Lehrstuhl für Betriebssysteme
Fakultät für Informatik
Technische Universität München

Grundlagen Rechnernetze und Verteilte Systeme

IN0010, SoSe 2018

Übungsblatt 11

2. Juli – 6. Juli 2018

Hinweis: Mit * gekennzeichnete Teilaufgaben sind ohne Lösung vorhergehender Teilaufgaben lösbar.

Aufgabe 1 Network Address Translation

In dieser Aufgabe soll die Weiterleitung von IP-Paketen (IPv4) bei Verwendung eines NAT-fähigen Routers betrachtet werden. Für die Zuordnung zwischen öffentlichen und privaten IP-Adressen verfügt ein NAT-fähiger Router über eine Abbildungstabelle, die die Beziehung zwischen lokalem und globalem Port speichert. Viele NAT-fähige Geräte speichern zusätzlich noch weitere Informationen wie die entfernte IP-Adresse oder die eigene globale IP-Adresse (z. B. wenn der Router mehr als eine globale IP besitzt). Davon wollen wir hier absehen.

Abbildung 2 zeigt die Netztopologie. Router R1 habe NAT aktiviert, wobei auf IF1 eine private und auf IF2 eine öffentliche IP-Adresse verwendet werde. Router R2 nutze kein NAT. PC2 habe bereits mit Server 2 kommuniziert, wodurch der Eintrag in der NAT-Tabelle von R1 entstanden ist (siehe Abbildung 2). Wählen Sie dort, wo Sie die Freiheit haben, sinnvolle Werte für die IP-Adressen und Portnummern.

a)* Geben Sie PC1 und Interface 1 von R1 eine passende IP-Adresse. Das Subnetz ist 10.0.0.0/24.

Möglich sind zum Beispiel:

- PC1: 10.0.0.1
- R1 IF1: 10.0.0.254

b)* PC1 sende baue nun eine HTTP-Verbindung zu Server 2 auf. Geben Sie die Felder für die Quell-IP, Ziel-IP, Quell-Port, Ziel-Port und TTL des IP- bzw. TCP-Headers für die Pakete an den drei markierten Stellen in Abbildung 2 an. Geben Sie außerdem neu entstehende Einträge in der NAT-Tabelle von R1 an.

Siehe Abbildung 2.

- **Zwischen PC1 und R1:** TTL = 64
  Wichtig ist beim Quell-Port, dass dieser größer als 1023 ist (da Nummern kleiner 1024 Well-Known-Ports repräsentieren und nicht als Quell-Ports verwendet werden). Außerdem sollte er nicht größer sein als 65535, da Portnummern 16 bit lang sind. Der Zielport ist mit TCP 80 (HTTP) vorgegeben.
- **R1 und R2** TTL = 63
  R1 tauscht die private Quell-IP durch seine eigene öffentliche IP-Adresse aus. Der Quell-Port wird (wenn nicht schon anderweitig belegt) für gewöhnlich beibehalten. Andernfalls wird auch dieser geändert, z. B. inkrementiert. Die genaue Wahl der Portnummer hängt vom jeweiligen NAT-Typ ab. Wir behalten die Portnummern sofern möglich einfach bei. An dieser Stelle wird auch ein neuer Eintrag in der NAT-Tabelle erzeugt: [10.0.0.1, 3627, 3627].
- **Zwischen R2 und Server 2** TTL = 62
  Keine Änderung, da ein gewöhnlicher Router IP-Adressen und Portnummern nicht verändert. Die TTL wird aber natürlich dekrementiert.

c) Server 2 antworte nun PC1. Geben Sie in Abbildung 3 analog zur vorherigen Teilaufgabe die Header-Felder an den drei benannten Stellen sowie neu entstehende Einträge in der NAT-Tabelle von R1 an.

Wir nehmen an, dass der Server Pakete mit TTL = 64 versendet.

- **Zwischen Server 2 und R2** TTL = 64
  Der Server adressiert die Antwort zunächst an R1 (wohin auch sonst?).

