# インターネットはプロトコルでつながっている
# 当日資料

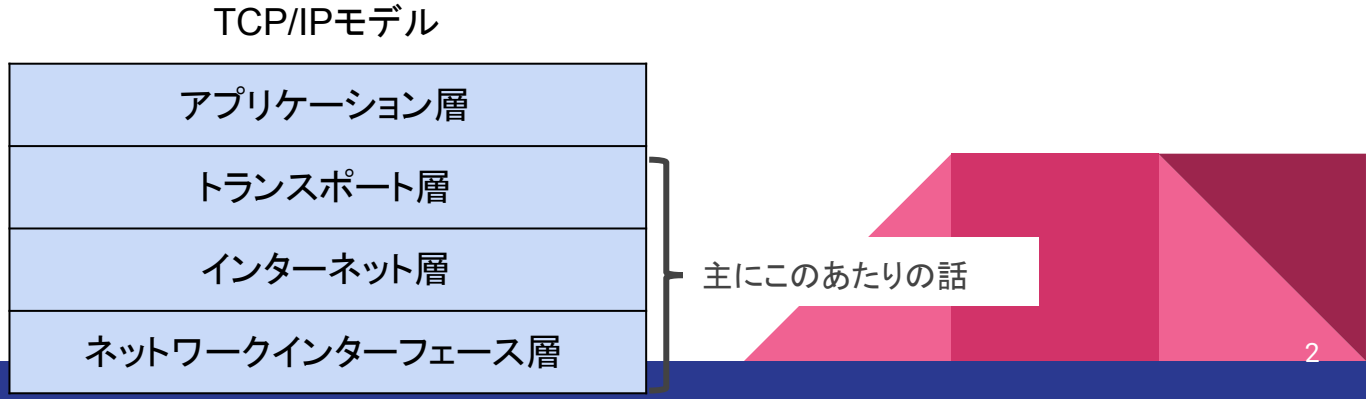
## ネットワーク運用チュートリアル2019

2019-11-26 InternetWeek 2019

株式会社朝日ネット
Megumi Takagi

# 申し送り

- 本資料はインターネット接続に際し使われる様々なプロトコルを紹介することを目的としています
- 主に，ネットワークインターフェース層～トランスポート層を取り上げます
- 正確性よりも分かりやすさを優先しているため，詳細について実際と異なる場合があります
- 特定の組織・企業等を宣伝する意図はありません
- 本資料は事前資料から一部ページの加筆修正を行っていますが，お手元の資料が事前資料でも特に問題ありません



# 自己紹介

たかぎ めぐみ (@Motsuo_p)

株式会社朝日ネット　ネットワーク部 (社会人2年目)

・ISP，VNEバックボーンの設計構築運用（ASAHIネット，v6コネクト）

・家庭用ルータの動作検証

InternetWeek2019 プログラム委員 兼 NOCチームリーダ

JANOG若者支援プログラム サポーター

# 目次

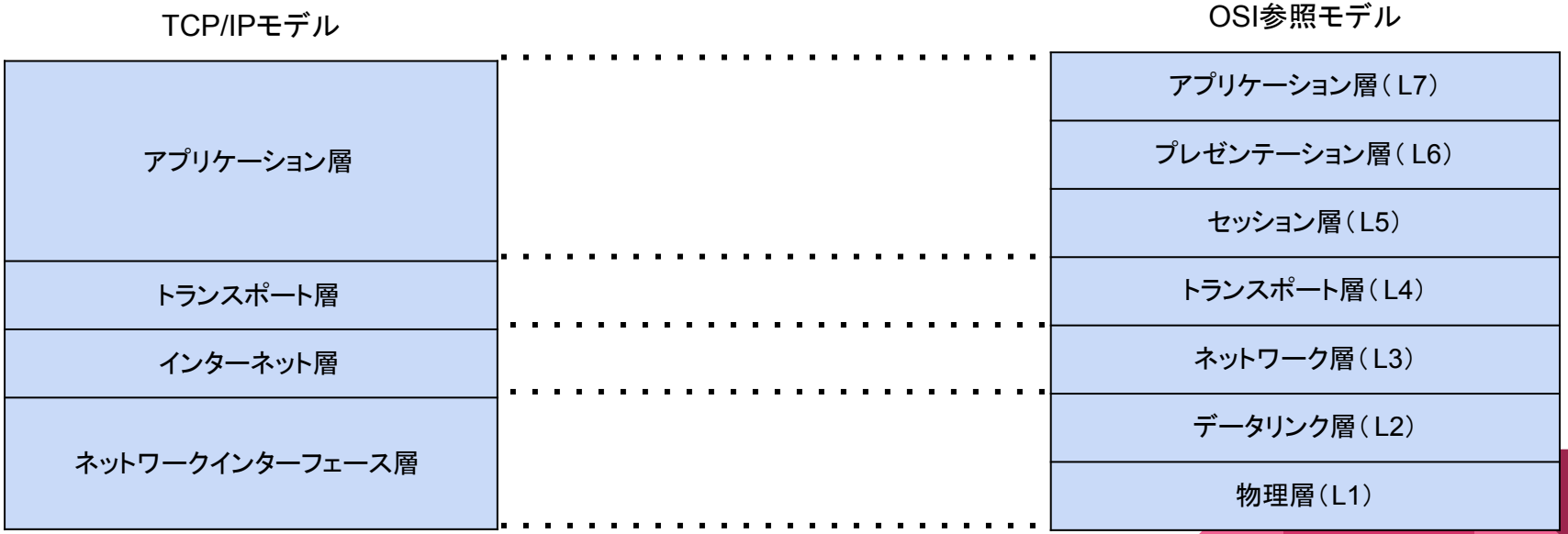
- プロトコルについて
  - プロトコルとは
  - プロトコルスタックとは（TCP/IPモデル）
- パケット・フレームについて
  - データとヘッダ
  - カプセル化と非カプセル化
  - Protocol Data Unit
- IP通信の例
  - HTTPリクエストとHTTPレスポンス
- 名前解決について
  - DNSの仕組み
- 動的ルーティングについて
  - IGPとEGP
  - BGP
- BGP
  - ASとベストパスセレクション
  - ピアリングと経路選択
- まとめ

# プロトコルとは

- 相手と会話するために決めるルール，約束事（表現方法＋話す手法）


# プロトコルスタックとは

- “stack” - 積み重ね。機能毎「**層**」に分けて考える




分け方・とらえ方の違いのみで，本質は同じ

# TCP/IP モデル

- 現在インターネット通信で主に使用されているプロトコルスタック

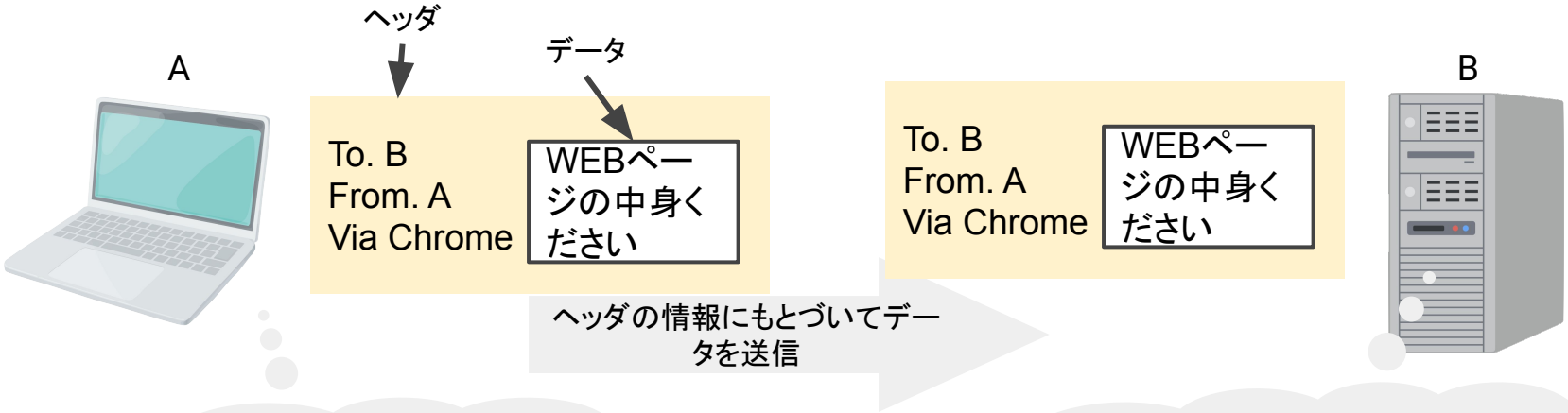
TCP/IPモデル

| |
|---|
| アプリケーション層<br>（HTTP，FTP，SMTP，DHCP，DNS等） |
| トランスポート層<br>（TCP，UDP等） |
| インターネット層<br>（IP） |
| ネットワークインターフェース層<br>（Ethernet，PPP，ATM等） |

