Eserciziario - Reti di calcolatori

VR443470

febbraio 2023

Indice

Prefazione

Questo eserciziario è stato creato con il solo scopo di ripassare i concetti principali, attraverso i temi d'esame presentati durante gli anni accademici. Il documento è scritto da uno studente universitario. Di conseguenza, qualsiasi concetto proposto ***potrebbe*** essere errato. Si consiglia al lettore, di far riferimento al libro di testo e al professore per avere informazioni dettagliate e attendibili.

Il testo propone diversi temi d'esame con le classiche 3 domande di teoria iniziali e 3 esercizi. Ad ogni quesito teorico viene fornita una risposta completa, cercando di esporre i concetti in maniera chiara. Allo stesso modo, gli esercizi pratici presentano dei grafici e i vari calcoli per cercare di essere i più delucidanti possibili.

Infine, il documento presenta anche una serie di riferimenti agli esercizi affrontati e una tabella contenente le domande proposte nei temi d'esame. In questo modo, il lettore avrà un riferimento ordinato nel caso in cui, per esempio, voglia rivedere un argomento specifico.

Fonti delle informazioni presentate nel documento:

- Docente del corso di Reti di calcolatori - UniVR: Carra Damiano
- Libri del corso consigliati dal docente:
  1. Libro utilizzato dal sottoscritto:
     - Autori: *James F. Kurose, Keith W. Ross*
     - A cura di: *Antonio Capone, Sabrina Gaito*
     - Titolo: Reti di calcolatori e internet. Un approccio top-down. (7^a edizione)
     - ISBN-13: 978-8891902542
     - Link Amazon (no ref)
  2. Altro libro consigliato:
     - Autori: *Andrew S. Tanenbaum, David J. Wetherall*
     - Traduttore: *Dario Maggiorini, Sabrina Gaito*
     - Titolo: Reti di calcolatori. (5^a edizione)
     - ISBN-13: 978-8891908254
     - Link Amazon (no ref)

1 Temi d'esame

1.1 Esame - 05/02/2013

1.1.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si descriva il problema del "terminale nascosto" (*hidden terminal problem*) nelle Wireless LAN e la soluzione adottata dallo standard 802.11.

2. In riferimento al livello di rete, si spieghi che cosa succede quando un host si connette ad una rete ed ha bisogno di ricevere un indirizzo IP (non è necessario andare nei dettagli dei protocolli, è sufficiente descrivere a grandi linee i messaggi scambiati).

3. L'header del protocollo UDP contiene solo 4 campi: Source Port, Destination Port, Length e Checksum. Si spieghi brevemente a cosa servono tali campi.

Risposte

1. Si supponga esistano tre stazioni A, B e C. La stazione A e C stanno trasmettendo certi dati alla stazione B. In questo scenario, si introduce il problema del terminale nascosto, il quale nasce per **due motivi**:

   - Gli **Ostacoli fisici**. Potrebbero essere presenti nell'ambiente impedimenti fisici che non consentono alle due stazioni trasmettenti (A, C) di sentirsi a vicenda, nonostante la destinazione sia la medesima.
   - **Fading (evanescenza)**. Potrebbe manifestarsi un fenomeno fisico, appunto l'evanescenza, che crea una collisione. Questo problema non viene rilevato dalla stazione ricevente.
     In altre parole, il trasmittente invia il segnale, ma a causa di una continua variazione d'intensità del segnale tra il valore massimo e quello minimo (*fading*), il destinatario riceve un segnale compromesso pensando che sia corretto. Questo accade perché il destinatario non può rilevare la collisione.

   La **soluzione** presentata dal **protocollo MAC 802.11** è quella di includere uno schema di prenotazione. Vengono creati due frame di controllo chiamati: RTS (*request to send*) e CTS (*clear to send*). Il primo viene inviato dal mittente al destinatario, indicando il tempo di connessione nel campo DATI; il secondo viene inviato dal destinatario una volta ricevuto l'RTS. Il *clear to send* viene **inviato in broadcast**, in questo modo viene comunicato a ciascun host collegato di non disturbare la comunicazione.
   Tale soluzione risolve il problema poiché il frame DATI viene trasmesso solamente una volta prenotato il canale tramite il frame RTS.

2. I messaggi scambiati durante l'assegnazione dell'indirizzo IP ad un host, sono 4: DHCP discover, DHCP offer, DHCP request, DHCP ACK:

   (a) **DHCP discover**, ovvero individuazione del server. Il mittente si collega alla rete e cerca di individuare un server. Per farlo, invia un segmento UDP, chiamato DHCP discover, in broadcast a tutti i server presenti nella rete. Al momento della connessione, il mittente ha come indirizzo IP speciale 0.0.0.0 e come indirizzo IP destinatario 255.255.255.255 (*broadcast address*).

   (b) **DHCP offer**, ovvero offerta del server. Qualsiasi server presente all'interno della rete, che è interessato a fornire un indirizzo IP al mittente, risponde al segmento precedente inviando un DHCP offer. Anche in questo caso, il destinatario invia il messaggio in broadcast specificando inoltre:

   - ID univoco rappresentante l'identificativo del messaggio ricevuto;
   - Indirizzo IP offerto al mittente;
   - Maschera della sottorete;
   - Durata di connessione (*lease time*), ovvero la durata di tempo in cui l'indirizzo IP offerto sarà valido.

