



# Computer Networking

## Data-link layer

Prof. Andrzej Duda  
duda@imag.fr

http://duda.imag.fr

1

## Data Link Layer

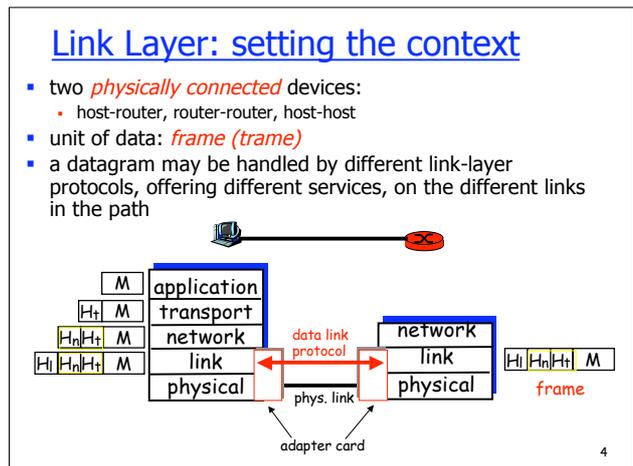
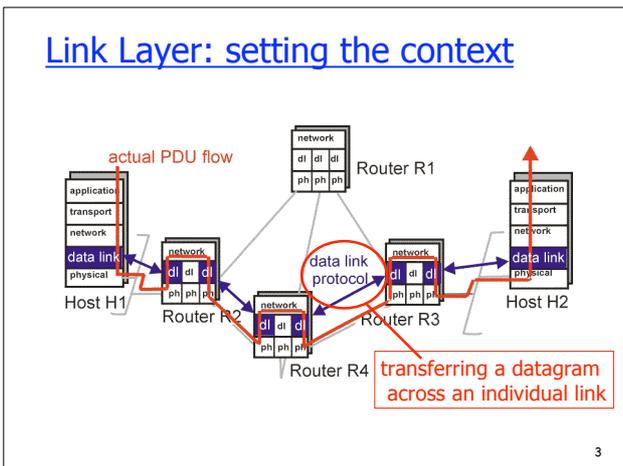
**Our goals:**

- Understand principles behind data link layer services:
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - LAN interconnection
- Instantiation and implementation of various link layer technologies

**Overview:**

- Link layer services
- Point-to-point protocol
  - PPP
- Later on:
  - LANs:
    - Ethernet
    - 802.11
  - link layer addressing, ARP
  - LAN interconnection
    - hubs, bridges, switches

2



### Link Layer Services

- Framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - implement channel access if shared medium,
  - 'physical addresses' used in frame headers to identify source, dest
    - different from IP address!
- Reliable delivery between two physically connected devices:**
  - we learned how to do this already (cf. Transport Layer)
  - seldom used on low bit error link (fiber, some twisted pair)
  - wireless links: high error rates
    - link-level reliability to avoid end-end retransmission

5

### Link Layer Services (more)

- Flow Control**
  - pacing between sender and receivers
- Error Detection**
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- Error Correction**
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- Half-duplex and full-duplex**

6

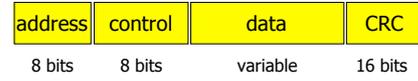
## Generating polynomials

- CCITT
  - $g(x) = x^{16} + x^{12} + x^5 + 1$ 
    - detects all simple errors, doubles, odd number of errors, bursts of 16 and less, 99.997% of 17, 99.998% of 18 or more
  - $g(x) = x^{16} + x^{15} + x^2 + 1$
- Ethernet
  - $g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

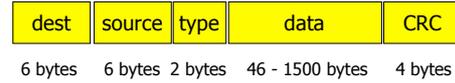
7

## Examples of protocols

- CRC - *Cyclic Redundancy Check*
- X.25 - LAPB, PPP, IEEE 802 LLC



- Ethernet



8

## Error detection in TCP/IP

- Checksum TCP/IP*
  - sum of 16 bit words with carry in 1's complement
  - carry is added
  - 1's complement
- Characteristics
  - detects all simple errors
  - Undetected Packet Error Rate
    - 1 in  $16 \cdot 10^6$  to 1 in  $10^{10}$

9

## Example

7 bit words:

```

0000010
1011011
1101100
0000011
11001100
-----
1001101
0110010
    
```

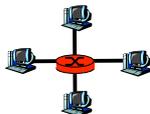
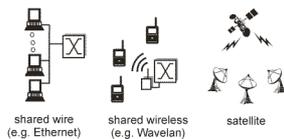
A curved arrow points from the carry bit '1' in the second row of the addition to the first bit of the result '1' in the third row.

