ネットワーク レイヤ
- IP (Internet Protocol) -

(0) 概要
(1) IP (Internet Protocol)
(2) Addressing
(3) ARP
(4) Routing
(5) Address Discovery (e.g., DHCP)
(6) ICMP

パケットの転送方法
- 輸送システムに例えると -

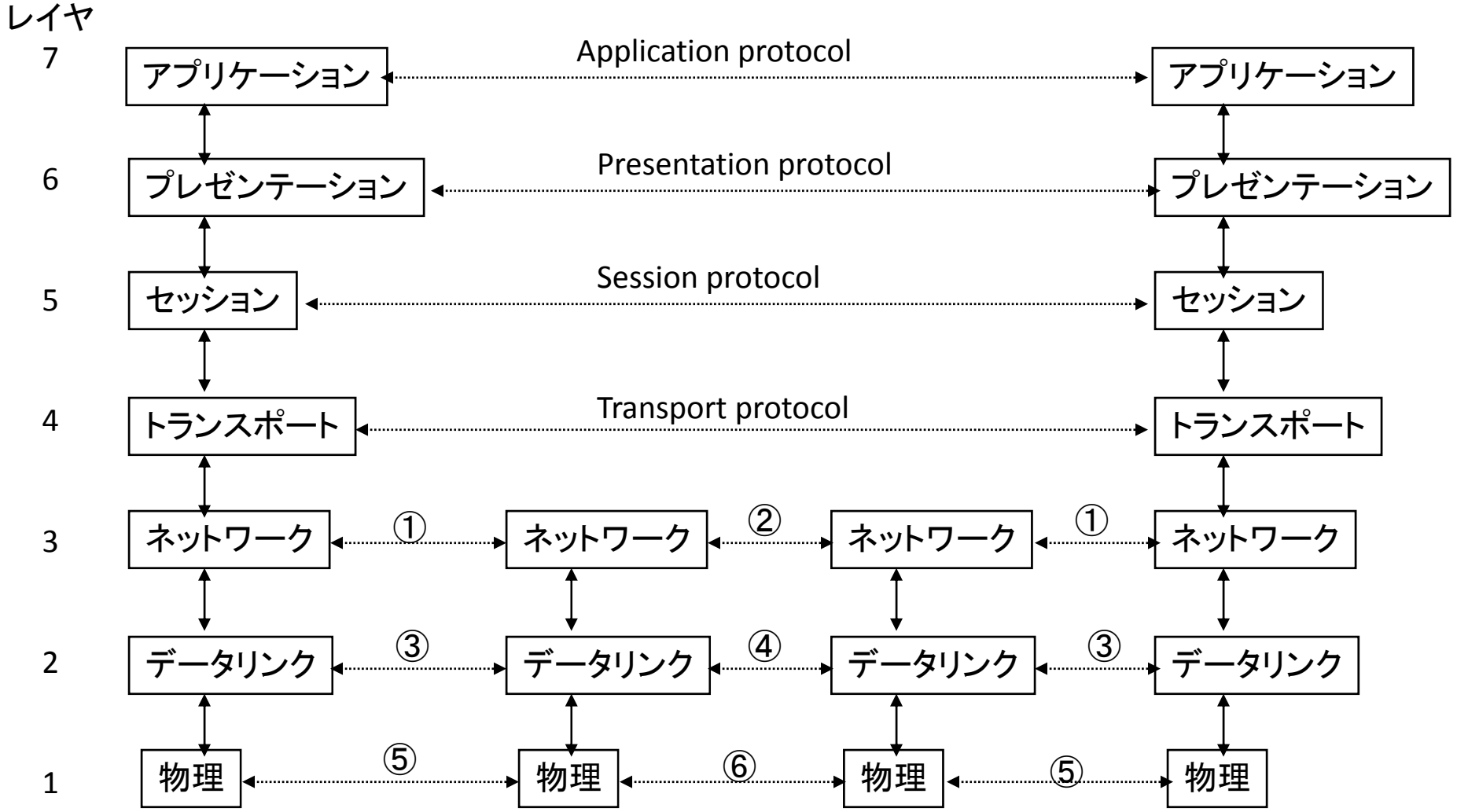
L4 : アプリケーションデータ ＝＝ 輸送したいもの
L3 : IPパケット ＝＝ 人
L2 : データリンクフレーム ＝＝車、電車、飛行機
L1 : 物理層 ＝＝ 道路、線路、空/滑走路

必要になるもの ;
(1) レイヤ間でのインターフェース
e.g., 電車への乗り方、道路の走り方
(2) 乗換場所、交差点
(3) レイヤの統一規格

インターネットの父 Dr.Robert Kahn氏の話

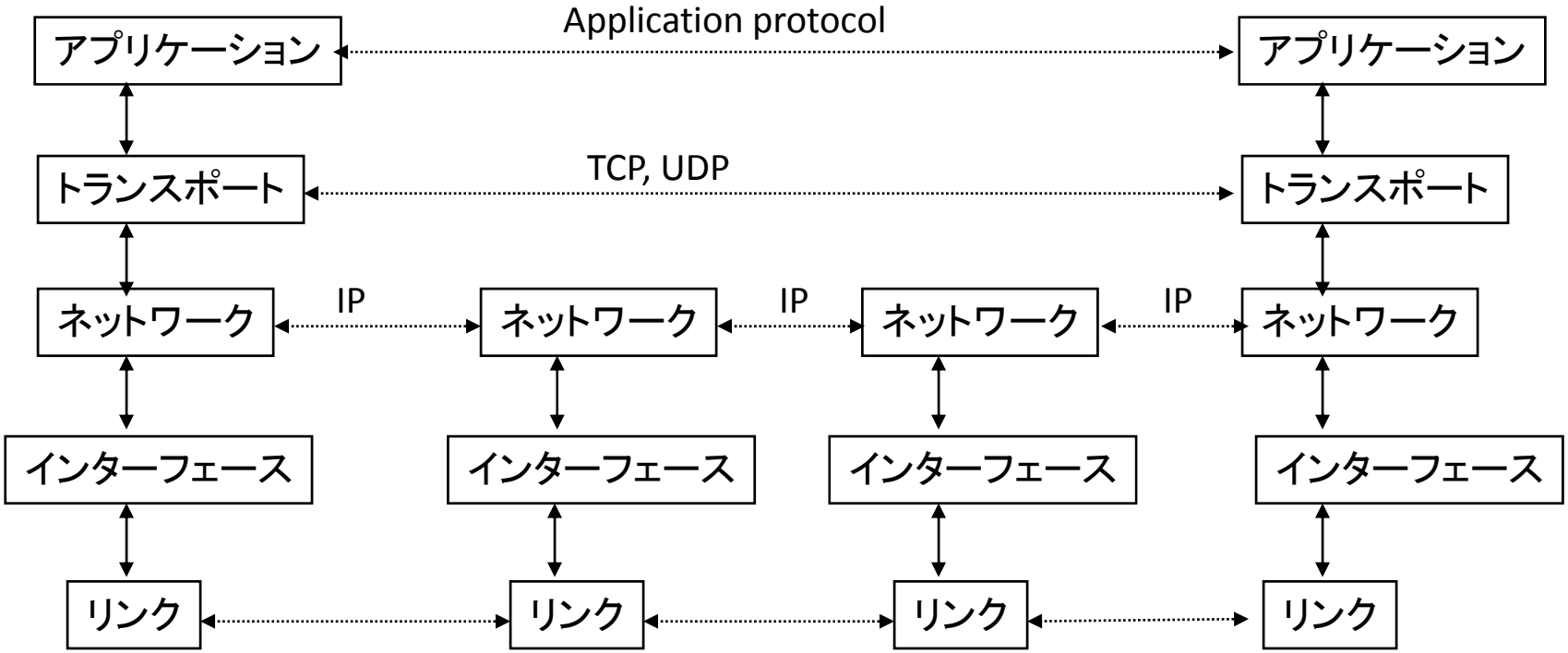
- Packet Networking 技術に関する研究がたくさんあった。
  - でも、誰も、実際に動かそうとしなかった。
  - だから、Vinton Cerf (現 MCI上級副社長)に実装させた。
  - でも、、、TCPの Virtual Connectionによる実装は、失敗だったかなぁ。。。。
- インターネットは、ロジカルなアーキテクチャ。実装形態としての TCP/IP と ネットワーク機器
  - 複数のメディアを利用可能にする“環境”(アーキテクチャ)の提供がいろいろな意味で “鍵” となる。

OSIの7層モデル



① : Network Layer Host-Router Protocol (e.g., ES-IS)
②, ④,⑥ : サブネットプロトコル (e.g., IS-IS,NNI,)
③ : Data link Layer Host-Router Protocol (e.g, UNI)
⑤ : Physical Layer Host-Router Protocol (e.g., UNI)

