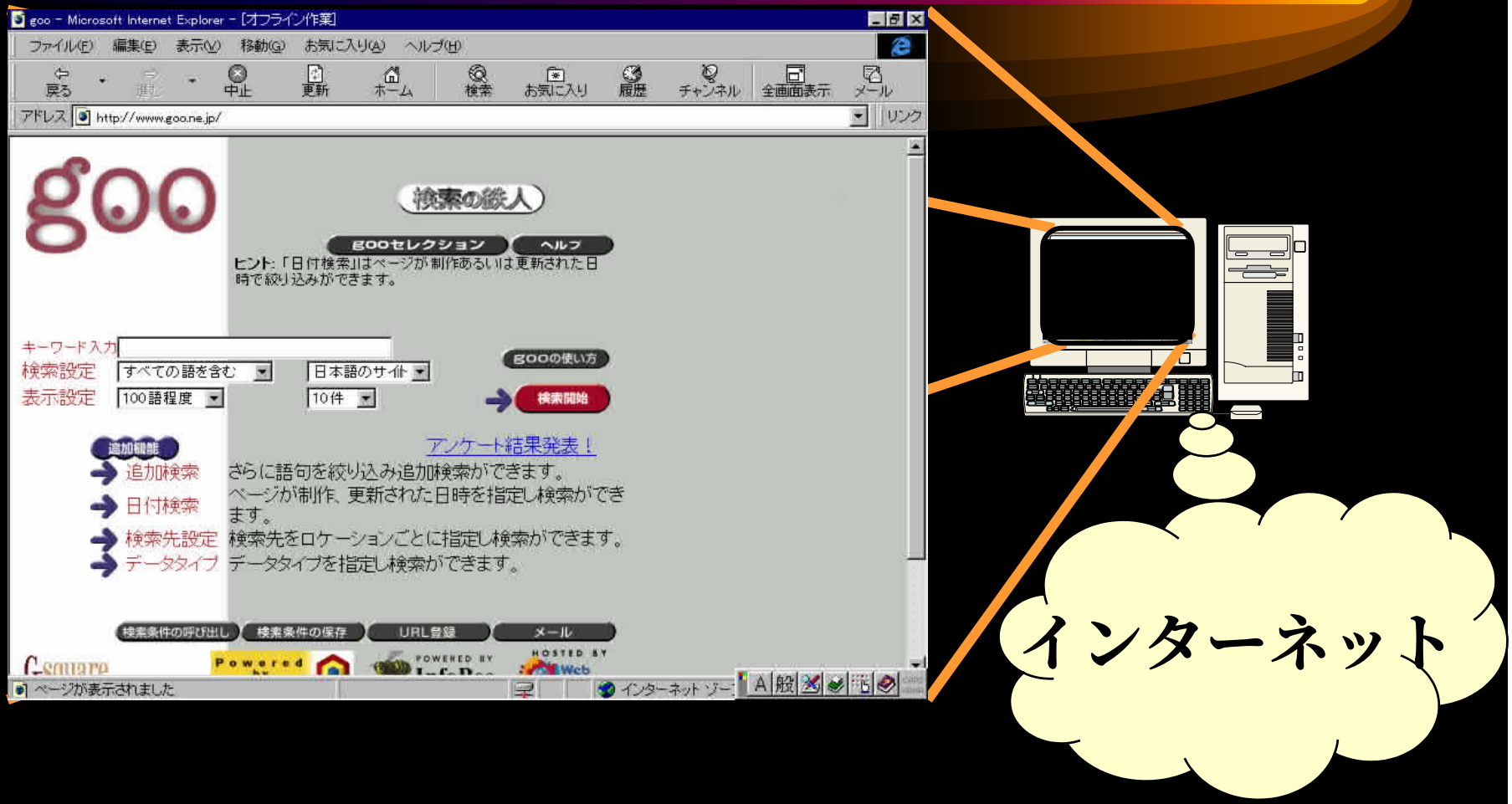
インターネットの向こう側

慶應義塾大学　環境情報学部
村井　純
jun@sfc.keio.ac.jp

ブラウザの向こう側は どうなってるの？



インターネットの構造

- Webブラウザの先では?
  - 小人が働いているわけではない :-)
    - 離れた二つのエンティティを結ぶ仕組み
    - 二つのエンティティで理解が共有できる言葉
- 役割分担
  - 階層的な構造
  - コミュニケーションエンティティでの分担
    - クライアント・サーバ

人間のコミュニケーション

人間とコミュニケーション

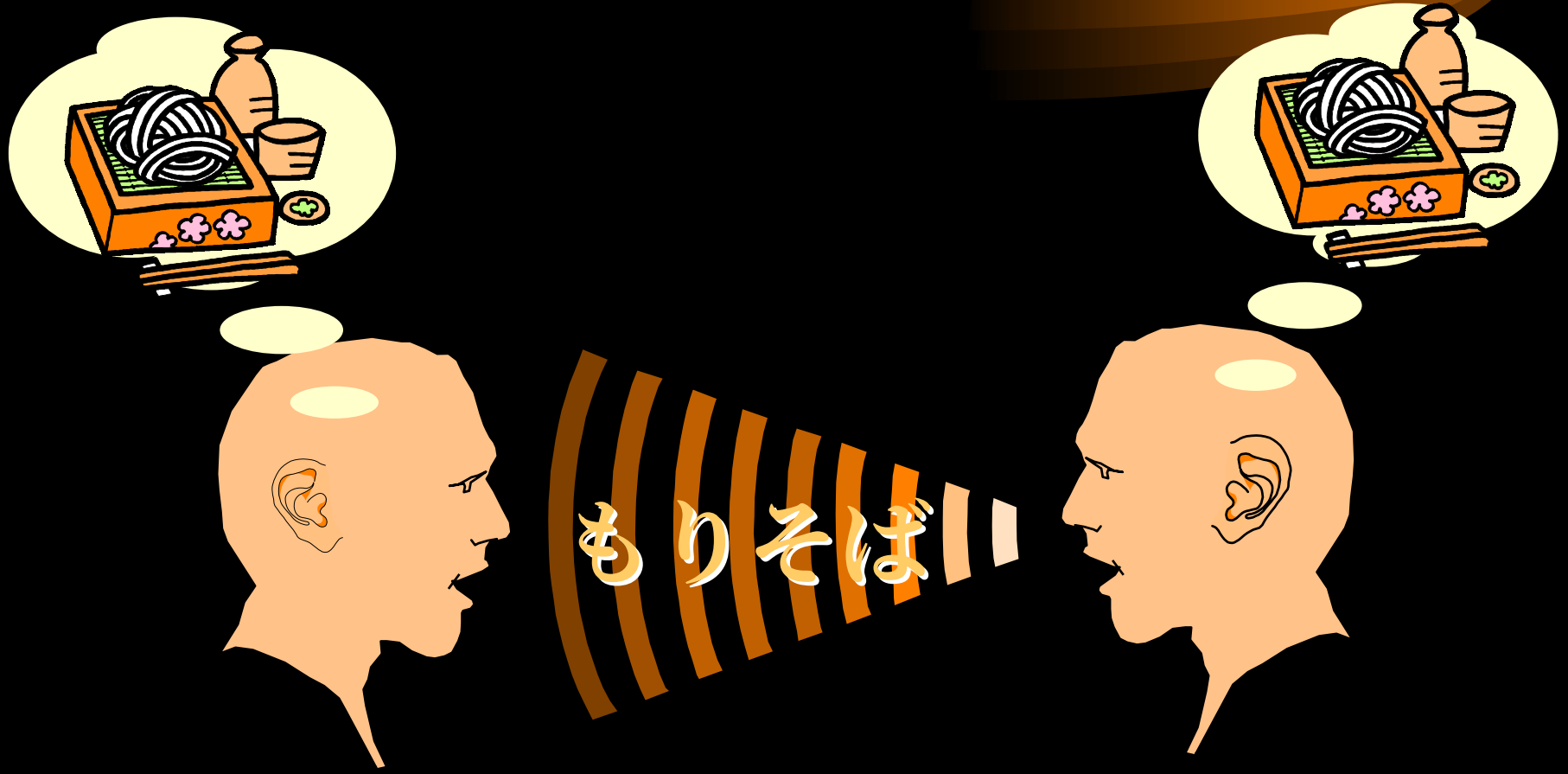
- 思ったことを伝えるには？
  - 言葉、身振り手振り
  - 何か動作をすることで相手に伝える
- どうして伝わるの?
  - 言葉
    - 発した言葉の意味と同じ意味を相手も知っている
  - 身振り手振り
    - 相手の動作から伝えたい意味を推測

言葉の伝わり方

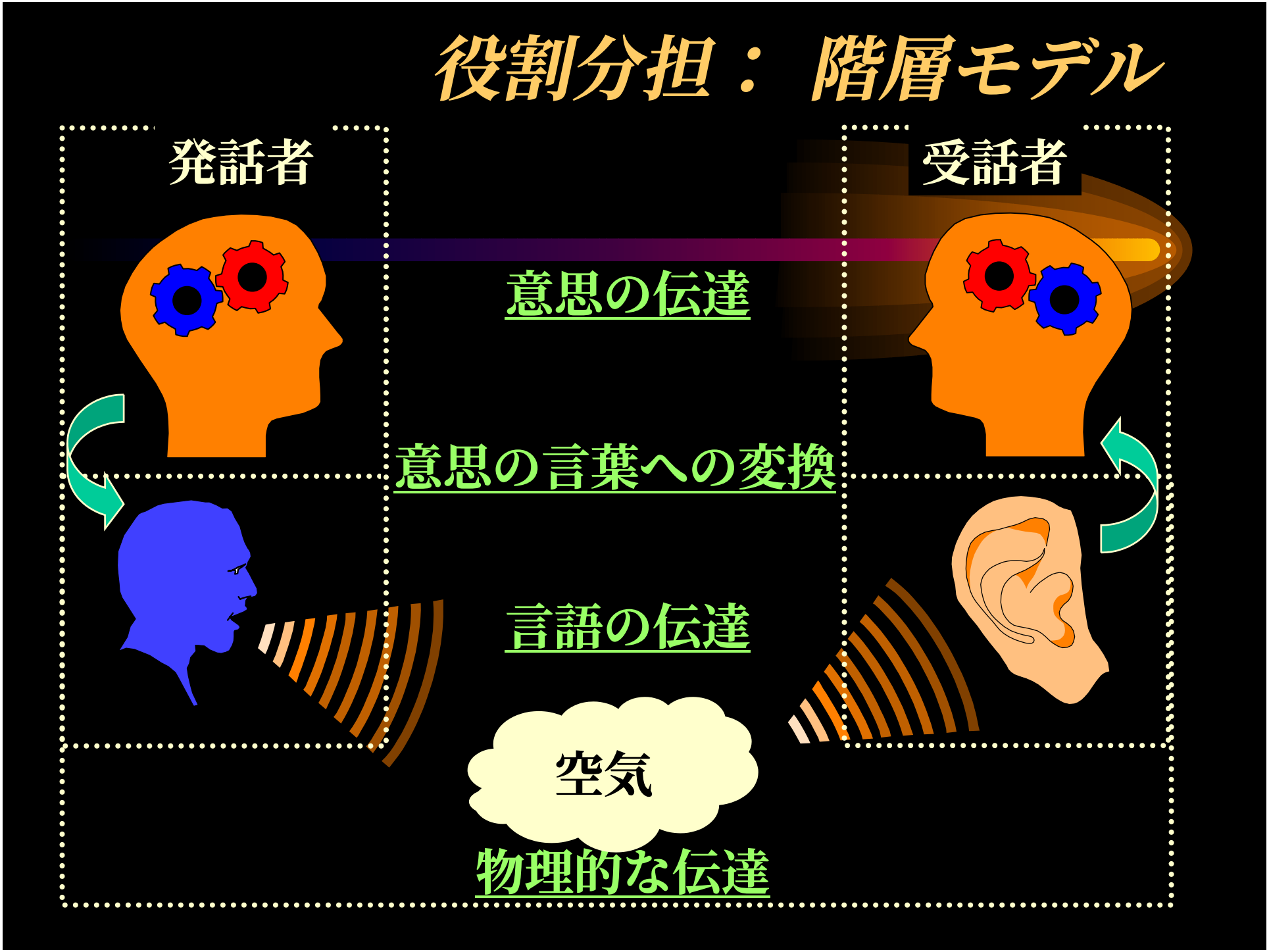
- 脳の中で思ったことが言葉になり．．．
- その言葉を発するために声帯を震わせ．．．
- 声帯の震えが空気に振動として伝わり．．．
- 振動した空気が相手の鼓膜の震わせ．．．
- 鼓膜の振動のパターンが脳に伝わり．．．
- そのパターンと一致したことを連想する

コミュニケーションのための役割分担

- 「もりそば」はどちらも知っている



役割分担： 階層モデル



階層構造

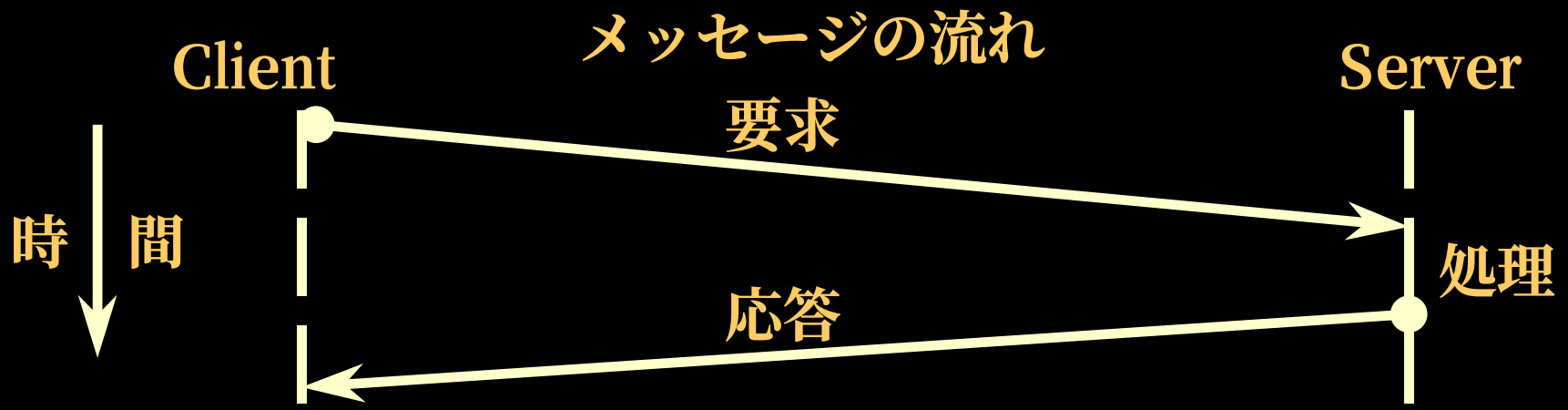
- 相手の同じ役割の部分と協調
  - 空気の振動からの変換　（耳・鼓膜）
  - 空気の振動への変換　　（喉・声帯）
- 階層ごとに役割を持つ
  - 物理的なつながり（空気）
  - コミュニケーション相手とのつながり（声）
  - 意味とイメージとのつながり（言葉）

クライアント・サーバモデル

- 通信相手とのタイミング
  - ２つのプログラムのコミュニケーション
    - ２つが非同期だとうまく通信できない
    - 一方が待ち受けし続け、一方がリクエストを送る
      → クライアント・サーバモデル
- クライアント
  - 能動的にサービス提供を促す側
- サーバ
  - 受動的にサービス提供する側

クライアント・サーバ

- クライアントとサーバの役割分担
  - クライアント
    - 通信を開始するアプリケーション
  - サーバ
    - クライアントからの要求を待ち受けるアプリケーション



クライアント・サーバの例

- サーバ（サービスを提供する側）
  - finger サーバ　　（fingerd）
  - WWWサーバ　　（httpd)
  - 体育予約サーバ
- クライアント（サービスを受ける側）
  - finger クライアント　　(finger)
  - WWWクライアント　　（Netscape）
  - 体育予約クライアント

モデル化・抽象化

- 「お話をする」ということ
  - 登場人物
    - 話をする人，話を聞く人
    - どこにいるの?
    - 何人いるの?
  - 何によって?
    - 相手の言葉と自分の言葉
    - 空気
    - 身振り手振り
- モデル化・抽象化
  - 相手と自分
    - 1対1
    - 1対多
    - 多対多
    - スコープ
  - お話の手順・方式
    - プロトコル
    - 伝送方式

コンピュータネットワーク

コンピュータネットワーク

- どうして **線** だけじゃだめなの？
  - **線** だけで可能なこと、不可能なこと
    - 信号を送ること → 可
    - 情報を送ること → 不可
  - 情報を送るには、その情報をどうやって送るか予め決めておかなければならない
    - 誰に？
    - どういう風に？

→ **プロトコル**

コンピュータネットワーク

- 複数の相手への同時通信
  - デジタル情報の複製
  - 多重化
- 物理的に同じ線に繋がっていないホストへのデータの転送
  - 中継 (forwarding)

ディジタルとアナログ
情報量

デジタル？ アナログ？

- デジタル ？
  - digit: 指、アラビア数字（0～9）
  - digital: 指の、数字の、計数型の

  → 数値による表現
- アナログ?
  - analogy: 類似、相似
  - analog: 類似物、相似物

  → 類似・相似による表現

デジタルの特徴

- すべてを同じように扱える
  - 一つの方法が様々な情報に利用できる
- 数学の魔法
  - 暗号化
    - 数値をある決まりに従って置き換える
    - 決まり → 鍵
    - HALの例 (IBMから 1 引く)
  - 圧縮
    - 重複がある場合は、同じことを表現すればいい
    - テレビの碁盤の目

暗号化

- HAL
  - 2001年宇宙の旅に出てきたコンピュータ会社
  - 1997年1月12日設立 :-)
  - H = I - 1, A = B - 1, L = M - 1
  - IBM = {H,A,L} + 1
  - Caesar 暗号

圧縮

- テレビの画面の碁盤の目
  - 頻繁に動いているところとそうでないところ
  - 動いているところだけを考えればいい

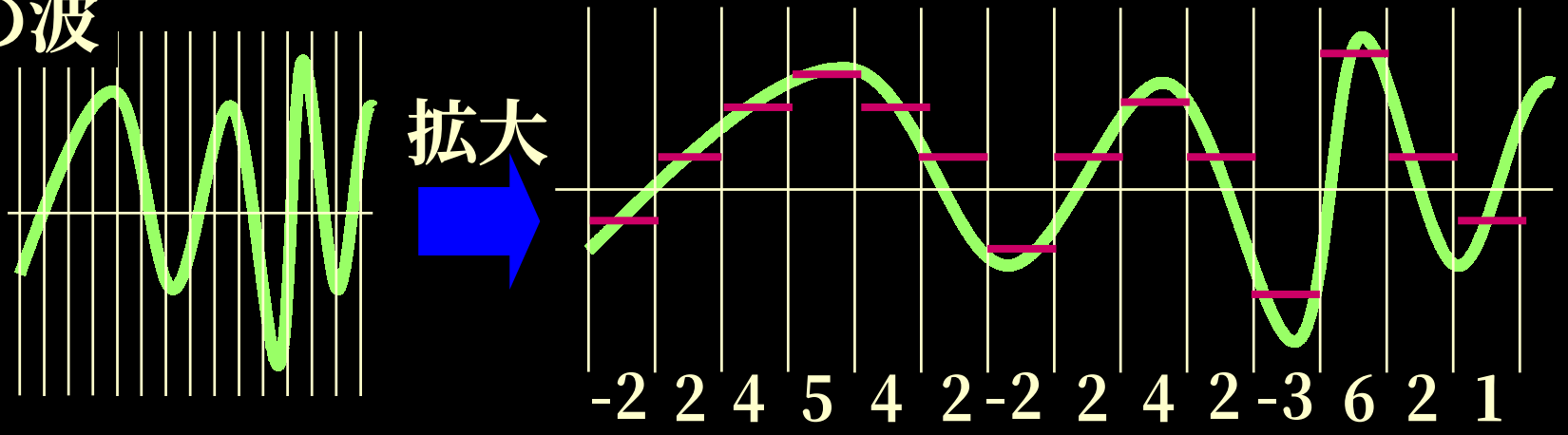
デジタル表現

- 文字
  - 数字を文字に割り当てる → 文字コード

A 0x41　　a 0x61

  - ASCII(American Standard Code for Information Interchange)
- 音
  - 音の波を一定の時間で区切り、波の大きさを数値化

