TKN WS16/17 - Rechnernetze und Verteilte Systeme - Fragenkatalog

BLOCK 2

Vervöllständigen Sie den Satz: Die Primitive **bind()**...

a. wird dazu verwendet, eine Verbindung auf einem Socket anzunehmen. Der Befehl blockiert, bis eine entsprechende Anfrage eingeht. Falsch

b. initiieren aktiv eine Verbindung auf einem Socket.

c. weisst einem Socket eine lokale Adresse zu.

d. terminiert einen Deskriptor und beendet die Datenübertragung.

Vervollständigen Sie den Satz: Die Primitive **listen()**...

a. ...wird dazu verwendet, eine Verbindung auf einem Socket anzunehmen. Der Befehl blockiert, bis eine entsprechende Anfrage eingeht.

b. ...initiieren aktiv eine Verbindung auf einem Socket.

c. ...sendet eine Nachricht an einen anderen Socket.

d. ...signalisiert die Bereitschaft auf eingehende Verbindungen zu horchen.

Vervollständigen Sie den Satz: Die Primitive **close()**...

a. ...wird dazu verwendet, eine Verbindung auf einem Socket anzunehmen. Der Befehl blockiert, bis eine entsprechende Anfrage eingeht.

b. ...terminiert einen Deskriptor und beendet die Datenübertragung.

c. ...weisst einem Socket eine lokale Adresse zu.

d. ...erstellt einen Socket und gibt einen Deskriptor zurück.

Vervollständigen Sie den Satz: Die Primitive **connect()...**

a. ...erstellt einen Socket und gibt einen Deskriptor zurück.

b. ...initiieren aktiv eine Verbindung auf einem Socket.

c. ...signalisiert die Bereitschaft auf eingehende Verbindungen zu horchen.

d. ...sendet ein Datagram an einen anderen Socket.

Vervollständigen Sie den Satz:

Ein **Handshake** (Austausch von Paketen vor den eigentlichen Nutzdatenpaketen) zwischen Sender und Empfänger wird **nur bei verbindungsorientierter Kommunikation** zwingend benötigt.

Vervollständigen Sie den Satz: **Portnummern** dienen dazu...

a. ...verschiedene Anwendungen oder Prozesse als Kommunikationsendpunkt auf einem einzigen Computer eindeutig zu identifizieren.

b. ...verschiedene Switches beim Circuit-Switching eindeutig zu identifizieren.

c. ...verschiedene Rechner als Kommunikationsendpunkt innerhalb eines verteilten Systems eindeutig zu identifizieren.

d. ...verschiedene Bytes innerhalb eines "zuverlässigen Bytestroms" eindeutig zu identifizieren.

Welche Aussage ist **aus Sicht einer Applikation** richtig?

a. Bei Datagram Service und reliable Bytestream findet die Übertragung in Einheiten fester Länge, sogenannten Datagrammen statt.

b. Bei Datagram Service und reliable Bytestream findet die Übertragung in Einheiten fester Länge, statt. Diese heißen bei Datagram Service Datagramme und bei reliable Bytestream Streams.

c. Bei Datagram Service findet die Übertragung in Einheiten fester Länge, sogenannten Datagrammen statt, wobei bei reliable Bytestream ein Datenstrom übertragen wird.

d. Bei Datagram Service und reliable Bytestream wird ein Datenstrom übertragen.

Was bedeutet das **Ende-zu-Ende-Prinzip**?

a. Das Prinzip besagt, dass Daten in einem zuverlässigen Bytestrom übertragen werden sollen. Falsch

b. Das Prinzip ist eine Vermittlungstechnik, bei der zwischen Sender und Empfänger eine physikalische Leitung geschaltet wird.

c. Funktionalität, die über das Internet-Protokollhinaus geht, soll in den Endsystemen und nicht im Netzwerkkern implementiert werden.

d. Bedeutet, dass es im Netz keine Warteschlangen gibt, sondern nur in den jeweiligen End-Systemen.

e. Das Prinzip sagt mir nichts.

Welchen Befehl können Sie unter keinen Umständen verwenden, um auf einem **Datagram Socket** zu lesen?

a. listen() b. recvfrom() c. read() d. recv()

Auf einem Rechner werden zwei Dienste angeboten, die über das Netzwerk erreichbar sein sollen. Wie wird in der Regel entschieden, an **welchen der Dienste** eingehende Daten **weitergeleitet** werden?

a. Auf einem Rechner kann es grundsätzlich immer nur einen von außen erreichbaren Dienst geben

b. Anhand der Portnummer

c. Anhand der IP-Adresse

d. Anhand der MAC-Adresse

Welchen Befehl verwenden Sie in der Regel, um wie im folgenden Diagramm eine **Nachricht** auf einem **Datagram-Socket** zu **empfangen**:

a. listenfrom()

b. readfrom()

c. scanfrom()

d. recvfrom()

Stellen Sie Sich vor, Sie sind Entwickler einer Software zum Videostreaming.
Vervollständigen Sie den Satz:
Sie müssen auf **Apllikationsebene** damit rechnen, dass **nicht alle Daten** vollständig ankommen, ...

a. ... nur wenn Sie verbindungslose Datenübertragung verwenden.

b. ... sowohl wenn Sie verbindungslose als auch verbindungsorientierte Datenübertragung verwenden.

c. ... weder wenn Sie verbindungslose noch wenn Sie verbindungsorientierte Datenübertragung nutzen.

d. ... nur wenn Sie verbindungsorientierte Datenübertragung verwenden.

