REPORT HOMEWORK 2 – 2042158 RUCCI

Topologia di rete:

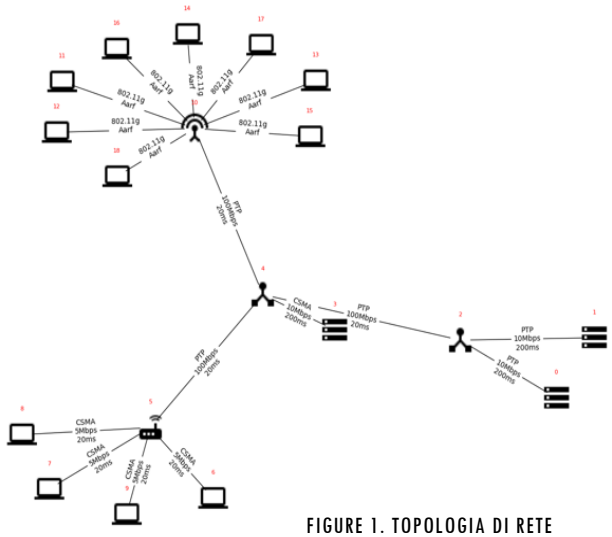

FIGURE 1. TOPOLOGIA DI RETE

- UDP burst traffic of 1784 B for each packet starting at 0.74 s
  Sender: Node 17 Receiver: Server 0
- UDP burst traffic of 1614 B for each packet starting at 3.59 s
  Sender: Node 12 Receiver: Server 1
- UDP burst traffic of 1536 B for each packet starting at 3.25 s
  Sender: Node 9 Receiver: Server 0
- Wi-Fi operating in Ad Hoc mode, stationary devices, no random walk
- UDP Echo application with Client 15 and Server 3
- Size of packet: 1319 Bytes
- Periodicity: 20ms
- MaxPackets: 250

Descrizione progettuale:

La rete è dotata di diversi metodi di comunicazione, tra cui Wi-fi, CSMA e collegamenti PTP. La sua gestione prevede una suddivisione oculata in diverse LAN, ciascuna delle quali è stata assegnata con maschere di rete adeguate al fine di minimizzare l'utilizzo degli indirizzi IP. Per la LAN Wi-fi, è stata utilizzata una subnet mask di /28, permettendo così di contenere fino a 9 host. Lo stesso principio è stato applicato alla LAN composta da 5 dispositivi connessi tramite CSMA. Infine, per gli altri collegamenti, come quelli PTP e il collegamento CSMA tra il server 3 e il router 4, è stata adottata una subnet mask di /30, poiché coinvolgono solamente 2 dispositivi.

Topologia e ritardi di rete:

1) ***Individuare le varie topologie note che compongono la rete.***

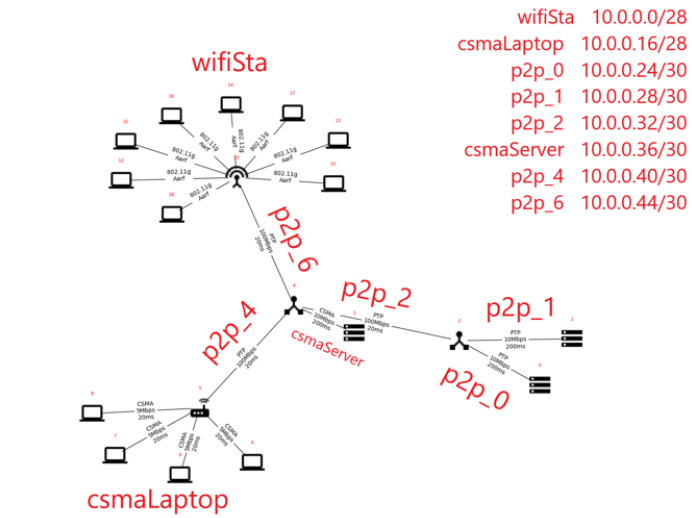

FIGURE 2. DENOMINAZIONE DELLE RETI

Sono individuabili 8 LAN così come mostrate in Fig.2, in particolare la topologia è suddivisa in 2 sotto-topologie:

- Topologia *peer-to-peer* : p2p_0, p2p-1, p2p_2, csmaServer, p2p-4, p2p_6;
- Topologia a *stella*: wifiSta, csmaLaptop;

2) Ricostruzione del percorso dei pacchetti attraverso la rete di tutti i flussi simulati (no Echo Application) usando Wireshark ed evidenziando i filtri per isolare i singoli flussi dello strato di trasporto tra le tracce.

Abbiamo analizzato i file pcap generati tramite Wireshark, aprendo i file relativi alle varie interfacce e cercando di isolare i singoli flussi attraverso il filtro "udp.stream eq" e riconoscendo i nodi attraverso gli indirizzi IP precedentemente assegnati alle LAN.

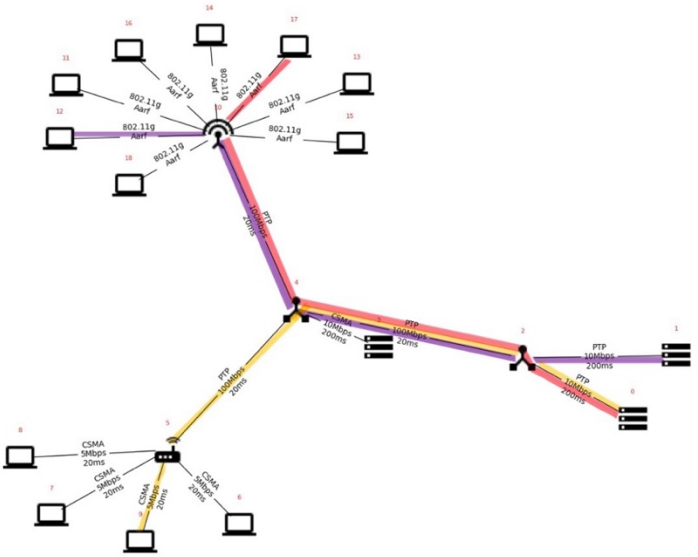

FIGURE 3. RAPPRESENTAZIONE DEI FLUSSI DI DATI

Tale analisi ci ha permesso di individuare un'interruzione di due flussi, quello tra il nodo 17 e server 0, e quello tra il nodo 12 e server 1; in particolare abbiamo notato che sull'interfaccia di uscita dal router Wi-fi, non c'è traccia di nessun pacchetto proveniente da questi host. Ipotizzando un congestionamento della rete e quindi un ritardo di inoltro di tali pacchetti, abbiamo prolungato, per fini di studio, la durata del main a 30s osservando, come atteso, la ricezione da parte dei server dei flussi "rallentati". I pacchetti del flusso tra nodo 9 invece giungono a destinazione entro la fine della simulazione.

In Fig.3 abbiamo rappresentato i flussi come segue

- Flusso Node 17 -> Server 0 in rosso;
- Flusso Node 12 -> Server 1 in viola;
- Flusso Node 9 -> Server 0 in giallo;

Infine abbiamo utilizzato il filtro "ip.src== (indirizzo nodo)" per isolare i singoli flussi.