   (c) **DHCP request**, ovvero richiesta. Una volta che il mittente ha valutato tutte le offerte, risponde al server inviando un segmento DHCP request. Da adesso, i messaggi non saranno più in broadcast ma *end-to-end*.

   (d) **DHCP AKC**, ovvero conferma. Il destinatario riceve la richiesta del mittente e risponde con un messaggio di conferma (ACK). Con quest'ultimo messaggio viene convalidato l'indirizzo IP, e gli altri campi, proposti al mittente.

3. Il protocollo UDP ha come intestazione solo quattro campi tutti da 16 bit. Il campo **porta di origine** (*Source Port*) e **porta di destinazione** (*Destination Port*), vengono utilizzati principalmente dal destinatario. In particolare, il *socket* del destinatario, utilizzando questi due campi, saprà a quale processo passare i dati presenti nel segmento UDP. Il campo ***Checksum*** viene inserito dal mittente per consentire al destinatario di verificare l'integrità del pacchetto. Dato che esso è una sequenza di bit calcolata tramite un algoritmo, il destinatario calcola questo valore e verifica che il pacchetto non sia stato compromesso. Infine, la **lunghezza** (*Length*) rappresenta la somma dell'intestazione (*header*) e il campo DATI. Dato che quest'ultimo è variabile, grazie a questo campo si è a conoscenza della lunghezza ed è possibile distinguere un segmento UDP dal successivo.

1.2 Esame - 05/07/2013

1.2.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si descriva l'algoritmo CSMA nella sua variante Collision Detection (CSMA-CD), indicando il motivo che ha portato all'introduzione di tale variante.

2. L'header del protocollo IP contiene un campo chiamato "Time to live" (TTL): si spieghi come viene utilizzato tale campo e perché è stato introdotto.

3. Si descriva la fase di chiusura della connessione nel TCP, indicando i messaggi scambiati e i principali campi dell'header utilizzati durante tale fase.

Risposte

1. L'algoritmo CSMA-CD (*Carrier Sense Multiple Access - Collision Detection*), come dice il nome, è un algoritmo utilizzato al livello logico per gestire il problema dell'accesso multiplo. Esso consente di evitare collisioni e di conseguenza, non incorrere in perdite di frame.
   Nonostante l'esistenza di altri algoritmi, questo risulta il migliore, con un'efficienza quasi perfetta. La **differenza sostanziale** tra l'algoritmo CSMA e la variante CSMA-CD, riguarda la rilevazione della collisione. Nonostante entrambi condividano il principio di rilevamento della portante, ovvero il controllo del canale prima della trasmissione, soltanto la variante implementa anche la *collision detection*. L'**algoritmo** è così strutturato:

   (a) Il datagramma proveniente dal livello di rete viene elaborato dal livello di collegamento. Si ottiene così il frame che viene successivamente allocato nel buffer pronto per procedere all'invio.

   (b) L'host si mette in ascolto del canale:
   - Se è libero, inizia la trasmissione del frame;
   - Se è occupato, si mette in attesa finché non viene liberato il canale.

   (c) Durante la trasmissione del frame, l'host mittente rimane in ascolto e verifica che sul canale non ci siano altre trasmissioni da parte di altri host:
   - Se non ci sono segnali d'interferenza, la trasmissione si conclude con successo;
   - Se uno o più host prova ad occupare il canale, l'host mittente interrompe *immediatamente* la trasmissione del frame.

   (d) Questo passaggio viene eseguito solamente se la trasmissione è stata interrotta. Quindi, l'host mittente attende un tempo casuale, chiamato *backoff time*, e alla fine si rimette in ascolto del canale (punto b).

2. Nell'header del protocollo IP è presente un campo chiamato "Time to live" che indica il tempo di vita di tale datagramma. La sua **introduzione** è stata necessaria per evitare che un datagramma, non rimanga per sempre in circolo all'interno di una rete.
   Il suo **utilizzo** è il seguente. Il mittente, al momento della creazione del datagramma, inserisce nell'header IP, nel campo TTL, un numero indicante il tempo di vita massimo. Successivamente all'invio, qualsiasi router elabori quel preciso datagramma, effettuerà un decremento di un'unità nel campo TTL. Se il valore di quest'ultimo dovesse raggiungere il valore zero, allora il datagramma verrà scartato.

3. La fase di chiusura di una connessione TCP è strutturata in tre fasi (si assume che il client voglia comunicare la chiusura al server, ma può accadere anche il contrario):

   (a) Il client invia un segmento contenente nell'header il campo FIN, che si trova nei campi di flag, posto ad 1.

   (b) Il server risponderà con un ACK e aggiungerà anch'esso il campo FIN ad 1 solamente se non dovrà inviare altri dati al client.
   Nel caso in cui dovesse trasmettere altri dati, risponderà con un ACK, invierà i dati richiesti e concluderà la connessione, inviando un segmento con il campo FIN ad 1.

