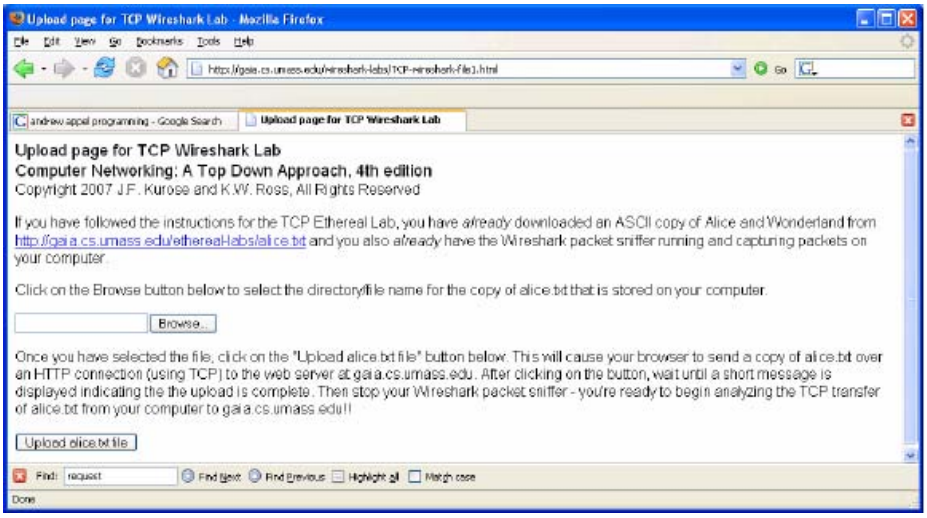
Il protocollo TCP

Obiettivo

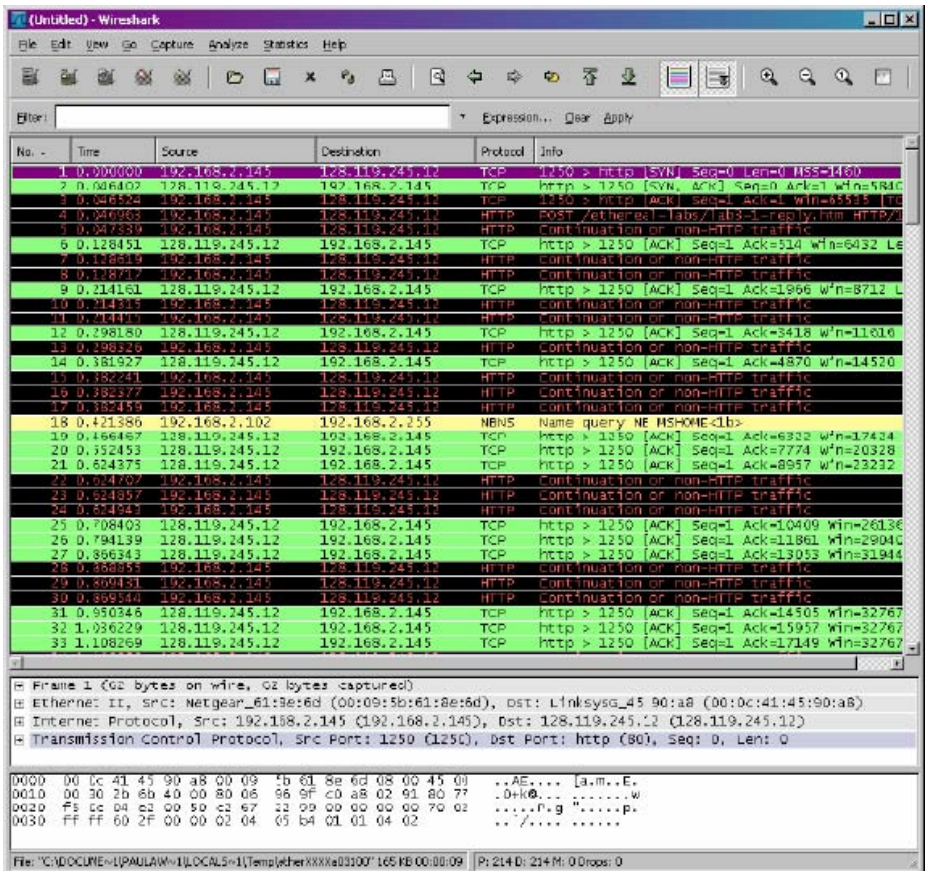
In questo esercizio studieremo il funzionamento del protocollo TCP. In particolare analizzeremo la "traccia" di segmenti TCP scambiati tra il vostro calcolatore ed un server remoto. Prima di iniziare la nostra esercitazione su TCP, useremo Wireshark per ottenere la "traccia" del trasferimento di un file dal vostro computer ad un server remoto utilizzando il protocollo TCP.