- **Zwischen R2 und R1** TTL = 63
  R2 ändert (außer der TTL) nichts.
- **Zwischen R1 und PC1:** TTL = 62
  R1 nutzt den Eintrag in der seiner NAT-Tabelle um die private IP-Adresse des tatsächlichen Empfängers zu ermitteln. Anschließend werden Ziel-IP und Ziel-Port (wenn nötig) ausgetauscht und das Paket weitergeleitet.

d)* Server 1 baut nun ebenfalls eine TCP-Verbindung zu Server 2 auf Port 80 auf. Dabei wählt er zufällig den Absender-Port 13059. Beschreiben Sie das am NAT auftretende Problem und wie dieses gelöst wird.

Es gibt eine Kollision mit dem ersten Eintrag in der NAT-Tabelle: Der NAT-Router kann bei Antworten von Server 2 nicht mehr unterscheiden, ob diese für PC1 oder Server 2 bestimmt sind, da als einziges Unterscheidungsmerkmal die globale Portnummer existiert.

Die Lösung besteht darin, dass der NAT-Router vor der Erzeugung neuer Einträge prüft, ob der jeweilige Port bereits in Verwendung ist. Ist dies der Fall, wählt der NAT-Server eine zufällige Portnummer aus dem Bereich der Ephemeral Ports (oder inkrementiert die Portnummer) und speichert sowohl die lokale als auch die neue globale Portnummer ab. Bei eingehenden Paketen wird in den L4-PDUs die Portnummer zurückübersetzt.

e)* R1 erhält von PC3 ein an `131.159.24.19:13059` adressiertes Paket. Wie wird R1 mit diesem Paket verfahren? Welche Probleme können sich daraus ergeben?

R1 wird die Zieladresse des Pakets gemäß der NAT Tabelle übersetzten und an PC2 weiterleiten, obwohl der ursprüngliche Eintrag für Server2 angelegt wurde. PC2 erhält ein „unerwartetes“ Packet und muss damit umgehen können. Die fälschlicherweise oft angenommene Firewallfunktion des NAT kann hierbei nicht ermöglicht werden.

f) Ergibt sich für PC2 ein Problem, wenn dieser ein „zufälliges“ Paket mit TCP-Payload auf einem Port mit einer bestehenden Verbindung erhält?

Das Paket besitzt wahrscheinlich eine andere Absender-IP und einen anderen Source Port und wird somit nicht der bestehenden Verbindung zugeordnet. Wenn Absender-IP und Source Port „zufällig“ übereinstimmen, so fällt die Sequenznummer des Pakets (mit hoher Wahrscheinlichkeit) nicht in den aktuell gültige Empfangsfenster und wird somit verworfen.

g)* Welche weiteren Unterscheidungskriterien könnten von einem NAT-Router verwendet werden?

Globale IP (wenn mehrere Interfaces/IP Addressen am Router konfiguriert sind), remote IP, remote Port sowie die Protokollnummer (TCP oder UDP).

h)* Welches Problem tritt auf, wenn PC1 einen Echo Request an Server 2 sendet?

Da ICMP keine Portnummern verwendet, kann der NAT-Router keinen Eintrag erzeugen. Die Antwort wird daher verworfen.

i) Beschreiben Sie eine mögliche Lösung für das in der vorherigen Teilaufgabe aufgetretene Problem.

Der NAT-Router könnte im Falle von ICMP Paketen zusätzlich zur Protokollnummer den ICMP-Identifier als Ersatz für die fehlenden Portnummern verwenden. In diesem Fall muss der NAT-Router aber in jedem Fall auch zwischen den IP-Protokollen (TCP, UDP, ICMP usw.) unterscheiden.

j) Welches Problem ergibt sich, wenn ein NAT-Router ICMP TTL-Exceeded Nachrichten empfängt und an den Empfänger (Absender des auslösenden Pakets) weiterleiten möchte? Wie kann dieses Problem umgangen werden?

