


(TCP)/IP on (small) embedded systems



Roel Jonkman

Introduction

- Small embedded systems (MCU's) running RTOS'es or bare metal.
 - 1MB or less RAM.
 - < 0.5Ghz (+-)
- Various small embedded TCP/IP stacks are readily available.
 - Commercially supported as well as open source. (BSD and GPL)
 - Quality varies somewhat. (wildly)
 - Getting TCP implementations correct is hard.
 - Infinite variety of potential tcp client implementations.

Challenges

- Context: Embedded processor used for control of a safety critical device.
(medical, UAV, car, truck etc.)
- Resource Concerns:
 - RAM.
 - CPU cycles.
- Security
 - Denial of service
 - Most if not all embedded stacks lack rudimentary firewall features.

L4 Protocol Choices

- TCP
 - Reliable.
 - Stream oriented, no message delineation.
- UDP (unicast)
 - Unreliable
 - Message oriented
 - Point to point
- UDP (multicast)
 - Unreliable
 - Message oriented
 - Multiple destinations as well as sources.

TCP

- Reliable, therefore 'easy' to work with from an application standpoint.
- No message delineation.
 - Need to implement your own on top. (Not so easy after all.)
- Per connection buffer sized according to TCP window size.
 - Needs to be at least a couple segments (MTU) worth.
 - 64k is max without using window scaling.
 - Large bandwidth delay products are (potentially) problematic.
- Can only handle a few connections.
 - RAM constraints.
- Likely does not behave well with misbehaving wireless links.
 - Untenable delay/jitter.
 - Breakdown of congestion window.
- Server ports are subject to Denial of Service.

UDP(unicast)

- Point to point
- Message delineated
 - Realistically limited to the maximum IP MTU - 8 bytes.
 - Can go up to ~64k.
 - Buffers have to 64k on both ends.
 - Subject to IP fragment loss.
- Need at least 1 UDP message size buffer per 'connection'.
- Possible to have more than a handful of 'connections'

UDP multicast

- UDP unicast pros/cons apply
- Specific address space, 224.0.0.0/4
- Stack needs to have some minimal primitives to join and leave multicast groups.
 - This should emit IGMP messages to downstream routers.
- Turns UDP into 'broadcast'.
 - This has implications for bridged/non routed networks.
- Single socket on device to communicate with many.
 - Resource efficient.

Resource constraints.

- Embedded systems can handle a limited number of point to point 'connections'.
 - TCP more limited than UDP.
- CPU overhead can be significant.
 - Usually 1 thread is required to be 'high priority'.
- Quality of stacks varies wildly.

Security concerns.