音の波



デジタル情報の表記

- 0 と 1 を便宜的に使う(2進数)
  - 1bit = [0, 1]
    - 0か1しか入らない桁のこと
    - “Yes” or “No”
- 32bit = 0, 1の列が32個並ぶ

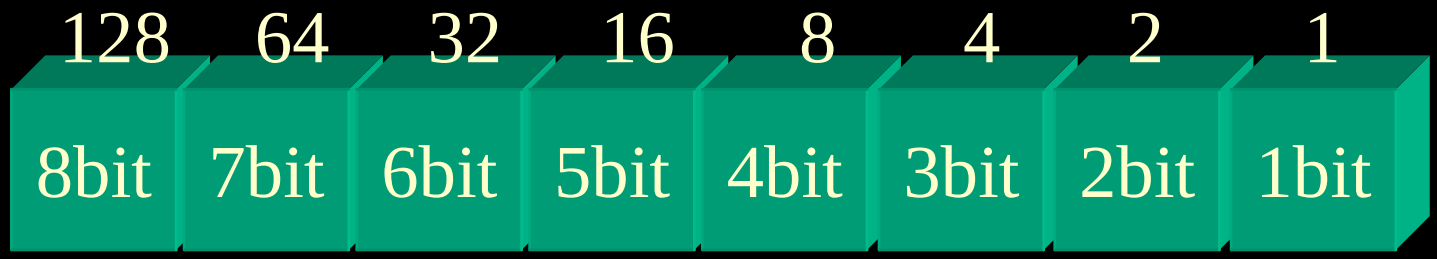
  例) 01101101011010110101011010111
- CD
  - 44.1KHz、16bit サンプリング
- 衛星放送
  - 32KHz or 48KHz、16bitサンプリング

Bit・Byte・Octet

- bit
  - 0または1だけが入る一桁
  - 1bit で区別できるのは $2^1 = 2$
- byte/octet
  - 1octet = 8bit
  - 8bit ≠ 1byte (但し、多くの場合は=)
  - 1octet で区別できるのは $2^8 = 256$
- K(キロ) / M(メガ)
  - 1K = 2^{10} = 1024　　1000 = 10^3
  - 1M = 2^{20} = 1048576　　1000000 = 10^6

2進数・2の冪乗

- 2進数
  - 一つの桁に0、または1だけが入る
  - 多くのコンピュータは内部で2進数を利用
- 2の冪乗
  - 32bit ではいくつの区別ができる？
  - $2^{32} = 2^{10} \times 2^{10} \times 2^{10} \times 2^{2}$　　$10^{3\times3} \times 4$　　40億
- 2進数から10進数
  - 上の箱に当てはめ、1の箱のところの数字を足す



CDの情報量

- 44.1KHzの間隔で16bitの幅のサンプリング
  - 44.1 (Hz)× 1000 × 16 (bit) = 705600 bit/sec
  - 約 74 分 = 74 × 60 (sec) = 4440 (sec)
  - これで計算すると 約373MB
- 実際にはエラー訂正用のデータが含まれる
- CD-ROM の容量は 約650MB

衛星放送の情報量

- 48KHzの間隔で16bitの幅のサンプリング
  - 48 (Hz)× 1000 × 16 (bit) ＝ 768000 bit/sec
  - １時間の番組では 60 × 60 (sec) ＝3600 (sec)
  - 329MB

伝送速度と時間

- CD-ROM 1枚 600MB を転送するには…
  - V.32 524288 (sec) 約146時間
  - ISDN Bチャネル 76800 (sec) 約21.3時間
  - T1 3200 （sec) 約53分
  - Ethernet 480 (sec) 8分
  - T3 106 (sec) 1.78分
  - OC-3 31 (sec)
  - OC-48 1.9 (sec)

ネットワークアーキテクチャ

ネットワークアーキテクチャ

- プロトコルって？
  - コミュニケーションを行うための決め事
  - 同じプロトコルしかコミュニケーションできない
  - peer to peer
- 階層構造　(Layering)
  - 役割分担を明確にする
    - 人のコミュニケーションの例

      思考 → 声帯 → 空気 → 鼓膜 → 理解
  - コミュニケーションの相手と同じ構造が必要

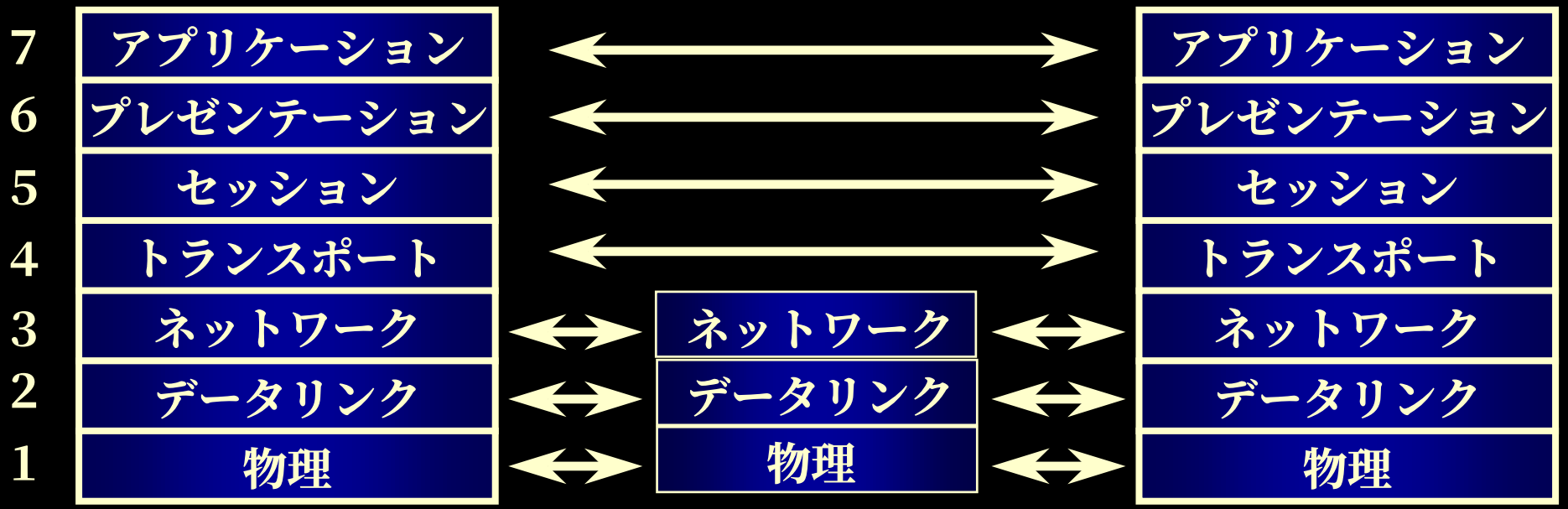
インターネットと鉄道網

- 客(データ)は必ず目的駅(コンピュータ)に到達する
- 駅には乗換え駅(ゲートウェイ)と普通の駅がある
- 各路線(ネットワーク)は自律運用、かつ、相互協調
- 選択可能な経路、経路の選択は知的な作業
- 鉄道網の「オーナー」はいない

「東京乗り換え案内」より

OSI 参照7層モデル

- ネットワークの世界の役割分担
- 思考と声帯が連携しているのと同じく　n層は　n+1層とn-1層と連係する
- n層のことは n層でしか理解できない
- コミュニケーションの間にあるシステムは第1層から第3層でリレーする



OSI参照7層モデルと鉄道モデル

名前	場所	役割	仕事
Application	世界	要求-要求	わがまま
Presentation			
Session	駅付近(切符)	客-客	客-窓口
Transport	改札付近	窓口-窓口	窓口-改札
Network	ホーム上	駅-駅	改札-ホーム
Datalink	「路線」	ホーム-ホーム	ホーム-電車
Physical	線路	車両・線路	車両・線路


物理層
データリンク

物理的な接続

- 物理層
  - 電気的な信号
- データリンク
  - 信号の出し方
  - 信号のフォーマット

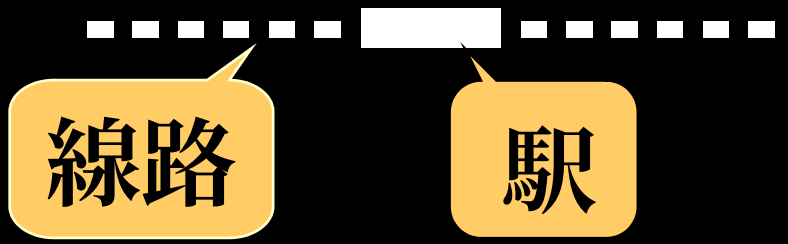
➡情報を伝えるために必要な技術

線路

- 物理的に敷かれたレール
  - 江ノ島から相模大野まで
- いろいろな種類がある
  - モノレール ： コンクリート
  - 電車 ： 鉄
  - リニアモーターカー ： 磁力？

物理層

- 鉄道モデルでいう線路が物理層
  - 物理的な接続
- ネットワークではケーブル・ファイバなど
  - Ethernet ― 同軸ケーブル、より対線
  - FDDI ― 光ファイバ
- 電気信号を通す

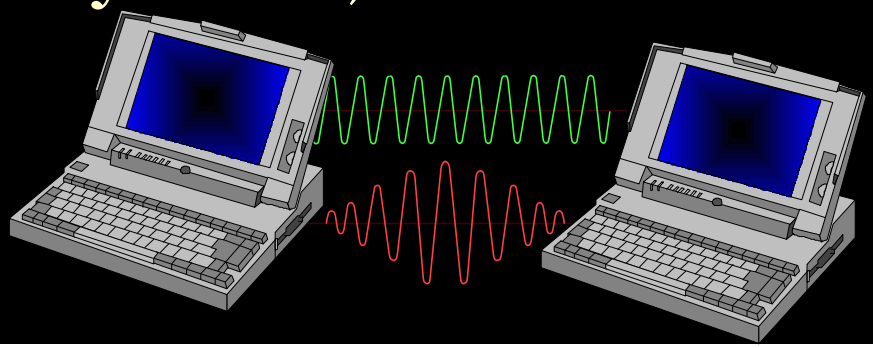


路線

- 線路上を走る電車
  - 人を運ぶ
- ホームからホームへ
  - 路線が違うとホームも異なる
  - 例）　山の手線と東海道線
- 同一路線では乗り降りは一度だけ
  - 例）：小田急江ノ島線
    - 藤沢 ― 湘南台

データリンク層

- 物理的に接続されているホスト間の通信
  - Ethernet, FDDIなど
- データをフレームという形で送る
  - 決まったフォーマット
  - ＭＡＣアドレス
  - 上位層タイプ
  - CRC（Cyclic Redundancy Check)



既存のネットワーク技術

- Ethernet(同軸ケーブル，より対線　など)
- FDDI/TPDDI(光ファイバ，より対線 など)
- ATM(光ファイバ)
- 無線LAN（2.4GHz帯の電波など）

新しいネットワークメディア

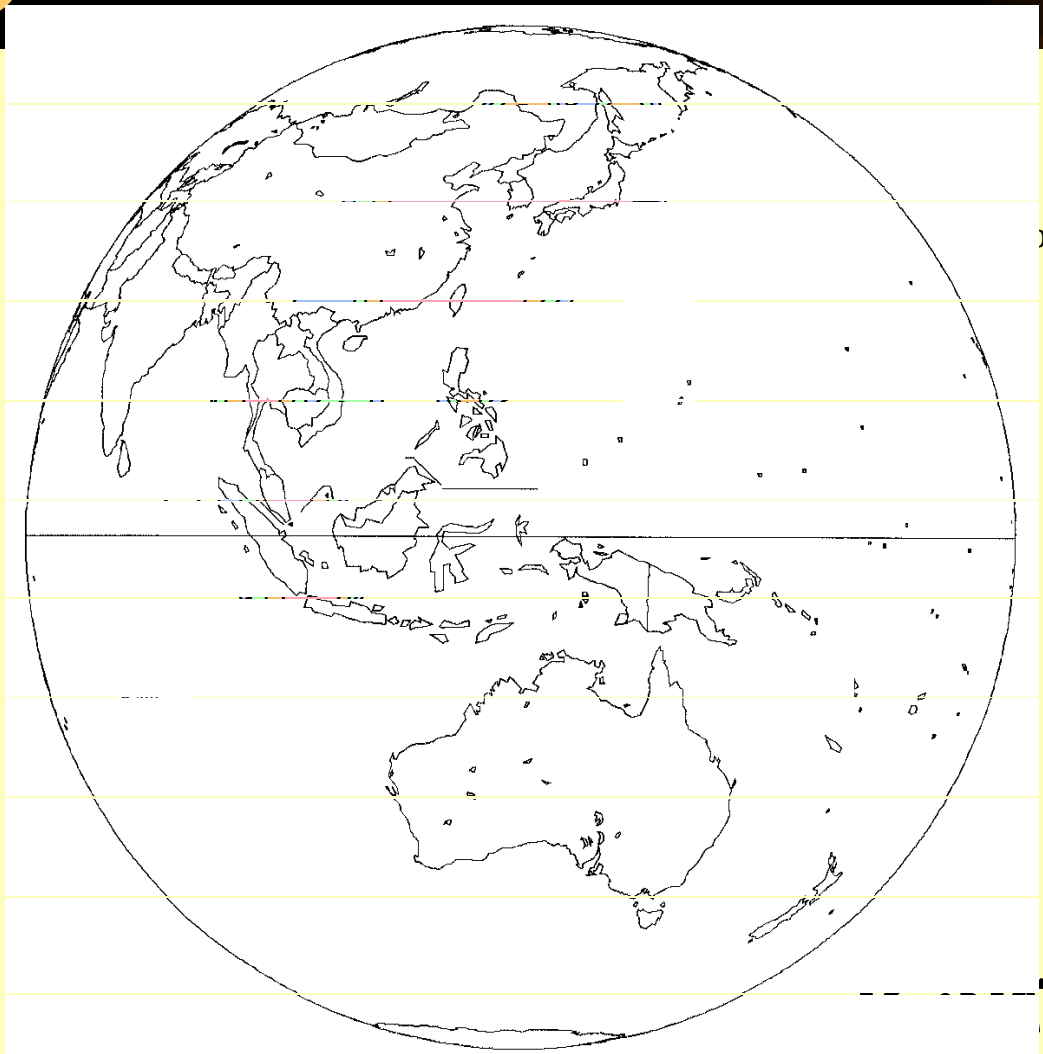
- 衛星通信
  - VSAT
  - 1.8mのアンテナ
  - 2Mbps
  - 500msの遅延
- CATV
  - 専用のチューナ
  - Up-Link 2Mbps, Down-Link 30Mbps
  - 同軸ケーブル

WISH-Internet with Satellite Harmonization

- 2Mbps X 7
- 75cm to 1.8m dishes
- Domestic foot print
- Integrated with the terrestrial testbed
- Multicast Internet development

WISH domestic topology

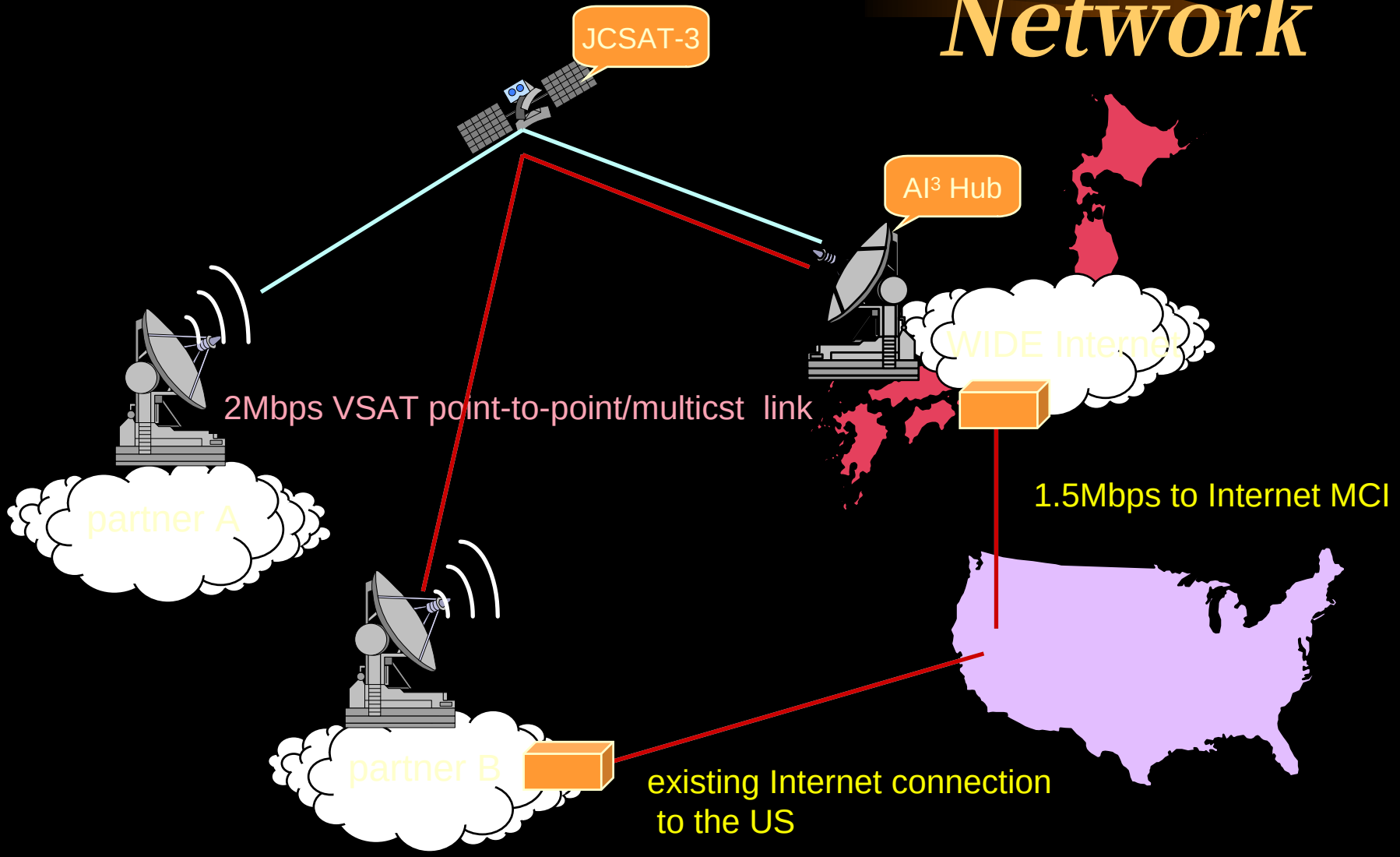
JCSAT-3 Asian Zone Beam



:3.6m

T

Topology of AI³ Testbed Network



CATV/有線放送

- 同軸ケーブルによる物理的な接続
- 帯域： 2Mbps～30Mbps
- 既存の設備（ネットワーク）を利用
- 東急ケーブルテレビが来年度よりサービスを開始
  - 月額1,800円
- 大阪有線が実験を開始
  - 参考：
    - 初期費用：30,000円
    - 月額：6,000円

