データリンクの基礎技術

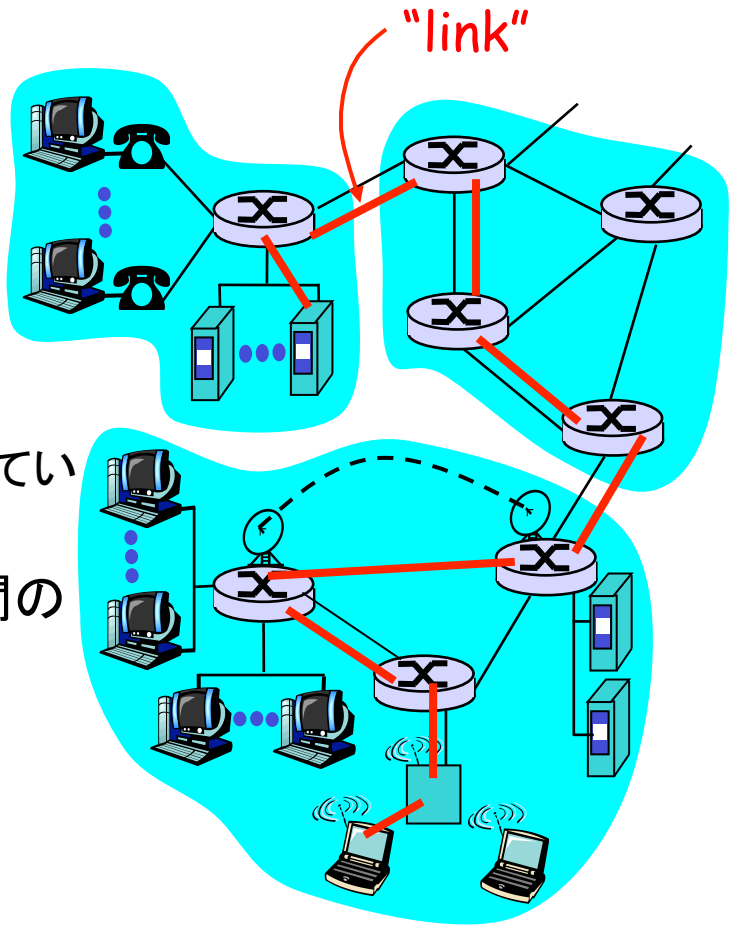
Suguru Yamaguchi
Nara Institute of Science and Technology
Department of Information Science

概要

- データリンクの提供している機能は何か
- 使用されている技術
  - デザイン・コンセプト
  - エラー処理(error control)
  - フロー制御(flow control)
  - 基本的なプロトコル

Introduction

- TCP/IPにおけるデータリンク層は？
  - Point-to-Point Protocol ＆ Medium Access Layer (MAC)
- いくつかの技術:
  - node:ホストやルータのこと
  - link:近隣ノード間を繋ぐ通信パス
    - Wired links, Wireless links, LANs
  - frame:レイヤ2のパケット、カプセル化されている
- データリンク層は、リンクを通じて各ノード間のデータグラム転送を司る



データリンクの基本的な機能

- パケットスイッチングのための基本機能を提供
  - 物理層の転送機能を用いてパケットレベルのスイッチングを行っている
    - フレーム交換
    - アドレスからノードを識別
- 物理層の特徴を考慮したデータ転送機能を提供
  - 安定したデータ転送
  - 伝搬遅延誤り
  - ネットワーク層とのインタフェース
  - 特に共通のインタフェースが提供されていれば、媒体間の差異を吸収できる
  - IEEE802.2 と IEEE802.x

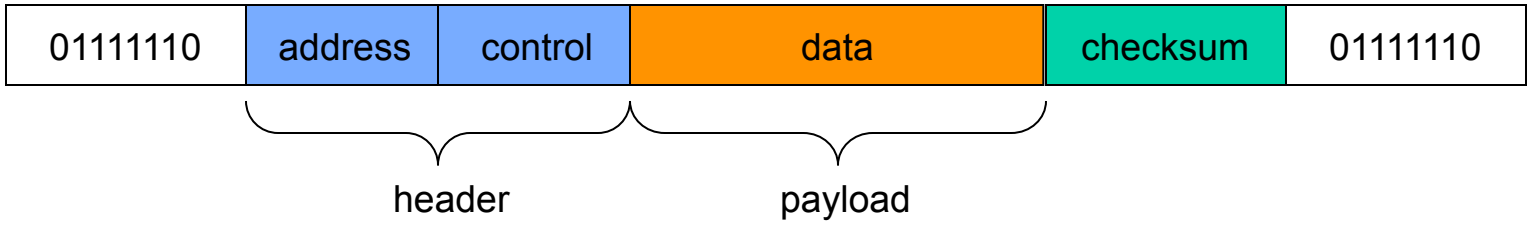
データリンク層のサービス

- フレーム化、リンクアクセス
  - データグラムをカプセル化(headerやtrailerの付与)によってフレームに変換
  - 媒体間のチャネルアクセス
  - 発信元、送信先を識別するためにヘッダに付与するMACアドレスを提供
- フロー制御:
  - 送信／受信ノード間のデータ送信を制御
- エラー検知:
  - 信号劣化やノイズによって発生するエラーを検知
  - パケット受信側がエラーを検出
- エラー訂正:
  - パケット再送要求を出さずにビット誤りを発見し、訂正する
- 半二重(half-duplex)と全二重(full-duplex)
  - 半二重では各ノードやエンドは同時に通信することができない

Framing

Frame

- Data link layer protocol data unit (PDU)
  - フレームの境界を定義
- 障害の切り分け
  - フレーム長に合わせて、ビット列内にエラー検知/訂正コードを付与s
- Frame header
  - エラー検知とフロー制御のための情報
  - 制御情報

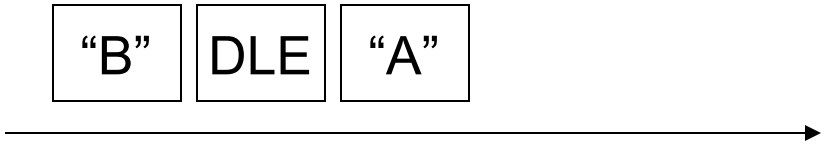


フレームの取り出し (1)