   (c) Il client, risponderà con una conferma, inviando un ACK al server.

1.2.2 Esercizio 1 - CSMA persistent

1.2.3 Esercizio 2 - Subnetting e tabella di routing

1.2.4 Esercizio 3 - Controllo della congestione TCP

1.3 Esame - 25/02/2014

1.3.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Per consentire il risparmio di energia nelle Wireless LAN (WLAN), le stazioni utilizzano il cosiddetto "Network Allocation Vector" (NAV): si spieghi che cos'è il NAV e come viene utilizzato.

2. In riferimento al livello di rete, si spieghi, anche attraverso un esempio, che cos'è il Network Address Translation (NAT), specificando per quale motivo tale funzionalità è stata introdotta.

3. In riferimento al livello di trasporto, si spieghi che cosa sono le "porte note" (Well-Known Ports) e il motivo per cui sono state introdotte.

Risposte

1. Il NAV (Network Address Vector) è un meccanismo utilizzato principalmente nel protocollo 802.11. Il **vantaggio principale** è il risparmio di energia da parte degli host all'interno di una WLAN. Il motivo è spiegato qui di seguito.
   Quando un host vuole comunicare con il destinatario (Access Point), invia una richiesta di invio (RTS, *request to send*) dopo aver attesto un tempo breve chiamato DIFS (*Distributed Inter-Frame Space*). Se il canale è inattivo, il destinatario risponde con un messaggio di conferma di invio (CTS, *clear to send*), solamente dopo aver atteso un breve periodo di tempo chiamato SIFS (*Short Inter-Frame Space*). A questo punto, il destinatario crea un'allocazione in cui imposta il valore di NAV, che corrisponderà al tempo totale di trasmissione del frame. Una volta terminata la trasmissione, il valore di NAV sarà a zero e dunque, il destinatario sarà di nuovo disponibile a ricevere dati (dopo ovviamente, l'invio dell'ACK di ricezione del datagramma al mittente).
   Ogniqualvolta un host vuole comunicare con l'Access Point, viene verificato se il canale è attivo o meno. Per cui, vi è un dispendio di energia e per tal motivo, il NAV evita di far effettuare una serie ripetitiva di richieste da parte degli host. Questo perché il destinatario (AP) comunica il valore di NAV ad ogni richiesta, così che il mittente possa aspettare ed evitare molteplici richieste.
   Quindi, valore di NAV diverso da zero, vuol dire canale occupato. Mentre, NAV uguale a zero, vuol dire canale libero.

2. All'interno di una rete locale, spesso vengono utilizzati degli indirizzi IP privati (il motivo esula dalla domanda, per cui non verrà esposto). Ma quest'ultimi non possono comunicare direttamente su Internet. Quindi, il **problema** che nasce riguarda la comunicazione al di fuori della rete locale.
   La **soluzione** è la NAT (*Network Address Translation*). Il router, abilitato al NAT, ha il compito di rappresentare la rete con un unico indirizzo IP, quindi, come se la rete locale fosse un unico dispositivo. Tuttavia, nel caso della ricezione di un pacchetto dal mondo esterno (fuori dalla rete locale), com'è possibile capire chi è il destinatario all'interno della rete?
   In questo caso, la **soluzione** prevede il salvataggio di una tabella all'interno del router, chiamata NAT *translation table*. Essa è composta da una colonna indicante l'indirizzo IP utilizzato su Internet con il numero di porta e l'indirizzo IP utilizzato nella rete locale con il numero di porta. Quindi, ogni riga avrà due indirizzi IP (privato e pubblico) con ogni numero di porta di ogni host.

3. Per porte note si intende un insieme di porte (range di valori $0 - 1023$) che sono associate ad applicazioni lato server. Questo bisogno nasce dal seguente **problema**.
   Nel momento in cui un client dovrà inviare un pacchetto ad un server, sarà sicuramente a conoscenza del suo numero di porta (sorgente) poiché assegnato dal sistema operativo, ma non sarà a conoscenza della porta del server.
   La **soluzione** è stata l'introduzione delle Well-Know Ports. Quindi, in base al protocollo utilizzato, il client inserirà un determinato valore nel campo "porta di destinazione". Per esempio, con il protocollo HTTP, il client inserirà il valore 80.

1.3.2 Esercizio 1 - ALOHA

1.3.3 Esercizio 2 - Subnetting e tabella di routing

1.3.4 Esercizio 3 - Controllo della congestione TCP + variante

1.4 Esame - 07/01/2019

1.4.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si descriva (ad esempio in pseudo-codice, corredato da commenti) l'algoritmo CSMA nella sua variante Collision Detection (CSMA-CD), indicando il motivo che ha portato all'introduzione di tale variante.
2. Si spieghi brevemente la funzionalità di frammentazione dei pacchetti IP, incluso le motivazioni e gli apparati che effettuano la frammentazione / deframmentazione.
3. L'header del protocollo UDP contiene solo 4 campi: Source Port, Destination Port, Length e Checksum. Si spieghi brevemente a cosa servono tali campi.

