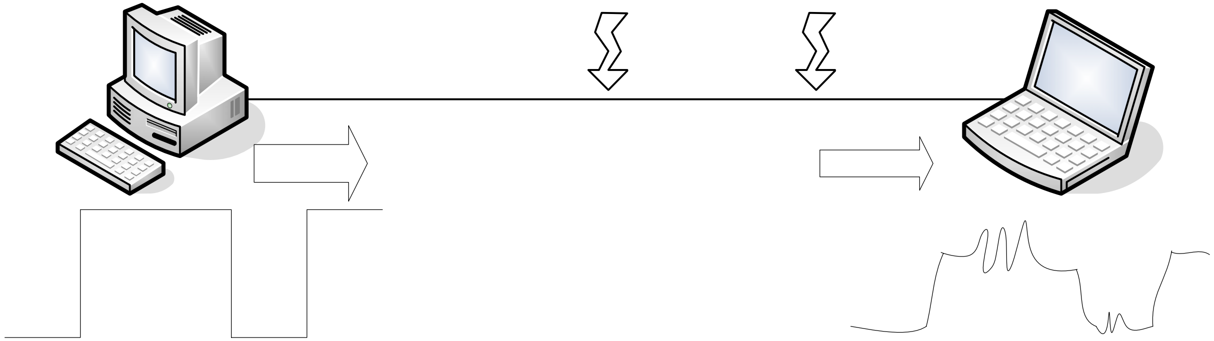
2.2 物理層

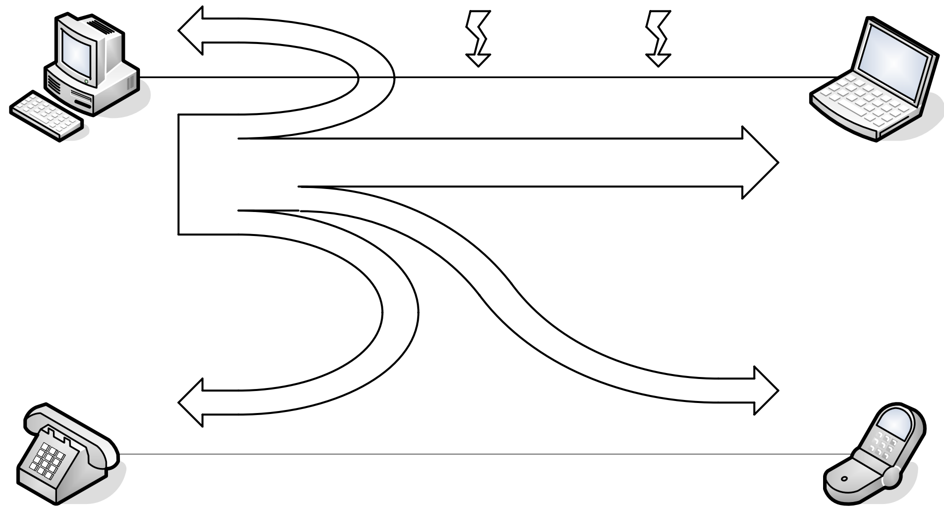
どれだけ情報を送ることができるか？



- 伝送中に信号が減衰する
- 周波数によって減衰量が異なるため波形が歪む
- 周囲からの雑音が混入する
- 広い周波数帯域を使えば速く送れる

→信号対雑音比を周波数軸上で積分（シャノンの原理）

雑音



- 歪み・エコー
  - きちんと計算すれば除去可能
- 近端漏話・遠端漏話
  - チャンネルを分けることで対策可能
- 周囲からの雑音
  - 一旦混入すると除去困難