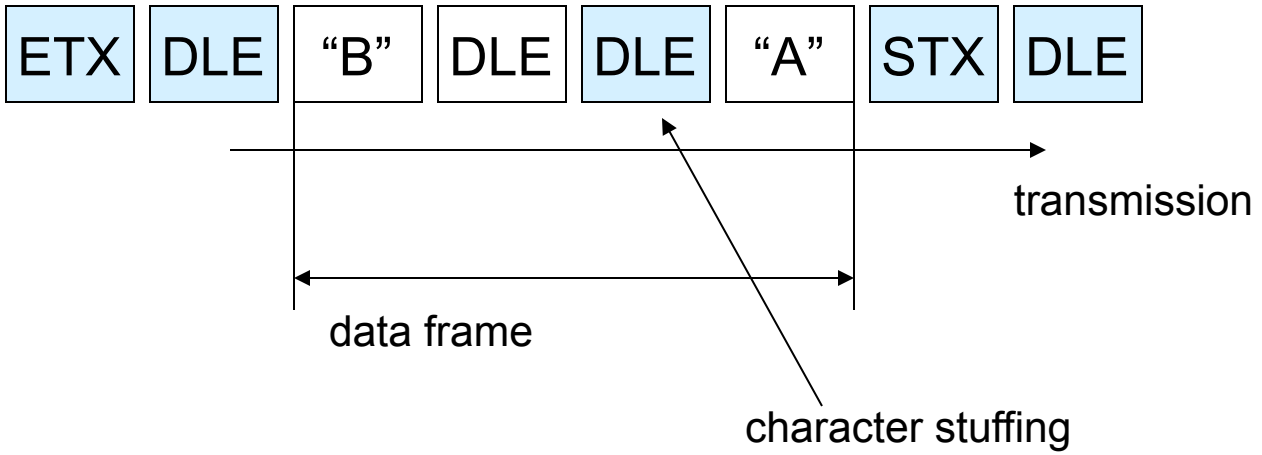
- 文字ベースでのフレーム取り出し
  - 物理層は文字単位でデータを転送する
    - CCITT X.21
      - 物理層フレームは8bit
      - コードフォーマットはASCII(IA5)
  - データ転送の最初と最後に特別な文字を挿入
    - STX, ETX
  - DLE (Data Link Escape)によって特別な文字をデータとして識別
    - バイナリデータの転送では特別な文字がパターン出現する可能性あり
    - character stuffing

フレームの取り出し (2)

Original Data



Transmission Sequence



フレームの取り出し (3)

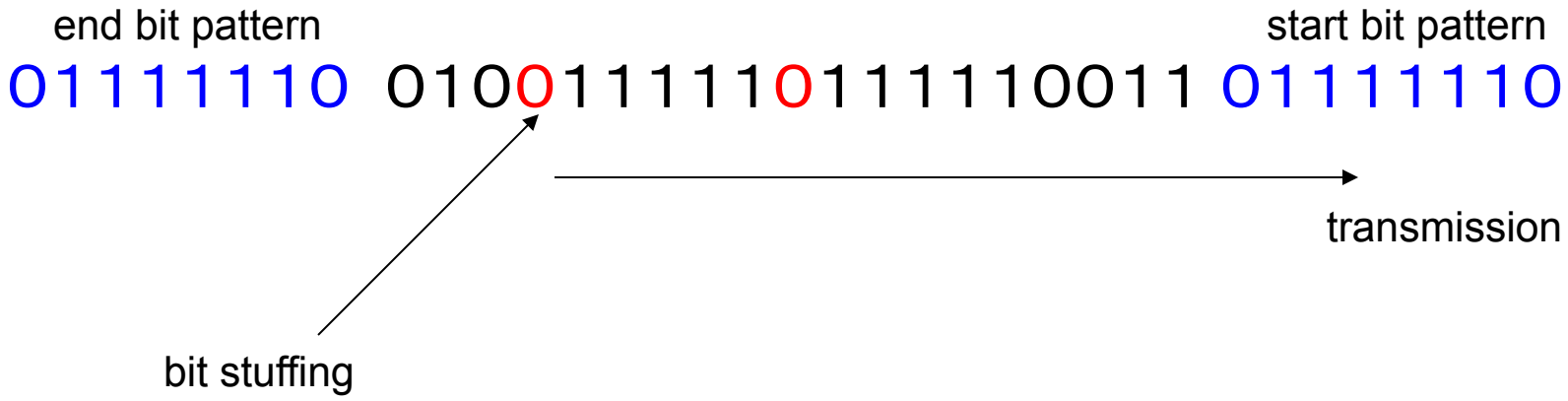
- ビット列をベースとしたフレームの取り出し
  - 特定のビット列をデータの最初と最後に挿入
    - 同期
  - データ列に現れる「同じ」ビット列を加工
    - bit stuffing
      - 特定のビット列がフレームの最初と最後にだけ常に存在するようにする
  - e.g.
    - 特別なビット列：01111110
    - 送信側ではデータ中に5ビット”1”が続いたら”0”を挿入
    - 受信側ではデータ中に5ビット”1”が続いたら次に来る”0”を削除

Frame Synchronization (4)