10

## Multiple Access Links and Protocols

Three types of "links":

- point-to-point (single wire, e.g. PPP)
- broadcast (shared wire or medium; e.g. Ethernet, 802.11)



- NBMA - Non Broadcast Multiple Access (ATM, X.25)

11

## Point to Point Data Link Control

- One sender, one receiver, one link: easier than a broadcast link:
  - no Media Access Control
  - no need for explicit MAC addressing
  - e.g., dialup link, ISDN line
- Popular point-to-point DLC protocols:
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link used to be considered "high layer" in protocol stack!)

12

### PPP (*Point-to-Point Protocol*)

- Point to point data link
  - dial-up (modem) connexions, fiber (SONET/SDH)
- Data framing, error detection
- Transparent data transmission
  - avoid transmitting characters interpreted by the modem
- Data
  - multi-protocol: IP packets, IPX packets, others
- Header compression - IP, TCP
- Authentication

13

### Associated protocols

- LCP (*Link Control Protocol*)
  - activate a link
  - negotiate options
  - test
- IPCP (*IP Control Protocol*)
  - network layer address negotiation:
    - hosts/nodes across the link must learn/configure each other's network address
- PAP (*PPP Authentication Protocol*)
  - password exchange

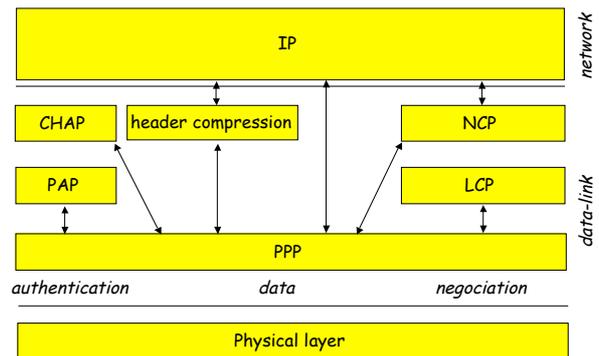
14

### Associated protocols

- CHAP (*Challenge Handshake Authentication Protocol*)
  - server sends a challenge (random number)
  - dial-up host encrypts it using a common secret password
  - sends the results
  - server does the same and compares
  - call NCP (*Network Control Protocol*) to finish with network level configuration
- NCP (*Network Configuration Protocol*)
  - network layer negotiation
  - depends on the network protocol
- IPCP (*IP Configuration Protocol*)
  - NCP for IP: configure the network layer
  - address negotiation
    - assign a temporary IP address to dial-up host
  - decide whether to use IP/TCP header compression

15

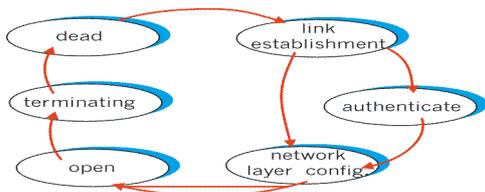
### Protocols



16

### PPP data control protocol

- PPP-LCP establishes/releases the PPP connection; negotiates options
- Starts in DEAD state
- Options: max frame length; authentication protocol, call-back
- Once PPP link established, IPCP moves in (on top of PPP) to configure IP network addresses etc.



17

### Data framing

- Frame format inherited from first packet switching networks
  - synchronous transmission with error recovery (HDLC)
  - Transpac (X.25 LAPB)
- PPP
  - start and end of frame delimitation
  - transparent data transmission
    - how to transmit characters used for delimitation?
    - byte stuffing
  - error detection - polynomial code

18

## PPP - frame formats

| flag   | address | control | protocol              | data | CRC     | flag   |
|--------|---------|---------|-----------------------|------|---------|--------|
| 8 bits | 8 bits  | 8 bits  | 8 or 16 bits variable |      | 16 bits | 8 bits |

- Flag: 01111110 - 0x7e
- Address: 11111111
- Control: 00000011
- Protocol: LCP, IPCP, IP
- Data: 1500 bytes by default
- CRC: 16 bit CCITT polynomial code

19

## PPP - frame formats

| flag   | address | control | protocol              | data | CRC     | flag   |
|--------|---------|---------|-----------------------|------|---------|--------|
| 8 bits | 8 bits  | 8 bits  | 8 or 16 bits variable |      | 16 bits | 8 bits |

- Transparent transmission - Byte stuffing
  - 0x7e → 0x7d 0x5e
  - 0x7d → 0x7d 0x5d
  - character X < 0x20 → 0x7d 0xYY
  - 0xYY = X + 0x20
  - example : 0x03 → 0x7d 0x23

20

## Example PPP connection (logs)

- PPPoE connecting to service
- PPPoE connection established.
- Connect: ppp0 <--> socket[34:16]
- sent [LCP ConfReq id=0x1 <mru 1492> <asynctest 0x0> <magic 0x1f3f807b> <pcomp> <accomp>]
- rcvd [LCP ConfReq id=0x81 <mru 1500> <auth chap MD5> <magic 0x4b4dcf02>]
- sent [LCP ConfAck id=0x81 <mru 1500> <auth chap MD5> <magic 0x4b4dcf02>]
- rcvd [LCP ConfRej id=0x1 <asynctest 0x0> <pcomp> <accomp>]
- sent [LCP ConfReq id=0x2 <mru 1492> <magic 0x1f3f807b>]

