

Fig. 4.1 Source end network layer connected through set of IP routers for data packets from an IP address and communicating with IoT/M2M IoT Apps and Services layer using TCP/IP suite of Application protocols

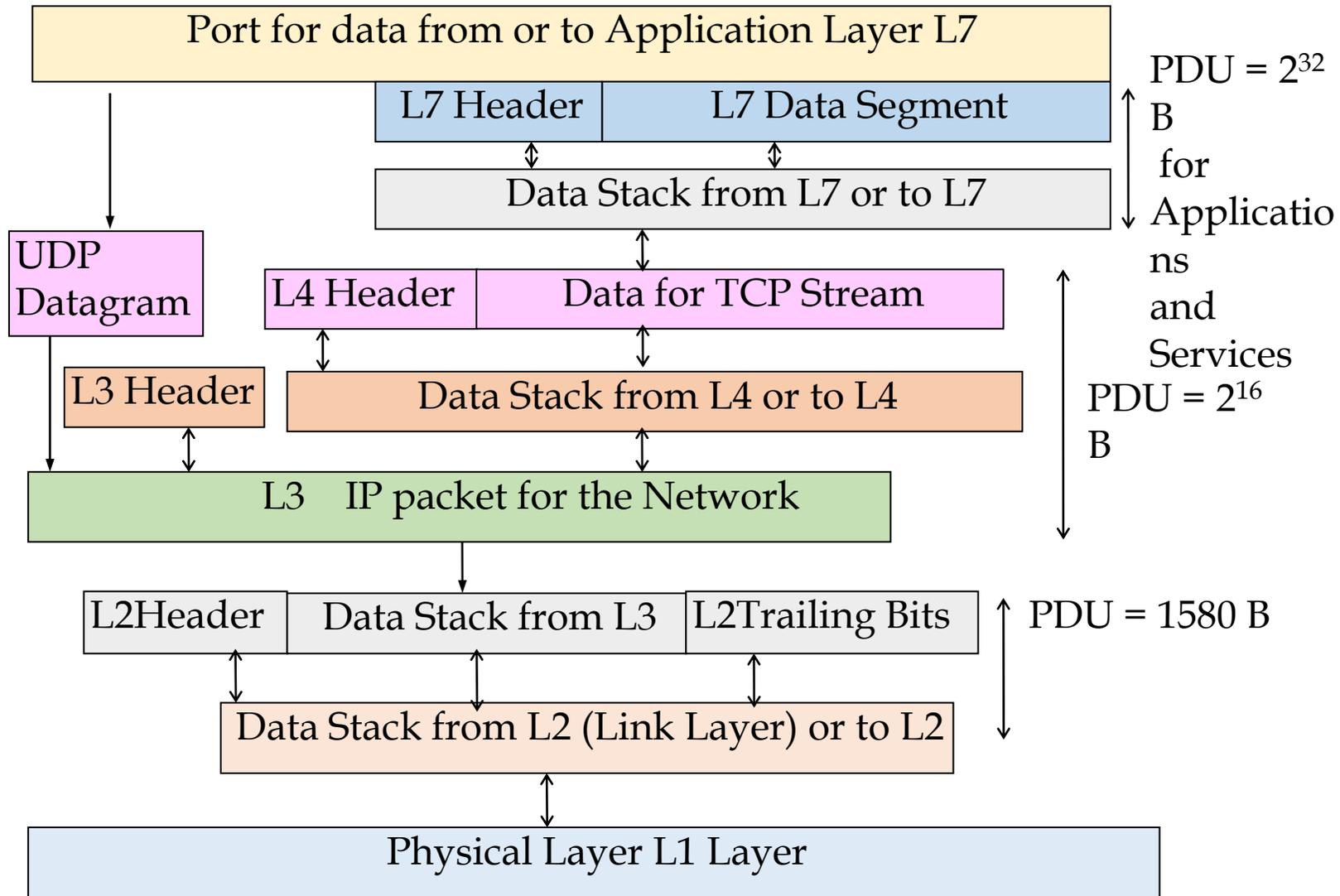
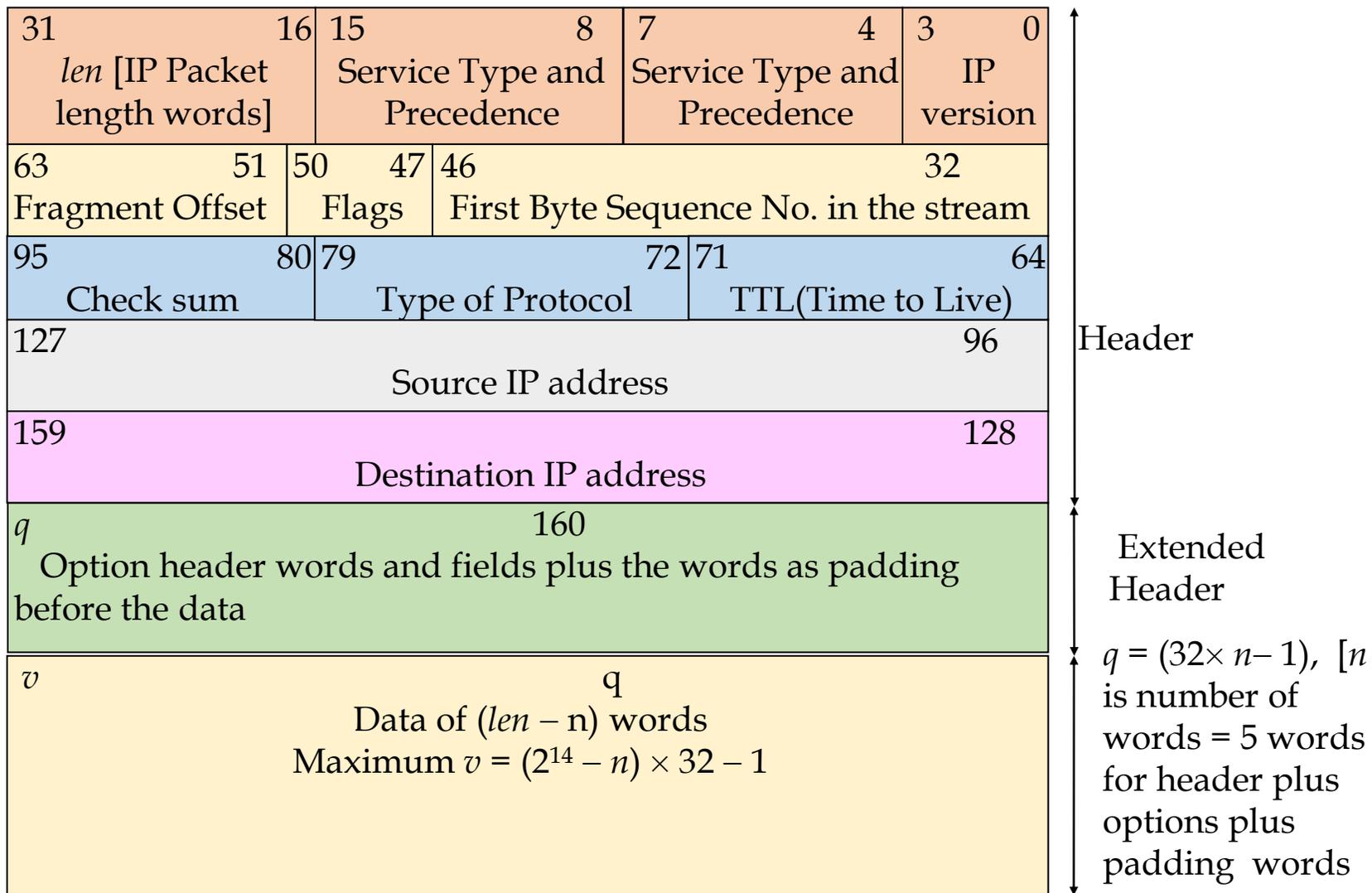