Risposte

1. La risposta è stata data a pagina 8 poiché è identica.
2. Si definisce l'unità massima di trasmissione (MTU), la massima quantità di dati che un frame a livello di collegamento può trasportare. Viene introdotto questo concetto per motivare il fatto che, a causa di limitazioni del collegamento, o a causa dei diversi protocolli utilizzati, il datagramma deve essere frammentato in più frame. Ecco il **motivo** della nascita di questa tecnica.
   Le sue **fasi** sono le seguenti:
   (a) **Creazione del datagramma** da parte dell'host. All'interno di esso vengono inseriti gli indirizzi sorgente e destinazione, e anche un numero identificativo (campo *Identification*). Il motivo di quest'ultimo, è la possibilità di rappresentare, con un codice univoco, ogni datagramma e quindi l'appartenenza allo stesso insieme da parte di più frame frammentati.
   (b) **Frammentazione datagramma** da parte del router. Nel farlo, esso contrassegna i frammenti con gli indirizzi di sorgente e destinazione, e con il valore del campo *Identification*.
   (c) **Deframmentazione datagramma** da parte del destinatario. Una volta ricevuto il pacchetto, il destinatario esamina il campo *Identification* per raggruppare tutti i frame appartenenti allo stesso datagramma frammentato. Inoltre, viene utilizzato il flag M per capire se il frame preso in considerazione è un pezzo intermedio (valore 1), oppure se è il frame finale (valore 0).
   Infine, per comprendere l'ordine corretto dei vari frame, viene utilizzato il campo offset. Quest'ultimo indica la distanza dal primo frame, il quale solitamente ha un identificativo pari a 000. Il campo offset viene utilizzato anche per capire se un frame è stato perso durante la comunicazione.

   Possibile **esempio**: link
3. La risposta è stata data a pagina 5 poiché è identica.

1.5 Esame - 05/02/2019

1.5.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si spieghi il funzionamento del protocollo ARP (Address Resolution Protocol), senza necessariamente entrare nei dettagli del protocollo stesso, specificando il motivo per cui è stato introdotto tale protocollo.

2. L'header del protocollo IPv6 contiene il campo "extension header": si spieghi come viene utilizzato e perché è stato introdotto.

3. Si descriva la modalità di instaurazione di una connessione TCP, specificando i messaggi scambiati e i campi più significativi dell'header utilizzati durante tale fase.

Risposte

1. Il protocollo ARP (Address Resolution Protocol) è stato introdotto per un **motivo** semplice. A livello di rete, ogni host viene identificato tramite il proprio indirizzo IP. Invece, a livello di collegamento, vi è un identificativo chiamato indirizzo MAC (Media Access Control) univoco (grazie allo standard IEEE, non possono esistere due schede di rete con lo stesso indirizzo). Sorge spontaneo il **problema**, ovvero come è possibile effettuare la risoluzione degli indirizzi da IP a MAC o viceversa.
   La **soluzione** proposta dal protocollo ARP, è quella di inserire all'interno della RAM di ogni nodo una tabella chiamata tabella ARP. Il suo contenuto è formato da tre colonne contenenti l'indirizzo IP, l'indirizzo MAC e il Time to live (TTL). Quest'ultimo campo, è stato introdotto per evitare che una riga rimanga all'interno della tabella per sempre. Quindi, una volta finito il tempo di vita (TTL), la voce viene eliminata dalla tabella.
   Inoltre, il protocollo ARP fornisce un pacchetto ARP, costruito dal trasmittente, che include vari campi inclusi l'indirizzo IP e MAC del mittente e destinatario. Questi pacchetti hanno lo stesso formato sia per quanto riguarda la richiesta che la risposta. Il motivo dell'introduzione di tale pacchetto, è dovuta al fatto che ogni nodo trasmittente interroga tutti gli altri nodi per ottenere l'indirizzo MAC corretto.

2. Nel protocollo IPv6 è presente un campo chiamato *next header*. Il suo compito è quello di indicare se è presente o meno un'intestazione successiva. Più precisamente, ha un doppio ruolo:

   (a) Se non è presente un'intestazione successiva, questo campo indica il protocollo utilizzato per trasportare il campo dati (*payload*). Per esempio TCP o UDP.

   (b) Se è presente un'intestazione successiva, allora il valore rappresenta il tipo di *extension header*.

   L'*extension header* è composto da due campi obbligatori: *next header* (8 bit) e *header length* (8 bit). Gli altri campi sono lasciati liberi. Il campo:

   - *Next header* ha lo stesso compito descritto sopra.
   - *Header length* identifica la lunghezza dell'estensione.

   Il **motivo** dell'introduzione di questo campo è strettamente legata all'estensibilità del protocollo. Se in futuro venisse deciso di introdurre uno o più campi, grazie all'*extension header*, sarà possibile aggiungere i nuovi valori senza cambiare drasticamente l'intero protocollo.