Die Berkeley Socket-API bietet verschiedene Arten von Sockets an. In der Vorlesung wurde vor allem auf "Zuverlässiger Byte-Strom" und "Unzuverlässiger Datagram-Service" eingegangen. Wie werden in der API diese beiden Services jeweils genannt?

Unzuverlässiger Datagram-Service **SOCK_DGRAM**

Zuverlässiger Byte-Strom **SOCK_STREAM**

BLOCK 3

Ein Client nutzt **synchrones RPC**, um eine entfernte Prozedur mit Rückgabewert aufzurufen.

- Der **Client** braucht initial **10 Millisekunden**, um seinen eigenen lokalen **Client Stub** mit den jeweiligen Parametern aufzurufen.
- Der **Server** braucht **17 Millisekunden**, um seine lokale **Prozedur** aufzurufen und das Ergebnis zu erhalten.
- Die Verarbeitungsverzögerung (**Processing Delay**) für jede Sende- oder Empfangsoperation bei Client und Server sind jeweils **2 Millisekunden**.
- Alle übrigen Verzögerungen (**Queueing Delay** (Warteschlangenverzögerung), **Transmission Delay** (Übertragungsverzögerung) und **Propagation Delay** (Ausbreitungsverzögerung)) addieren sich zwischen Client und Server in jeder Richtung auf jeweils **5 Millisekunden**.
- **Marshalling** und **Unmarshalling** benötigen jeweils **1 Millisekunden**.

Wie lange dauert ein Aufruf vom lokalen Aufruf des Client Stubs bis zur Rückgabe des Ergebnisses an die aufrufende Prozedur? Geben Sie Ihr Ergebnis in Millisekunden ohne Einheit an.

➔ **Die richtige Antwort ist: 49,00**

- Der **Client** braucht initial **6 Millisekunden**, um seinen eigenen lokalen **Client Stub** mit den jeweiligen Parametern aufzurufen.
- Der **Server** braucht **15 Millisekunden**, um seine lokale **Prozedur** aufzurufen und das Ergebnis zu erhalten.
- Die Verarbeitungsverzögerung (**Processing Delay**) für jede Sende- oder Empfangsoperation bei Client und Server sind jeweils **2 Millisekunden**.
- Alle übrigen Verzögerungen (**Queueing Delay** (Warteschlangenverzögerung), **Transmission Delay** (Übertragungsverzögerung) und **Propagation Delay** (Ausbreitungsverzögerung)) addieren sich zwischen Client und Server in jeder Richtung auf jeweils **7 Millisekunden**.
- **Marshalling** und **Unmarshalling** benötigen jeweils **2 Millisekunden**.

Wie lange dauert ein Aufruf vom lokalen Aufruf des Client Stubs bis zur Rückgabe des Ergebnisses an die aufrufende Prozedur?

➔ **Die richtige Antwort ist: 51,00**

Sie kennen Sich aus dem privaten und universitären Umfeld mit dem Konzept der “elektronischen Post” (E-Mail) aus. Nutzen sie das Beispiel des Abrufs von E-Mails durch Clients bei einem Server, um die folgende Frage für das **Client-Server-Prinzip** im Allgemeinen zu beantworten:

Welche Art von Adresse haben Client und Server im Client/Server-Modell aus der Vorlesung:

a. Nur der Client muss in der Regel eine bekannte feste von außen erreichbare IP-Adresse haben.

b. Der Server hat in der Regel immer eine dynamische IP-Adresse.

c. Nur der Server sollte in der Regel eine bekannte feste von außen erreichbare IP-Adresse haben.

d. Sowohl Client als auch Server müssen in der Regel eine bekannte feste von außen erreichbare IP-Adresse haben.

Sie kennen aus der Vorlesung verschiedene **Arten von Transparenz**. Vervollständigen Sie den Satz:

- _____ versteckt verschiedene Repräsentationen der Daten und wie auf die Resource zugegriffen wird.

- **Replikationstransparenz (Replication Transparency)** versteckt dass es mehrere Kopien derselben Ressource gibt.

Sie kaufen in einem Online-Shop ein. Sie haben bereits 50 Stück Artikel Nr. 5 im Warenkorb. Welche der folgenden Operation, mit denen sie den Shop beauftragen können, ist dabei wahrscheinlich **idempotent**?

a. Gib den Inhalt des Warenkorbs zurück.

b. Füge ein Stück des Artikels Nr. 5 hinzu.

c. Erniedrige die Menge des Artikel Nr. 5 um 3.

d. Erhöhe die Menge des Artikel Nr. 5 um 3.

