Il protocollo TCP

Obiettivo

Durante questa esercitazione studieremo il funzionamento del protocollo TCP. In particolare analizzeremo la "traccia" di segmenti TCP scambiati tra il vostro computer ed un server remoto.

Prima di iniziare la nostra esercitazione su TCP, avremo bisogno di Wireshark per ottenere la "traccia" del trasferimento di un file dal vostro computer ad un server remoto utilizzando il protocollo TCP.

Procedura Windows

- Aprite il web browser. Andate all'indirizzo <u>http://gaia.cs.umass.edu/wireshark-labs/alice.txt</u> 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*. Però non cliccate *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:

📶 (Un	titled) - W	iresharl	k																	_ []	×
Ble	Edit <u>V</u> iew	<u>G</u> o 9	apture	<u>A</u> nalyzi	e Statis	tics	Help														
8	¥ 8	(@(-	þ		×	Pg.	<u>_</u>	9	⇔	¢	ø	Ŧ	⊉	C		•	, Q	0	7	
Elter:										T D	pressio	n <u>C</u>	lear g	<u>А</u> рріү							
No	Time		Source				Destina	tion		Pre	stacol	lnfo									×
	1 0.00	0000	192.1	68.2.	145		128.1	19.2	45.12	TC	P	1250) > h	ittp	SYN	5eq-0	Len	-0 MSS	CIA 60		
	2 0.04	6402	142.1	19.24 6 9 1	5.12 145		192.1	168.2	1.145 4 5 1 3	TC IC	P.	1250		.250	LEYN,	ACK]	Seq=I	J ACK:	=⊥ wnr	1=515410	
	4 0.04	6963	192.1	68.2.	145		128.1	19.2	45,12	HT	TP	POST	/et	:here	al-1a	abs/lab	3-1-1	reply.	htm H	ITTP/1	1
	5 0.04	7339	192.1	68.2.	145		128.1	19.2	45.12	нт	ΤP	Cont	rinua	tior	or r	non-HTT	P tri	affic			
	6 D.12	8451	128.1	19.24	5.12		192.1	L68.Z	.145	TC	P.	http) > 1	.250	[ACK]	Seq=1	Ack:	=514 \	rfin=64	32 Le	
	7 D.12 P 0.11	8619 074 7	192.1 102.1	88.7. 20 1	145		128.1	19.2	45.12	HT	TP	Cont	innua	tion	I OF F	DON-HTT	P Tri	attic sffic			
	B 0.21	41.61	178.1	9 0. 21	5.12		107.1	68.7	2.145	TC	0	http		2.50	[ACE]	Sect-1	Ack:	-1.966	Win-P	2717 L	
	10 0.21	4315	192.1	68.1.	145		128.1		45.12	HT	TP	CONT	-	wier		IDN-HTT	PITH	di li la	wini=s		
	11 0.21	4415	192.1	68.2.	145		128.1	19.2	45.12	HT	ΤP	Cont	inua	tion	or r	non-HTT	P tri	affic			
	12 0.29	8180	128.1	19.24	5.12		192.1	L68.2	2.145	TC	P.	http) > 1	2.50	[ACK]	Seq-1	. Ack	-3418	Win-1	1616	
	13 0.29	8326	192.1	68.Z.	145		128.1	19.2	45.12	HT	TP	Cont	inua	tior	or r	non-HTT	P tri	affic			183
	14 D.3E	1927	128.1	19.24	5.12		192.1	.68. z	2.145	то	P.	nttp) > 1	.2.50	[ACK]	seq-1	ACK •	4870	whn=3	4520	
	15 D 38 15 D 38	2241	192.1	90.2. 68 7	145		128-1	19.2	MS 12	HT	TD	Cont	finua	tion			D TP:	affic			
	17 0.36	2459	192.1	68.1.	145		128.1	19.7	45.12	HT	TP	CONT	inua	tion			PT	affic			
