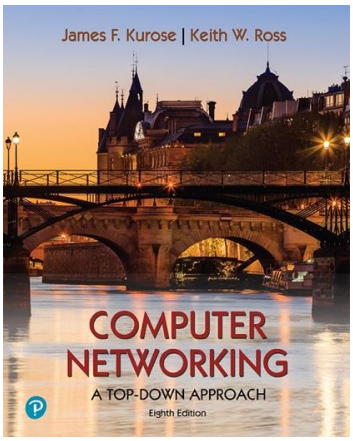
Wireshark Lab: TCP v8.0

Supplement to *Computer Networking: A Top-Down Approach, 8th ed.*, J.F. Kurose and K.W. Ross

"Tell me and I forget. Show me and I remember. Involve me and I understand." Chinese proverb



Adattato per il corso di Sistemi Operativi e Reti – Corso di laurea in Informatica – Università della Calabria.

In questa sperimentazione di laboratorio esamineremo in dettaglio il comportamento del celebre protocollo TCP. Lo faremo analizzando una traccia dei segmenti TCP inviati e ricevuti durante il trasferimento di un file da 150 KB (contenente il testo delle avventure di Alice nel paese delle meraviglie di Lewis Carrol) dal tuo computer a un server remoto. Studieremo l'uso da parte di TCP dei numeri di sequenza e di riconoscimento per fornire un trasferimento dati affidabile; vedremo l'algoritmo di controllo della congestione di TCP - avvio lento ed eliminazione della congestione - in azione; ed esamineremo il meccanismo di controllo del flusso pubblicizzato dal ricevitore di TCP. Considereremo anche brevemente la configurazione della connessione TCP e analizzeremo le prestazioni (velocità effettiva e tempo di andata e ritorno) della connessione TCP tra il tuo computer e il server.

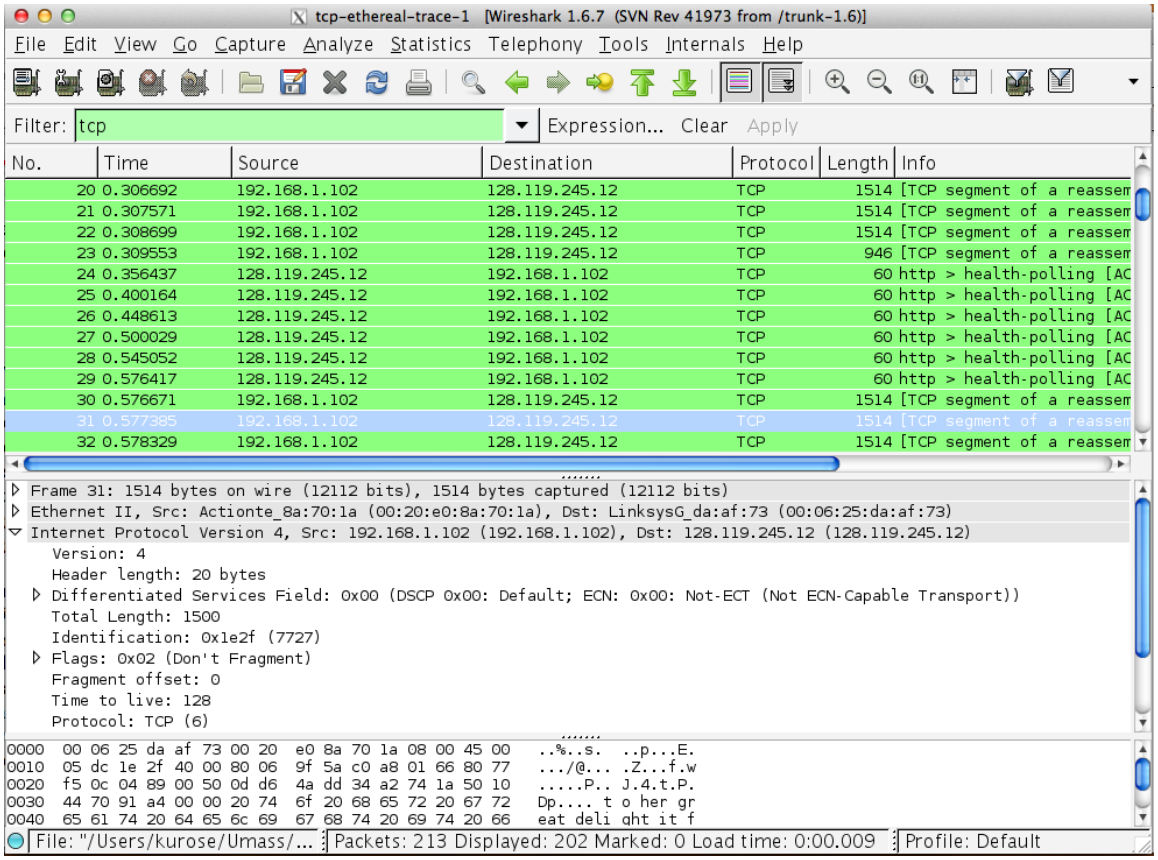
1. Catturare una conversazione TCP dal tuo calcolatore a un server remoto