Eine Bank benutzt **Remote Procedure Call (RPC)** für Konto-Transaktionen.
Welche der folgenden Operation ist **nicht idempotent**?

a. Setze den Kontostand des Kontos mit der Kontonummer 31415926 auf 100 Euro. Falsch

b. Ziehe dem Konto mit der Kontonummer 31415926 100 Euro ab.

c. Rufe den Kontostand des Kontos mit der Kontonummer 31415926 ab.

d. Sortiere die Transaktionen des Kontos mit der Kontonummer 31415926 nach dem Datum.

7. Betrachten Sie das folgende Programm. Was gibt dieses Programm unter jedem der folgenden **Parameterübergabe-Mechanismen** aus?

```c
#include <stdio.h>
void foo(int a, int b) {
    a = 3;
    a = a + b;
    b = a + 2;
}
int main() {
    int x = 5;
    foo(x, x);
    printf("%d", x);
}
```

→ Call-by-reference: 8 , Call-by-value: 5

```c
#include <stdio.h>
void bar(int a, int b) {
    a = 5;
    a = a + b;
    b = a - 2;
}
int main() {
    int x = 3;
    bar(x, x);
    printf("%d", x);
}
```

→ Call-by-reference: 8 , Call-by-value: 3

Vervollständigen Sie den Satz:

Mit der at-least-once **Semantik** wird im Kontext von **RPC** sichergestellt, dass die entfernte Prozedur **mindestens einmal** ausgeführt wird.

Sie haben in der Vorlesung den Remote **Procedure Call (RPC)** Aufruf kennengelernt. Eine Art von RPC ist folgender Grafik abgebildet. Um einen welchen Aufruf handelt es sich bei dem abgebildeten Aufruf in jedem Fall?

a. Idempotenter Remote Procedure Call

b. Synchroner Remote Procedure Call

c. Remote Procedure Call mit exactly-once Semantik

d. Asynchroner Remote Procedure Call

Block 4

Welche Aussage ist im Kontext von **P2P** (laut Vorlesung) **falsch**?

a. Knoten sind in ihrer Funktion gleichwertig. Man nennt sie deshalb auch Peers.

b. P2P-Systeme sind in der Regel selbstorganisierend.

c. Jedes P2P Netz braucht einen zentralen Server. Bei Bittorrent wird dieser Tracker genannt.

d. Knoten in einem P2P Netz verlassen oft unerwartet für andere Knoten das Netz, sind also nicht zuverlässig.

Welche der folgenden Aussagen über **P2P-Netze** ist im Allgemeinen falsch?

a. Dienste und Ressourcen können zwischen allen teilnehmenden Peers ausgetauscht werden.

b. Die Verfügbarkeit der Peers kann nicht vorausgesetzt werden, da Peers dem P2P-Netz ständig beitreten bzw. das Netz verlassen. Man nennt dies auch "Churn".

c. Zentralisierte P2P-Systeme, welche einen zentralen Server zur Verwaltung benötigen, um zu funktionieren, haben grundsätzlich keine Skalierbarkeitsprobleme.

d. Es gibt sowohl P2P-Netze mit zentralem Server als auch komplett dezentrale P2P-Netze.

Welche Aussage zum **HTTP Caching** ist falsch?

a. Durch den Header "If-modified-since" kann der Client angeben, dass er eine resource nur benötigt, wenn diese verändert wurde.

b. Durch den Header "expires" kann der Server angeben, wie lange die angefragte Resoruce gültig bleibt.

c. Durch den Header "no-cache" kann der Server angeben, dass die Resource nicht gecached werden soll.

d. Durch den Header "cache-control" kann der Client angeben, wie lange er beabsichtigt die Resource im Cache vorzuhalten.

Welche der folgenden Aussagen über **Napster** ist falsch?

a. Napster ist ein dezentrales P2P-System ohne zentralen Server.

b. Bei Napster muss nur ein Knoten kontaktiert werden, um herauszufinden, welcher Knoten eine bestimmte Datei anbietet.

c. Napster benutz einen zentralen Server, der eine Liste von Kandidaten zurückgibt, die die Datei anbieten.

d. Napster benutzt einen zentralen Server, welches die Implementierung einer anspruchsvolle Suchmaschinen auf dem Index-System vereinfacht.

Welche Aussage trifft auf das **Napster-Modell** nicht zu?

a. Napster benutzt zentralisierte Server, um zu speichern, welcher Peer welche Datei anbietet, so dass ein Abschalten dieser Server das gesamte P2P-Netz zum Erliegen bring ("Single Point of Failure").

b. Napster benutzt zentralisierte Server, um zu speichern, welcher Peer welche Datei anbietet, so dass jede Suchanfrage an die Server geschickt werden muss. Eine wesentliche Problem ist dabei die Skalierbarkeit zu gewährleisten.

c. Napster benutzt zentralisierte Server, um zu speichern, welcher Peer welche Datei anbietet, so dass eine Zensur der Inhalte an der Stelle dieser Server besonders einfach ist.

d. Bei Napster müssen Anfragen durch große Teile des P2P-Netzes geflutet werden, damit die kontaktierten Peers antworten können, falls sie eine entsprechende Datei anbieten.