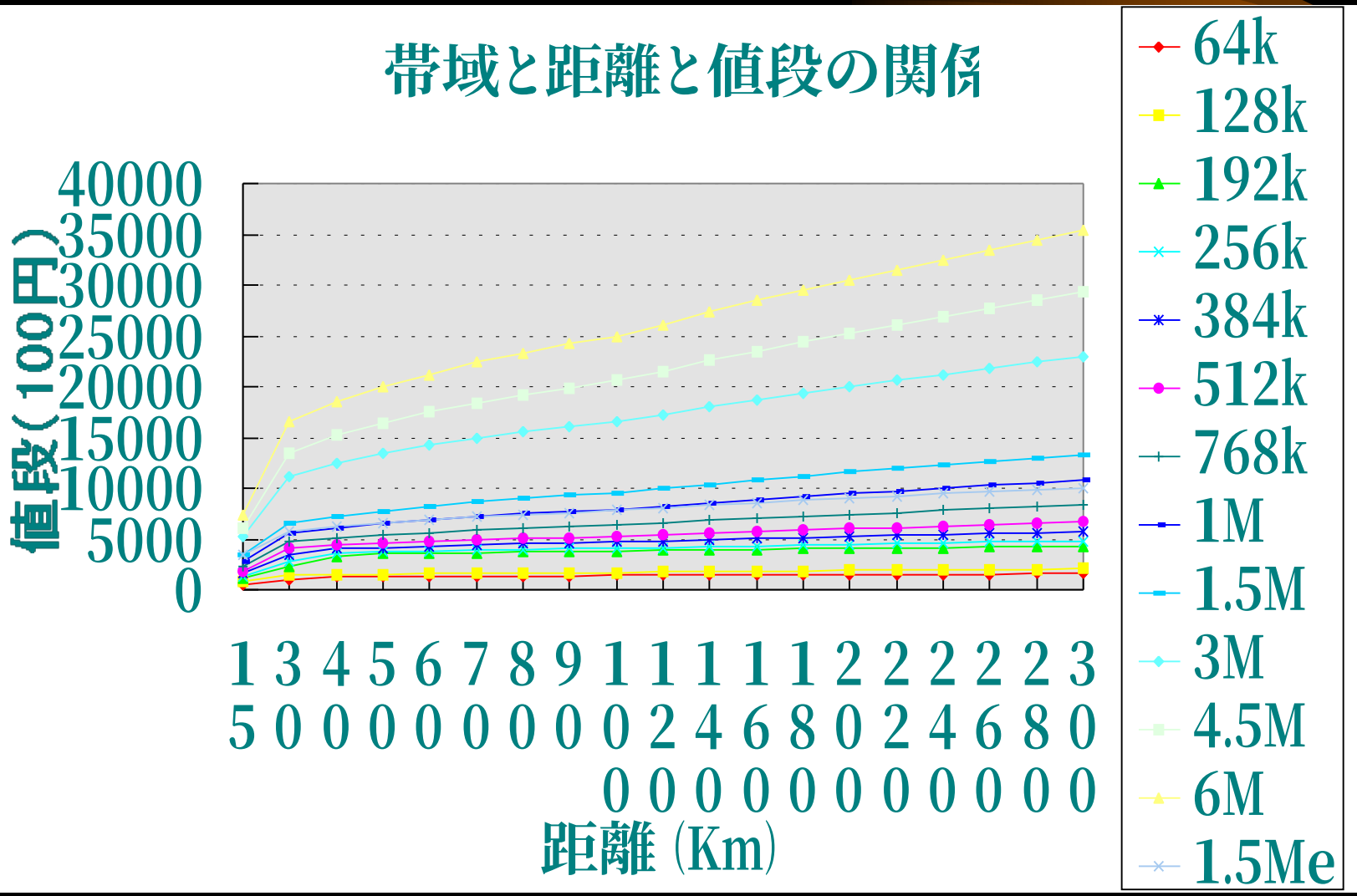
ネットワークの太さ

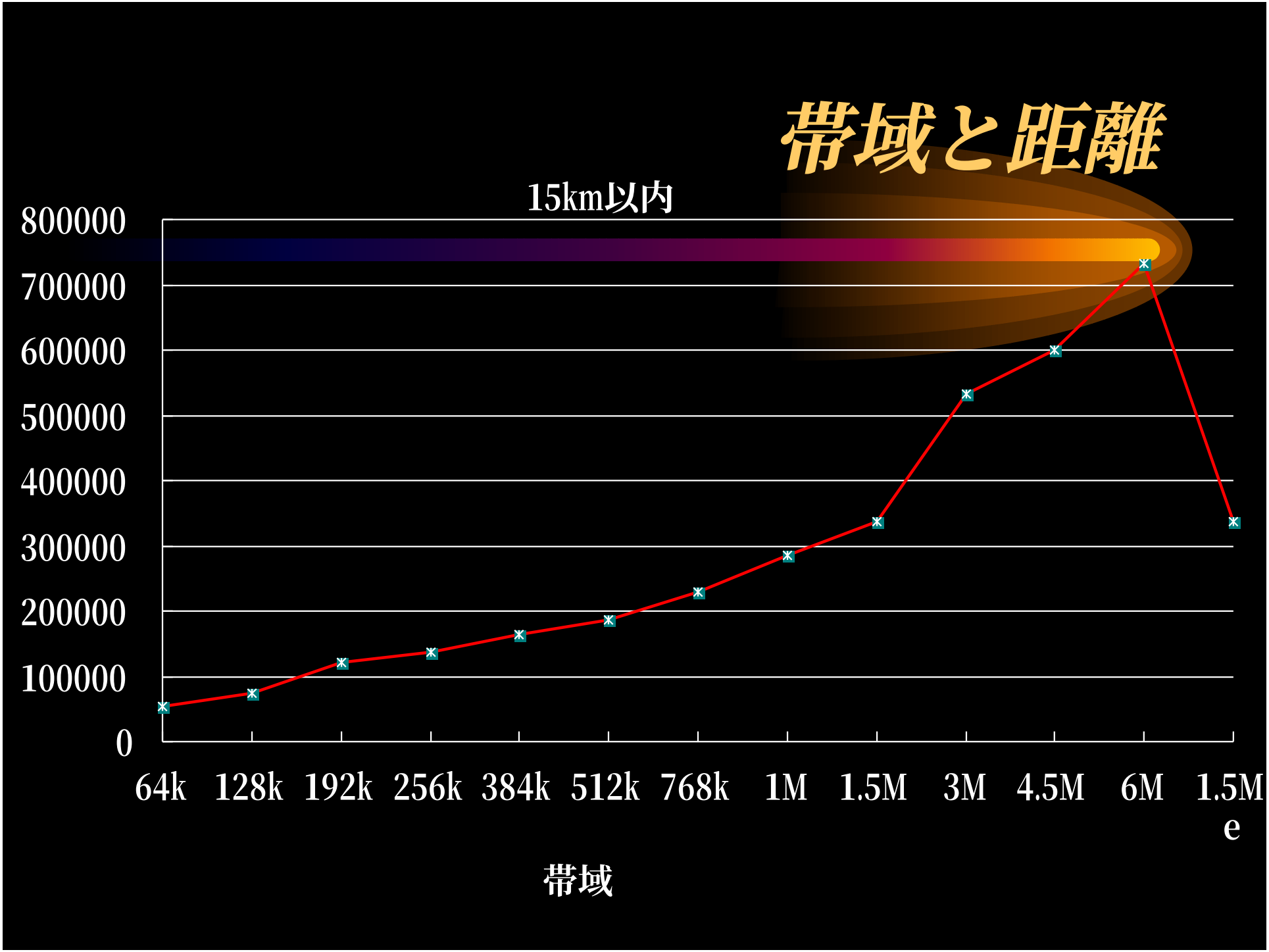
- 帯域
  - 単位時間当たりに送れるデータ量
  - bps(Bit Per Second)
- 一般に帯域が大きい(太い)回線ほどコストが高い

帯域と距離

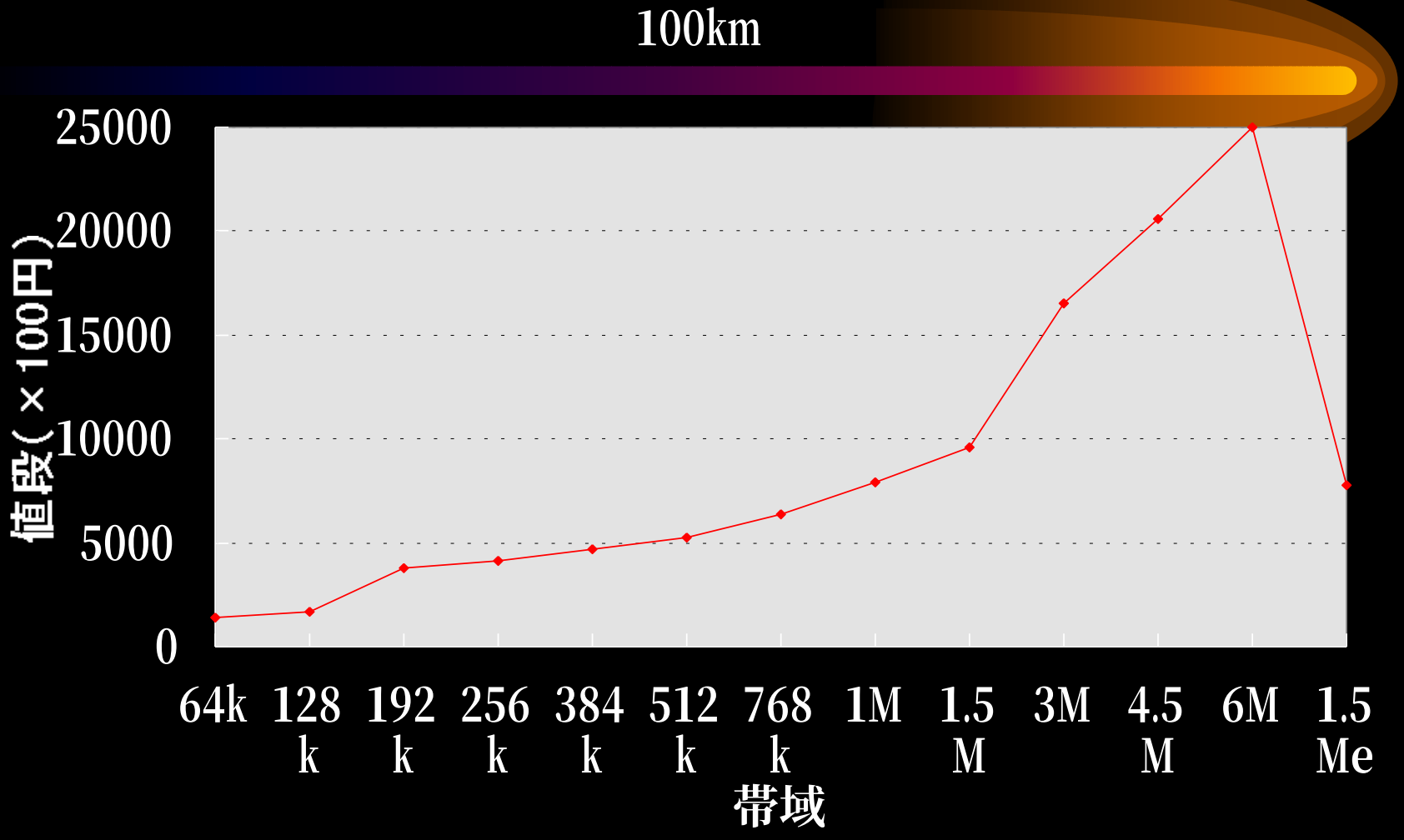
- 帯域
  - ネットワークを流せるデータ量
  - メディアによって帯域が異なる
    - 太い・細い
- 距離
  - 通信する2地点間の距離
- コスト
  - 帯域の大きさ
  - 距離の遠さ

帯域と距離

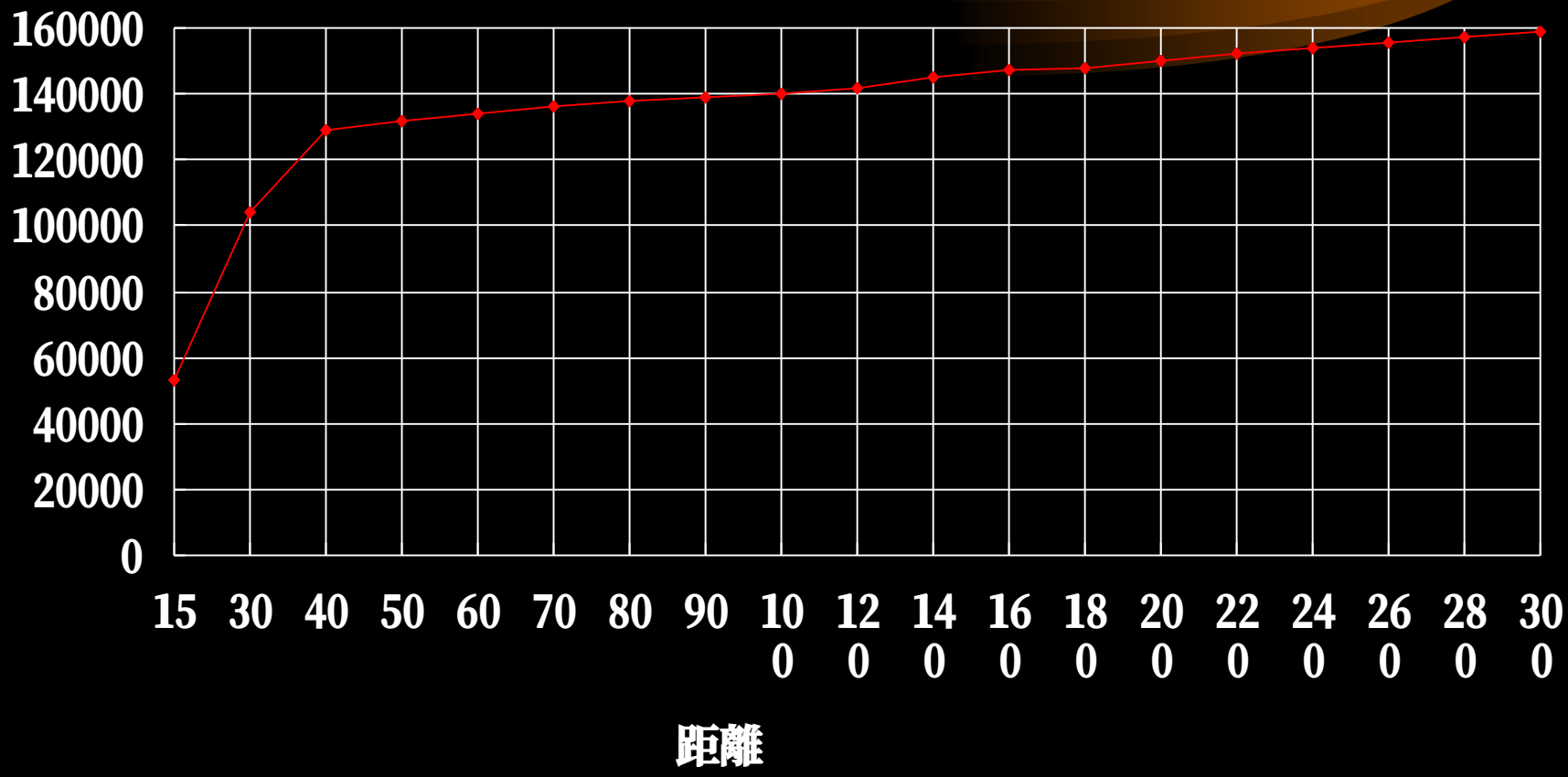

帯域と距離と値段の関係
値段(100円)
40000
35000
30000
25000
20000
15000
10000
5000
0
15
30
40
50
60
70
80
90
100
120
140
160
180
200
220
240
260
280
300
距離 (Km)
64k
128k
192k
256k
384k
512k
768k
1M
1.5M
3M
4.5M
6M
1.5Me


帯域と距離
15km以内
800000
700000
600000
500000
400000
300000
200000
100000
0
64k
128k
192k
256k
384k
512k
768k
1M
1.5M
3M
4.5M
6M
1.5M
e
帯域

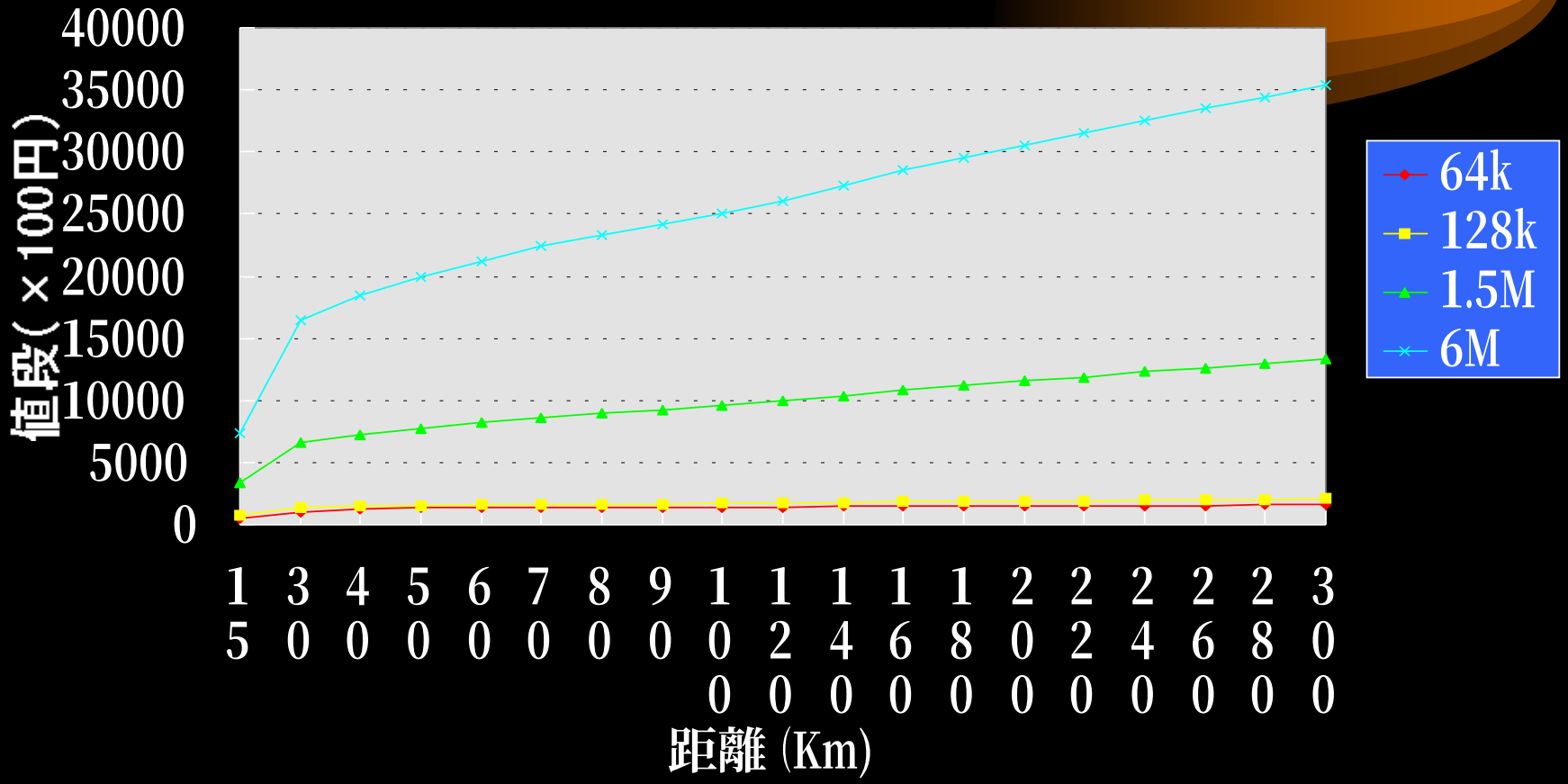
帯域と距離


100km
25000
20000
15000
10000
5000
0
値段(×100円)
64k
128k
192k
256k
384k
512k
768k
1M
1.5M
3M
4.5M
6M
1.5Me
帯域

帯域と距離


値段
64k
160000
140000
120000
100000
80000
60000
40000
20000
0
15
30
40
50
60
70
80
90
100
120
140
160
180
200
220
240
260
280
300
距離

帯域と距離


各帯域と距離と値段
値段(×100円)
40000
35000
30000
25000
20000
15000
10000
5000
0
15
30
40
50
60
70
80
90
100
120
140
160
180
200
220
240
260
280
300
距離 (Km)
64k
128k
1.5M
6M

CSMA/CD

- みんな耳を立てている
- しゃべりたい時は相手の名前をデータに付ける
- 同時に誰かが喋りだしたら話すのを止める
- さいころを振って出た目だけ待ってもう一度離す
- Carrier Sense, Multiple Access, Corigion Ditection

ARP (Address Resolution Protocol)

- Ethernetアドレス
  - Media Access Controlアドレス
  - 48bit
  - 隣のホストを指定する
  - IPの処理部分が利用するプロトコル

  - IPアドレスの宛先が…
    - 同じネットワーク内
      - MACアドレスは同一ホストを指定
    - 異なるネットワーク
      - MACアドレスはルータを指定

ARP
動作と原理

- 隣のホストのリスト
  - ARPテーブル
  - IPアドレスとMACアドレスの対
- リストの作成
  - 「このIPアドレスを持っている人はいますか?」
    - IPアドレスを指定
    - 全員に対して送る(Broadcast Message)
  - 「そのIPアドレスは私が持っています」
    - 送り主に対して送る(Uni-cast Message)

インターネットと信頼性

- 信頼性とは?
  - Flow Control
  - Congestion Control
  - Re-Transmission
  - CRC
  - Checksum