Original Data

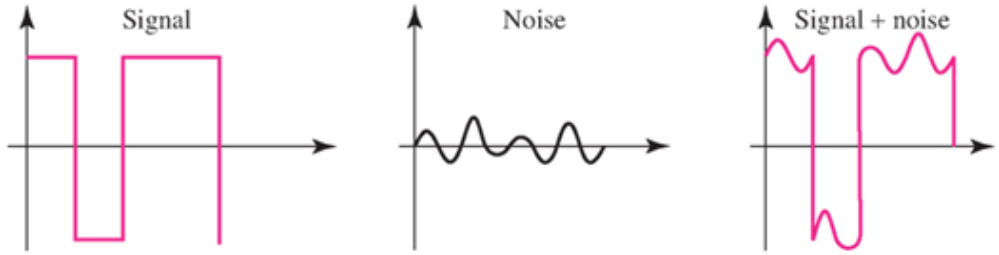
01011111111111110011

Transmission Sequence

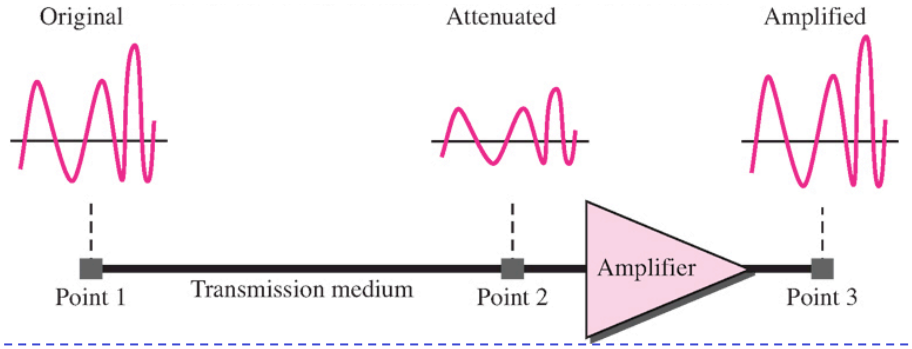


Error Control, Correction & Detection Schemes

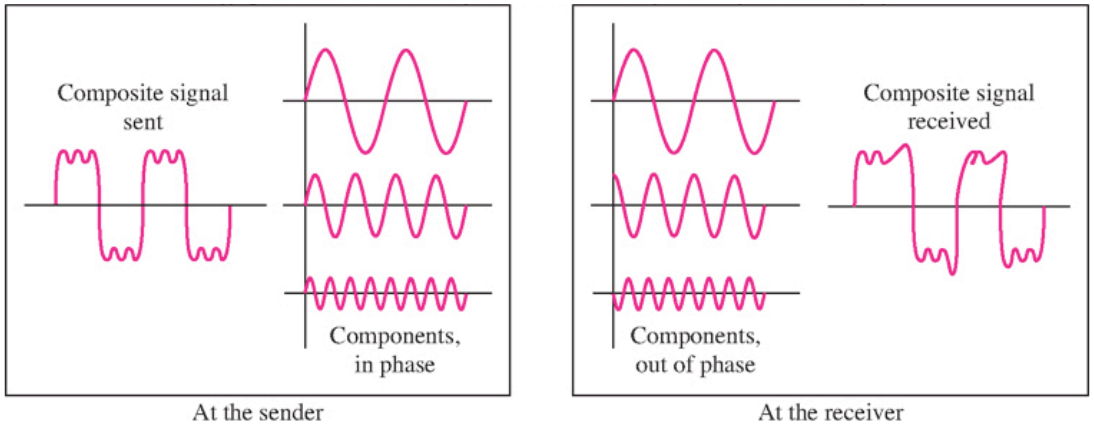
Error in Physical Layer



Noise



Attenuation



Distortion

Error Control

- 目的
  - 通信路上でに伝送誤りを発見・訂正
  - 送信フレームが正しく送られているかどうか
  - 送信フレームの順番が正しいかどうか
- 技術
  - 「フレーム」という概念の導入
  - 符号技術
    - 誤り訂正符号(Error Correction Code)
    - 誤り検出符号(Error Detection Code)
  - プロトコル技術
    - タイマ(Timer)
    - 再送(Retransmission)

Parity Bitによる誤り検出

- 1ブロックのデータに付与する1ビットの符号
  - 転送中にデータが破損していないかを保証する
  - 1ビットの誤り検出
    - 2ビット以上の誤りは検出できない場合がある
- 7ビット毎にパリティビットを付与
  - データ転送プロトコルがOdd parityを付与した場合、Odd parityとなる
  - Even Parityを付与する場合は、Even Parityとなる

Odd parity
- Initial value: 1010101 (four 1s)
- Parity bit added: 1
- Transmitted value: 10101011
- Result: odd parity (five 1s)

Even parity
- Initial value: 1010101 (four 1s)
- Parity bit added: 0
- Transmitted value: 10101010
- Result: even parity (four 1s)

Cyclic Redundancy Check(CRC) (1)

- 連続したビット誤り(burst error)を検出
- フレームの後ろに誤り検出用のCRCチェックビットを付加
- ビット列の多項式表現
  - *n*-bit sequence
    - $(a_{n-1}, \ldots\ldots, a_1, a_0)$ ※ a_i : 0 or 1
  - 多項式表現
    - $F(x)=a_{n-1}x^{n-1}+\ldots+a_2x^2+a_1x+a_0$
- 多項式演算
  - modulo 2
  - 排他的論理和 (eXclusive OR)と等価

Cyclic Redundancy Check (2)

- CRCの計算手順
  - 生成多項式$G(x)$ ただし次数は r $(r < m)$
  - mビットのフレームを送信するとし、その多項式表現を$M(x)$(次数 $m-1$)とする
  - フレームに r 個の”0”を付けたもの $x^rM(x)$ を作成
  - 多項式$x^rM(x)$ を$G(x)$で割った余り$R(x)$ を計算
  - 送信する符号語 $F(x)$を作る
    - $F(x) = x^rM(x) + R(x)$

Cyclic Redundancy Check (3)

- 誤りの検出
  - *F(x)* は常に *G(x)*で割り切れる
  - 割り切れなかった場合は誤りが存在
  - rビット以下の誤りは検出可能
  - それ以外の誤りは *G(x)*に依存

標準化されたCRC (1)

- 代表的な標準
- CRC-12
  - $x^{12}+x^{11}+x^3+x^2+x+1$
- CRC-16
  - $x^{16}+x^{15}+x^2+1$
- CRC-32
  - $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
- CRC-CCITT
  - $x^{16}+x^{12}+x^5+1$

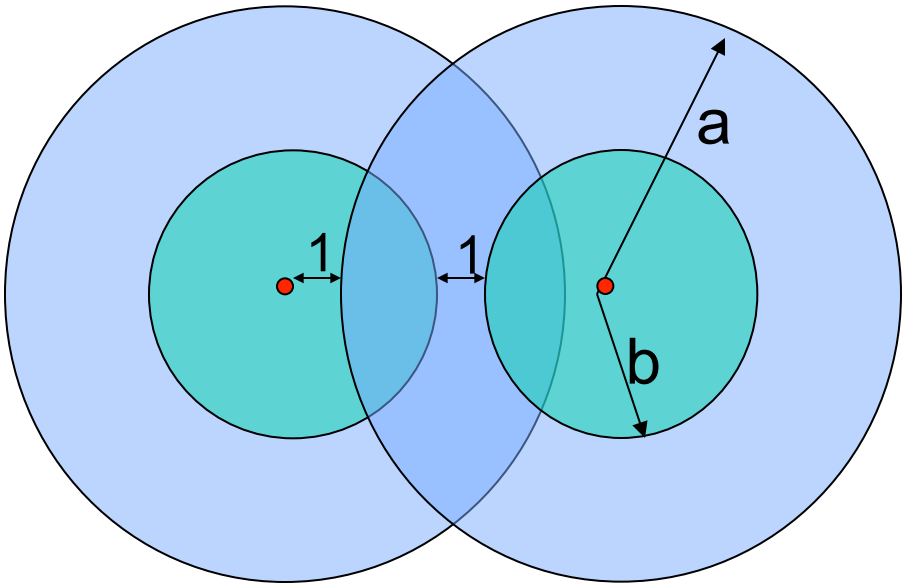
- 上記以外にも標準化されたCRCがいくつかある

Standardized CRC (2)

