Ludwig-Maximilians-Universität und Technische Universität München
Prof. Dr. Heinz-Gerd Hegering

Übungen zur Vorlesung Rechnernetze I, WS 2007/2008

Übungsblatt 10

Besprechung in der Übung am 28./30. Januar 2008.

Aufgabe 10.1 Adressierung von entfernten Anwendungen

1. Welche drei Informationsteile benötigt eine Anwendung (application), um eine andere Anwendung, die auf einem entfernten Rechner (host) läuft, über eines von mehreren Transportprotokollen zu adressieren?
2. Wie heißen diese Informationsteile bei den in der Internet-Welt verwendeten Protokollen? Wie nennt sich die Schnittstelle zwischen den Anwendungen und dem API (application programming interface), das das Betriebssystem als Verbindung zum Netzwerk-Protokoll-Stack auf einem Host bereitstellt?
3. Optional für Fortgeschritte: Informieren Sie sich, wie dies bei systemnaher Programmierung umgesetzt wird. Am besten eignet sich dafür ein Beispiel in der Programmiersprache C, weil höhere Programmiersprachen (z.B. Python, Java, Perl) viele Details bereits kapseln und so vor dem Programmierer verbergen.

Aufgabe 10.2 Nicht-persistente und persistente Verbindungen bei HTTP

Die Nutzung von Diensten auf entfernten Systemen ist eine der Hauptanwendungen in verteilten Systemen. Die Verbindung zu diesen Diensten kann dabei auf unterschiedliche Arten erfolgen, welche im folgenden am Beispiel des Hypertext Transfer Protokolls (HTTP) untersucht werden sollen.

1. HTTP erlaubt sowohl nicht persistente, als auch persistente Verbindungen. Erklären Sie kurz diese beiden Verfahren! Beschreiben Sie ebenfalls das Konzept des Pipelinings bei persistenten Verbindungen.
2. Geben Sie die Definition der Round Trip Time (RTT) (nach Kurose) an!
3. Anhand der Übertragungszeiten einer HTML-Seite, für deren Anzeige entweder keine, 9 oder 99 weitere Objekte nachgeladen werden, sollen nun die Unterschiede zwischen den verschiedenen Verbindungsmöglichkeiten aufgezeigt werden. Es sollen folgende Bedingungen gelten:

   Hinweis: Die Annahmen hier sind eine extreme Vereinfachung, die nur fuer kleine Seiten und ohne Komplikationen gilt.

   - Ein Verbindungsaufbau dauert 1 RTT.
   - Es dauert jeweils eine halbe RTT, bis eine Anfrage den Server erreicht bzw. bis eine Antwort/Übertragung den Client erreicht. Die eigentliche Übertragungszeit der Daten ist dabei zu vernachlässigen.
   - Weitere Verzögerungen im Client/Server (z.B. verursacht durch andere Prozesse) oder in Transitsystemen werden nicht berücksichtigt.
   - Hinweis: Die Annahmen hier sind eine extreme Vereinfachung, die nur für kleine Seiten und ohne Komplikationen gilt.

   (a) Vervollständigen Sie die unten angegebene Tabelle, welche die einzelnen Verfahren gegenüberstellt.

   Vorgehen: Visualisieren Sie zunächst für alle fünf Verfahren grafisch die Kommunikation durch zwei senkrechte Zeitachsen für Client und Server mit Pfeilen für den Datenaustausch. Zeichnen Sie die Zeitachsen nebeneinander für parallele Verbindungen. Beschriften Sie die Zeichnungen sinnvoll.

   Leiten Sie dann aus der grafischen Darstellung die Ergebnisse ab.

   Wichtig: Beim Pipelining soll folgende Vereinfachung gemacht werden: Es überlagern sich ab der 2. Anfrage des Clienten zeitlich die Anfrage (Richtung: von Client zu Server) und die Antwort des Servers auf die vorherige Anfrage (Rückrichtung).

Verbindungen \ Übertragung von	HTML	HTML + 9 Obj.	HTML + 99 Obj.
1 Verbindung, nicht persistent	2	20	200
max. 10 Verbindungen parallel, nicht pers.	2		
1 persistente Verb.	2		
1 persistente Verb. mit Pipelining	2		
3 persistente Verb. mit Pipelining	2		

(b) Warum ist es sinnvoll, Techniken wie persistente Verbindungen und Pipelining im HTT-Protokoll zur Verfügung zu stellen und nicht (nur) auf parallele Verbindungen zu setzen?

