Data Link Layer

Design Issues

- Uses the physical layer to send and receive bits over a communication channel
- Defines an interface to the network layer
- Deals with transmission errors
- Regulates the flow of data
    - unless an eager sender overwhelms a slow receiver
- IN CASE OF A SHARED CHANNEL
    - Provides Medium Access Control (MAC) services

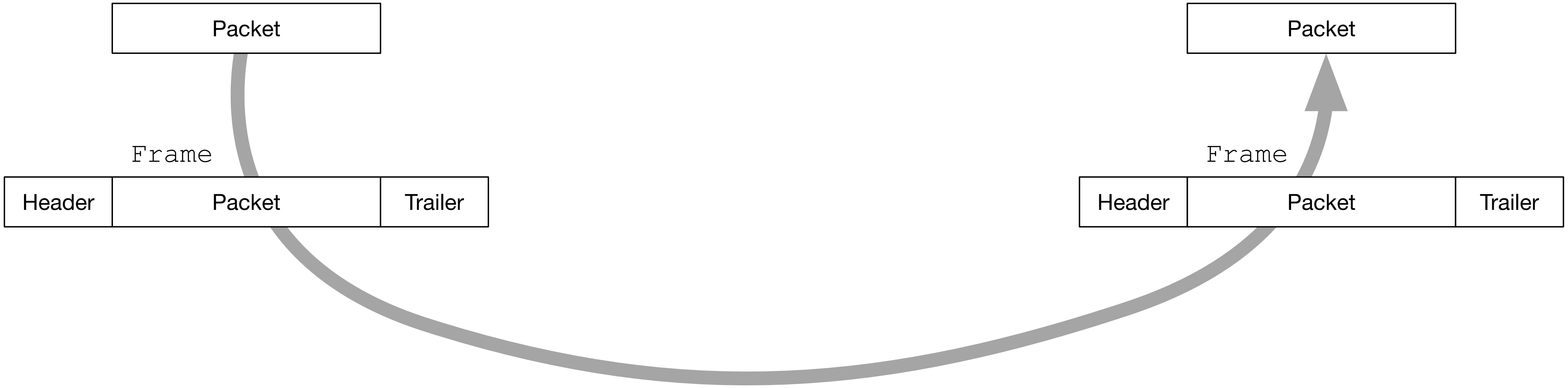
Services provided to network layer

- Data link layer can provide
  - unacknowledged connectionless service
  - acknowledged connectionless service
  - acknowledged connection-oriented service

Framing

- Physical layer can send a raw bit stream
- Network layer provides a bit stream or a stream of packages
- Usually, data link layer breaks network layer stream into *frames*
- Frames carry metadata for
    - Error detection / control
    - flow control
    - acknowledgments

Framing


Packet
Packet
Frame
Frame
Header
Packet
Trailer
Header
Packet
Trailer

Framing

- Frames need to be delimited
    - So that receiver can recognize the frame and its parts
    - Four methods:
        - Byte count
        - Flag bytes with byte stuffing
        - Flag bits with bit stuffing
        - Physical layer coding violations

Start	Content	Acks, Windows	Checksum	Finish	Start	Content	Acks, Windows	Checksum	Finish	Start	Conten

Byte Count

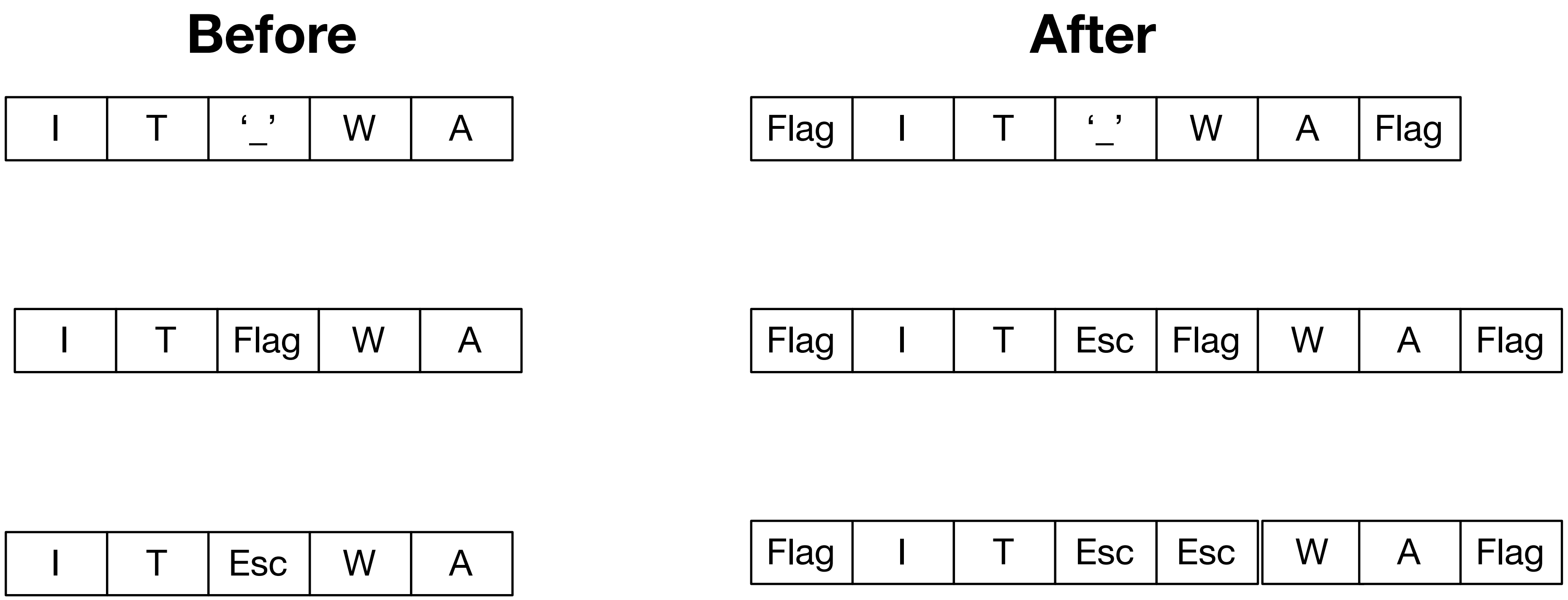
- Can start out a frame with the byte count.
  - Receiver reads the byte count
  - Throws byte count away
  - Reads as many bytes as in the byte count as the frame
  - Divide the following byte stream into frames

0x04	0x03	0x0a	0x32	0xf1	0x05	0x04	0x04	0x04	0x04	0x04	0x03	0xff	0x0b3	0x0b3

  - Once synchronization is lost, cannot recover framing
  - Byte counts can never be corrupted