TTL-Exceeded Nachrichten sind eigene ICMP Nachrichten, deren Identifier nicht im NAT eingetragen wurde (Nachrichten werden nicht im eigenen Netzwerk generiert, sondern von Rechnern außerhalb). Eine Zuordnung zum Empfänger ist somit nicht möglich. ICMP TTL Exceeded enthalten neben dem ICMP Header auch den IP Header und die ersten 8 Payload Bytes des auslösenden Pakets[1]. Darüber kann das NAT die auslösende

[1] Nach RFC 4443 (ICMPv6): „As much of invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU.“ Echte Implementierungen setzen dies für IPv4 analog um.

Verbindung Identifizieren. Bei TCP und UDP sind hier die Portnummern zu finden, bei ICMP Nachrichten der ursprüngliche Identifier.

k)* Nun möchte PC3 eine HTTP-Verbindung zu Server 1 aufbauen. Kann dies unter den gegebenen Umständen funktionieren? (Begründung!)

PC3 kann das Paket nicht direkt an die Adresse 10.0.0.10 adressieren, da es sich hierbei um eine private IP-Adresse handelt, welche im Internet nicht geroutet wird. Wenn PC3 die öffentliche IP von R1 kennt, hinter dem sich Server 1 befindet, so kann er das Paket zwar an die IP-Adresse von R1 und TCP80 adressieren. R1 hat jedoch (soweit aus der Aufgabenstellung hervorgeht) keinen passenden Eintrag in der NAT-Tabelle und kann daher den Empfänger des Pakets nicht ermitteln.

l) Wie könnte das Problem unter Beibehaltung des NATs umgangen werden?

Im NAT kann eine statische Weiterleitung, ein sogenanntes Portforwarding, eingetragen werden.
Beispiel: 10.0.0.10 80 80 Darüber kann Server 1 auf der IP Adresse von R1 über den Router R1 von außen auf Port 80 erreicht werden.

Aufgabe 2 Domain Name System (DNS)

Hinweis: Angelehnt an Endterm 2015

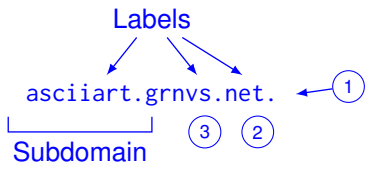
Zentrale Aufgabe des Domain Name Systems (DNS) ist es, menschenlesbare Namen auf IP-Adressen abzubilden, die dann für die Wegwahl auf der Netzwerkschicht verwendet werden können. Bei dem Namen `asciiart.grnvs.net.` handelt es sich um einen sog. *Fully Qualified Domain Name (FQDN)*.

a)* Was ist der Unterschied zwischen einem vollqualifizierten Domain Name (FQDN) und einem nicht-(voll)qualifizierten?

Ein FQDN endet stets mit `.`, d. h. der Wurzel des Name Spaces. Ein nicht-qualifizierter Domain Name hingegen kann ein einzelnes Label oder eine geordnete Liste durch Punkte getrennter Labels sein, die relativ zu einer anderen Wurzel als `.` zu sehen sind.

b)* Benennen Sie die einzelnen Bestandteile des FQDNs, sofern es dafür gängige Bezeichnungen gibt.

1. Root (Beginn des Namensraums)
2. Top Level Domain (TLD)
3. Second Level Domain



Da im Alltag zumeist nicht explizit zwischen einem „FQDN“ (also mit terminierenden Punkt) und „Domain Name“ (also ohne terminierenden Punkt) unterschieden wird, da es kontextabhängig klar ist, was von beiden gerade gemeint ist, werden wir[2] im Folgenden auch nur noch dann den Root-Punkt setzen, wenn wir dies besonders hervorheben bzw. deutlich machen wollen.

In Abbildung 1 sind ein PC sowie eine Reihe von Servern dargestellt. Wir nehmen an, dass PC1 den Router als Resolver nutzt. Der Router wiederrum nutzt einen Resolver von Google unter der IP-Adresse 8.8.8.8 zur Namensauflösung. Ferner nehmen wir an, dass der Google-Resolver gerade neu gestartet wurde (also insbesondere keine Resource Records gecached hat) und rekursive Namensauflösung anbietet.