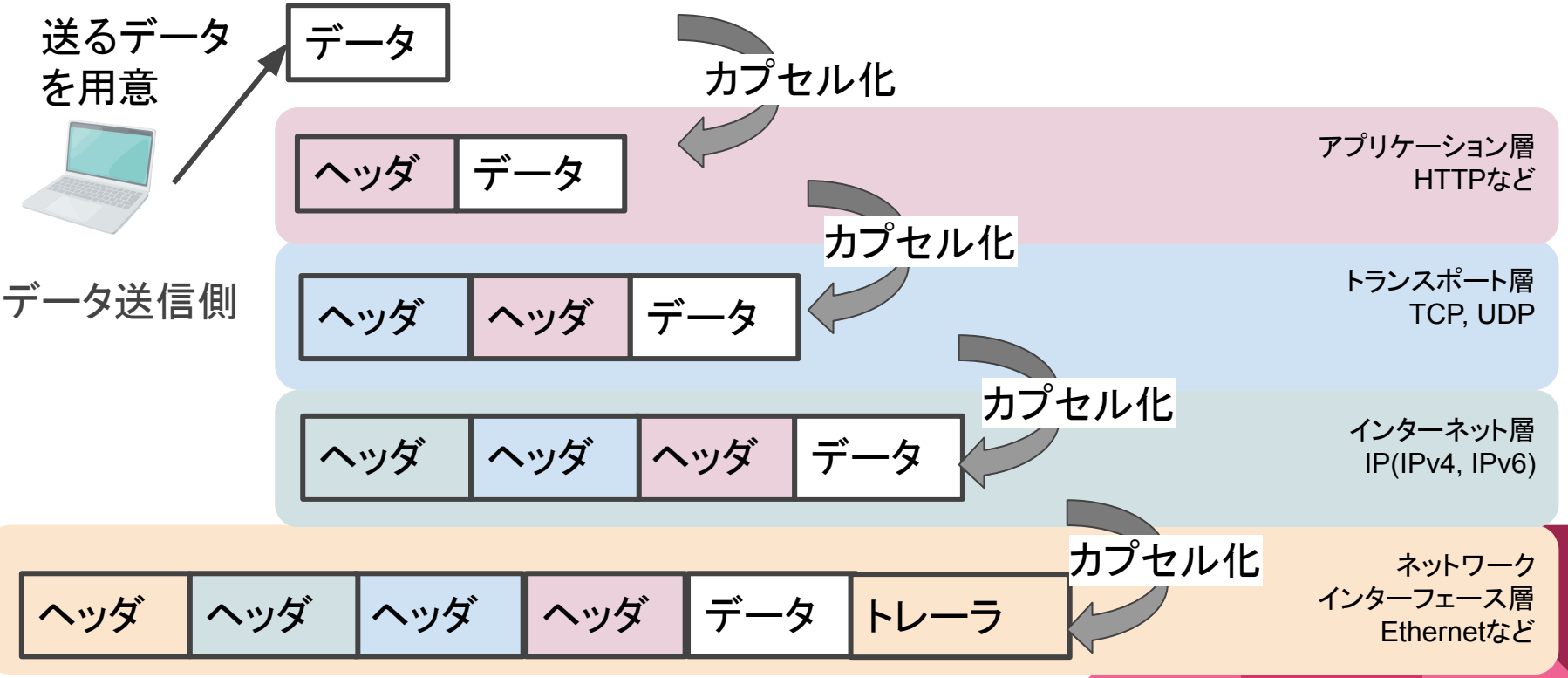
プロトコルスタックイメージ

# データとヘッダ

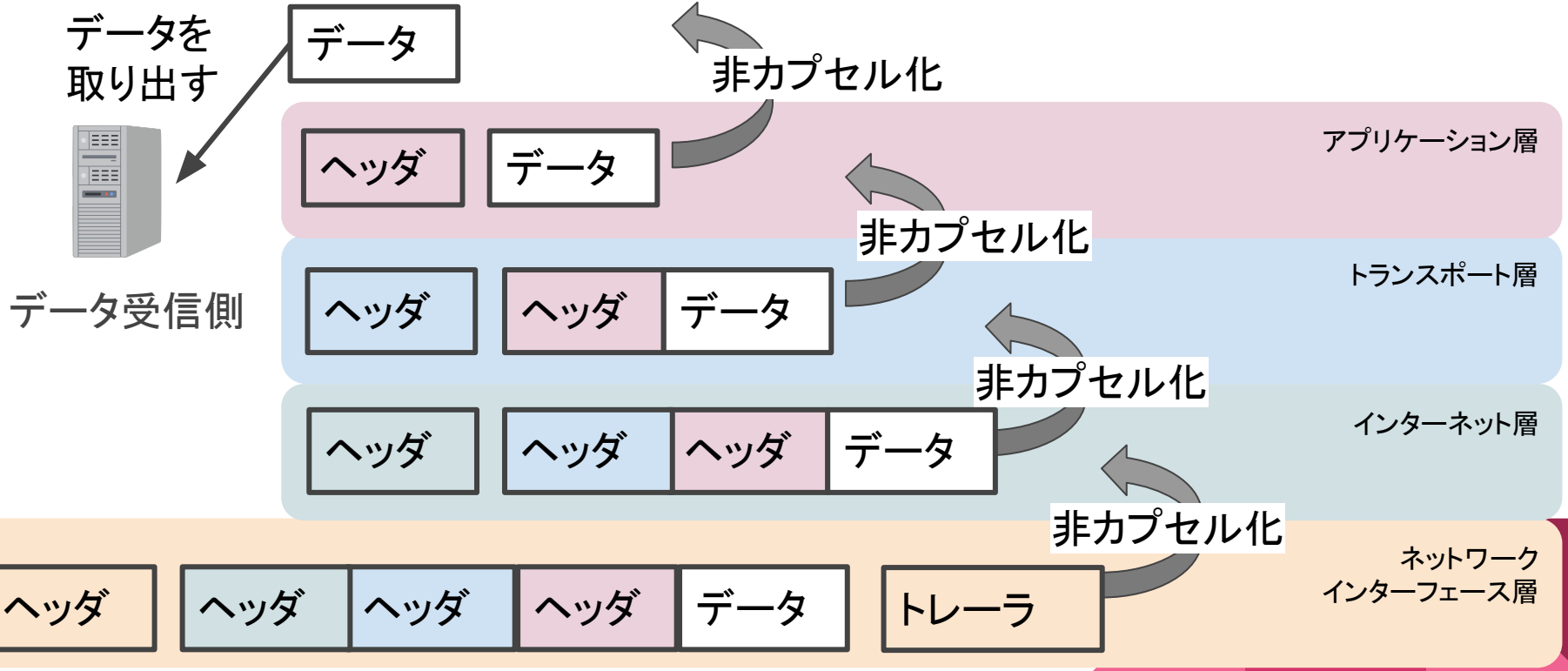
- 送信するデータに，各層の処理に必要な情報「**ヘッダ**」を付加する



# カプセル化と非カプセル化

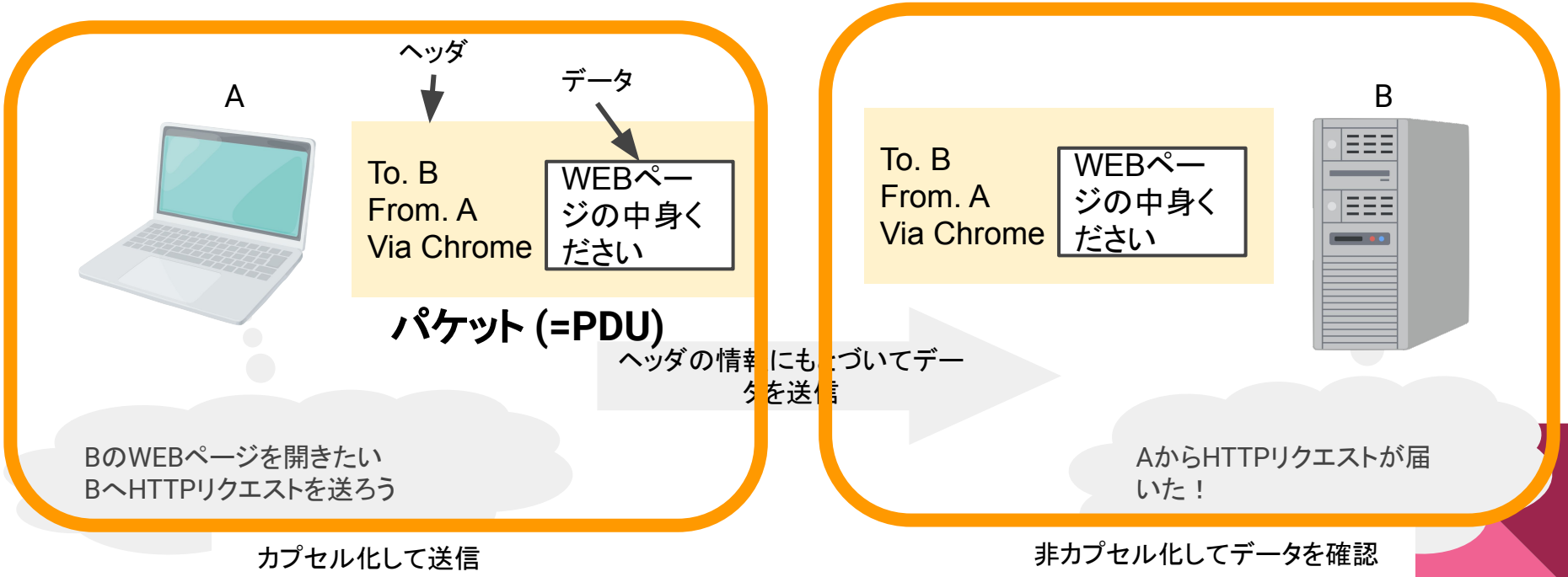


# カプセル化と非カプセル化



# データはPDUとして送受信される

- PDU (Protocol Data Unit) = カプセル化されたデータ（**パケット**とよばれる）



# PDUの名称

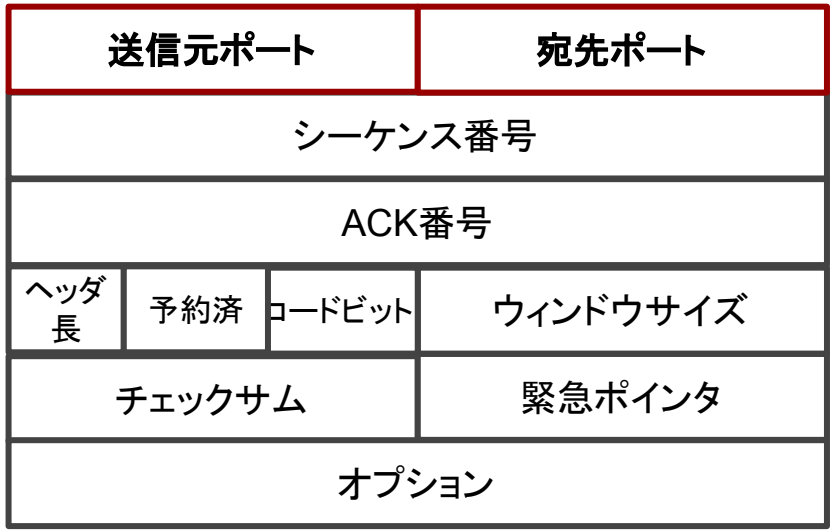
- **セグメント**※ e.g. TCPセグメント・UDPセグメント ※パケットと呼ばれることが多い


- **パケット** e.g. IPパケット

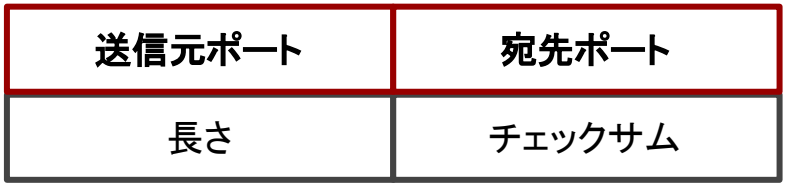

- **フレーム** e.g. Ethernetフレーム

# TCPヘッダ，UDPヘッダ

## TCPヘッダ※



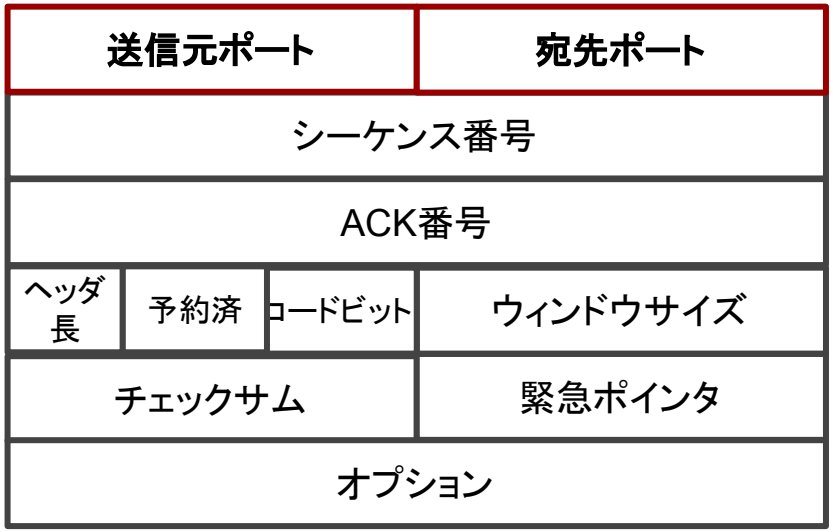

## UDPヘッダ※




※ここではトランスポート層で付与されるヘッダ部分（他層のヘッダ，データを除いた部分）を　TCP/UDPヘッダとしています

# TCPとUDP

## TCPヘッダ※




## TCP

・確認応答を行う
・シーケンス番号でデータの順番を管理
・輻輳制御

といった動きをする。

「信頼性のある通信」を実現

※ここではトランスポート層で付与されるヘッダ部分（他層のヘッダ，データを除いた部分）を　TCP/UDPヘッダとしています

# TCPとUDP

## UDP

・確認応答は行わない（コネクションレス）
・信頼性は確保しない
・処理が単純で遅延が少ない

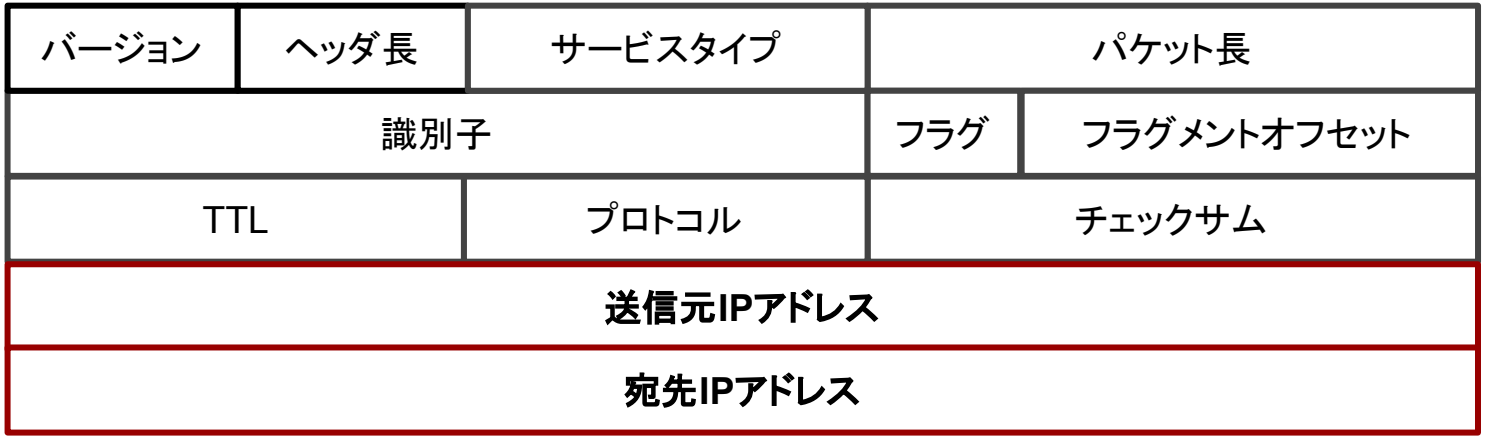
UDPヘッダ※

| 送信元ポート | 宛先ポート |
| --- | --- |
| 長さ | チェックサム |

※ここではトランスポート層で付与されるヘッダ部分（他層のヘッダ，データを除いた部分）を　TCP/UDPヘッダとしています

# IPヘッダ，ICMPヘッダ

## IPヘッダ




## ICMPヘッダ

# Ethernetヘッダ


※Ethernetフレームはヘッダの他に，トレーラ が付く

| 宛先MAC アドレス | 送信元MAC アドレス | タイプ | インターネット層 ～アプリケーション層のヘッダ＋データ | トレーラ （FCS） |
|---|---|---|---|---|

※ トレーラは，エラーチェックのフィールドとして使用される

# 今回注目したい要素　～相手へ届けるために～

- トランスポート層：「宛先ポート」「送信元ポート」（TCP or UDP）
- ネットワーク層　：「宛先IPアドレス」「送信元IPアドレス」
- データリンク層　：「宛先MACアドレス」「送信元MACアドレス」

通信はかならず送受信 → 宛先情報 と 送信元 (送り主) 情報 がセット

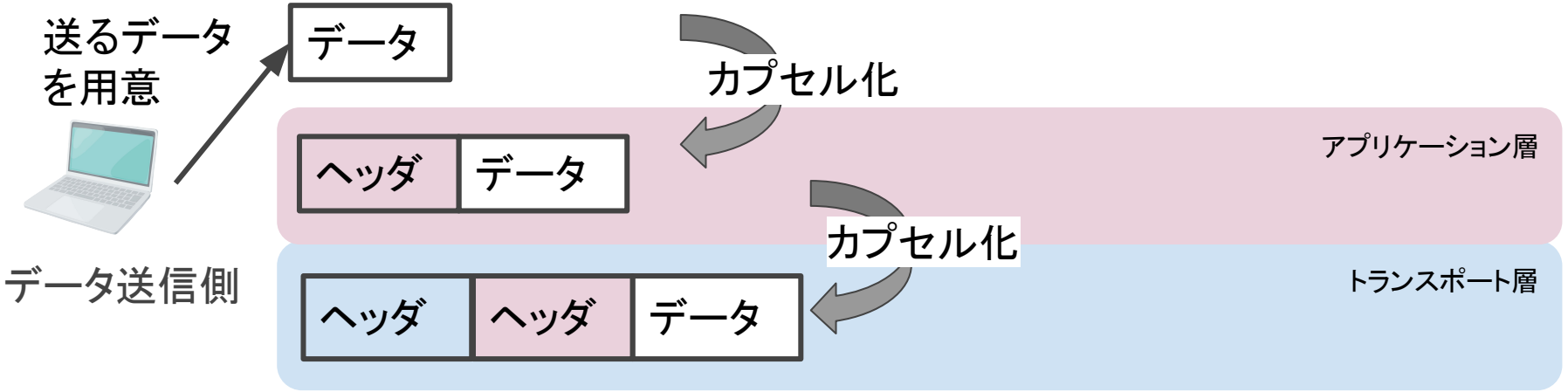
# WEBページのデータを要求する（リクエスト）

- WEBページのデータなので，使用するプロトコルは HTTP → 宛先ポートはTCPの80番

192.168.1.100 のWEB
ページデータください

| | |
|---|---|
| HTTPリクエストデータ<br>(HTTPヘッダ含む) | 宛先ポート 80<br>送信元ポート 65500<br>（TCP） |

送信元ポート番号は，1025～65535 のランダムポートから割り当てられる。

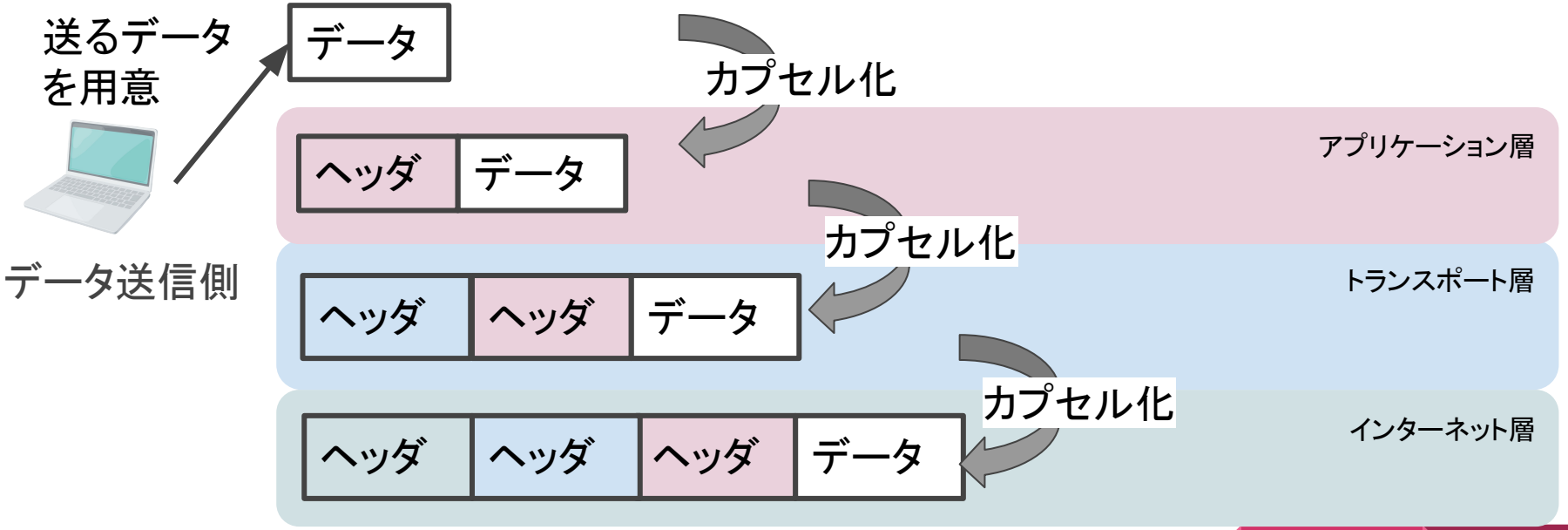


# WEBページのデータを要求する(リクエスト)

- 宛先192.168.1.100, 送信元 192.168.2.1 の場合

192.168.1.100 のWEBページデータください

| | | |
|---|---|---|
| HTTPリクエストデータ<br>(HTTPヘッダ含む) | 宛先ポート 80<br>送信元ポート 65500 | 宛先IPアドレス<br>192.168.1.100<br>送信元IPアドレス<br>192.168.2.1 |