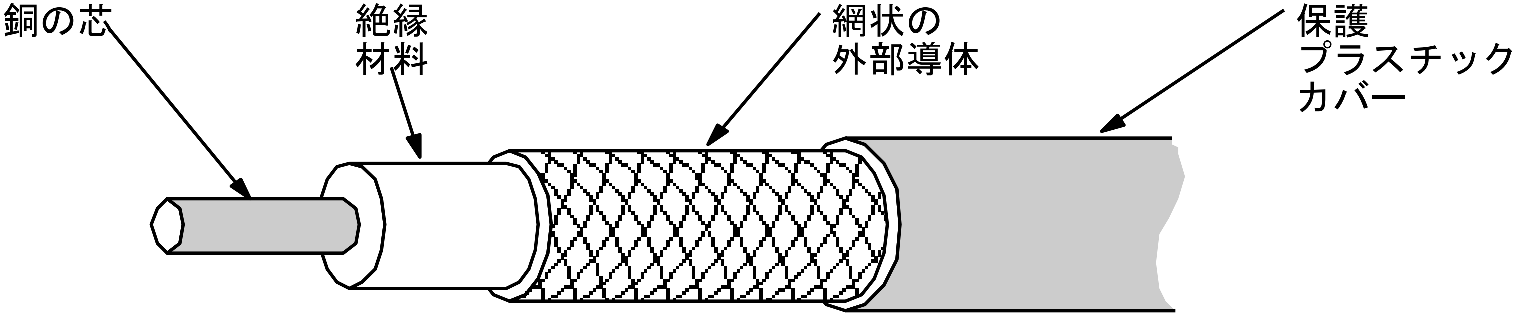
平行ケーブル

- 屋内電話線
- コンピュータとモデムをつなぐケーブル
- 数十kbps、十数m

より対線

- 電話線（日本ではカッド構造）
  - ADSL: 数Mbps、数km
  - VDSL: 数十Mbps、100m程度
- LANケーブル
  - カテゴリ3: 10Mbps程度、100m
  - カテゴリ5: 100Mbps程度、100m
  - エンハンストカテゴリ5、カテゴリ6: 1Gbps

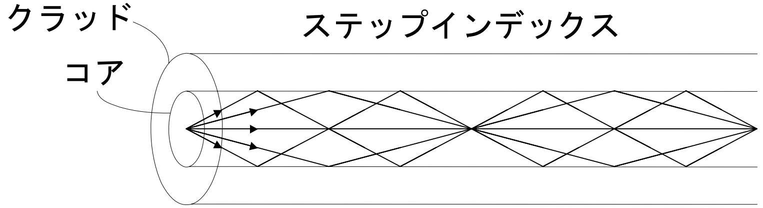
同軸ケーブル



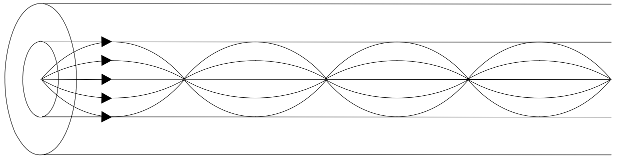
- Ethernet: 10Mbps、500m
- CATV: アナログ750MHz、数km