Prima di iniziare la nostra esplorazione del TCP, dovremo utilizzare Wireshark per ottenere una cattura del trasferimento via TCP di un file dal tuo computer a un server remoto. Lo farai accedendo a una pagina Web che ti permetterà di inserire il nome di un file memorizzato sul tuo computer (che contiene il testo ASCII di Alice nel Paese delle Meraviglie), e quindi trasferire il file su un server Web utilizzando il metodo HTTP POST. Stiamo utilizzando il metodo POST anziché il metodo GET poiché vorremmo trasferire una grande quantità di dati dal tuo computer a un altro computer. Durante l'upload terremo attivo Wireshark per poter registrare la traccia dei segmenti TCP inviati e ricevuti dal tuo computer.

Fai quanto segue:

- Avvia il tuo browser web.
- Vai su http://gaia.cs.umass.edu/wireshark-labs/alice.txt e recupera una copia ASCII di Alice nel paese delle meraviglie.

- Archivia questo file da qualche parte sul tuo computer.
- Successivamente vai a http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.
- Carica il file precedentemente scaricato
- Dovresti vedere una schermata Wireshark che assomiglia a:



2. Primo sguardo

Prima di analizzare in dettaglio il comportamento della connessione TCP, diamo una visione di alto livello della cattura.
• Per prima cosa, filtra i pacchetti visualizzati nella finestra di Wireshark inserendo "tcp" (minuscolo, senza virgolette e non dimenticare di premere Invio dopo aver inserito!) Nella finestra delle specifiche del filtro di visualizzazione verso la parte superiore della finestra di Wireshark.

Quello che dovresti vedere è una serie di messaggi TCP e HTTP tra il tuo computer e gaia.cs.umass.edu. Dovresti vedere l'handshake a tre vie iniziale contenente un messaggio SYN. Dovresti vedere un messaggio HTTP POST. A seconda della versione di Wireshark che stai utilizzando, potresti vedere una serie di messaggi di "Continuazione HTTP" inviati dal tuo computer a gaia.cs.umass.edu. Ricorda che questo è il modo in cui Wireshark indica che ci sono più segmenti TCP utilizzati per trasportare un singolo messaggio HTTP. Nelle

versioni più recenti di Wireshark, vedrai "[segmento TCP di una PDU riassemblata]" nella colonna Info del display di Wireshark per indicare che questo segmento TCP conteneva dati che appartenevano a un messaggio di protocollo di livello superiore (nel nostro caso, HTTP). Dovresti anche vedere i segmenti TCP ACK restituiti da gaia.cs.umass.edu al tuo computer.

(Da qui in poi, se vuoi puoi usare un file di cattura pre-fabbricato, anziché la cattura personalizzata: prendi il file chiamato *tcp-ethereal-trace-1* all'interno dell'archivio ZIP in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip)

Se possibile, quando rispondi a una domanda, annota oppure prendi uno screenshot dei pacchetti all'interno della cattura che hai usato per rispondere alla domanda posta.