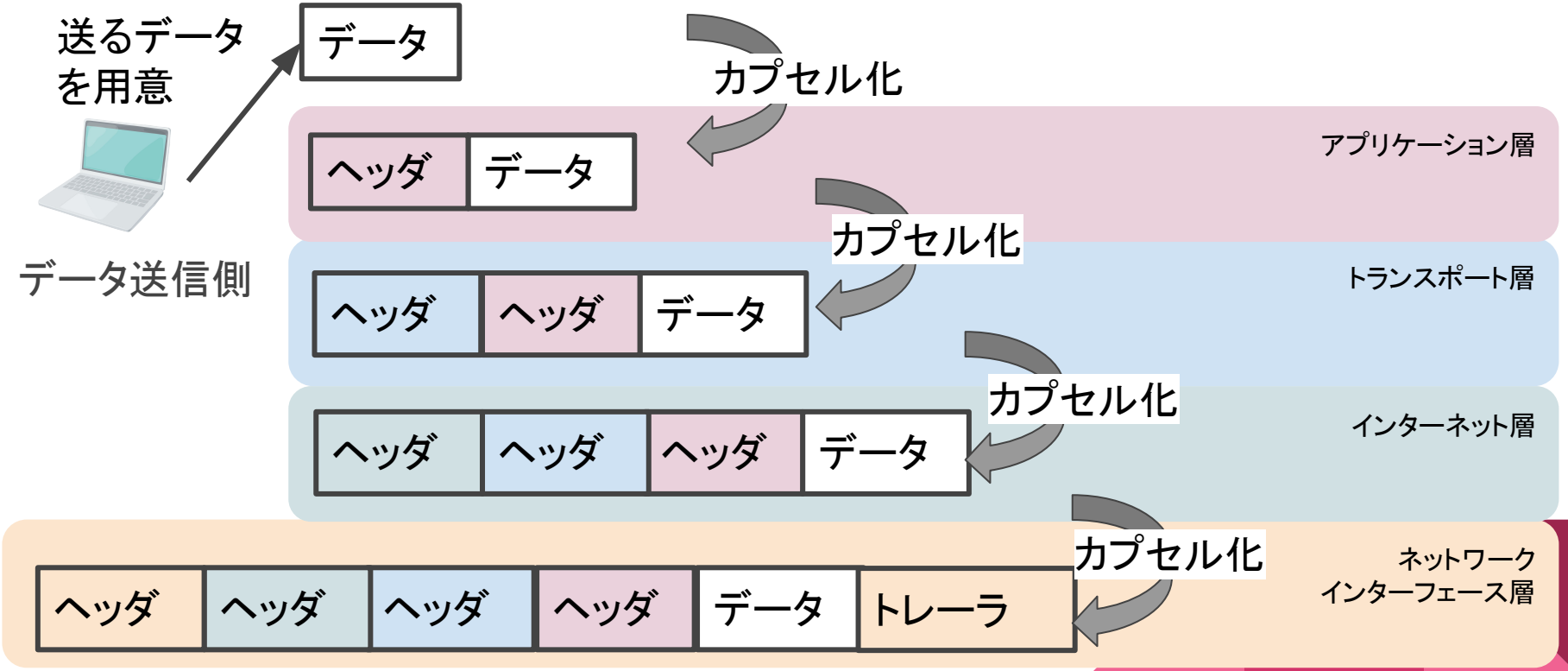


# WEBページのデータを要求する(リクエスト)

- 宛先 00-00-5E-EF-10-00-00-11，送信元 00-00-5E-EF-10-00-00-AA の場合

192.168.1.100 のWEBページデータください

| | | | |
|---|---|---|---|
| HTTPリクエストデータ<br>(HTTPヘッダ含む) | 宛先ポート 80<br>送信元ポート 65500 | 宛先IPアドレス<br>192.168.1.100<br>送信元IPアドレス<br>192.168.2.1 | 宛先MACアドレス<br>00-00-5E-EF-10-00-00-11<br>送信元MACアドレス<br>00-00-5E-EF-10-00-00-AA |



# WEBページのデータを要求する(リクエスト)

192.168.2.1

192.168.1.100 の
WEBページデータください

| | | | |
|---|---|---|---|
| HTTPリクエストデータ<br>(HTTPヘッダ含む) | 宛先ポート 80<br>送信元ポート 65500 | 宛先IPアドレス<br>192.168.1.100<br>送信元IPアドレス<br>192.168.2.1 | 宛先MACアドレス<br>00-00-5E-EF-10-00-00-11<br>送信元MACアドレス<br>00-00-5E-EF-10-00-00-AA |

スイッチ・ルータといったネットワーク機器が宛先へパケットを転送

192.168.1.100

受信したパケットを非カプセル化してデータ確認

WEBページデータリクエストですね。

# WEBページのデータを受け取る（レスポンス）

192.168.2.1

受信したパケットを非カプセル化してデータ確認

192.168.1.100 のWEB
ページデータ受け取り

スイッチ・ルータといったネットワーク機器が宛先へパケットを転送

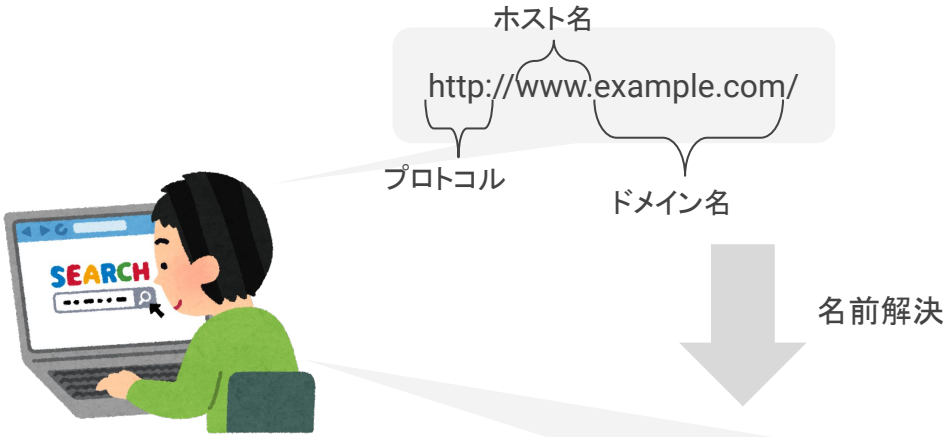
192.168.1.100

WEBページデータを
送ります。

| | | | |
|---|---|---|---|
| HTTPレスポンス<br>データ<br>(HTTPヘッダ含む) | 宛先ポート 65500<br>送信元ポート 80 | 宛先IPアドレス<br>192.168.2.1<br>送信元IPアドレス<br>192.168.1.100 | 宛先MACアドレス<br>00-00-5E-EF-10-00-00-AA<br>送信元MACアドレス<br>00-00-5E-EF-10-00-00-11 |

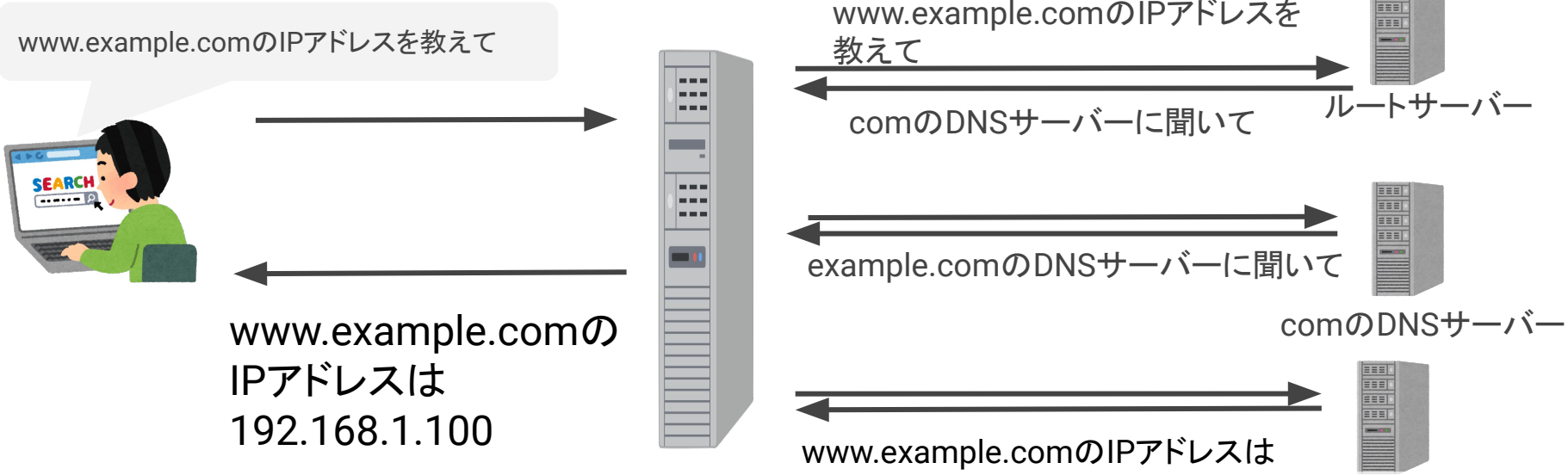
# DNS - ドメイン名とIPアドレス

- DNS :ドメイン名 (名前) とIPアドレスの変換＝名前解決 を行う




www.example.com のIPアドレスは 192.168.1.100 と分かったので,
宛先IPアドレスは 192.168.1.100

# 名前解決の流れ

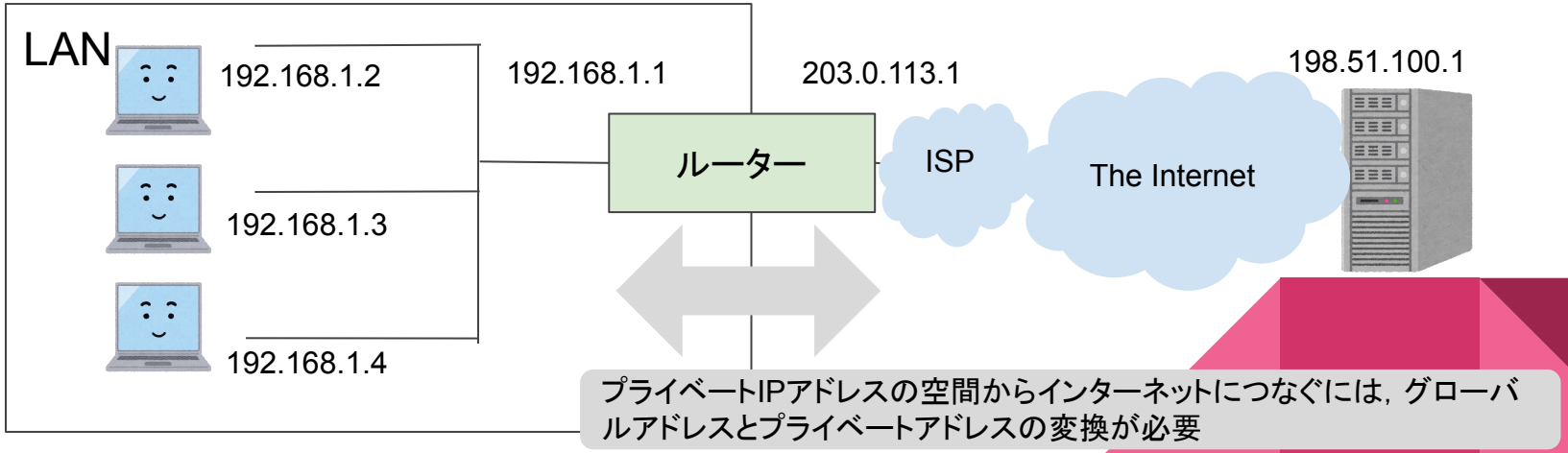


# ルートサーバとTLD

- ルートサーバー
  - A～Mまで13クラスタ存在（ルートヒントファイルにIPアドレスの情報が載っている)
  - 各所在 https://root-servers.org/
- TLD (一例)
  - com, org, net, gov, …… (genericTLD)
  - jp, cn, ru, …… (country code TLD)
  - IANAのTLDリスト https://www.iana.org/domains/root/db

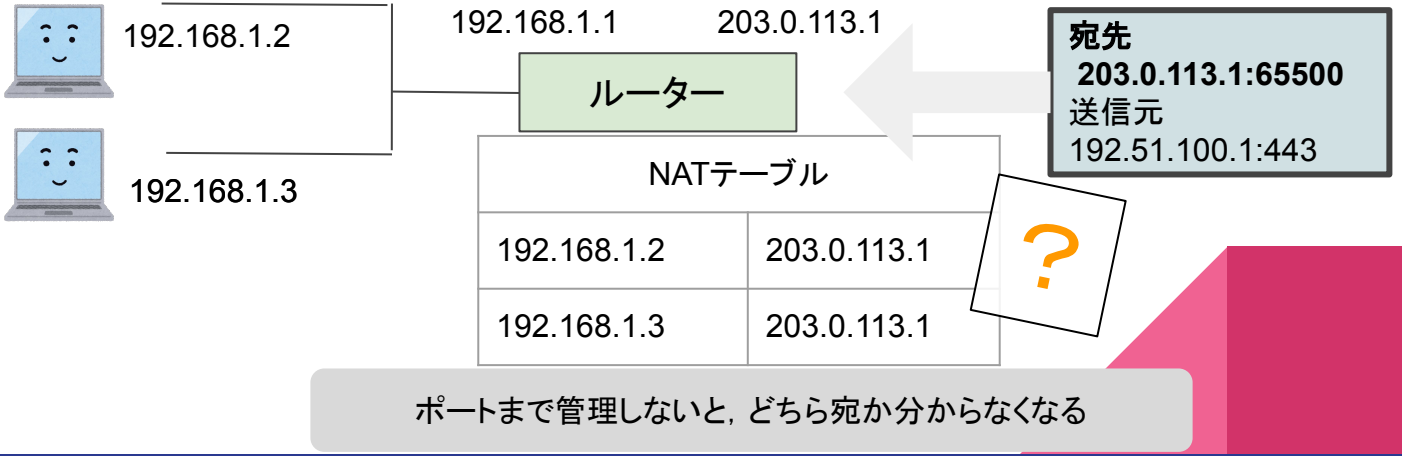
# プライベートアドレスとグローバルアドレス

- グローバルIPアドレス：インターネットで使用できるユニークなIPアドレス
  - 「パブリックIPアドレス」とも呼ばれる（英語ではパブリックが主流）
- プライベートIPアドレス：LANで使用できるIPアドレス（インターネットでは使用できない）