光ファイバ

- ステップインデックス

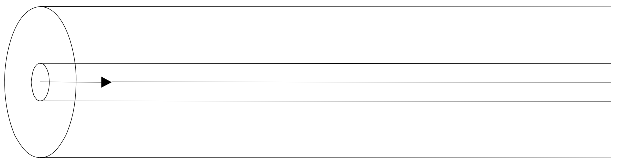


- グレーデッドインデックス
  - コア径50μmまたは62.5μm
  - 100Mbps、2km
  - 1Gbps、数百m



光ファイバ（続き）

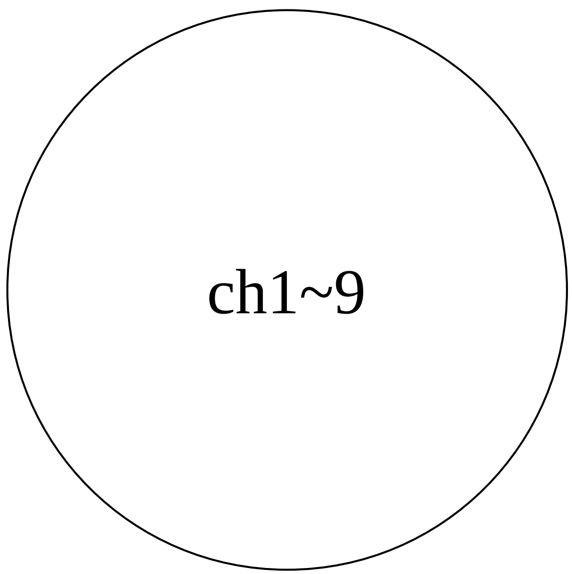
- シングルモード
  - コアア径9μm程度
  - 数十Gbps、数十～数百km
- 波長多重(Wavelength Division Multiplex)
  - 最大数百波長を1本の芯線で伝送可能

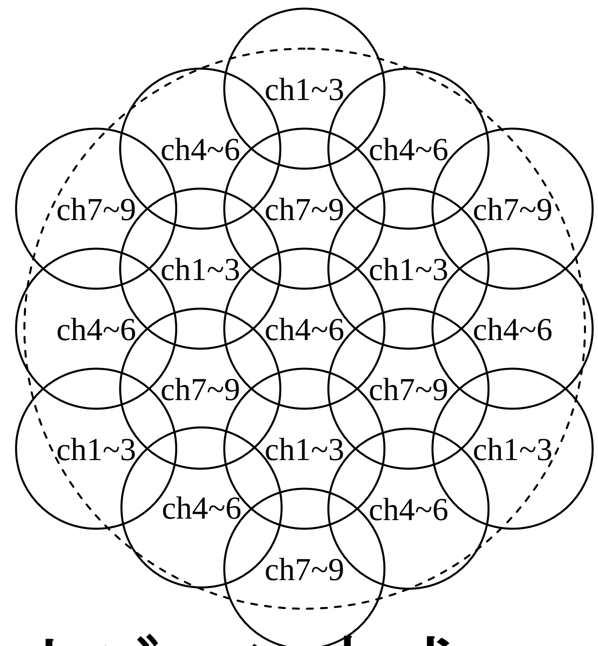


無線

- 携帯電話
  - PDC: 28800bps、数km
  - PHS: 128kbps、数百m
  - 3G: 最大数Mbps 、数km
- 無線LAN
  - IEEE 802.11b: 2.4GHz帯、11Mbps、数十m
  - IEEE 802.11a: 5GHz帯、54Mbps、屋内のみ
  - IEEE 802.11g: 2.4GHz帯、54Mbps、屋外可