Fig. 4.2 TCP/IP suite four layers model generating data stack for the network, and for physical layer during Internet communication



Data Packet (stack) from or to Transport layer (Maximum Size 2^{14} words = 2^{16} B)

Fig. 4.3 Data stack received or transmitted at or to transport layer, and packet consisting of IP header fields of 160 bits and extended header (*n* - 5) words (when required) plus data stack of maximum *v* words from or for the transport layer

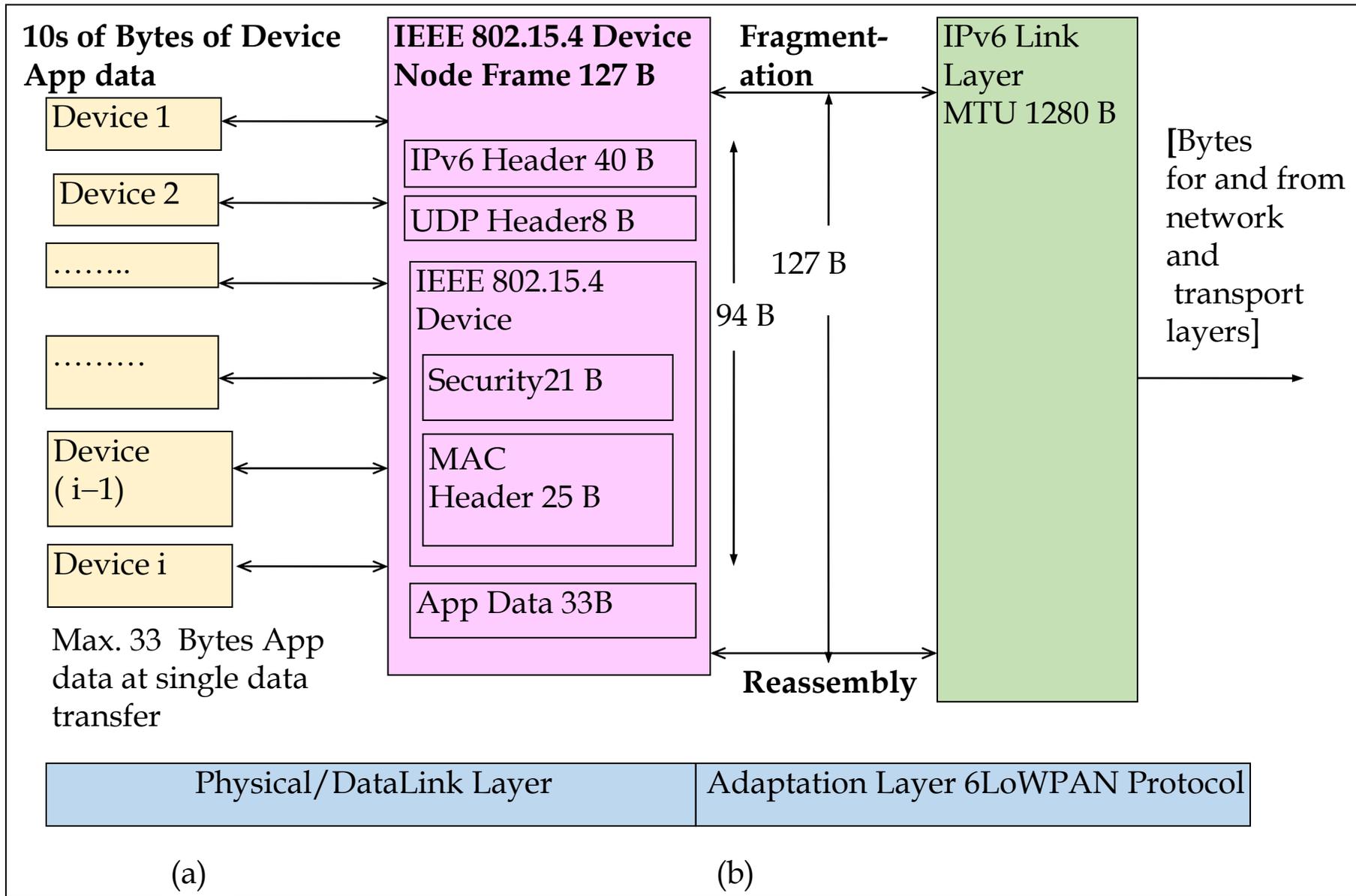


Fig. 4.4(a) Physical layer IEEE 802.15.4 network devices (b) Adaptation layer 6LoWPAN protocol 127 B fragmented frames reassembly into IPv6 maximum 1280 B or fragmentation of IPv6 MTU 1280B into 127 B frames for transfer to a device.