Was beinheltet die Definition von "**Representational State Transfer**" nicht?

a. Bei einer Anfrage wird durch den Server eine Darstellung des Zustand einer Resource an den Client übertragen

b. RESTful webservices werden über HTTP angesprochen und geben immer XML oder HTML zurück.

c. Ein Client kann eine vorgeschlagene Darstellung einer Ressource übertragen, um den Zustand zu ändern Falsch

d. Die Darstellung von Zuständen enthält Links, mit denen der Client neue Zustandsübergänge initiieren kann

HTTP ist ein zustandloses Protokoll. Mit welcher Technik können Server z.B. trotzdem den Warenkorb einer Nutzers speichern?

a. HTTP ist nicht zustandslos **b. Cookies** c. Caches Falsch d. Canaries

Welche **HTTP-Methode** muss bei dem Beispiel jeweils verwendet werden:

Sie wollen ein Dokument abrufen, wobei die Operation idempotent sein soll **→ GET**

Sie wollen ein Dokument speichern, wobei die Operation idempotent sein soll **→ PUT**

Im Kontext von **REST** ist eine Resource.....

_____

Welche der folgenden Aussagen über **P2P-Netze** ist im Allgemeinen falsch?

a. Die Verfügbarkeit der Peers kann nicht vorausgesetzt werden, da Peers dem P2P-Netz ständig beitreten bzw. das Netz verlassen. Man nennt dies auch "Churn".

b. Zentralisierte P2P-Systeme, welche einen zentralen Server zur Verwaltung benötigen, um zu funktionieren, haben grundsätzlich keine Skalierbarkeitsprobleme. Richtig

c. Dienste und Ressourcen können zwischen allen teilnehmenden Peers ausgetauscht werden.

d. Es gibt sowohl P2P-Netze mit zentralem Server als auch komplett dezentrale P2P-Netze.

Wie funktioniert die in der Vorlesung vorgestellte **Join-Operation** in einer **DHT** mit Hilfe von **Chord**?

a.
- Alle Knoten müssen dem beitretenden Knoten bekannt sein
- Senden eines join()-Requests an den Nachfolger
- Der zukünftige Nachfolger trägt den neuen Knoten als sein Vorgänger ein
- Der zukünftige Nachfolger sendet ein notify() an den neuen Knoten, woraufhin der zukünftige Nachfolger als Nachfolger beim neuen Knoten eingetragen wird
- Aktualisierung des Nachfolger-Pointers beim Vorgängers im Zuge der regelmäßig stattfindenden stabilize() Nachrichten

b.
- Mindestens ein Knoten muss dem beitretenden Knoten bekannt sein
- Ermittlung des zukünfigen Nachfolgers per Weitterreichung des join()-Requests
- Der zukünftige Nachfolger trägt den neuen Knoten als sein Vorgänger ein
- Der zukünftige Nachfolger sendet ein notify() an den neuen Knoten, woraufhin der zukünftige Nachfolger als Nachfolger beim neuen Knoten eingetragen wird
- Aktualisierung des Nachfolger-Pointers beim Vorgängers im Zuge der regelmäßig stattfindenden stabilize() Nachrichten

c.
- Alle Knoten müssen dem beitretenden Knoten bekannt sein
- Senden eines join()-Requests an den Nachfolger
- Der zukünftige Nachfolger trägt den neuen Knoten als sein Vorgänger ein
- Der zukünftige Vorgänger wird durch notify() benachrichtigt, woraufhin er den neuen Knoten als Nachfolger einträgt

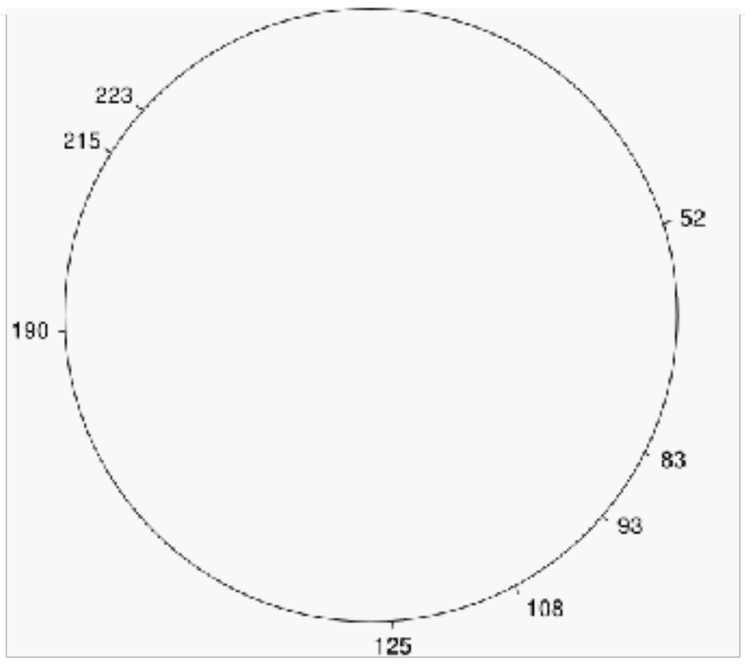
d.
- Jeder Knoten kennt seine(n) Nachfolger
- Route die Anfrage (ID, Daten) mit Hilfe der Zeiger auf Nachfolger an den für ID zuständigen Knoten
- Der zuständige Knoten antwortet dem anfragendem Knoten direkt