TCP/IPの5層モデル



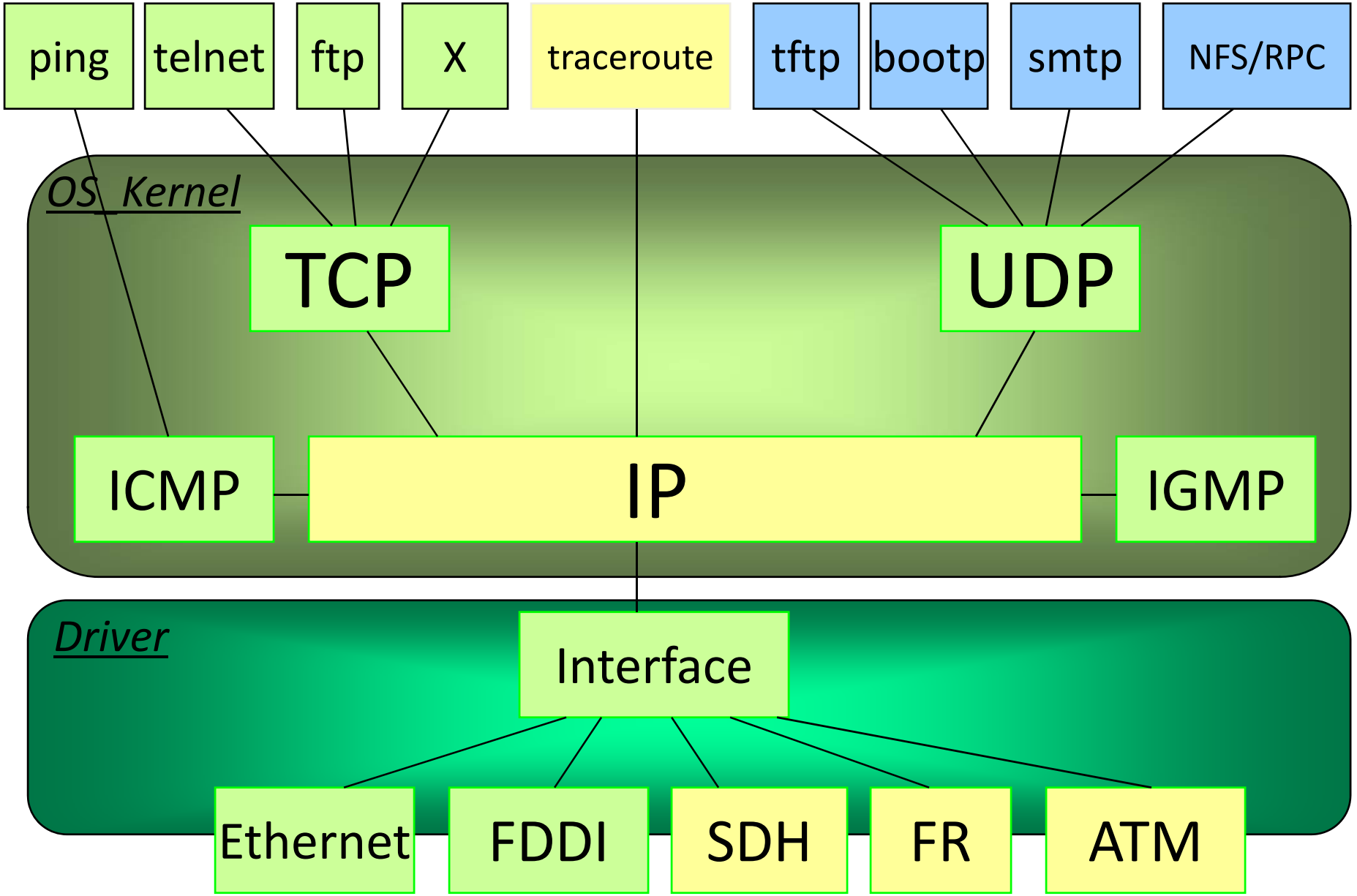
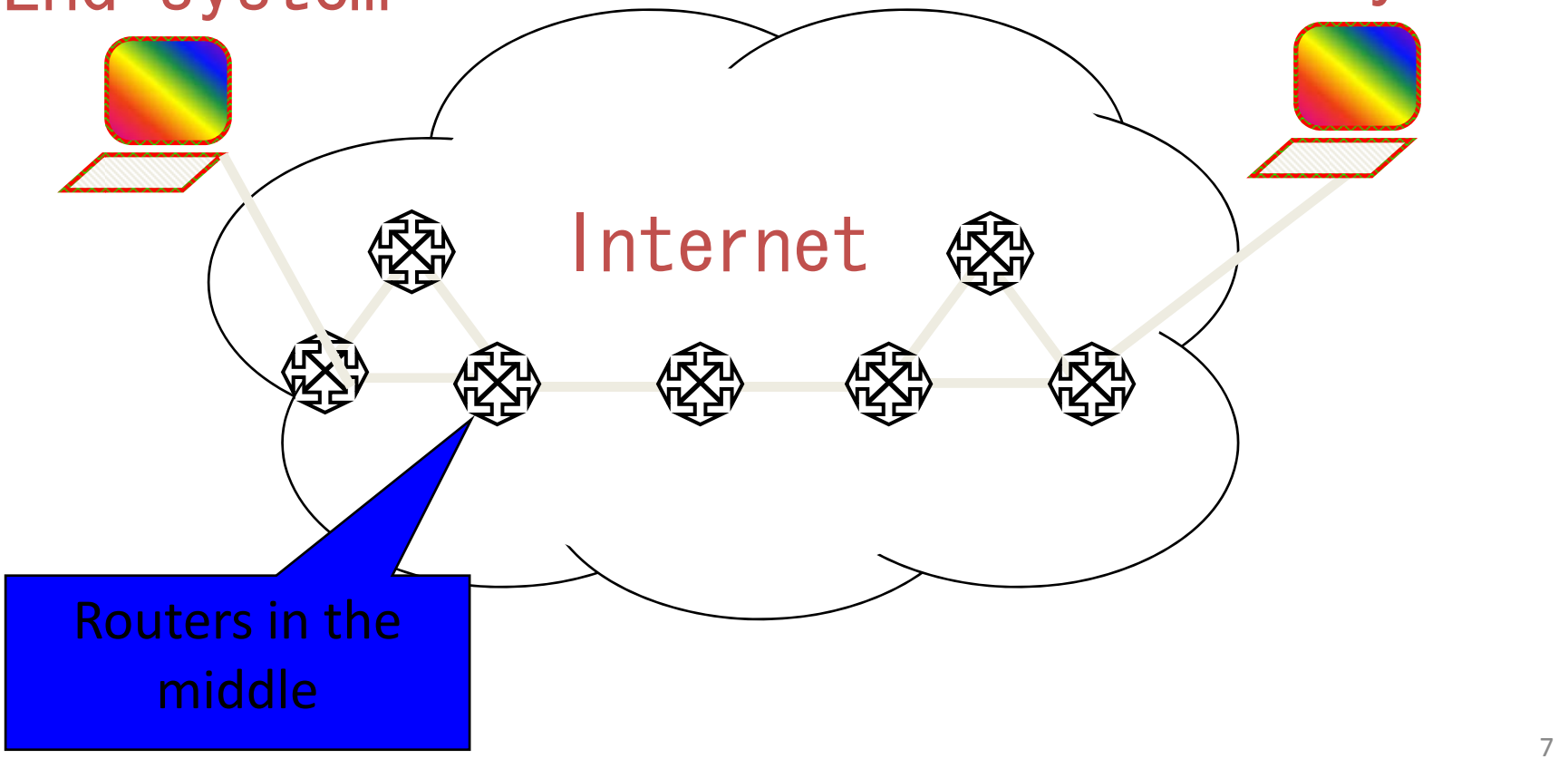


図1-20. インターネットシステムにおけるソフトウェア構造の例

Internet “end-to-end model”

透明な情報“流通”の重要性
転送/伝送 ➔ 流通



IPシステム(Internet)って？

1. オープンシステム
   - どこの会社のものでも 使えます。
2. グローバルシステム
   - どこからでも 管理制御 できます。
3. 自由な情報の流通
   - いろいろな目的に利用できます。
4. (デジタル)情報には “Semantic” がない
   - 専用システムの呪縛が解けます。
5. データリンクを選ばず
   - リンクがあれば、つなげられます !!

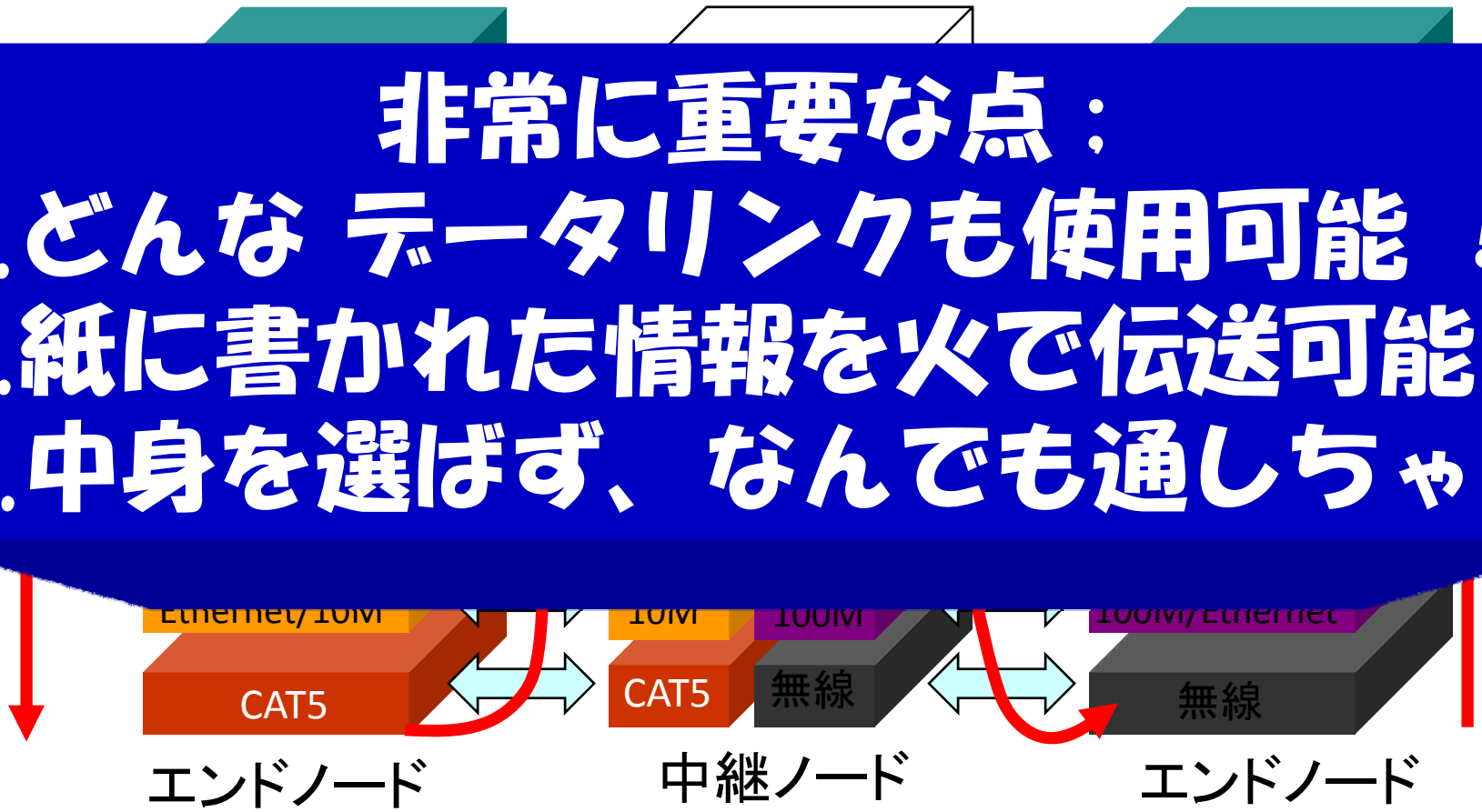
インターネットを用いた
デジタル情報の伝送

非常に重要な点：

1.どんな データリンクも使用可能 !!!

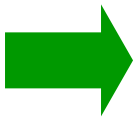
2.紙に書かれた情報を火で伝送可能 !!

3.中身を選ばず、なんでも通しちゃう



例えば。。。。

- “既設“ xx を使って
  - ADSL ＝ 電話線
  - PLC(Power Line Communication) ＝ 電力線
    - Live E! での実例
  - 無線LAN ＝ 非免許対象周波数
    - マンションの共用ルーム防犯カメラの実例



システムの設計・構築に必要なコストの削減効果

パケット

- **pack･et**　*n.*　（出典：EXCEED英和辞典）
  - 小包；（小さい）束；
  - 【コンピュータ】 パケット (データの転送の単位)；

IPパケットの転送
- 東大からコロンビア大へ -

運びたいもの： 映画

媒体(CD, MD, DVD)、符号化(AVI, MPEG,DV)

経路はいっぱいあるねぇ。。

1. 本郷 ➔ 成田空港
2. 成田空港 ➔LGA/JFK/EWR
3. LGA/JFK/EWR ➔ マンハッタン

(*) おっと、船もあったし、AMTRAK(電車) も
飛行機が嫌いな人もいた。。。。
なるべく速く行きたい人もいた。。。。

IPパケットの転送
- 東大からコロンビア大へ -

利用する交通機関(データリンク)も いっぱいあるねぇ。。

1. 足(徒歩ともいう)
2. 自転車
3. 自動車
4. 電車
5. 船
6. 飛行機

(*) たまには予約も必要。。。。混んでると
乗れないこともあるなぁ。。。

IPパケットの転送
- 東大からコロンビア大へ -

物理層もいろいろあるねぇ。。。。

1. 道路: 砂利道、ぬかるみ、舗装路
   交通事故
2. 空　: 高度と経路で値段違うのよねぇ。。。
   墜落
3. 電車: 乗り心地、、、
   列車事故(遅延、行けなくなる)
4. 海　: 天候次第でえらい違い
   台風に巻き込まれると遭難、、、

IPパケットの転送
- 東大からコロンビア大へ -

<< 電話、放送、インターネットの違い >>

電話 ： 先に偵察隊(シグナリング)が飛び、
糸があるか確認し、確保し、報告。
その後に、実際にものを送る。

放送 ：面倒なので、とりあえず、送り続ける。
受信者側が、受け取るかどうか決める。
(SPAMに似ている)

インターネット： つながっているかどうかの情報
は報告されている(経路制御)。
つながっていれば、転送しちゃう。
届かないかもしれない。。。。。

通信資源 vs 処理負荷 vs 通信品質

- 放送 ; 常に資源確保
  - 通信資源を浪費し、通信品質を常に確保
  - エンドノードは、SPAMフィルター機能を実装
- 電話; 必要な時に On-Demandに資源確保
  - 実は、ベストエフォート型のサービス。
  - エンドノードは、最も、単純(?)
- インターネット ; いつでも流せる
  - 最も、通信資源の利用コストが小さい。
  - エンドノードが自力で通信品質を確保

