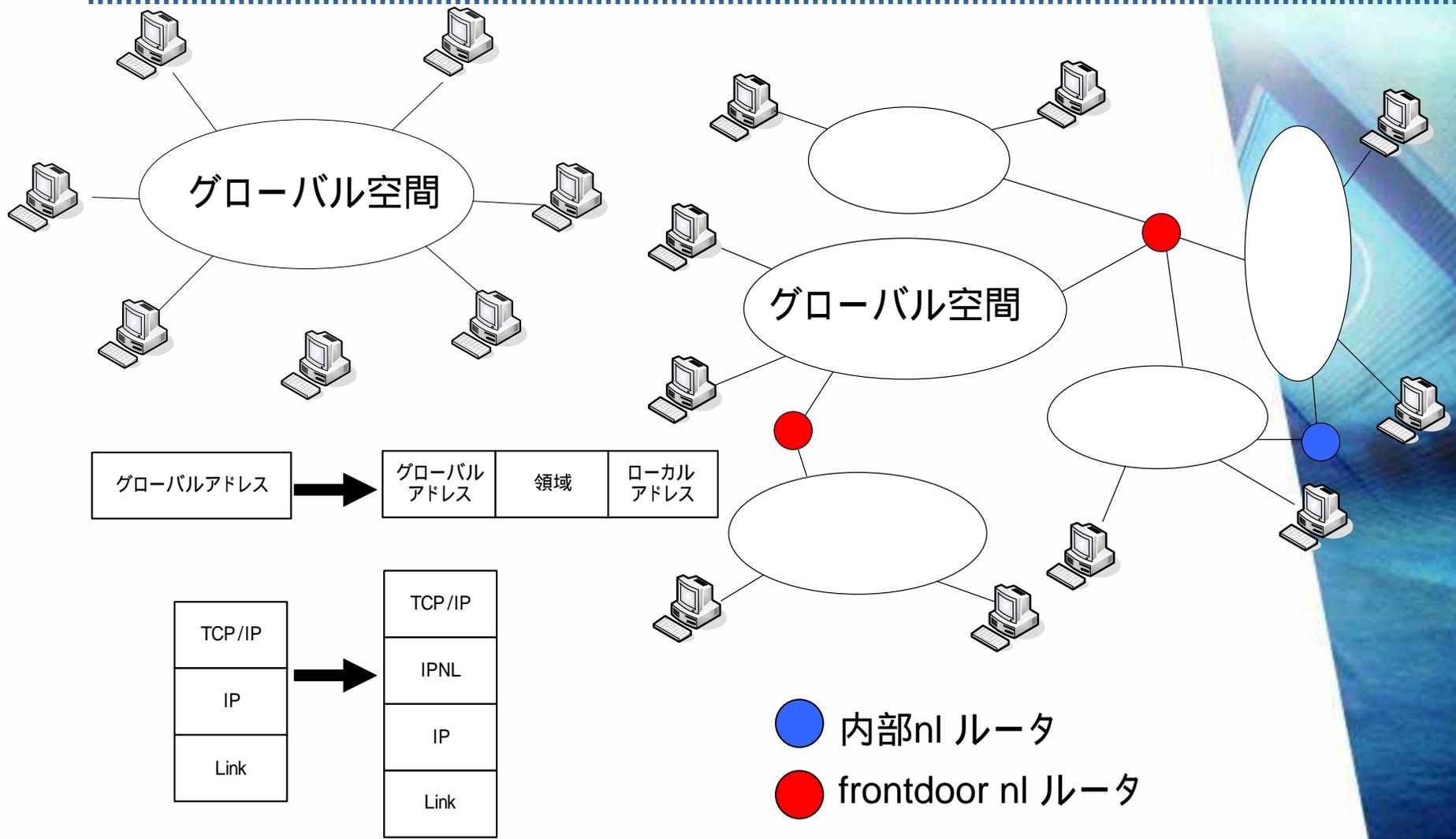
- 非常にシンプルな方法
- 異なるプライベート空間同士でも通信できる
- オーバーヘッドはそれほど大きくない
- IPv6が広まると利点がなくなる

## IPNL (IP Next Layer)

---

- IPより上に新しい層を加える
- IPv4の機能をそのまま使うことが可能
- パケットにはエンドツーエンドのホストの識別子
- FQDN (Fully Qualified Domain Names) ホスト名 + ドメイン
- NATボックスはnl-ルータと呼ばれる
- Site内部のnl-ルータを内部ルータとも呼ぶ
- プライベート領域を中央でつなぐnl-ルータはfrontdoor nl-ルータまたはfrontdoorと呼ばれる