Flag Bytes

- Use a special flag byte as beginning and end marker
- Use another byte as an escape if the flag appears in the text
- Use a double escape in order to escape an escape



Before	After
I T ‘_’ W A	Flag I T ‘_’ W A Flag
I T Flag W A	Flag I T Esc Flag W A Flag
I T Esc W A	Flag I T Esc Esc W A Flag

Flag bytes

- What is the encoding of

Flag	I	T	Esc	Esc	W	A	Flag

Flag bytes

- Stuffing bytes adds to the size of the stream
  - If a text contains 1/256 flags and 1/256 escapes per byte, then we add 2/256 bytes or about 1% to the size of the text
- Flag bytes is used in the Point-to-Point Protocol
- Flag bytes assume that all data is in the form of bytes

Flag bits

- Developed for the High-level Data Link Control protocol
- Each frame starts and ends with the pattern 01111110 = 0x7E
- To prevent this pattern from appearing in the text of the frame
  - After five consecutive ones, sender inserts a zero
  - After receiving five consecutive ones, receivers drops the following zero.

Flow Control

- True simplex traffic without feedback:
  - Sending pictures from Mars - Mariner expeditions
  - All messages / frames are sent with additional error control bits
  - Receiver can reconstruct original message even if the received message is distorted by noise

Flow Control

- To provide reliable, connection-oriented service
  - Receiver provides sender with feedback
    - in form of special control frames
  - Receiver resends frame after negative feedback
    - or sometimes based on a timer
      - because a frame can be completely lost in a noise burst
  - Usually, frames are numbered

Flow control

- Feedback-based flow control
  - Receiver tells sender when it is ok to send more frames
- Rate-based flow control
  - Sender has mechanism that limits the number of frames per time unit

- A Network Interface Card (NIC) runs at “wire speed” if it can handle all frames that can appear on the wire
  - Then there is no need for flow control at the data link layer

Error Detection / Correction

- Error detection:
    - Receiver can tell that message contains errors
- Error correction:
    - Receiver can reconstruct true message
- Errors:
    - Burst errors (a high error rate for the duration of the burst or several bits in a row are transmitted in error)
    - Erasures (receiver’s physical layer cannot decode a bit)
    - Random errors (bits switch value)

Flow Control

- Sender's network layer gives packets to the data link layer
- Data link layer needs to get exactly the same sequence of packets delivered
- Feedback protocols:
    - Data can get destroyed or corrupted
    - Acks can get destroyed or corrupted
        - Frames can be resend by mistake
    - Need frame numbering and timers

Flow-Control

- Automatic Repeat reQuest (ARQ) / Positive Acknowledgment with Retransmission (PAR)
    - Frames are numbered with a single bit
        - Increment switches 0 to 1 and vice versa
    - Sender sends frame, then waits for Ack
        - If Ack does not arrive in time, then resends frame
        - NEVER sends a new frame without an Ack
        - Increments frame number
    - Receiver sends an Ack for each successful frame
        - Increments expected frame number
        - Can send a No-Ack to ask for resending a corrupted frame

Exercise

- Play through all scenarios to determine whether a frame can be lost or duplicated.

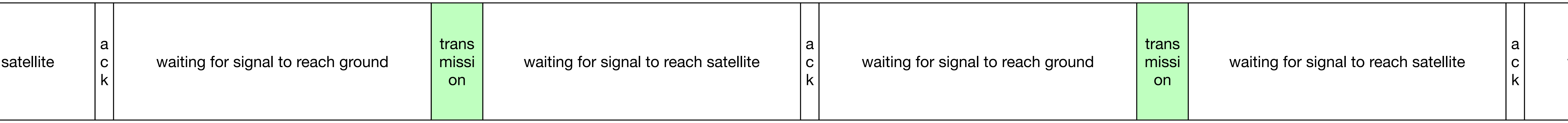
- Frame fine, Ack fine
- Frame bad, no Ack sent
- Frame bad, No-Ack sent, and received
- Frame fine, Ack sent, and lost
- Frame bad, No-Ack sent and lost

Flow Control

- ARQ does not work with long delays
    - Exercise:
        - Satellite link has 100 msec delay and 100Mbps bandwidth
        - How much of the channel can actually be used
        - Assume frames of 10Kb.

Answer

- Time to send a frame:
  - 10Kb/100Mbps = $10^4/10^9$ sec = 10^{-5} sec = 10^{-2} msec