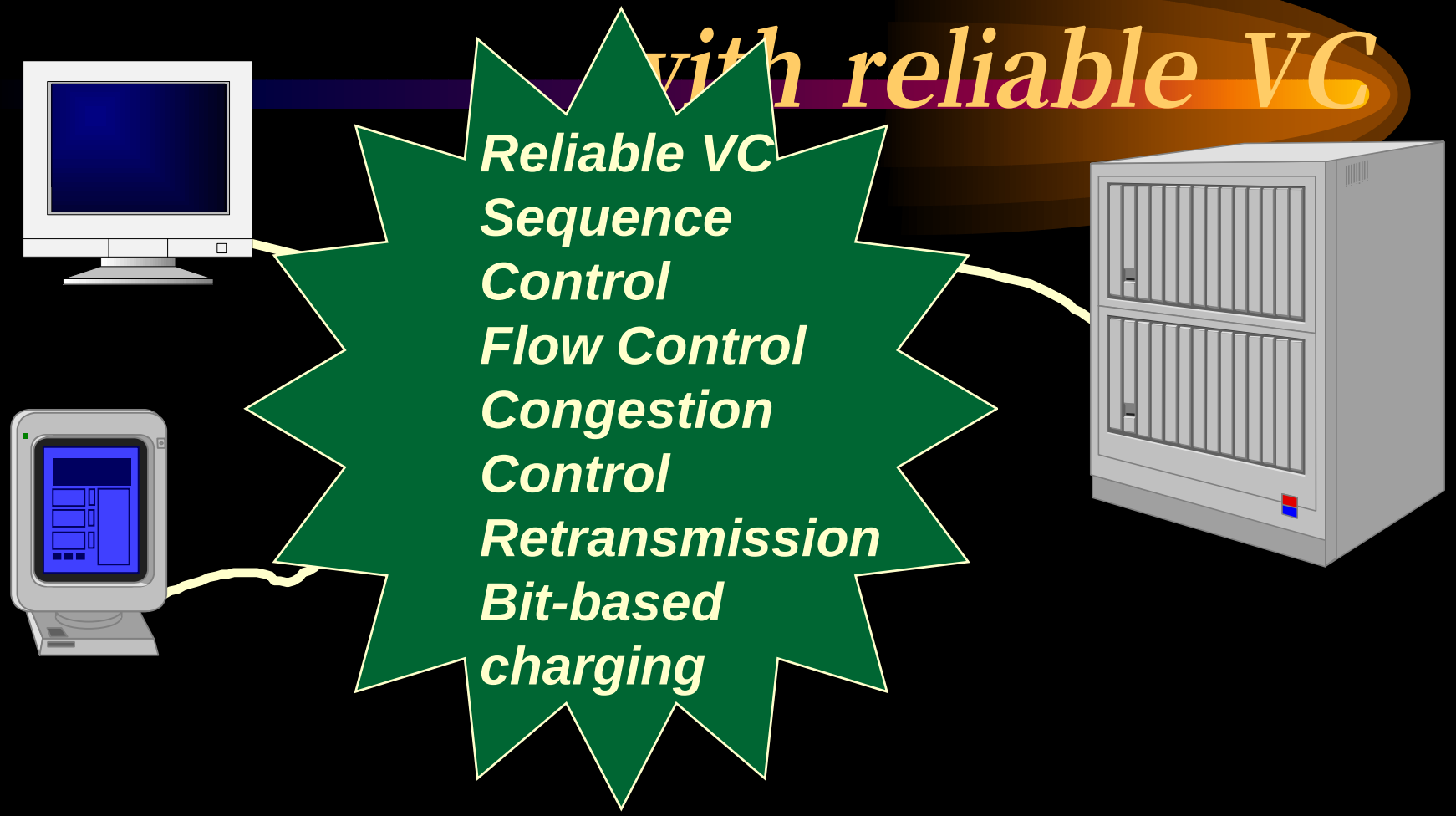
通信形態

- バーチャルサーキット型
  - 仮想的なパイプを通した転送
  - コネクション型
- データグラム型
  - データを小分けにし、バラバラに転送
  - コネクションレス型

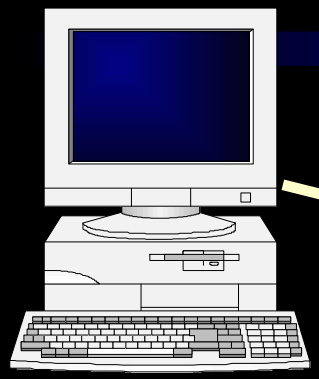
バーチャルサーキット vs データグラム

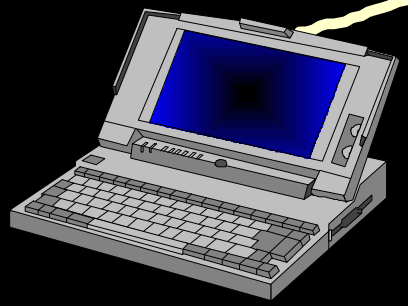
- FAX vs 手紙
  - 紙に書かれた文書の転送
  - 手紙の通信
    - 送信者がポストに入れた順番では届かない
      - 送信者が送った順番と受信者が受け取った順番は異なる
    - 送り手の動作
      - 単純
      - 宛先を指定
      - ポストに投函
  - FAXの通信
    - 送信者がFAXに入れた順番に文書が受信される
      - 送信者が送った順番と受信者が受け取った順番は同じ
    - 送り手の動作
      - 複雑
      - 相手を指定
      - FAX間の初期化
      - データの転送
      - 終了操作

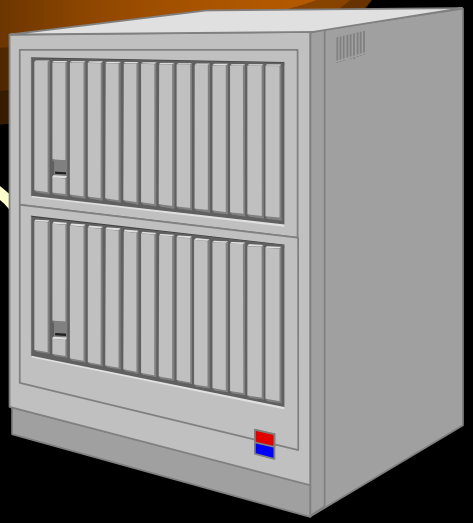
Data Communication Network with reliable VC



Internet
on unreliable DG

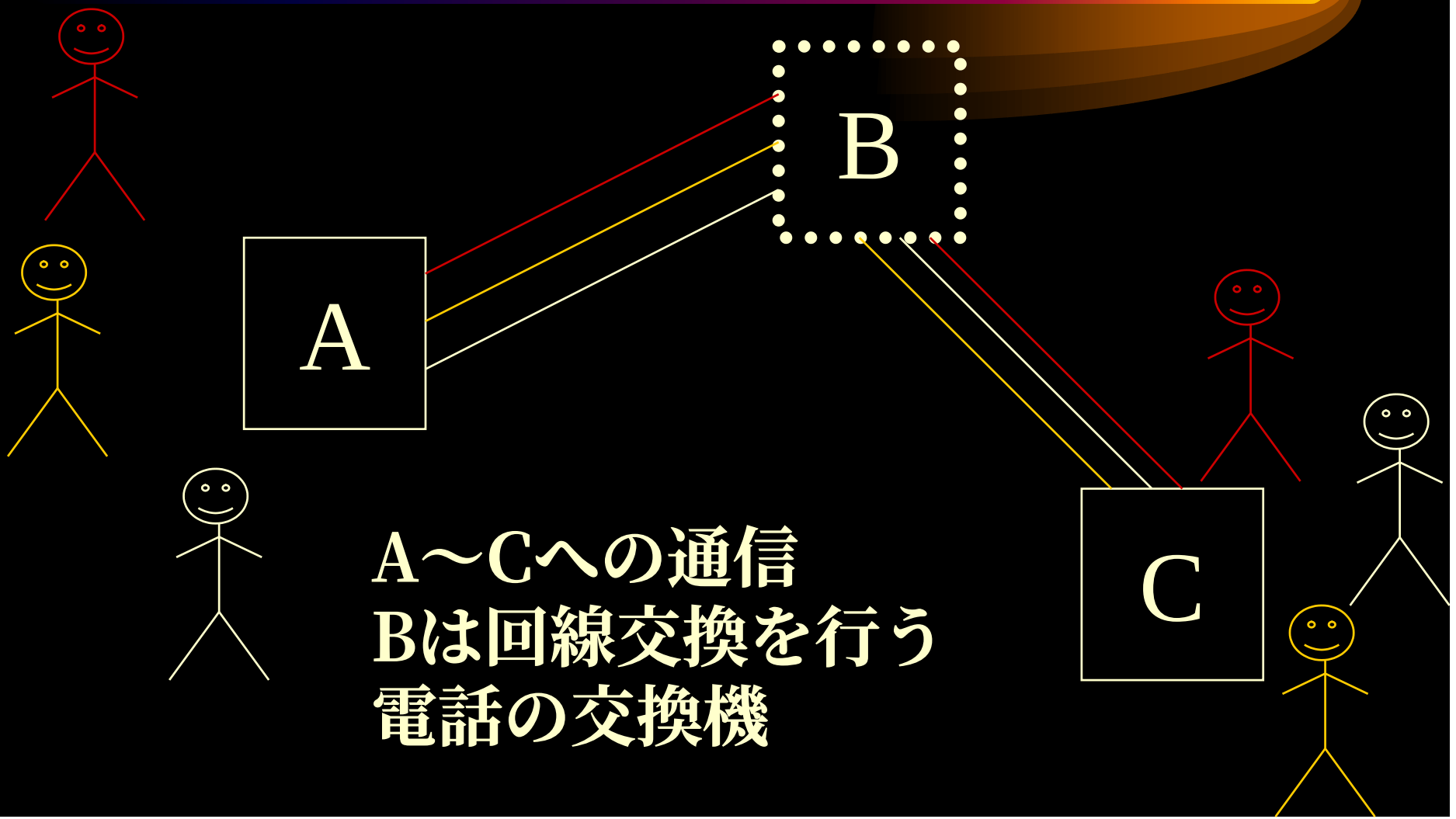






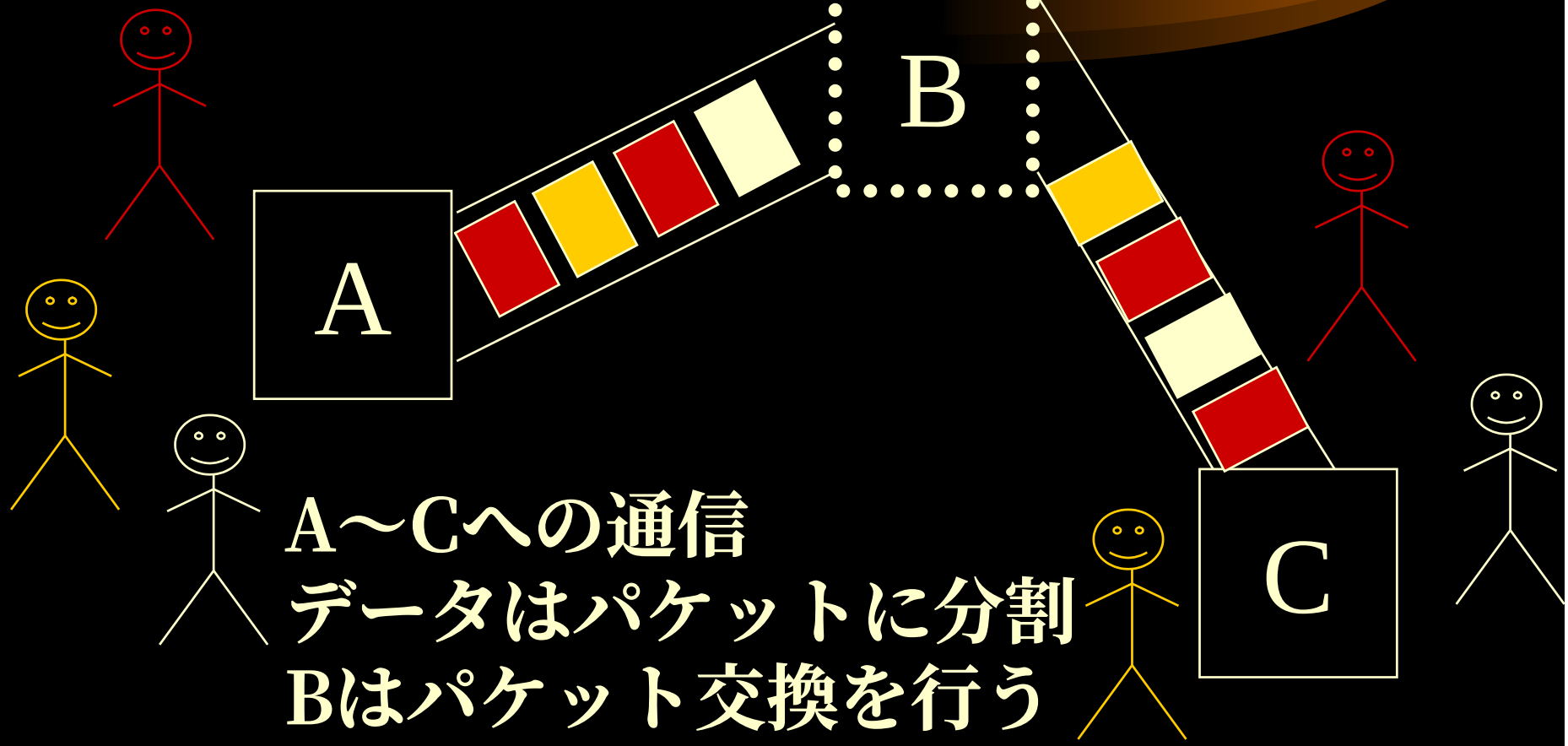
Unreliable Datagram
No Sequence Control
No Flow Control
No Congestion Control
No Retransmission
Fixed-rate charging

回線交換方式



A～Cへの通信
Bは回線交換を行う
電話の交換機

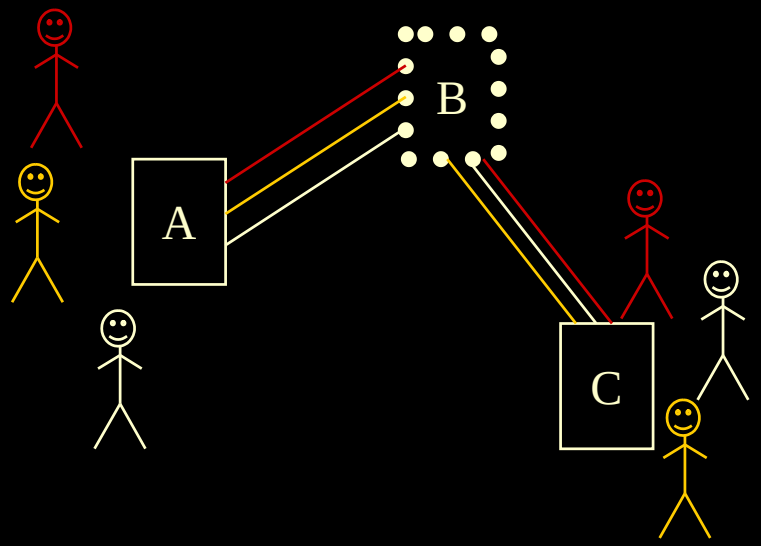
パケット交換方式



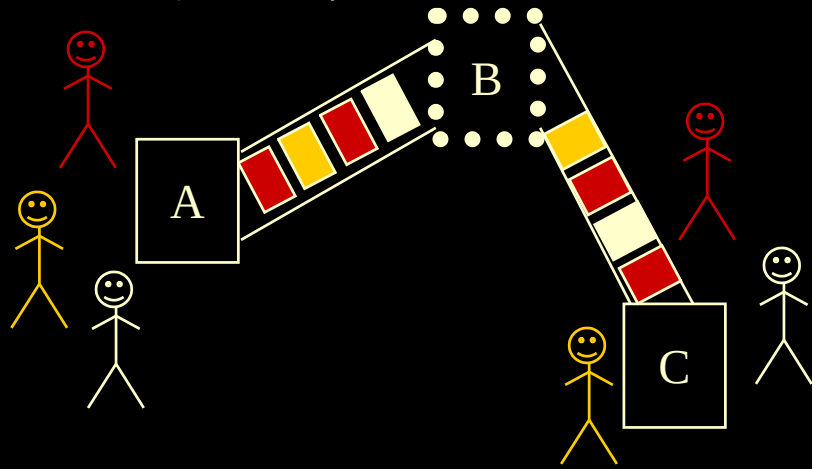
A～Cへの通信
データはパケットに分割
Bはパケット交換を行う

データ転送の中継方式

- 回線交換方式
  - 2点間を回線で結ぶ
  - バーチャルサーキット
  - 電話システム



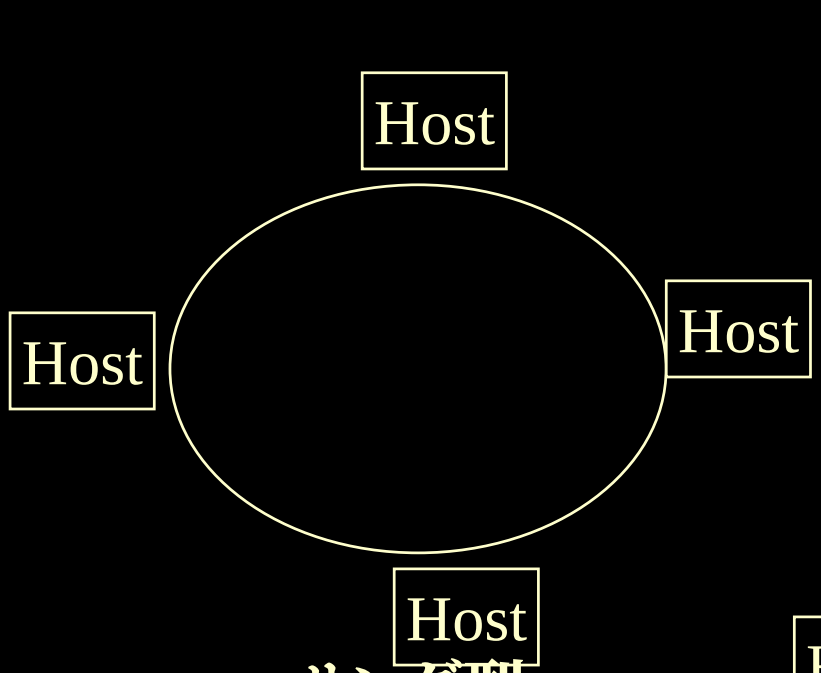
- パケット交換方式
  - 2点間をパケットで結ぶ
  - データグラム
  - インターネット



