

Eserciziario - Reti di calcolatori

VR443470

gennaio 2023

Indice

<i>Prefazione</i>	4
1 Temi d'esame	5
1.1 Esame - 05/02/2013	5
1.1.1 Domande sulla teoria	5
1.1.2 Esercizio 1 - CSMA persistent	7
1.1.3 Esercizio 2 - Subnetting e tabella di routing	7
1.1.4 Esercizio 3 - Controllo della congestione TCP	7
1.2 Esame - 05/07/2013	8
1.2.1 Domande sulla teoria	8
1.2.2 Esercizio 1 - CSMA persistent	9
1.2.3 Esercizio 2 - Subnetting e tabella di routing	9
1.2.4 Esercizio 3 - Controllo della congestione TCP	9
1.3 Esame - 25/02/2014	10
1.3.1 Domande sulla teoria	10
1.3.2 Esercizio 1 - ALOHA	11
1.3.3 Esercizio 2 - Subnetting e tabella di routing	11
1.3.4 Esercizio 3 - Controllo della congestione TCP + variante	11
1.4 Esame - 07/01/2019	12
1.4.1 Domande sulla teoria	12
1.4.2 Esercizio 1 - CSMA persistent	13
1.4.3 Esercizio 2 - Subnetting e tabella di routing	13
1.4.4 Esercizio 3 - Controllo della congestione TCP	13
1.5 Esame - 07/01/2019	14
1.5.1 Domande sulla teoria	14
1.5.2 Esercizio 1 - CSMA persistent	14
1.5.3 Esercizio 2 - Subnetting e tabella di routing	14
1.5.4 Esercizio 3 - Controllo della congestione TCP	14
1.6 Esame - 05/02/2019	15
1.6.1 Domande sulla teoria	15
1.6.2 Esercizio 1 - CSMA persistent	15
1.6.3 Esercizio 2 - Subnetting e tabella di routing	15
1.6.4 Esercizio 3 - Controllo della congestione TCP	15
1.7 Esame - 18/02/2020	16
1.7.1 Domande sulla teoria	16
1.7.2 Esercizio 1 - CSMA persistent	16
1.7.3 Esercizio 2 - Subnetting e tabella di routing	16
1.7.4 Esercizio 3 - Controllo della congestione TCP	16
1.8 Esame - 10/01/2022	17
1.8.1 Domande sulla teoria	17
1.8.2 Esercizio 1 - CSMA persistent	17
1.8.3 Esercizio 2 - Subnetting e tabella di routing	17
1.8.4 Esercizio 3 - Controllo della congestione TCP	17
1.9 Esame - 01/02/2022	18
1.9.1 Domande sulla teoria	18
1.9.2 Esercizio 1 - CSMA persistent	18
1.9.3 Esercizio 2 - Subnetting e tabella di routing	18
1.9.4 Esercizio 3 - Controllo della congestione TCP	18

1.10	Esame - 22/02/2022	19
1.10.1	Domande sulla teoria	19
1.10.2	Esercizio 1 - CSMA persistent	19
1.10.3	Esercizio 2 - Subnetting e tabella di routing	19
1.10.4	Esercizio 3 - Controllo della congestione TCP	19
1.11	Esame - 18/07/2022	20
1.11.1	Domande sulla teoria	20
1.11.2	Esercizio 1 - CSMA persistent	20
1.11.3	Esercizio 2 - Subnetting e tabella di routing	20
1.11.4	Esercizio 3 - Controllo della congestione TCP	20
1.12	Esame - 27/09/2022	21
1.12.1	Domande sulla teoria	21
1.12.2	Esercizio 1 - CSMA persistent	21
1.12.3	Esercizio 2 - Subnetting e tabella di routing	21
1.12.4	Esercizio 3 - Controllo della congestione TCP	21
1.13	Esame - 09/01/2023	22
1.13.1	Domande sulla teoria	22
1.13.2	Esercizio 1 - CSMA persistent	22
1.13.3	Esercizio 2 - Subnetting e tabella di routing	22
1.13.4	Esercizio 3 - Controllo della congestione TCP	22
2	Indice per ogni tipologia di domanda	23
3	Indice per ogni tipologia d'esercizio	23

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 ACK**, 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.

- 1.1.2 Esercizio 1 - CSMA persistent
- 1.1.3 Esercizio 2 - Subnetting e tabella di routing
- 1.1.4 Esercizio 3 - Controllo della congestione TCP

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 atteso 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).

Ogniquale volta 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.

- 1.4.2 Esercizio 1 - CSMA persistent
- 1.4.3 Esercizio 2 - Subnetting e tabella di routing
- 1.4.4 Esercizio 3 - Controllo della congestione TCP

1.5 Esame - 07/01/2019

1.5.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1.5.2 Esercizio 1 - CSMA persistent

1.5.3 Esercizio 2 - Subnetting e tabella di routing

1.5.4 Esercizio 3 - Controllo della congestione TCP

1.6 Esame - 05/02/2019

1.6.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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 - 18/02/2020

1.7.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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 - 10/01/2022

1.8.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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 - 01/02/2022

1.9.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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 - 22/02/2022

1.10.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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 - 18/07/2022

1.11.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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 - 27/09/2022

1.12.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

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

1.13 Esame - 09/01/2023

1.13.1 Domande sulla teoria

Le domande di teoria sono le seguenti:

1.13.2 Esercizio 1 - CSMA persistent

1.13.3 Esercizio 2 - Subnetting e tabella di routing

1.13.4 Esercizio 3 - Controllo della congestione TCP

- 2 **Indice per ogni tipologia di domanda**
- 3 **Indice per ogni tipologia d'esercizio**