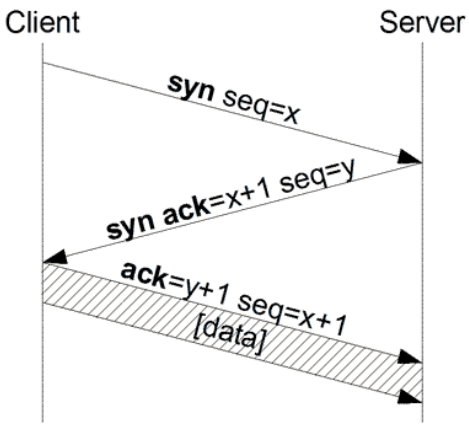
10.3 Aufbau von TCP-Verbindungen

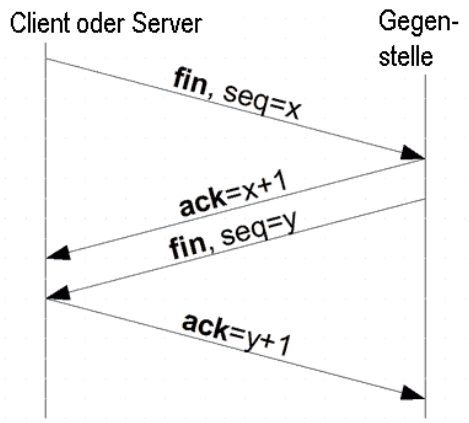


Beim Aufbau einer TCP-Verbindung kommt der so genannte **Drei-Wege-Handshake** zum Einsatz. Der Rechner, der die Verbindung aufbauen will, sendet dem anderen ein SYN-Paket (von engl. *synchronize*) mit einer Sequenznummer x. Die Sequenznummern sind dabei für die Sicherstellung einer vollständigen Übertragung in der richtigen Reihenfolge und ohne Duplikate wichtig. Es handelt sich also um ein Paket, dessen SYN-Bit im Paketkopf gesetzt ist (siehe TCP-Header). Die Start-Sequenznummer ist eine beliebige Zahl, deren Generierung von der jeweiligen TCP-Implementierung abhängig ist. Sie sollte jedoch möglichst zufällig sein, um Sicherheitsrisiken zu vermeiden.

Die Gegenstelle **(siehe Skizze)** empfängt das Paket. Ist der Port geschlossen, antwortet sie mit einem TCP-RST um zu signalisieren, dass keine Verbindung aufgebaut werden kann. Ist der Port geöffnet, sendet sie in einem eigenen SYN-Paket im Gegenzug ihre Start-Sequenznummer y (die ebenfalls beliebig und unabhängig von der Start-Sequenznummer der Gegenstelle ist). Zugleich bestätigt sie den Erhalt des ersten SYN-Pakets, indem sie die Sequenznummer x um eins erhöht und im ACK-Teil (von engl. *acknowledgment* = Bestätigung) des Headers zurückschickt.

Der Client bestätigt zuletzt den Erhalt des SYN/ACK-Pakets durch das Senden eines eigenen ACK-Pakets mit der Sequenznummer y+1. Dieser Vorgang wird auch als „*Forward Acknowledgement*" bezeichnet. Außerdem sendet der Client den Wert x+1 aus Sicherheitsgründen ebenso zurück. Dieses ACK-Segment erhält der Server, das ACK-Segment ist durch das gesetzte ACK-Flag gekennzeichnet. Die Verbindung ist damit aufgebaut.

10.4 Verbindungsabbau



Der geregelte Verbindungsabbau erfolgt ähnlich. Statt des SYN-Bits kommt das FIN-Bit (von engl. *finish* = Ende, Abschluss) zum Einsatz, welches anzeigt, dass keine Daten mehr vom Sender kommen. Der Erhalt des Pakets wird wiederum mittels ACK bestätigt. Der Empfänger des FIN-Pakets sendet zuletzt seinerseits ein FIN-Paket, das ihm ebenfalls bestätigt wird.

Obwohl eigentlich vier Wege genutzt werden, handelt es sich beim Verbindungsabbau auch um einen Drei-Wege-Handshake, da die ACK- und FIN-Operationen vom Server zum Client als ein Weg gewertet werden.

Zudem ist ein verkürztes Verfahren möglich, bei dem FIN und ACK genau wie beim Verbindungsaufbau im selben Paket untergebracht werden. Die *maximum segment lifetime* (MSL) ist die maximale Zeit, die ein Segment im Netzwerk verbringen kann, bevor es verworfen wird. Nach dem Senden des letzten ACKs wechselt der Client in einen zwei MSL andauernden Wartezustand (*Waitstate*), in dem alle verspäteten Segmente verworfen werden. Dadurch wird sichergestellt, dass keine verspäteten Segmente als Teil einer neuen Verbindung fehlinterpretiert werden. Außerdem wird eine korrekte Verbindungsterminierung sichergestellt. Geht ACK y+1 verloren, läuft beim Server der Timer ab, und das LAST_ACK Segment wird erneut übertragen.

10.5 Beispiel für eine TCP-Datenübertragung

Der Sender schickt sein erstes TCP-Segment mit einer Sequenznummer SEQ=1 (variiert) und einer Nutzdatenlänge von 1460 Byte an den Empfänger. Der Empfänger bestätigt es mit einem TCP-Header ohne Daten mit ACK=1461 und fordert damit das zweite TCP -Segment ab dem Byte Nummer 1461 beim Sender an. Dieser schickt es dann mit einem TCP-Segment und SEQ=1461 an den Empfänger. Dieser bestätigt es wieder mit einem ACK=2921 und so weiter. Der Empfänger braucht nicht jedes TCP-Segment zu bestätigen, wenn diese zusammenhängend sind. Empfängt er die TCP-Segmente 1–5, so braucht er nur das letzte TCP-Segment zu bestätigen. Fehlt zum Beispiel das TCP-Segment 3, weil es verloren gegangen ist, so kann er nur die 1 und die 2 bestätigen, 4 und 5 jedoch noch nicht. Da der Sender keine Bestätigung für die 3 bekommt, läuft sein Timer ab, und er verschickt die 3 noch einmal. Kommt die 3 beim Empfänger an, so bestätigt er alle fünf TCP-Segmente. Der Sender startet für jedes TCP-Segment, welches er auf die Reise schickt, einen Timer (RTT).

Ablauf einer TCP-Datenübertragung (Quelle: Wikipedia)