Was ist die allgemeine Formel zum Ausrechnen des i-ten Wertes in der Finger Table des Knotens mit der ID n bei Chord?

a. n + 2i (mod 2m) b. m + 2n (mod 2i) c. n + 2m (mod 2i) d. i + 2n (mod 2m)

Eine **Distributed Hash Table** benutzt **Chord** als Implementierung.
- Die Keys haben eine Länge von 8 Bits.
- Es sind 8 Knoten vorhanden.

Die IDs der Knoten sind in der Grafik verzeichnet:



- Welches ist der Eintrag mit **i=6** in der Finger Table des Knoten mit der **ID n=223**?

→ Die richtige Antwort ist: 52

- Welches ist der Eintrag mit **i=6** in der Finger Table des Knoten mit der **ID n=93**?

→ Die richtige Antwort ist: 190

Block 5

Was beinhaltet die Definition von "**Representational State Transfer**" **nicht**?

a. Die Darstellung von Zuständen enthält Links, mit denen der Client neue Zustandsübergänge initiieren kann Falsch

b. RESTful webservices werden über HTTP angesprochen und geben immer XML oder HTML zurück.

c. Ein Client kann eine vorgeschlagene Darstellung einer Ressource übertragen, um den Zustand zu ändern

d. Bei einer Anfrage wird durch den Server eine Darstellung des Zustand einer Resource an den Client übertragen

Ein **HTTP-Server** kann eine Verbindung nach einer Anfrage sofort beenden oder auf weitere Anfragen warten. Bitte vervollständigen Sie:

Eine **anhaltende**, **dauernde Verbindung** über **mehrere Anfragen** hinweg nennt man dabei:

a. rigoros b. kontinuierlich c. kohäsiv **d. persistent**

Eine **Verbindung**, die **nach jeder Anfrage direkt geschlossen** wird, nennt man dabei:

a. nicht-kohäsiv **b. nicht-persistent** c. lose d. instationär

Welche Bedingung muss nicht unbedingt erfüllt sein, damit ein Webservice **"Cachable"** ist, also z.B. bei einem Proxy zwischengespeichert werden kann?

a. Zustandslosigkeit (Statelessness)

b. Veränderbar (modifiable) sein

c. einheitliche Schnittstelle (uniform interface)

d. Selbstbeschreibend (Self-descriptive) sein

Was sind **Cookies** (im Sinne der Vorlesung)?

a. Informationen, die durch den Server auf einem HTTP-Client gespeichert werden und mit jeder weitern Anfrage an den Server geschickt werden

b. Ein Pufferüberlaufschutzverfahren in der Computerprogrammierung

c. Eine Technik, die einem HTTP-Benutzer anzeigt, in welcher Verzweigung er sich innerhalb einer Applikation befindet.

d. Cookies ist die Bezeichnung für die Menge idempotenter HTTP-Methoden

Ordnen Sie die **HTTP-Methoden** richtig ihren jeweiligen Beschreibungen zu:

Anfrage ein Dokument idempotent zu speichern → **PUT**

Daten liefern, die zu einem Dokument hinzugefügt werden sollen → **POST**

Im Kontext von **REST** ist eine Resource…

→ **Alles, was wichtig genug ist, um referenziert werden**

,

Block 6

Für die **Synchronisation** von 4 verteilten Systemen A, B, C und D soll der **Berkeley Algorithmus** verwendet werden. System A hat dafür einen **Time Daemon**.

Zu Beginn des Resynchronisationsintervalls haben die Uhren die folgenden Werte:
A/11542, B/11567, C/11554, D/11530.

Der Time-Daemon verwendet die **Mittelwertbildung** zur Ermittlung der Uhrzeit.
Wie groß ist der **Offset** von System A bzw. um wie viel muss er seine Uhr vor- oder zurückstellen? Geben Sie Ihr Ergebnis als positiver oder Negativer Zahlenwert ohne Einheit an!

→ **Antwort: 5,75**

Zu Beginn des **Resynchronisationsintervalls** haben die Uhren die folgenden Werte:
A/11562, B/11558, C/11566, D/11548.

Der **Time-Daemon** verwendet die **Mittelwertbildung** zur Ermittlung der Uhrzeit.
Wie groß ist der **Offset** von System A bzw. um wie viel muss er seine Uhr vor- oder zurückstellen? Geben Sie Ihr Ergebnis als positiver oder Negativer Zahlenwert ohne Einheit an!