用語の定義

- ノード (e.g., 自宅、駅、空港)
  - インターネットに接続できる機器
- ルータ・中継ノード (e.g., 駅、交差点)
  - 受け取ったパケットを転送できる(バケツリレー）ノード
- ホスト・エンドノード (e.g., 自宅)
  - ルータ・中継ノード以外のノード
    - 通常、家庭や会社にあるPCはホスト・エンドノード

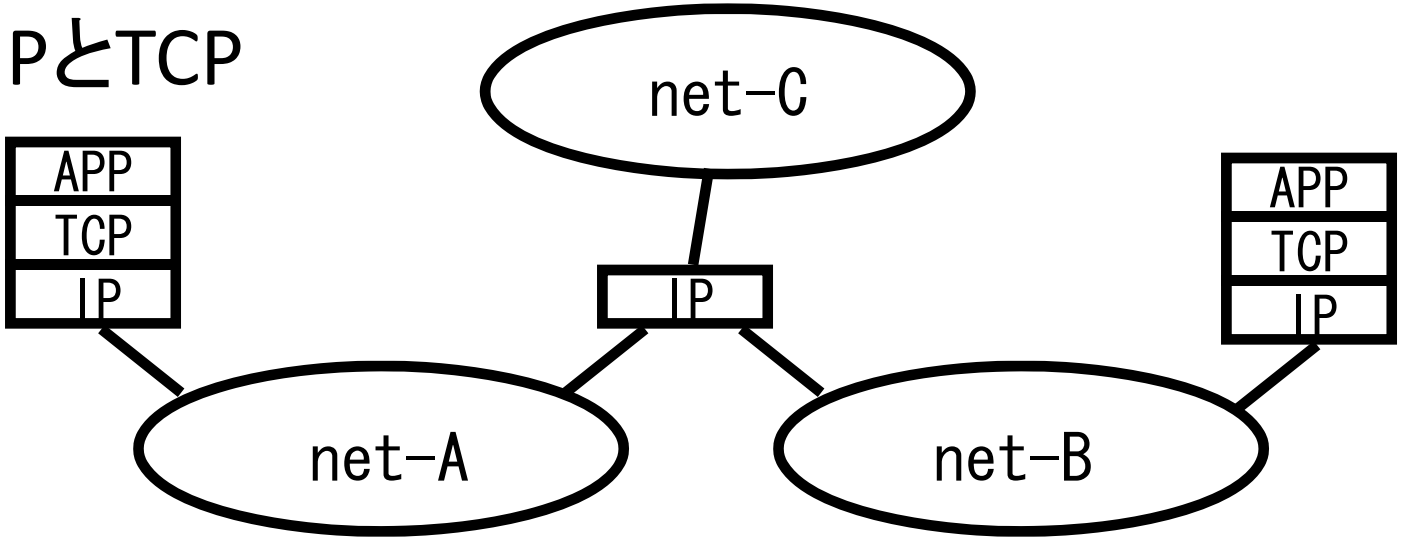
ルータとノード

- いろいろなリンク(乗り物)を収容する。
- 人を適切なリンク(乗り物)に乗せる
  - ➔ 乗り物がたくさんあると悩んでしまう。。。。
- 車の交差点は特殊なケース
- 交差点(i.e., ルータ)人が乗り物を乗り換える
- どこで降りようか/乗り換えようか？
  - ➔ 経路制御(Routing Protocol)の役目
- どの乗り物(席、時間)を使おうか？
  - ➔ リンクアクセス制御(MAC (Media Access Control) Protocol)
- 混んでいる乗り換え点では、事故が絶えない
  - ➔ 渋滞、衝突、乗り遅れ

IP - インターネット・プロトコル

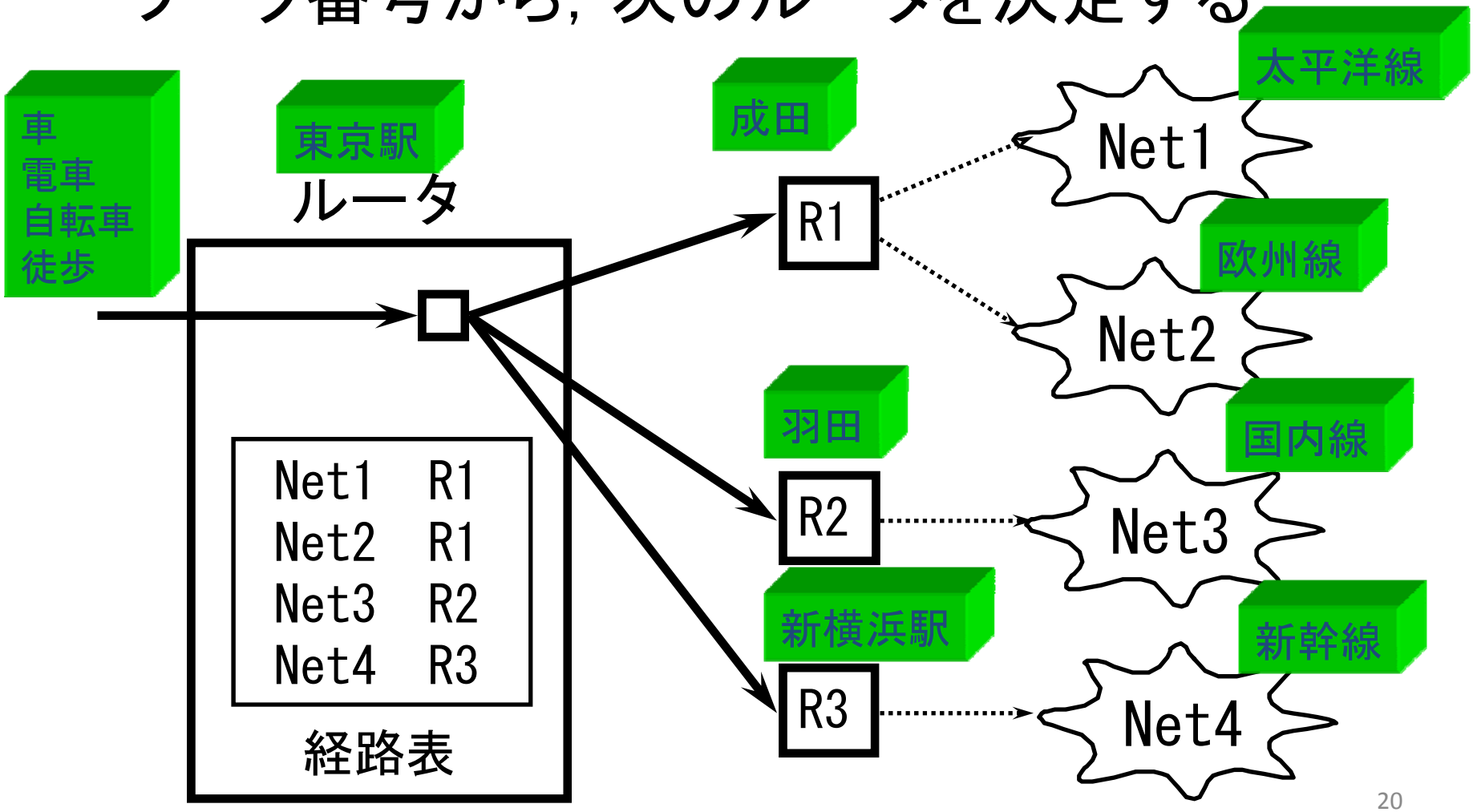
- IPパケット(= 人)

- IPとTCP



経路制御

- ルータは受信したパケットの送り先のネットワーク番号から，次のルータを決定する



IP(Internet Protocol) ～機能と特徴～

- **機能**
  - ホスト(インターフェース)の識別
    - 32bits(or 128bits)の識別子
    - IPアドレス
  - データの中継
    - 乗換駅の例： 乗換え
    - ルータ・ゲートウェイ
  - データの分割
    - 一度に送れないデータを分割、再構成
- **特徴**
  - 信頼性のない通信を提供
    - データグラム型
    - 「失敗したらごめんね」
    - 最大限努力する
  - 大規模な環境
    - 処理が単純
  - 仕様が公開
    - RFC791

IPの仕事

- 指定されたコンピュータへデータを届ける
  - 相手の指定(Addressing)
  - データの中継(Routing)
- データの大きさの制限
  - 大きなデータは分割(Fragment)
  - 受け取ったら元に戻す(Reassemble)

みなさんからの質問

- なんで、アクトビラ(acTVila) って あるの?
  - http://actvila.jp/

  ➔国際(International) vs グローバル(Global)
    - FMラジオの周波数

      (海外 88～108MHZ, 日本だけが76～90MHZ)
    - 携帯電話
    - ISDN
- “オープン” ってなんですか?
  - マイクロソフトは、「オープンスタンダード 」な会社と呼ばれています。

アドレス

- 2つの目的
  - 目的のノード(のインターフェース) を識別するための識別子
  - 目的のノード(のインターフェース)へパケットを届ける(Routing)ためのヒント
- アドレスの構造
  - 階層化(Hierarchy)と集約化(Aggregation)
- 効率的な Routingのためのアドレス構造
  - 究極的には、個別の識別子でやりたい
  - 現実的には、集約化したアドレス空間を定義
    - アドレス空間の大きさ(＝桁数)

アドレス構造

- 効率的な Routingのためのアドレス構造
  - 可変長 vs 固定長
    - 情報理論的には、可変長で、情報の多いもの(多くのノードを収容する空間)には、短いシンボル(アドレス長)を割り当てるべき。
      ➔ 転送すべき総情報量を極小化できる。
    - ということで。
      - 電話：CC(可変長)、TLA(可変長)、NLA(可変長)、SLA(固定長)
      - インターネット：昔は固定長、今はすべて可変長
      - 放送(ラジオ/テレビ)：特に関係なし

アドレス構造

- 電話システム (E.163)

CC:	TLA:	NLA:	SLA(4桁):
北米＝ 1	東京＝03	東大＝5684	1234
日本＝ 81	横浜＝045	片平＝986	
	筑後＝0942	羽犬塚＝53	

- インターネットシステム
  - 昔： 8ビットごとにAggregate
    - クラスA (8ビット)、クラスB(16ビット)、クラスC(24ビット)
  - 今： 完全に可変長に Aggregate

オーダー。。。

- 階層化と集約化が、運命の分かれ道
  - 構造を 持たない アドレス
    - MACアドレスとか
    - オーダー n ； o(n) と書く。
  - 構造を持つアドレス
    - 階層化
      - 下手な階層化 ➔ 実は 依然として o(n) になる。
      - 上手な階層化 (集約化との組み合わせ) ➔ o(log n) になる。

