

1. L'algoritmo di Slow Start è usato da TCP Reno [V/F]

2. Il throughput può essere maggiore del bitrate [V/F]

3. Il protocollo UDP usa l'ACK per riscontrare i datagrammi [V/F]

4. La funzionalità di flow control adatta la finestra di invio del mittente per non sovraccaricare il destinatario [V/F]

5. Il protocollo TCP fornisce la funzionalità controllo della congestione [V/F]

6. Il ritardo di accodamento è il tempo che serve a un bit per attraversare il mezzo di trasmissione [V/F]

7. Il valore del tempo di timeout dopo il quale considerare un pacchetto perso dipende dalla velocità di trasmissione [V/F]

8. UDP implementa un handshake a due vie invece che tre [V/F]

9. Il protocollo FTP usa il protocollo di trasporto TCP [V/F]

10. Il protocollo Go-Back-N usa l'ACK cumulativo [V/F]

1. vero * 2. falso, e' limitato dal bitrate 3. falso 4. vero 5. vero

6. falso 7. falso, dipende dalla velocità di propagazione

8. falso 9. vero 10. vero

* domanda non in programma per l'esonero del 09/04/2024

Esercizio 1 (4 punti)

Dato il seguente esempio di messaggio:

```
GET /usr/users/doc HTTP/1.1
```

```
Host: www.esamedireti.it
```

```
Connection: close
```

```
User-agent: Mozilla/4.0
```

```
Content-type: gif/jpeg
```

```
Content-encoding: MIME-version 1.0
```

```
If-Modified-Since: Wed, 25 Jan 2021 09:23:24
```

Rispondere alle seguenti domande:

1. Spiegare brevemente il ruolo ed effetto di ogni riga del messaggio (scrivere la descrizione direttamente accanto al messaggio sopra).

2. Che tipo di messaggio è? A quale protocollo si riferisce?

3. Spiegare brevemente il funzionamento del protocollo e quali altri protocolli dello stack sono coinvolti

1) GET /usr/users/doc HTTP/1.1 richiesta HTTP di un oggetto web
Host: www.esamedireti.it l'hostname del server soggetto della richiesta
Connection: close non mantenere aperta la connessione
User-agent: Mozilla/4.0 il browser di colui che fa la richiesta
Content-type: gif/jpeg tipo dell'oggetto web richiesto
Content-encoding: MIME-version 1.0 la codifica accettata dal richiedente
If-Modified-Since: Wed, 25 Jan 2021 09:23:24 richiesta GET condizionata (check cache)

2) È un messaggio di richiesta condizionata HTTP 1.1.

3) Il server HTTP 1.1 apre un socket TCP sulla porta 80 per accogliere le richieste. Il client invia (tramite TCP) un messaggio di richiesta HTTP 1.1, come GET, HEAD o PUT (e altri). Il server riceve la richiesta, e risponde con un messaggio HTTP 1.1 contenente una riga di stato, un header, e l'eventuale oggetto web richiesto, i messaggi HTTP 1.1 sono in formato ASCII. Un oggetto web può fare riferimento ad altri oggetti, per ogni oggetto inviato e' necessaria una nuova connessione TCP. Il segmento TCP sarà incapsulato in un datagramma IP, e poi inviato al client.

! gli esercizi 2 e 3 sono assenti in quanto duplicati e già svolti in altri fogli di esercizi.

Un host TCP A sta ricevendo un byte stream da B. Ha già ricevuto e riscontrato i byte fino al 4000 (numero di sequenza iniziale 1). Spiegare quali azioni esegue e che tipo di messaggi invia l'host TCP A in seguito ai seguenti eventi:

1. A riceve un segmento di 1000 byte con numero di sequenza pari a 3001.
2. In seguito all'evento 1, A riceve un segmento di 1000 byte con numero di sequenza pari a 6001.
3. In seguito all'evento 2, A riceve un segmento di 1000 byte con numero di sequenza pari a 5001.
4. In seguito all'evento 3, A riceve un segmento di 1000 byte con numero di sequenza 4001.

1. pacchetto già ricevuto, ignora ed invia ACK 4001.

buffer = []

2. pacchetto fuori ordine, byte [6001-7000] nel buffer, invio ACK 4001 (duplicato).

buffer = [6001..., 7000]

3. pacchetto fuori ordine, byte [5001-6000] nel buffer, invio ACK 4001 (duplicato).

buffer = [5001..., 7000]

4. gap riempito, byte [4001-7000] inviati al livello applicativo, invio ACK 7001 (cumulativo)

buffer = [4001..., 7000] \Rightarrow elabora \Rightarrow buffer = []