Procedura

- Aprite il web browser. Andate all'indirizzo http://gaia.cs.umass.edu/wireshark-labs/alice.txt e recuperate il file di testo *Alice in Wonderland*. Memorizzate questo file sul vostro hard disk.
- Andate su http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.
- Vi dovrebbe apparire una schermata di questo tipo:



- Cliccate su *Browse* per inserire il nome del file che avete memorizzato sul vostro computer contenente il testo di *Alice in Wonderland*. Non cliccate ancora su *Upload alice.txt file*.
- Adesso avviate Wireshark e iniziate ad ascoltare il traffico di rete.
- Ritornate al vostro web browser e cliccate su *Upload alice.txt file* per caricare il file sul server *gaia.cs.umass.edu.server*. Una volta caricato il file vi verrà visualizzato un breve messaggio di congratulazioni.
- A questo punto terminate la sessione di cattura di pacchetti su Wireshark. La vostra finestra di Wireshark dovrebbe essere simile a quella riportata di seguito:



Se non vi è possibile catturare questi pacchetti con Wireshark allora potete scaricare un file di cattura già confezionato per voi da questo URL : https://www.mat.unical.it/informatica/Reti%20di%20Calcolatori?action=AttachFile&do=view&target=tcp-trace
Memorizzate questo file sul vostro computer e apritelo da Wireshark attraverso il comando *Open* del menu *File*.

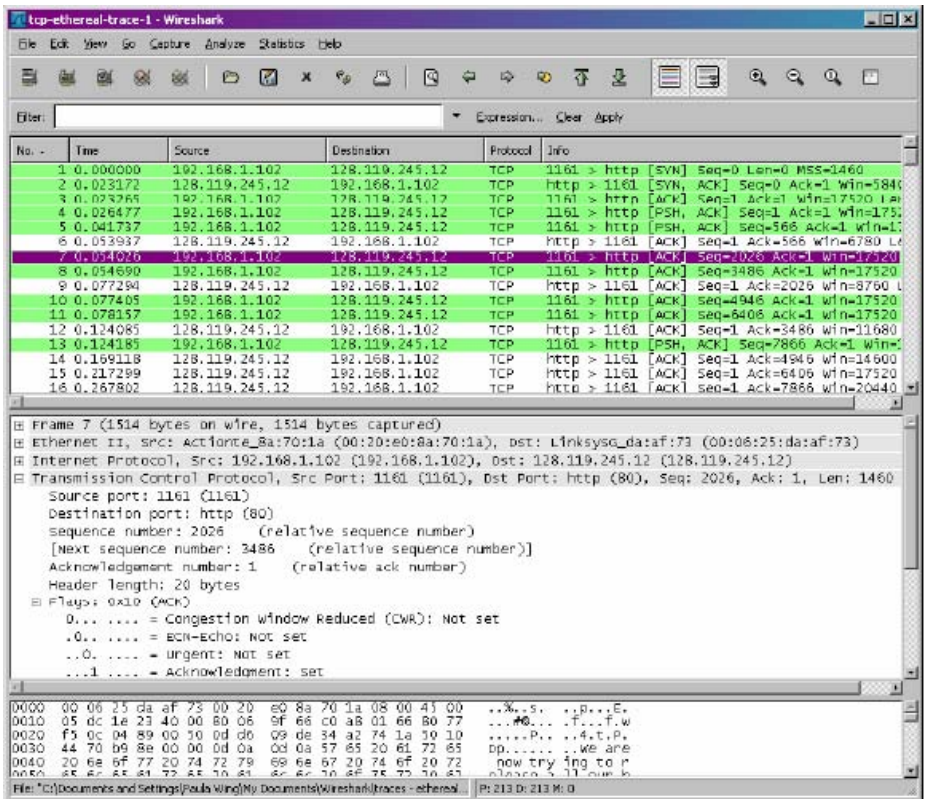
Domande

Filtrate i pacchetti visualizzati nella finestra di Wireshark inserendo la parola “tcp” nella casella testuale per la specifica del filtro. Ciò che dovreste vedere è una serie di messaggi TCP e HTTP scambiati tra il vostro computer ed il server *gaia.cs.umass.edu*.