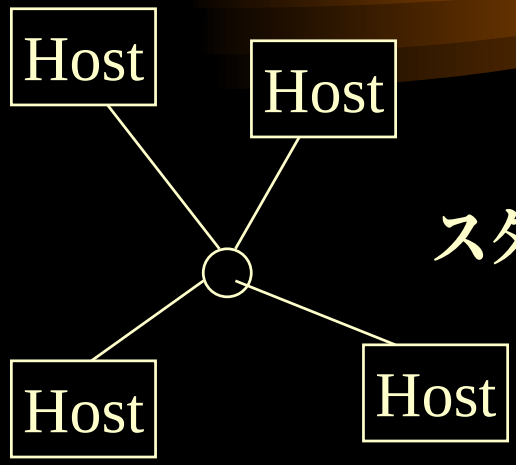
ネットワークの構成

- LAN
  - スター型，バス型，リング型
- LAN間接続
  - ポイントーポイント接続

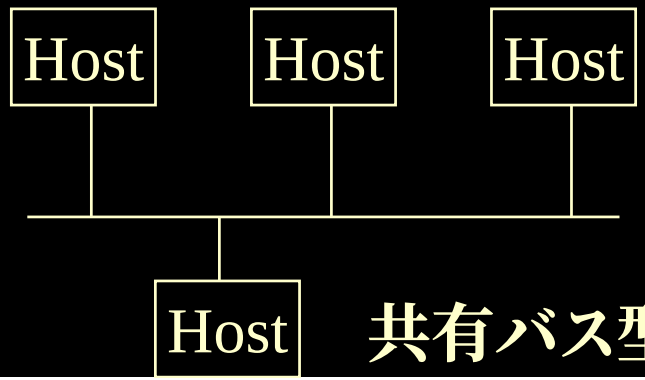
ローカルネットワークの接続形態



リング型



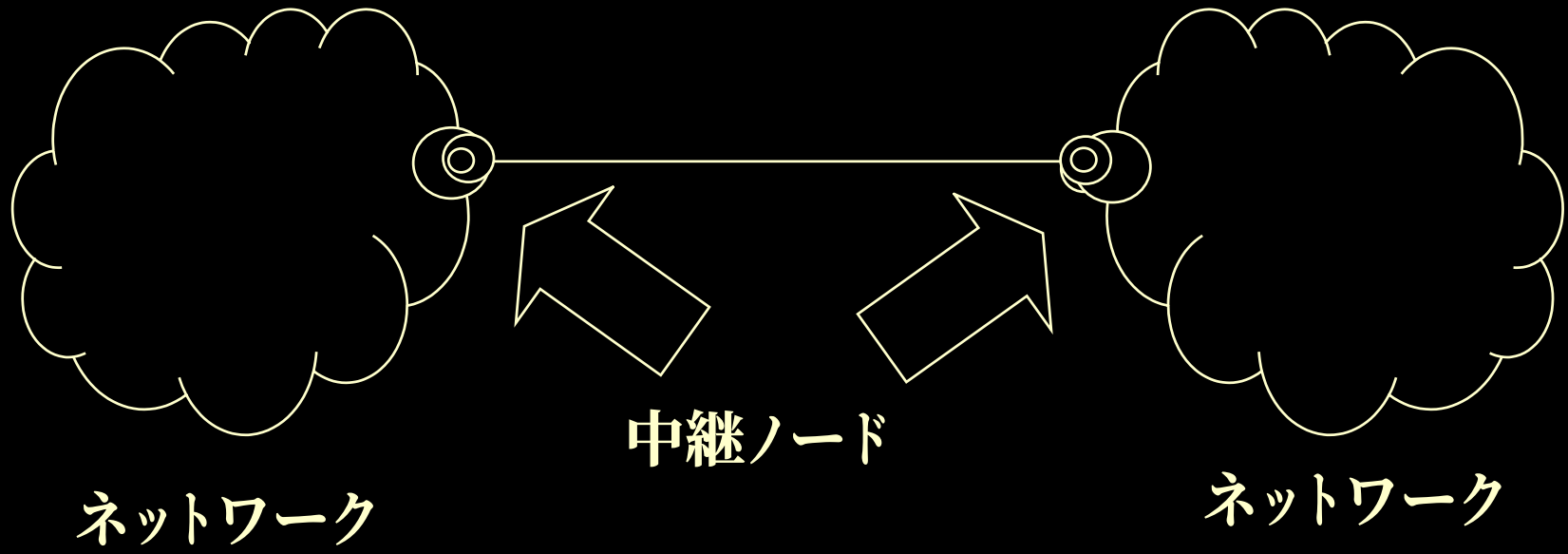
スター型



共有バス型

ポイント — ポイント型接続

1対1の接続方法，2地点間接続



IP

IP ～機能と特徴～

- 機能
  - ホストの識別
    - 32bit 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(インターネットプロトコル)

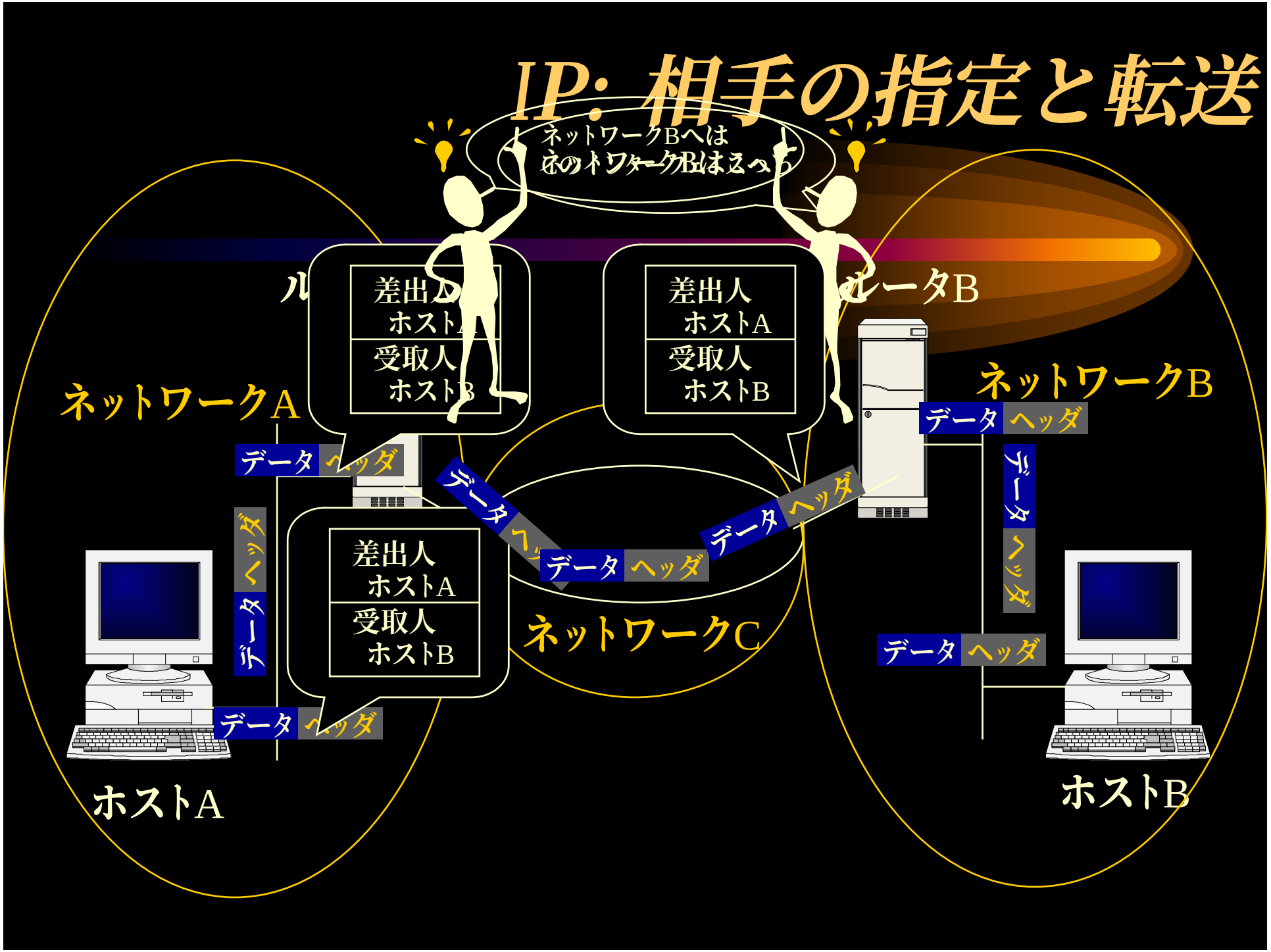
- インターネットの最も基礎となるプロトコル
- OSI参照モデルの第３層
- インターネットプロトコルの役割
  - ネットワークやホストの識別（Addressing:アドレス）
  - データ配送の経路の決定（Routing:経路制御）
  - 別のネットワークへのデータの転送（Forwarding)
  - データの分割（Fragmentation)

IP(インターネットプロトコル) 続き

- データグラム型の通信
  - 信頼性がない
  - 仕組みが単純
- Best Efforts(最善努力型)
  - エラーの訂正は行わないが、エラーの報告は行う
  - ICMP (Internet Control Message Protocol)

乗換駅

- 乗換駅とは?
  - 鉄道の路線と路線が交わるところ
  - 例： 藤沢駅
    - ＪＲ東海道線
    - 小田急江ノ島線
    - 江ノ島電鉄
  - 客の行き先で路線を選ばなくてはならない
  - 例： 湘南台から品川
    - 湘南台： 藤沢に行く
    - 藤沢： 東海道線に乗り換え


IP: 相手の指定と転送
ネットワークBへは
この ネットワークBは どっち
差出人
ホストA
受取人
ホストB
差出人
ホストA
受取人
ホストB
ルータB
ネットワークA
ネットワークB
ネットワークC
データ
ヘッダ
ホストA
ホストB

経路制御(Routing)

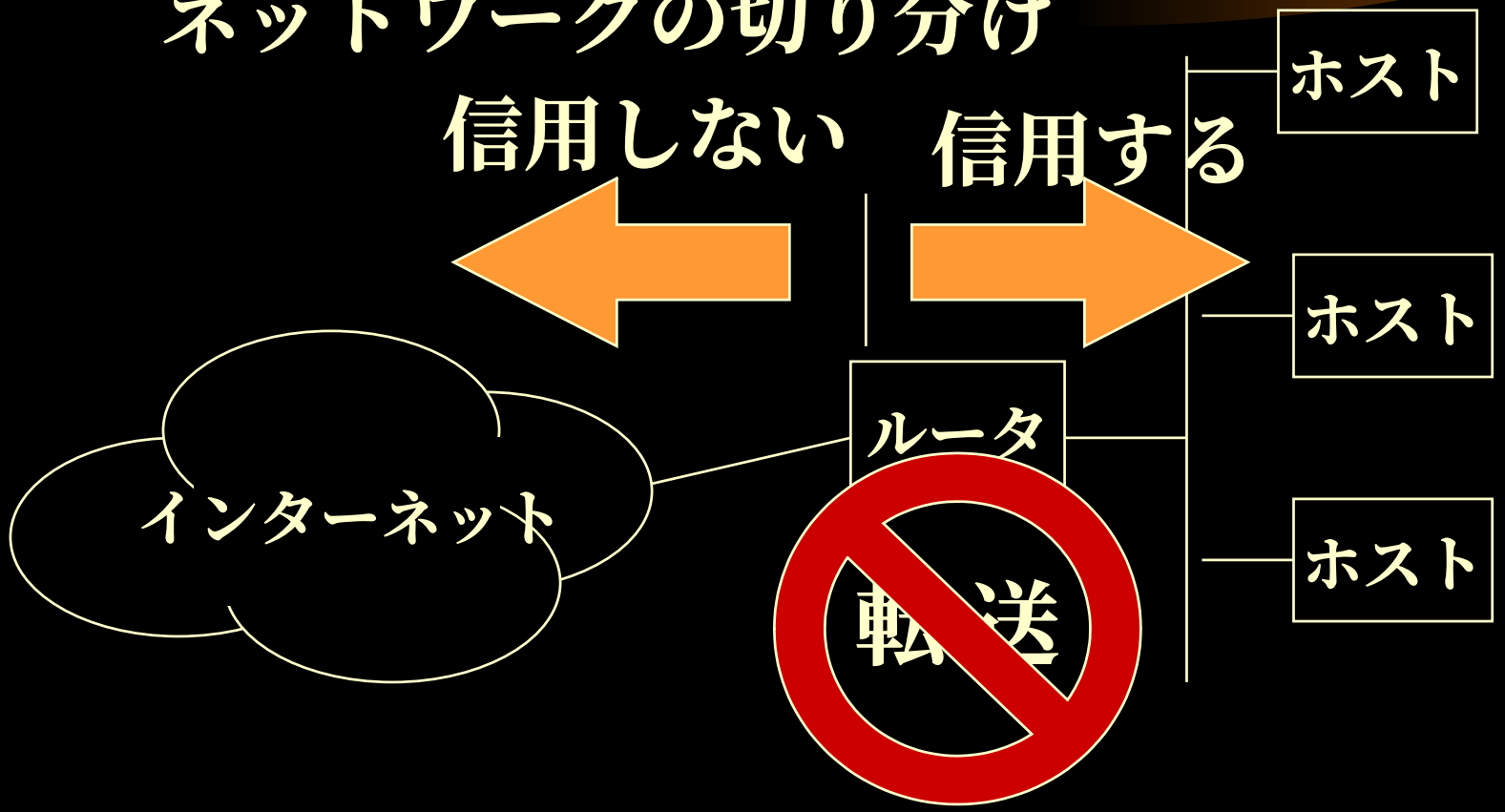
- 鉄道モデルの乗換駅 ＝ ルータ(Router)
- ルータは複数の路線に接続している
  - 藤沢駅
    - 小田急江ノ島線
    - JR東海道線
    - 江ノ島電鉄
- 行き先に応じて違う路線に乗せ換える
- ルータは複数のネットワークに接続

経路制御（Routing）その２

- どの路線（経路）に乗せればいいか？
  - 目的のホストがどの経路に接続してるか？
- 経路制御表
  - 経路の選択、制御に用いる
  - どの路線にはどのホストが接続しているという情報（経路情報）をまとめた表
  - すべてのホストはまとめきれない
    - IPアドレスのネットワーク部
    - default という経路

ファイアウォール

- 信用するネットワークと信用できないネットワークの切り分け



インターネット構築

- ネットワーク層
  - ホスト・ホスト間の通信の提供
  - 中間システムは転送のみを行う
  - エンドシステムはなんでも行う
- データリンク層
  - LAN内の通信
  - 電気信号的な接続

ルータとホスト

- ルータ(Router)
  - 中間システム(Immediate System)
  - パケット交換を専門にする
  - ネットワーク・ネットワーク間を結ぶ
- ホスト(Host)
  - エンドシステム(End System)
  - パケットの交換は行わない
  - 主にサービスの利用と提供

IPアドレス

- 32bitのアドレス空間
  - 32bitだといくつ識別できる？
  - インターネットに参加するためのID
  - 重複してはいけない
  - 通信対象を識別する
- 構造化されたアドレス
  - ネットワークを表すネットワークアドレスとホストを表すホストアドレスから成り立つ
  - ネットマスクによる柔軟な構造

IP: 相手の指定
Addressing

駅は路線名と組み合わせると一意に決まる

鉄道の例

小田急線　湘南台駅
JR山手線　恵比寿駅

路線の名前　駅の名前

ネットワークとホストを別々に識別すれば良い

ネットワーク部とホスト部

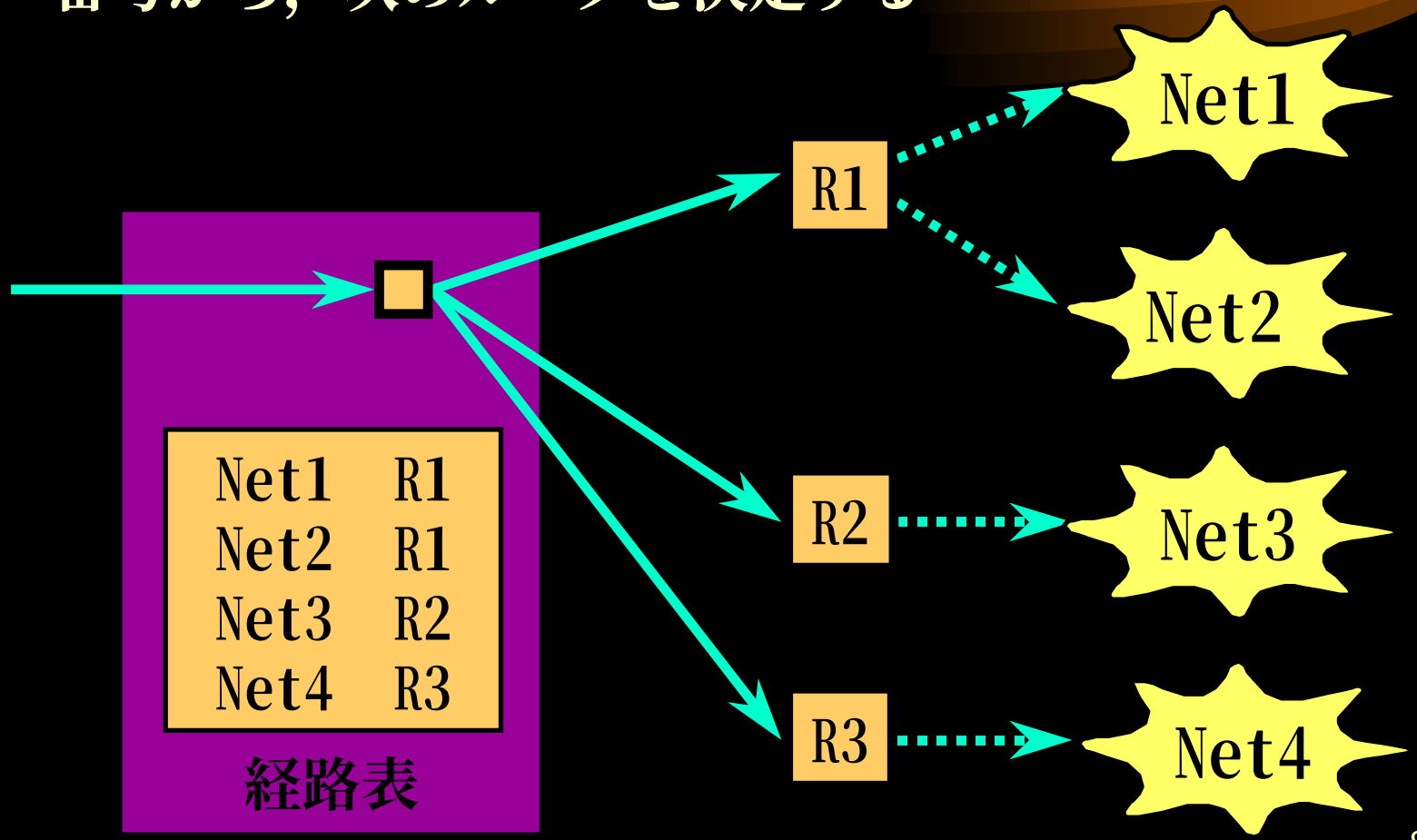
- 鉄道網の例
  - ○○線
    - 路線を表す部分
  - ××駅
    - 路線の中の駅を表す部分

山手線新宿駅

小田急線新宿駅

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



ネットワーク部とホスト部

- IPアドレスの構造化
- ネットワークの形態(トポロジ)を表す
- ネットワーク部
  - 世界中でネットワークを一意に表す
- ホスト部
  - ネットワークの中でホストを一意に表す
- サブネットマスク
  - ネットワーク部とホスト部の切れ目を表す

名前とIPアドレス

- コンピュータは数字を扱う方が得意
  - コンピュータはアドレスが分ればＯＫ
- 人間は数字を扱うのは得意ではない
  - 数字より名前の方が扱いやすい
- 名前
  - ホストの名前　(例：　mail0)
  - 組織の名前　　（例：　sfc.keio.ac.jp)

インターネット上の名前の取扱い

ドメイン名

- sfc.keio.ac.jp
  - jp (国を示す)
    - ISO3166 で決められている国名
    - 例外がある (アメリカ:com, edu/イギリス: uk)
  - ac （組織の属性・性質）
    - 学術研究機関（academic）
  - keio （組織の名称）
    - 慶應義塾大学
  - sfc (組織内のサブドメイン)
    - 湘南藤沢キャンパス (Shonan Fujisawa Campus)

DNS(Domain Name System)

- ホスト名
  - ホストの名前と組織の名前からなる名前
  - 例： mail0．sfc．keio．ac．jp
- ホスト名からIPアドレスを検索する
  - ホスト名とIPアドレスのデータベース
  - 世界中から検索できなければならない
- 構造化されたデータベース
  - ドメイン
  - サブドメイン