3. L'instaurazione di una connessione TCP viene chiamata *handshaking* a tre vie poiché vengono scambiati tre pacchetti. I **messaggi inviati e ricevuti** sono i seguenti:

   - Il **client** invia al server un **segmento SYN**. Tale pacchetto è speciale, perché privo del campo Dati. Presenta inoltre il campo SYN ad uno e un valore, chiamato numero di sequenza iniziale (*client initial sequence number*), generato dal client e posto nel campo numero di sequenza (*sequence number*).
   - Il **server** risponde al client con un **segmento SYNACK**. All'arrivo del pacchetto, il server inoltre alloca le varie risorse per essere pronto a scambiare i dati con il client.
     Ancora una volta, il pacchetto costruito è speciale poiché presenta solo tre valori significativi:
     - **SYN** a uno per comunicare che si è in una fase iniziale.
     - **ACK number** uguale al valore del numero di sequenza iniziale del client (*client initial sequence number*), più uno.
     - **Numero di sequenza** che è analogo a quanto è stato fatto per il client. Quindi, il server genera un numero iniziale di sequenza (*server initial sequence number*) e lo inserisce nel campo.
   - Il **client** conferma la ricezione del messaggio del server con un **segmento ACK**. A questo punto anche il client alloca le varie risorse per essere pronto a scambiare i dati con il server.
     Inoltre, il client può decidere se inviare il pacchetto di conferma o inviare il pacchetto contenente sia la conferma che le informazioni da inviare al server. Qualsiasi scelta prenda il client, è certo che i seguenti campi sono così definiti:
     - **SYN** a zero per comunicare che la connessione è stata stabilita.
     - **ACK number** uguale al valore del numero di sequenza iniziale del server (*server initial sequence number*), più uno.

1.6 Esame - 18/02/2020

1.6.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si descriva (ad esempio in pseudo-codice, corredato da commenti) l'algoritmo CSMA nella sua variante Collision Detection (CSMA-CD), indicando il motivo che ha portato all'introduzione di tale variante.

2. L'header del protocollo IP contiene un campo chiamato "Time to live" (TLL): si spieghi come viene utilizzato tale campo e il perché è stato introdotto.

3. L'header TCP contiene un campo chiamato "Sequence Number" (SN) e "Acknowledge Number" (AckN): spiegare il loro significato (Che cosa contengono tali campi?) e come vengono utilizzati (Chi imposta il loro valore? I campi vengono impostati in ogni segmento?).

Risposte

1. La risposta è stata data a pagina 8 poiché è identica.

2. La risposta è stata data a pagina 8 poiché è identica.

3. Questi due campi sono stati implementati per rendere affidabile la trasmissione dei dati.
   Durante l'**instaurazione di una connessione** tra client e server, entrambi i campi vengono utilizzati:

   - Inizialmente il client utilizza il SN per inserire il proprio numero di sequenza iniziale (*client initial sequence number*).
   - Successivamente, il server risponde al client e inserisce in SN il suo proprio numero di sequenza iniziale (*server initial sequence number*), e in AckN il valore del numero di sequenza iniziale del client incrementato di uno.
   - Infine, il client risponde al server inserendo nel campo AckN il numero di sequenza iniziale del server incrementato di uno.

Invece, **durante la comunicazione** di informazioni tra client e server:

- Nel campo Sequence Number viene inserito il numero presente nel flusso di byte del primo byte di segmento (si veda esempio sotto).
- Nel campo AckN viene inserito il valore del numero di sequenza del byte successivo che l'host mittente aspetta dal destinatario (si veda esempio sotto).

Esempio SN: Host A vuole inviare $500'000$ byte all'Host B. Si ipotizzi che la MSS (*Maximum Segment Size*) sia pari a $1'000$ byte e che il primo byte del flusso abbia come SN il valore 0. Dunque, i pacchetti vengono frammentati in 500 frame ($500'000 \div 1'000$) e inviati all'Host B. La numerazione dei pacchetti sarà: il primo $SN = 0$, il secondo $SN = 1000$, il terzo $SN = 2000$ e così via.
Esempio AckN: Host A ha ricevuto un segmento dall'Host B contenente i byte da 0 a 535 e un altro segmento contenente i byte da 900 a 1000. Per qualche motivo, l'Host A è ancora in attesa del segmento contenente i byte da 536 a 899. Il prossimo segmento inviato dall'Host A, avrà come AckN il valore 536. In questo modo, l'Host A sta comunicando all'Host B il flusso di dati mancante ($536 - 899$).

1.6.2 Esercizio 1 - CSMA persistent

1.6.3 Esercizio 2 - Subnetting e tabella di routing

1.6.4 Esercizio 3 - Controllo della congestione TCP

1.7 Esame - 10/01/2022

1.7.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si descriva il problema del "terminale nascosto" (hidden terminal problem) nelle Wireless LAN e la soluzione adottata dallo standard 802.11.
2. L'header del protocollo IP contiene un campo chiamato "Time to live" (TLL): si spieghi come viene utilizzato tale campo e il perché è stato introdotto.
3. Si descriva la modalità di instaurazione di una connessione TCP, specificando i messaggi scambiati e i campi più significativi dell'header utilizzati durante tale fase.