3) Vi sono dei bottlenecks nella rete? Se si, individuare gli eventuali link e discutere eventuali contromisure e soluzioni

Partendo dal flusso tra nodo 17 e server 0, il bottleneck è il collegamento con la velocità di trasmissione minore PTP con il router 2 (10 Mbps). Analizzando gli altri flussi, per il nodo 15 e server 3, il bottleneck è nel collegamento CSMA tra il server 3 e il nodo 4 (10 Mbps). Nel flusso tra il nodo 9 e server 0, il bottleneck è nel collegamento CSMA tra il nodo 9 e il router 5 (5 Mbps). Infine, nel flusso tra il nodo 12 e server 1, il bottleneck è il collegamento PTP tra il server 1 e il router 2 (10 Mbps).
Si potrebbe considerare l'aumento delle velocità di trasmissione dove c'è un significativo decremento rispetto al resto della rete o la riduzione del traffico totale scambiato. Nel nostro caso, i flussi tra i nodi 12 e server 1 e 17 e server 0 subiscono un notevole ritardo, probabilmente dovuto alla grande quantità di traffico generato dal nodo 15, che satura il canale di comunicazione.

4) Calcolare il throughput medio del flusso di strato di trasporto (no Echo Application) a tempo t=2.0s.

Per il calcolo del throughput medio nell'intervallo [0,2]s, ci aspettavamo la presenza di traffico proveniente esclusivamente dal nodo 17, ma dall'analisi dei file pcap è risultato che tali pacchetti risultano presenti in un periodo di tempo successivo ai 2 secondi. I filtri utilizzati per l'analisi sono stati "*frame.time <= "1970-01-01 01:00:02.000000000+0100" && !(ip.src==10.0.0.6)*", dove 10.0.0.6 è l'indirizzo IP del nodo 15 (ECHO Application).Successivamente, usando le statistiche fornite direttamente da Wireshark sulla dimensione dei pacchetti (media) su ogni file pcap, abbiamo moltiplicato tale dimensione per il numero totali di pacchetti: *(10314,97)+(364)=1733,91 b* .
A questo punto il throughput medio si ottiene dividendo il calcolo precedente per il tempo impiegato : *tMedio= bit totali/2*60s=866,95 b/s*

5) *Calcolare il throughput medio del flusso di strato di trasporto (no Echo Application) a tempo t=5.0s. Commentare eventuali cambiamenti rispetto alla domanda precedente.*

Il calcolo procede secondo il metodo precedentemente descritto, con l'unica variazione che, a differenza del punto precedente, ora tutti i nodi stanno trasmettendo pacchetti, e risultano presenti pacchetti su quasi tutte le interfacce dove si prevedeva la presenza di flusso di dati. Ciò suggerisce un aumento dei bit totali e, di conseguenza, un incremento del throughput. Le aspettative sono confermate dal seguente calcolo:
bitTotali=*(343*229,78)+(7*784)+(3*112,67)+(3*1036)+(6*664,3)+(9*88,22)+(2*918)+(2*918)+(4*1148)+(44*808,27)=136338,21 b*
throughput= *27267,6 b/s*

Modalità Ad Hoc per lo strato Wi-Fi:

1. ***Tutti i frame ricevono l'acknowledgement? Spiegare perché.***

   Dalle analisi del file pcap *task-10.1* (interfaccia del router, nodo q0, tra sé stesso e gli host), risulta che non tutti i pacchetti immesi nella rete Wi-Fi vengano confermati da un ACK.
   Attraverso l'applicazione del filtro "*wlan.fa.retry==True*" è concessa la visualizzazione dei soli pacchetti che presentano la flag "*Retry*" appartenente al protocollo *IEE 820.11* posta uguale a 1, che indica che tale pacchetto è la ritrasmissione di un precedente pacchetto che non è stato confermato come ricevuto correttamente, e di conseguenza che non ha ricevuto un *ACK*.