名前の構造化

- 電話番号

0466ー47ー5111

市外局番　市内局番　加入者番号

- 住所

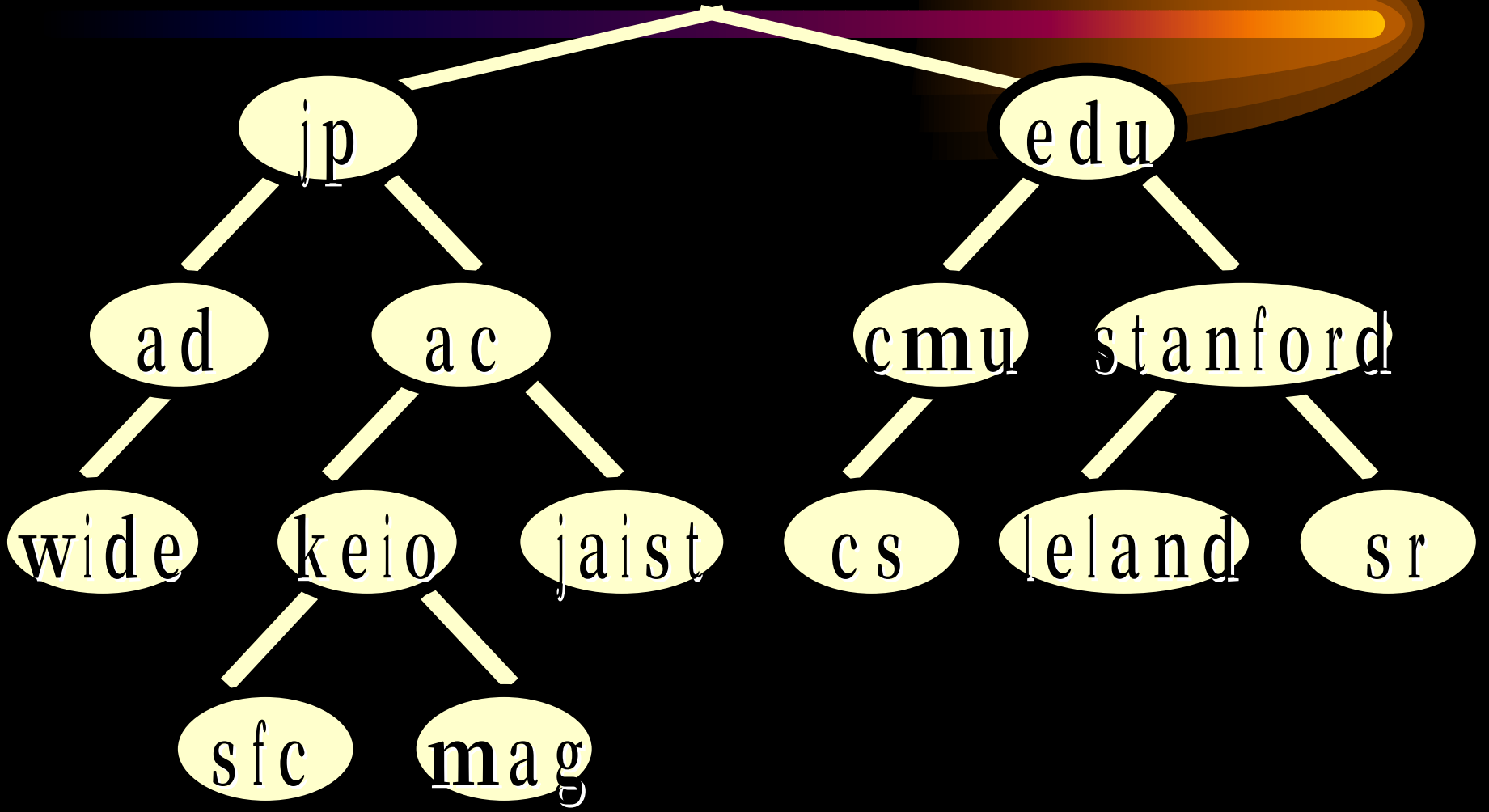
神奈川県　藤沢市　遠藤　5322

都道府県　市町村　地区　番地

DNS(Domain Name System)

- ホスト名
  - ホストの名前と組織の名前からなる名前
  - 例： mail0.sfc.keio.ac.jp
- ホスト名からIPアドレスを検索する
  - ホスト名とIPアドレスのデータベース
  - 世界中から検索できなければならない
- 構造化されたデータベース
  - ドメイン
  - サブドメイン

DNSの構造



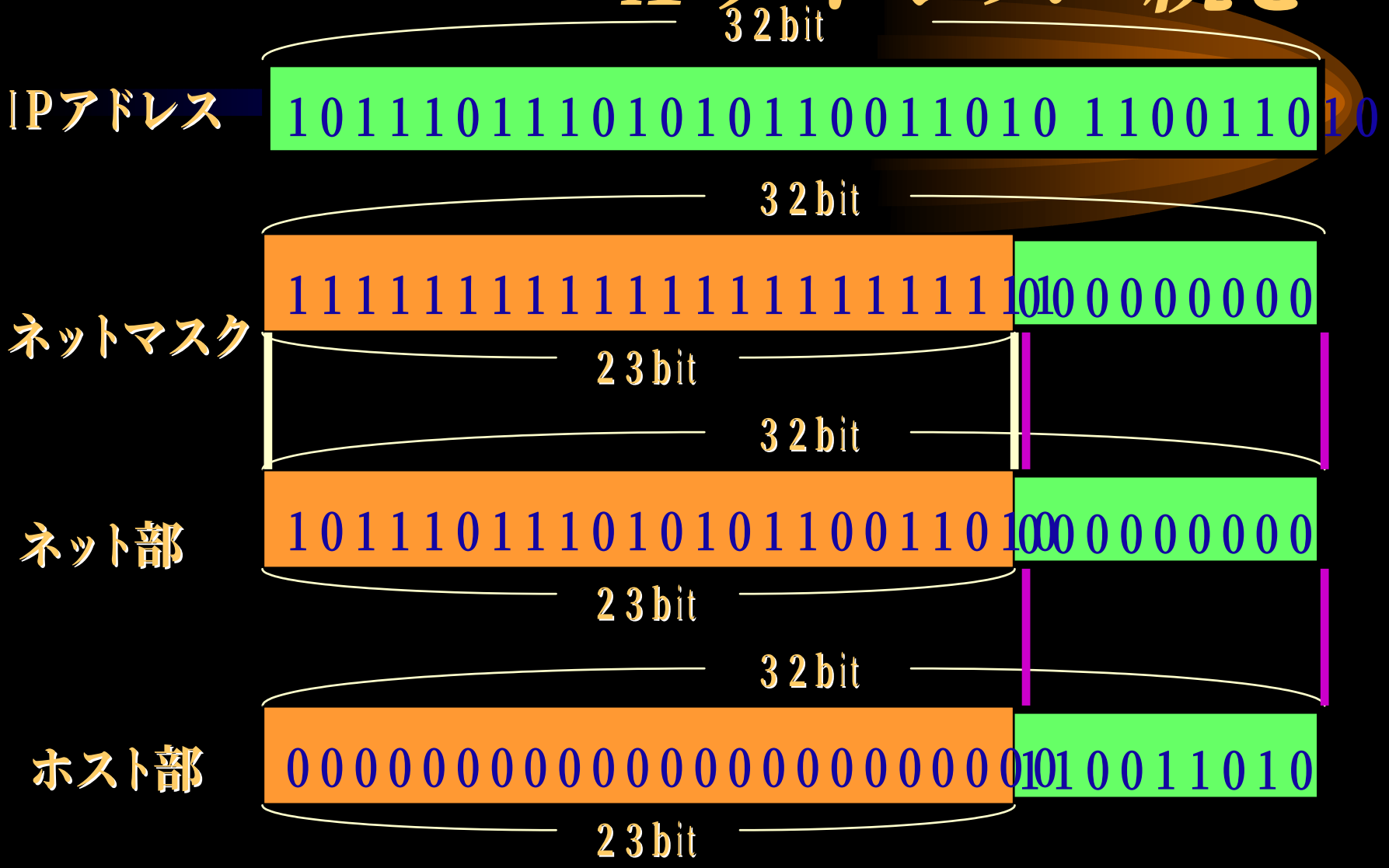
名前とIPアドレス

- コンピュータは数字を扱う方が得意
  - コンピュータはアドレスが分ればＯＫ
- 人間は数字を扱うのは得意ではない
  - 数字より名前の方が扱いやすい
- 名前
  - ホストの名前　(例：　mail0)
  - 組織の名前　　（例：　sfc.keio.ac.jp)

IPアドレス

- 32bitのアドレス空間
- インターネットに参加するためのID
  - 世界中で唯一自分を証明するためのもの
  - 重複してはいけない
  - 自分と相手を認識する
  - 1つのインターフェイスに1つのアドレス
- 構造化されたアドレス
  - ネットワークを表すネットワークアドレスとホストを表すホストアドレスから成り立つ
  - ネットマスクによる柔軟な構造

IPアドレス　続き


32bit
IPアドレス
10111011101010110011010 11001101 0
32bit
ネットマスク
11111111111111111111111 10100000000
23bit
32bit
ネット部
10111011101010110011010 00000000
23bit
32bit
ホスト部
00000000000000000000000 10100110 10
23bit

IPアドレスとネットマスク

- mail0.sfc.keio.ac.jpと通信したい！
  - DNSでアドレスを取得　— 133.27.4.120
  - ネットワークアドレスを探す
    - ホストによって異なる
      - 経路制御表にネットマスクが埋め込まれている
      - クラスによるネットワーク部との比較 etc....
  - 経路表にしたがって次のホスト（ルータ）を指定
  - ルータに向かってパケットを送信

ＩＰアドレス　続き

- 歴史的にクラスという方法で構造化を行っていた
  - クラスビット（アドレスの先頭数ｂｉｔ）による識別
  - ５つのクラス
    - クラスA　（１２６ネットワーク、１６７７７２１４ホスト）
    - クラスB　（１６３８２ネットワーク、６５５３４ホスト）
    - クラスC　（２０９７１５０ネットワーク、２５４ホスト）
    - クラスD　（実験に利用）
    - クラスE　（予約されている）

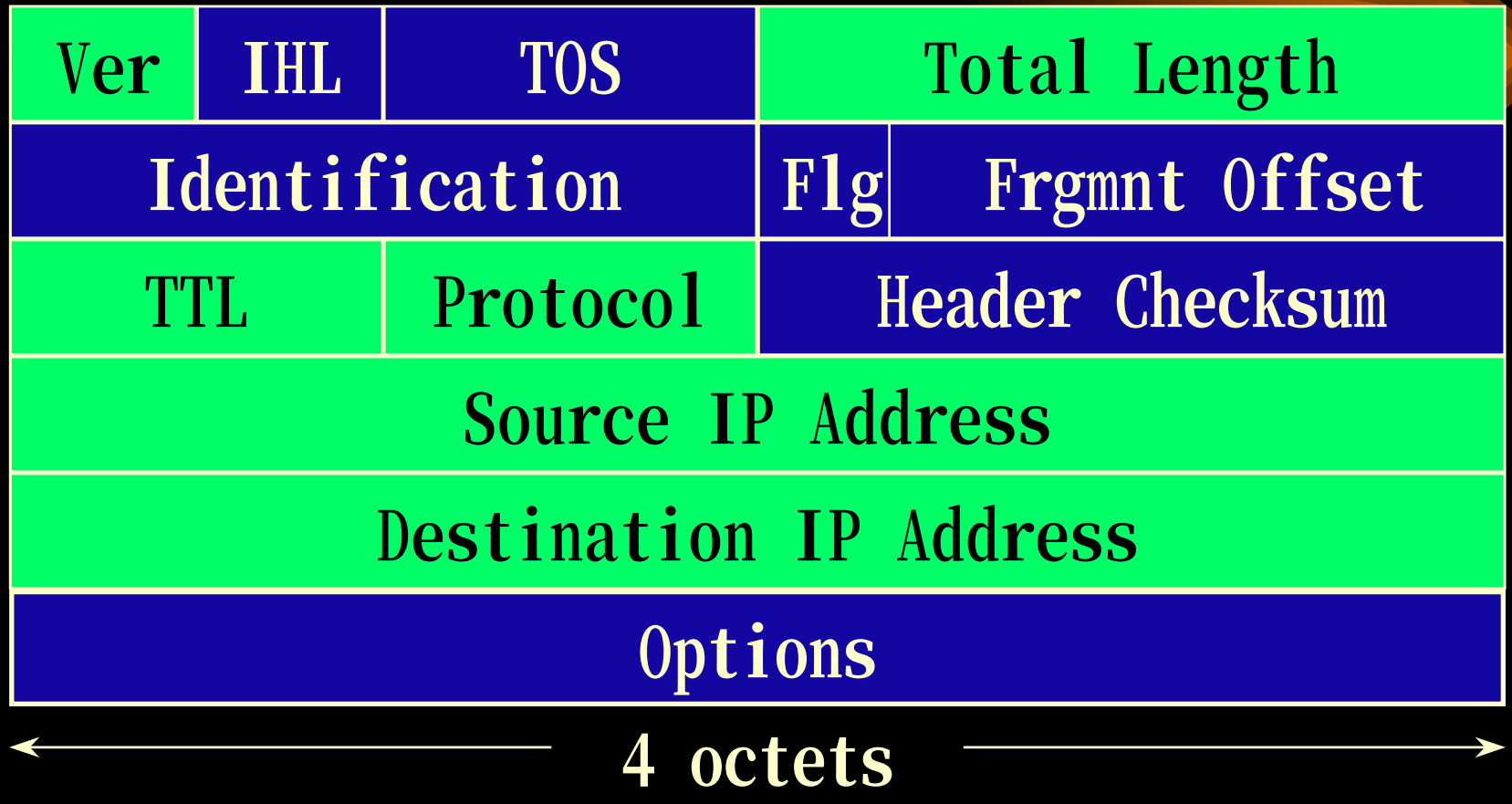
IPアドレス表記法

- 通常は10進数
  - 133.27.4.120　(10進数)
- コンピュータの中では2進数
  - 10000101 00011011 00000100 01111000 (2進数)
- 二つの特別なアドレス
  - ネットワークアドレス
    - ホスト部が 0
    - 133.27.4.0
  - ブロードキャストアドレス
    - 全員に届く
    - 255.255.255.255 や　133.27.5.255

IPヘッダ

- データグラムの配送に必要な情報を含む
  - 送信者アドレス
  - 受信者アドレス
  - データグラムの長さ　etc. ．．．
- 一つ一つのデータグラムに付加される
  - VCではない

IPヘッダ


Ver
IHL
TOS
Total Length
Identification
Flg
Frgmnt Offset
TTL
Protocol
Header Checksum
Source IP Address
Destination IP Address
Options
4 octets

IPヘッダ

- Ver.
  - バージョンフィールド　現在は 4
- IHL
  - ヘッダの長さ　4octet を 1としてカウント
- TOS
  - Type of Service　配送のタイプ
- Total Length
  - IPデータグラム全体の大きさ　（1octetでカウント）

IPヘッダ（続き）

- Identification
  - もともとのデータを識別する
  - Fragmentation の際に必要
- flg （Flag)
  - Fragmentされたか、Fragmentしてはならないか
- Fragment Offset
  - 元のデータのどこでFragmentか
  - 8octets 単位で計られる

IPヘッダ（続き）

- TTL （Time To Live）
  - このデータグラムの寿命
  - 基本的には1つのホストを経由すると1減る
  - 初期値は64
- Protocol
  - 上位層が何か
- Header Checksum
  - ヘッダ部分の検証
  - データが壊れていないか

IPヘッダ（続き）

- Source Address
  - 送信元ホストのアドレス
- Destination Address
  - 送信先ホストのアドレス
- Option
  - 特別な機能を提供

IPとエラー

- エラー
  - データグラムの紛失
  - checksum の不一致
  - Fragment したデータの一部欠落
- エラー訂正　vs　エラー報告
  - IPはエラーの訂正は行わない
  - 送信者にエラーの報告を行う
  - 送信者はアプリケーションにエラーを報告する

ICMP(Internet Control Message Protocol)

- IPが送信者にエラーを知らせる機能
  - 送信先への到達不能
  - 送信元へのフロー制御
  - 経路変更
  - TTL＝０ の時 パケットの破棄
- 送信先への到達可能テストとその状態
  - ping コマンド

ping の例

```
[9:27am]shonan~(@_@)ping altavista.digital.com
PING altavista.digital.com (204.123.2.66): 56 data bytes
64 bytes from 204.123.2.66   icmp_seq=0 time=224 ms
64 bytes from 204.123.2.66   icmp_seq=1 time=223 ms
64 bytes from 204.123.2.66   icmp_seq=2 time=225 ms
64 bytes from 204.123.2.66   icmp_seq=3 time=229 ms
^C

----altavista.digital.com PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 223/225/229
[9:27am]shonan~(-_-)
```

ping の動作

- ICMP　Echo Request と Echo Reply
  - Echo Requestを受けたホストはEcho Replyを返す
- パケットを出すと同時にタイマをスタート
- TTLは255（最大）に設定
- それぞれのメッセージにＩＤが振る
- 同IDのパケットを受け取ったらタイマをストップ
- 時間とTTLを表示

traceroute の例

```
[9:27am]shonan~(-_-)traceroute altavista.digital.com
traceroute to altavista.digital.com (204.123.2.69), 30 hops max, 40 byte packets
 1  cisco-sfc.sfc.wide.ad.jp (133.4.29.3)  60 ms  92 ms  10 ms
 2  jp-gate.wide.ad.jp (133.4.11.1)  2 ms  3 ms  2 ms
 3  jp-entry.wide.ad.jp (133.4.1.2)  3 ms  3 ms  4 ms
 4  us-entry.wide.ad.jp (133.4.43.2)  138 ms  142 ms  131 ms
 5  border1-serial2-1.SanFrancisco.mci.net (204.70.32.13)  222 ms  220 ms  230 m
s
 6  borderx1-fddi0-0.SanFrancisco.mci.net (204.70.2.164)  232 ms  236 ms  239 ms
 7  barrnet.SanFrancisco.mci.net (204.70.158.102)  222 ms  227 ms  226 ms
 8  paloalto-br1.bbnplanet.net (131.119.0.193)  221 ms  224 ms  231 ms
 9  decwrl.bbnplanet.net (4.0.1.58)  225 ms  223 ms  225 ms
10  digital-gw1.pa-x.dec.com (204.123.0.241)  227 ms  224 ms  222 ms
11  altavista.digital.com (204.123.2.69)  223 ms  224 ms  227 ms
```

tracerouteの動作

- ポート番号30000以上のUDPパケット
- TTLを１から順に大きくする
- TTL=0になったホストから ICMP Time Exceeded メッセージを受ける
- ICMP Destination Port Unreachable メッセージを受けるまで繰り返す

経路制御（Routing） その３

Routing tables

Destination	Gateway	Flags	Refcnt	Use	Interface
127.0.0.1	127.0.0.1	UH	1	10055	lo0
133.12.30.2	133.4.27.1	UGH	0	67	le0
133.4.105.0	133.4.31.6	UG	0	0	le1
133.4.27.0	133.4.27.4	U	19	59513	le0
133.4.29.0	133.4.29.8	U	4	141350	nf0
133.4.30.0	133.4.29.2	UG	6	8178	nf0
133.4.31.0	133.4.31.9	U	12	81608	le1
133.4.34.0	133.4.29.1	UG	2	806038	nf0
133.4.42.0	133.4.29.5	UG	1	4605	nf0
133.4.46.0	133.4.29.6	UG	2	129764	nf0
133.4.68.0	133.4.29.1	UG	0	0	nf0
133.4.68.32	133.4.29.1	UG	0	0	nf0
133.4.68.64	133.4.29.1	UG	0	0	nf0
224.0.0.0	133.4.27.4	U	0	2964152	le0
default	133.4.29.3	UG	18	1854553	nf0

トランスポート層

TCP/UDP

トランスポート層

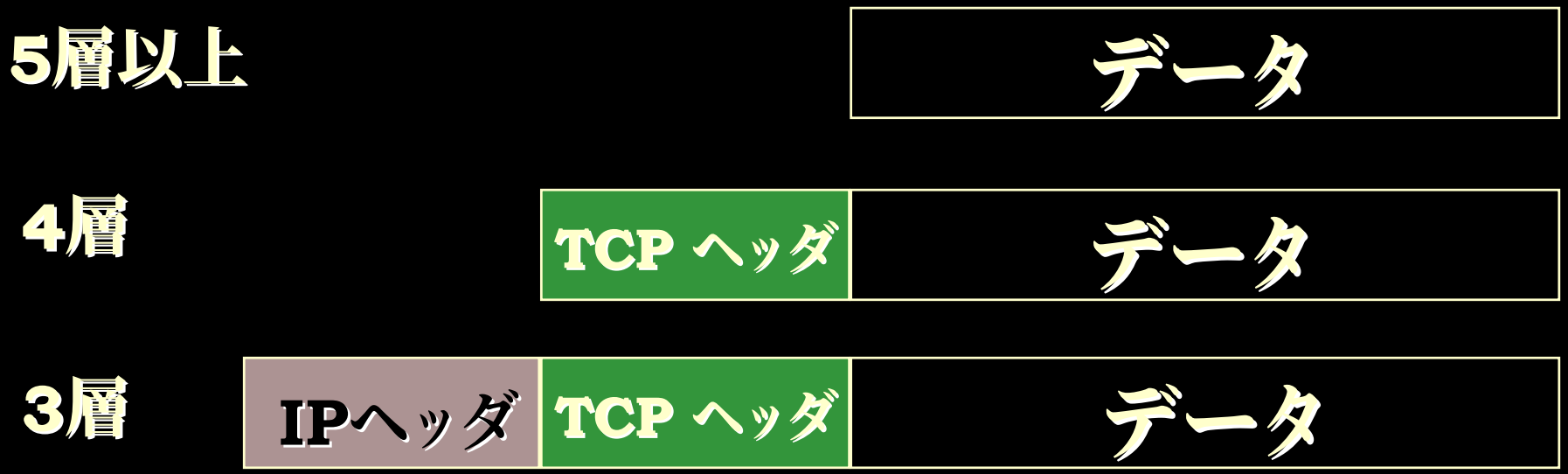
- マシンまで到達したデータをアプリケーションと結び付ける
- TCP
- UDP

