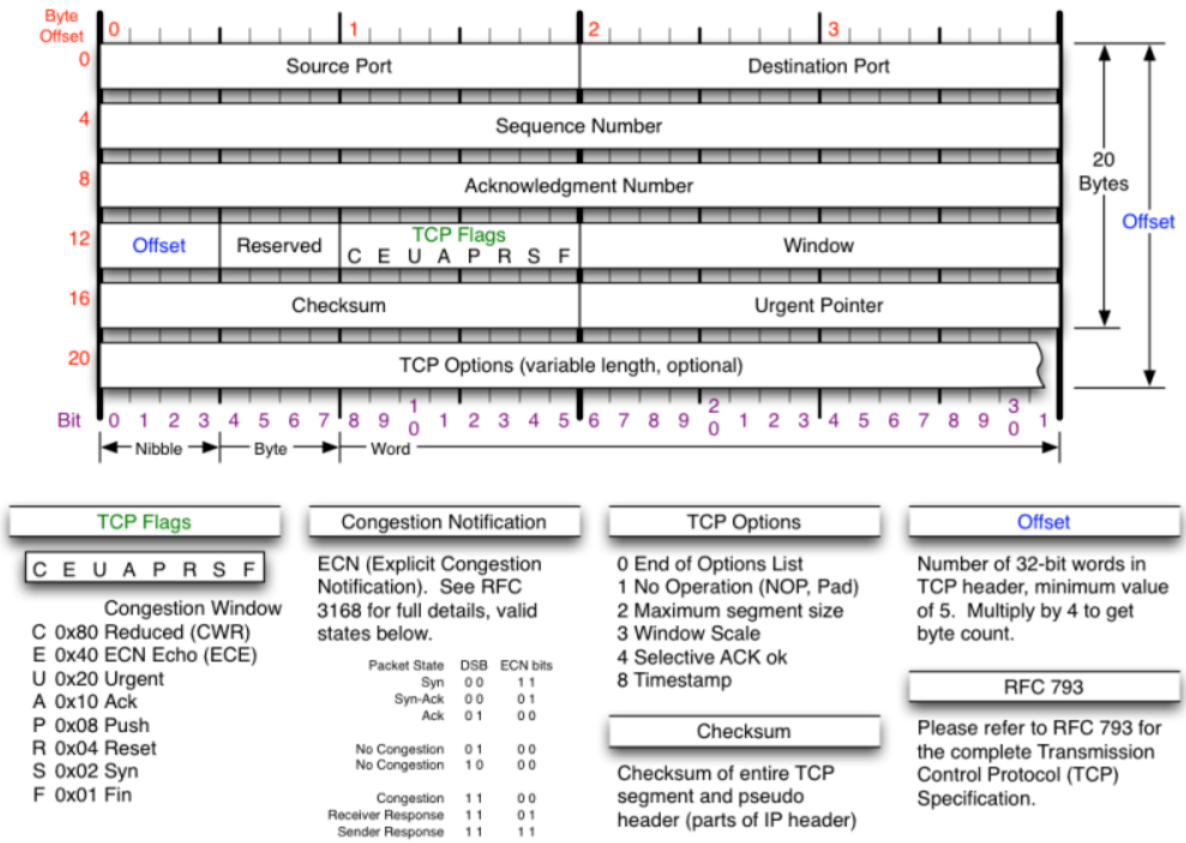
Scan Type	Option	Description
TCP Connect Scan	`-sT`	This is the most basic form of TCP scanning. Nmap attempts to establish a connection to every target port. It's not stealthy as it completes the TCP three-way handshake.
SYN Scan (Stealth Scan)	`-sS`	Also known as half-open scanning, it sends a SYN packet and awaits a response. A SYN/ACK response indicates the port is listening (open), while a RST (reset) is indicative of a non-listener. Does not establish a full TCP connection, making it stealthier than a TCP connect scan.
UDP Scan	`-sU`	Used for identifying open UDP ports. This scan sends a UDP packet to every targeted port. Since UDP is connectionless, the scan types are less conclusive and slower compared to TCP scans.
ACK Scan	`-sA`	This scan is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered. It sends an ACK packet and expects a RST in return.
Window Scan	`-sW`	Similar to ACK scan, but it examines the TCP window size of the RST packets returned to infer whether a port is open or closed. Less common in modern networks.
Maimon Scan	`--scanflags URGACKPSHRSTSYNFIN`	Sends packets with the FIN, PSH, and URG flags set, named after Uriel Maimon who discovered this technique. Useful for evading certain types of firewall rules.
FIN Scan	`-sF`	Sends a FIN packet, which should be ignored by open ports in a compliant TCP stack, potentially bypassing some firewalls or packet filters. Closed ports should respond with a RST.
Xmas Scan	`-sX`	Sends packets with the FIN, PSH, and URG flags set, lighting up the packet like a Christmas tree. Useful for evading firewall rules, similar to the Maimon scan.
Null Scan	`-sN`	Sends a packet with no TCP flags set. This scan type is designed to bypass certain firewall rules, as a compliant TCP stack should ignore such packets.
Idle Scan	`-sI [zombie host]`	A stealthy scan method that uses a "zombie" host to scan a target. It exploits the predictable IP ID flaw to indirectly determine if a port is open or closed.
SCTP INIT Scan	`-sY`	Used to probe SCTP (Stream Control Transmission Protocol) enabled machines. It sends SCTP association initiation requests to determine if ports are listening.
SCTP COOKIE ECHO Scan	`-sZ`	Similar to the SCTP INIT scan, but uses COOKIE ECHO chunks instead. This method is less obtrusive and can bypass some security measures.
IP Protocol Scan	`-sO`	This scan determines which IP protocols (TCP, ICMP, IGMP, etc.) are supported by the target machine. It's different from port scanning as it's looking at IP level protocols.
Custom TCP Scan	`--scanflags [flags]`	For customizing TCP flags in the packet sent to the target, providing flexibility for advanced scanning techniques or research purposes.



This illustration is particularly relevant when discussing Nmap and its various scan techniques. In Nmap scans, different types of TCP packets are crafted and sent to the target system, and the response (or lack thereof) is analyzed to determine the state of the ports. The TCP header plays a crucial role in this process:

1. **Source and Destination Ports**: These fields are used by Nmap to specify the target ports for scanning.
2. **Sequence and Acknowledgment Numbers**: These are used in establishing and maintaining a connection. Nmap manipulates these values in different scans, especially in SYN scans where it initiates a connection without completing the TCP three-way handshake.
3. **Flags**: This is where Nmap's functionality shines. Nmap manipulates the TCP flags (URG, ACK, PSH, RST, SYN, FIN) for different scanning techniques. For example:
    - In a SYN scan (**-sS**), Nmap sets the SYN flag to initiate a connection and then listens for a response.
    - In a FIN scan (**-sF**), the FIN flag is set, and compliant TCP stacks should ignore such packets if the port is open.
    - Xmas (**-sX**) and Null (**-sN**) scans use unusual combinations of flags to probe systems in a way that might evade detection.
4. **Window Size**: This field can be used in certain types of scans (like Window scan **-sW**) to infer the state of a port based on how the target system's TCP stack responds.
5. **Checksum**: Nmap ensures the correctness of its packet construction through the checksum.
6. **Other Fields (Data Offset, Reserved, Urgent Pointer, Options)**: Typically, less manipulated in standard Nmap scans but are integral to the structure and functionality of TCP packets.

Understanding the TCP header structure is essential for comprehending how Nmap operates and interprets responses from target systems and is invaluable in both offensive and defensive cybersecurity practices.