- 用途
  - CRC-12 は6ビット文字表現に対して使用
  - CRC-16 と CRC-CCITT は8ビット文字表現に対し有効
- 検出特性(CRC-16 and CRC-CCITT)
  - 全ての単一および二ビット連続誤り
  - 全ての奇数個のビット誤り
  - 16ビット以下のバースト誤り
  - 99.997%の17ビットバースト誤り
  - 99.998%18ビット以上バースト誤り

誤り訂正符号

- Error Correction Code
  - ハミング符号(Hamming code)の利用
  - 符号語間の距離
    - a個の誤り検出：a+1以上
    - b個の誤り訂正：2b+1以上



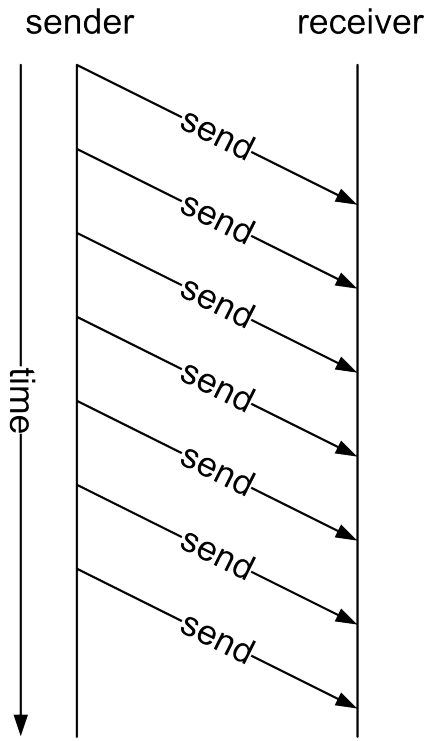
Flow Control

Flow Control

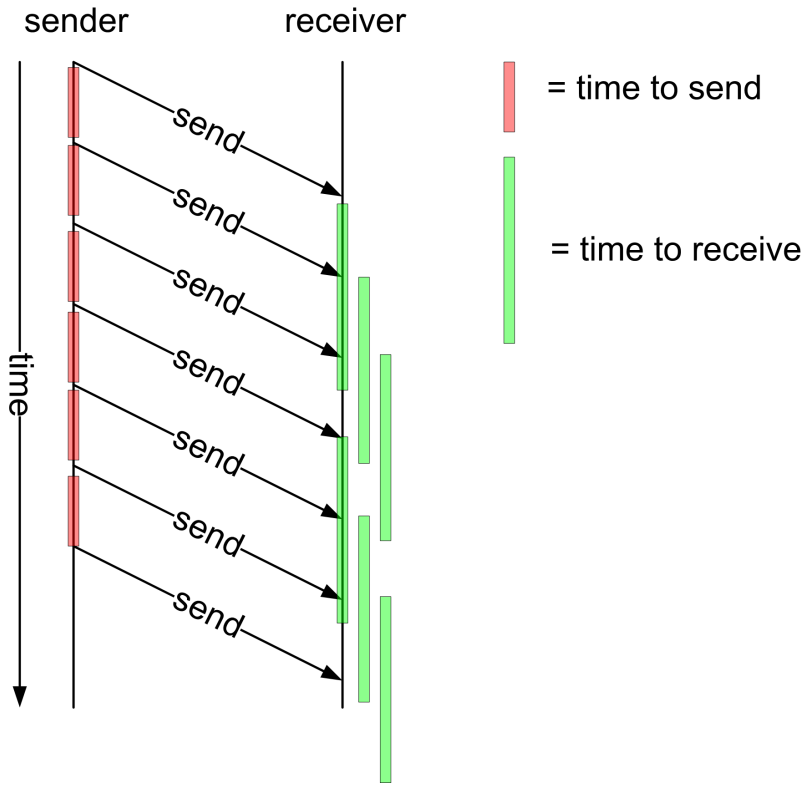
- 連続するDLLフレームをどのように送信するかを扱う
- 2つの役割:
  - 消失フレームの回復
  - バッファオーバフローを回避
- ネットワーク層は送信したフレーム量と同じ量を受信しなければならない
- Automatic Repeat Request (ARQ)
  - Stop-and-wait
  - Go-back-N
  - Selective-repeat