        (例) 100を単位に階層化＆集約化

        2階層で1万、3階層で100万、4階層で10億 と指数関数で増加

        ➔ でも、、、、管理する情報量は、常に 100のオーダー。

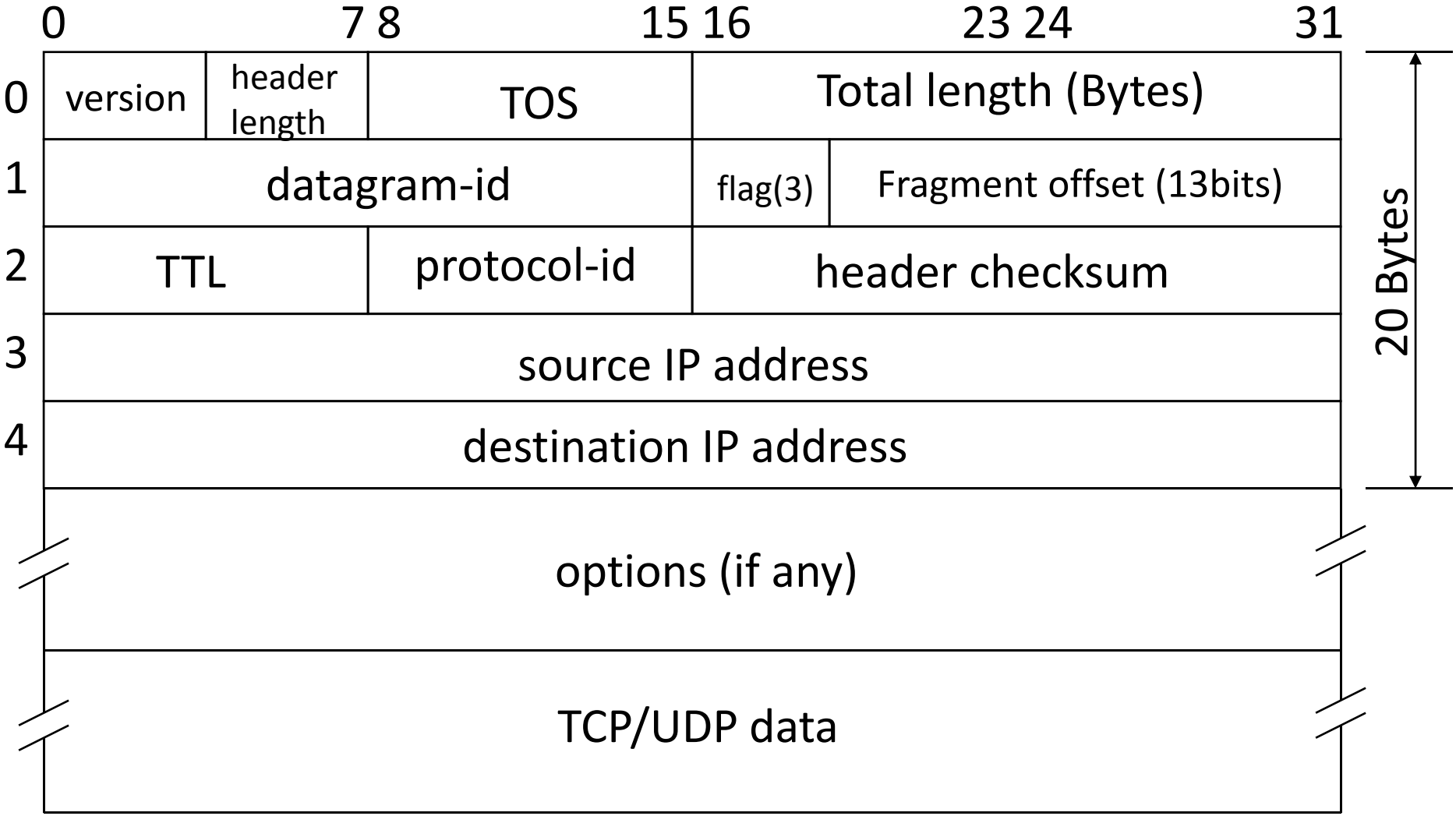
(*) 必要になるもの；2つのアドレスのマッピング/解決

ネットワーク レイヤ

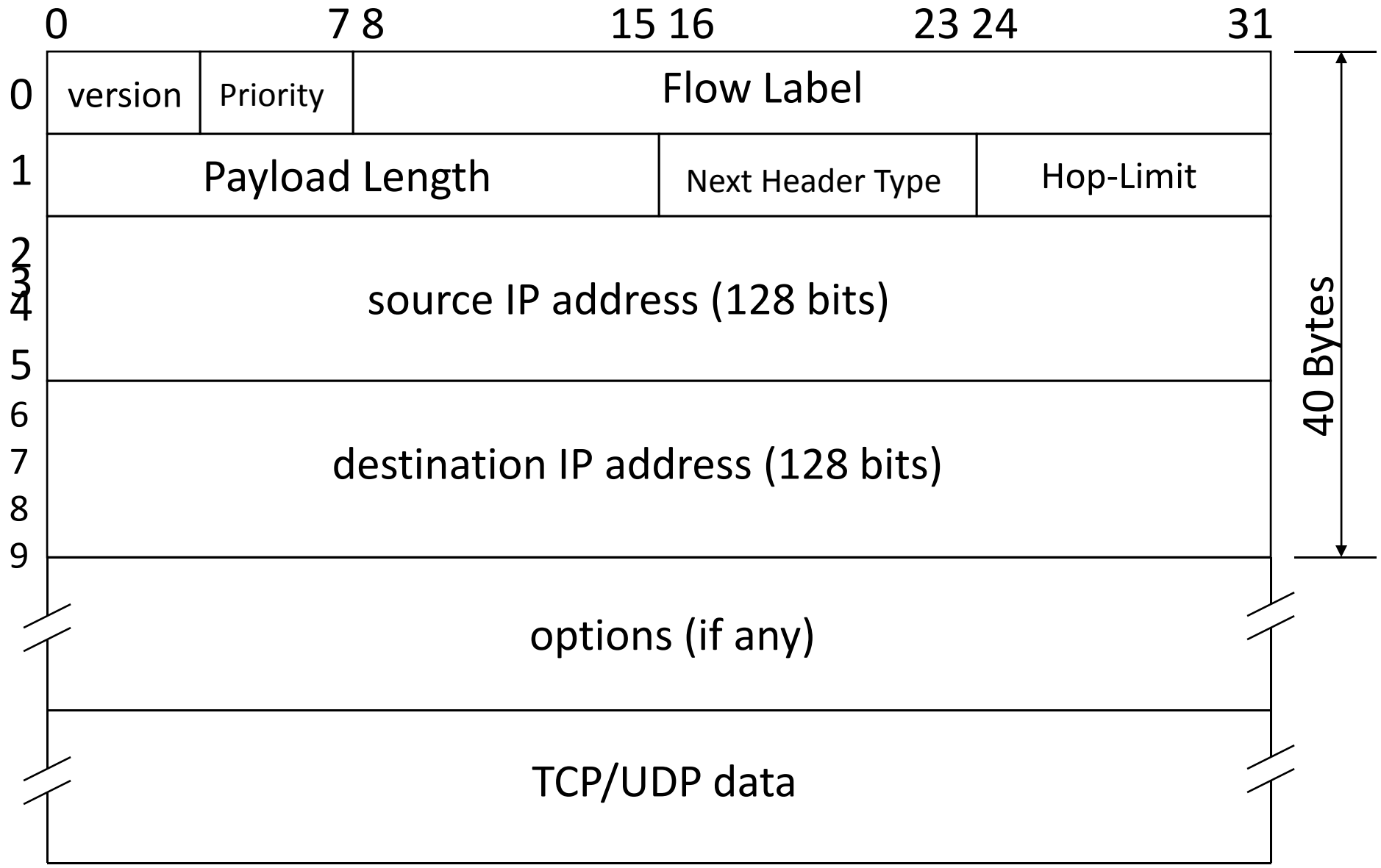
- IP (Internet Protocol) -

→ (1) IP (Internet Protocol)
(2) Addressing
(3) ARP
(4) Routing
(5) Address Discovery (e.g., DHCP)
(6) ICMP/IGMP

IP version 4 (IPv4) Header Format

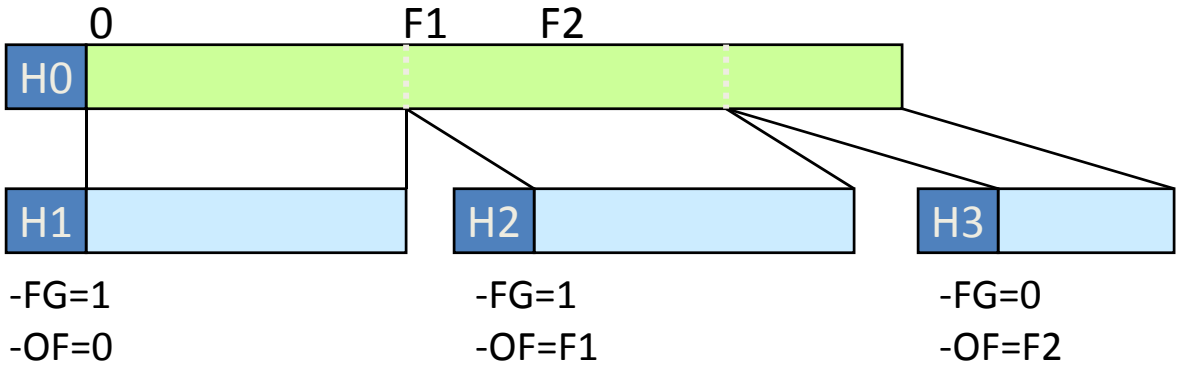


IP version 6 (IPv6) 基本Header Format



Fragment

- リンクMTU以上のパケットはFragment。
  (要は、大き過ぎるパケットは、切り刻む)
- すべてのパケットが同一のsrc_IP, dst_IPを持つ。
- Flag=フラグメントの継続を示す(最後=“0”、他=“1”)
- Offset ; Byte単位の表現(先頭=“0”)



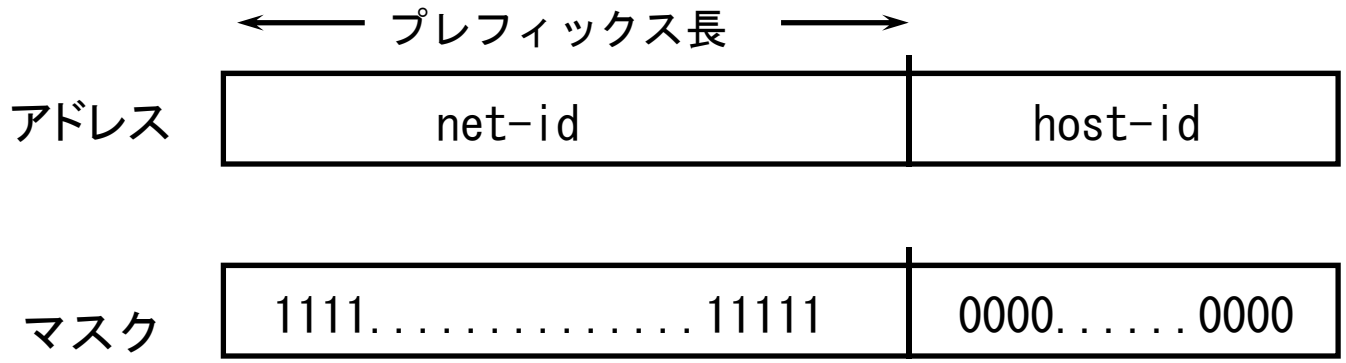
ネットワーク レイヤ
- IP (Internet Protocol) -

(1) IP (Internet Protocol)
→ (2) Addressing
(3) ARP
(4) Routing
(5) Address Discovery (e.g., BOOTP)
(6) ICMP/IGMP

ネットマスクとプレフィックス

- ネットワーク部は可変

- プレフィックス長によるネットワーク番号表記



例）133.201.2/24

プレフィックス長

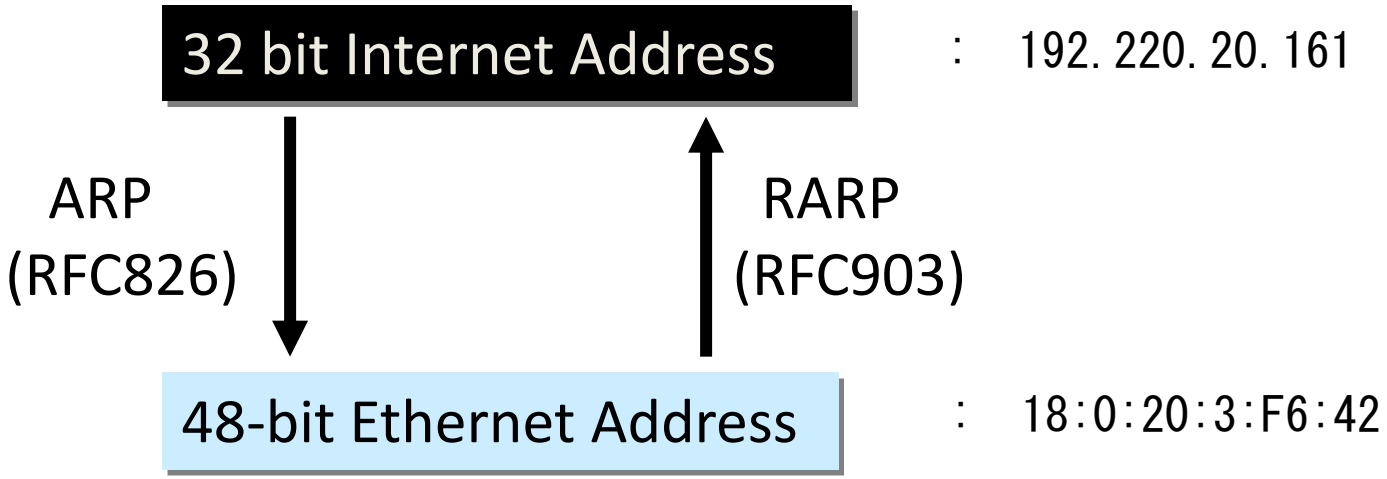
ネットワーク番号

ネットワーク レイヤ

- IP (Internet Protocol) -

(1) IP (Internet Protocol)
(2) Addressing
→ (3) ARP
(4) Routing
(5) Address Discovery (e.g., DHCP)
(6) ICMP/IGMP