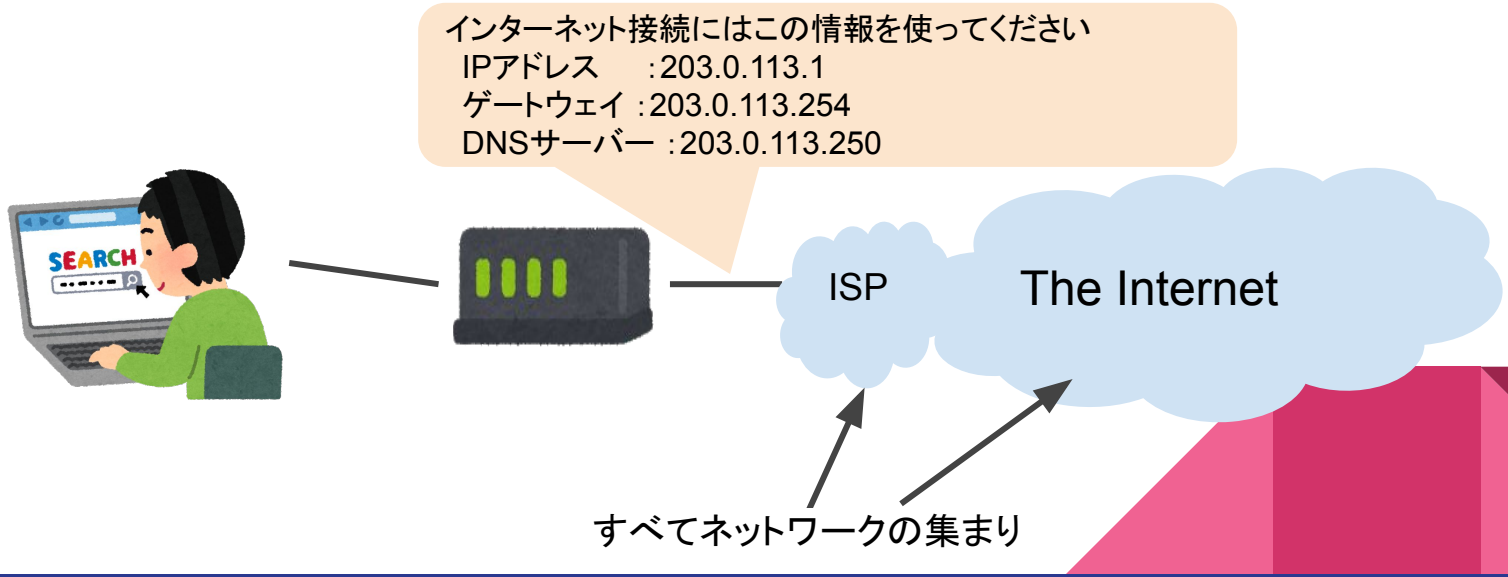
# アドレス変換：NATとNAPT (IPマスカレード)

- NAT (Network Address Translation)　：IPアドレスの１対１ 変換
- NAPT (Network Address Port Translation)　：IPアドレス＋ポート番号の組み合わせを変換
- １つのグローバルアドレスを複数のプライベートアドレスで使う場合，**NAPT** が必要



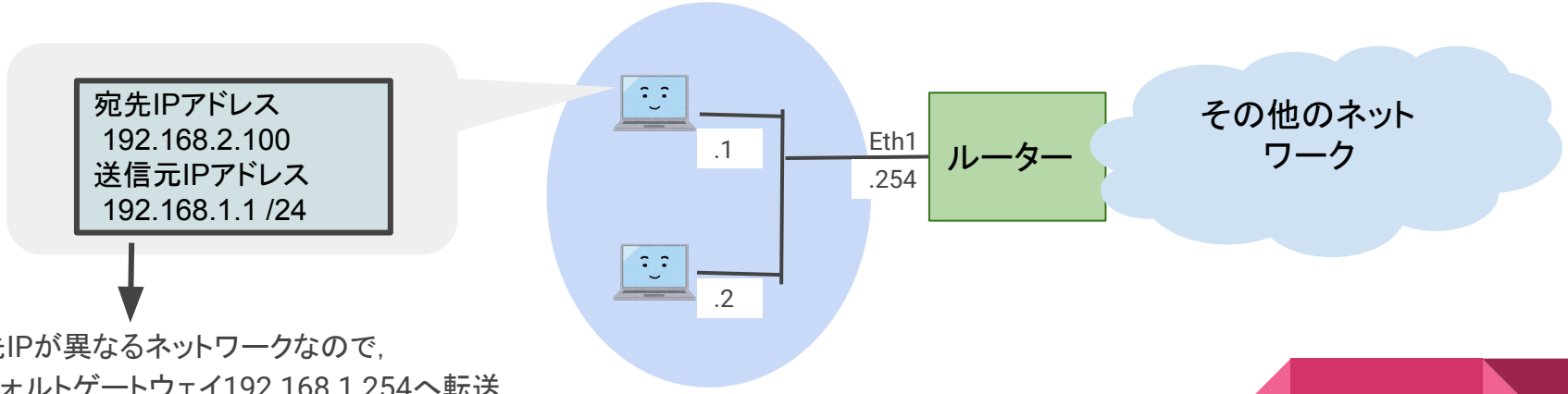
# インターネットへ繋がるとは

- インターネットへ繋がるとは，任意の宛先グローバルIPアドレスへパケットを届けられること
- ISPはインターネットへの接続性を提供する



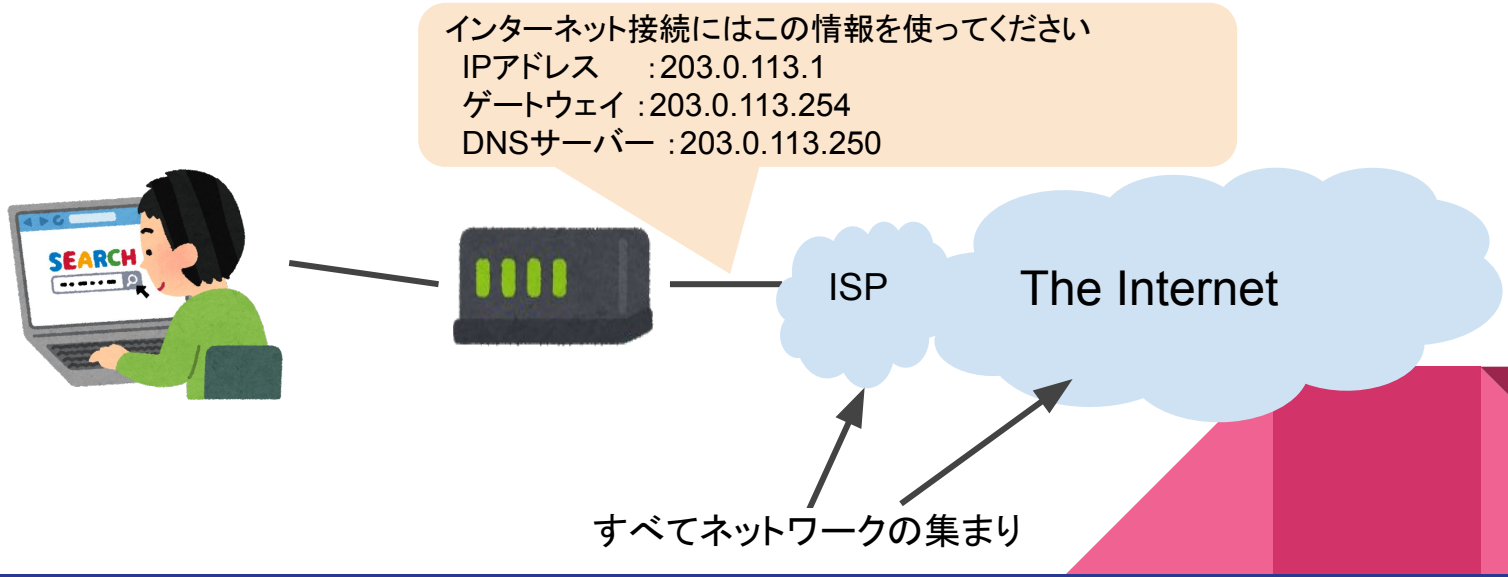
# デフォルトゲートウェイ

- 異なるネットワーク宛てのパケットの場合，PCはデフォルトゲートウェイへ転送する

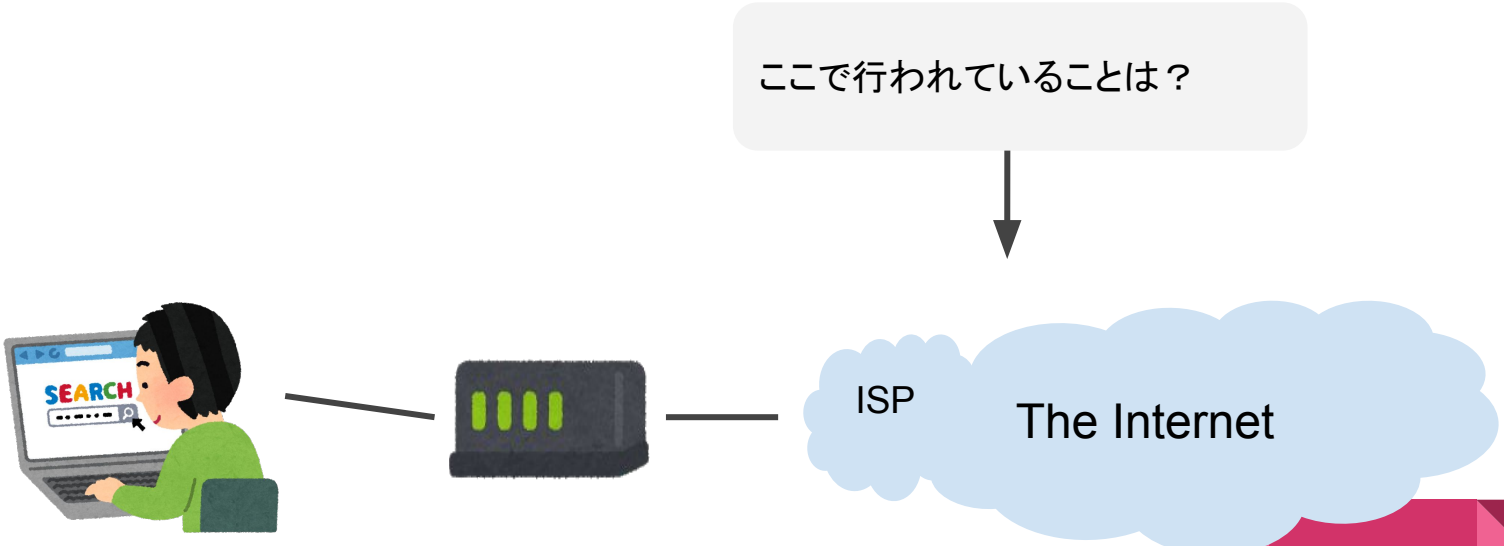


# インターネットへ繋がるとは

- インターネットへ繋がるとは，任意の宛先グローバルIPアドレスへパケットを届けられること
- ISPはインターネットへの接続性を提供する

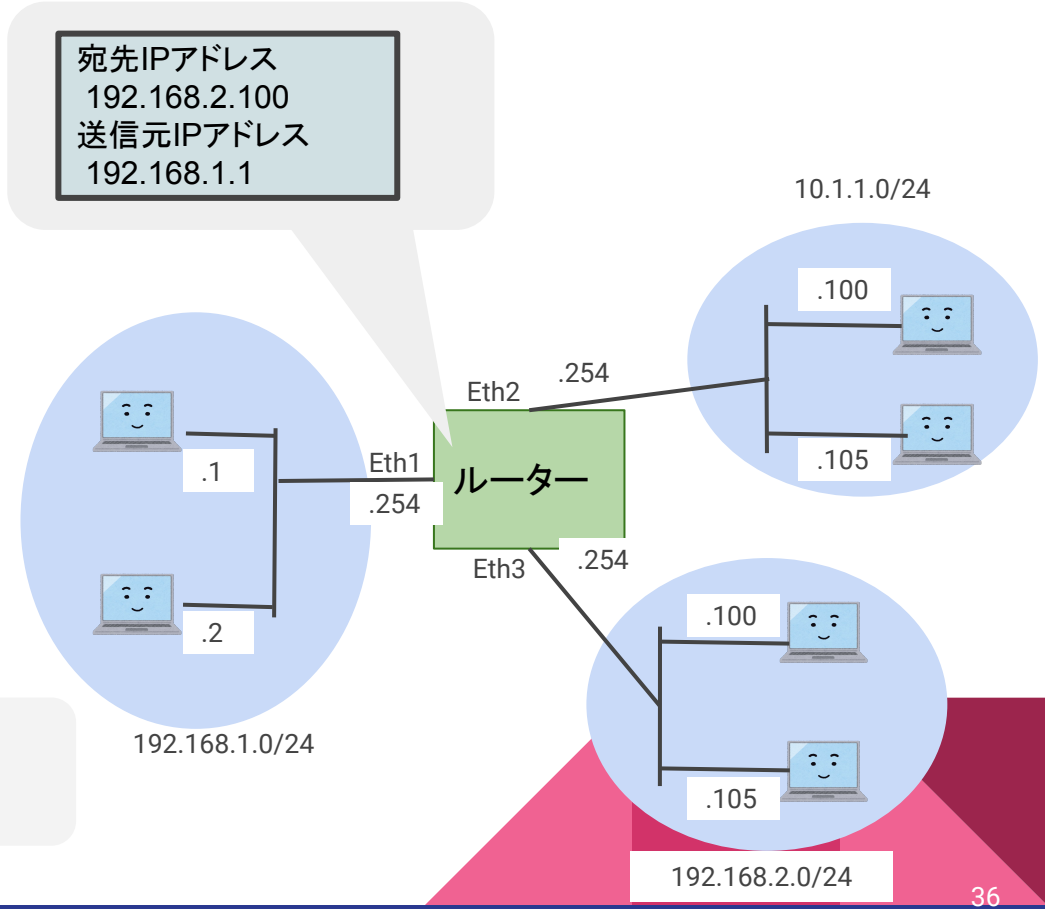


# インターネットへ繋がるとは



# ルーティング

| ルーティングテーブル | | |
|---|---|---|
| 接続ネットワーク | インターフェース | ネクストホップ |
| 192.168.1.0/24 | Eth1 | Directly connected |
| 192.168.2.0/24 | Eth3 | Directly connected |
| 10.1.1.0/24 | Eth2 | Directly connected |