21

## Example PPP connection (logs)

- ConfReq: ask for modification of default values
- ConfAck: new values accepted
- ConfNack: new values rejected, but can be negotiated
- ConfRej: values cannot be negotiated, contains acceptable values
- MRU: maximum reception unit
- id: match Req with Rep
- magic: detect loops between two equipments (should be different for both sides)
- asynctest: transcoding map Asynchronous Control Character Map: a series of bits, if 1 - the character is transcoded using 0x7d escape character, if 0 - no transcoding
- pcomp: protocol field compression (on 1 byte)
- accomp: access and control fields compression

22

## Example PPP connection

- rcvd [LCP ConfAck id=0x2 <mru 1492> <magic 0x1f3f807b>]
- sent [LCP EchoReq id=0x0 magic=0x1f3f807b]
- rcvd [CHAP Challenge id=0x1 <1a5c7a4446bf4a51bc15b170dfbb66ae>, name = "BSGRE102"]
- ChapReceiveChallenge: rcvd type CHAP-DIGEST-MD5
- sent [CHAP Response id=0x1 <2c99f036ee3190e0364d6bad10e4b971>, name = "monlogin@isp"]
- rcvd [LCP EchoRep id=0x0 magic=0x4b4dcf02]
- rcvd [LCP ConfReq id=0x12 <auth pap> <magic 0x794cb2ad>]
- sent [LCP ConfReq id=0x3 <mru 1492> <asynctest 0x0> <magic 0x5893ae1e> <pcomp> <accomp>]
- lcp\_reqci: returning CONFACK.

23

## Example PPP connection

- sent [LCP ConfAck id=0x12 <auth pap> <magic 0x794cb2ad>]
- rcvd [LCP ConfNak id=0x3 <mru 1500>]
- sent [LCP ConfReq id=0x4 <asynctest 0x0> <magic 0x5893ae1e> <pcomp> <accomp>]
- rcvd [LCP ConfAck id=0x4 <asynctest 0x0> <magic 0x5893ae1e> <pcomp> <accomp>]
- sent [LCP EchoReq id=0x0 magic=0x5893ae1e]
- sent [PAP AuthReq id=0x1 user="monlogin@isp" password=<hidden>]
- rcvd [LCP EchoRep id=0x0 magic=0x794cb2ad]
- rcvd [PAP AuthAck id=0x1 ""]
- sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <compress VJ Of 01>]
- rcvd [IPCP ConfReq id=0x48 <addr 192.168.254.254>]
- ipcp: returning Configure-ACK

24

## Example PPP connection

- sent [IPCP ConfAck id=0x48 <addr 192.168.254.254>]
- rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
- sent [IPCP ConfReq id=0x2 <addr 0.0.0.0>]
- rcvd [IPCP ConfNak id=0x2 <addr 82.65.101.110>]
- sent [IPCP ConfReq id=0x3 <addr 82.65.101.110>]
- rcvd [IPCP ConfAck id=0x3 <addr 82.65.101.110>]
- ipcp: up
- local IP address 82.65.101.110
- remote IP address 192.168.254.254

25

## Example PPP connection (traces)

- Point-to-Point Protocol
  - Address: 0xff
  - Control: 0x03
  - Protocol: IP (0x0021)
- Internet Protocol, Src Addr: 62.147.72.195 (62.147.72.195), Dst Addr: 129.88.38.1 (129.88.38.1)
  - Protocol: TCP (0x06)
- Transmission Control Protocol, Src Port: 53475 (53475), Dst Port: ssh (22), Seq: 1928148501, Ack: 0, Len: 0

26

## Example PPP connection (traces)

- Point-to-Point Protocol
  - Address: 0xff
  - Control: 0x03
  - Protocol: IP (0x0021)
- Internet Protocol, Src Addr: 129.88.38.1 (129.88.38.1), Dst Addr: 62.147.72.195 (62.147.72.195)
  - Protocol: TCP (0x06)
- Transmission Control Protocol, Src Port: ssh (22), Dst Port: 53475 (53475), Seq: 1558034509, Ack: 1928148502, Len: 0

27

## Not provided by PPP

- error correction/recovery
- flow control
- sequencing

28

## Data Link Layer: Summary

- Principles behind data link layer:
  - structure the information sent over the wire
    - frame structure
  - may add some functions
    - TCP/IP: error and flow control done at upper layers
  - using a link between two connected devices
    - point-to-point
    - sharing a broadcast channel
- PPP
  - used in many contexts
    - modems, ADSL, POS
  - authentication and accounting

29