Die autoritativen Nameserver für die jeweiligen Zonen sind in Tabelle 1 gegeben.

Zone	autoritativer Nameserver
`.`	`d.root-servers.net.`
`com., net.`	`a.gtld-servers.net.`
`google.com.`	`ns1.google.com.`
`grnvs.net.`	`bifrost.grnvs.net.`

Tabelle 1: Zonen mit zugehörigen autoritativen Nameservern

c)* Erläutern Sie den Unterschied zwischen einem *Resolver* und einem *Nameserver*.

Nameserver sind autoritativ für eine oder mehrere Zonen („Bereiche“), d. h. sie besitzen eine gültige und aktuelle Kopie der gesamten Zone, für die sie autoritativ sind.

Resolver hingegen extrahieren mittels eine Reihe iterativer Anfragen an die jeweils autoritativen Nameserver die benötigten Information aus dem DNS und geben diese an den anfragenden Client zurück. Resolver können Einträge für begrenzte Zeit cachen, so dass bei erneuter Anfrage derselben Resource Records der Prozess nicht wiederholt werden muss.

d)* Welche Funktion erfüllen `d.root-servers.net` und `a.gtld-servers.net`?

Der Root-Nameserver ist autoritativ für die Rootzone, d. h. er kennt die Nameserver, welche für die einzelnen TLDs verantwortlich sind, so z. B. `a.gtld-servers.net` als einen der autoritativen Nameserver für net-Domains.

[2] for the sake of notational brevity

`a.gtld-servers.net` kennt wiederum die zuständigen Nameserver für alle Second-Level-Domains unterhalb der net-TLD.

e)* Erklären Sie den Unterschied zwischen iterativer und rekursiver Namensauflösung.

Rekursive Namensauflösung bedeutet, dass eine DNS-Anfrage an einen Resolver gestellt wird. Dieser wird das endgültige Ergebnis zurücksenden.

Bei iterativer Auflösung hingegen werden schrittweise die autoritativen Nameserver der einzelnen Zonen angefragt.

f) Zeichnen Sie in Abbildung 1 alle DNS-Nachrichten (Requests / Responses) ein, die ausgetauscht werden, sobald PC1 auf `asciiart.grnvs.net.` zugreift. Nummerieren Sie die Nachrichten gemäß der Reihenfolge, in der sie zwischen den einzelnen Knoten ausgetauscht werden.

s. Abbildung 1.

