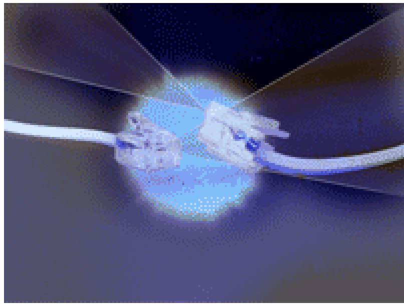
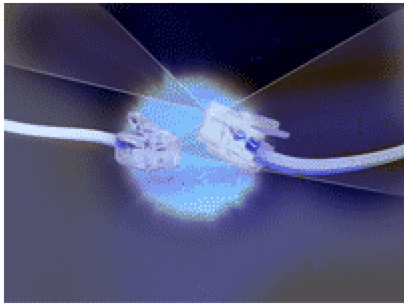


Error Detection

- Detect errors in transmitted signal by including redundant information
- Simple technique: transmit a second copy of the message
  - Discard message if two copies differ
  - Inefficient (only half transmitted bits are data)
  - Misses error if same bit is corrupted in both copies

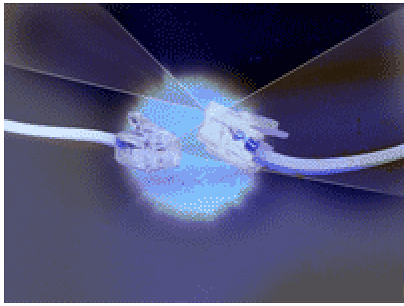
Error Detection (cont.)

- Better methods are available – send *k* bits of redundant data for *n* data bits, where $k \ll n$
  - In Ethernet, frames of up to 12,000 bits require only 32 bits of extra data
- Data is redundant because it must be computable from message data, using an algorithm common to sender and receiver
- An error-detecting code is any group of extra bits added to message
  - A *checksum* is a special case that uses addition to compute the code



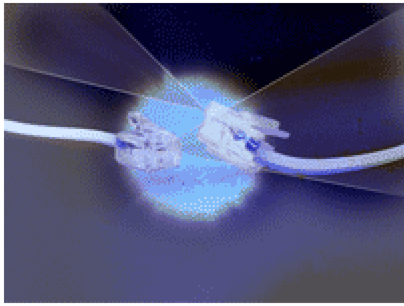
Two-Dimensional Parity

- Based on the simple parity scheme
  - Add an extra bit to a 7-bit code to balance the number of 1s in the byte (either even or odd)
- Two-dimensional parity adds a similar computation for each bit position across all bytes in the frame
  - Adds one parity byte to the frame
  - Can detect all 1-, 2-, and 3-bit errors in a frame, and most 4-bit errors



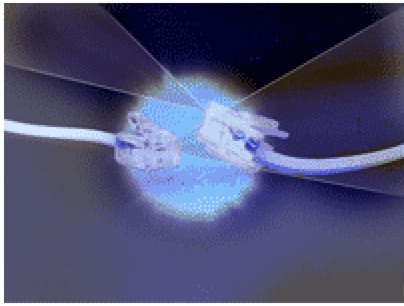
2D Parity (cont.)

- Example (using odd parity):
  - 0101010 0
    1100110 1
    0001101 0
    1000100 1
    1111011 1
    1010010 0
    1010011 0
  - Added 14 bits to frame – one parity bit for each of the 6 data bytes, plus an eight-bit parity byte



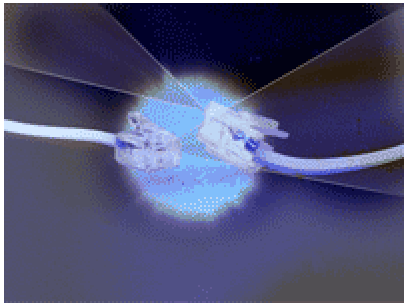
Internet Checksum

- Add up all the data in the frame, and append the resulting sum to the frame
  - Treats data as sequence of 16-bit integers
  - Uses one's complement addition
    - Carry out from MSB added to result
- Only adds 16 bits for any length frame
- Can miss some 2-bit errors
- Fast to compute, usually sufficient (because a better code used at link level)



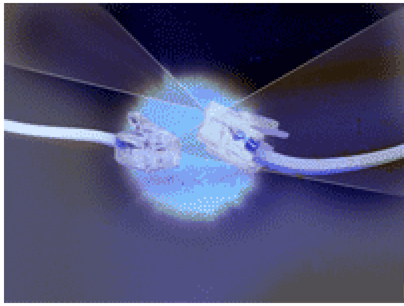
Cyclic Redundancy Check (CRC)

- Based on finite-field mathematics
- Consider (n+1)-bit message as representing a degree-n polynomial
  - Each bit is coefficient of corresponding power of x – MSB is power of highest-order term
  - For example, 100101 represents $x^5 + x^2 + 1$
- Also need a *divisor* polynomial, $C(x)$, with degree k



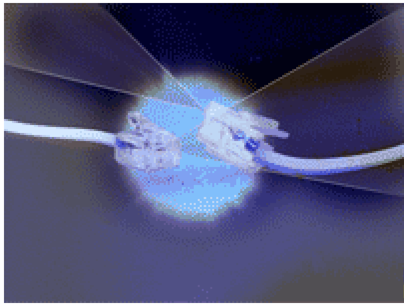
CRC (cont.)