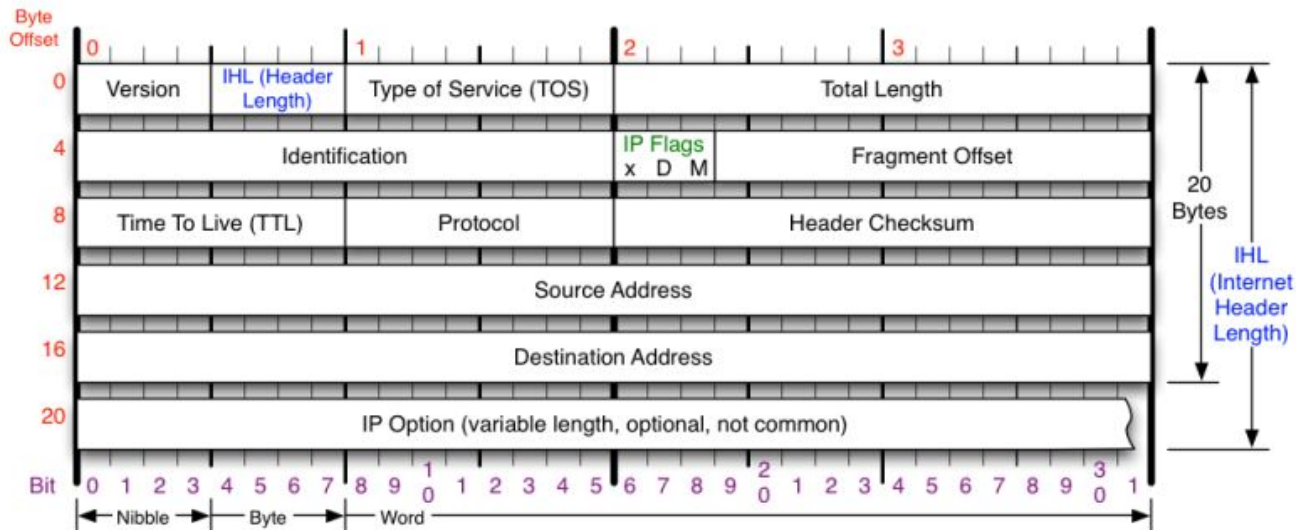
- Denial of service
 - Syn flood of open TCP ports.
 - Resource exhaustion of state entries.
 - Doesn't take much to do this.
 - Syn cookies is the mitigation.
 - None of the embedded stacks I've come across implement this.
- No port/address firewalling features.
 - Flood a device with UDP packets to an open port.
 - Likely to cause significant CPU usage.
 - Can be somewhat mitigated by choosing thread priorities carefully.

Practical examples.

- Multiple Peer publish/subscribe:
 - Use UDP multicast
- Peer to Peer RPC
 - Use UDP unicast.
- Log downloads (intermittent.)
 - Use separate command to enable/disable server port.
 - Limit exposure of open port to 'when needed'.
 - Implement TFTP instead of FTP.
 - Most likely need to 'augment' TFTP a bit.
 - TFTP is very primitive.
- Telemetry data.
 - Use UDP multicast.
 - TCP is liable to cause unacceptable delays and jitter.