Risposte

1. La risposta è stata data a pagina 5 poiché è identica.
2. La risposta è stata data a pagina 8 poiché è identica.
3. La risposta è stata data a pagina 15 poiché è identica.

1.7.2 Esercizio 1 - CSMA persistent

1.7.3 Esercizio 2 - Subnetting e tabella di routing

1.7.4 Esercizio 3 - Controllo della congestione TCP

1.8 Esame - 01/02/2022

1.8.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si spieghi il funzionamento del protocollo ARP (Address Resolution Protocol), senza necessariamente entrare nei dettagli del protocollo stesso, specificando il motivo per cui è stato introdotto tale protocollo.

2. Non tutti gli indirizzi IP possono essere assegnati agli host: ve ne sono alcuni, definiti "indirizzi speciali" che hanno un particolare utilizzo. Tra questi, vi sono: l'indirizzo composto da tutti "1", l'indirizzo composto da tutti "0", l'indirizzo in cui l'host ID è composto da tutti "1", l'indirizzo in cui l'host ID è composto da tutti "0". Per ciascuno dei 4 indirizzi si spieghi il loro utilizzo.

3. In riferimento al livello di trasporto, si spieghi che cosa sono le "porte note" (Well Known Ports) e il motivo per cui sono state introdotte.

Risposte

1. La risposta è stata data a pagina 14 poiché è identica.

2. Questi indirizzi IP speciali, vengono chiamati indirizzi IP privati, cioè non possono essere assegnati a nessun host:

   - Indirizzo con tutti "1", utilizzato, per esempio, nel protocollo DHCP. Rappresenta un indirizzo di broadcast in cui il client, non conoscendo l'indirizzo della rete e non potendo dunque utilizzare il Direct Broadcast, invia il segmento di DHCP discover a tutti gli host della rete con l'indirizzo 255.255.255.255.
   - Indirizzo con tutti "0", utilizzato nel protocollo DHCP. Viene impiegato nel momento della prima connessione di un client in una rete. All'inizio, esso non ha un indirizzo IP e durante la prima fase di DHCP discover, gli viene assegnato un indirizzo IP speciale: $(0.0.0.0)_{10}$.
   - Indirizzo host ID con tutti "1", rappresenta un indirizzo di Direct Broadcast. Ovvero, un indirizzo di rete del genere ha lo scopo di far ricevere il messaggio inviato a tutti gli host della rete. I bit appartenenti al suffisso sono a uno.
   - Indirizzo host ID con tutti "0", rappresenta un indirizzo di rete. Ovvero, i bit del suffisso sono impostati a zero e non possono essere utilizzati da un host poiché identificano la rete stessa.

3. La risposta è stata data a pagina 11 poiché è identica.

1.8.2 Esercizio 1 - CSMA persistent

1.8.3 Esercizio 2 - Subnetting e tabella di routing

1.8.4 Esercizio 3 - Controllo della congestione TCP

1.9 Esame - 22/02/2022

1.9.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si spieghi che cosa si intende, quando si parla della funzionalità del livello 2, per *framing* e si descriva una delle possibili tecniche con un semplice esempio.

2. In riferimento al livello di rete, si spieghi che cosa succede quando un host si connette ad una rete ed ha bisogno di ricevere un indirizzo IP (non è necessario andare nei dettagli dei protocolli, è sufficiente descrivere a grandi linee i messaggi scambiati).

3. Si descriva la fase di chiusura della connessione nel TCP, indicando i messaggi scambiati e i principali campi dell'header utilizzati durante tale fase.

Risposte

1. Al livello di collegamento, o logico, esiste un **problema** chiamato *framing* (in italiano, delimitazione delle trame): quando il mittente finisce la fase del livello di collegamento (livello 2), passa il risultato al livello fisico. A questo strato, tutti i dati vengono rappresentati con dei bit e l'invio avviene tramite onde elettromagnetiche (e non solo). Il destinatario, riceve tali segnali e deve essere in grado di distinguere ogni frame per ricostruire i dati inviati dal mittente. Quindi, come è possibile distinguere un frame dagli altri?
   Esistono tre **soluzioni**:

   (a) **Intervalli temporali** è una soluzione banale che presenta una criticità. Innanzitutto, il suo funzionamento è elementare: viene inviato una trama, viene attesto un intervallo temporale, viene inviata la trama successiva, viene atteso un intervallo temporale e così via. La criticità di questa tecnica è che a causa di fenomeni fisici, i segnali vengono distorti con la conseguente perdita o cambiamento di informazioni.

   (b) **Contatore di caratteri (*character count*)** è una soluzione che modifica l'header. Infatti, viene inserito un nuovo campo nell'header in cui viene specificato il numero di Byte da inviare/ricevere. La criticità presente in questo caso, riguarda eventuali errori del valore del campo da parte del mittente. Quindi, un banale errore di programmazione potrebbe causare un'alterazione delle informazioni con questa soluzione.

   (c) **Byte di flag e *bit stuffing*** è una delle soluzioni migliori. Viene scelta una sequenza di bit (byte) da inserire all'inizio e alla fine della trama. Così facendo il destinatario può eseguire un banale controllo all'inizio e alla fine per capire la delimitazione di ogni trama. Inoltre, nel caso in cui ci fosse un errore durante l'inserimento della sequenza, il problema si ripercuote solo sue due trame e non a cascata.