- Compute transmission *P(x)*, which is *n*+1 bit message *M(x)* with *k* redundant bits added
- Choose error check code to make *P(x)* evenly divisible by *C(x).*
  - Receiver can compute *P(x)* / *C(x)*, and if remainder is 0, message is error-free
- Use *modulo 2* arithmetic
  - *B(x)* can be divided by *C(x)* if degree of *B* is >= degree of *C*
  - Remainder obtained by subtracting modulo 2 (XOR)



CRC (cont.)

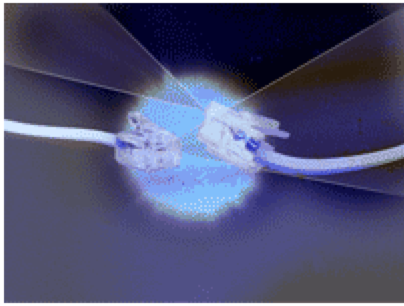
  - For example, the remainder of 10010 / 11001 = 10010 – 11001 (mod 2) = 1011
- To generate $P(x)$
  - Add k 0s to $M(x)$ to form $T(x)$
  - Divide $T(x)$ by $C(x)$ and find remainder
  - Subtract remainder from $T(x)$
- This result should be evenly divisible by $C(x)$



CRC Example

- Suppose $M(x)$ = 11010, $C(x)$ = 1011
  - $T(x)$ = 110100000
  - $T(x)/C(x)$:

```
1011/110100000
     1011
      1100
      1011
       1110
       1011
        1010
        1011
         00100
          1011
          1111            (remainder)
```



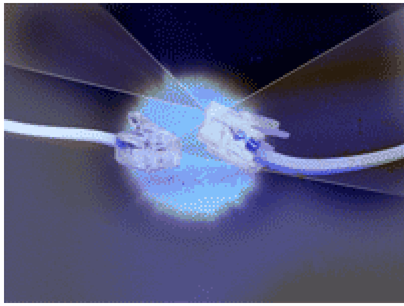
CRC Example (cont.)

- So *P(x)* = T(x) – remainder = 110101111
- Check:

```
1011/110101111
     1011
      1100
      1011
       1111
      1011
        1001
        1011
         01011
          1011
          0000
```

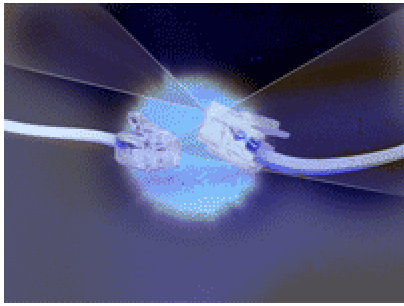
Choosing CRC Polynomial

- If receiver computes non-zero remainder, error occurred in message.
- Want to choose $C(x)$ to minimize chance that $P(x) + E(x) / C(x)$ will be 0 (if so, error would be undetected)
  - This can only happen if $E(x)$ is evenly divisible by $C(x)$
  - Choose $C(x)$ so it won’t evenly divide into common errors



Choosing CRC Polynomial (cont.)

- Types of errors:
  - Single bit (i.e. x^i) – won’t evenly divide by any $C(x)$ with 1 for first and last term
  - Double-bit errors – detected by any $C(x)$ with a factor containing at least three ones
  - Odd number of errors – detected by any $C(x)$ with the factor $(x + 1)$
  - Any burst error of $< k$ bits



Common CRC Polynomials

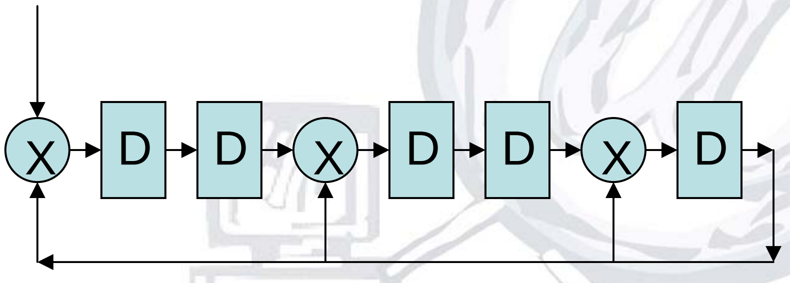
- CRC *C(x)*

CRC	*C(x)*
CRC-8	100000111
CRC-10	11000110011
CRC-12	1100000001101
CRC-16	11000000000000101
CRC-CCITT	10001000000100001
CRC-32	100000100110000010001110110110111

- Ethernet, 802.5 use CRC-32
  HDLC uses CRC-CCITT
  ATM uses CRC-8, CRC-10, CRC-32

CRC in Hardware

- Can easily implement the algorithm using a *k*-bit shift register and XOR gates
  - Example for $C(x) = x^5 + x^4 + x^2 + 1$



  - The contents of register after all message bits shifted in, with *k* 0s appended, is the CRC