Collateral

IP



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

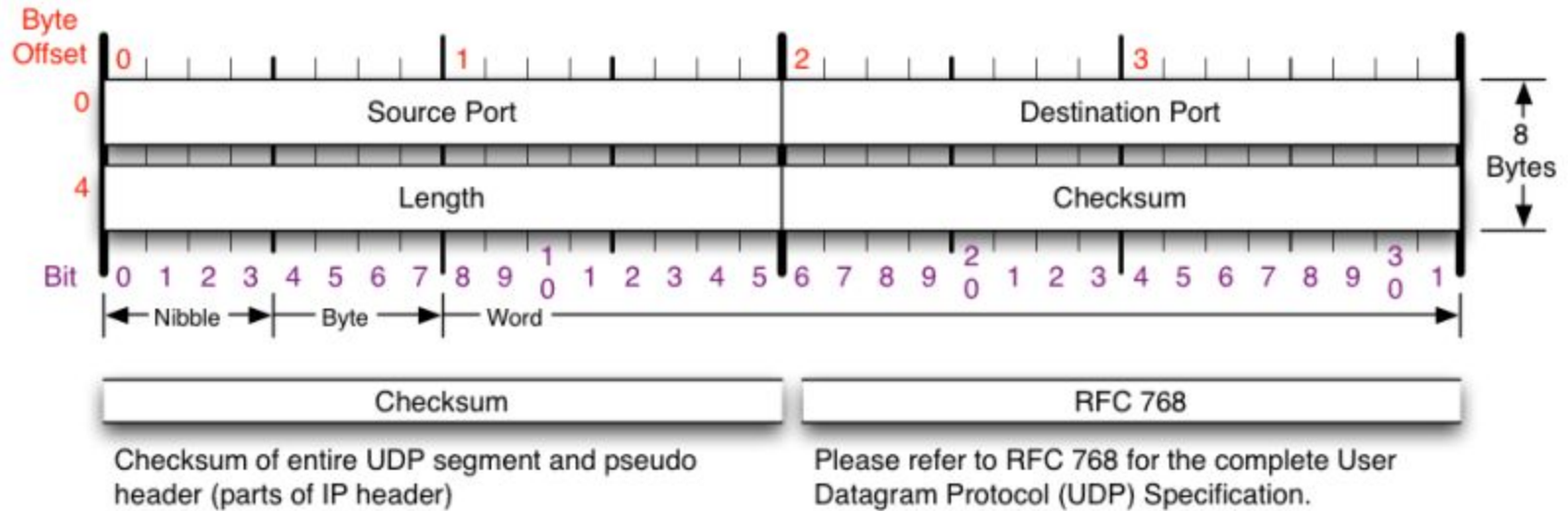
x D M

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

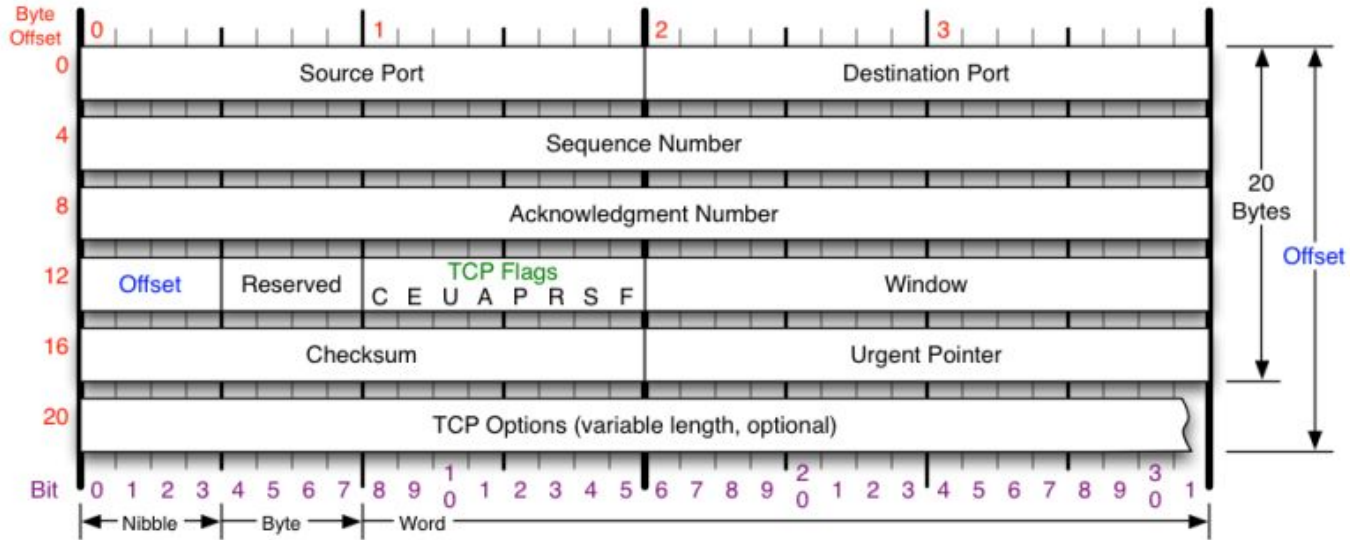
RFC 791

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

UDP



TCP



TCP Flags

C E U A P R S F

Congestion Window

C 0x80 Reduced (CWR)
 E 0x40 ECN Echo (ECE)
 U 0x20 Urgent
 A 0x10 Ack
 P 0x08 Push
 R 0x04 Reset
 S 0x02 Syn
 F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Receiver Response	1 1	0 1
Sender Response	1 1	1 1

TCP Options

0 End of Options List
 1 No Operation (NOP, Pad)
 2 Maximum segment size
 3 Window Scale
 4 Selective ACK ok
 8 Timestamp

Checksum

Checksum of entire TCP segment and pseudo header (parts of IP header)

Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.