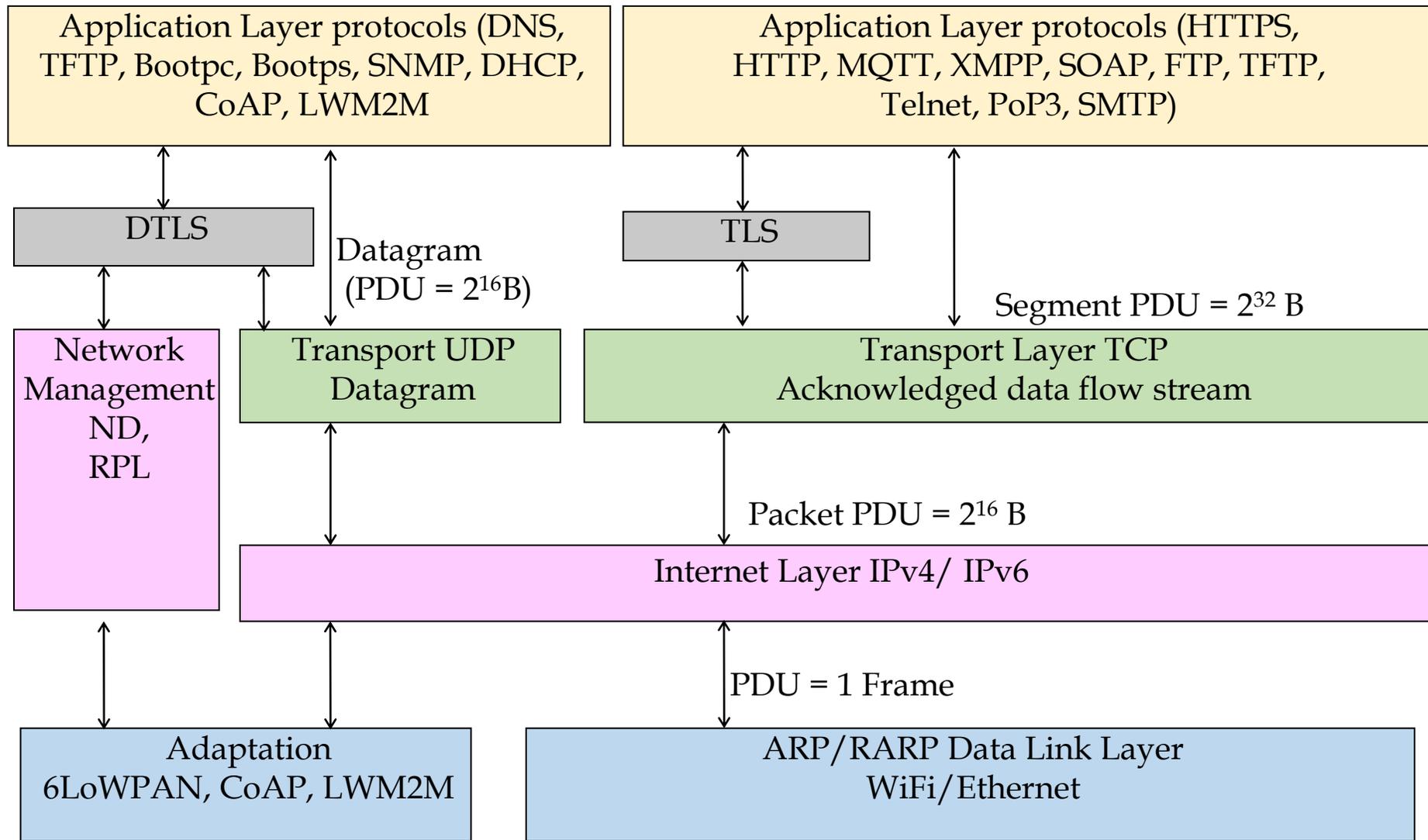


Fig. 4.5 IoT TCP/IP Suite of Protocols for Internet

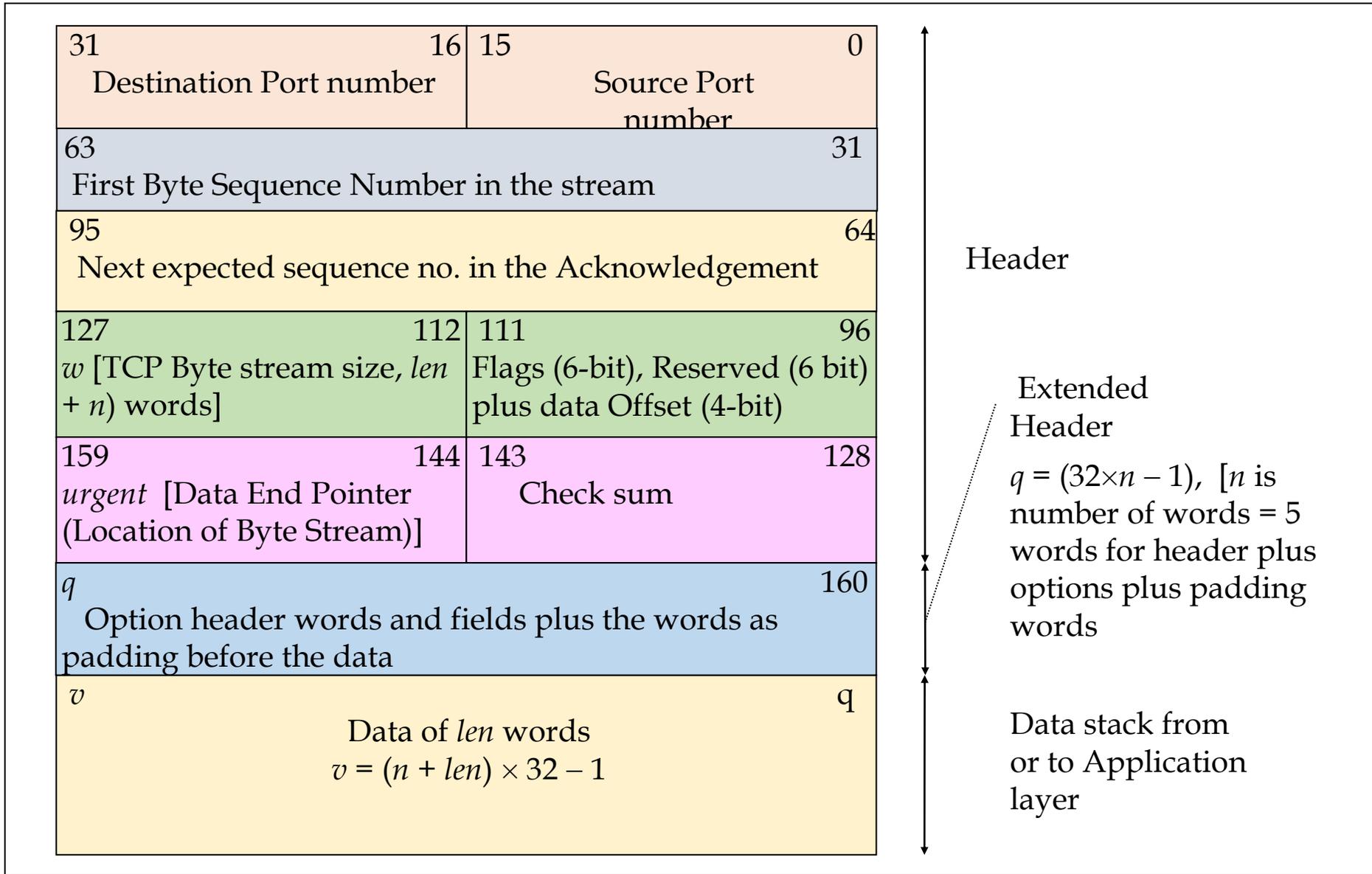


Fig. 4.6 Data stack received or transmitted at or to transport layer stream consisting of TCP header field 160 bits and extended header ($n - 5$) words when required plus data stack of len words from or for the Application layer