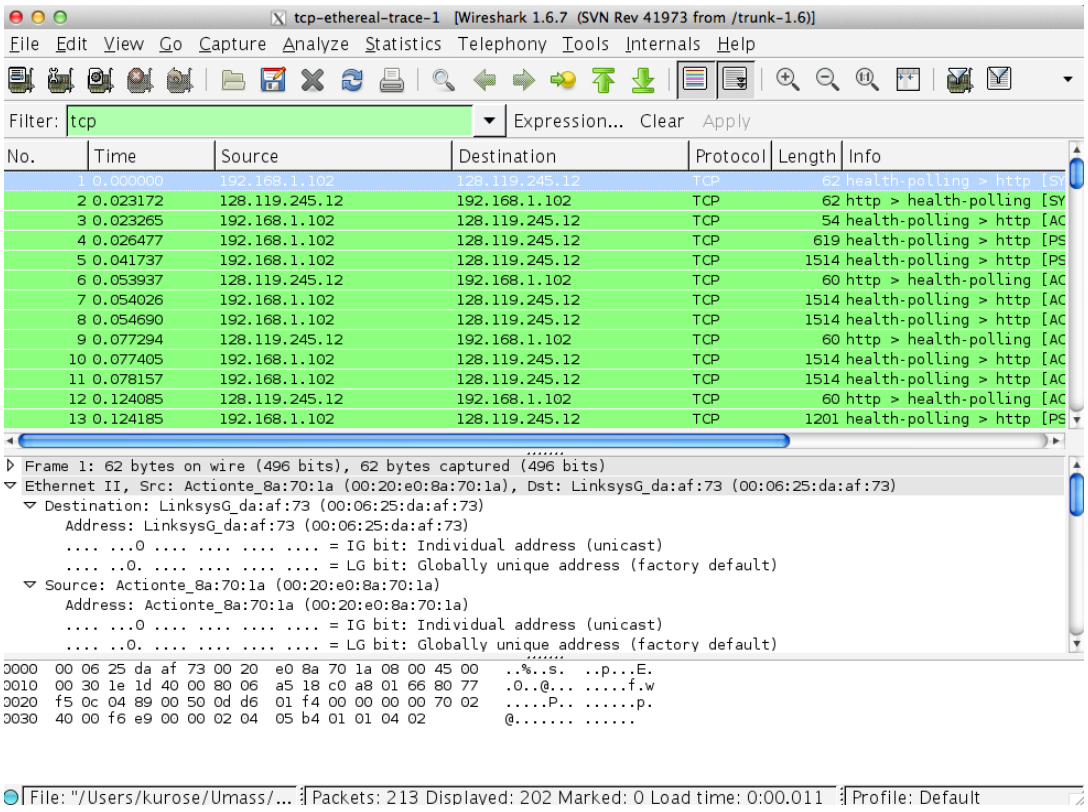
1. Qual è l'indirizzo IP e il numero di porta TCP utilizzati dal computer client (origine) che sta trasferendo il file a gaia.cs.umass.edu? Per rispondere a questa domanda, è probabilmente più semplice selezionare un messaggio HTTP ed esplorare i dettagli del pacchetto TCP utilizzato per trasportare questo messaggio HTTP, utilizzando i "dettagli della finestra di intestazione del pacchetto selezionato" (fare riferimento alla Figura 2 nella "Guida introduttiva Wireshark"- Lab se non sei sicuro delle finestre di Wireshark).

2. Qual è l'indirizzo IP di **gaia.cs.umass.edu**? Su quale numero di porta invia e riceve segmenti TCP per questa connessione?

Se stai usando un tuo file di cattura, rispondi alla seguente domanda:

1. Qual è l'indirizzo IP e il numero di porta TCP utilizzati dal computer client (origine) per trasferire il file su gaia.cs.umass.edu?

Poiché questa esercitazione riguarda TCP piuttosto che HTTP, cambiamo la finestra "Elenco dei pacchetti acquisiti" di Wireshark in modo che mostri le informazioni sui segmenti TCP contenenti i messaggi HTTP, piuttosto che sui messaggi HTTP. Per fare in modo che Wireshark esegua questa operazione, seleziona Analizza→ Protocolli abilitati. Quindi deseleziona la casella HTTP e seleziona OK. Ora dovresti vedere una finestra di Wireshark simile a:



Questo è ciò che ci interessa esaminare: una serie di segmenti TCP inviati tra il tuo computer e gaia.cs.umass.edu. Useremo la traccia del pacchetto che hai catturato (e/o la traccia del pacchetto **tcp-ethereal-trace-1** in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip) per studiare il comportamento del TCP nel resto di questo testo.