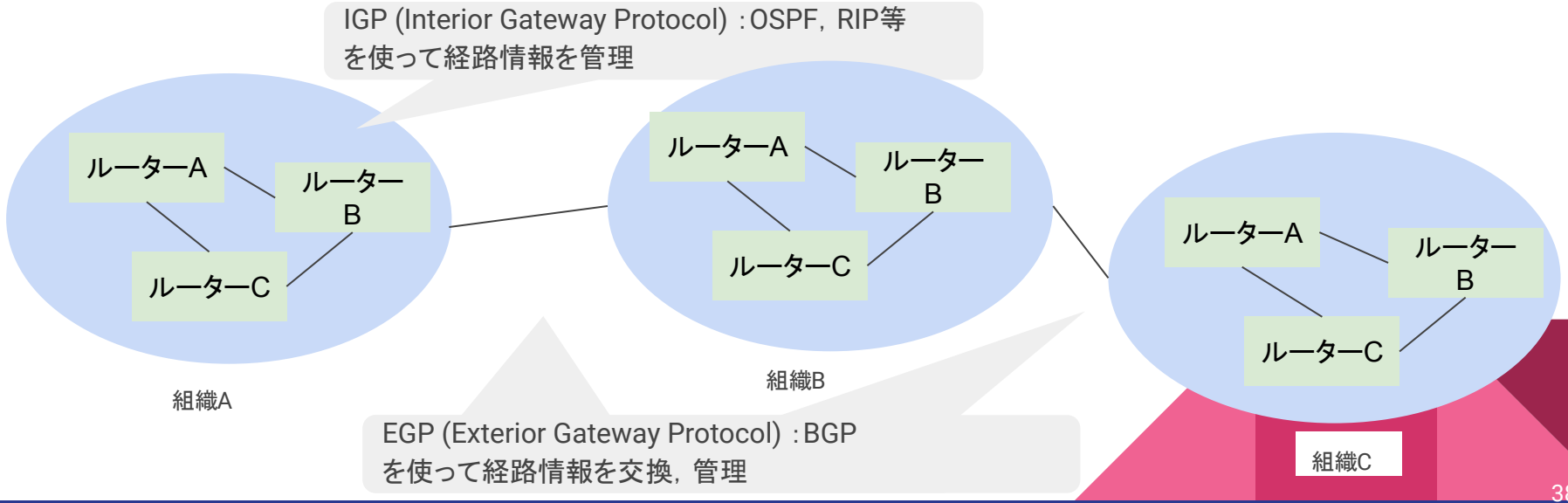

ルーターのルーティングテーブルにもとづいて，
パケットはEth3から192.168.2.0/24へ転送する

# 静的ルーティングと動的ルーティング

- 静的ルーティング（スタティックルーティング）
  - 「この送信元でこの宛先なら，ネクストホップは　XXX 」と固定的に設定
- 動的ルーティング（ダイナミックルーティング）
  - 「このルーターはこのネットワークに繋がっている」という情報を **ルーティングプロトコル** で動的に集めて最適な経路を動的生成（ RIP，OSPF，IS-IS，BGP等）

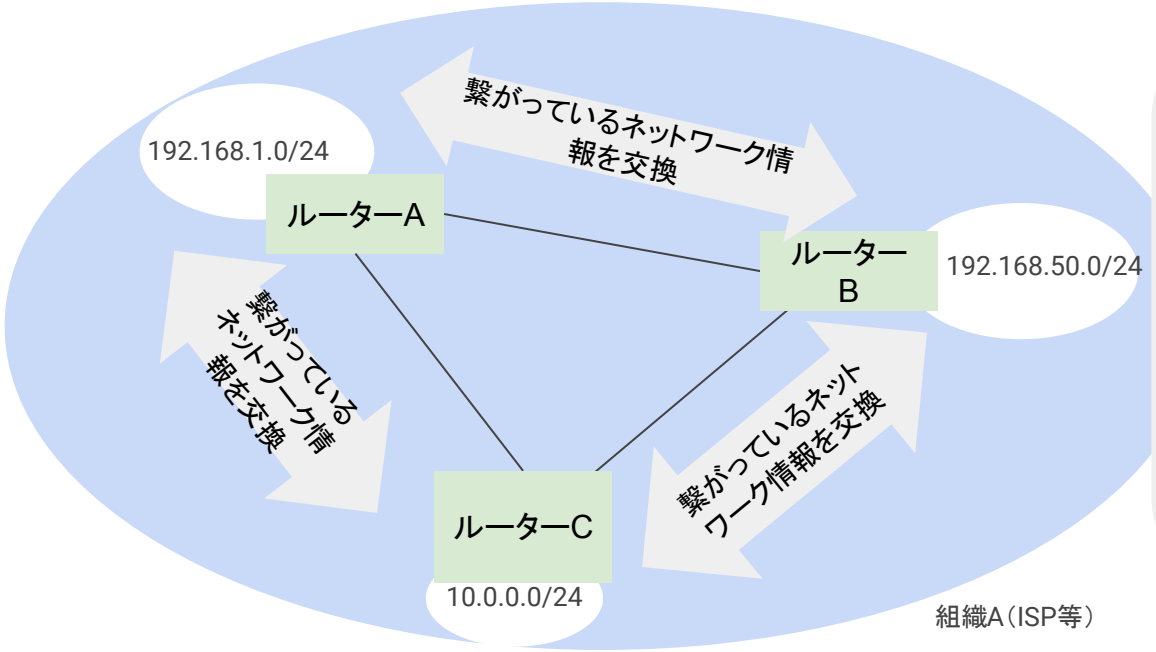
# インターネットとルーティング

- インターネットはネットワークの集合
  - → 異なるネットワークへパケットを転送するために ルーティングテーブル を使う
  - → ルーティングテーブル を作るために **ルーティングプロトコル** を使う



# ルーティングプロトコル：IGP

- 組織内のルーティングプロトコルは，組織内で決める
  - → 組織によって何のルーティングプロトコルを使っているかは様々




OSPFの場合
　ルーター同士 自分が繋がっているネットワーク情報を交換し，ネットワーク全体のマップを作成

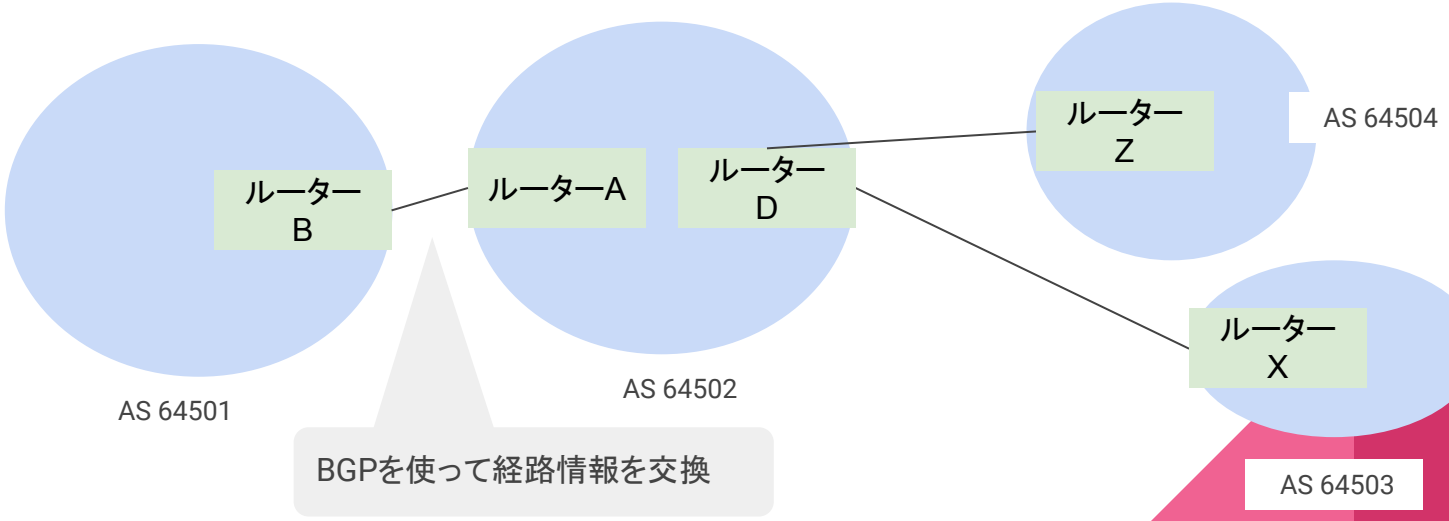
↓

ルーティングテーブルを作成

| ルータAのルーティングテーブル（簡略） |
| --- |
| 192.168.50.0/24宛ならルータBへ転送 |
| 10.0.0.0/24宛ならルータCへ転送 |

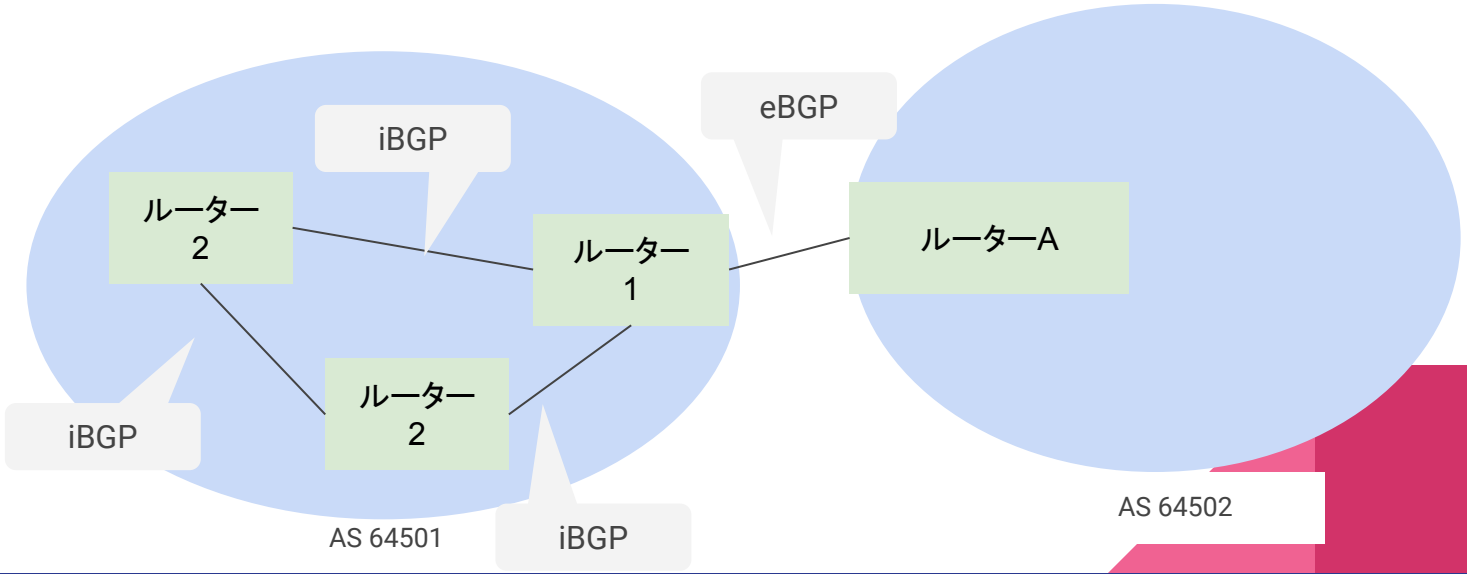
# ルーティングプロトコル：BGP

- 組織間のルーティングプロトコルは組織間で共通でなければならない
  - →現在インターネットでは **BGP (Border Gateway Protocol)** が使われている
- BGPでは **AS** (Autonomous system) を単位として経路情報を扱う
- インターネット通信をするとき，自分も宛先もどこからしらの ASに所属している

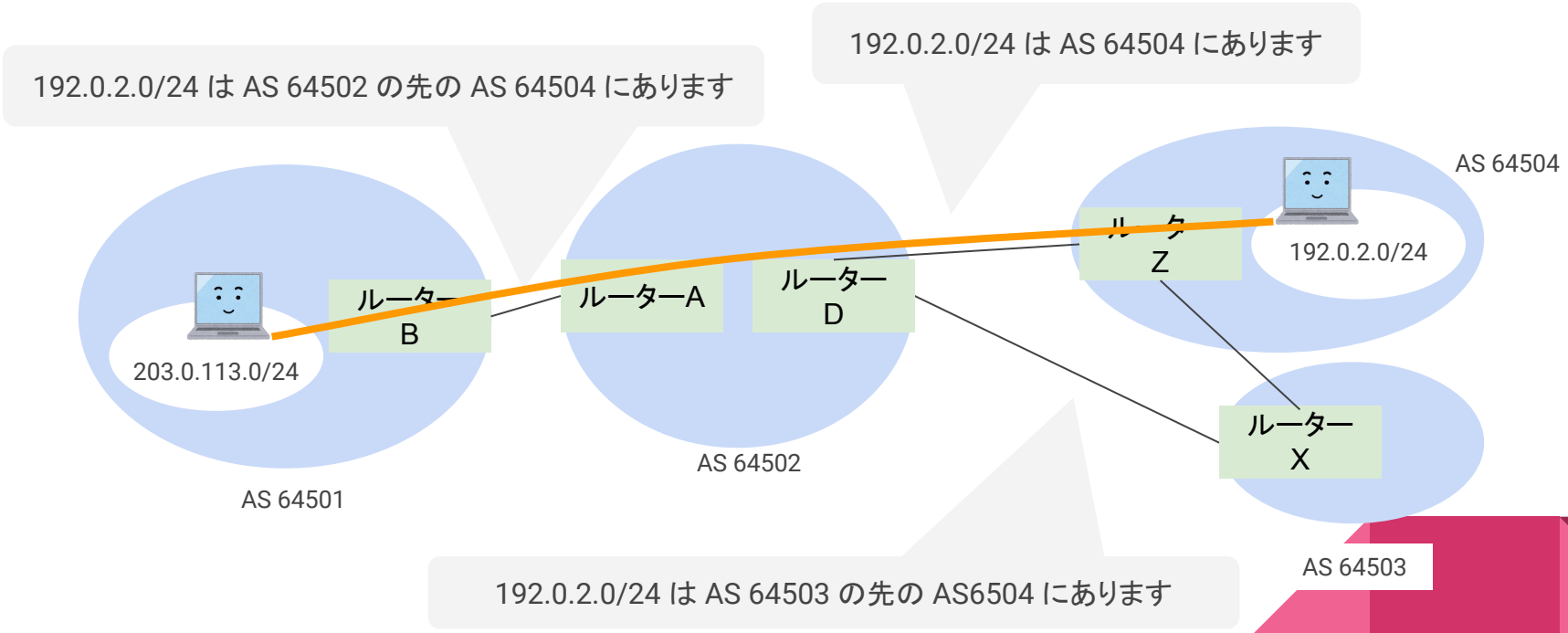


# BGP:ピアリング

- **ピア** をBGPルーター同士で確立
- 最初にすべての持っている経路情報を伝えた後は，TCPコネクション（179番）を張りっぱなし。
- もし最初に渡した経路情報から変化があったら，その差分だけを送信（ Update）