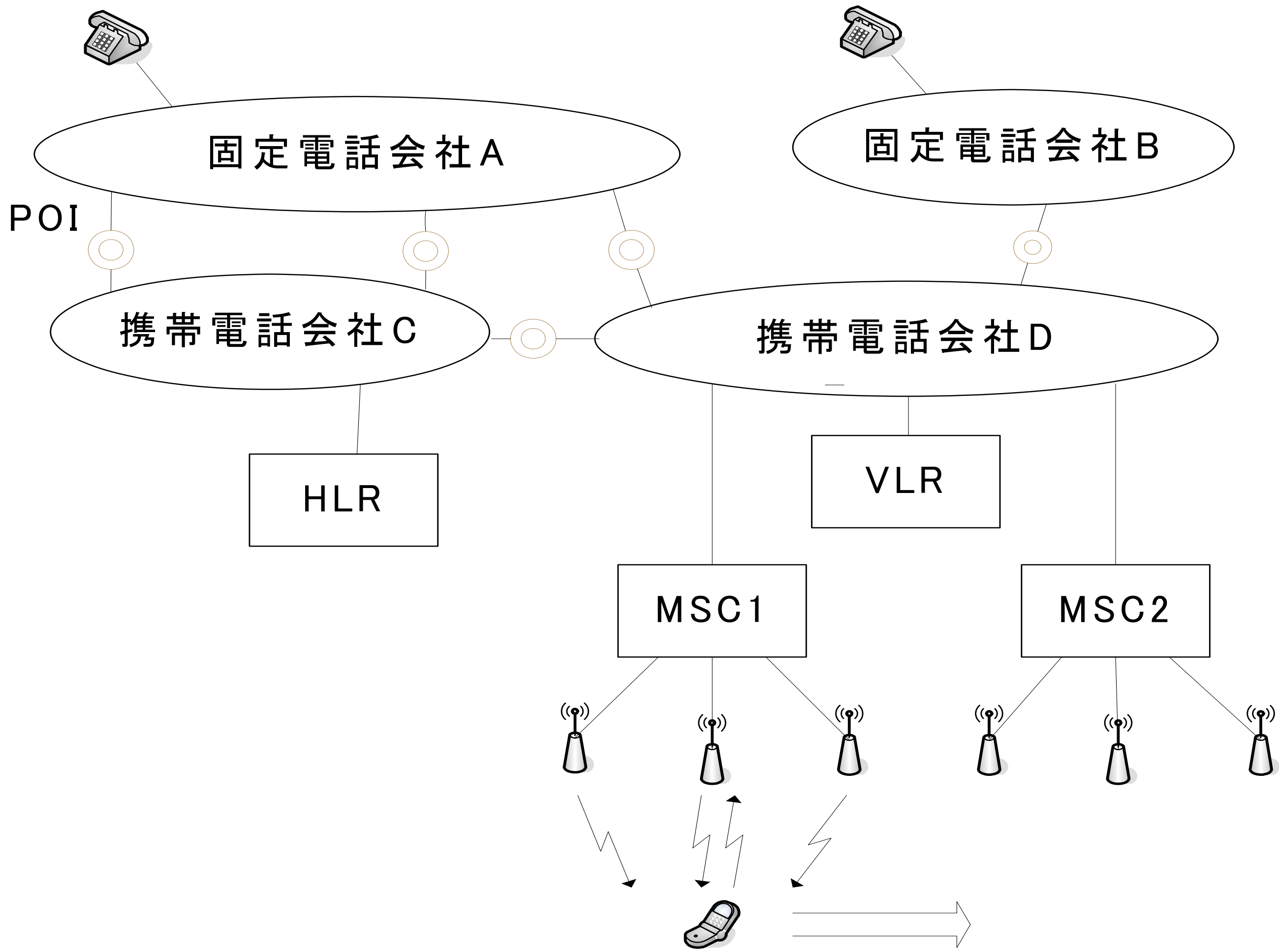
小ゾーン方式





- 大ゾーン方式
  9人までサービス可能
- 小ゾーン方式
  57人までサービス可能

位置登録



CDMA
(Code Division Multiple Access)

- 1ビットを*m*チップ使って伝送（*m*=128程度）
- 各局に固有のチップ列を割り当てる
  - A:00011011 (-1 -1 -1 +1 +1 -1 +1 +1)
  - B:00101110 (-1 -1 +1 -1 +1 +1 +1 -1)

  →A+(-B): (0 0 -2 +2 0 -2 0 +2)
- 各局は受信信号と自局のチップ列の内積を取る
  - A: 0+0+2+2+0+2+0+2=8 → 1と認識
  - B: 0+0+(-2)+(-2)+0+(-2)+0+(-2)=-8 → 0と認識

2.3 データリンク層

データリンク層のおもな機能

- 同期・誤り制御
  – フレーム化
- 媒体アクセス制御
  – 通信媒体が共有されている場合（LAN、無線）

誤り検出符号、誤り訂正符号

- mデータビット＋r冗長ビット
- 符号語間のハミング距離：対応するビットが何箇所異なっているか
- 符号全体のハミング距離：符号語間のハミング距離の最小値
- dビットの誤り検出→距離d+1が必要
- dビットの誤り訂正→距離2d+1が必要

単一誤り訂正：ハミング符号

- 2^mの符号語のそれぞれに対してn個の誤りが対応→$(n+1)2^m \leqq 2^n$→$(m+r+1) \leqq 2^r$
- $2^{r-1} \leqq n \leqq 2^r-1$のとき、2のべき乗の位置のビットをチェックビットに使う