3. TCP Basics

Rispondi alle seguenti domande per i segmenti TCP:

1. Qual è il numero di sequenza del segmento TCP SYN utilizzato per avviare la connessione TCP tra il computer client e gaia.cs.umass.edu? Cosa c'è nel segmento che identifica il segmento come segmento SYN?
2. Qual è il numero progressivo del segmento SYN/ACK inviato da gaia.cs.umass.edu al computer client in risposta al SYN? Qual è il valore del campo ACK nel segmento SYN/ACK? In che modo gaia.cs.umass.edu ha determinato quel valore? Cosa c'è nel segmento che identifica il segmento come segmento SYN/ACK?
3. Qual è il numero di sequenza del segmento TCP contenente il comando HTTP POST? Nota che, avendo disabilitato la decodifica del protocollo HTTP, per trovare il comando POST, dovrai esplorare il contenuto del pacchetto nella parte inferiore della finestra di Wireshark, cercando un segmento con un "POST" all'interno del suo campo DATI.

4. Considera il segmento TCP contenente HTTP POST come il primo segmento nella connessione TCP. Quali sono i numeri di sequenza dei primi sei segmenti nella connessione TCP (incluso il segmento contenente HTTP POST)? A che ora è stato inviato ogni segmento? Quando è stato ricevuto l'ACK per ogni segmento? Data la differenza tra il momento in cui è stato inviato ciascun segmento TCP e il momento in cui è stato ricevuto il relativo riconoscimento, qual è il valore RTT per ciascuno dei sei segmenti?
5. Se hai voglia di scoprire di più, utilizza Internet per scoprire cos'è l'**EstimatedRTT**. Qual è il suo valore dopo la ricezione di ogni ACK? Si supponga che il valore iniziale di EstimatedRTT sia uguale all'RTT misurato per il primo segmento e quindi venga calcolato utilizzando l'equazione per EstimatedRTT per tutti i segmenti successivi.
   Suggerimento: EstimatedRTT = (1 - α)*EstimatedRTT + α*SampleRTT **α = 0.125**

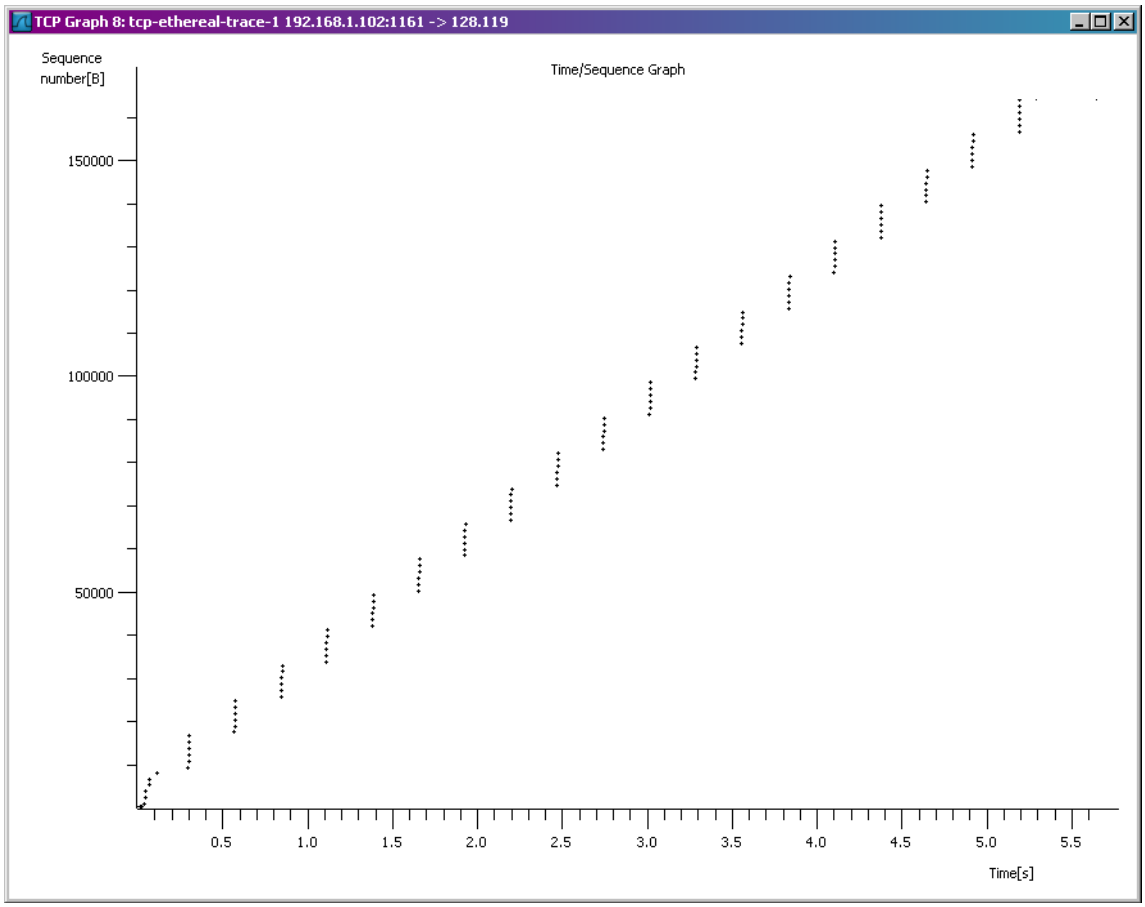
Nota: Wireshark ha una bella funzionalità che ti consente di tracciare l'RTT per ciascuno dei segmenti TCP inviati. Selezionare un segmento TCP nella finestra "elenco dei pacchetti acquisiti" che viene inviato dal client al server gaia.cs.umass.edu. Quindi selezionare: Statistics → TCP Stream Graph → Round Trip Time Graph.