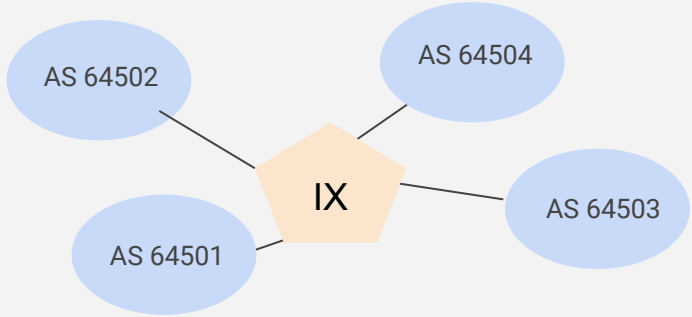


# BGP:ピアリング



# “インターネットへの経路” を手に入れるために

パブリックピアリング (Internet Exchange を利用)

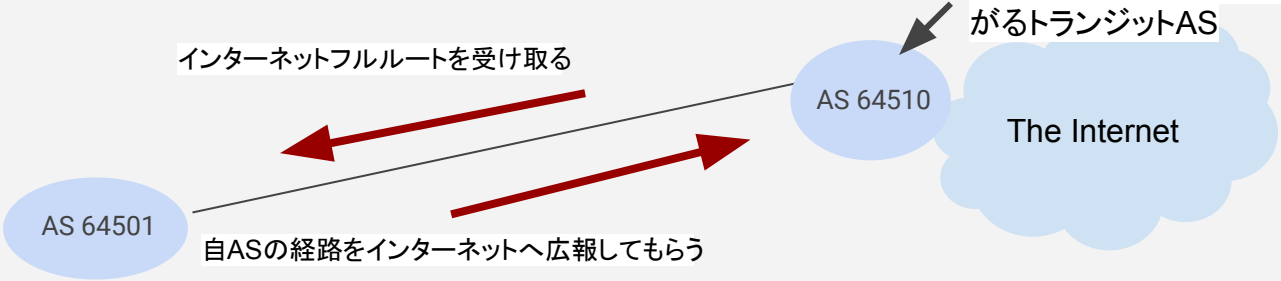



プライベートピアリング ( PNI)

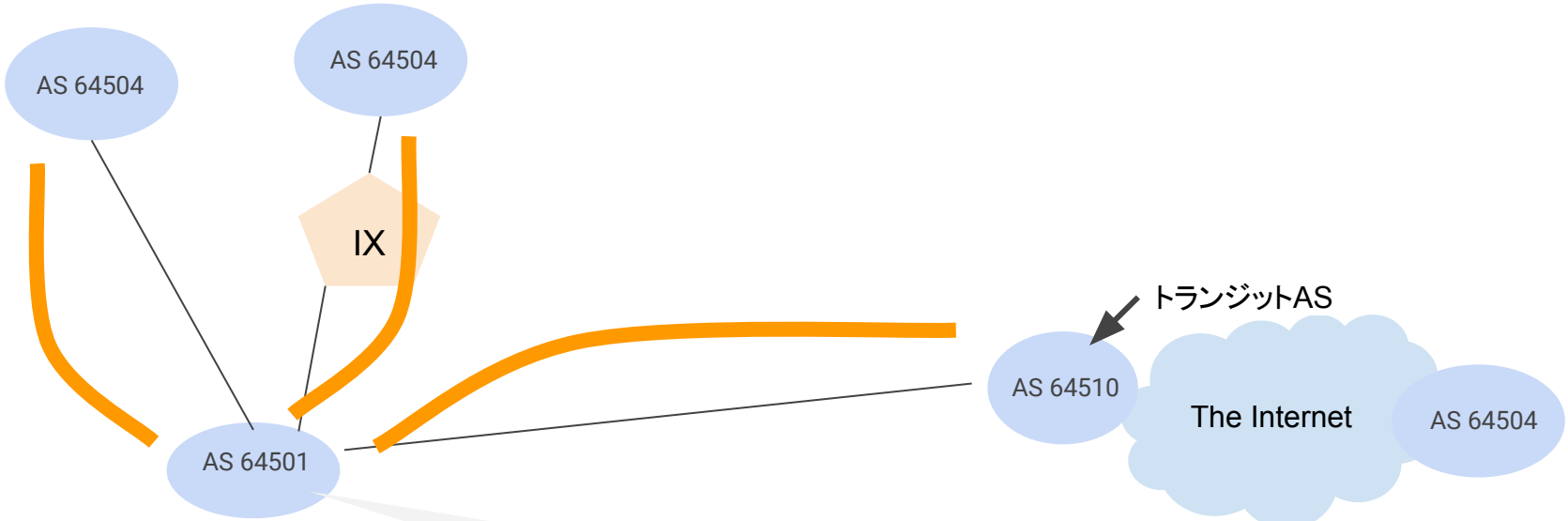

※PNI : Private Network Interconnect

トランジット

インターネット上のすべてのISP (ネットワーク) に繋がるトランジットAS



# どの経路が最適か：ベストパスの選択




AS64503のネットワークへ到達するには
・Private Peering でもらった経路
・Public Peeringでもらった経路
・トランジットでもらった経路
が選択肢にあるけど・・・

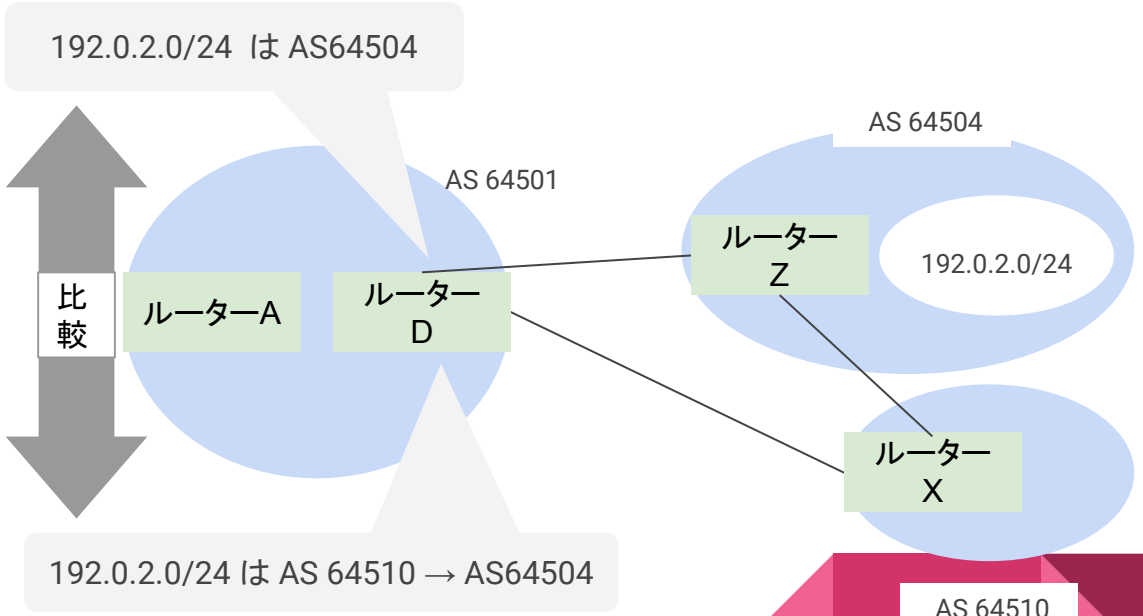
# どの経路が最適か：ベストパスの選択

- BGPで扱う経路情報には様々な付加情報（パス属性）が含まれる
  - 同じ宛先への経路情報でも，パス属性が異なる
  - →パス属性を比較して，どれが最適経路かを判断

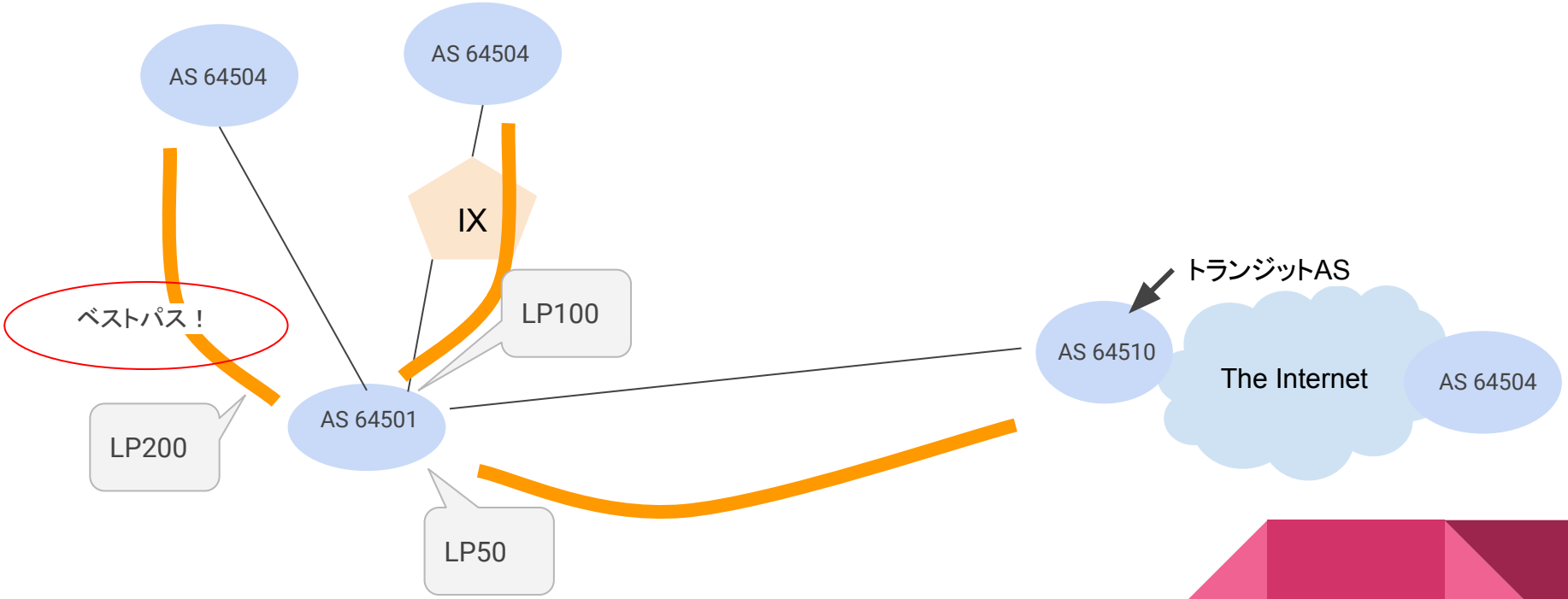
# BGP：ベストパスセレクション

どれが最適な経路（ベストパス）かを決める

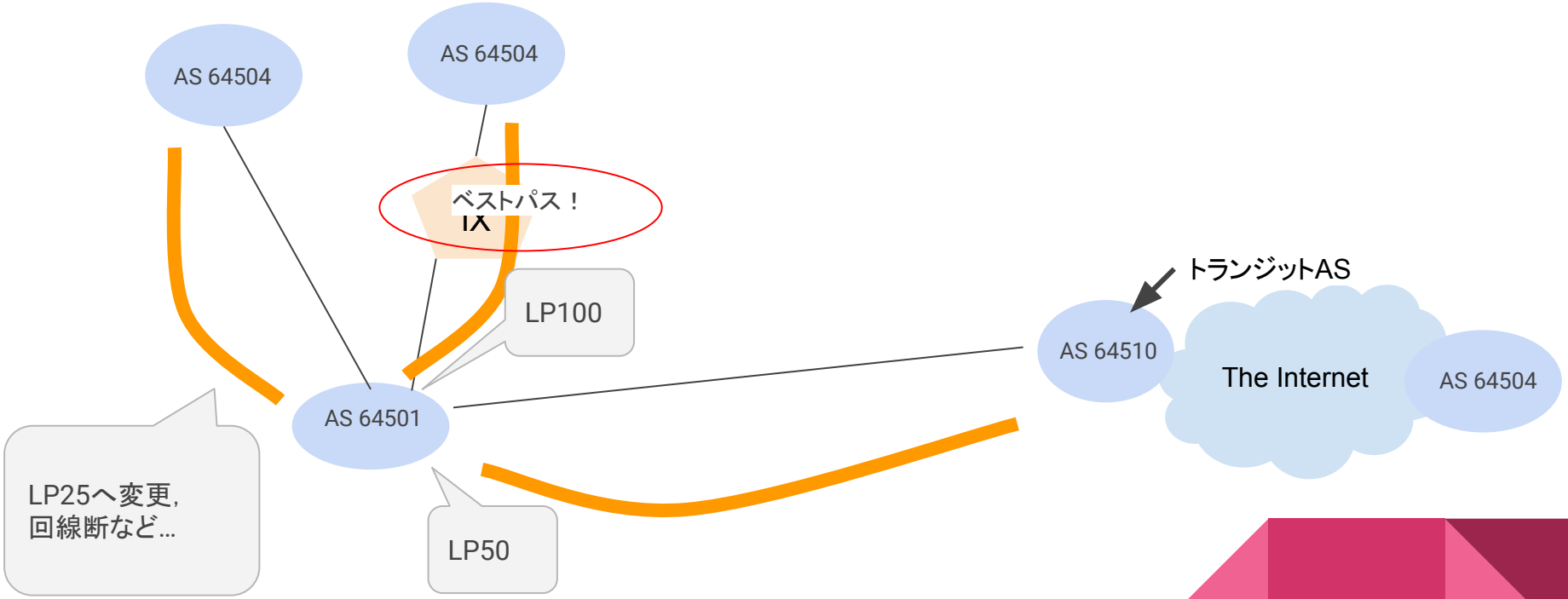
| 判断順序 | BGPベストパス選択アルゴリズム |
|---|---|
| 2 | LOCAL_PREF属性の値が最も大きい |
| 3 | LOCALで生成された経路 |
| 4 | ASパス長が最短 |
| 5 | MED属性の値が最も小さい |
| … | … |