制限しない(Unrestricted)場合

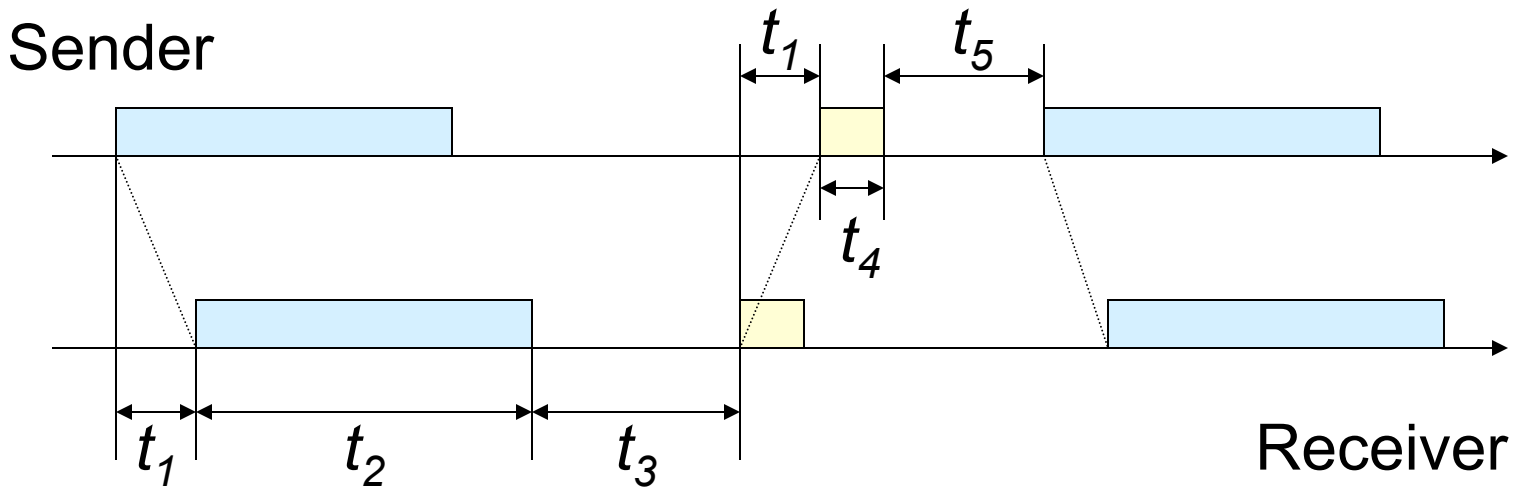
- 送信側は送信する
- 受信側は受信するのを待つ



- 問題：受信側のバッファが溢れる可能性



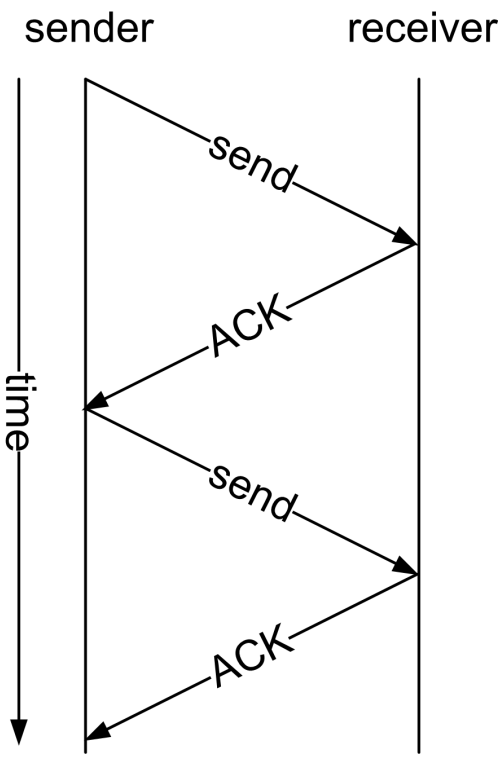
Stop-and-wait ARQ (1)



t_1: Round Trip Time
t_2: Frame Transmission Time
t_3: Frame Processing Time
t_4: ACK Transmission Time
t_5: ACK Processing Time

Stop-and-wait ARQ (2)

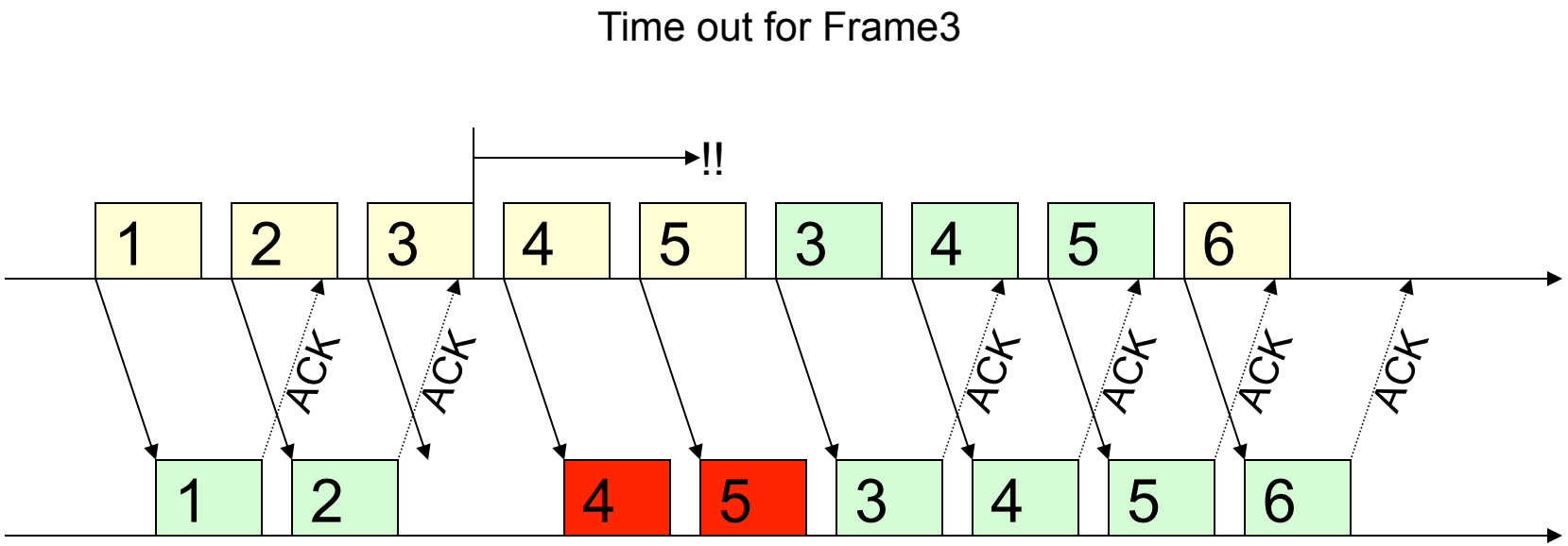
- 手法
  - 1フレーム送信毎にACKフレームを待つ
  - 送信側タイマは$2t_1+t_2+t_3+t_4$以上に設定
  - 送信側タイマがタイムアウトするとフレームを再送
- 特徴
  - 単純
  - バッファは送信/受信側それぞれで1フレーム分でよい
  - 回線使用効率は非常に悪い



Go-back-N ARQ (1)

- 手法
  - Stop-and-Wait Protocolを改善
  - 連続してフレームを送出
  - 誤りが発生した場合
    - 誤りが発生したフレームを正しく受信するまで、受信したフレームを破棄
    - 正しく受信するまでACKも返さない

Go-back-N ARQ (2)