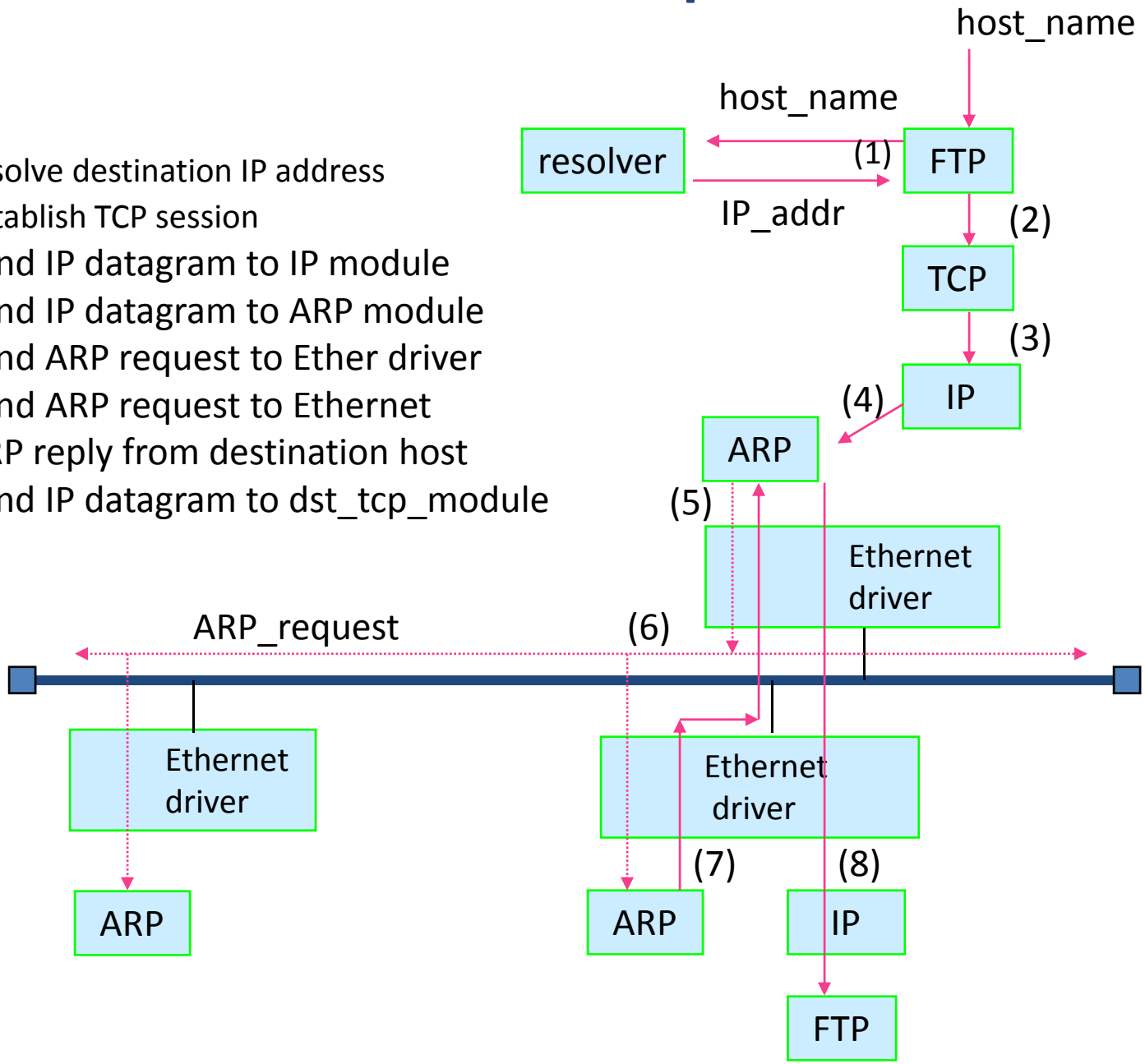
ARP(Address Resolution Protocol)
RARP(Reverse ARP)



(*) IPv6 では、Neighbor Discovery Protocol が提供

ARP when we use ftp

(1) resolve destination IP address
(2) establish TCP session
(3) send IP datagram to IP module
(4) send IP datagram to ARP module
(5) send ARP request to Ether driver
(6) send ARP request to Ethernet
(7) ARP reply from destination host
(8) send IP datagram to dst_tcp_module



ネットワーク レイヤ

- IP (Internet Protocol) -

(1) IP (Internet Protocol)
(2) Addressing
(3) ARP
→ (4) Routing
(5) Address Discovery (e.g., DHCP)
(6) ICMP/IGMP

Routing

[Routingの種類]

(a) 静的ルーティング(static_routing, default_routing)
(b) 動的ルーティング(dynamic_routing)

[Routing Process(Dynamic Routing)が行う仕事]

(i) 経路情報の広告(to 隣接ノード)
(ii) 経路の計算
(iii) パケットの送信・転送・受信

[パケットの受信時の手続き]

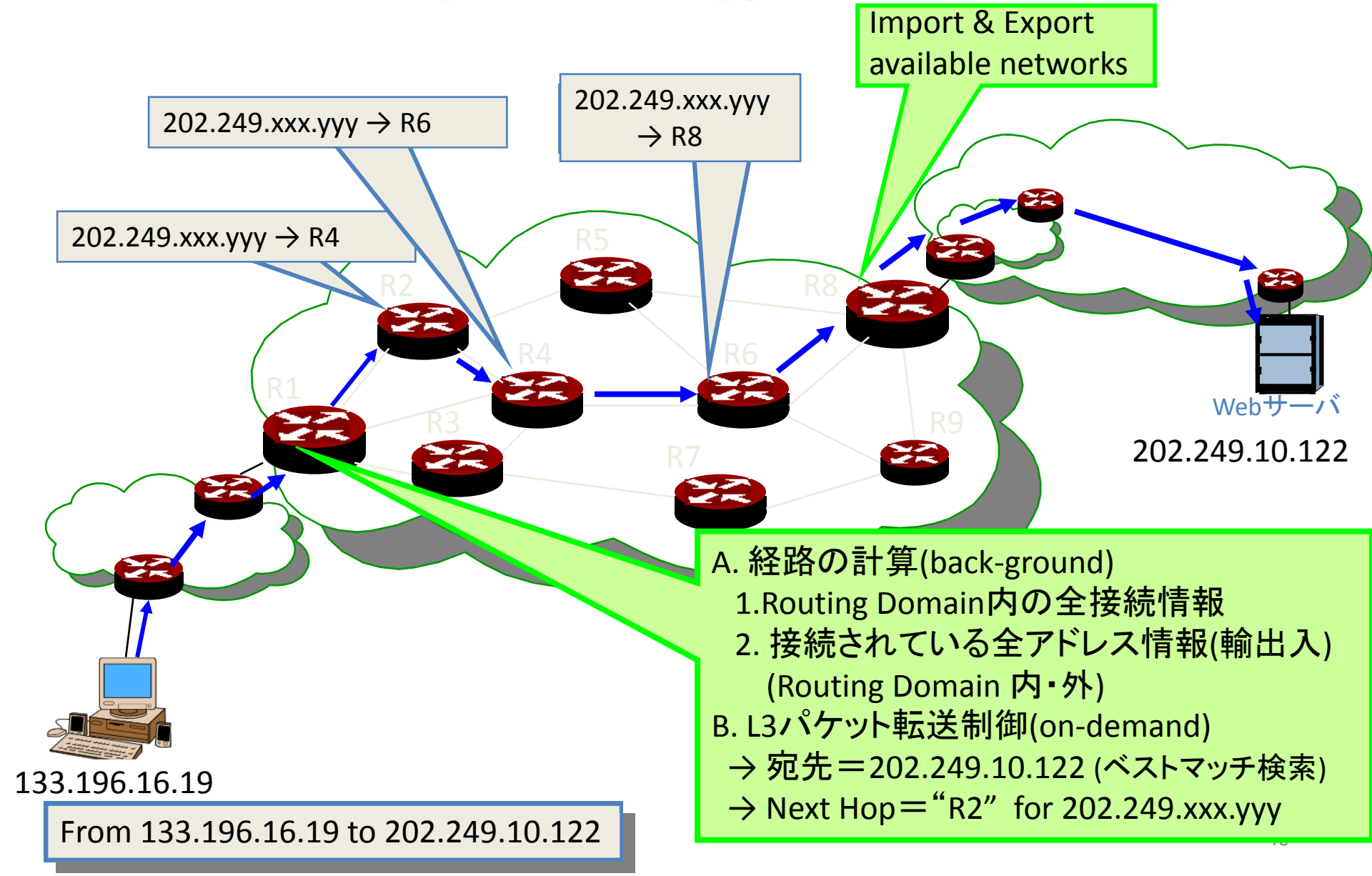
(1) host_address の検索
(2) network_address の検索
(3) defaultエントリー の検索

経路制御のお仕事。

- 数学的な表現
  - あるノード(目的地)をルート(根) とするツリーを形成する。ツリーは、多数存在可能であるが、あるコスト関数を用いて 唯一のツリーを選択させる。
  - これを、すべてのノードに対して生成する。
  - 全世界規模の完全なツリーを管理することは困難なので、上手に 再帰的(Recursive)な構造を持ち込む。
- ツリー ;
  - 閉じた部分を持たないトポロジー