- Quali sono l’indirizzo IP ed il numero di porta TCP utilizzati dal computer (sorgente) che sta trasferendo il file a *gaia.cs.umass.edu*?
- Qual è l’indirizzo IP di *gaia.cs.umass.edu*? Su quale numero di porta sta inviando e ricevendo segmenti TCP per questa connessione?

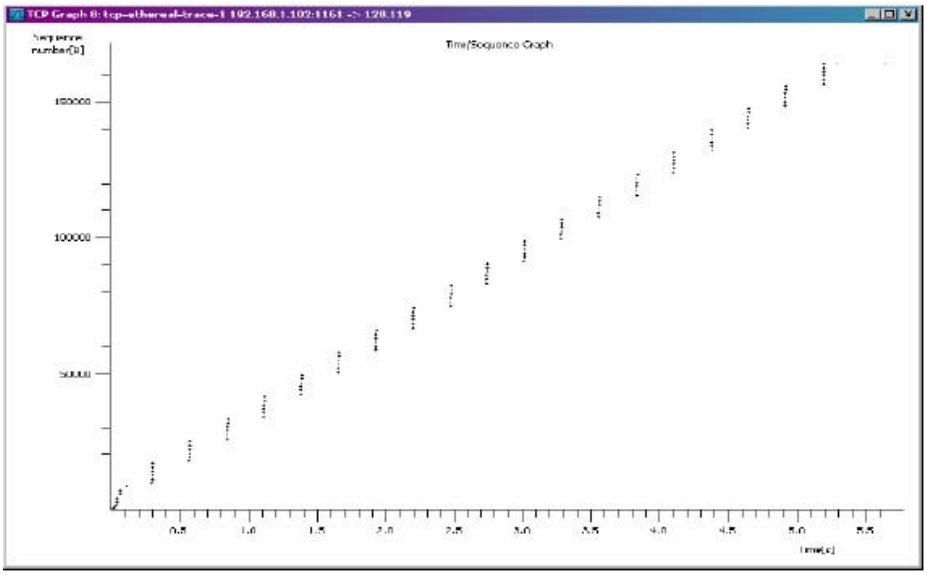
Se siete stati in grado di creare il vostro file di cattura, senza scaricarlo dal sito del corso, rispondete a questa domanda:

- Quali sono l’indirizzo IP ed il numero di porta TCP utilizzati dal vostro computer (sorgente) per trasferire il file a *gaia.cs.umass.edu*? Qual è il primo segmento che indica una conversazione tra gaia.cs.umass.edu e il pc locale?



- Qual è il numero di sequenza del segmento TCP SYN che è stato usato per iniziare la connessione TCP tra il computer client e *gaia.cs.umass.edu*? Cos'è che identifica il segmento come segmento SYN?
- Qual è il numero di sequenza del segmento SYN/ACK inviato da *gaia.cs.umass.edu* al computer client in risposta al SYN? Qual è il valore del campo ACKNOWLEDGEMENT NUMBER nel segmento SYN/ACK? Come ha determinato *gaia.cs.umass.edu* quel valore? Cos'è che identifica il segmento come segmento SYN/ACK?
- Qual è il numero di sequenza del segmento TCP contenente il comando HTTP POST?
- Considerate il segmento TCP contenente il comando HTTP POST come primo segmento della connessione TCP. Quali sono i numeri di sequenza dei primi sei segmenti della connessione TCP (incluso il segmento contenente il comando HTTP POST)? In che istante è stato inviato ciascun pacchetto? Quando è stato ricevuto l'ACK per ciascun segmento? Data la differenza tra quando ciascun segmento TCP è stato inviato e quando il suo ACK è stato ricevuto, qual è il valore di RTT per ciascun dei sei segmenti? Qual è il valore di *EstimatedRTT* dopo la ricezione di ciascun ACK? Il valore di *EstimatedRTT* per il primo segmento è uguale alla misura di RTT, e poi per i segmenti successivi viene calcolato usando l'equazione vista a lezione.
- Qual è la lunghezza di ciascuno dei primi sei segmenti TCP?
- Ci sono dei segmenti ritrasmessi nel file di cattura? Cosa devi verificare per poter rispondere a questa domanda?
- Tipicamente quanti segmenti vengono confermati con un solo pacchetto che trasporta un ACK? Potete identificare i casi dove il ricevitore sta rispondendo con degli ACK a più di uno dei segmenti ricevuti?
- Qual è il throughput (byte trasferiti per unità di tempo) per la connessione TCP? Spiegate come calcolare questo valore.