Aufgabe 10.3 Praktische Versuche: HTTP im Protokoll-Analysator Ethereal

Ein Netzwerk-Protokoll-Analysator visualisiert die Vorgänge beim Senden und Empfangen von Daten über das Netz. Damit leistet er bei der Ausbildung und bei der Fehlersuche wertvolle Dienste.

Benutzen Sie das freie Programm Ethereal zur Protokoll-Analyse aus einer Trace-Datei. Sie finden das Programm auf `http://www.ethereal.com`, falls es noch nicht auf Ihrem System installiert ist. Es ist für viele Plattformen erhältlich, u.a. Linux, Windows, MacOS X. Auf vielen Linux-Systemen ist es Teil der Standard-Installation, Sie starten es von der Shell aus mit `ethereal`.

Auf der Webseite zur Übung finden Sie die Trace-Datei zu dieser Aufgabe. Trace-Dateien enthalten quasi 1:1-Kopien von Paketen, so wie sie im Netzwerk verschickt/empfangen werden.

Im vorliegenden Fall wurden nacheinander mit einigen Sekunden Abstand die Webseiten `http://www.tum.de`, `http://www.lmu.de` und `http://www.google.de` mit einem grafischen Browser abgerufen, welcher dann die zusätzlich zur Anzeige benötigten Dateien, wie Style-Dateien und Bilder, nachlädt. Die Trace-Datei wurde so gefiltert, dass nur solche Pakete drin geblieben sind, die unmittelbar HTTP-Inhalte transportieren.

Laden Sie die Trace-Datei `tum-lmu-google.capture` in Ethereal. Machen Sie sich dann zunächst mit dem Programm vertraut, insbesondere mit dem Menü. Nutzen Sie die Conversation List, um alle TCP-Verbindungen aufzulisten. In deren Einträgen können Sie sich durch Rechtsklick auf die Einträge die passenden Pakete anzeigen lassen. Auch durch Rechtsklick auf die einzelnen Pakete öffnet sich ein Kontext-Menü.

Ziel der Aufgabe ist die Erstellung von Übersichten nach folgendem Beispiel (Quelldaten sind hier die Pakete zu www.google.de am Ende der Trace-Datei):

```
Google, www.google.de

Anfangszeit des gesamten Abrufs: 29.031 s
Endzeit des gesamten Abrufs: 29.292 s
Zeitdifferenz: 261 ms

TCP-Verbindung 1: Client 192.168.218.94:1286 <-> Server 64.233.183.104:80
Server-String aus HTTP-Header: GWS/2.1

Dateiname       Status- Content-Type    Länge   Start   Ende    Dauer
                Code                    [Byte]  [s]     [s]     [ms]
-----------------------------------------------------------------------
/               200     text/html       1332    29.032  29.106  74
favicon.ico     200     image/x-icon    1406    29.241  29.292  51

TCP-Verbindung 2: Client 192.168.218.94:1285 <-> Server 64.233.183.104:80
Server-String aus HTTP-Header: GWS/2.1

Dateiname       Status- Content-Type    Länge   Start   Ende    Dauer
                Code                    [Byte]  [s]     [s]     [ms]
```

```
-----------------------------------------------------------------------
logo.gif        200     image/gif       9121    29.138  29.182  44

Verfahren: Es wurden persistente Verbindungen genutzt, Pipelining kam nicht zur
           Anwendung. Der Browser nutzte 2 parallele TCP-Verbindungen.
```

Hinweis: Versuchen Sie, das Beispiel für Google detailliert nachzuvollziehen, bevor sie die Daten zu www.tum.de und www.lmu.de auswerten.

Aufgabe:

1. Erstellen Sie zwei Übersichten (eine für www.tum.de, eine für www.lmu.de jeweils inkl. aller Objekte, die vom Browser zur Anzeige der Seite nachgeladen werden), nach dem Beispiel für Google.