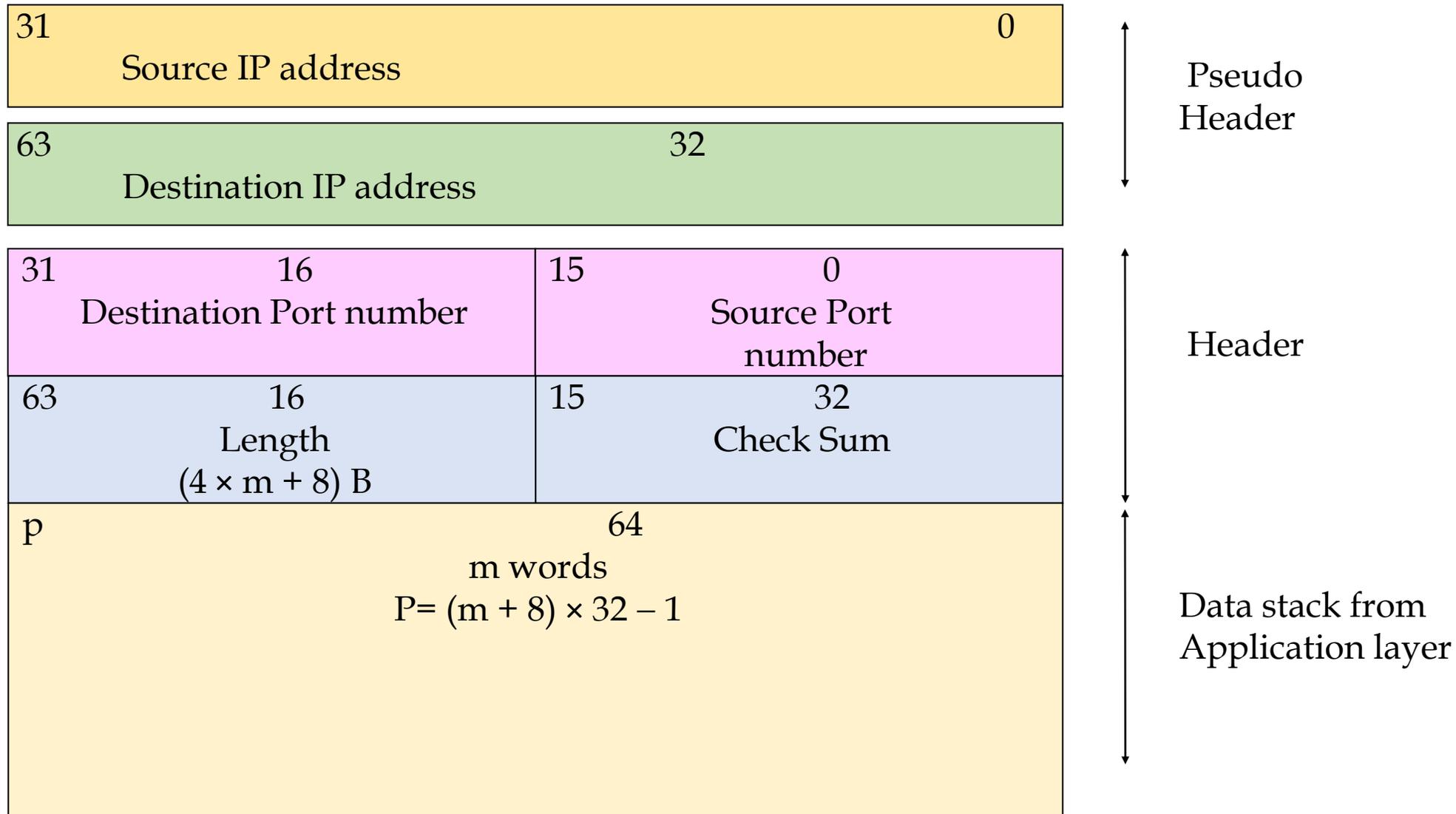


Fig. 4.7 Transport Layer UDP Header field with data stack from the Application layer and Pseudo header of 2 words (64 bits) for source and destination IP addresses

Lesson 6

Application Layer Protocols: HTTP, HTTPS and Other Ports

A TCP/IP message

- Sent from specified port at transmission end and to the specified port at the receiver end, else receiver port does not listen

TCP/IP suite number of Application layer protocols

- HTTP, HTTPS, FTP, Telnet and Others
- A concept of Port: Each port uses a distinct protocol at an Application layer
- A port uses a protocol for sending and receiving messages.

Application layer HTTP Port

- **HTTP** (Hyper Text Transfer Protocol)
- Port number = 80
- HTTP port sends Application data stack at the output to the lower layer using the HTTP protocol
- An HTTP web server listens to port 80 only and responds to port 80 only
-

Application layer HTTP Port

- Uses a URL for example `http://www. mheducation.com/`.
- Default port taken as 80 for the URL. Port number can be specified after the TLD, for example, after `.com` in URL `http://www. mheducation.com:80/`.

HTTPS (HTTP over Secure Socket Layer or TLS)

- Port number = 443. An HTTPS port sends a URL, for example, https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.
- Here, TLD is .org

https://en.wikipedia.org/wiki/List_

- domain name is wikipedia.org.
- Sub domain name is en.
- Resource relative URL is /wiki/List_of_TCP_and_UDP_port_numbers

HTTP Client and Server

- HTTP standard protocol
- Requests a URL specified web page (resource)