Adesso esaminiamo la quantità di dati inviati per unità di tempo dal client al server. Per farlo useremo una delle utilità grafiche di Wireshark per il protocollo TCP – il *Time SequenceGraph (Stevens)* – per rappresentare i nostri dati in maniera grafica. Selezionate un segmento TCP e poi cliccate su *Statistics →TCP Stream Graph → Time SequenceGraph(Stevens)*. Dovreste vedere un grafico simile a quello seguente:



Qui ciascun punto rappresenta un segmento TCP inviato, ed è ottenuto incrociando il suo numero di sequenza con l'istante di tempo al quale è stato inviato.

- Usate il tool *TimeSequenceGraph(Stevens)* per visualizzare il grafico (tempo, numeri di sequenza) dei segmenti che sono stati inviati dal client a *gaia.cs.umass.edu*. Potete identificare dove inizia e dove finisce la fase di slowstart di TCP? E dove termina la congestion avoidance? Commentate i modi in cui i dati misurati differiscono dal comportamento ideale di TCP che abbiamo studiato a lezione.
- Rispondete a ciascuno dei due quesiti precedenti per il file di cattura che avete generato quando avete trasferito un file dal vostro computer a *gaia.cs.umass.edu*.

Il protocollo UDP

Durante questa esercitazione vedremo anche il protocollo di trasporto UDP. Iniziate una sessione di cattura di pacchetti in Wireshark e poi fate qualcosa che causerà l'invio e la ricezione di pacchetti UDP da parte del vostro host.

Opzionalmente, potete confezionare un piccolo software di chat basato sul protocollo UDP.

Esercizio Opzionale:
Si progetti un piccolo software di chat basato sul protocollo UDP: il software deve consentire una chat tra due utenti i quali conoscono mutuamente il proprio IP e la propria porta di ascolto. Si possono usare questi sorgenti di partenza:
http://systembash.com/content/a-simple-java-udp-server-and-udp-client/

Si noti che il protocollo di chat non deve prevedere la semplice modalità "botta e risposta" ma entrambi gli interlocutori devono poter inviare del testo in qualsiasi momento dall'altra estremità della conversazione. Ricorrere alla programmazione multithread se necessario.

Che cosa potreste fare altrimenti per catturare del traffico UDP? Se non riuscite a catturare dei pacchetti con Wireshark allora potete scaricare un file di cattura già confezionato per voi dall'indirizzo

https://www.mat.unical.it/informatica/Reti%20di%20Calcolatori?action=AttachFile&do=view&target=udp-trace.pcap

In tal caso memorizzate il file sul vostro computer ed apritelo con Wireshark.
Dopo avere terminato la sessione di cattura, impostate il filtro in modo che Wireshark vi mostri solo pacchetti UDP inviati e ricevuti dal vostro host. Scegliete uno di questi pacchetti ed espandete i campi UDP nella finestra di dettaglio.

Domande:

- Selezionate un pacchetto. Da questo pacchetto determinate quanti campi ci sono nell'intestazione(*header*) UDP. Individuate anche i nomi di questi campi.
- Dal campo *content* del pacchetto, determinate la lunghezza (in byte) di ciascuno dei campi dell'intestazione UDP.
- Il valore del campo *Length* indica la lunghezza di cosa? Verificatelo nel pacchetto UDP che avete catturato.
- Qual è il numero massimo di byte che può essere incluso in un *payload* UDP?
- Qual è il numero di porta più grande possibile?
- Qual è il numero di protocollo per UDP? Date le vostre risposte sia nella notazione esadecimale che in quella decimale.
- Su quali campi viene determinato il checksum UDP? Reperite questa informazione tramite un motore di ricerca.

