7.2 Adressierung

ZUSATZ

7.2.1 IP-Adressen

Damit ein Rechner im Netz ansprechbar ist, braucht er eine eindeutige Adresse. Im Protokollstapel TCP/IP ist das die **IP-Adresse**.

Eine IPv4-Adresse umfasst 32 Bit, aufgeteilt in vier Gruppen zu 8 Bit (1 Byte), dezimal geschrieben als Werte von 0 bis 255 — zum Beispiel `192.168.1.1`. Damit sind $2^{32} = 4\,294\,967\,296$ Adressen möglich.

Jede Adresse zerfällt in zwei Teile: die **Netz-ID** vorne und die **Host-ID** hinten. Welcher Teil wie lang ist, legt die **Subnetzmaske** fest: Sie enthält vorne lauter Einsen und hinten lauter Nullen.

	Binär	Dezimal
Rechner 3	`11000000 10101000 00001001 00000011`	192.168.9.3
Rechner 65	`11000000 10101000 00001001 01000001`	192.168.9.65
Subnetzmaske	`11111111 11111111 11111111 00000000`	255.255.255.0

Beide Rechner gehören zum selben Netz, weil sie in den ersten drei Bytes übereinstimmen — genau dort, wo die Maske Einsen hat.

Netz-ID berechnen. Man schreibt IP-Adresse und Maske binär untereinander und verknüpft sie Bit für Bit mit UND.

Beispiel: IP `192.168.79.5`, Maske `255.255.192.0`

	Binär	Dezimal
IP-Adresse	`11000000 10101000 01001111 00000101`	192.168.79.5
Subnetzmaske	`11111111 11111111 11000000 00000000`	255.255.192.0
UND-Verknüpfung	`11000000 10101000 01000000 00000000`	**192.168.64.0**

Die Netz-ID ist also `192.168.64.0`; für die Host-ID bleiben 14 Bit.

Wie viele Rechner passen in ein Netz? Bei der Maske `255.255.255.0` bleiben 8 Bit für die Host-ID, also $2^8 = 256$ Kombinationen. Zwei davon sind vergeben: lauter Nullen bezeichnet das Netz selbst, lauter Einsen ist die Broadcast-Adresse. Es bleiben **254** nutzbare Adressen.

Allgemein: Bei h Host-Bits sind es $2^h - 2$ Rechner.

7.2.2 MAC-Adressen und ARP

Jede Netzwerkkarte besitzt eine **MAC-Adresse** (Media Access Control) mit 48 Bit, die der Hersteller fest einbrennt und die weltweit eindeutig ist. Sie wird in sechs Zweiergruppen hexadezimal geschrieben, etwa `00:1A:2B:3C:4D:5E`; die ersten drei Bytes kennzeichnen den Hersteller.

Innerhalb eines Netzes werden Pakete anhand der MAC-Adresse zugestellt, nicht anhand der IP. Die Übersetzung von IP nach MAC erledigt das **ARP** (Address Resolution Protocol): Der Rechner fragt per Rundruf „Wer hat 192.168.1.7?" und merkt sich die Antwort im ARP-Cache.

Merke

Die MAC-Adresse gehört zur **Hardware** und wird nicht zugewiesen; die IP-Adresse gehört zum **Netz** und wird vergeben. Deshalb gilt die MAC-Adresse nur innerhalb des eigenen Netzes — über den Router hinaus interessiert sie niemanden.

Stolperstein

„Unveränderlich" stimmt nur für die eingebrannte Adresse selbst. Softwareseitig lässt sie sich überschreiben, und aktuelle Betriebssysteme tun das von sich aus: Smartphones und Notebooks melden sich in einem WLAN, das sie nicht kennen, mit einer **zufällig erzeugten** MAC-Adresse an und benutzen für jedes Netz eine andere. Der Grund ist Datenschutz — sonst ließe sich ein Gerät allein am Funkverkehr über Orte hinweg wiedererkennen. Für die Schule heißt das: Dasselbe Notebook kann im Schul-WLAN und im Heimnetz mit verschiedenen MAC-Adressen auftauchen. Für die Zustellung im Netz ändert das nichts — eindeutig muss die Adresse nur dort sein, wo sie gerade benutzt wird.