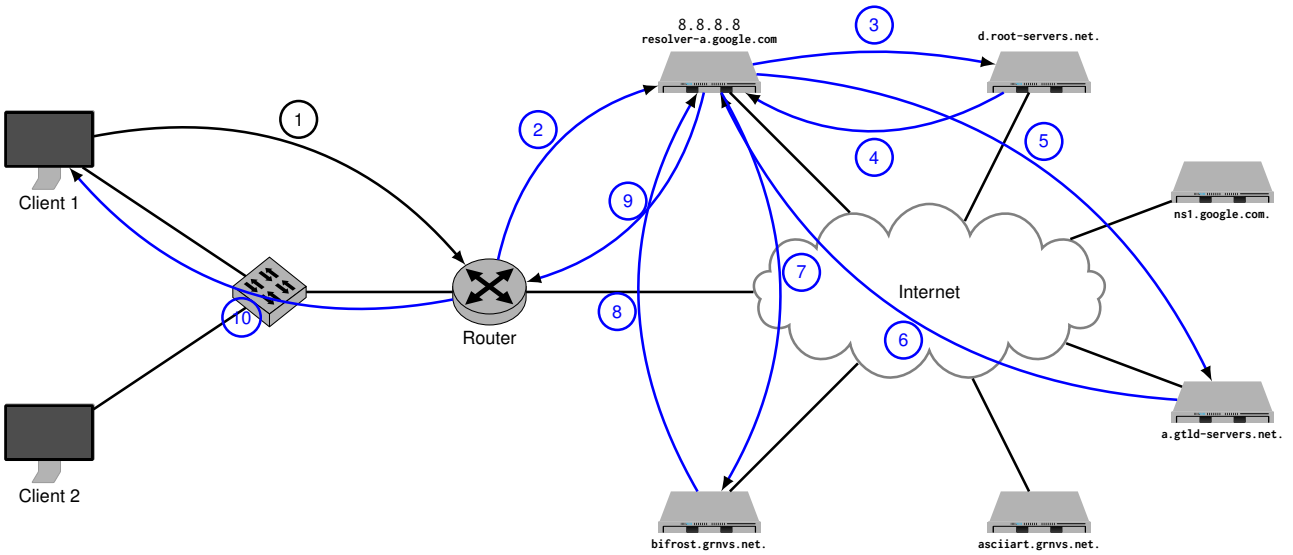


Abbildung 1: Vorlage zu Aufgabe 2f)

g)* Wie wird im DNS sichergestellt, dass kein bösartiger Nameserver Anfragen für andere Domänen beantwortet? (Wir gehen davon aus, dass keine Man-in-the-Middle-Angriffe möglich sind.)

Dies wird lediglich indirekt dadurch sichergestellt, dass während der iterativen Namensauflösung stets nur die jeweils autoritativen Nameserver kontaktiert werden. Sofern die

- Antwort des Rootservers zuverlässig war und
- die Antwort auf dem Weg vom Rootserver zum anfragenden Nameserver nicht modifiziert wurde

kann ein bösartiger Nameserver keine falschen Antworten liefern – eben da er nie gefragt wird.

Selbstverständlich wird auf diese Weise nicht verhindert, dass DNS-Antworten mittels Man-in-the-Middle-Attacken abgefangen und modifiziert werden können. Dagegen helfen lediglich kryptographische Verfahren, wie sie in der DNSSEC-Erweiterung zu finden sind (nicht in der Vorlesung behandelt).