# IPNL ネットワーク形態



## IPNL アドレス

- IPNLヘッダは2種類のroutableなアドレスを運べる
- ホストが接続を始めるとき, 3つのアドレスが必要
- 自身のFQDN, EHIP,宛先のFQDN
- 最初のパケットにはFQDNが送られる
- 以降のパケットはIPNLアドレスだけを送る

### IPNL Address Format

|             |               |             |
|-------------|---------------|-------------|
| MRIP(4byte) | RN<br>(2byte) | EHIP(4byte) |
|-------------|---------------|-------------|

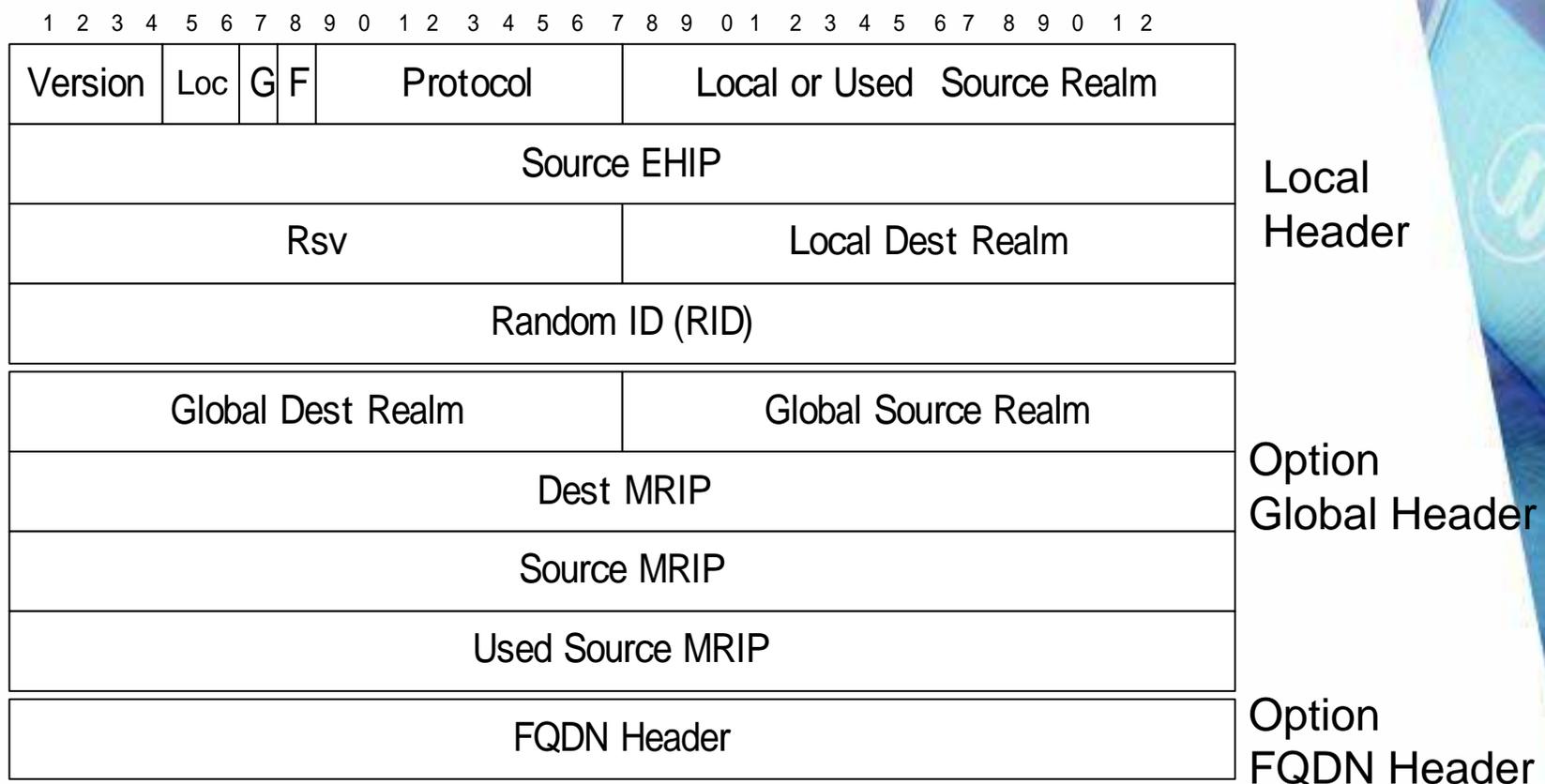