→ **Die richtige Antwort ist: -3,50**

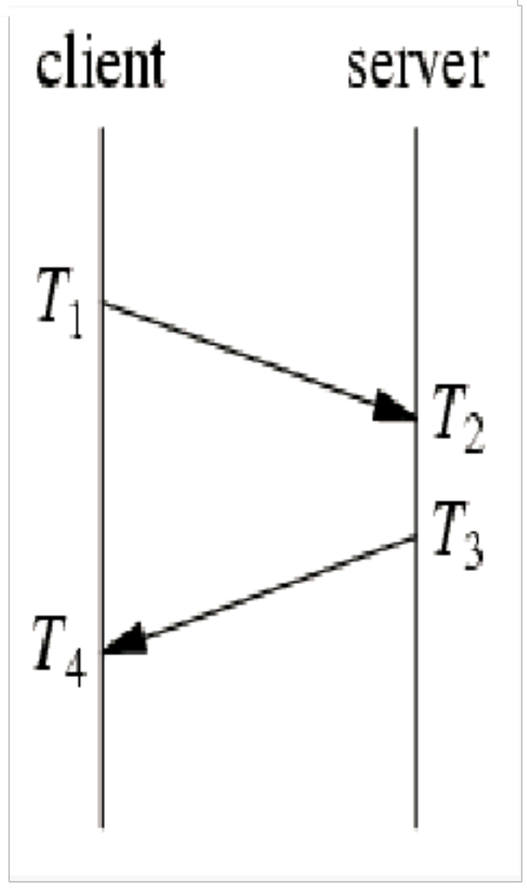
Zu Beginn des **Resynchronisationsintervalls** haben die Uhren die folgenden Werte:
A/11558, B/11544, C/11548, D/11542.

Der **Time-Daemon** verwendet die **Mittelwertbildung** zur Ermittlung der Uhrzeit.
Wie groß ist der **Offset** von System A bzw. um wie viel muss er seine Uhr vor- oder zurückstellen? Geben Sie Ihr Ergebnis als positiver oder Negativer Zahlenwert ohne Einheit an!

→ **Die richtige Antwort ist: -10,00**

NTP erhebt bei der Synchronisierung folgende **Zeitstempel**:
Nehmen Sie an T1=13,5, T2=16,1, T3=18,5 und T4=20,2.
Wie groß ist die gesamte RTT in beide Richtungen? Geben Sie Ihr Ergebnis ohne Einheit an!

→ **Antwort: 4,3**



Nehmen Sie an T1=11,3, T2=16,4, T3=19,0 und T4=23,2.
Wie groß ist die gesamte RTT in beide Richtungen?
Geben Sie Ihr Ergebnis ohne Einheit an!

→ **Antwort: 9,3**

Nehmen Sie an T1=11,3, T2=16,8, T3=18,2 und T4=23,3. Wie groß ist die gesamte RTT in beide Richtungen? Geben Sie Ihr Ergebnis ohne Einheit an!

→ **Antwort: 10,60**

Vervollständigen Sie den Satz: Ein **Stratum 1** Server ist ein Server, der...

a. ... keine eigene Zeitquelle hat und seine Zeit von Stratum 2 Servern bezieht.

b. ... direkt über eine sehr genaue Zeitquelle verfügt, etwa eine Atomuhr oder einen GPS-Empfänger.

c. ... nur Version 1 von NTP unterstützt.

d. ... eine sehr genaue Zeit anbietet, weil er mit einem Maximum an anderen Servern seine Zeit abgleicht.

Wovon können Sie in der Realität ausgehen?

a. Die Zeit, die ein Zeitserver zum Bearbeiten von Anfragen braucht, ist so klein, dass man sie immer vernachlässigen kann.

b. Uhren können immer ohne Gefahr unmittelbar auf eine frühere Zeit zurückgesetzt werden.

c. Die Round Trip Time (RTT) ist immer symmetrisch.

d. Zeiten lassen sich in einem verteilten System nie exakt synchronisieren.

Welche der folgenden Vorgehensweisen kann bei **der Zeitsynchronisation** in verteilten Systemen in der Regel zu **Problemen** führen?

a. Lokale Uhren sprunghaft zurückstellen.

b. Lokale Uhren beschleunigen.

c. Lokale Uhren sprunghaft vorstellen.

d. Lokale Uhren verlangsamen.

Wieso hat der **Offset** zwischen den Zeiten von Server und Client **keinen Einfluss** auf die Berechnung der **Round Trip Time (RTT)**?

a. Die Berechnung der RTT bezieht sich auf Client- und Server-Seite jeweils nur auf Zeitintervalle (also Differenzen), welche unabhängig von der tatsächlichen absoluten Zeit sind.

b. Der Offset zwischen den Zeiten von Server und Client hat einen Einfluss auf die Berechnung der Round Trip Time (RTT).

c. Es wird angenommen, dass die Uhren schon hinreichend genau synchronisiert sind, um den Offset zu vernachlässigen.

d. Der Offset zwischen den Zeiten von Server und Client hat genau dann keinen Einfluss auf die Berechnung der Round Trip Time (RTT), wenn diese symmetrisch ist, was bei NTP angenommen wird.