IP経路(ルーティング)制御

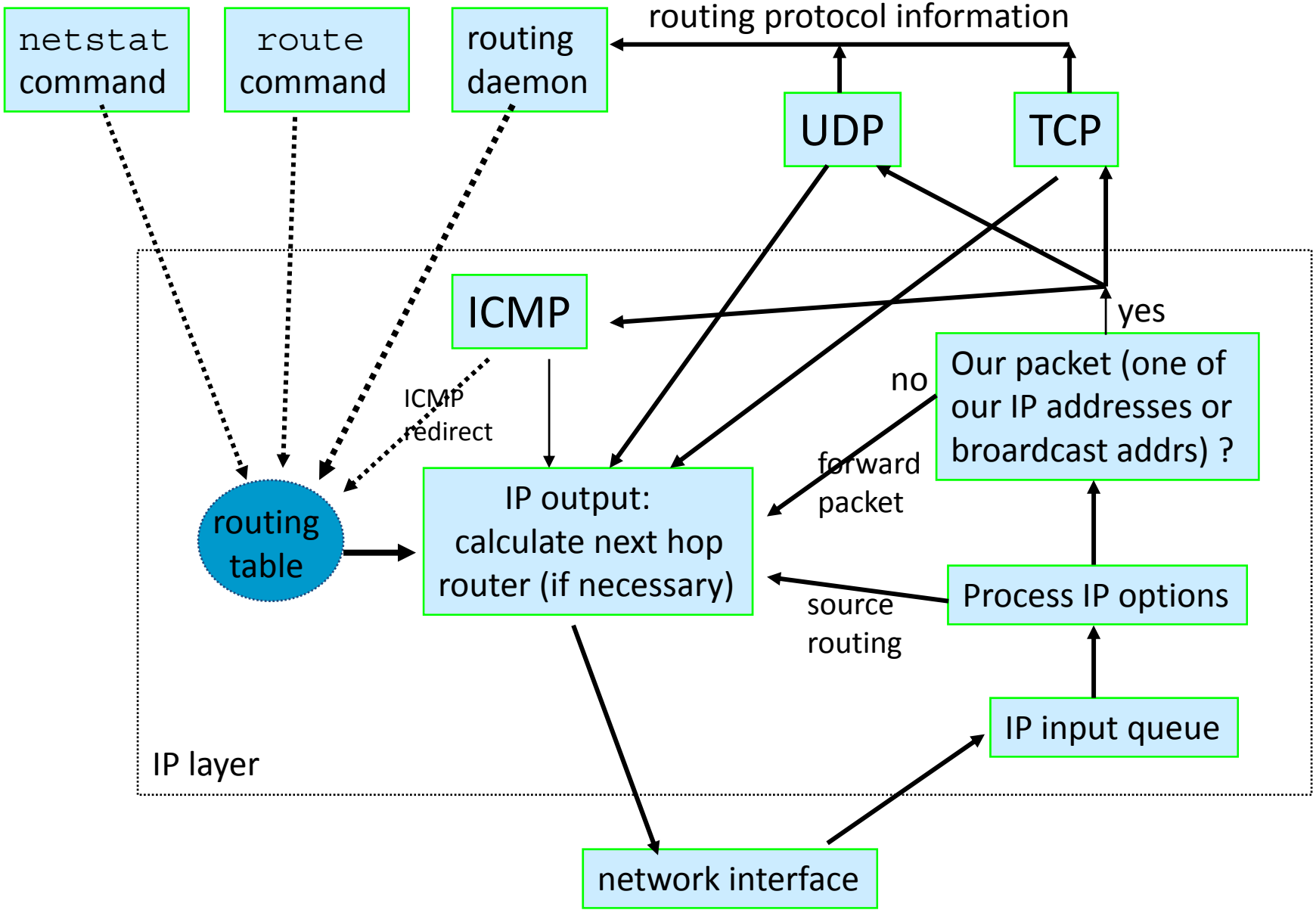
- What a dynamic routing protocol does -



A. 経路の計算(back-ground)
1.Routing Domain内の全接続情報
2. 接続されている全アドレス情報(輸出入)
(Routing Domain 内・外)

B. L3パケット転送制御(on-demand)
→ 宛先＝202.249.10.122 (ベストマッチ検索)
→ Next Hop＝“R2” for 202.249.xxx.yyy

From 133.196.16.19 to 202.249.10.122



<< Datagram processing in the node >>

Dynamic Routing Protocols

[Unicast Routing]

(1) IGP (Interior Gateway Protocol)

(a) RIP (Routing Information Protocol)

(b) OSPF (Open Shortest Path First)

(2) EGP (Exterior Gateway Protocol)

(c) BGP4 (Border Gateway Protocol version 4)

[Multicast Routing]

(i) DVMRP (Distance Vector Multicast Routing Protocol)

(ii) MOSPF (Multicast OSPF)

(iii) PIM (Protocol Independent Multicast protocol)

(iv) MBGP (Multicast BGP)

Dynamic Routing Protocols

[経路計算アルゴリズム]

(1) 距離ベクトル(Distance Vector) 方式　; DV型
(2) リンク状態 (Link State) 方式　; LS型
(3) パス ベクトル (Path Vector) 方式　; PV型
(4) ソースルーティング (Source Routing) 方式

Routing Type	DV型	LS型	PV型	Object	IPv6
RIP	Yes			routed	RIPng
OSPF		Yes		gated	OSPFv6
BGP4			Yes	gated	BGP4＋
DVMRP	Yes			mrouted	―
MOSPF	Yes		―	―	
PIM	n/a	n/a	n/a	―	―
MBGP			Yes	―	―

Distance Vector Routing Protocol

- Bellman-Ford アルゴリズム (or Bellman-Fullkerson)
  - 1969年頃のARPANETにおいて使用
  - Developed by Xerox-PARC as XNS-RIP
  - Nodeから到達可能な宛先アドレスへの距離(Distance)情報。すなわち、到達可能な宛先アドレスへの最適経路のベクトル情報を管理し、最も距離(Distance)が小さなj経路を、宛先ネットワークへの最適経路と判断する。
  - ルーティングドメインの接続情報は持たない。
    ① Distance Ventor：
    宛先ネットワークに対する最短の距離情報ベクトル
    ② 次ホップノード情報

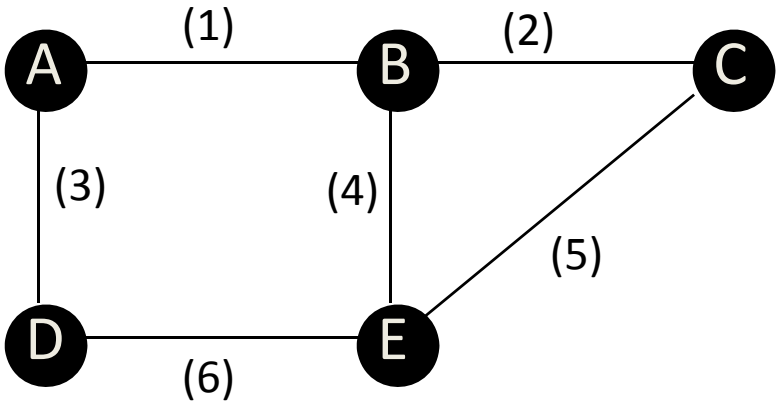
    {dst_net, distance, next_hop_node}

Routing Information Protocol (RIP)

- RIP for IPv4 ; RFC 1058/1721/1722/1723/1724
  RIP for IPv6 ; RFC 2080
- routed (ルートデーモン) in BSD, SunOS
  - 最大ホップ数 ; 15 ホップ
  - Cold-Start 方式; 経路の計算に最大450秒(=15x30秒)必要
  - 30秒ごとにDistance Vectorデータベース情報の交換
    → ノード/リンクの状態変化もホップホップに伝播
  - UDP(port=520)を用いたデータベース(Distance Vector)の交換。
  - 180秒返事がないと故障と判断する(keep-alive方式)
  - 最適経路の計算式;
    - D(i,j) ; distance vector
    - d(i,j) ; node_i と node_j 間の距離

$$D(i,j) = \min [d(i,k) + D(k,j)] \quad \text{(for all k)}$$

Routing Information Protocol (RIP)



(1) 0 sec

From A to	Link	Cost
A	local	0

B → A

From B to	Link	Cost
B	local	0

D → A

From D to	Link	Cost
D	local	0

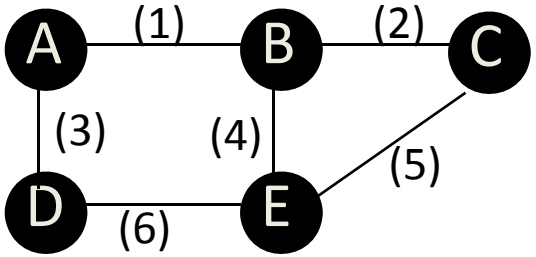
(2) 30 sec

From A to	Link	Cost
A	local	0
B	(1)	1
D	(3)	1

Routing Information Protocol (RIP)

(2) 30 sec

From A to	Link	Cost
A	local	0
B	(1)	1
D	(3)	1



B → A

From B to	Link	Cost
A	(1)	1
B	local	0
C	(2)	1
E	(4)	1

D → A

From D to	Link	Cost
A	(3)	1
D	local	0
E	(6)	1

(3) 60 sec

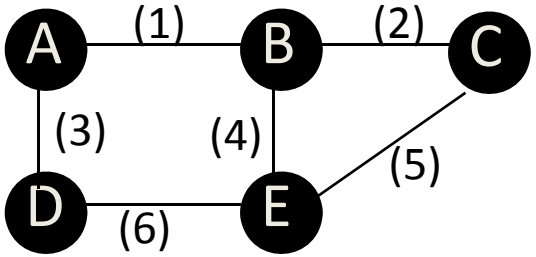
From A to	Link	Cost
A	local	0
B	(1)	1
C	(1)	2
D	(3)	1
E	(1)	2

Select one from even two path
→ {(1),2} vs {(3), 2}

Routing Information Protocol (RIP)

(3) 60 sec

From A to	Link	Cost
A	local	0
B	(1)	1
C	(1)	2
D	(3)	1
E	(1)	2



B → A

From B to	Link	Cost
A	(1)	1
B	local	0
C	(2)	1
D	(1)	2
E	(4)	1

D → A

From D to	Link	Cost
A	(3)	1
B	(3)	2
C	(6)	2
D	local	0
E	(6)	1

(4) 90 sec

From A to	Link	Cost
A	local	0
B	(1)	1
C	(1)	2
D	(3)	1
E	(1)	2

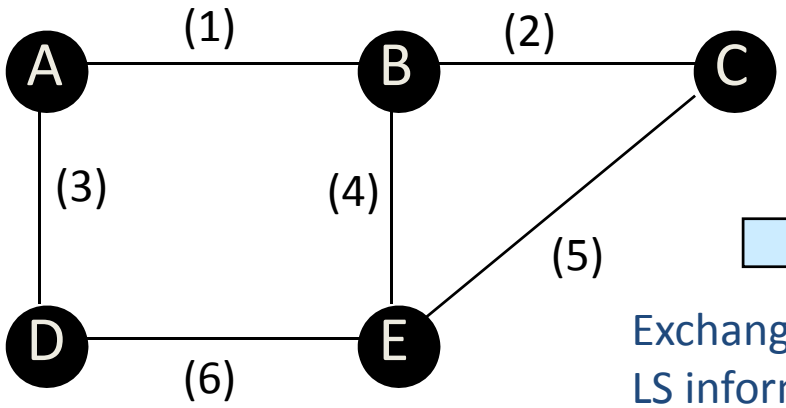
Link State Routing Protocol

- SFP (Shortest Path First) 方式

  - 1970年頃のARPANETにおいて使用
    → 大規模化への対応(Distance Vectorの課題の克服)

  - ルーティングドメイン内の各ノード間の接続トポロジー情報 と、ノード間の接続リンクのコスト値(Attribute value)のデータベースを持つ。
  - リンク状態データベースは、ルーティングドメイン内のすべてのノードで同一。
  - 同一のデータベース情報、同一の経路計算アルゴリズムを適用して各ノードで独立に最適経路(自ノードから宛先ネットワークへ最小コストで到達するために取るべき次ホップノードの選択)の計算を行う。
  - リンクコストの積算値が極小となる経路を「最小コストの経路」とする。
    → 宛先ネットワークごとに、最小コストを提供する
    Spanning_Tree の形成

Open Shortest Path First (OSPF)

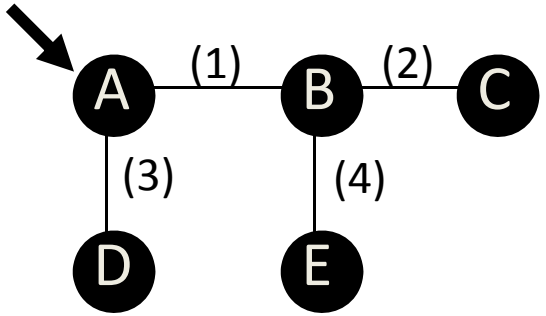


Exchange LS information between neighbor node

[Link State Data-Base]

From	To	Link-id	Distance
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Calculate spanning tree



Build routing table

At Node "A"