Chiaramente, è possibile, per caso, che all'interno della trama sia presente una sequenza identica. In questo caso, la situazione viene gestita utilizzando un valore, chiamato *bit stuffing*. Quindi, viene aggiunto uno zero alla fine del byte flag al momento della trasmissione, mentre viene rimosso uno zero alla fine del byte di flag al momento della ricezione.

Non viene presentato nessun esempio poiché la spiegazione risulta ampiamente chiara.

2. La risposta è stata data a pagina 5 poiché è identica.

3. La risposta è stata data a pagina 8 poiché è identica.

1.9.2 Esercizio 1 - CSMA persistent

1.9.3 Esercizio 2 - Subnetting e tabella di routing

1.9.4 Esercizio 3 - Controllo della congestione TCP

1.10 Esame - 18/07/2022

1.10.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si scriva lo pseudo-codice, corredato da commenti, dell'algoritmo CSMA nella variante Collision Detection (CSMA-CD). Si indichi inoltre il motivo che ha portato all'introduzione di tale variante.

2. L'header IP contiene un campo di 16 bit denominato "Identification": si spieghi che cosa contiene tale campo e come viene utilizzato.

3. In riferimento a livello di trasporto, si spieghi la differenza tra Smoothed Round Trip Time (SRTT) e Round Trip Time (RTT), e come ciascuno di questi viene calcolato. Infine, si indichi come viene calcolato il Retransmission Timeout (RTO).

Risposte

1. Lo pseudo-codice dell'algoritmo CSMA-CD è stato implementato utilizzando il linguaggio di programmazione C. Prima di vedere l'algoritmo si espone il **motivo** che ha portato a tale variante. Nel livello di collegamento è necessario gestire multipli accessi ad un canale. Oltre ad altri algoritmi esistenti, venne creato il CSMA che aveva una caratteristica principale: il rilevamento della portante. Ovvero, quando il mittente doveva trasmettere informazioni sul canale, prima di trasmettere verificava che non ci fossero altri segnali già in trasmissione.
   Nonostante ciò, venne creata una sua variante ancora più precisa, addirittura con un tasso d'efficienza praticamente perfetto. La caratteristica implementata dal CSMA-CD è il rilevamento della collisione, oltre a quello della portante. Quindi, il mittente prima di inviare il segmento, controlla il canale per vedere se è libero (rilevamento della portante). Se è disponibile, invia il segmento e rimane in ascolto: nel caso in cui riceva un segnale da parte di un altro host, interrompe immediatamente l'invio di informazioni sul canale (rilevamento delle collisioni).

L'algoritmo è il seguente:

```
bool result = false;
int backoff_time = randInt()
// finche' non sono stati trasmetti i dati, si rimane nel loop
while (not (result))
{
    // si controlla se il canale e' libero (true se libero,
    alt. false)
    if channel_is_free{
        // inizia la trasmissione dei dati, ma in parallelo...
        send_data()
        // ... viene ascoltato il canale durante invio dei
    dati.
        result = listen_channel()
        // se la funzione rileva un altro host che prova ad
    inviare un dato
        // ritorna il valore "false" e dunque attende un tempo
     chiamato backoff time.
        // Ovviamente se listen_channel esegue un return prima
     della fine di send_data(),
        // vuol dire che un altro host ha provato a comunicare
    , altrimenti terminano insieme.
        // Si assume che send_data() possa far terminare il
    processo listen_channel() tramite
        // un segnale. In tal caso, la funzione listen_channel
    () ritorna true
    }
    // result uguale a false solo se il canale non e' libero
    oppure se l'invio dei dati e'
    // stato disturbato da un altro host (quindi
    listen_channel() ha bloccato l'esecuzione tornando false)
    if result == False
        timeout(random_time)
    else
        output("Tutto corretto")
}
```

2. Al livello 2 (di collegamento o logico) il protocollo IP provvede a eseguire l'inserimento di un valore nel campo *Identification*. Tale valore è indispensabile al destinatario. Infatti, nel momento in cui il router esegue la frammentazione del pacchetto ricevuto, inserisce in ogni frame il valore identificativo del pacchetto originario.
   Così facendo il destinatario, al livello 2, è in grado di ricostruire il pacchetto originario, raggruppando tutti i frame utilizzando il campo *Identification*.

3. Con **RTT** si indica il *Round Trip Time*, ovvero il tempo impiegato da un pacchetto per viaggiare dal client al server e poi tornare indietro. In altre parole è il tempo che intercorre tra l'invio del pacchetto e la ricezione della risposta. Invece, il ***Smoothed* RTT** rappresenta una stima del valore medio del RTT. Dunque, per ottenere è necessario eseguire una stima prendendo in considerazione il valore medio del *round trip time*.
   Infine, l'**RTO** (*retrasmission time-out*) è il tempo impiegato prima di ritrasmettere un pacchetto. All'invio di un pacchetto da parte del mittente, parte in parallelo un timer che indica il range di tempo entro quando il server deve rispondere. Quando scade tale tempo, il pacchetto viene inviato nuovamente. Inizialmente il valore è ad 1, dopo ogni perdita incrementa l'RTO per due rispetto all'RTO precedente ($\text{RTO}_{new} = \text{RTO}_{old} \times 2$).