6. Qual è la lunghezza di ciascuno dei primi sei segmenti TCP?
7. Qual è la quantità minima di spazio buffer disponibile notificata dal ricevente per la durata dell'intera cattura? La mancanza di spazio nel buffer del ricevitore limita mai il mittente?
8. Sono presenti segmenti ritrasmessi nel file di cattura? Cosa hai controllato per rispondere a questa domanda? Cosa controlla secondo te Wireshark per individuare i segmenti marcati come duplicati? Se nella tua cattura non sono presenti segmenti ritrasmessi, utilizza le catture del nostro archivio per visualizzarli.
9. Quanti dati normalmente riconosce il destinatario in un ACK? Riesci a identificare i casi in cui il destinatario conferma cumulativamente con un solo ACK più di un segmento ricevuto?
10. Qual è il throughput (byte trasferiti per unità di tempo) medio per la connessione TCP? Spiega come calcoleresti questo valore.

4. TCP congestion control in action

Esaminiamo ora la quantità di dati inviati per unità di tempo dal client al server. Piuttosto che calcolare manualmente questo dai dati grezzi presenti nella finestra di Wireshark, utilizzeremo una delle funzioni di rappresentazione grafica TCP di Wireshark, la Time-Sequence-Graph (Stevens) - per tracciare i dati.

1. Selezionare un segmento TCP nella finestra "Elenco dei pacchetti acquisiti" di Wireshark. Quindi selezionare il menu: Statistics → TCP Stream Graph → Time-Sequence-Graph (Stevens). Dovresti vedere un grafico simile al grafico seguente, che è stato creato in base ai pacchetti catturati nella traccia del pacchetto **tcp-ethereal-trace-1** in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip:



Qui, ogni puntino rappresenta un segmento TCP inviato, che traccia il numero di sequenza del segmento rispetto al momento in cui è stato inviato. Notare che un insieme di punti sovrapposti rappresenta una serie di pacchetti che sono stati inviati a raffica dal mittente, mentre è facile notare i momenti di silenzio dovuti alla finestra di invio che si è riempita.

2. Rispondi alle seguenti domande per i segmenti TCP che trovi nella traccia del pacchetto **tcp-ethereal-trace-1** in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip
    1. Utilizzare lo strumento di tracciamento del grafico della sequenza temporale (Stevens) per visualizzare il numero di sequenza rispetto al grafico del tempo dei segmenti inviati dal client al server gaia.cs.umass.edu. Riesci a identificare dove inizia e finisce la fase di avvio lento di TCP e dove prende il sopravvento la modalità congestion avoidance? Noti delle differenze rispetto al comportamento idealizzato del TCP che viene presentato nel libro? Quali?
    2. Rispondi a ciascuna delle due domande precedenti per la traccia che hai raccolto tu quando hai trasferito un file dal tuo 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:
https://www.mat.unical.it/ianni/SOR-Web/codice/UDPSocket/UDPServer.py
https://www.mat.unical.it/ianni/SOR-Web/codice/UDPSocket/UDPClient.py

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 lo ritenete necessario.
Che cosa potreste fare altrimenti per catturare del traffico UDP? Se non riuscite a catturare dei pacchetti UDP con Wireshark allora potete scaricare un file di cattura già confezionato per voi dall'indirizzo:
https://www.mat.unical.it/ianni/SOR-Web/esercitazioni/reti/EsercitazioneWireshark.zip

In tal caso memorizzate il file di cattura 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.
• Guardando il contenuto del pacchetto, verificate di saper determinare 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.