UDP (User Datagram Protocol) Cheatsheet

#NWT #EBE

Overview

UDP ist ein verbindungsloses, nicht bestätigendes Transportprotokoll, das auf dem Internet Protocol (IP) aufbaut.

Eigenschaften

- UDP ist schnell und einfach, da es keine Verbindungsherstellung, Bestätigungen oder Wiederholungen bietet.
- Es bietet keine Garantie für die Zustellung von Paketen oder die Reihenfolge, in der sie ankommen.
- Es ist ein "Best Effort"-Protokoll, das für Anwendungen geeignet ist, die eine schnelle Übertragung von Daten benötigen, bei denen eine geringfügige Verlusttoleranz akzeptabel ist.

Vorteile

- Geringer Overhead: Keine Verbindungsherstellung oder Bestätigungen bedeuten weniger Paketüberkopf.
- Schnelle Übertragung: Es bietet eine geringe Latenz und eignet sich gut für echtzeitkritische Anwendungen wie VoIP oder Videostreaming.

Nachteile

- Keine Fehlerkorrektur: UDP bietet keine Mechanismen zur Fehlererkennung oder -korrektur. Fehlerhafte Pakete werden verworfen.
- Keine Zuverlässigkeit: Es gibt keine Garantie dafür, dass Pakete ankommen oder in der richtigen Reihenfolge eintreffen.

Verwendungszwecke

- Echtzeitkommunikation: VoIP, Videostreaming, Online-Gaming.
- DNS (Domain Name System): Zur Auflösung von Domainnamen in IP-Adressen.
- SNMP (Simple Network Management Protocol): Zur Netzwerküberwachung und -verwaltung.

Headerstruktur

Feld	Größe (in Bytes)	Beschreibung
Source Port	2	Senderport
Destination Port	2	Empfängerport
Length	2	Länge des UDP-Datagramms (inklusive Header)
Checksum	2	Prüfsumme für Integritätsprüfung

Ports

- Ports dienen dazu, den Datenverkehr an die entsprechenden Anwendungen auf dem Zielhost weiterzuleiten.
- Standardport für UDP-Verbindungen: 1-65535

- Portnummern unter 1024 sind reserviert für bekannte Dienste und erfordern erhöhte Berechtigungen für die Verwendung.

Beispiel

```
import socket

# Sender
sender = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
receiver_address = ('localhost', 12345)
message = 'Hello, UDP!'
sender.sendto(message.encode(), receiver_address)
sender.close()

# Empfänger
receiver = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
receiver.bind(('localhost', 12345))
data, sender_address = receiver.recvfrom(1024)
print(f"Received message: {data.decode()} from {sender_address}")
receiver.close()
```