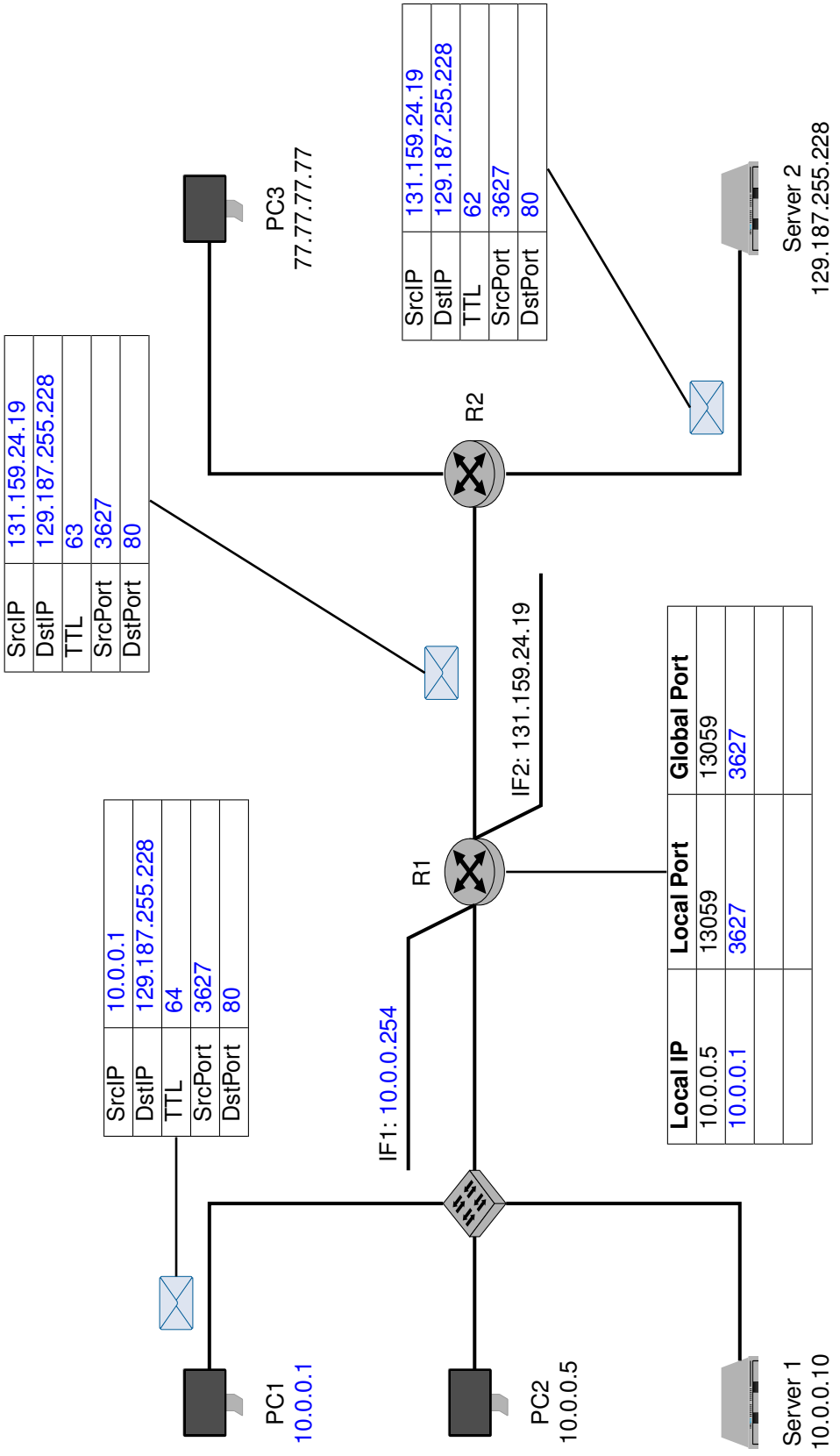


Abbildung 2: Lösungsblatt für Aufgabe 1a)/b)