satellite

ack

waiting for signal to reach ground

transmission

waiting for signal to reach satellite

ack

waiting for signal to reach ground

transmission

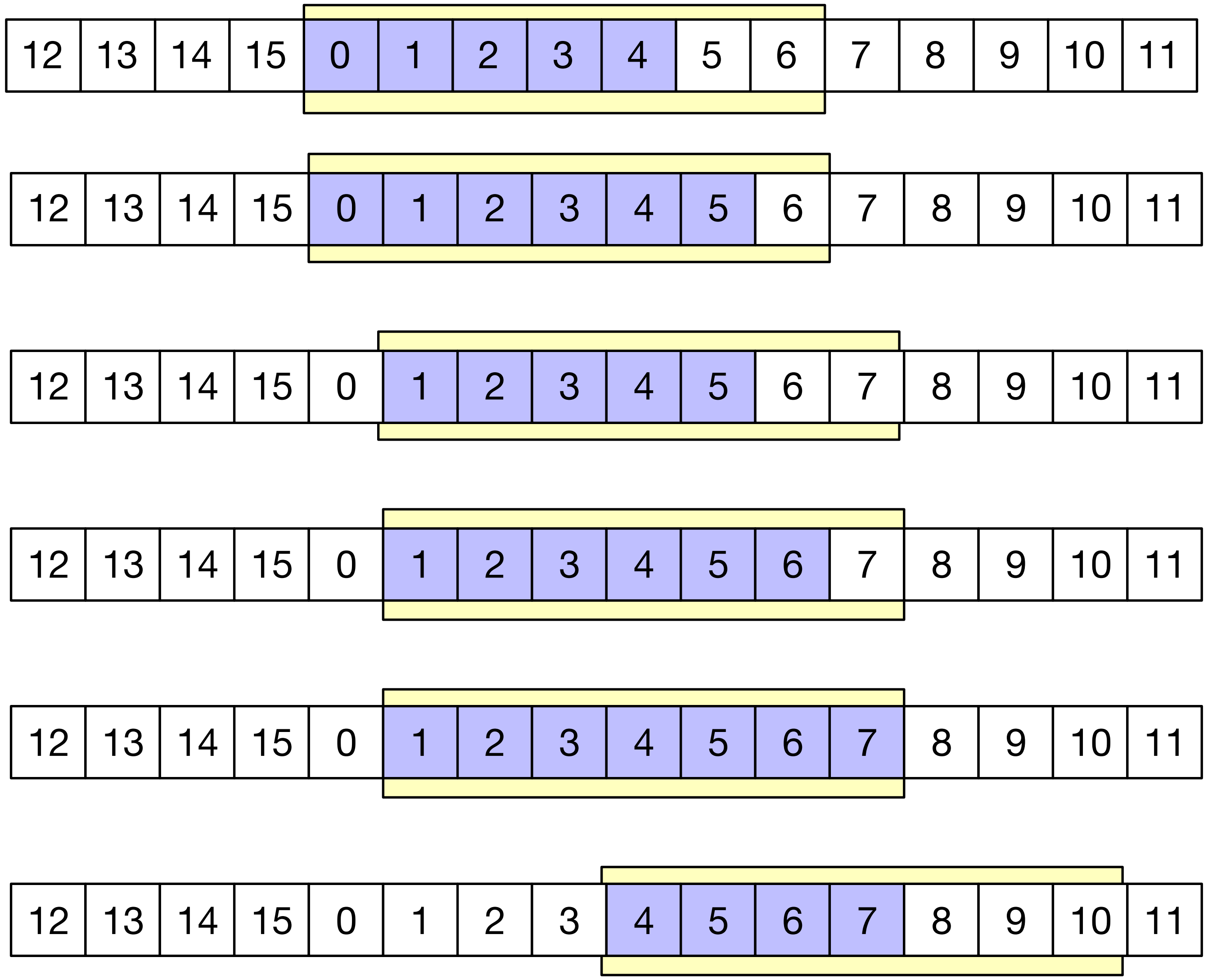
waiting for signal to reach satellite

ack

- Useful time proportion:
- $$\frac{10^{-2}}{2 \times 10^2 + 10^{-2}} = 0.0000499975$$

Sliding window protocol

- For better use of physical link, use sliding window protocols
    - Idea: Sender sends a number of frames without waiting for acks
    - When acks are missing, resend frames after a while
    - Fundamental problem: Frames need numbers, but numbers need to roll over
        - Use a sliding window: numbers repeat



Five frames sent

Six frames sent

Ack for frame 0 received

Seven frames sent

Eight frames sent, buffer
is full, cannot send anymore

Received acks for
frames 1, 2, 3,
can now send three
more frames

Sliding window protocol

- Sender needs to buffer all unacknowledged frames
- Acknowledgments can be done cumulatively
  - When we ack frame *n,* we also ack all preceding frames *n*-1, *n-2, …* in the sliding window
- Some protocols have resend messages for garbled frames, others just relay on not acknowledging them
- Selective repeat:
  - Receiver buffers all frames that arrive after a garbled one
  - Sends a resend request to sender
  - Gives frames to the network layer only in order received

Protocols

Serial Line Internet Protocol

SLIP

- Created informally in early 1980s
- Became a standard before

  RFC 1055 (1988) A Non-standard for Transmission of IP Datagrams over Serial Lines : SLIP

- Just provides framing
  - SLIP *END character* of a frame is 0xC0 = b11000000
  - Minor enhancement: precede the datagram with a SLIP end character
  - Uses *SLIP escape character* 0xDB = b11011011
  - Maximum frame size is 1006B, but can change between implementations

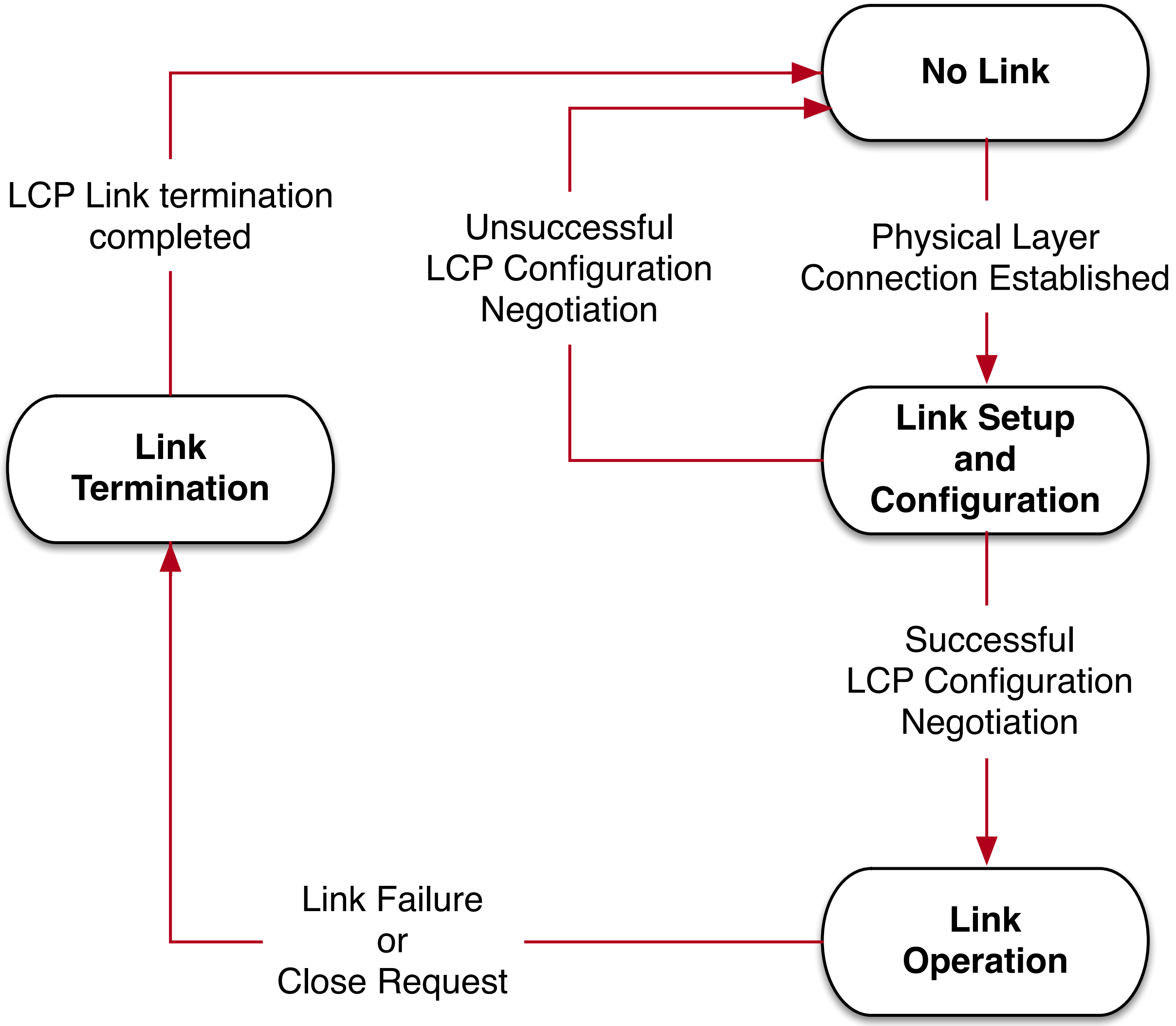
Point-to-Point Protocol (PPP)