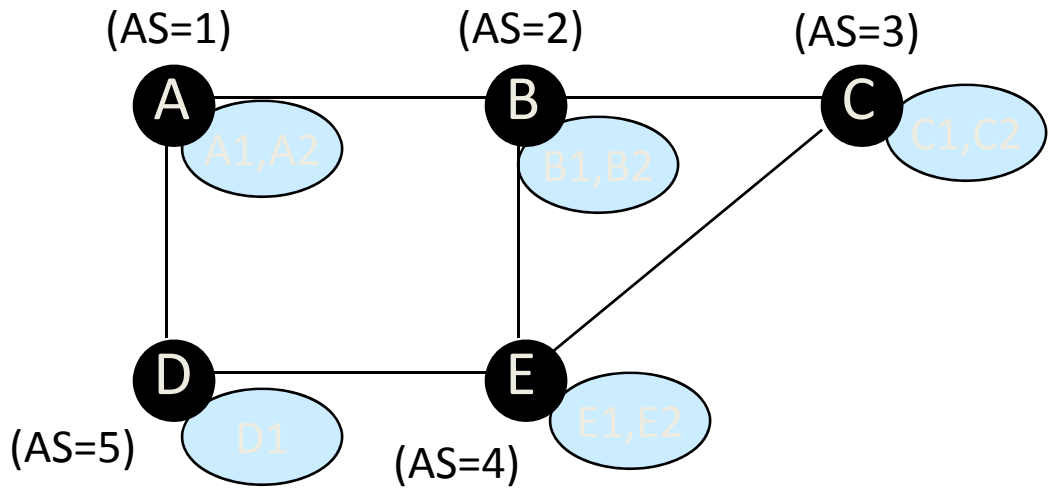
To	Next-Hop	Link-id
B	B	1
C	B	1
D	D	3
E	B	1

Path Vector Routing Protocol

- 1982年頃のRFC827 (EGP; Exterior Gateway Protocol)
  → NSFNETバックボーンと地域バックボーンの相互接続を目的として開発された(RFC1093)。
  → 各ネットワークの運用・制御ポリシーを反映させることのできるルーティングプロトコル。

- AS (Autonomous System)間での経路制御;
  16 bits で表現される 自律ネットワーク(AS番号)間で、各宛先ASのネットワークへ到達するための経路を、AS番号の順序列(Path―Vector)、あるいはAS番号の集合を用いて表現する。 各ASの境界ルータ(Boarder Router)は、隣接ルータ(Peering Router)と、自ASから、すべての到達可能な宛先ASへの到達経路(path)を広告する。 Path(AS番号の順序列)の集合情報を利用するので、Path-Vector方式と呼ばれている。

Path Vector Routing



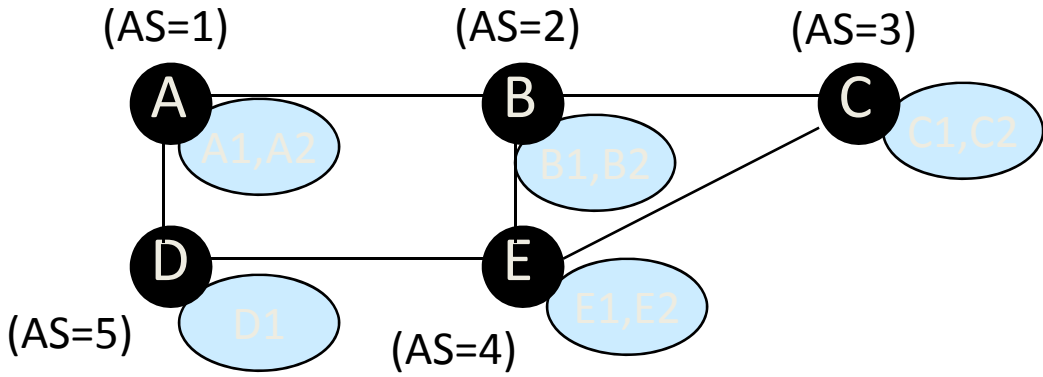
[Path Vector Data-Base in node A]

From	To	Path	Next-Router
A	B1	A,B	B
A	B2	A,B	B
A	C1	A,B,C	B
A	C2	A,B,C	B
A	D1	A,D	D
A	E1	A,D,E	D
A	E2	A,B,E	B

[Path Vector Data-Base in node C]

From	To	Path	Next-Router
C	B1	C,B	B
C	B2	C,B	B
C	A1	C,B,A	B
C	A2	C,E,D,A	B
C	D1	A,E,D	E
C	E1	A,E	E
C	E2	A,B,E	B

Path Vector Routing



(1) Step 1

From	To	Path	Next-Router
A	B1	A,B	B
A	B2	A,B	B
A	D1	A,D	D

[From B]

From	To	Path	Next-Router
B	E1	B,E	E
B	E2	B,E	E
B	C1	B,C	C
B	C2	B,C	C

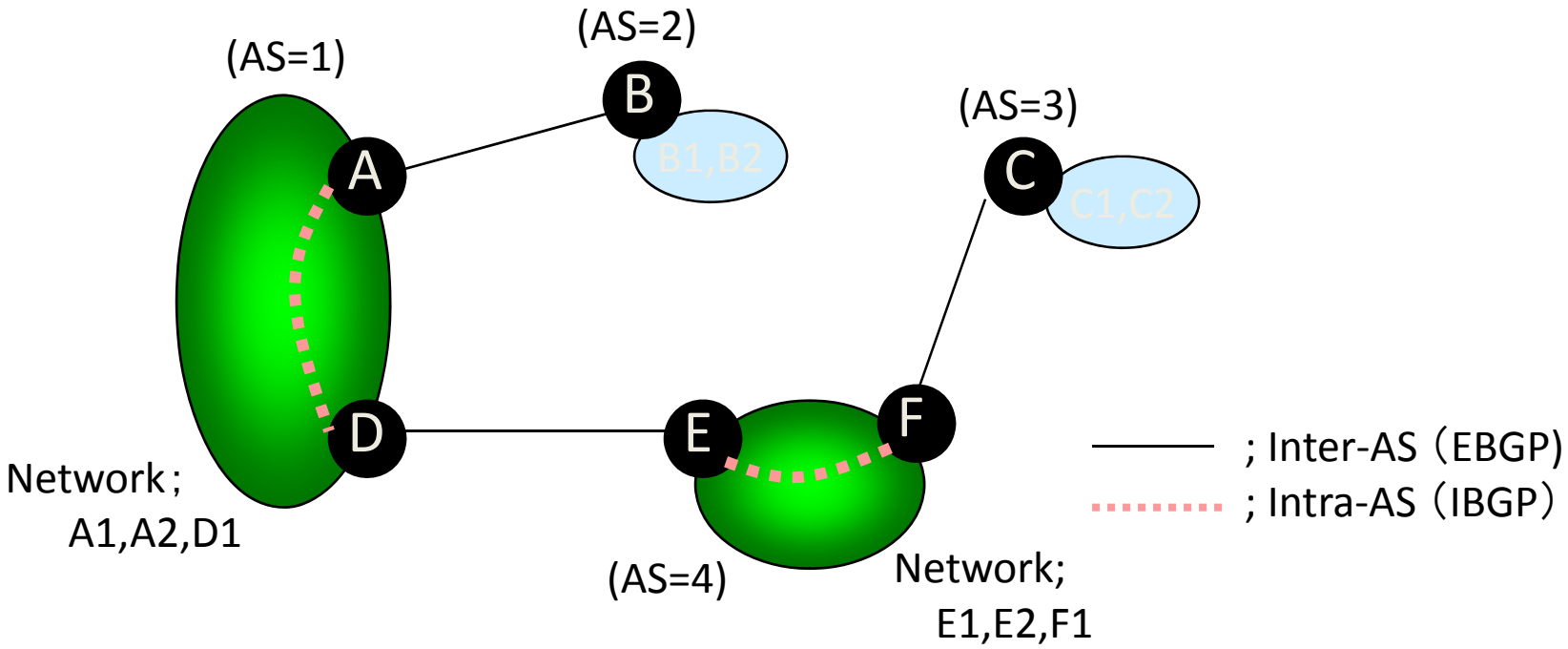
[From D]

From	To	Path	Next-Router
D	A1	D,A	A
D	A2	D,A	A
D	E1	D,E	E
D	E2	D,E	E

(2) Step 2

From	To	Path	Next-Router
A	B1	A,B	B
A	B2	A,B	B
A	C1	A,B,C	B
A	C2	A,B,C	B
A	D1	A,D	D
A	E1	A,D,E	D
A	E2	A,B,E	B

Path Vector Routing ; IBGP



（*）IBGP；（A,D）＆（E,F）

From B1 to E2；{AS=2,AS=1,AS=4}
経路；B →（A → D）→ E

From C2 to B2；{AS=3, AS=4, AS=1, AS=2}
経路; C →（F → E）→（D → A）→ B

経路制御アーキテクチャのまとめ

1. 状態管理に関する点（戦略）
   - 「パスの状態」のみを管理するという戦略
     - GW(=プロトコル変換)での状態管理 (O(np))
     - セッションごと(=On Demand型)の状態管理 ($O(n^2)$)

     ※「パス」≪「セッション」≪「パケット」
2. 大規模化に関する点
   i. 階層構造
   ii. (情報の)集約化
3. システムの持続性（Sustainability）に関する点
   i. AS単位での 運用の自律性
   ii. データリンク技術からの独立性

ネットワーク レイヤ

- IP (Internet Protocol) -

(1) IP (Internet Protocol)
(2) Addressing
(3) ARP
(4) Routing
→ (5) Address Discovery (e.g., DHCP)
(6) ICMP/IGMP

DHCP

(1) IP Address
   (i) Automatic allocation
   (ii) Dynamic allocation ; 貸し出し期間が存在する
   (iii) Static allocation ; Manual configuration
(2) Subnet mask
(3) MTU
(4) Broadcast Address Flavor
(5) Default Gateway
(6) Static Routes Lists

(*) DNSエントリーへの登録は行われない！
   → Internetからの直接アクセスが必要なHostには Static allocation を適用すべき。

DHCPの課題

1. セキュリティー機能
   - BOOTPの延長上のプロトコルであり、
     Security機能を盛りこめない….

2. DNSシステムへの名前の反映
   - サーバ系ノードにはDHCPが適用できず。
   - InternetへのNameの登録が困難…
   - 逆引き(IP_Address → host_name)は不可能
     ではないが、、、、
     正引き(host_name→IP_address)は難しい。