	18 0.42	1386	192.1	68.2.	102		192.1	168.2	2.255	NE	IN5	Name	: que	erv N	IB MSF	HOME <1b	2				183
	19 0.46	6467	128.1	19.24	5.12		192.3	L68. Z	.145	TO	P	http	> > 1	2.50	[ACK]	Seq=1	Ack:	=6322	Win=3	7424	
	ZD D.55	2453	128.1	19.24	5.12		192.1	L68.2	.145	TC	P	http) > 1	.2 50	[ACK]	Seq=1	. Ack:	=7774	Win=2	0328	
	21 0.62	4375	128.1	19.24	5.12		192.1	.68.2	2.145	TC	P	http	> 1	2.50	[ACK]	seq-1	. Ack	-8957	Win=2	3232	
	22 0.02	4707	192.1	08.2.	145		128.1	19.2	40.12	HT	TP	Cont	innua innua	tion	I OF F	100-HTT	P tri	attic affic			
	23 0.62 24 0.62	4037	192.1	00.2. 62 1	1월 5 1월 5		128.1	19.2	93,12 25 17		TP	CONC	inua inua	mior	I OF F		P LP	4111C			
	25 0.70	8403	128.1	19.24	5.12		192.1	68.2	.145	TC	P	http	1 > 1	2.50	[ACK]	Seg-1	Ack	-1.04.09) Win-	26136	
	26 D.79	4139	128.1	19.24	5.12		192.1	L68.2	.145	ŤĊ	P	http	> 1	250	[ACK]	Seg=1	Ack	-11861	. win-	29040	
	27 D.86	6343	128.1	19.24	5.12		192.1	L68.2	.145	TC	P	http) > 1	.250	[ACK]	Seq=1	. Ack:	=13053	s Wina	31944	
	2B D.86	8855	192.1	68.2.	145		128.1	19.2	45.12	HT	ΤP	cont	rinua	tior	I OF F	non-HTT	P Tri	affic			
	29 0.86	9431	192.1	68.2.	145		128.1	19.2	45.12	HT	TP	Cont	inua	tior	or r	100-HTT	P tra	affic			
	30 0.85 21 0.85	9544	122.1	10.24	5 4 5		102.1	60.7	43,12	11	IP .	Long	innua	2.50	Lock I	DDHHHH	H Th	14503	s ucho.	22767	
	32 1.03	6220	128.1	19.24	5.12		192.1	168.2	2.145	TC	5	http	1	2.50	LACK1	Seget	Acles	15053	z win-	32767	
	33 1.10	8269	128.1	19.24	5.12		192.1	168.2	.145	TO	P	http	51	2 50	TACK	Seg=1	Ack	17149	Win-	32767	123
1						_															2
									15												<u>.</u>
E Fr	amel (62 Dyn	es on	whre	, 62)	oyτe	s cap	oture	2CI)												
E ET	hernet	II, SM	'C: Ne	cgear	_61:80	9:60	(00)	:09:5	b:61:8	e:6d),	DST	: L1r	nksys	sg_4 :	5:90:	aa (bo	DC:4	1:45:	9D:aBj	2	
E Int	ternet	Protes	:ol, S	.c: 1	92.168	8.2.	145 ((192,	168.2.	145),	Dst:	128.	,119,	.245.	12 C	128.119	9.245	.12)			
E Tri	ansmiss	ian Ca	ontrol	Prot	pcol,	Sno	Port	:: 12	250 (12	50), 0)st P	ort:	http	р (ВС	1), S	eq: D,	Len:	0			
0000	00 DC	41 45	90 aB	00 0	19 55	61	8e 6	d 08	00.45	00			[a.m	E.							٦
0010	00 30	2b 6b	40 00	80 0	36 BC	9f	c0 a	8 02	91 80	77	. D+k@	a		w							
DOZD	f5 Dc	04 ez	00 50	CZ 6	17 ZZ	99	00 0	0 00	00 70	02	• • • • •	P.g	·	p.							
0030	FT FT	60 ZF	00 00	02 (14 05	104	OT 0	I 04	92		•• /•		• • • •	••							
File: "C:	ODDCLIME ₂	4 IPALL	UAV11 DO	4 Sec11	Tennlet	herXX	Xen31	10" 165	KB 00:00:	09 P: 3	214 D: 3	214 M: 0	Drope	e û							1