PPP Overview

- RFC 1134 (1989) —> RFC 1171 (1990): the PPP standard
- Uses the ISO High Level Data Link Control (HLDLC) protocol by IBM
- PPP provides:
    - More comprehensive framing
    - Allows multiple layer 3 protocols to be multiplexed on a single link
    - Uses error detection via CRC
    - Negotiates link parameters, including maximum frame size
    - Testing links before and during transmission
    - Support for authentication
    - Support for compression, encryption, and link aggregation
        - Link aggregation - two physical links can be used as one

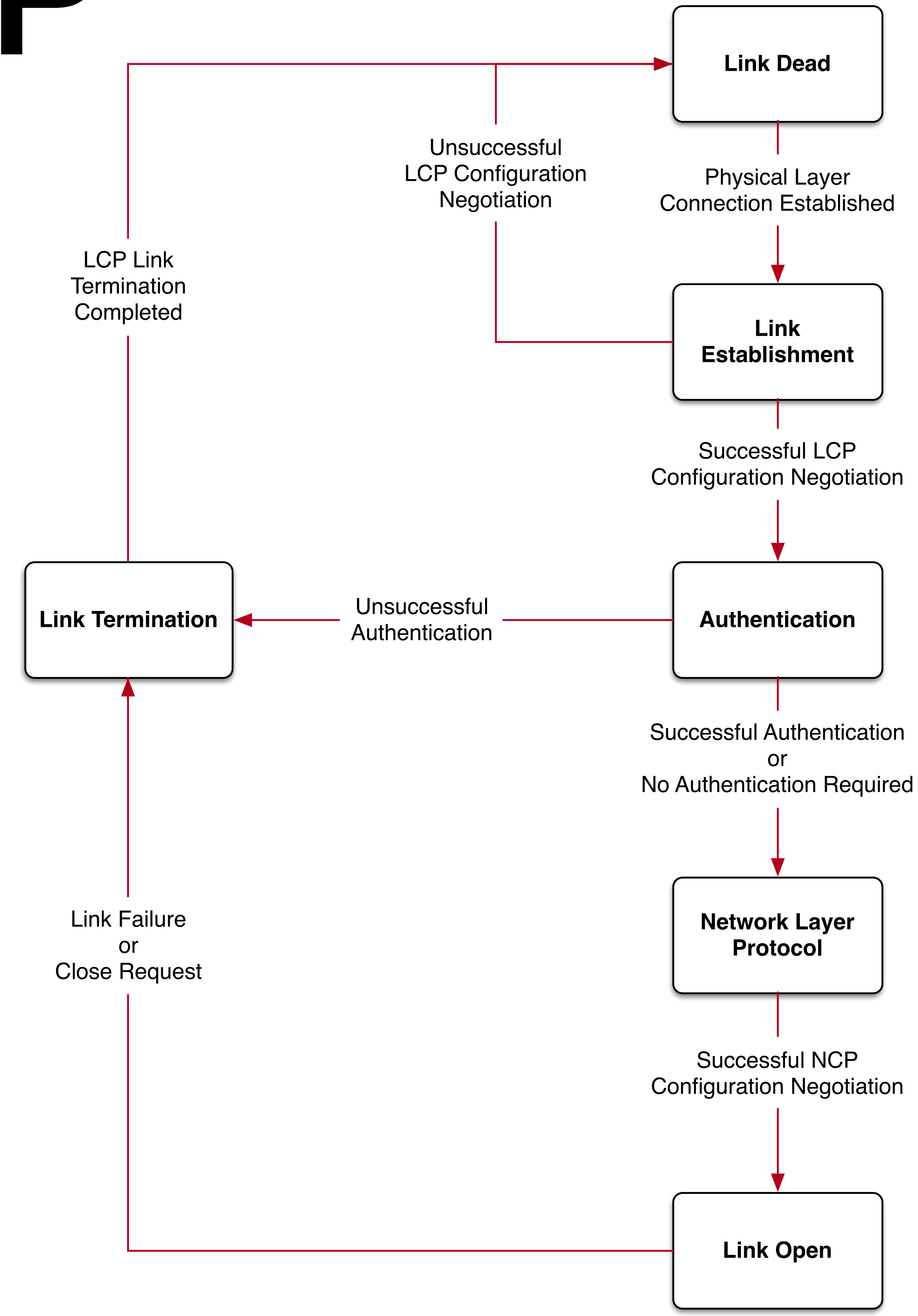
PPP

1. Devices make contact and set up a link
    - Agree on all parameters
    - Network Control Protocol (NCP) is selected according to layer 3 traffic
2. Link operation
3. Link termination
    - by either device


No Link
LCP Link termination completed
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
Link Termination
Link Setup and Configuration
Successful LCP Configuration Negotiation
Link Failure or Close Request
Link Operation