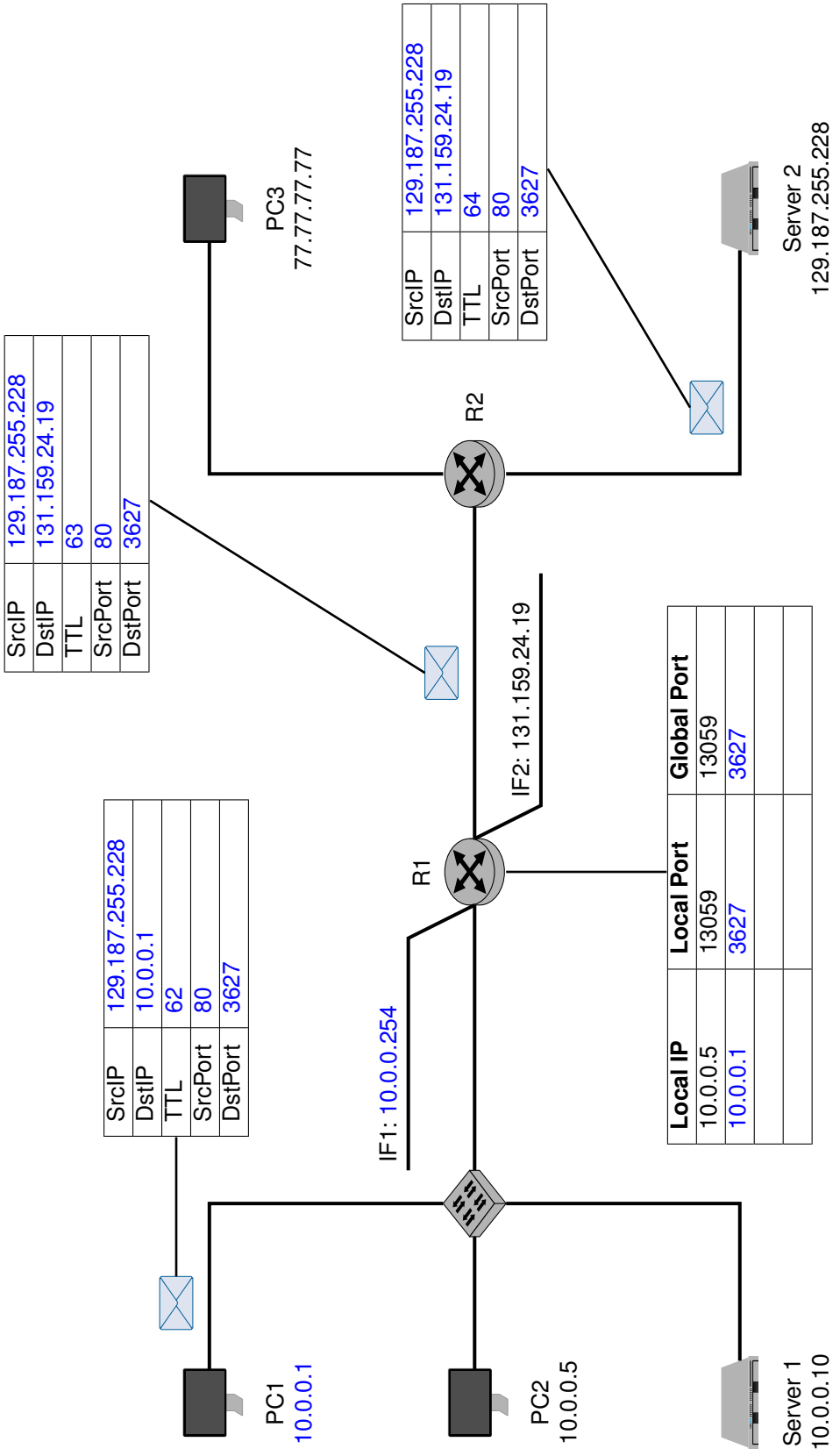


Abbildung 3: Lösungsblatt für Aufgabe 1c)

Aufgabe 3 Kompression: Huffman-Kodierung (Hausaufgabe)

Gegeben sei das Alphabet $\mathcal{A} = \{a, b, c, d\}$ und die Nachricht

$$m = \text{aabcbdacababbbbcbddbbbaababdbdbb} \in \mathcal{A}^{32}.$$

a)* Bestimmen Sie die Auftrittswahrscheinlichkeiten $p_i \in \mathcal{A}$ der einzelnen Zeichen in m.

Aus den Zeichenhäufigkeiten ergibt sich:

$$p_a = \frac{8}{32} = \frac{1}{4},\ p_b = \frac{16}{32} = \frac{1}{2},\ p_c = \frac{3}{32} \approx 0{,}09,\ p_d = \frac{5}{32} \approx 0{,}16$$

b) Bestimmen Sie den Informationsgehalt $I(p_i)$ der einzelnen Zeichen aus $\mathcal{A}$.

Für den Informationsgehalt erhalten wir:

$$\begin{aligned} I(p_a) &= -\log_2(p_a) = 2\,\text{bit} \\ I(p_b) &= -\log_2(p_b) = 1\,\text{bit} \\ I(p_c) &= -\log_2(p_c) \approx -1{,}6 + 5 = 3{,}4\,\text{bit} \\ I(p_d) &= -\log_2(p_d) \approx -2{,}3 + 5 = 2{,}7\,\text{bit} \end{aligned}$$

c) Die Nachricht m stamme aus einer Nachrichtenquelle X. Bestimmen Sie auf Basis der bisherigen Ergebnisse die Quellenentropie $H(X)$.