1.10.2 Esercizio 1 - CSMA persistent

1.10.3 Esercizio 2 - Subnetting e tabella di routing

1.10.4 Esercizio 3 - Controllo della congestione TCP

1.11 Esame - 27/09/2022

1.11.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si spieghi la differenza tra commutazione di circuito e commutazione di pacchetto.

2. Nel protocollo IPv6 c'è un campo denominato "extension header", che a sua volta contiene altri sotto-campi. Si spieghi quali sono tali sotto-campi, come vengono utilizzati e perché il campo "extension header" è stato introdotto.

3. Si consideri una connessione TCP già instaurata e lo scambio di segmenti dovuti ad una HTTP GET. Si indichi il valore assunto dai campi dell'header TCP "Sequence Number" (SN) e "Acknowledge Number" (AckN) in entrambe le direzioni, spiegando le ragioni dei valori indicati.

Risposte

1. Nella **commutazione di circuito**, le risorse richieste lungo il percorso (buffer, ecc.) vengono riservate per l'intera durata della comunicazione.
   Al contrario, nella **commutazione di pacchetto**, la sorgente divide i messaggi in parti più piccole (pacchetti con *header*) ed essi viaggiano lungo il percorso attraversando collegamenti e commutatori di pacchetti. Quindi non hanno una corsia dedicata a loro.
   Le differenze principali è che la prima (circuito) ha una comunicazione riservata che garantisce velocità costante, ma un grande spreco di risorse e altre complicazioni riguardo la larghezza di banda *end-to-end*. Al contrario, la seconda condivide il suo percorso riuscendo ad ottimizzare al meglio le risorse, ma aumentando le possibilità di un'eventuale perdita (buffer overflow di un nodo intermedio per esempio) e incremento del ritardo a seconda del numero dei nodi intermedi e dello *store and forward* (ovvero attesa dell'intero pacchetto prima di inviarlo ad un altro nodo).

2. La risposta è stata data a pagina 14 poiché è identica.

3. Durante l'instaurazione di una connessione tra client e server, il protocollo TCP utilizza la tecnica handshaking a tre vie e i campi hanno i seguenti valori:

   - Il client imposta il valore del SN al suo valore iniziale, quindi al numero di sequenza iniziale del client (*client initial sequence number*).
   - Il server come risposta, imposta il valore del SN al suo valore iniziale, quindi al numero di sequenza iniziale del server (*server initial sequence number*). Inoltre, nel valore di ack (AckN) inserisce il valore passatogli dal client incrementato di 1.
   - Il client risponde confermando la ricezione del messaggio inviando un pacchetto in cui come AckN è presente il valore passatogli dal server incrementato di 1.

Invece, durante una richiesta HTTP GET inviata dal client, il server risponde al messaggio inviando l'informazione richiesta. In questo caso, il messaggio inviato è un HTTP REPLY. Il protocollo HTTP delega il controllo dell'affidabilità al protocollo TCP.
Quindi, durante l'invio di una richiesta GET e la risposta REPLY, i campi hanno i seguenti valori:

- Nel campo SN viene inserito il numero presente nel flusso di byte del primo byte di segmento (per esempio, $0, 1000, 2000, 3000$).
- Nel campo AckN viene inserito il valore del numero di sequenza del byte successivo che l'host mittente aspetta dal destinatario.

Si veda la terza domanda a pagina 17 per capire meglio gli esempi.

1.11.2 Esercizio 1 - CSMA persistent

1.11.3 Esercizio 2 - Subnetting e tabella di routing

1.11.4 Esercizio 3 - Controllo della congestione TCP

1.12 Esame - 09/01/2023

1.12.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1. Si spieghi lo scopo e il funzionamento del protocollo ARP (Address Resolution Protocol), e in particolare i messaggi trasmessi dalle stazioni, specificando gli indirizzi usati.
2. Si spieghi brevemente la funzionalità di frammentazione dei pacchetti IP, mostrando un caso in cui essa è necessaria, gli apprati coinvolti sia nella frammentazione che nel riassemblaggio e i principali campi dell'header coinvolti.
3. L'header del protocollo UDP contiene solo 4 campi: Source Port, Destination Port, Length e Checksum. Si spieghi brevemente a cosa servono tali campi.

Risposte

1. La risposta è stata data a pagina 14 poiché è identica.
2. La risposta è stata data a pagina 12 poiché è identica.
3. La risposta è stata data a pagina 5 poiché è identica.

1.12.2 Esercizio 1 - CSMA persistent

1.12.3 Esercizio 2 - Subnetting e tabella di routing

1.12.4 Esercizio 3 - Controllo della congestione TCP

2 Indice per ogni tipologia di domanda

3 Indice per ogni tipologia d'esercizio