2. ***Vi sono delle collisioni nella rete? Spiegare perché. Come siete arrivati a questa conclusione?***

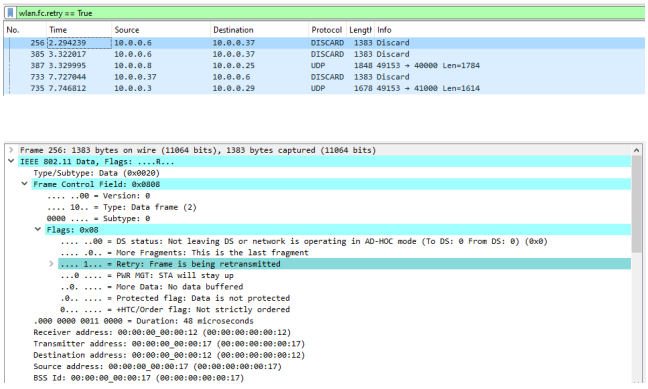

FIGURE 4. FLAG RETRY (*WLAN.FA.RETRY*)

Analizzando il medesimo file pcap menzionato nella domanda precedente e utilizzando la stessa flag impiegata nei pacchetti Wi-Fi per indicare una trategica della procedura CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Questa procedura è progettata per ridurre le collisioni e migliorare l'affidabilità delle comunicazioni wireless.

Nel contesto della ritrasmissione, la suddetta flag viene attivata, identificando così una collisione causata da un eccessivo flusso di dati immessi contemporaneamente nella rete durante il tempo disponibile per la simulazione.

3. ***Calcolare l'overhead complessivo della Echo Application.***

   Abbiamo analizzato il file pcap relativo al nodo 10 sull'interfaccia 1, applicando un filtro (*ip.src==10.0.0.6)* per visualizzare solamente i pacchetti provenienti dal nodo 17. Considerando prima i due frame scambiati tramite il protocollo ARP, i quali presentano una dimensione complessiva di 128 byte (64 byte ciascuno), a questa quantità, abbiamo aggiunto l'overhead totale derivante dai 250 pacchetti UDP. Per determinare l'overhead specifico di ciascun singolo pacchetto UDP, abbiamo sottratto dalla dimensione totale del pacchetto (1383 bytes), la quale includeva le varie intestazioni derivate dall'incapsulamento nei vari livelli, la dimensione del payload (1319 bytes come da specifica), risultando in un overhead di 64 bytes per ogni pacchetto.

   Dunque per la trasmissione di un singolo pacchetto avremo:
   128 bytes + 64 bytes = 192 bytes.

   Pertanto, la somma totale si attesta a
   *128 bytes + 64 bytes * 250 pacchetti = 16128 byte*.

4. ***Forzare l'uso di RTS/CTS nella rete per i pacchetti di strato applicativo, ci sono delle collisioni adesso? Spiegare il perché.***

   Dall'analisi del file pcap, è emerso che non si verificano collisioni durante la trasmissione dei pacchetti UDP. In particolare, prima dell'invio di ciascun singolo pacchetto UDP, viene sempre trasmessa una richiesta di invio RTS (Request to Send). Successivamente, viene mandato un messaggio CTS (Clear to Send), indicando che il canale di comunicazione è libero. Questo pattern denota che la comunicazione avviene in modo coordinato e senza interferenze, contribuendo così a evitare collisioni durante la trasmissione dei pacchetti.

5. Calcolare l'overhead complessivo della Echo Application forzando RTS CTS. Cosa cambia?

Esaminando il medesimo file pcap e con le stesse modalità del punto 3, si nota che le richieste ARP mantengono costante la dimensione in byte (128 byte complessivi) e anche il calcolo risulta simile al precedente.
Tuttavia, per ciascun pacchetto, il nodo 17 invia una richiesta di invio RTS (Request to Send) e riceve un consenso di invio CTS (Clear to Send), i quali presentano dimensioni rispettivamente di 20 byte e 14 byte.

La somma totale dei byte per ogni singolo pacchetto è quindi calcolata come segue:
(128 + 64 + 20 + 14 = 226 byte.

Estendendo questo calcolo per l'intero flusso di comunicazione, otteniamo la somma cumulativa dei byte, considerando 250 pacchetti:
*128 + 64 * 250 + 250 * (20 + 14) = 24628 byte.*

Quello che cambia è che forzando RTS/CTS ,stiamo aggiungendo bytes durante la trasmissione e dunque anche l'overhead ne risente.