例：$x_1=x_3+x_5+x_7+x_9+x_{11}$

$x_2=x_3+x_6+x_7+x_{10}+x_{11}$

$x_4=x_5+x_6+x_7+x_{12}$

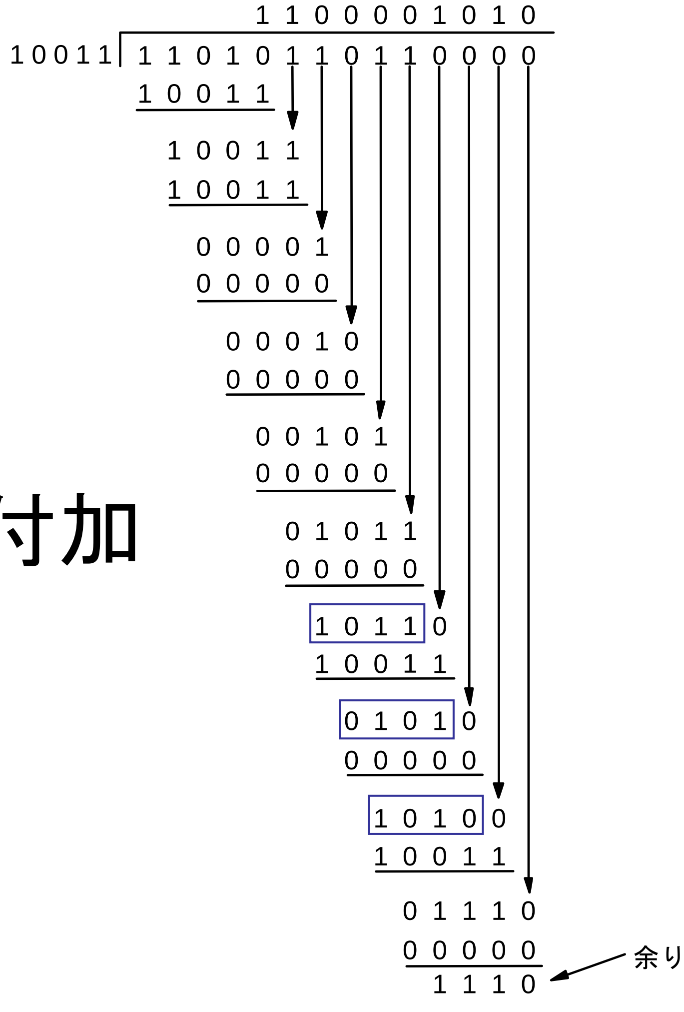
$x_8=x_9+x_{10}+x_{11}+x_{12}$

Cyclic Redundancy Check

- 偶数パリティ：偶3進数に対応
  - $10110001 = 3^7 + 3^5 + 3^4 + 1 = 2512$
- nビットの符号語をn-1次式と見なす
  - $a_1 x^{n-1} + a_2 x^{n-2} + \ldots + a_{n-1} x + a_n$
- 符号語がr次の生成多項式$G(x)$で「割り切れる」ようにする
  - $x^r M(x)$を$G(x)$で割った余りを$x^r M(x)$から引く

Cyclic Redundancy Checkの例

- 送るべきデータ
  1 1 0 1 0 1 1 0 1 1
- 生成多項式
  1 0 0 1 1
- データに0を4ビット付加
  1 1 0 1 0 1 1 0 1 1 0 0 0 0
- 余りを引いて送信
  1 1 0 1 0 1 1 0 1 1 1 1 1 0



Cyclic Redundancy Check

- 誤り$E(x)$が$G(x)$で割り切れなければ検出可能
  - $G(x)=(x+1)Q(x)$なら全ての奇数ビット誤りを検出可能
  - 長さ≦rの全てのバースト誤りを検出可能
    - $x^i(x^{k-1}+\ldots+1)$
- $G(x)$の例
  - $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

ハードウェア

- $G(x)=x^4+x+1$の例