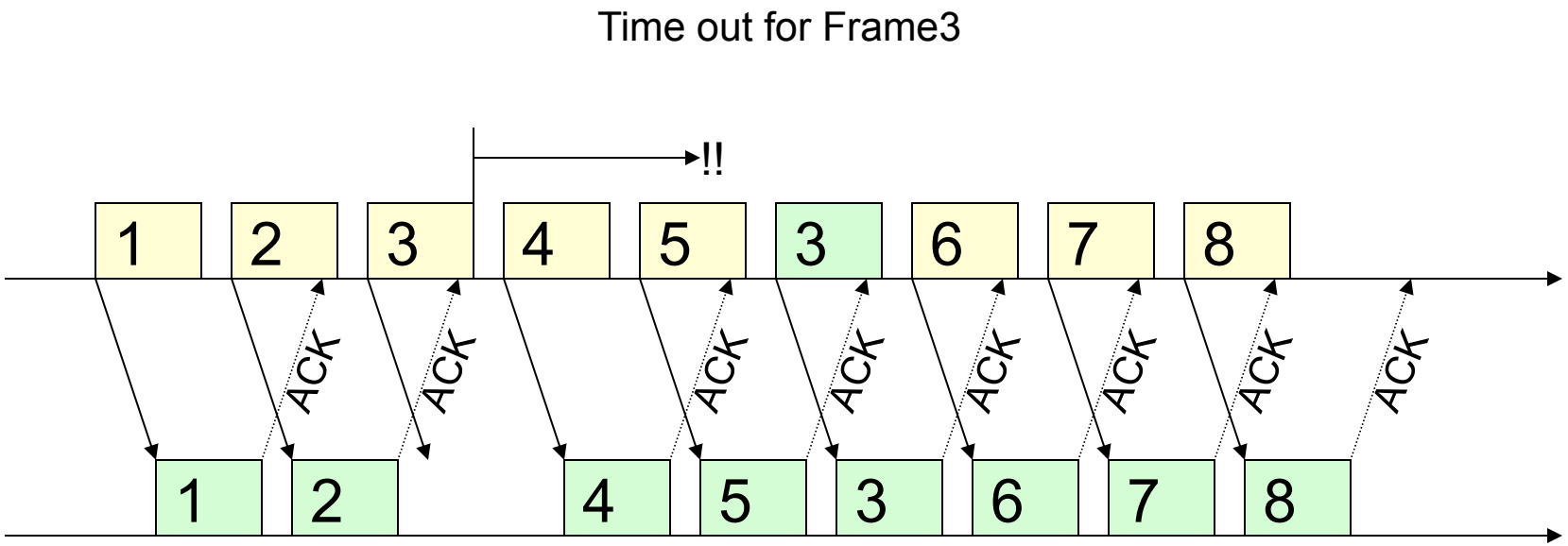
Go-back-N ARQ (3)

- 特徴
  - 誤りが発生すると正しく送られたフレームまでもが破棄され、無駄が大きい
    - 特に再送タイマを大きく設定する場合に不利
  - 送信側で再送のためのバッファが必要
    - ACKを受け取ってないフレームについては再送のためにバッファリングしておく必要あり
    - Go-back-Nの”N”はACKを待たずに送ることができるフレーム数 (on-the-fly frame数)
    - バッファは再送タイマの値に依存

Selective-Repeat ARQ (1)

- 方式
  - Go-back-Nプロトコルを改良
  - 必要なフレームだけを再送
- 特徴
  - 伝送効率がGo-back-Nより良い
    - 正しく受信されているフレームを無駄に破棄しない
  - 処理が複雑
    - 受信側でバッファが必要
    - 順番制御(フレームシーケンス制御)が必要

Selective-Repeat ARQ (2)



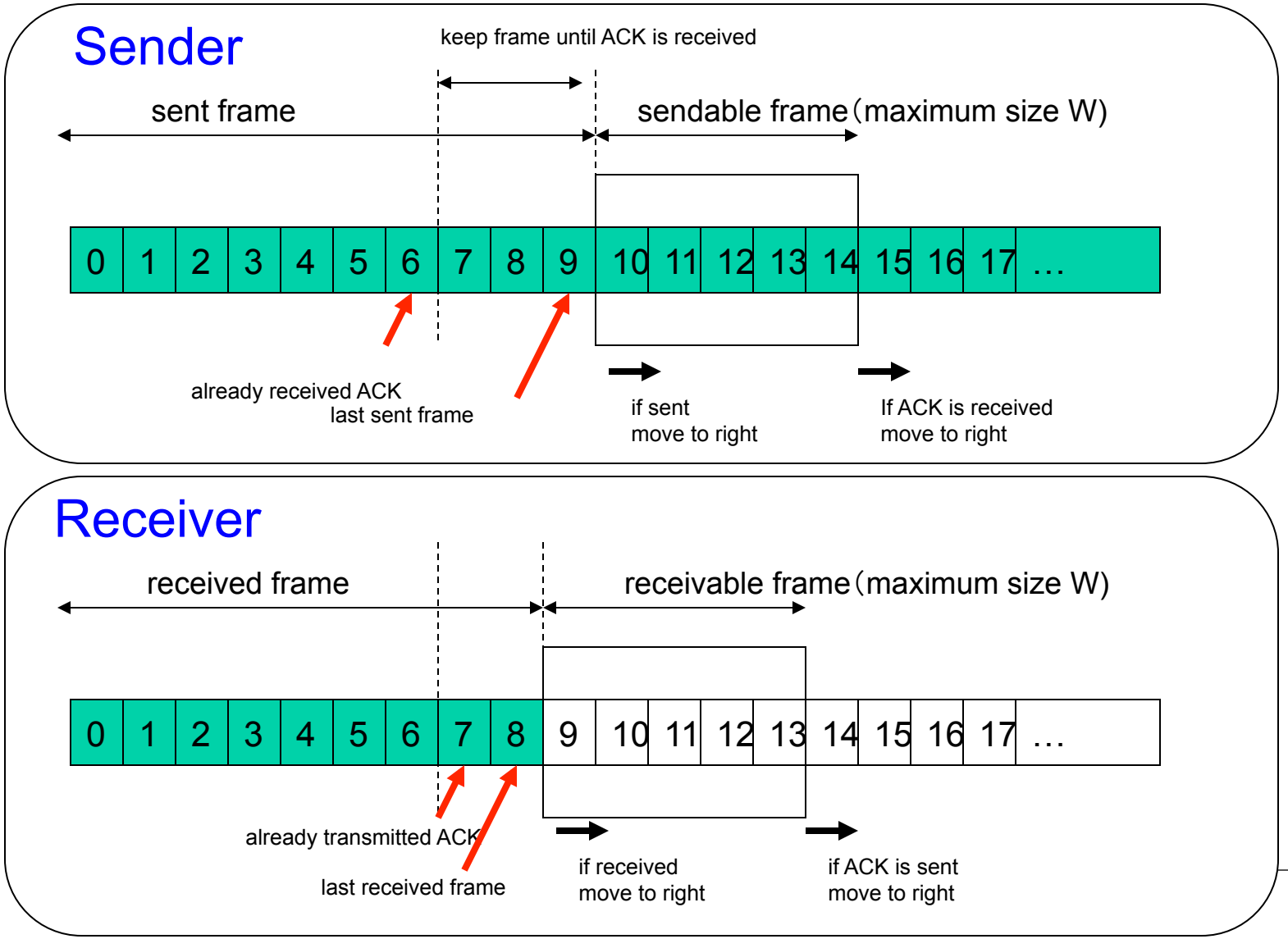
フロー制御

- Stop-and-WaitやGo-back-Nでは受信側は複雑な処理が発生しない
- Selective-Repeatでは受信側で順番制御
  - 受信側でバッファが存在
  - 送信側がうまく送らないと受信側バッファで溢れが発生する可能性あり
- フロー制御(Flow control)
  - 受信側と何らかの手法を用いて、バッファ溢れを抑制する