# どの経路が最適か：ベストパスの選択



# どの経路が最適か：ベストパスの選択



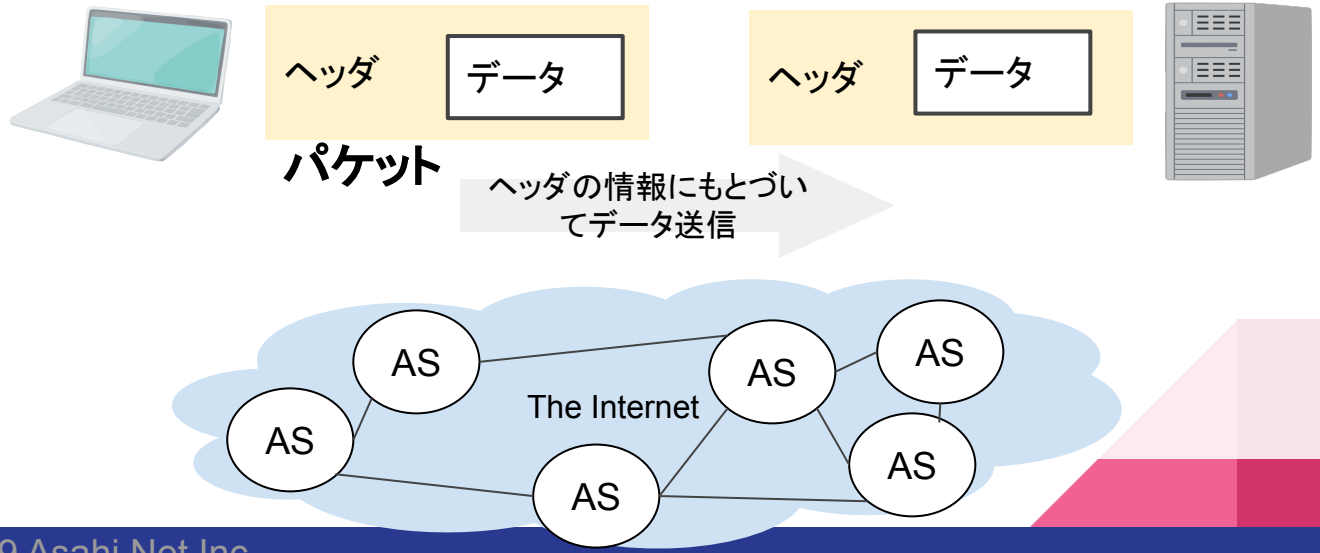
# どの経路が最適か：ベストパスの選択

- BGPで扱う経路情報には様々な付加情報（パス属性）が含まれる
  - 同じ宛先への経路情報でも，パス属性が異なる
  - →パス属性を比較して，どれが最適経路かを判断

・ピアリングを行うことで，同じ宛先について複数の経路情報を持っておく（冗長性）
・各回線の状態に応じてLP，MEDなどのパス属性を変更し，ベストパスをコントロール（トラフィックコントロール）

# まとめ

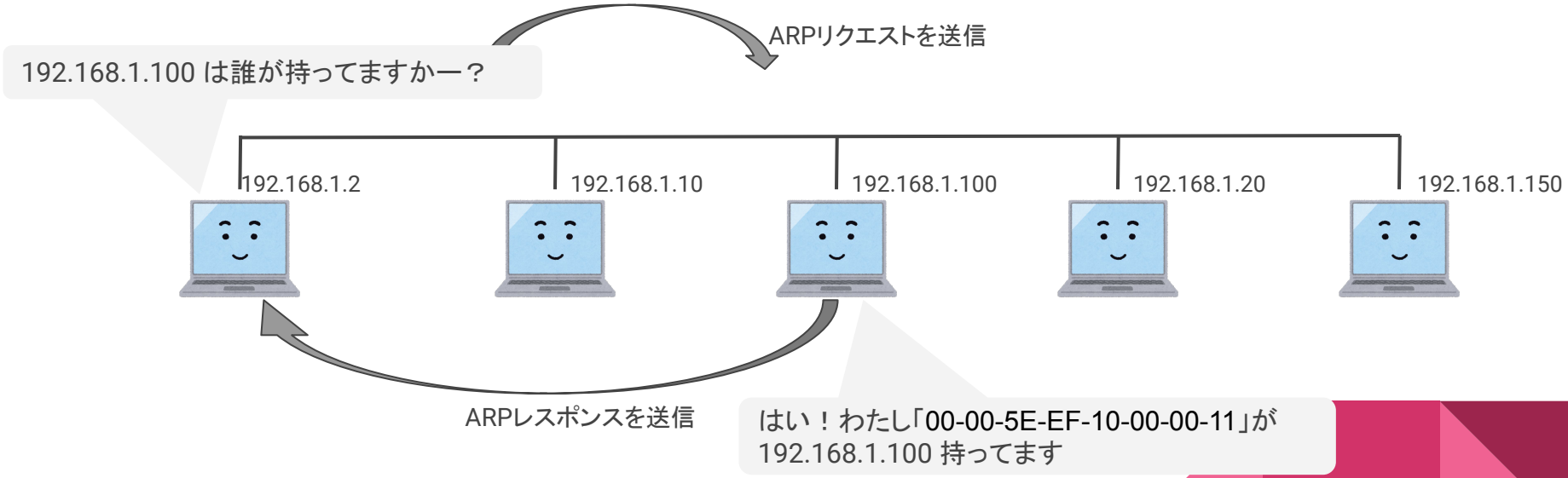
- インターネット通信では，データはパケットという単位でやり取りされる
- ヘッダには，各プロトコルでやり取りするのに必要な情報が含まれている
- インターネットは BGP というルーティングプロトコルで繋がっている



# APPENDIX

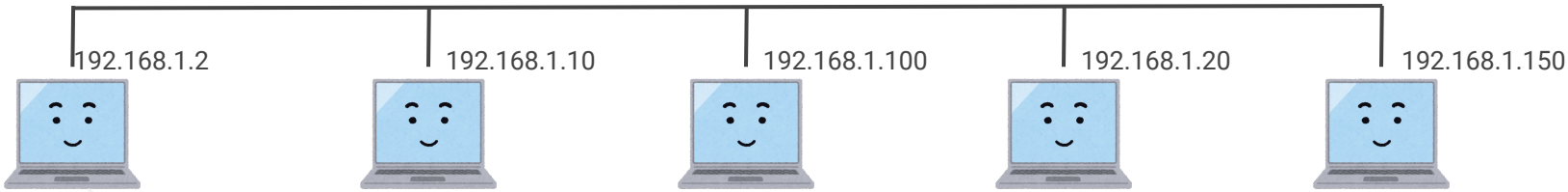
# ARP - IPアドレスとMACアドレスの橋渡し

- ARPによってIPアドレスから宛先MACアドレスを特定する



# ARP - IPアドレスとMACアドレスの橋渡し

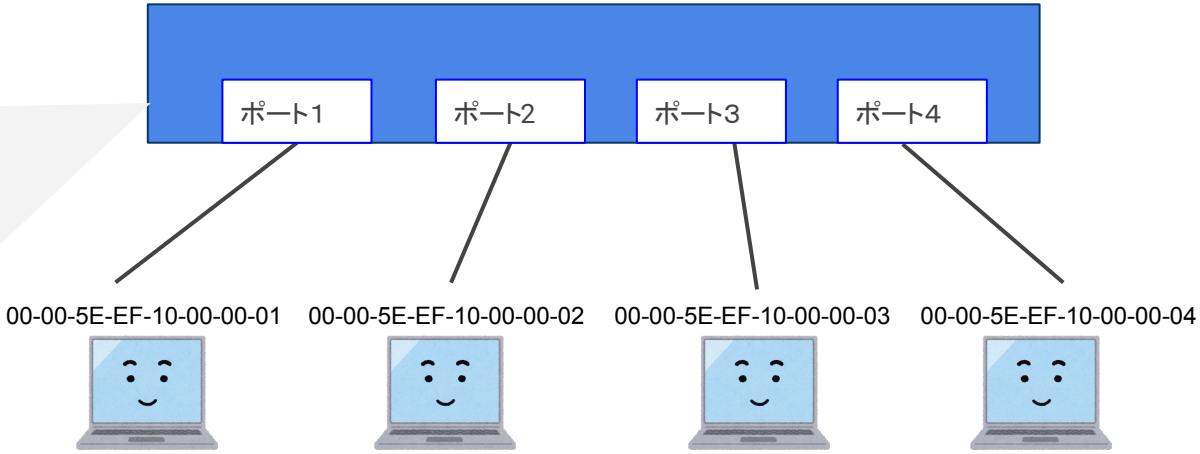
- ARPテーブル：IPアドレスとMACアドレスの対応リスト




| ARPテーブル | |
|---|---|
| IPアドレス | MACアドレス |
| 192.168.1.100 | 00-00-5E-EF-10-00-00-11 |

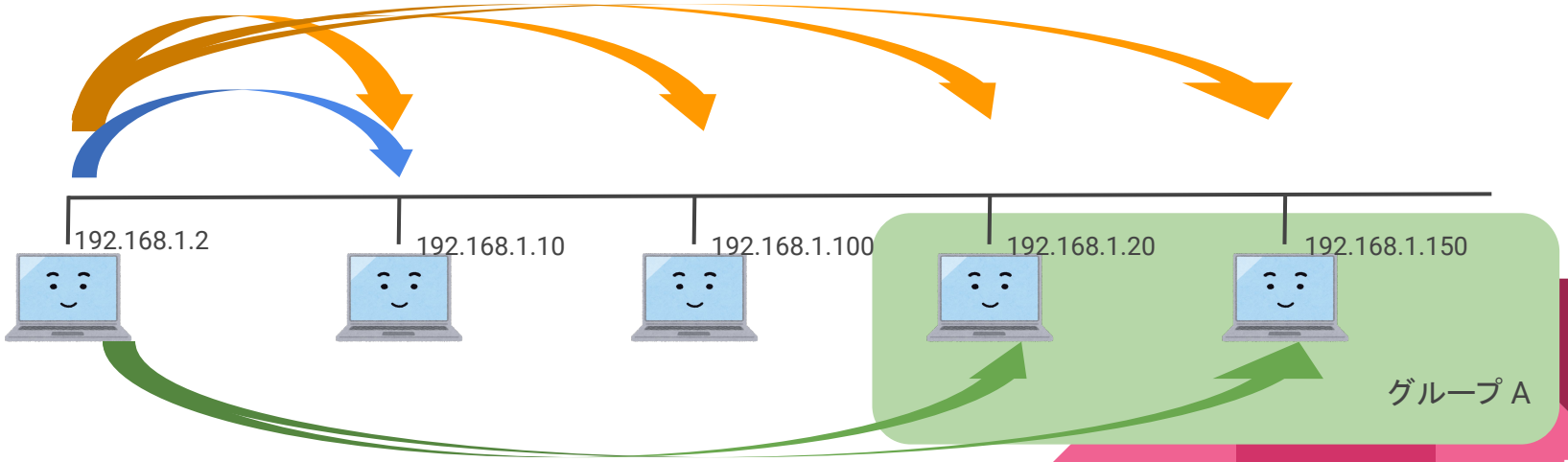
# スイッチ

| MACアドレステーブル | |
|---|---|
| ポート | 接続されているMACアドレス |
| 1 | 00-00-5E-EF-10-00-00-01 |
| 2 | 00-00-5E-EF-10-00-00-02 |
| 3 | 00-00-5E-EF-10-00-00-03 |
| 4 | 00-00-5E-EF-10-00-00-04 |



# ユニキャスト・ブロードキャスト・マルチキャスト

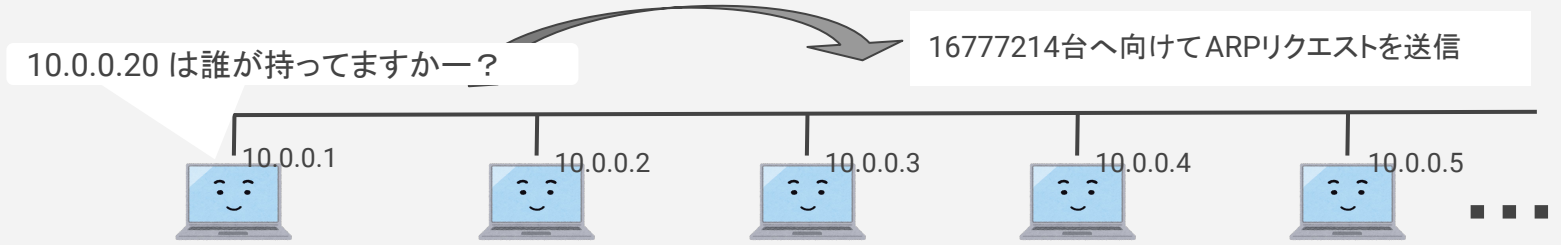
- [ユニキャスト]　　1：1 の通信。特定が持つ固有のアドレスを使用し，特定機器宛となる
- [ブロードキャスト] 1：全体 の通信。ブロードキャスト用のアドレスを使うことで全体宛送信となる
- [マルチキャスト]　1：N の通信。マルチキャスト用のアドレスを使うことで，特定グループ宛送信となる



# サブネット化

- ARPリクエスト→スイッチに接続している機器全体に送信
- クラスフルの考え方では，IPアドレスのビットパターンで同一ネットワーク（ブロードキャストが届く範囲）を区切っていた