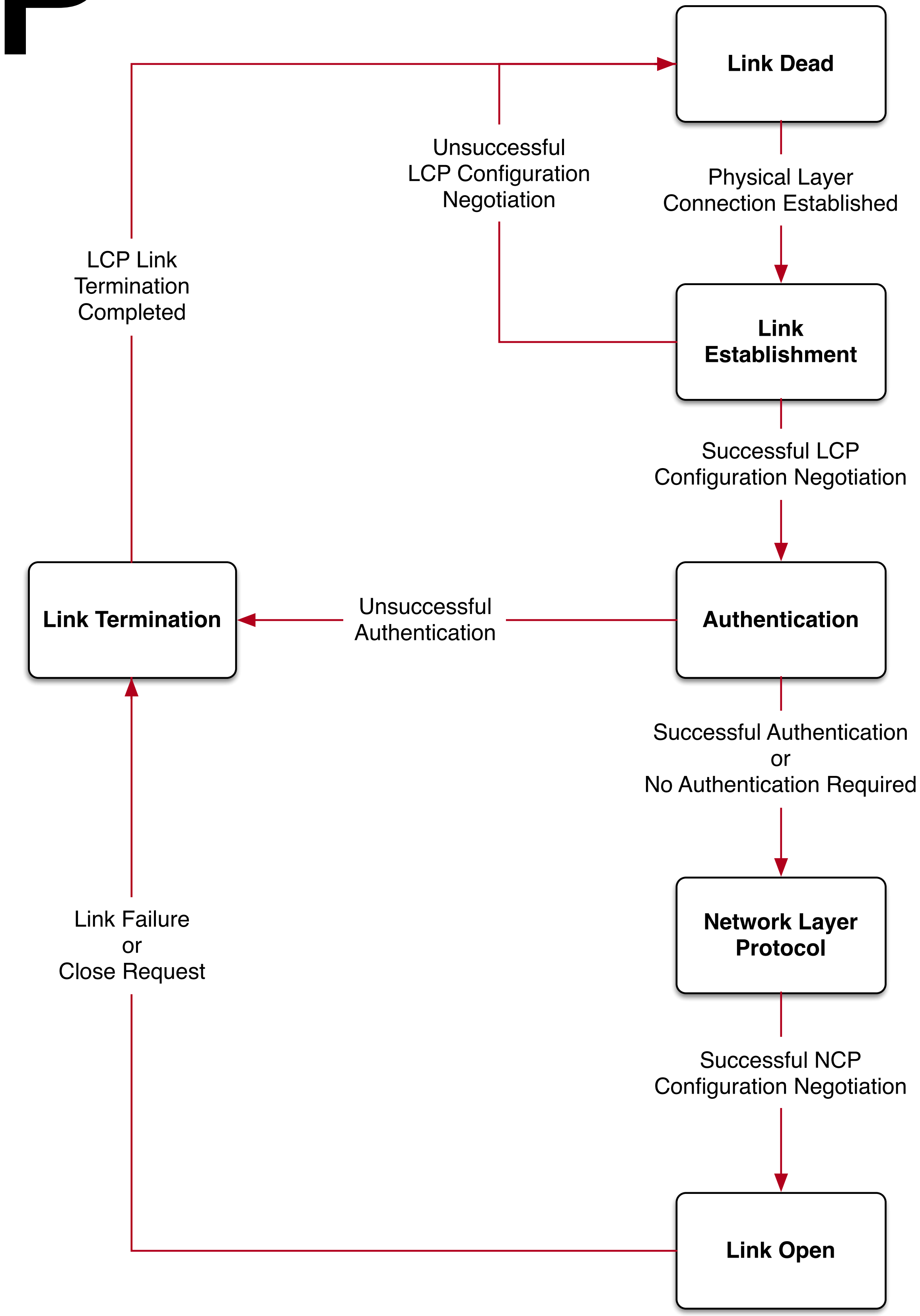
PPP

- Link Control Protocol used to set up link so that PPP can be used
- Links start out in the Link Dead phase
- When devices detect connection, they start establishment


Link Dead
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
LCP Link Termination Completed
Link Establishment
Successful LCP Configuration Negotiation
Link Termination
Unsuccessful Authentication
Authentication
Successful Authentication or No Authentication Required
Link Failure or Close Request
Network Layer Protocol
Successful NCP Configuration Negotiation
Link Open

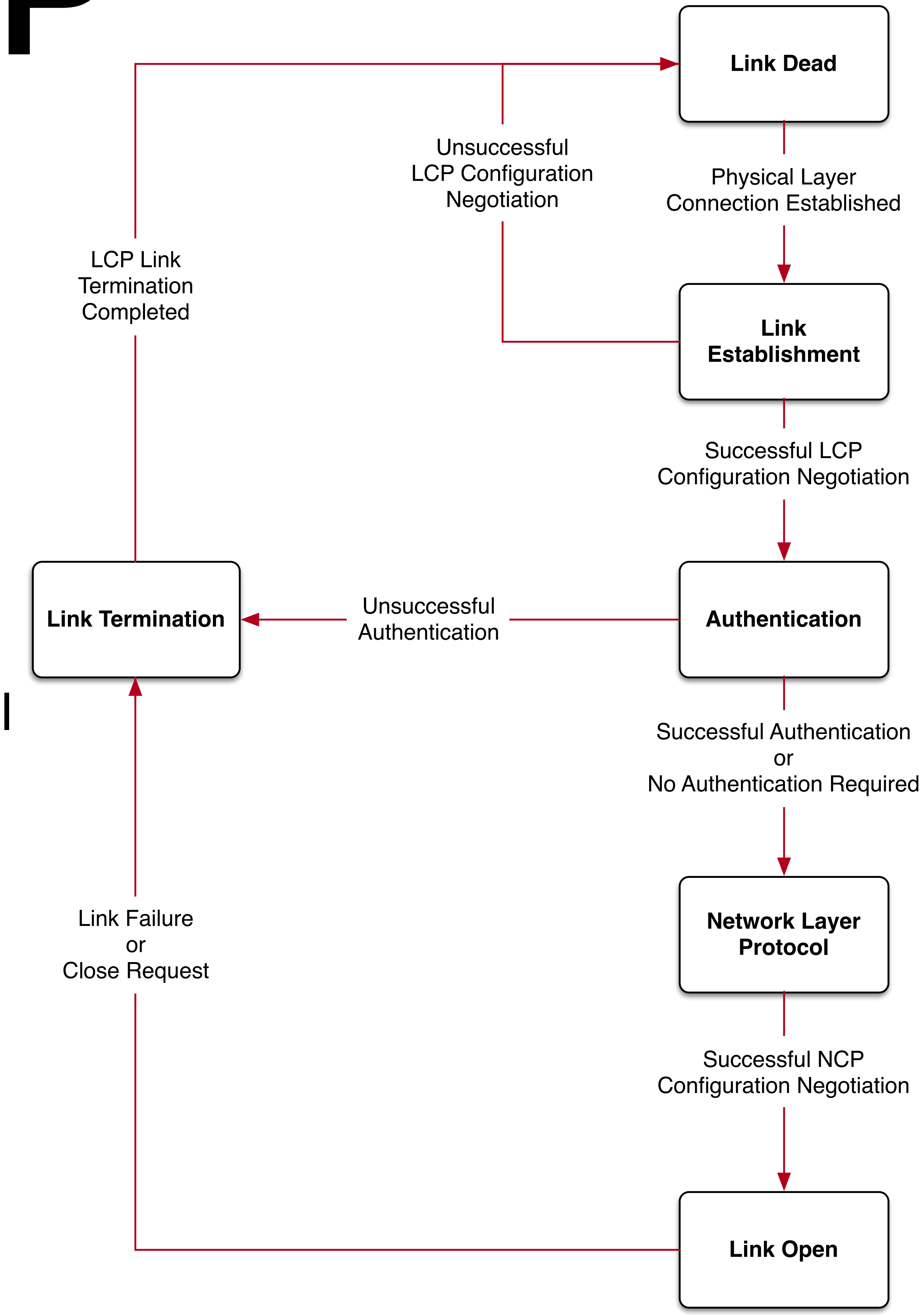
PPP

- Device A sends LCP configuration request to Device B
- Device B checks parameters
- If they work, send Ack
  - Successful negotiation
- If they do not work, send Nack
  - Unsuccessful negotiation