- Web-Server sends in response the requested resource
- .

HTTP Client and Server

- The HTTP client requests an HTTP server on the Internet
- The server responds by sending a response
- The response may be with or without processing

HTTP a Stateless Protocol

- An HTTP request assumed a fresh request as per the protocol
- No session or sequence number field or no field that is retained in the next exchange
- .

HTTP a Stateless Protocol

- A current exchange by an HTTP request independent of the previous exchanges. The later exchanges do not depend on the current one.

Cookie

- A text file which creates during a particular pair of exchanges of HTTP request and response
- The creation is either at a CGI or processing program, for example, JavaScript or script or at a client.

Cookie

- A prior exchange may then depend on this cookie
- An HTTP state management mechanism

HTTP a file transfer-like protocol

- Used more efficiently than the FTP
- On the other hand HTTP no command line overheads

HTTP Protocol Features

- Very light (a small format)
- Speedy as compared to other protocols, such as FTP
- Able to transfer any type of data to a client provided it is capable of handling that data.

HTTP Protocol Features

- Based on Object Oriented Programming System (OOPS)
- Eight HTTP specific specified methods 1. GET. 2. POST. 3. HEAD. 4. CONNECT. 5. PUT. 6. DELETE. 7. TRACE. 8. OPTIONS

Summary

We learnt

- Application Layer Protocols
- HTTP
- HTTPS
- FTP
- HTTP Features

Summary

We learnt

- Datagram
- Rest and RESTful
- Hypertext, Hyperlink
- XML

End of Lesson 7 on

Application Layer Protocols: HTTP, HTTPS, and Other Ports