   Jede der Übersichten soll folgende Daten enthalten:

   - Überschrift (einmal TUM, www.tum.de, einmal LMU, www.lmu.de)
   - Zeitstempel des Pakets mit der HTTP-Anfrage für die HTML-Seite in Sekunden
   - Zeitstempel des Pakets, mit dem das Nachladen des letzten Objektes abgeschlossen wird, in Sekunden
   - Zeitdifferenz (Ende-Anfang) in Millisekunden
   - Für jede TCP-Verbindung:
     - Quell-IP-Adresse:Quell-Port – Ziel-IP-Adresse:Ziel-Port. Ordnen Sie die Rollen Client und Server zu.

       Hinweise: Gehen Sie davon aus, dass alle Verbindungen vom Client aufgebaut werden.
     - Server-String aus den Kopfdaten des HTTP-Headers
     - Für jede TCP-Verbindung: Identifizieren Sie alle über diese Verbindung geladenen Dateien. Erstellen sie dazu für jede TCP-Verbindung eine Tabelle mit den Spalten: Dateiname (URL-Pfad kann weggelassen werden), HTTP-Status-Code, Content-Type, Dateilänge in Byte, Zeitpunkt der HTTP-Anfrage, Zeitpunkt der HTTP-Antwort, Zeitdifferenz. Der Zeitpunkt einer HTTP-Anfrage soll hier durch den Zeitstempel (Absolutwert) des Pakets festgelegt sein, das das 1. Byte der Anfrage überträgt, für die Antwort das Paket, das das letzte Byte transportiert.

       Markieren Sie zusätzlich die Zeitpunkte der Anfragen und Antworten auf einem Zeitstrahl. Verbinden Sie zusammengehörige Paare (Anfrage-Zeitpunkt, Antwort-Zeitpunkt) mit einem Bogen.

       Hinweis: Um die übertragenen Dateien zu identifizieren, reicht es nicht immer, nur die Kurzinfo anzuschauen, die Ihnen Ethereal in der Paket-Liste angibt. Nutzen Sie stattdessen Rechtsklick auf ein Paket, "Follow TCP Stream". Um wieder zur Paketliste aller Pakete zu kommen, löschen Sie den Filterausdruck im Display-Filter-Eingabefeld.
   - Identifizieren Sie das verwendete Verfahren (nicht-persistente Verbindungen, persistente Verbindungen, Pipelining ja/nein)

   Hinweis: Nutzen Sie in Ethereal die Displayfilter, sowie die Analyse-Funktionen im Menü! Das Einfärben (Coloring Rules) von zusammengehörigen Paketen erleichtert es, den Überblick zu behalten.

   Hinweis: IP-Adressen und ihre Zuordnung zu Hostnamen werden später in der Vorlesung behandelt, dehalb gibt es hier eine Hilfestellung: Für die in der Trace-Datei enthaltenen Pakete gilt: TU-Adressen beginnen mit 129, LMU-Adressen beginnen mit 141, Google-Adressen beginnen mit 64. Die Adresse 192.168.218.94 gehört zu dem Rechner, von dem die drei Seiten abgerufen wurden und auf dem auch der Protokoll-Analysator lief.

2. In der Trace-Datei finden sich folgende Besonderheiten:

   - Einmal wird die HTTP-Antwort komprimiert übertragen.
   - Einmal passt der Content-Type nicht zu den vom Server gesendeten Nutzdaten.
   - Einmal findet der HTTP-Server die gewünschte Datei nicht.

   Identifizieren Sie diese drei Stellen. Äußern Sie in allen drei Fällen eine Vermutung über die Ursachen.

Vertiefung (optional, ohne Punkte)

Benutzen Sie Ethereal im Live-Capture-Betrieb auf einem Rechner, auf dem Sie die administrative Hoheit haben und der nicht gleichzeitig von anderen Nutzern genutzt wird, möglichst in einem Netzwerk, in dem durch andere Rechner nur wenig Netztraffic entsteht. Hinweis: Sie benötigen meist root-Rechte oder müssen als Mitglied der Gruppe Administratoren eingeloggt sein, um den Live-Capture-Betrieb zu starten.

Beobachten Sie den HTTP-Verkehr, während Sie mit einem Browser im Web surfen und versuchen Sie, das vermittelte Wissen über das HTT-Protokoll aus der Vorlesung praktisch nachzuvollziehen. Experimentieren Sie mit den Einstellungen in ihrem Browser-Programm bezüglich persistenter Verbindungen und Pipelining.