Window Flow Control

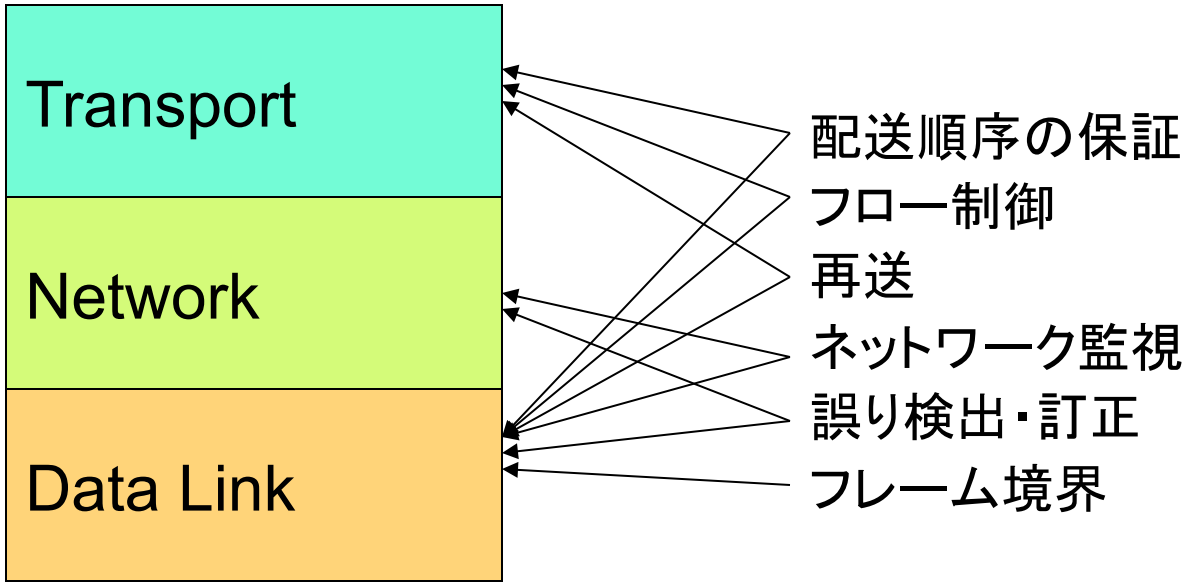
- ウィンドウ・フロー制御
  - W :ウィンドウサイズ
    - 受信側のバッファサイズと同じ
  - 送信側にて (1)素手に送信したフレームで、(2)ACKを受け取っていないフレーム数(バイト数)をNrとする
  - Nr < Wは送信を行う
  - Sliding window flow control

Windows Flow Control



機能分担

- 機能分担(Role allocation) :システムデザインに依存
- 方法は複数存在する

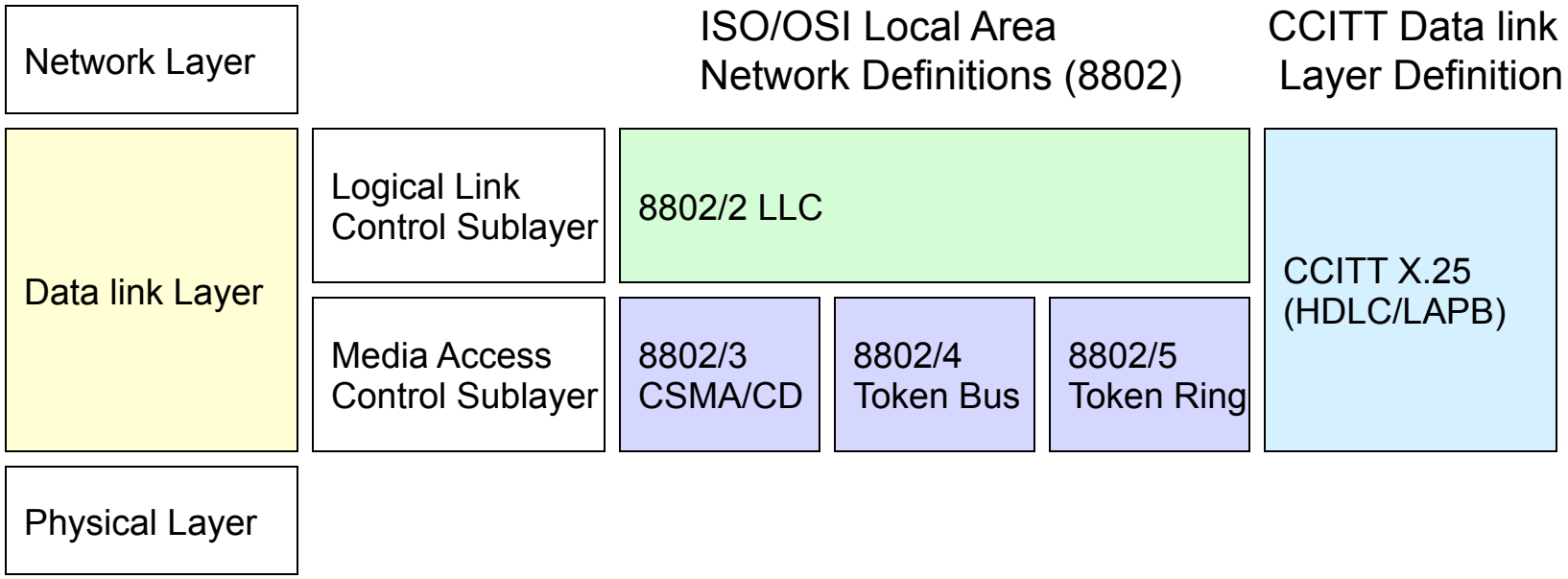


データリンクの副層

単一構造からサブレイヤ構造へ

- 1960年代の一対一接続(point-to-point connection)型のデータリンクの利用技術から、データリンクの基礎的な概念が形成された
- 1970年代後半からのLAN(Local Area Network)の標準化がサブレイヤ構造へ移行した
  - ISO/OSI 8802、IEEE802グループのアーキテクチャ
  - 共通のIEEE802.2 (ISO8802/2) として構成された Logical Link Control (LLC)
  - 各伝送メディアに特化されたIEEE802.x