Link Dead
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
LCP Link Termination Completed
Link Establishment
Successful LCP Configuration Negotiation
Link Termination
Unsuccessful Authentication
Authentication
Successful Authentication or No Authentication Required
Link Failure or Close Request
Network Layer Protocol
Successful NCP Configuration Negotiation
Link Open

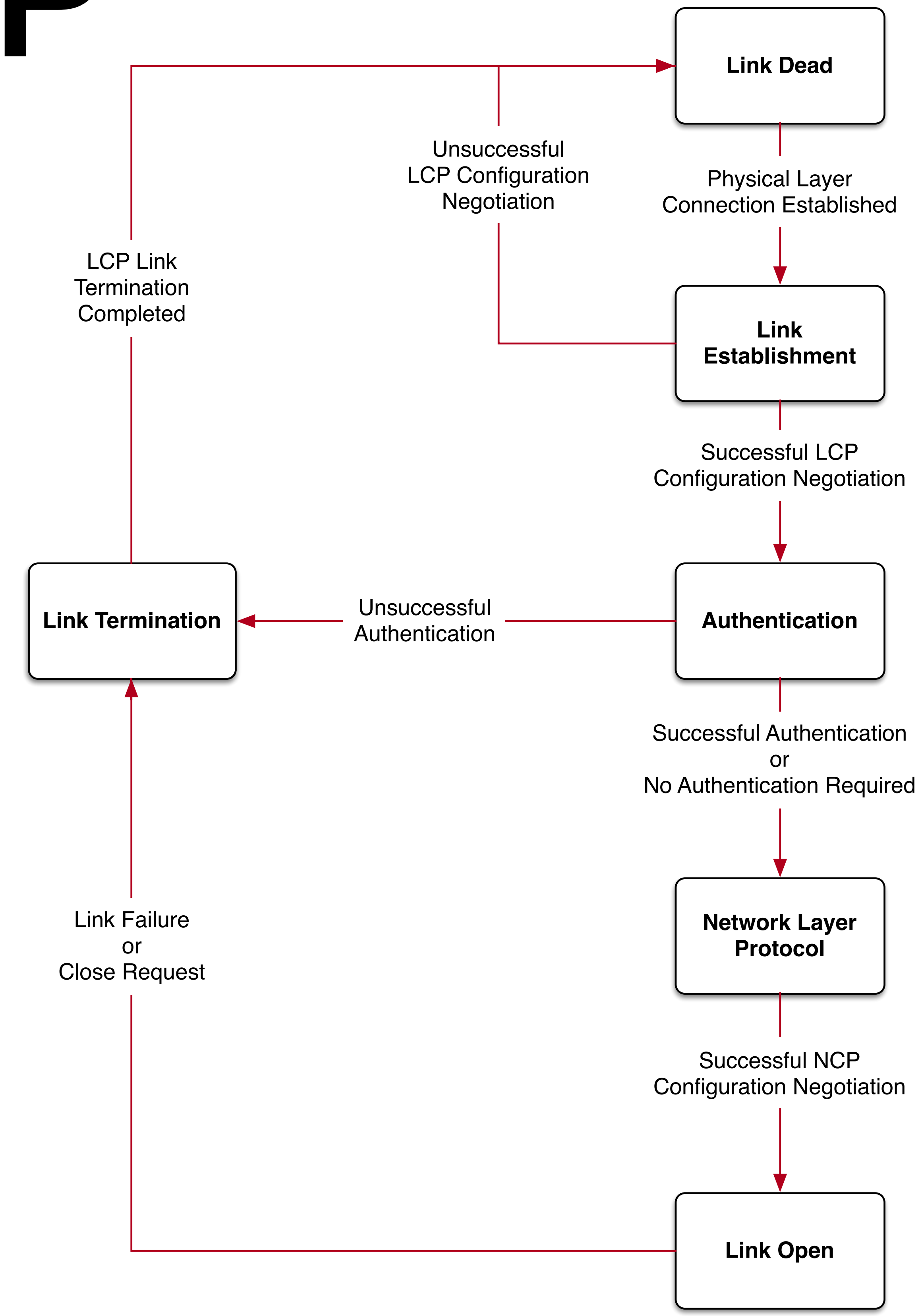
PPP

- LCP link configuration
  - Initiator sends a configure-request frame
    - Options for MRU, and Authentication & Quality protocols, Magic Number (in order to detect loopbacks), Protocol field compression in PPP frames, Address and Control field compression in PPP frames
  - Responder either agrees or disagrees with the proposal
    - Configure-Nack makes counter-proposals
    - Configure-Reject just rejects


Link Dead
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
LCP Link Termination Completed
Link Establishment
Successful LCP Configuration Negotiation
Link Termination
Unsuccessful Authentication
Authentication
Successful Authentication or No Authentication Required
Link Failure or Close Request
Network Layer Protocol
Successful NCP Configuration Negotiation
Link Open

PPP

- After link establishment, proceed to authentication phase
    - Needed for example for dial-up connections
    - Uses CHAP or PAP