7.2.3 IPv6

Der IPv4-Adressraum ist erschöpft; die letzten freien Blöcke wurden bereits vergeben. Deshalb gibt es **IPv6** mit 128 Bit Adresslänge, also $2^{128} \approx 3{,}4 \cdot 10^{38}$ Adressen. Geschrieben wird hexadezimal in acht Gruppen, etwa `2001:0db8:85a3:0000:0000:8a2e:0370:7334`, wobei führende Nullen und eine längste Nullfolge abgekürzt werden dürfen: `2001:db8:85a3::8a2e:370:7334`.

IPv6 ist längst kein Versuchsstadium mehr: Ein erheblicher Teil des weltweiten Verkehrs läuft darüber, meist parallel zu IPv4 auf demselben Gerät.

7.2.4 Die Netz-ID im Programm

Das bitweise UND, das Sie oben von Hand gerechnet haben, erledigt jede Programmiersprache mit einem einzigen Operator: `&` in Python, `and` in Pascal. Damit wird aus der Rechnung ein Dreizeiler. Die zweite Funktion zählt die Einsen der Maske und bestimmt daraus die Zahl der nutzbaren Adressen.

Python

```python
def netz_id(ip, maske):
    teile_ip = [int(t) for t in ip.split(".")]
    teile_maske = [int(t) for t in maske.split(".")]
    return ".".join(str(a & b) for a, b in zip(teile_ip, teile_maske))


def anzahl_hosts(maske):
    bits = sum(bin(int(t)).count("1") for t in maske.split("."))
    return 2 ** (32 - bits) - 2


print(netz_id("192.168.79.5", "255.255.192.0"))   # 192.168.64.0
print(anzahl_hosts("255.255.255.0"))              # 254
```

Object Pascal

```
function NetzID(const ip, maske: string): string;
var
  teileIP, teileMaske: array[1..4] of Integer;
  i, p: Integer;
  restIP, restMaske: string;
begin
  restIP := ip + '.';
  restMaske := maske + '.';
  for i := 1 to 4 do
  begin
    p := Pos('.', restIP);
    teileIP[i] := StrToInt(Copy(restIP, 1, p - 1));
    Delete(restIP, 1, p);
    p := Pos('.', restMaske);
    teileMaske[i] := StrToInt(Copy(restMaske, 1, p - 1));
    Delete(restMaske, 1, p);
  end;

  Result := '';
  for i := 1 to 4 do
  begin
    Result := Result + IntToStr(teileIP[i] and teileMaske[i]);
    if i < 4 then Result := Result + '.';
  end;
end;
```

Arbeitsauftrag 2

5. Bestimmen Sie Netz-ID und Host-ID für `10.0.5.130` mit der Maske `255.255.255.192`.
6. Wie viele Rechner können in einem Netz mit der Maske `255.255.255.128` sein?
7. Ein Netz braucht Platz für 500 Rechner. Welche Subnetzmaske wählen Sie? Begründen Sie.
8. Gehören `172.16.4.9` und `172.16.5.9` bei der Maske `255.255.254.0` zum selben Netz? Rechnen Sie es binär vor.
9. Wie viele Adressen sind bei IPv6 möglich? Vergleichen Sie die Größenordnung mit der Zahl der Sandkörner der Erde (geschätzt 10^{20}).
10. Ergänzen Sie das Programm aus Abschnitt 7.2.4 um eine Funktion, die zusätzlich die **Broadcast-Adresse** ausgibt — also die Netz-ID mit lauter Einsen im Host-Teil. Prüfen Sie Ihre Fassung an den Beispielen aus Aufgabe 5 und Aufgabe 8.
11. Schreiben Sie `anzahl_hosts` auch in Pascal. Hinweis: Zählen Sie die Einsen der Maske, indem Sie jedes Byte fortlaufend halbieren und den Rest aufsummieren.