Sublayers of the Data Link Layer



Logical Link Control (LLC) (1)

- Unacknowledged Connectionless Service (LLC type1)
  - 送信者が受信者と同期することなくフレームを送信
  - 誤りが発生しても特別な処理をしない
    - 上位層が誤り制御を実施
  - 高速、エラーの少ないリンク向けの制御方式
  - LAN, real-time traffic (voice/speech)

- Acknowledged Connection-oriented Service (LLC type2)
  - データ送信前にコネクションを設定
    - 送信側と受信側が同期した処理が可能
  - 信頼性 (誤り、順番)を保証
  - 処理は一般に複雑
  - 低速、エラーの多いリンクに有効

Logical Link Control (LLC) (2)

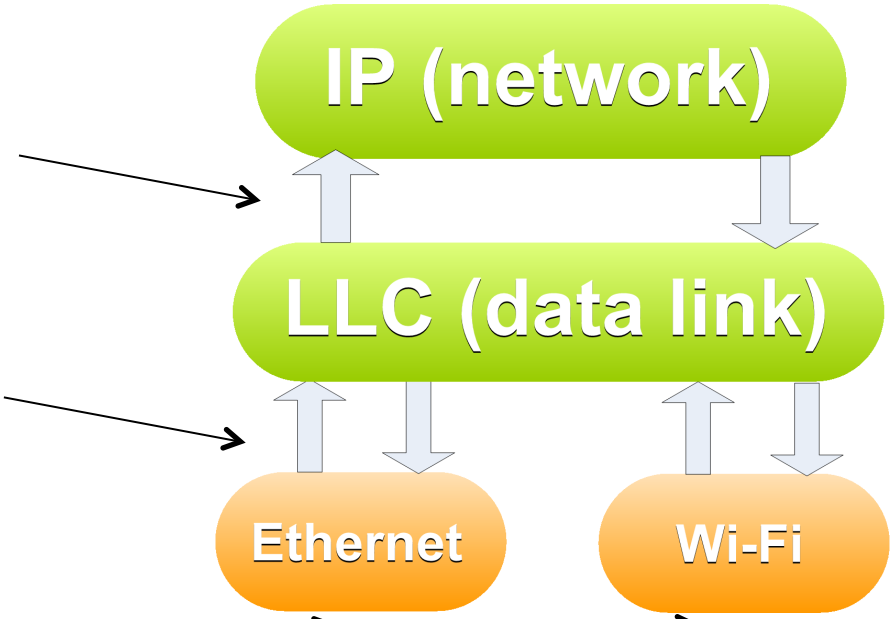
- Acknowledged connectionless Service (LLC type3)
  - 受信者は各フレーム毎に受信確認(acknowledge)
  - 誤り制御をデータリンク層で実施
  - 信頼性を向上
- Unacknowledged Connection-oriented Service
  - 信頼性(誤り、順番)を保証しない
  - 高速でエラーの少ないリンク向けの制御方式

Media Access Control (MAC) (1)

データリンク層はネットワーク層に向けて
パケット送受信機能を提供

物理層はデータリンク層へ向けた
バイナリ送受信機能を提供

複数のノードがメディア媒体へアクセス
する際、メディア媒体はそれぞれ異な
る制約を持つ

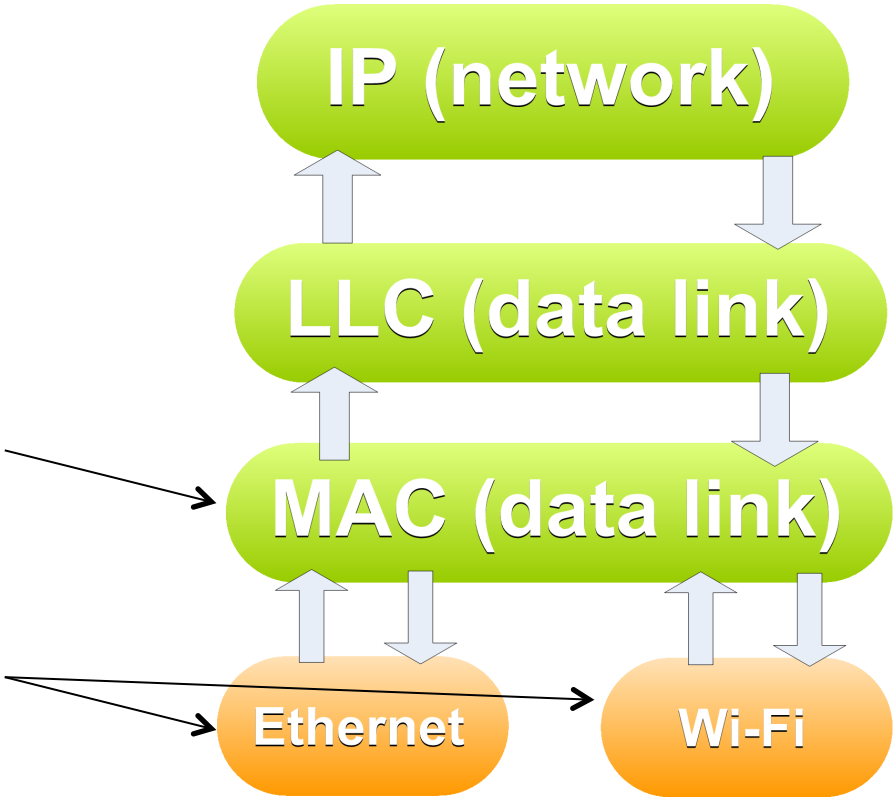


メディア媒体は長時間確保しておきたい

Media Access Control (MAC) (2)

MAC層はデータリンク層に向けて
媒体へのアクセス機能を提供

異なる伝送媒体にはそれぞれ
別々のプロトコルが必要



次回：チャネル割り当て(multiplexing)