Wie funktioniert der **Berkeley-Algorithmus**?

a. Der Client fragt den Server nach der Zeit
Der Server antwortet mit seiner lokalen Zeit
Aus der Zeit zwischen Anfrage und Antwort schätzt der Client die Round Trip Time (RTT)
Der Client setzt seine Zeit auf die Zeit des Servers plus der halben geschätzten RTT

b. Der Time-Daemon schickt allen anderen Systemen seine eigene Zeit.
Die Systeme berechnen ihren Offset zu dieser Zeit und antworten dem Server
Der Server berechnet einen Mittelwert aus den Werten der Systeme und schickt allen (inkl. sich selbst) den errechneten Offset für das System

c. Der Master ist an eine genaue Uhr angeschlossen und schickt diese Uhrzeit in einem bestimmten Intervall an alle angeschlossenen Systeme.
Da die Zeitstempel in der Hardware der Geräte gesetzt werden, kann der Delay vernachlässigt werden. Die Systeme setzen ihre Uhr entsprechend der Zeit des Servers.

d. Der Client fragt den Server nach der Zeit
Der Server antwortet mit zwei Zeitstempeln für die Ankunft der Anfrage und dem Abschicken der Antwort.
Aus der Zeit zwischen Anfrage und Antwort schätzt der Client die Round Trip Time (RTT).
Der Client errechnet einen Offset zum Server, der die Bearbeitungszeit der Servers nicht enthält.
Der Client setzt seine Uhr entsprechend

Block 7

Welche Aussage über **CIDR (Classless Inter-Domain Routing)** ist Richtig?

a. CIDR ermöglicht Netzwerke-Teile von IP Adressen mit beliebig vielen Bits, so dass die festen Klassen A, b und C nicht mehr nötig sind

b. CIDR erhöht die Anzahl der bits pro IP Adresse auf 128, so dass mehr Adressen zur Verfügung stehen

c. CIDR ermöglicht verschiedenen Nutzern innerhalb eines lokalen Netzes den Zugang zum Internet

d. CIDR verteilt innerhalb eines lokalen Netzes automatisch IP-Adressen

Wie hilft **NAT** das **Problem von knappen Internet-Adressen** zu lösen?

a. NAT ermöglicht verschiedenen Nutzern innerhalb eines lokalen Netzes den Zugang zum Internet über eine einzige öffentliche IP-Adresse

b. NAT ermöglicht Netzwerke-Teile von IP-Adressen mit Beliebig vielen bits, sodass so dass die festen Klassen A, b und C nicht mehr nötig sind

c. NAT hilft bei diesem Problem nicht

d. NAT erhöht die Anzahl der bits pro IP Adresse auf 128, so dass mehr Adressen zur Verfügung stehen

Was ist der Unterschied zwischen **Hub** und **Switch**?

a. Ein Switch benutzt das Ethernet Protokoll auf jedem Link, so dass jeder Link eine eigene Kollisionsdomäne ist. Beim Hub sind alle Hosts eine Kollosionsdomäne und erkennen Kollisionen selbst.

b. Ein Switch ist eine Hub mit mehr als 2 Ports

c. Hub arbeiten in der Regel auf Layer 3 (Network layer), wohingegen Switches auf Layer (Data link layer) Routing Entscheidungen erlernen und treffen,

d. Switch und Hub unterscheiden sich bis auf den Begriff technologisch nicht

Was ist der Unterschied zwischen **Router** und **Bridge**?

a. ROUTER arbeiten in der Regel auf Layer 3 (Network layer), wohingegen BRIDGES auf Layer (Data link layer) Routing Entscheidungen erlernen und treffen.

b. Ein Router ist eine Bridge mit mehr als 2 Ports

c. Router und Bridge unterscheiden sich bis auf den Begriff technologisch nicht

d. Ein Router sendet ankommende Frames immer an alle Ausgänge, während eine Bridge diese nur an den Ausgang sendet, an dem sie benötigt werden.

Was is der Unterschied zwischen **Peering** und **Transit**?

a. Beim Peering tauschen gewöhnlich relativ gleichwertige Internet Server Providers (ISPs) den Zugang zu Ihren Kunden aus. Beim Transit bietet (meist verkauft) ein ISP den Zugang zu allen Zielen in seiner Routing-Tabelle.

b. Peering geschieht auf layer 2, wohingegen Transit auf Layer 3 staafindet.

c. Peering entscheidet anhand einer Tabelle, auf welchen Ausgang ankommende Pakete weitergeleitet werden. Diese Tabelle wird mittels Transit-Algorithmen gefüllt.

d. Es gibt keinen Unterschied

Was ist der Unterschied zwischen **Routing** und **Forwarding**?

a. Forwarding entscheidet anhand einer Tabelle, auf welchen Ausgang ankommende Pakete weitergeleitet werden. Diese Tablle wird mittels Routing-Algorithmen gefüllt.

b. Routing passiert zwischen großen Autonomen Systemen (AS) und Forwarding zwischen den ISPs und den Endkunden.

c. Es gibt keinen Unterschied.

d. Routing geschieht auf Layer 2, wohingegen Forwarding auf Layer 3 stattfindet.