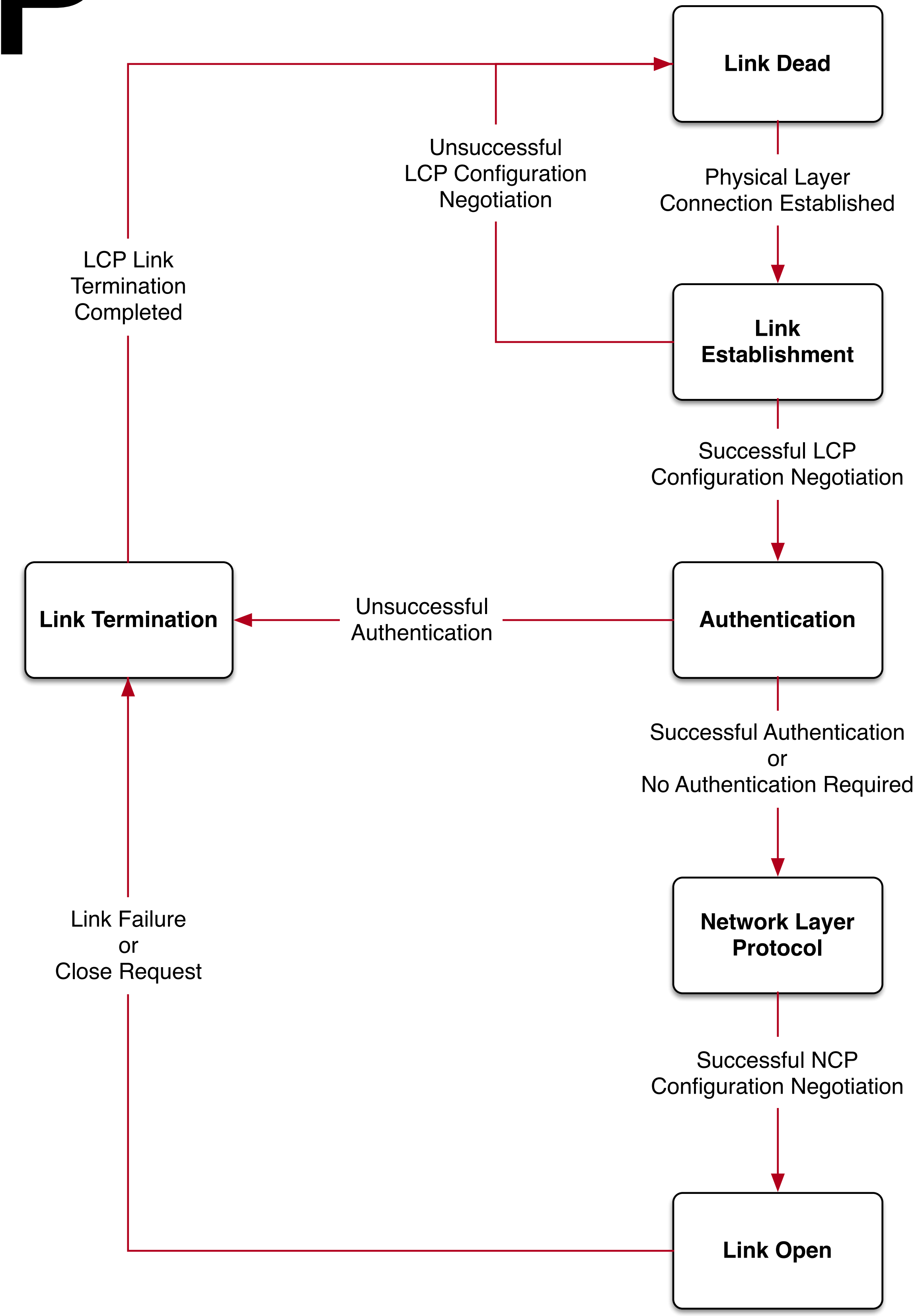

Link Dead
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
LCP Link Termination Completed
Link Establishment
Successful LCP Configuration Negotiation
Link Termination
Unsuccessful Authentication
Authentication
Successful Authentication or No Authentication Required
Network Layer Protocol
Link Failure or Close Request
Successful NCP Configuration Negotiation
Link Open

PAP & CHAP

- PAP
  - Initiator sends a password and name in an Authentication Request
  - Responder decides whether to accept
- PAP is insecure
- CHAP
  - Uses three-way hand-shake:
    - Responder sends a challenge
    - Initiator encrypts challenge with shared key
    - Responder checks and indicates success or failure to initiator

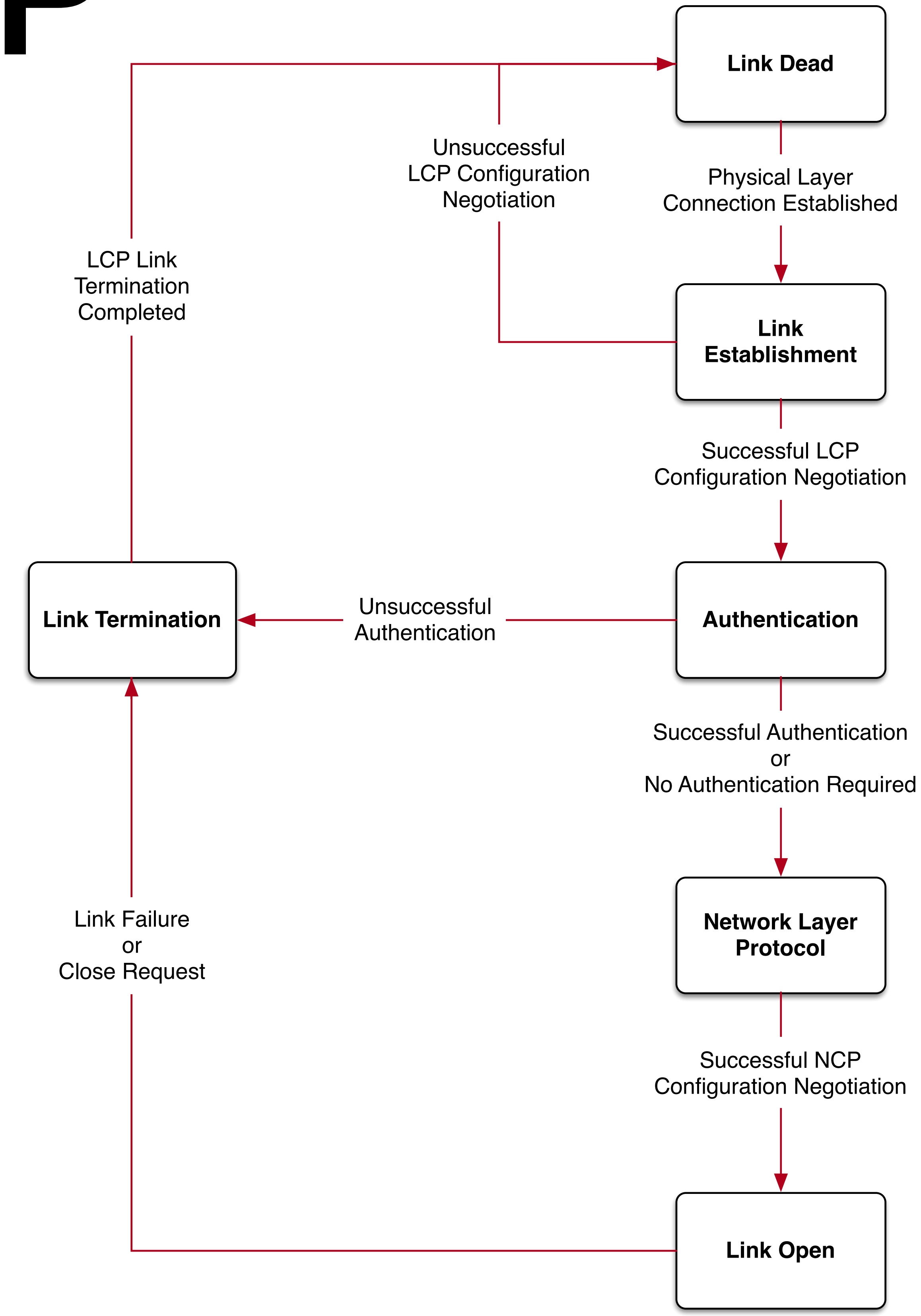
PPP

- Network Control Protocol
  - Lightweight version of LCP
  - Optimizes according to Layer 3 traffic
    - Originally, PPP was for IP only
    - Special protocol versions for:
      - IPv4, IPv6, IPX (Novell NetWare suite), NetBios, AppleTalk
  - There can be more than one NCP connection over the same link