クラスフルの考え方だと：10.0.0.0 ～ 10.255.255.255 が同一ネットワーク。

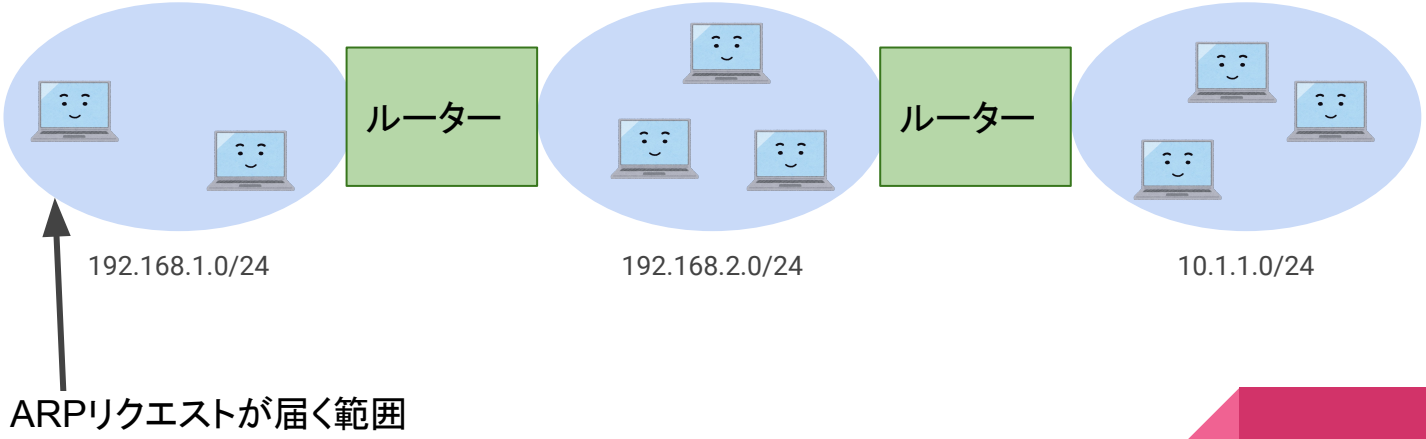



大量の機器にARPリクエストを送信するのは非効率

**サブネット化** (ネットワークを分割) してブロードキャストの範囲を区切る

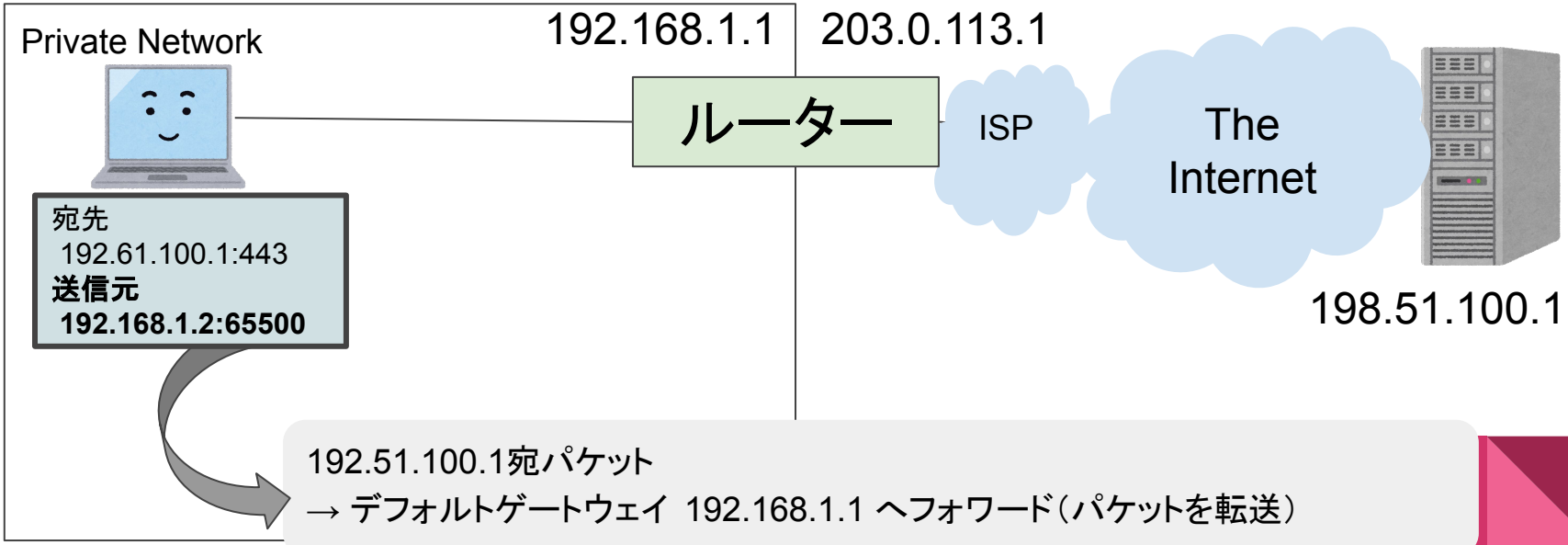
# ルーター

- ルーターによって異なるネットワーク間の通信が可能になる

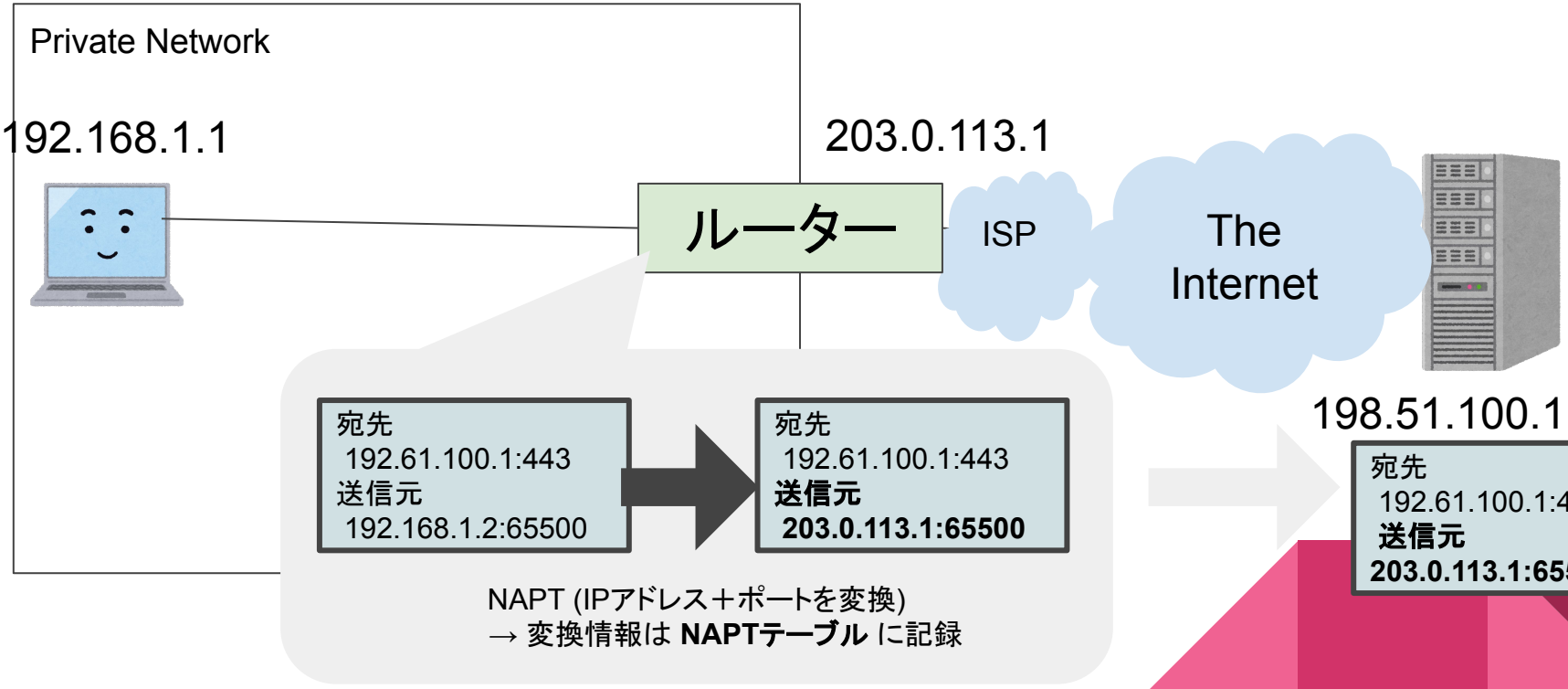


# NAPTの流れ - アドレスとポート番号の変換

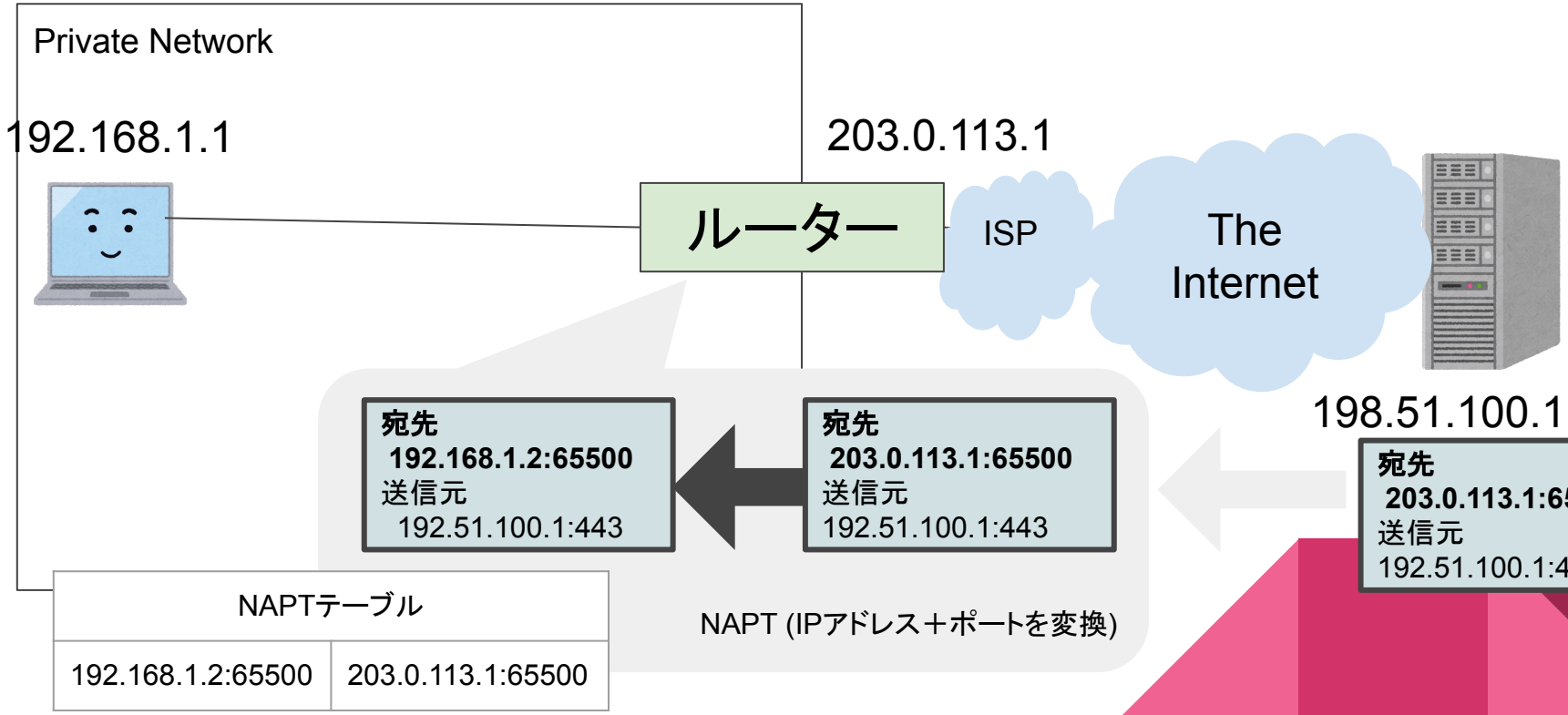
- IPアドレス（＆ポート）を変換するための機能



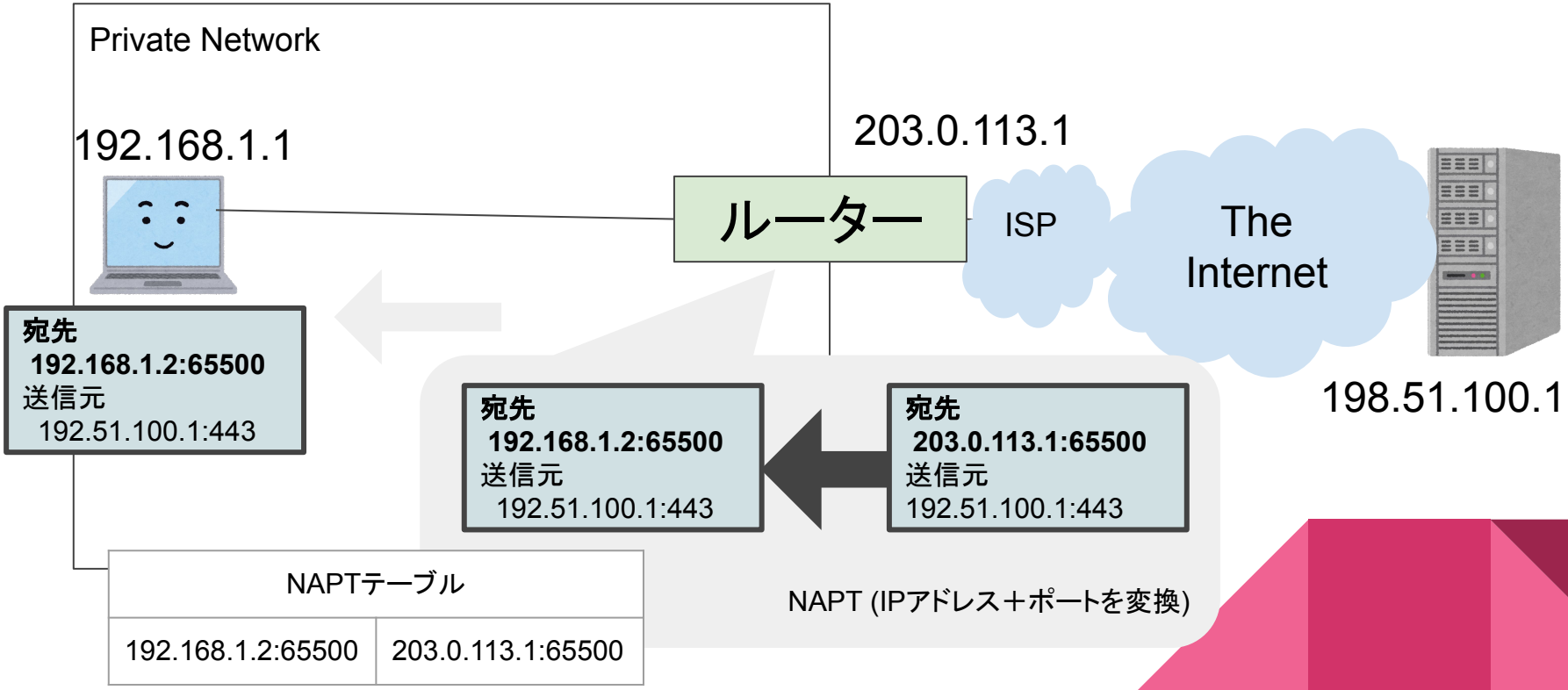
# NAPTの流れ - アドレスとポート番号の変換



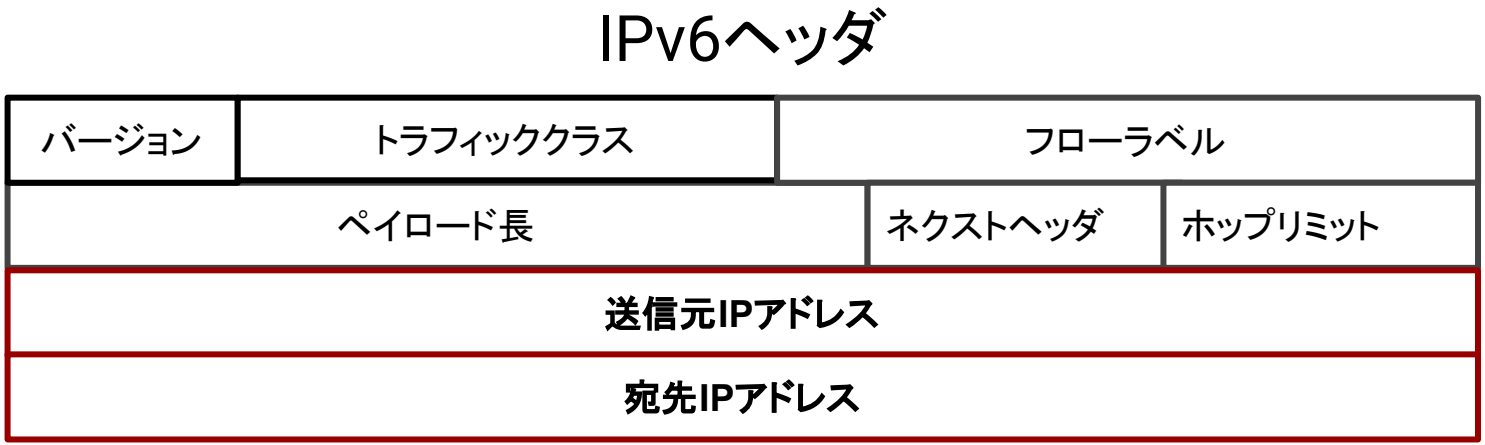
# NAPTの流れ - アドレスとポート番号の変換



# NAPTの流れ - アドレスとポート番号の変換



# IPv6ヘッダ




- IPv6アドレス：128bit、16ビットずつ8つに「：」で区切り、の16進数で表記
  - 2001:0db8::1
- IPv4アドレス：32bit、8ビットずつ4つに「．」で区切り、10進数で表記
  - 203.0.113.1

IPv6では，「リンクローカル」「ユニークローカル」「グローバル」という **スコープ**を用いて通信範囲を制御