Welche für einen Host erlaubte IP befindet sich **im Netz 192.168.1.0** mit der **Netzmaske 255.255.255.0**?

a. 192.168.1.5

b. 192.168.1.255

c. keiner der oben genannten

d. 192.168.2.5

Sie haben soeben Adresse per **DHCP** bezogen. **Wie lange** ist sie **gültig**?

a. bis der sogenannte DHPC-Lease abgelaufen ist.

b. bis Time to Live (TTL) abgelagen ist.

c. Gründsätzlich 24 Stunden

d. bis die Hälfte des sogenannte DHPC-Lease abgelaufen ist.

Sie haben soeben Adresse per **DHCP** bezogen. **Wann** wird diese in der Regel **erneut**?

a. WENN die Hälfte des sogenannte DHPC-Lease abgelaufen ist.

b. bis Time to Live (TTL) abgelagen ist.

c. Gründsätzlich 24 Stunden

d. bis der sogenannte DHPC-Lease abgelaufen ist.

Wofür wird das **ARP-Protokoll** verwendet?

a. Es löst IP-Adressen in MAC-Adressen auf.

b. Es verhindert Kollisionen bei Ethernet

c. Es löst Hostnamen in IP-Adressen auf

d. Es erstellt auf Knoten im Internet eine Forwarding-Tabelle mit IP-Adressen und dem jeweiligen Ausgang.

Was ist eine Eigenschaft der **MAC-Adresse**?

a. Sie sollte weltweit einzigartig sein

b. Sie sollte ausschließlich durch Software gesetzt werden

c. Sie sollte mit möglichst vielen Nullen beginnen

d. Sie sollte aus der IP-Adresse generiert werden

Vollständigen:

Bei IP Ist die Adresse **127.0.0.1** für den **Loopback** reserviert

und die Adresse **255.255.255.255** für den **Broadcast**

Block 8:

Womit muss man bei einem Netz, welches **Best-Effort** anbietet, nicht rechnen?

a. Es ist mit allen hier genannten Effekten zu rechnen. Richtig

b. Pakete umgeordnet ankommen

c. Pakete können beschädigt werden

d. Pakete können verloren gehen

2. Wie interpretiert **TCP Paketverluste**?

a. TCP sendet verlorene Pakete nur nochmals, aber bezieht Paketverluste ansonsten nicht mit ein.

b. Als Überlast (ein Router auf dem Weg hat das Paket wegen voller Warteschlangen verworfen)

c. Als Verlust durch Kollision auf dem Medium (kommt bei drahtlosen Netzen häufig vor)

d. Als Aufforderung, die Verbindung zu schließen.

3. Wenn eine **TCP-Verbindung** einseitig geschlossen wird, darf der andere Verbindungspartner keine Daten mehr schicken. Eine auswählen:

Wahr

Falsch

5. Wie groß muss bei Go-Back-N der Puffer auf **Empfängerseite** mindestens sein?

a. So groß wie das Window + 1.

b. So groß wie das Window.

c. 0 bzw. 1 (nur das aktuelle Paket)

d. So groß wie das Window - 1.

Wie groß muss bei Go-Back-N der Puffer auf **Senderseite** mindestens sein?

a. 0

b. So groß wie die Hälfte des Windows.

c. So groß wie das Window.

d. 1

Was ist der Unterschied zwischen **Flow- und Congestion-Control**?

a. Flow-Control dient dazu, den Empfänger nicht zu überlasten, wohingegen Congestion-Control versucht eine Überbelastung im Netz zu verhindern.

b. Flow-Control dient dazu, das Netz nicht zu überlasten, wohingegen Congestion-Control versucht eine Überbelastung des Empfängers zu verhindern.

c. Flow-Control und Congestion-Control dienen gemeinsam dazu, alle Daten vollständig zu empfangen.

d. Flow-Control dient dazu, alle Daten in der richtigen Reihenfolge zu empfangen, wohingegen Congestion-Control versucht eine Überbelastung des Empfängers zu verhindern

Mit Hilfe welcher **zwei Mechanismen** erkennt ein Sender bzw. Empfänger bei Verwendung eines **ARQ Protokolls Fehler**?

a. Mit Timern und Sequenznummern.

b. Immer ausschließlich mit NACKs (negative acknowledgment) und Sequenznummern.

c. Mit Ports und IP-Adressen.

d. Mit Handschake und Connection close.

Angenommen es wird **TCP** genutzt, der Sequenznummernraum entspricht deshalb den übertragenen Bytes und der Empfänger hat gerade Byte 4711 empfangen, dann schreibt Sequenznummer der Empfänger in sein ACK-Paket die Sequenznummer 4711.

Wahr

Falsch

Welche Pakete werden beim TCP-Handshake ausgetauscht?

a. FIN, ACK **b. SYN, SYN + ACK, ACK** c. FIN, (Data) + ACK, FIN, ACK d. SYN, ACK