MRIP: Middle Realm IP Address

RN: Realm number

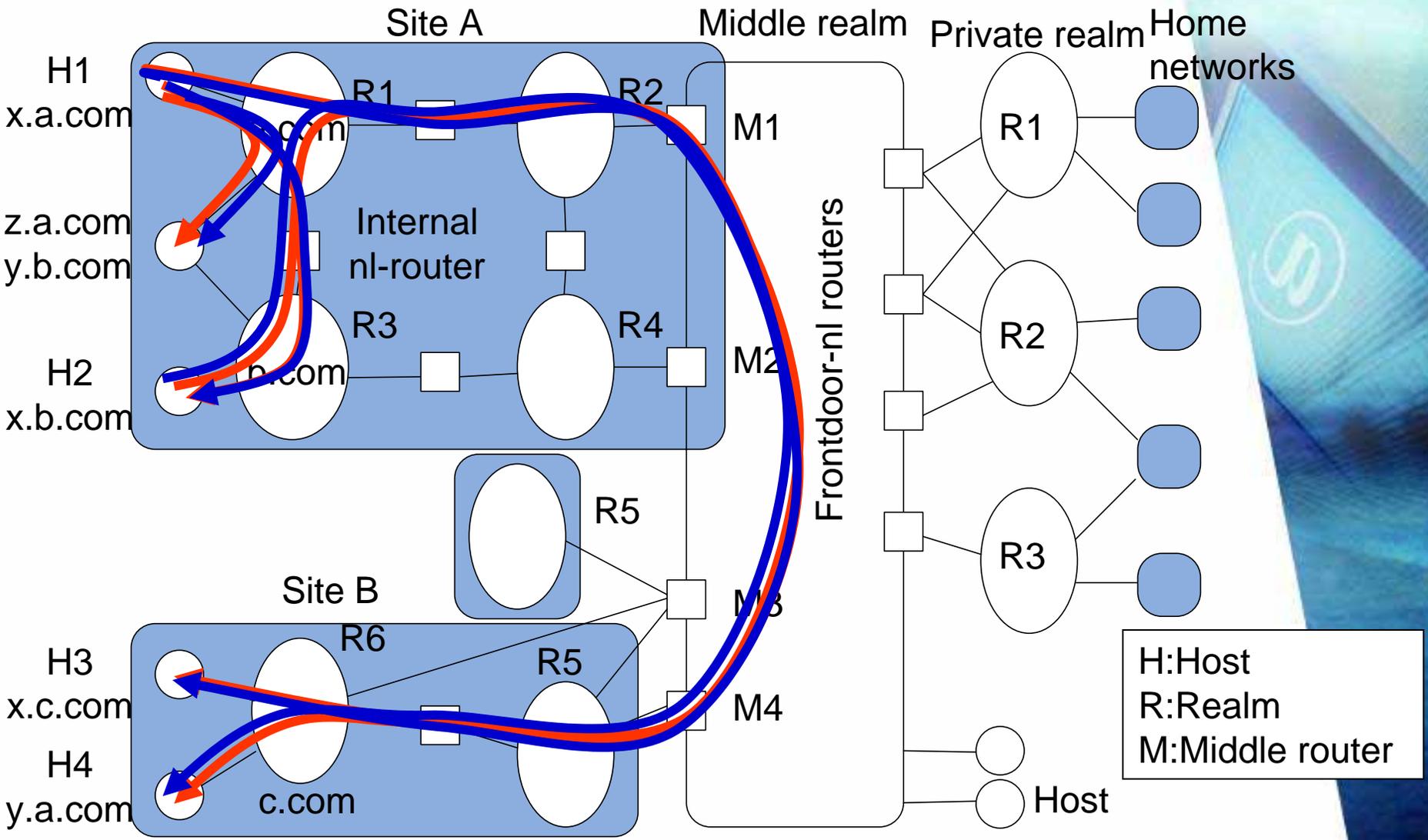
EHIP: End Host IP

# IPNL フォーマット

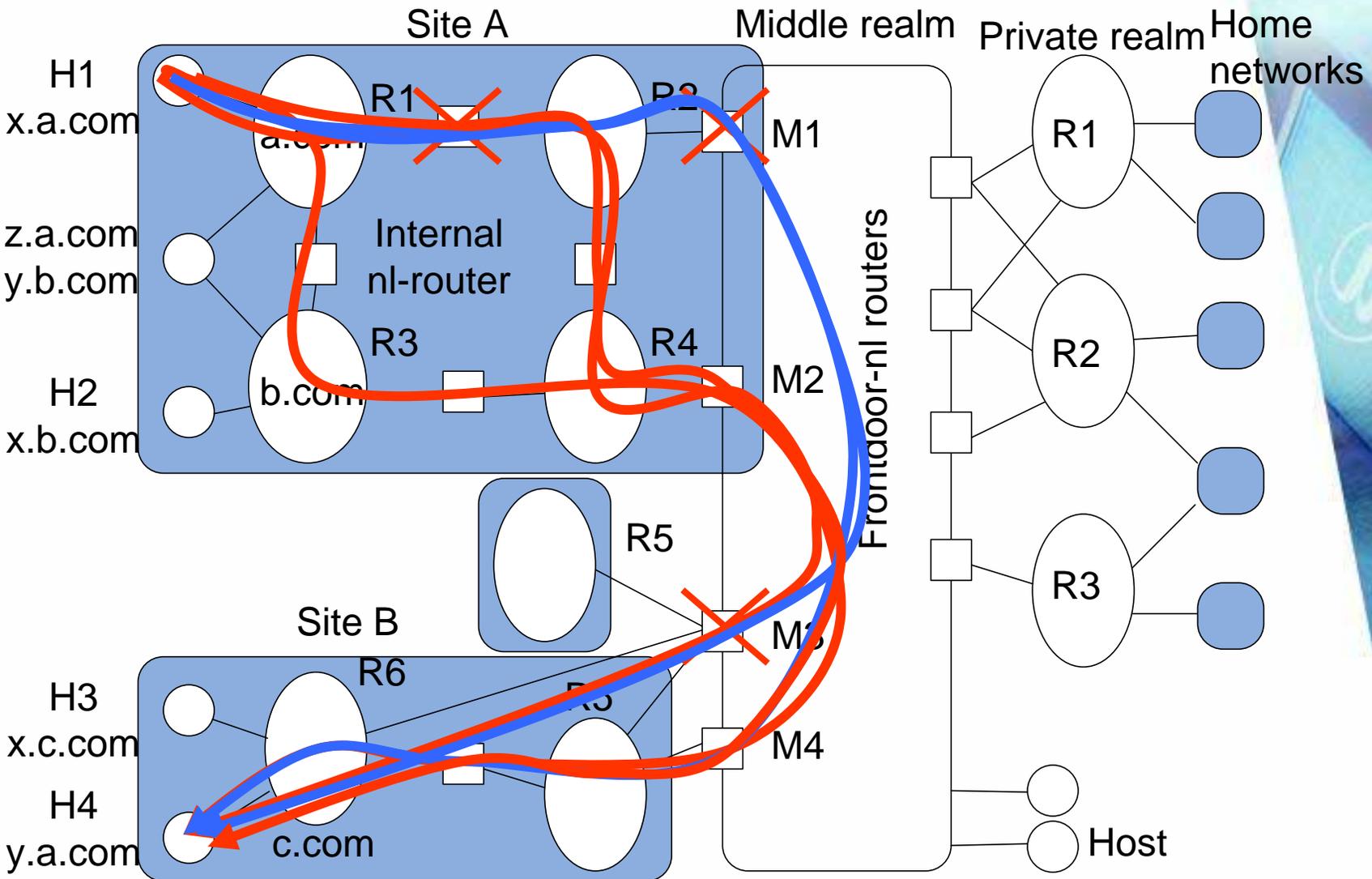
## IPNL Header Format



# IPNL ルーティング1



# IPNL ルーティング2



# IPNL ルーティング3

H1 H4

ホップ値  
R1-R2

R1-R2が故障

ホップ値  
R1-R3

ホップ値: 次の転送先のルータのアドレスを示す

|   |                  |
|---|------------------|
|  | Source MRIP      |
|  | RN               |
|  | Used Source MRIP |
|  | Dest MRIP        |

H1 H3

故障前



ルータ M1  
が故障

M2がパケット受信時



M2に変更

M4がパケット受信時



M1, M3が  
同時に故障



M2に変更



M4に変更

## IPNL 評価

---

- Linux2.2.16カーネル
- NICカードは3Com 3c905B 10/100Mbpsを使用
- IPNLによるオーバーヘッドを測るベンチマーク
- 1500バイトのパケットを99.1Mbpsのスループット

## IPNL まとめ

---

- 異なるプライベート空間同士でも通信できる
- ホストが移動してもFQDNがサポート
- 障害が起こっても通信することが可能
- オーバーヘッドもほとんどない

---

# 終わり