トランスポート層の役割

- 届いたデータをどのプログラムに渡せばいいか?
  - ホスト内でアプリケーションを識別
  - ポート番号
- アプリケーションに通信の質の選択を提供
  - 複雑だが信頼性を保証する通信
    - Connection Oriented
    - Virtual Circuit
  - 単純だが信頼性を保証しない通信
    - Connection Less

ネットワーク層とトランスポート層

- トランスポート層でもヘッダを付ける
- ネットワーク層ではトランスポート層ヘッダがついたデータにネットワーク層ヘッダを付加する



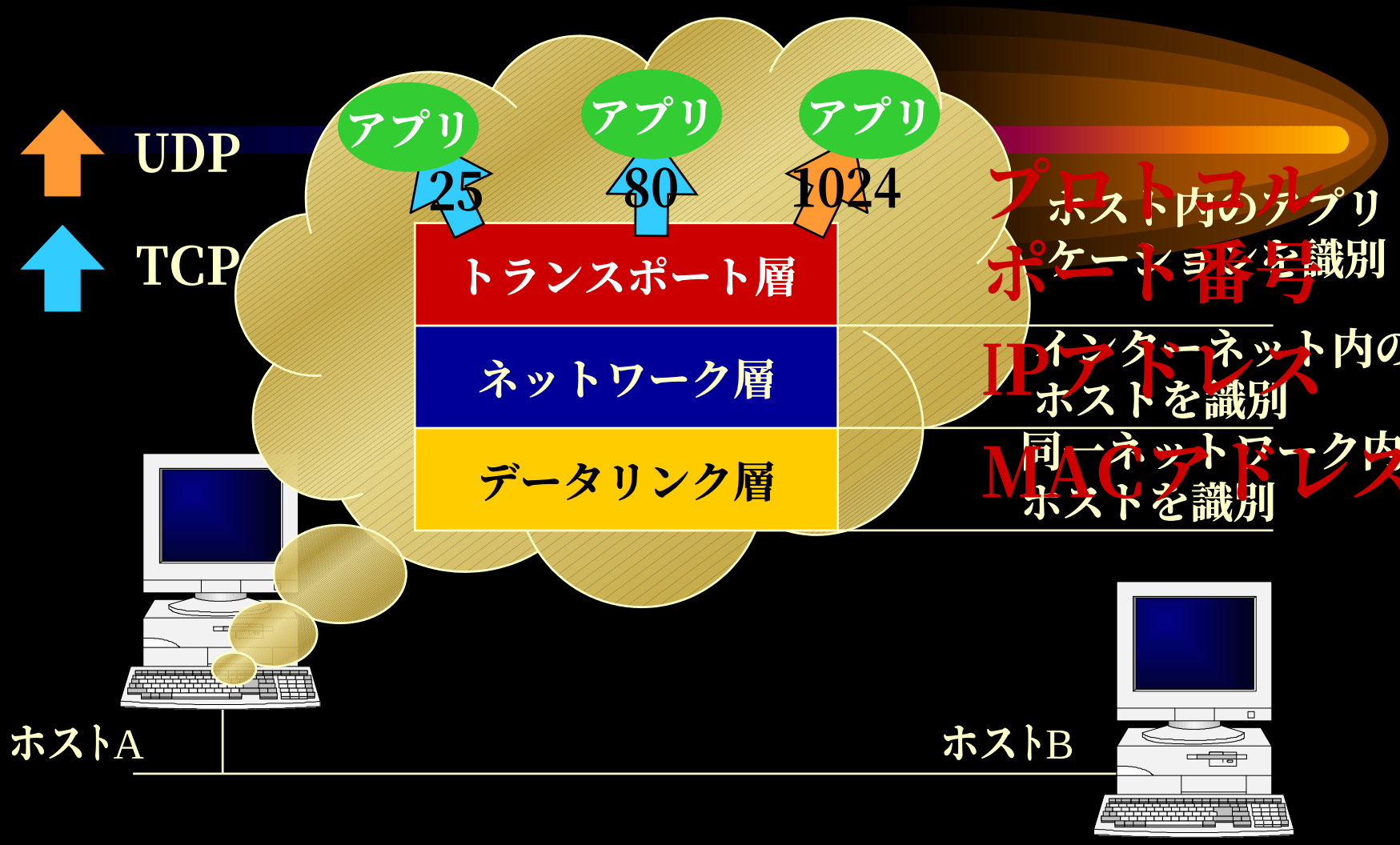
ポート番号

- ネットワーク層によるマシンまでのデータ配送
- 一つのホスト内でアプリケーションごとに割り振りが必要
  - 同一ホスト内でのアプリケーションの識別
- アプリケーションの識別子
  - プロトコル（TCP・UDP）
  - ポート番号（プロトコルに固有）

つまり…

- インターネット上のアプリケーションの識別
  - トランスポート層のプロトコル
  - ポート番号
  - IPアドレス
- ネットワーク上の連続するデータの流れ
  - フロー

ネットワーク層とトランスポート層



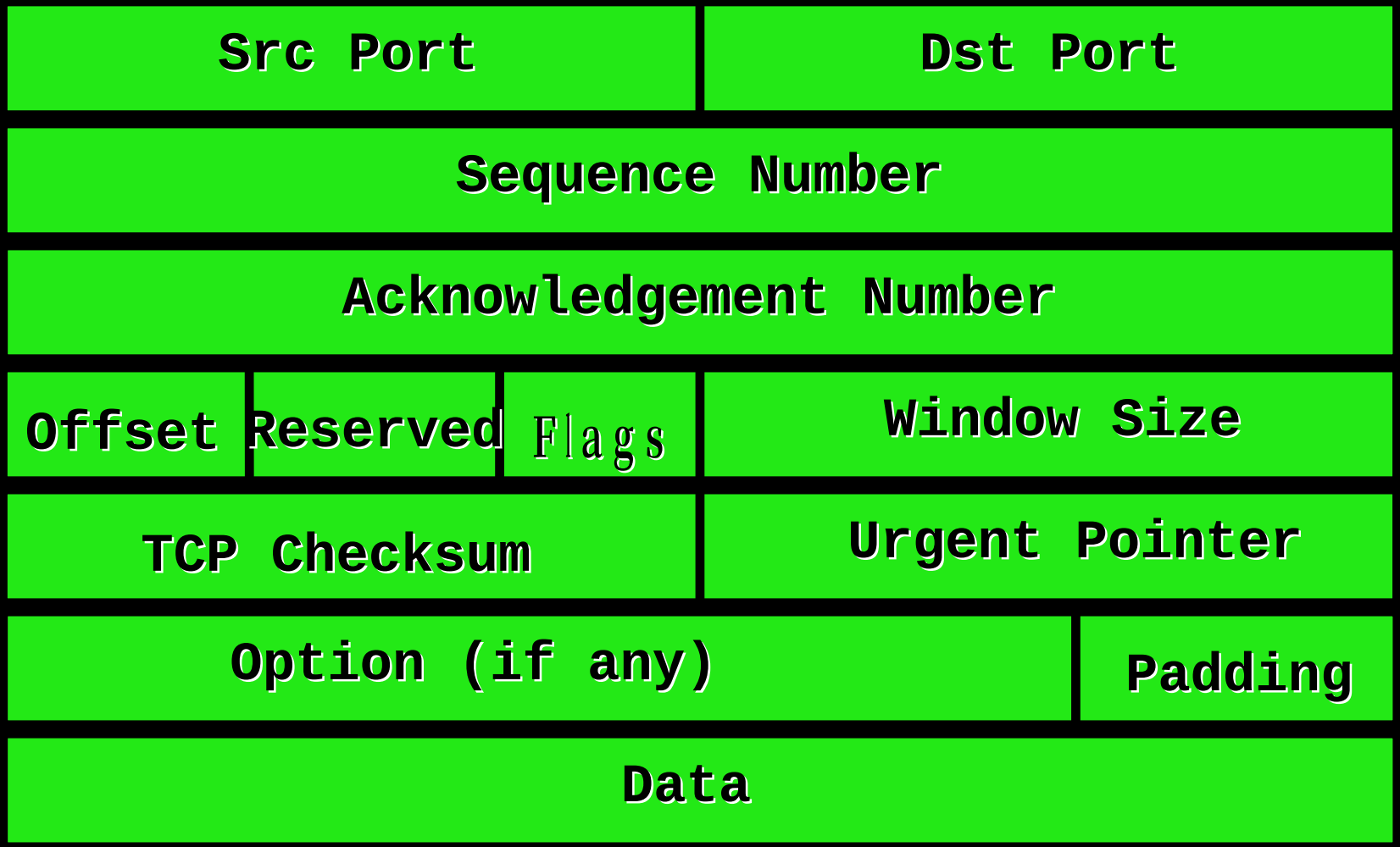
ネットワーク層・トランスポート層
IPアドレスとポート番号の応用

- ネットワーク管理
  - セキュリティ保持のため
    - Firewall (ファイアウォール)
- サービスの質を考慮した通信
  - フローラベル

TCP (Transmission Control Protocol)

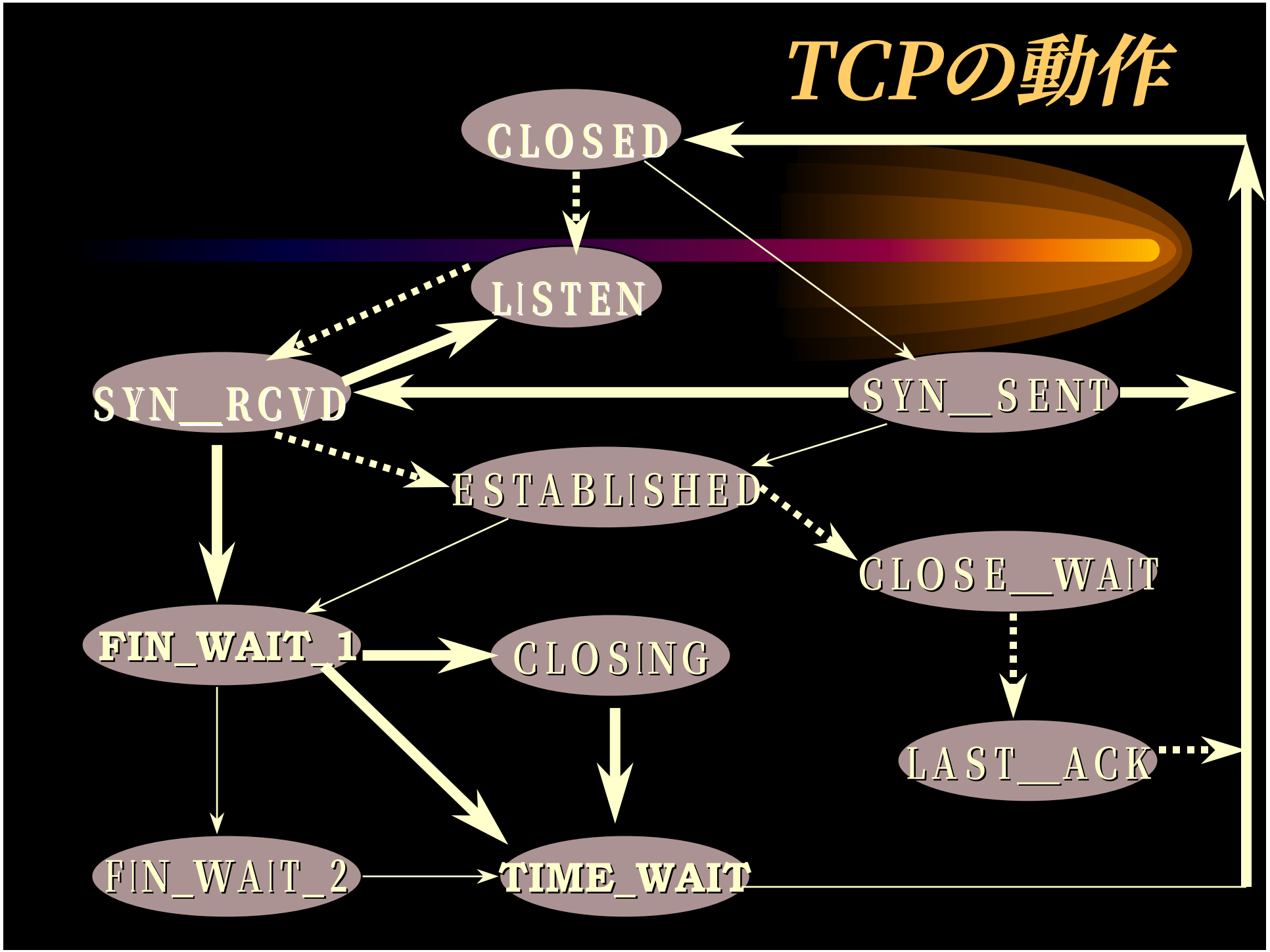
- データ転送に関する制御を行う
- 具体的には……
  - IPに信頼性を加える
    - VC型通信
    - エラー検出とエラー訂正
    - フロー制御
    - 順序の再構成
  - データ転送に関するユーザインターフェイス

TCPヘッダ

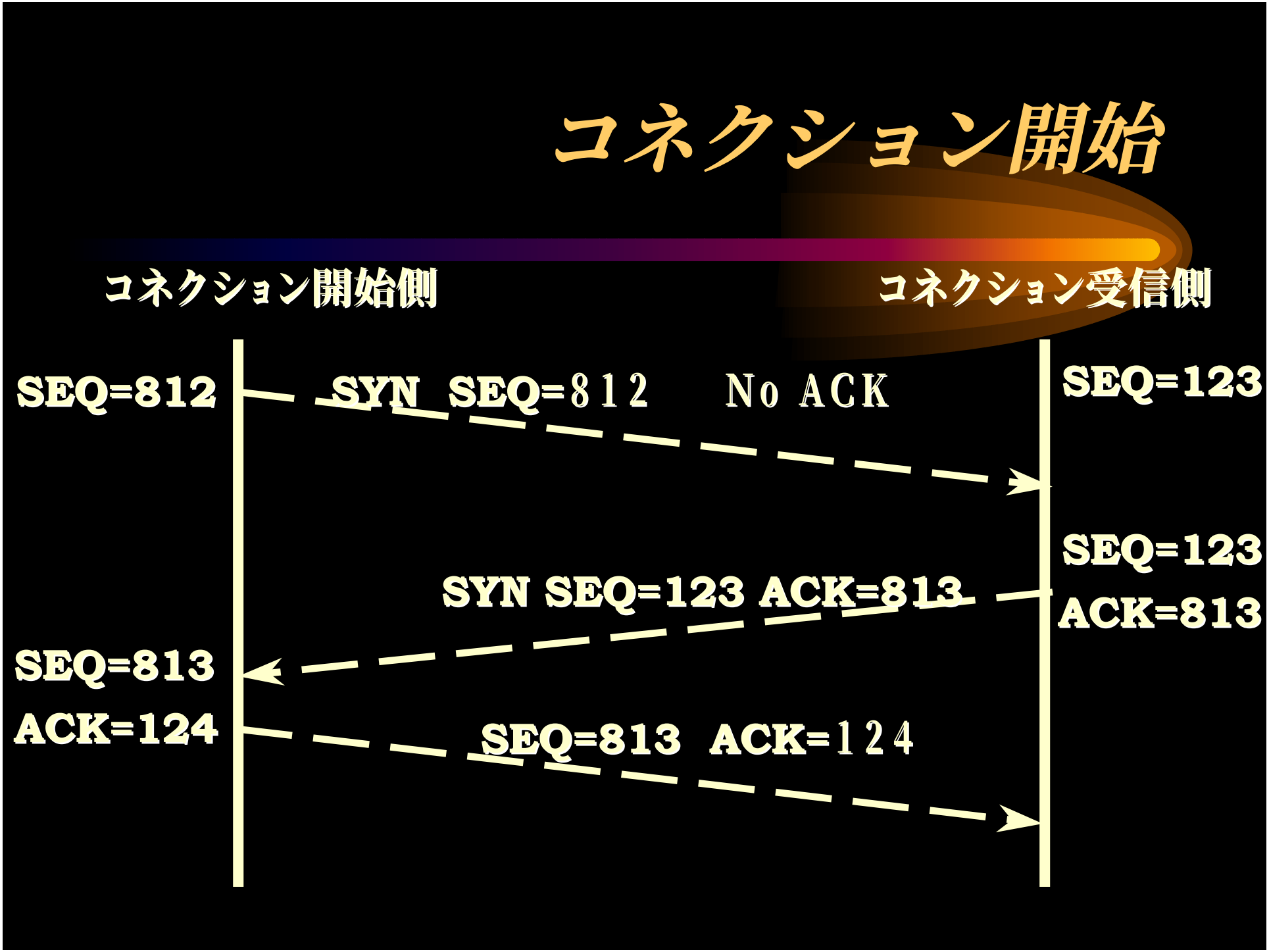

Src Port
Dst Port
Sequence Number
Acknowledgement Number
Offset
Reserved
Flags
Window Size
TCP Checksum
Urgent Pointer
Option (if any)
Padding
Data

Flags Code

- FlagsはTCPがコネクション制御のために利用する(6種類)
  - URG： Urgent Pointerが有効
  - ACK： Acknowledge bitが有効
  - PSH： セグメントをすぐに上位層へ渡す
  - RST： エラーによる強制的なクローズ
  - SYN： コネクションセットアップの同期をとる
  - FIN ： コネクションを終了する


TCPの動作
CLOSED
LISTEN
SYN_RCVD
SYN_SENT
ESTABLISHED
CLOSE_WAIT
FIN_WAIT_1
CLOSING
LAST_ACK
FIN_WAIT_2
TIME_WAIT

コネクション開始


コネクション開始側
コネクション受信側
SEQ=812
SYN SEQ=812 No ACK
SEQ=123
SEQ=123
SYN SEQ=123 ACK=813
ACK=813
SEQ=813
ACK=124
SEQ=813 ACK=124

3-Way Handshake

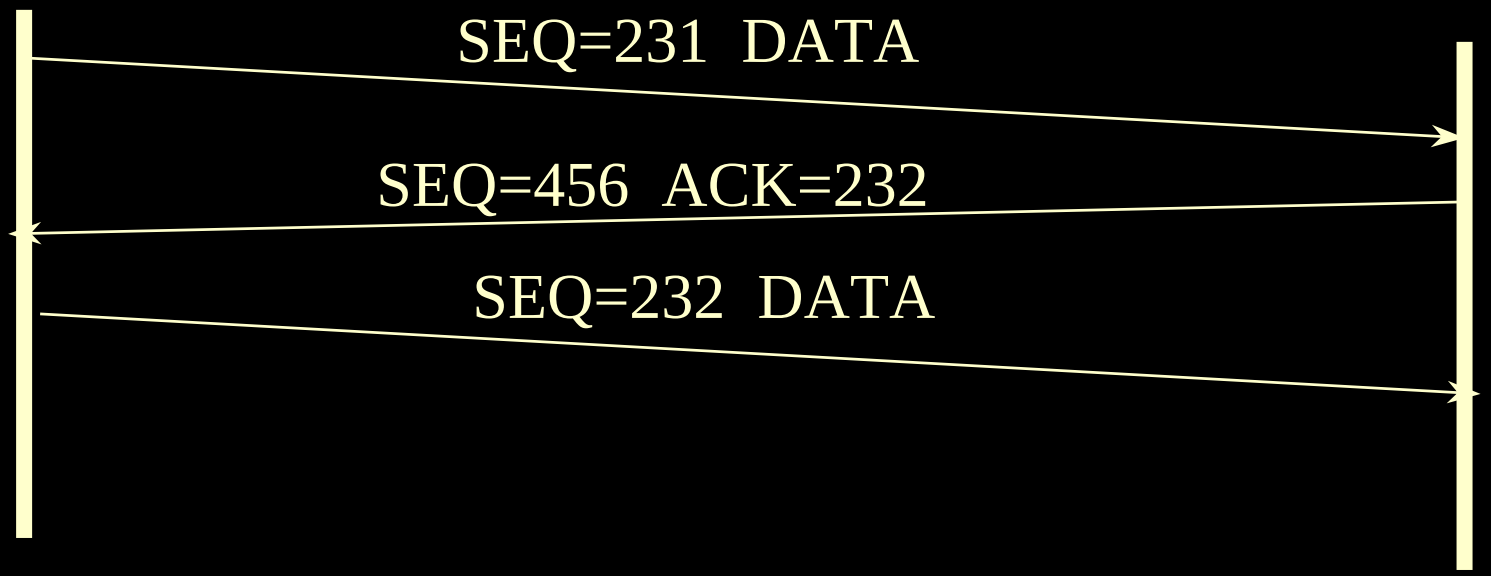
- コネクションのセットアップ
  - ホスト1、ホスト2間で
    - ホスト1はSYN　Flagのついたパケットを送る
    - SYNを受けたホスト2は相手との同期を取るためにSYN Flagのついたパケットを送る。この時いっしょに受け取ったSYNに対するACK Flagもつける
    - ホスト2からSYNを受け取ったホスト1はACKを返す
  - SYN と ACKが相乗り
    - Piggy back