Link Dead
Physical Layer Connection Established
Link Establishment
Unsuccessful LCP Configuration Negotiation
Successful LCP Configuration Negotiation
Authentication
Unsuccessful Authentication
Link Termination
LCP Link Termination Completed
Successful Authentication or No Authentication Required
Network Layer Protocol
Successful NCP Configuration Negotiation
Link Open
Link Failure or Close Request

PPP

- NCP uses Configure-Request, Configure-Ack, Configure-Nack, ConfigureReject to establish parameters
- Example IPv4:
  - Van Jacobson TCP/IP header compression
  - Specify IP address
  - Request other device supplies IP address (for dial up)
- NCP connection can be closed without closing the LCP connection



Link Dead
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
LCP Link Termination Completed
Link Establishment
Successful LCP Configuration Negotiation
Link Termination
Unsuccessful Authentication
Authentication
Successful Authentication or No Authentication Required
Network Layer Protocol
Link Failure or Close Request
Successful NCP Configuration Negotiation
Link Open

PPP

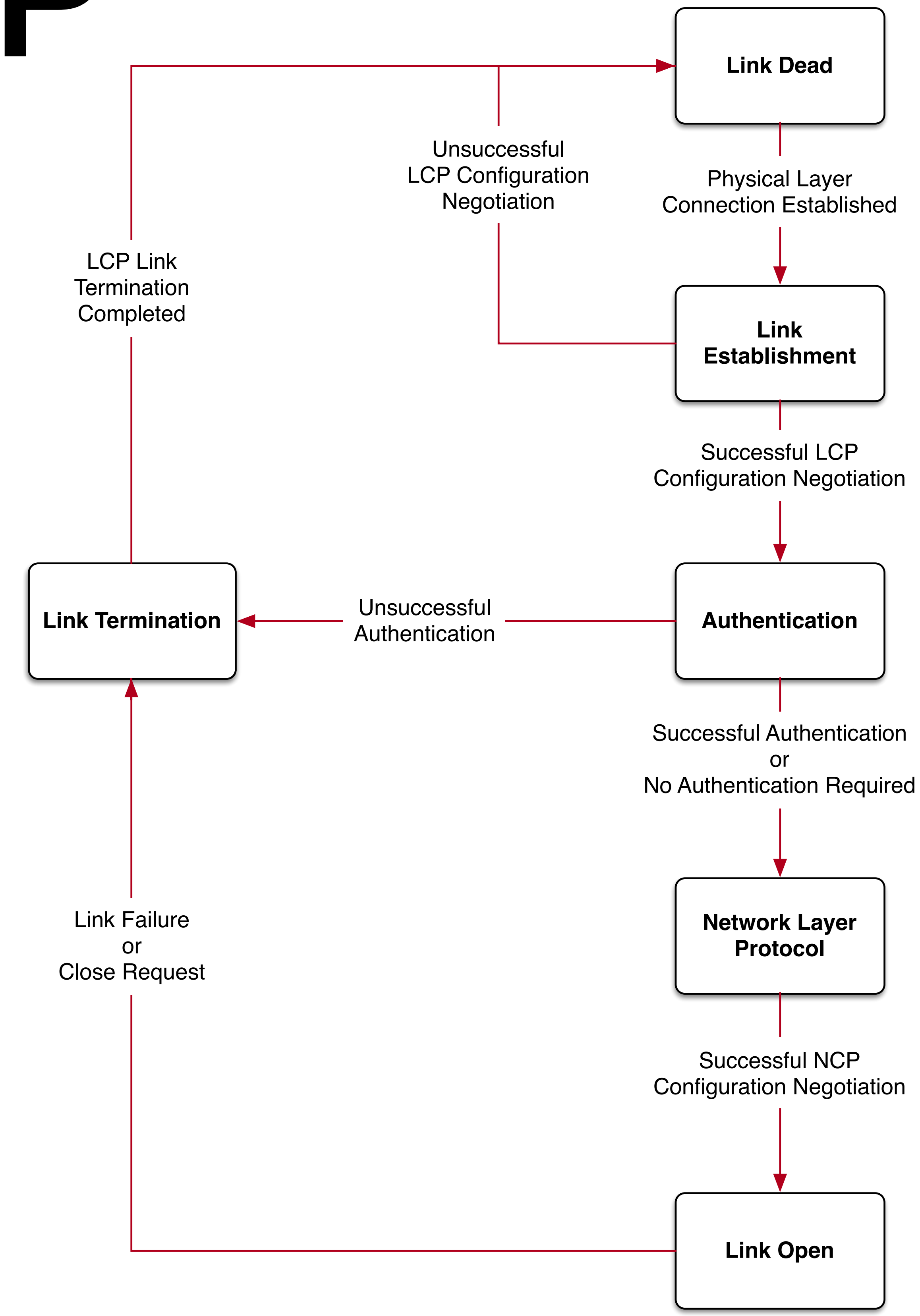
- PPP Link Quality Monitoring and Reporting (LQM / LQR)
- Can periodically create statistics
    - Number of frames
    - Number of bytes
    - Number of errors
    - Number of discarded frames
- Devices can use LQR to react to changes in the quality of the link

PPP

- PPP Compression Control Protocol (PPP CCP)
  - Sets up one of several compression algorithms for data
- PPP Encryption Control Protocol (PPP ECP)
  - Sets up one of several encryption protocols
- PPP Multilink Protocol
  - Optional feature of PPP implementations
  - Example: Used to combine several physical channels defined on the same physical medium
  - Each frame is divided into fragments that are sent over the different channels

PPP

- Link maintenance
  - Any device can use echo-requests to test the link
- Link termination
  - Any device can send a terminate-request message
  - Other device sends terminate-ack message



Link Dead
Unsuccessful LCP Configuration Negotiation
Physical Layer Connection Established
LCP Link Termination Completed
Link Establishment
Successful LCP Configuration Negotiation
Link Termination
Unsuccessful Authentication
Authentication
Successful Authentication or No Authentication Required
Link Failure or Close Request
Network Layer Protocol
Successful NCP Configuration Negotiation
Link Open