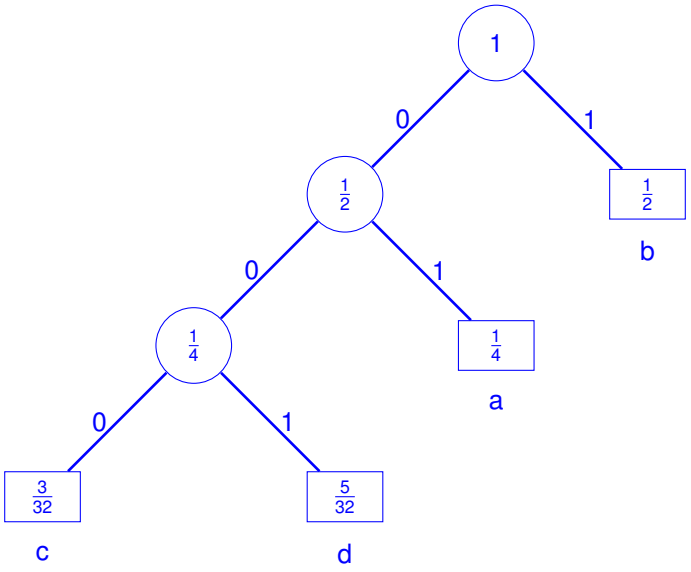
Die Quellenentropie ist nichts weiter als die mit den Auftrittswahrscheinlichkeiten gewichtete Summe des Informationsgehalts der Einzelzeichen:

$$H(X) = \sum_{i \in \mathcal{A}} p_i I(p_i) = \frac{1}{4} \cdot 2\,\text{bit} + \frac{1}{2} \cdot 1\,\text{bit} + 0{,}09 \cdot 3{,}4\,\text{bit} + 0{,}16 \cdot 2{,}7\,\text{bit} = 1\,\text{bit} + 0{,}306\,\text{bit} + 0{,}432\,\text{bit} = 1{,}738\,\text{bit}$$

Dies bedeutet, dass sich die Zeichen der Quelle X mit durchschnittlich 1,738 bit pro Zeichen kodieren lassen.

d) Bestimmen Sie nun einen binären Huffman-Code C für diese Nachrichtenquelle.

Siehe Vorlesungsfolien. Beginnend bei den beiden Zeichen mit der geringsten Auftrittswahrscheinlichkeit wird ein Baum beginnend bei den Blättern (den Zeichen) konstruiert. Dabei werden in jedem Schritt stets die beiden Knoten bzw. Blätter zusammengefasst, so dass die Summe der Auftrittswahrscheinlichkeiten über alle Knoten bzw. Blätter minimal ist:



Die Kanten werden mit 0 bzw. 1 beschriftet. Der Code lässt sich nun einfach ablesen, indem man von der Wurzel ausgehend die Kantenbeschriftungen abliest: $C = \{a \mapsto \texttt{01}, b \mapsto \texttt{1}, c \mapsto \texttt{000}, d \mapsto \texttt{001}\}$

Zeichen mit hoher Auftrittswahrscheinlichkeiten erhalten kurze Codewörter. Außerdem lässt sich leicht überprüfen, dass C präfixfrei ist: Kein Codewort ist ein Präfix eines anderen Codeworts. Die Zeichen sind jeweils nur an den Blättern des Baums definiert, nicht jedoch an den inneren Knoten. Dies erleichtert die Dekodierung.

e) Bestimmen Sie die durchschnittliche Codewortlänge von C.

Die durchschnittliche Codewortlänge ergibt sich aus der mit den Auftrittswahrscheinlichkeiten gewichteten Summe der Codewortlängen. Sei $l(c)$ die Länge eines Codeworts in C und $c(i)$ die Funktion, welche ein Zeichen $i \in \mathcal{A}$ auf ein Codewort aus C abbildet. Dann erhalten wir:

$$\dot{l}_C = \sum_{i \in \mathcal{A}} p_i \cdot l(c(i)) = \frac{1}{4} \cdot 2\,\text{bit} + \frac{1}{2} \cdot 1\,\text{bit} + 0{,}09 \cdot 3\,\text{bit} + 0{,}16 \cdot 3\,\text{bit} = 1\,\text{bit} + 0{,}27\,\text{bit} + 0{,}48\,\text{bit} = 1{,}75\,\text{bit}$$

f) Vergleichen Sie die durchschnittliche Codewortlänge von C mit der Codewortlänge eines uniformen[3] Binärcodes.

Der kürzeste uniforme Code hat eine durchschnittliche Codewortlänge von $\dot{l}_U = 2$. Die Ersparnis beträgt also etwa 12,5 %.

[3] Ein Code heißt *uniform*, wenn alle Codewörter dieselbe Länge aufweisen.