コネクションの終了

- コネクションの終了はFIN　Flagによって行う
- コネクションの終了は一方ごとにできる

データの転送

- Sequence NumberとAcknowledge Numberによって、データの順番を保証
- ACKが帰ってきたら次のデータを送る
- ACKが帰ってこなかったら前のデータを再転送

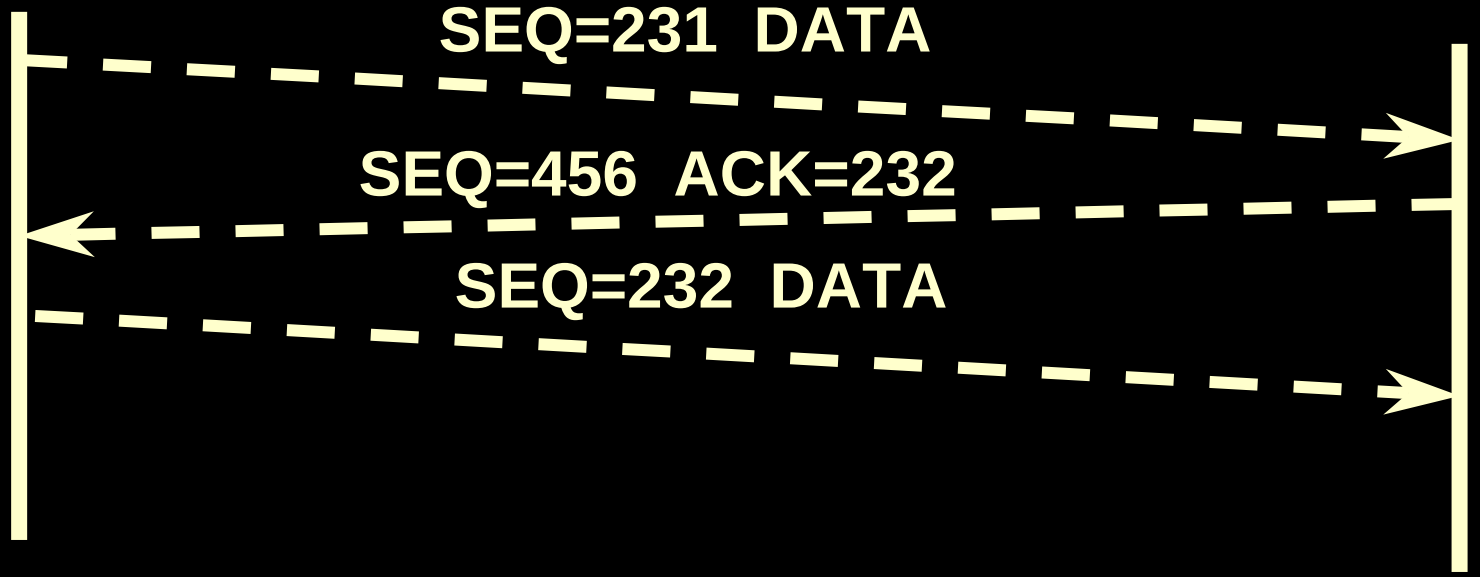


データ転送（続き）

- いちいちACKを待つと効率が悪い
- 推測を元にある程度先送りした方が良い
- だが、先送りしすぎるとACKがなかなか帰ってこない
- 幅を決めなければならない

データの転送

- Sequence NumberとAcknowledge Numberによって、データの順番を保証
- ACKが帰ってきたら次のデータを送る
- ACKが帰ってこなかったら前のデータを再転送

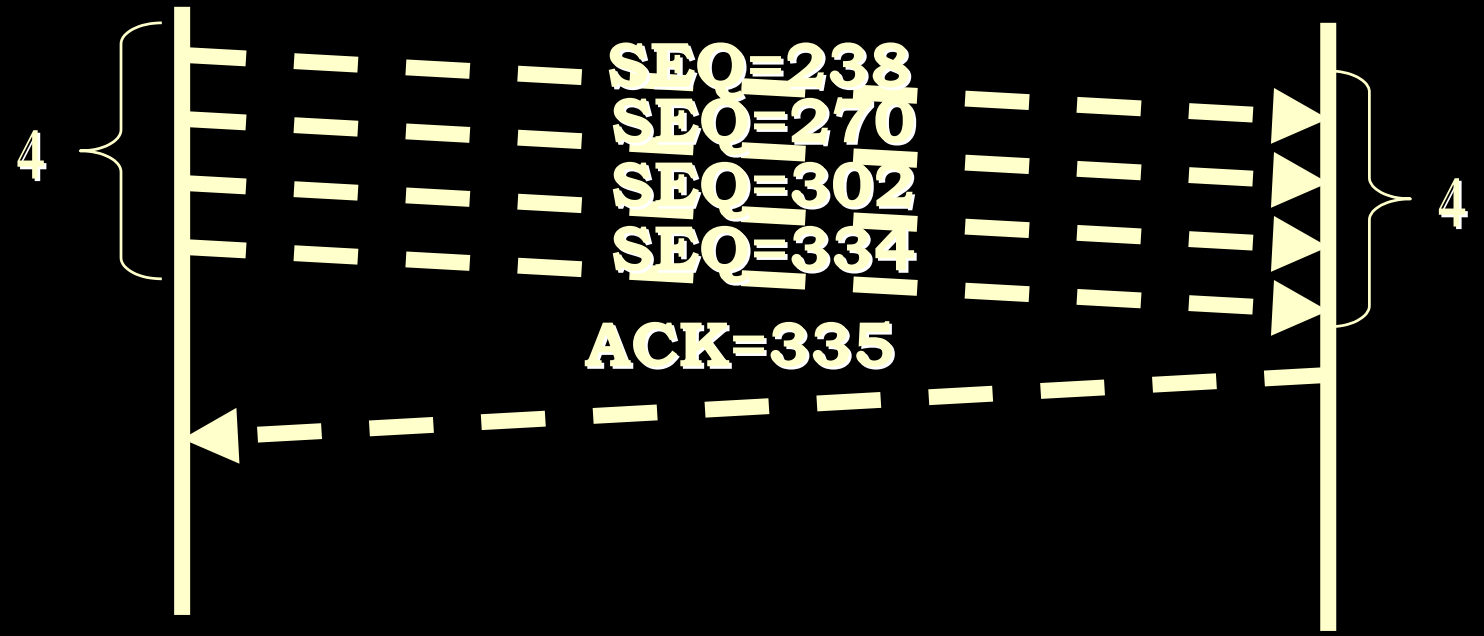


データ転送（続き）

- いちいちACKを待つと効率が悪い
- 推測を元にある程度先送りした方が良い
- だが、先送りしすぎるとACKがなかなか帰ってこない
- 幅を決めなければならない

ウィンドウコントロール

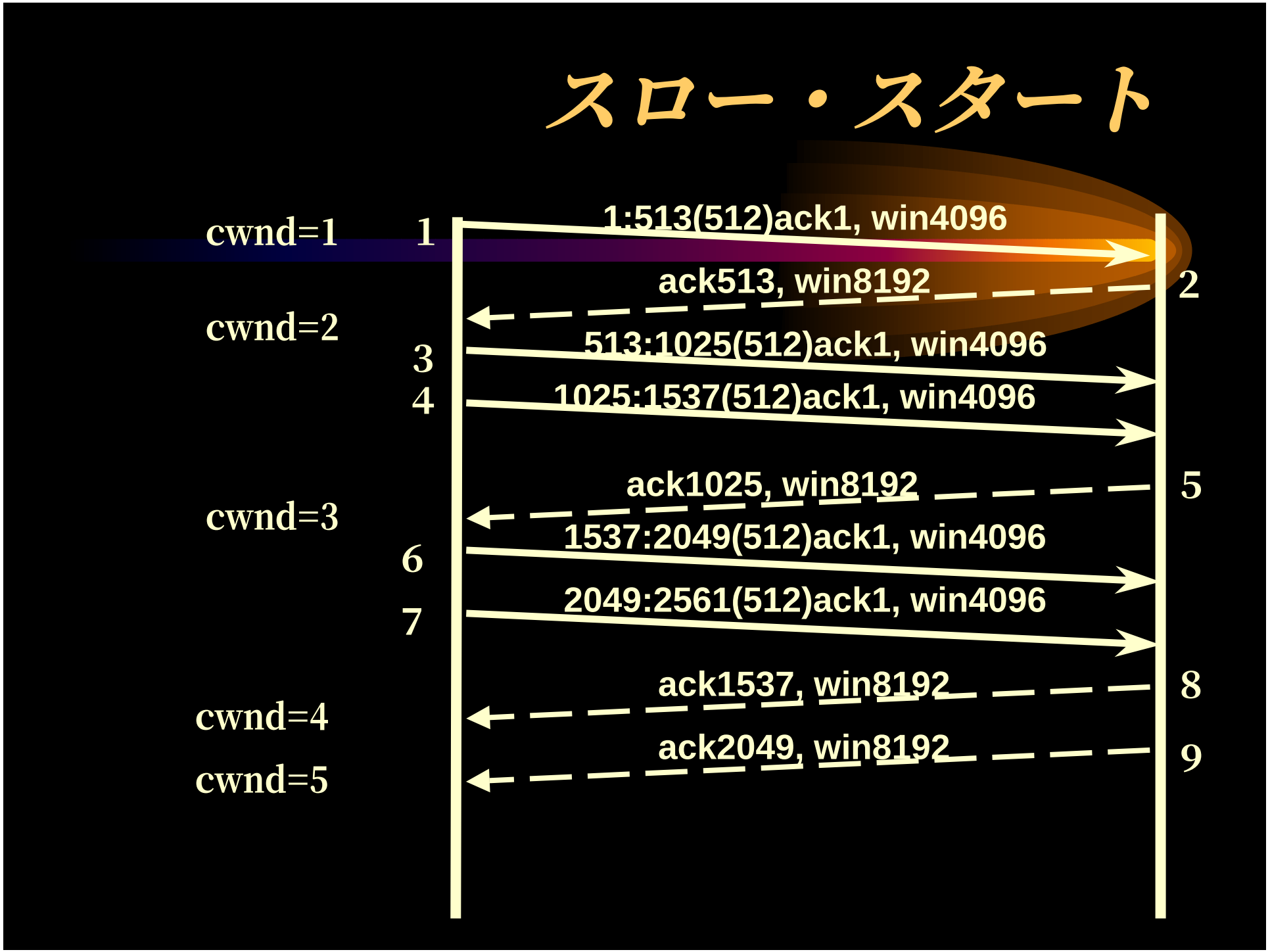
- 先送りする幅を決める仕組み
- いくつのパケット？



スロー・スタート

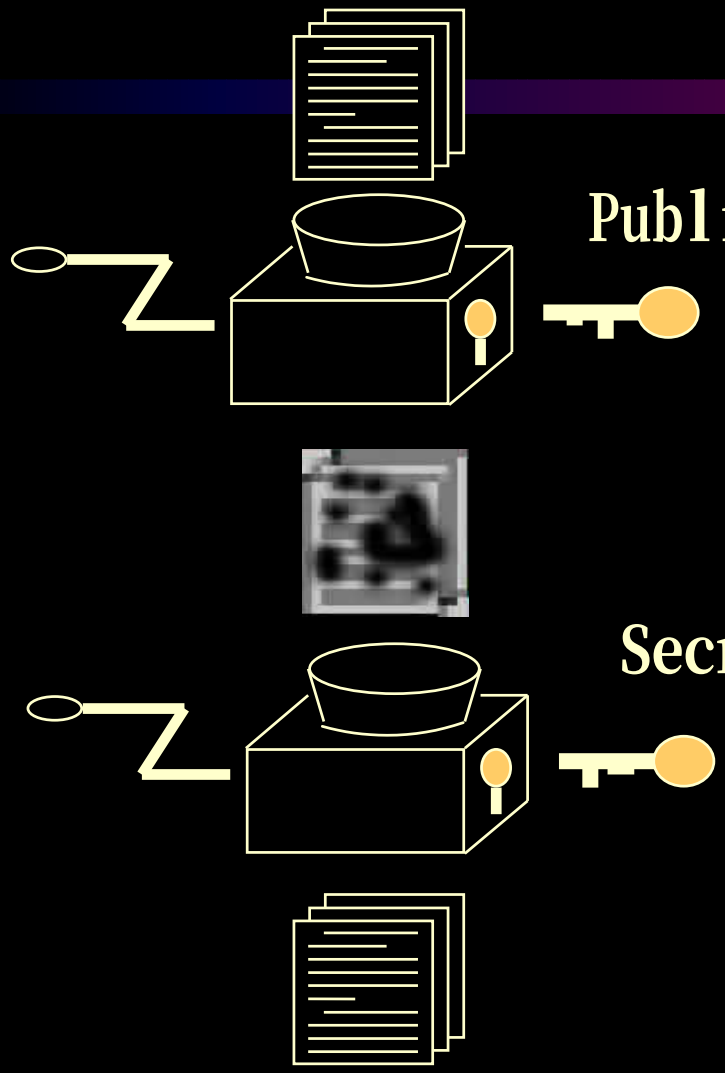
- ネットワークに送り出されるパケットの通信速度を他方のエンドから返される確認応答の通信速度を同期させるためのもの
- 送り手のTCPに別のウィンドウを追加する。
  - 輻輳ウィンドウ(cwnd)
- 新しいコネクションが確立
- 輻輳ウィンドウをセグメント1つに初期化
- ACKが受け取られるたびに、輻輳ウィンドウは1セグメントずつ増やされる

スロー・スタート


cwnd=1
1
1:513(512)ack1, win4096
ack513, win8192
2
cwnd=2
513:1025(512)ack1, win4096
3
1025:1537(512)ack1, win4096
4
ack1025, win8192
5
cwnd=3
1537:2049(512)ack1, win4096
6
2049:2561(512)ack1, win4096
7
ack1537, win8192
8
cwnd=4
ack2049, win8192
9
cwnd=5

セキュリティ・信頼性

公開キーによるセキュリティ



- 閉じる鍵
  - 公開鍵
- 開く鍵
  - 秘密鍵

セキュリティ

- 成りすまし・盗聴・改竄
  - 成りすまし：
    - あるホストのふりをし不正にデータを　操作すること
  - 盗聴：
    - 盗み聞きし，通信の内容を知ること
  - 改竄
    - ネットワーク中に流れているデータを書き換えること

信頼性とセキュリティ

- 誰に対して信頼性を保証するか
- どのレイヤ
- アプリケーションとアプリケーション
- ノードとノード
- ポートとポート

UNIXとTCP/IP

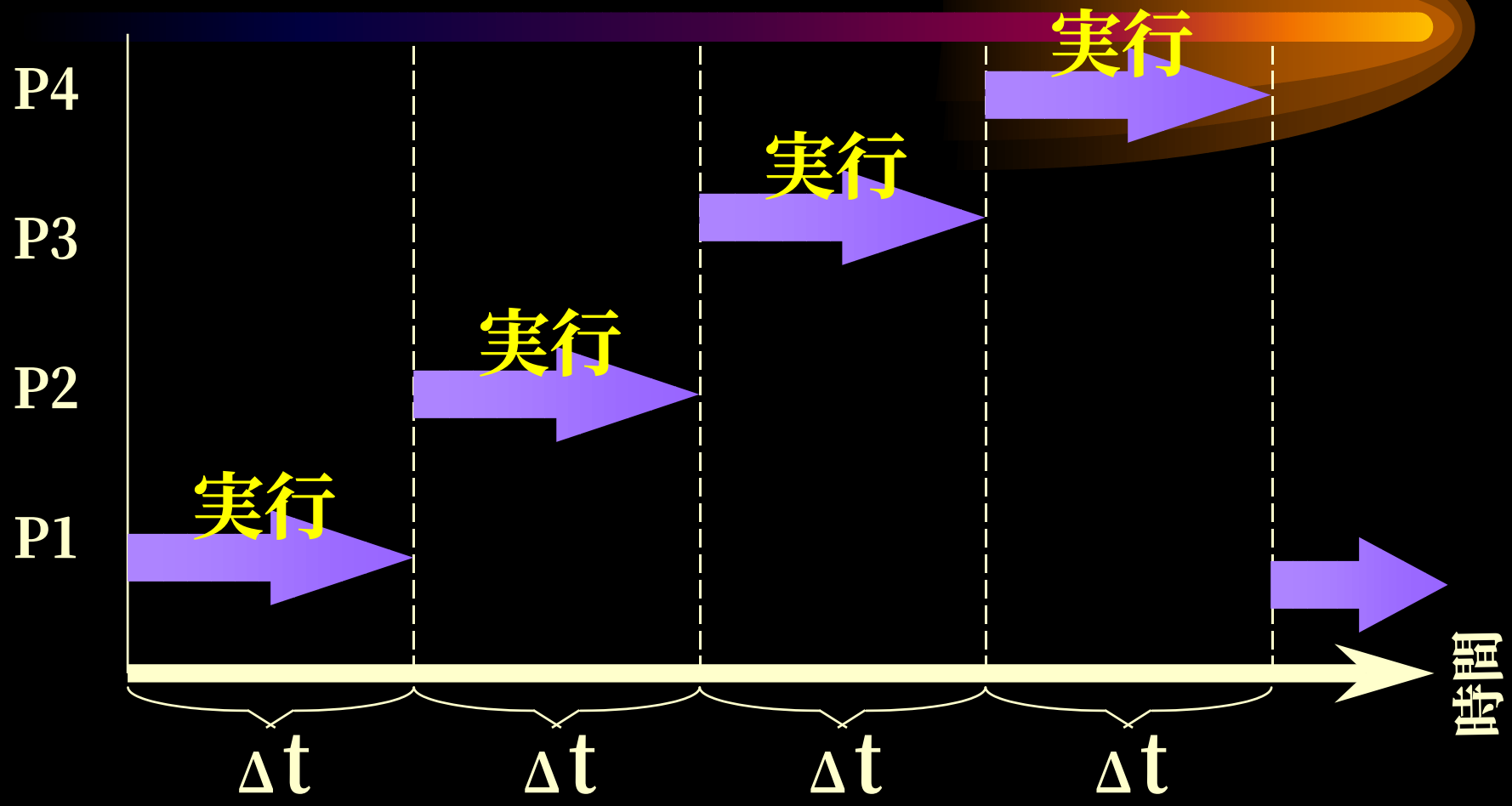
UNIXオペレーティングシステム

- Uni-plexed Information Computing System
- 優れた抽象
  - 取り扱うデータは全てファイル
  - open, close, read, write, seek, ioctl……
- ソースコードの公開
  - 教育機関を中心に普及
  - BSD (Berkley Software Distribution)
- C言語で記述
  - 移植性が高い

UNIXとTCP/IP

- 4.2BSD から
  - カーネルサービスの一部として実装
- 資源利用の工夫
  - mbuf　(Memory Buffer/Murai buffer? :)
- 割り込みとタイムアウト
  - ハードウェアからソフトウェアへ
  - クリティカルセクション(Critical Section)
  - slow time out と fast time out

時分割システム（ＴＳＳ）



Internet FAX Machine

- Internet FAX Machine
- Example(panasonic)
  - TCP/IP
  - MIME/SMTP
  - G3
  - sending/receiving FAX
  - Internet and PTN(G3) relay
  - OCR email address

Stream と Flow Control

- Stream
  - 送信元と送信先の間のデータ転送
  - 連続したデータの転送
  - Virtual Circuit
- Flow Control
  - 送信元と受信先の組
  - データの転送のコントロール
  - 例： TCP

階層構造


アプリケーション
7
プレゼンテーション
6
セッション
5
トランスポート
4
ネットワーク
3
データリンク
2
物理
1