ネットワーク レイヤ

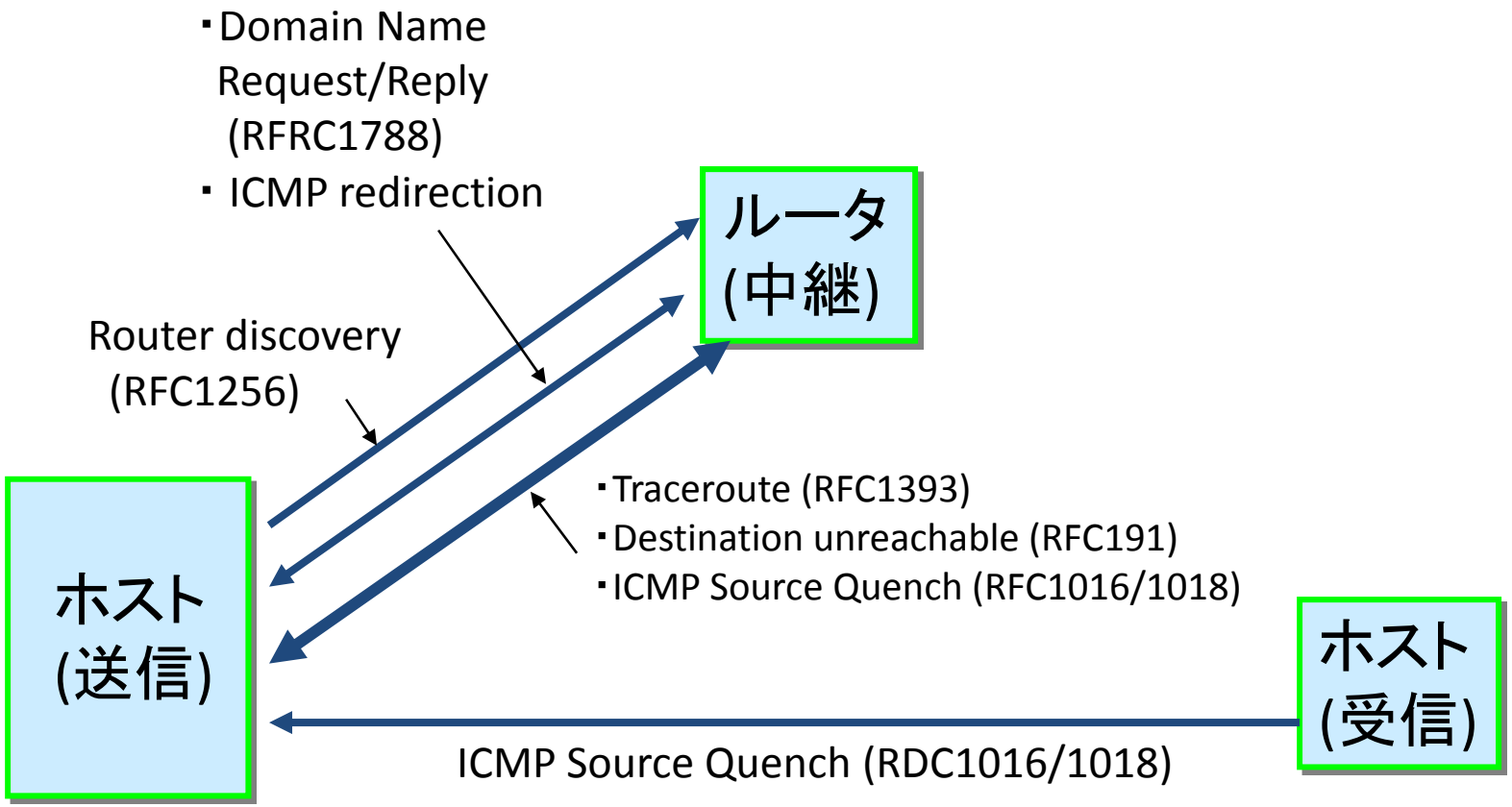
- IP (Internet Protocol) -

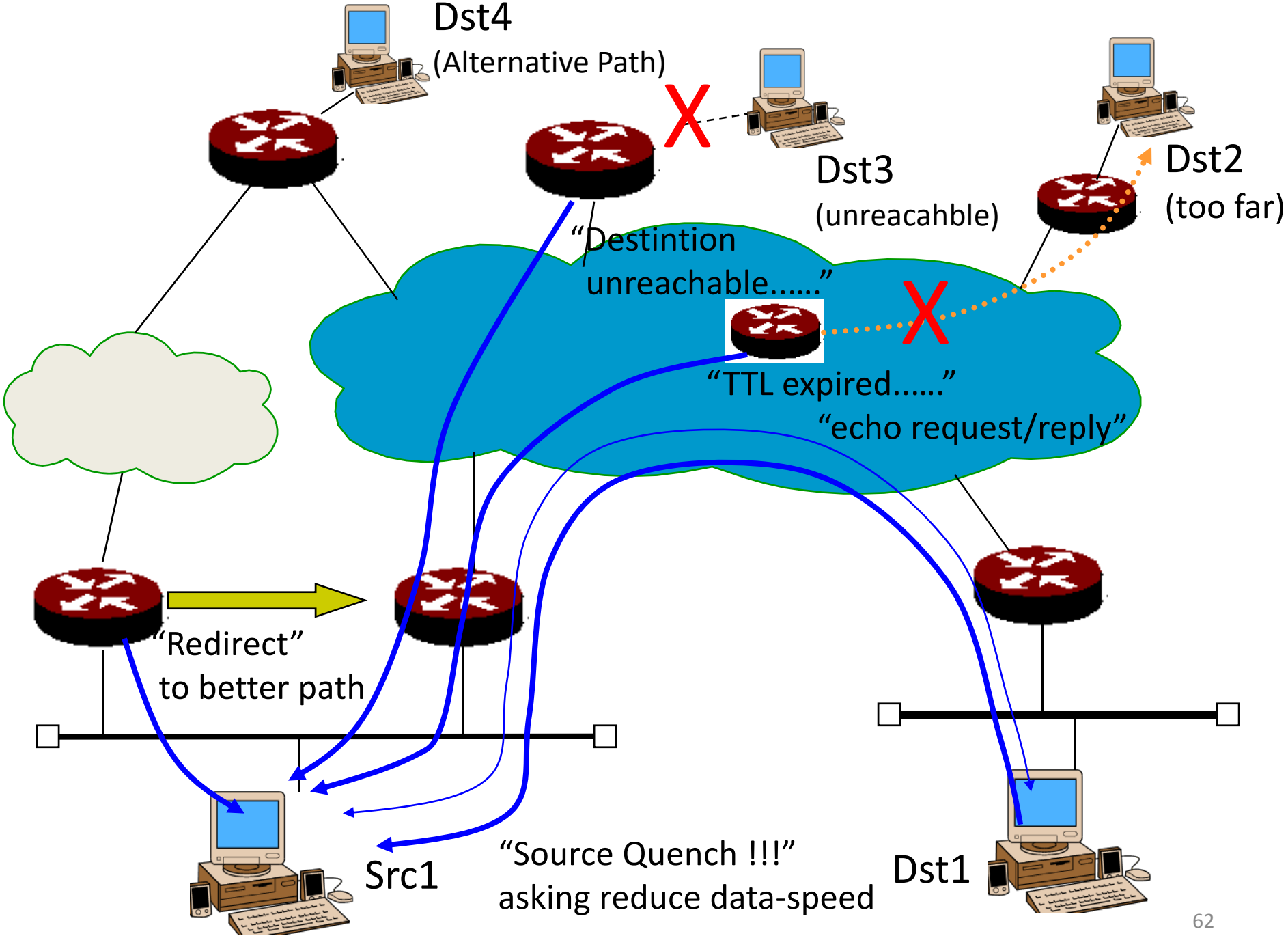
(1) IP (Internet Protocol)
(2) Addressing
(3) ARP
(4) Routing
(5) Address Discovery (e.g., BOOTP)
→ (6) ICMP/IGMP

ICMP/IGMP

- ICMP; Internet Control and Management Protocol
  - IPの管理・制御
  - ICMPを利用した重要なアプリケーション
    - (i) ping
    - (ii) traceroute
    - (iii) Router_Discovery
- IGMP; Internet Group Management Protocol
  - マルチキャストメンバーの管理

ICMP


・Domain Name Request/Reply (RFRC1788)
・ ICMP redirection
ルータ (中継)
Router discovery (RFC1256)
・Traceroute (RFC1393)
・Destination unreachable (RFC191)
・ICMP Source Quench (RFC1016/1018)
ホスト (送信)
ホスト (受信)
ICMP Source Quench (RDC1016/1018)


Dst4
(Alternative Path)
Dst3
(unreacahble)
Dst2
(too far)
"Destintion unreachable....."
"TTL expired....."
"echo request/reply"
"Redirect"
to better path
Src1
"Source Quench !!!"
asking reduce data-speed
Dst1

Ping Program

・Ping 実行時のディスプレイ表示

```
bsdi% ping svr4
PING svr4 (140.252.13.34): 56 data bytes
64 bytes from 140.252.13.34: icmp_ser=0 ttl=255 time=0 ms
64 bytes from 140.252.13.34: icmp_ser=1 ttl=255 time=0 ms
64 bytes from 140.252.13.34: icmp_ser=2 ttl=255 time=0 ms
      :                :                :
```

・tcpdump 出力

```
1 0.0                   bsdi > svr4: icmp : echo request
2 0.003733 (0.0037)     svr4 > bsdi: icmp : echo reply

3 0.998045 (0.9953)     bsdi > svr4: icmp : echo request
4 1.001747 (0.0037)     svr4 > bsdi: icmp : echo reply

5 1.997818 (0.9961)     bsdi > svr4: icmp : echo request
6 2.001542 (0.0037)     svr4 > bsdi: icmp : echo reply

7 2.997610 (0.9961)     bsdi > svr4: icmp : echo request
8 3.001311 (0.0037)     svr4 > bsdi: icmp : echo reply
      :                :                :
```

Ping Program
- with IP RR(Record Route) Option-

・Ping 実行時のディスプレイ表示

```
bsdi% ping -R slip
PING slip (140.252.13.65): 56 data bytes
64 bytes from 140.252.13.65: icmp_ser=0 ttl=254 time=280 ms
RR:     bsdi  (140.252.13.66)
        slip  (140.252.13.65)
        bsdi  (140.252.13.35)
        svr4  (140.252.13.34)
64 bytes from 140.252.13.65: icmp_ser=0 ttl=254 time=280 ms (same route)
64 bytes from 140.252.13.65: icmp_ser=0 ttl=254 time=270 ms (same route)
        :                        :                        :
```

・tcpdump 出力

```
1 0.0                   svr4 > slip: icmp : echo request (ttl 32,
                        id 35835, optlen=24 RR(39)= RR{#0.0.0.0/
                        0.0.0.0/0.0.0.0/0.0.0.0/0.0.0.0} EOL)
2 267746 (0.2677)       slip > svr4: icmp : echo reply (ttl 254,
                        id 1976, optlen=24 RR(39)= RR{#140.252.13.65/
                        140.252.13.35/#00.0.0/0.0.0.0/0.0.0.0} EOL)
```

Traceroute Program

・traceroute 実行時のディスプレイ表示

```
bsdi% traceroute nic.ddn.mil
traceroute to nic.ddn.mil (192.112.36.5), 30 hops max, 40 bytes packets
 1 netb.tuc.noao.edu (140.252.1.183)  218 ms  227 ms  233 ms
 2 gateway.tuc.noao.edu (140.252.1.4) 233 ms  229 ms  204 ms
 3 butch.telecom.arizona.edu (140.252.104.2) 204 ms 228 ms 234 ms
 4 Gabby.Telecom.Arizona.EDU (128.196.128.1) 234 ms 228 ms 204 ms
 5 NSIgate.Telecom.Arizona.EDU (192.80.43.3) 233 ms 228 ms 204 ms
 6 JPL1.NSN.NASA.GOV (128.161.88.2) 234 ms  590 ms  252 ms
 7 JPL3.NSN.NASA.GOV (192.100.15.3) 238 ms  223 ms  234 ms
 8 GSFC3.NSN.NASA.GOV (128.161.3.33) 293 ms  318 ms  324 ms
 9 GSFC8.NSN.NASA.GOV (192.100.13.8) 294 ms  318 ms  294 ms
10 SURA2.NSN.NASA.GOV (128.161.166.2) 323 ms 319 ms 294 ms
11 nsn-FIX-pe.sura.net (192.80.214.253) 294 ms  318 ms  324 ms
12 GSI.NSN.NASA.GOV (128.161.252.2)  293 ms  318 ms  324 ms
13 NIC.DDN.MIL (192.112.36.5) 324 ms  321 ms  324 ms
```

(*) `-g` オプションにより、loose/strict なsource routingを行いながらtracerouteが可能

```
bsdi% traceroute -g  sh.wide.ad.jp  nic.ddn.mil
```

Traceroute Program

・Ping 実行時のディスプレイ表示

```
bsdi% traceroute slip
traceroute to slip (140.252.13.65), 30 hops max, 40 bytes packets
 1 bsdi (140.252.13.35)  20 ms  10 ms  10 ms
 2 slip (140.252.13.65)  120 ms  120 ms  120 ms
```

・tracerouteの動作原理;

→ ICMP time exceed message (type=11,code=0)を利用する。

1. TTL=1 ; On_the-LinkのNext_Hop_Node より先には到達できない。
   → Unreach destination_host due to Time_Exceeded
2. 2度(合計3度) 同一のTTLでICMPパケットの送信
3. TTL ← TTL+1
4. TTL=TTL+1 ; 1_hop先のNodeまでは到達可能になる。
   If reachable → Stop
   If unreachable
   {
   2度(合計3度)同一のTTLでICMPパケットの送信;
   goto step3.;
   }

ネットワークレイヤ
- IP (Internet Protocol) -

(0) 概要
(1) IP (Internet Protocol)
(2) Addressing
(3) ARP
(4) Routing
(5) Address Discovery (e.g., DHCP)
(6) ICMP