- Esaminate una coppia di pacchetti UDP in cui il primo pacchetto è inviato dal vostro host mentre il secondo pacchetto è una risposta al primo. Descrivete la relazione che c'è tra i numeri di porta nei due pacchetti.

Domanda extra:

- Catturate un piccolo pacchetto UDP. Verificate a mano il checksum di questo pacchetto. Mostrate il procedimento ed i passi che avete seguito per svolgere questo compito.

Il protocollo IP

Obiettivo

Nell'ultima parte dell' esercitazione studieremo anche il funzionamento del protocollo IP. In particolare analizzeremo i pacchetti IP generati durante una esecuzione del comando tracert (traceroute con Linux). Ricordiamo che questo comando ha un comportamento differente tra linux e windows. Con Linux, traceroute invia all'host di destinazione una serie di pacchetti UDP; con Windows, il comando tracert invia messaggi ICMP. Questa distinzione è necessaria per comprendere le differenze tra i pacchetti rilevati usando due sistemi operativi appartenenti a famiglie differenti. Una similitudine nel funzionamento è comunque presente. Entrambi i comandi (tracert e traceroute) inviano pacchetti con un valore del campo TTL che varia da pacchetto a pacchetto. In generale la sequenza di pacchetti che
viene inviata sulla rete ha il campo TTL modificato secondo il seguente schema:

- Il primo messaggio ha TTL=1;
- Il secondo ha TTL=2, e così via. In questo processo, ogni router che viene attraversato decrementa di uno il valore del campo TTL fino ad arrivare a TTL=1. Quando a un router perviene un messaggio con TTL=1, il router risponde al mittente inviando un messaggio ICMP di errore (Time To Live exceeded). Con questa procedura, il sistema è capace di mappare i passi successivi
che un pacchetto segue durante il suo percorso sulla rete per raggiungere la destinazione.

Procedura Windows

- Aprite la console dei comandi di Windows;
- Fate partire Wireshark e iniziate ad ascoltare il traffico di rete.
- Dalla console dei comandi di Windows, eseguite il seguente programma "tracert hostname";
- Quando il comando tracert finisce la sua esecuzione, interrompere lo sniffing della rete.

Dalla schermata di Wireshark è possibile individuare una serie di pacchetti ICMP di tipo "Echo Request" generati dalla vostra macchina. Inoltre, è possibile individuare una serie di pacchetti ICMP di tipo TTLexcedeed pervenuti alla vostra macchina dai vari router che i messaggi ICMP hanno attraversato.

Domande Windows:

- Individuare il primo messaggio ICMP inviato dalla vostra macchina ed espan dere il protocollo IP corrispondente al pacchetto inviato. Qual'è l'indirizzo IP della vostra macchina?
- Da quanti byte è costituito l'header del pacchetto IP che state analizzando? Da quanti byte è composto il payload del pacchetto IP? Fornire una spiegazione della procedura per il calcolo del payload del pacchetto;
- Il pacchetto IP in analisi è frammentato? Come fare a stabilire se il pacchetto è frammentato?

Ordinare i pacchetti rilevati dalla rete secondo il mittente. Identificare i pacchetti inviati dalla vostra macchina con protocollo ICMP e rispondere alle seguenti domande:

- Quali sono i campi del pacchetto IP che cambiano tra un messaggio e l'altro ?
- Quali sono i campi che restano costanti?
- Quali sono i campi che devono restare costanti e quali quelli che ci si aspetta che debbano cambiare? Perché?
- Descrivi e giustifica i valori che vengono visualizzati nel campo Identification del pacchetto IP;

- Che valori assumono i campi Identification e TTL di questi pacchetti IP?
- I valori di questi campi restano invariati per tutti i pacchetti inviati dal router più vicino (first hop)? Perché?

• Identificare i messaggi ICMP di tipo TTL-exceeded che sono pervenuti alla vostra macchina. Che valori assumono i campi Identification e TTL di questi pacchetti IP?
• I valori di questi campi restano invariati per tutti i pacchetti inviati dal router più vicino (first hop)? Perché?