Se non vi è possibile catturare questi pacchetti con Wireshark allora potete scaricare un file di cattura già confezionato per voi dall'indirizzo

https://www.mat.unical.it/informatica/Reti di Calcolatori?

<u>action=AttachFile&do=get&target=tcp-trace.</u> Memorizzate questo file sul vostro computer e apritelo da Wireshark attraverso il comando *Open* del menu *File*.

Domande Windows:

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*?

Adesso andate tutti su Analyze \rightarrow Enabled Protocols e poi deselezionate il box HTTP e cliccate su OK. Adesso dovreste vedere una schermata simile a quella riportata di seguito:

📶 top-ethereal-trace-1 -	Wireshark					_							
⊟le <u>E</u> dit Vjew <u>G</u> o <u>C</u> a	spture Analyze Statistics	det											
	🕷 🗈 🕅 x	% 🛆 🛛 🛇	କ କ ବ ଟ୍	7 2 T		ବ୍ଦ୍ 🗆							
Biter: Expression Clear Apply													
No Time	Source	Destination	Protocol Info				<u> </u>						
1 0.000000	102.168.1.102	128.119.245.12	TCP 1161	> http [S]	VN] Seq=0 Len-	-0 M55-1460							
3.0.023265	192.168.1.102	192,108,1,102	TCP 1161	> 1101 [S > http [A	YN, ACKI SEQHU CKI Seqhi Ack:	/ ACKHI WINHDA 1 Win=17520 (844 Let						
4 0.026477	192.168.1.102	128.119.245.12	TCP 1161	> http [P	SH, ACK] Seq=1	L Ack=1 Win=1	75.						
5 0.041737	192.16B.1.102	128,119,245,12	TCP 1161	> http (P	SH, ACK] Seq=5	i66 Ack−1 win	-1.						
6 0.053937	128.119.245.12	192.16B.1.102	TCP http	> 1161 [A	CK] Seq-1 ACk.	-566 win=6780	1488						
7 0.034020	192,108,1,102	128,119,243,12	TCP 1101	> http [A	CK Sec=2020 /	ACK-1 W10-175. ACK-1 W10-175	20						
9 0.077294	128.119.245.12	192.168.1.102	TCP http	> 1161 [A	CK] Seg=1 Ack:	=2026 Win=876	δĩ						
10 0.077405	192.16B.1.102	128,119,245,12	TCP 1161	> http [A	ск] seq=4946 л	4ck=1 win=175)	20						
11 0.078157	192.16B.1.102	128.119.245.12	TCP 1161	> http [A	CK] Seq=6406 A	Ack-1 win-1753	20						
12 0.124085	103 168 1 102	192,108,1,102	TCP Http TCP 1161	> 1101 [AV	CKJ SECHI ACK·	*5480 W10=1100 7966 Ack=1 W50	80.00						
14 0.169118	128, 119, 245, 12	192.16B.1.102	TCP http	> 1161 A	CK] Sec=1 Ack:	=4946 Win=1460	00						
15 0.217299	128.119.245.12	192.168.1.102	TCP http	> 1161 [A	CK] Seq=1 Ack:	=6406 Win=1752	20 🛛						
16 0.267802	12B.119.245.12	192.16B.1.102	TCP http	-> 1161 FA	ск] sed=1 Ack-	-7866 win-204-	40 🖃						
<u>×</u>							<u></u>						
⊞ Frame 7 (1514 by	∕tes on wire, 1514 b	ytes captured)					-						
⊞ Ethernet II, sro	:: Actionte_8a:70:1a	a (00:20:e0:8a:70	:1a), DST: Link	sysg_da:af	173 (00:06:25	:da:af:73)							
⊞ Internet Protoco	ol, Snc: 192.168.1.1	.02 (192,168,1,10)	2), Dst: 128.11	L9.245.12 (128,119,245,1	2)							
E Transmission Con	ntral Protocol, Src	Port: 1161 (1161)), Dst Part: ht	tp (80), s:	ieq: 2026, Ack	; 1, Len; 14€	50						
Source port: 1	1161 (1161)												
Destination po	ort: http (80)												
sequence numbe	er: 2026 (relativ	/e_sequence_numbe	r)				- 11						
[Next sequence	e number: 3486 (1	elative sequence	number)]				- 10						
Acknowledgement number: 1 (relative ack number)													
Header length: 20 bytes													
E Flags: 0×10 (A	КСК)						- 18						
D =	Congestion Window #	educed (CWR): No	t set										
.0 =	ECN-Echo: Not set						- 13						
	urgent: Not set												
1	Acknowledgment: set												
<u>4</u>							<u>.</u>						
0000 00 06 25 da -	af 73 00 20 e0 8a.	70 la 08 00 45 00 co pp 01 66 pp 73)%s	pE.			-						
D020 f5 0c D4 89	40 00 80 08 91 88 00 50 0d d6 09 de	CU AB UI 66 B0 77 34 a2 74 la 50 10)	4.t.P.			- 13						
0030 44 70 b9 8e	00 00 0d 0a 0d 0a	57 65 20 61 72 65	Dp	we are									
0040 20 6e 6f 77 1	20 74 72 79 69 6e	67 20 74 6f 20 72	naw try in	g to r			1						
File: "C: Documents and Settin	ngsl/Paula Wing(Ny Documents)	Wiresharklitraces - ethereal.	P: 213 D: 213 M: 0				1						

• 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?
- Riabilitate la visualizzazione del protocollo IP dal menu *Analyze/Protocols*. 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 connessine 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-Sequence-Graph(Stevens)* – per rappresentare i nostri dati in maniera grafica.

Selezionate un segmento TCP e poi cliccate su *Statistics* \rightarrow *TCP Stream Graph* \rightarrow *Time-Sequence-Graph(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 *Time-Sequence-Graph(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*.

II protocollo UDP

Obiettivo

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. Che cosa potreste fare? Come prima, se non riuscite a catturare dei pacchetti con Wireshark allora potete confezionato scaricare un file di cattura già per voi dall'indirizzo https://www.mat.unical.it/informatica/Reti di Calcolatori? action=AttachFile&do=get&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 Windows:

- 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.
- Cercate la parola "UDP" da Google e determinate i campi sui quali viene fatto il checksum UDP.
- 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 (traceroot con linux).

Ricordiamo che questo comando ha un comportamento differente tra linux e windows. Con linux, traceroot unvia 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 TTL-excedeed pervenuti alla vostra macchina dai vari router che i messaggi ICMP hanno attraversato.

Domande Windows:

- Individuare il primo messaggio ICMP inviato dalla vostra macchina e espandere 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 devono cambiare? Perché?
- Descrivi e giustifica i valori che vengono visualizzati nel campo Identification del pacchetto IP;

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?