

TCP VEGAS, WESTWOOD+ E VENO, GRAFICAMENTE

1. Introduzione	2
2. TCP Vegas, Westwood+ e Veno	3
3. Simulazione	3
3.1 ns-3	3
3.2 Implementazione	4
4. Risultati	6
5. Conclusioni	9

1. Introduzione

Il protocollo TCP (Transmission Control Protocol), secondo Wikipedia, “è un protocollo di rete a pacchetto di livello di trasporto, appartenente alla suite di protocolli Internet, che si occupa di controllo della trasmissione ovvero rendere affidabile la comunicazione dati in rete tra mittente e destinatario.”

Sono state sviluppate molte versioni nel corso degli anni e ognuna di esse gestisce la congestione della rete e la perdita di pacchetti durante una trasmissione in modo diverso.

In questa relazione saranno mostrati dei grafici che mostrano il comportamento di tre protocolli in particolare: TCP Vegas, TCP Westwood+ e TCP VenO.

Per la creazione dei grafici è stato utilizzato NS3, un simulatore di reti: al fine di una migliore comprensione del progetto saranno descritte le scelte implementative più rilevanti.

2. TCP Vegas, Westwood+ e Veno

TCP Vegas implementa un algoritmo di controllo della congestione basato sull'RTT che tenta di prevenire la perdita di pacchetti. Campiona continuamente l'RTT e calcola la differenza tra il throughput effettivo e quello stimato. Aumenta o decrementa il valore della finestra di congestione in base a questo valore, confrontandolo con due valori soglia.

TCP Westwood+ utilizza il paradigma di controllo della congestione AIAD (Additive Increase/Adaptive Decrease). Quando si verifica una perdita, invece di dimezzare il valore della congestion window come nelle versioni precedenti, questo protocollo cerca di stimare la larghezza di banda della rete e utilizza il valore stimato (BWE) per regolare quello della congestion window.

TCP Veno migliora l'algoritmo di Reno per gestire in modo più efficace la perdita casuale di pacchetti nelle reti utilizzando il metodo di Vegas nella stima del collo di bottiglia della rete per distinguere perdite random di pacchetti da una rete congestionata. Inoltre utilizza un sistema AIMD (Additive Increase/Multiplicative Decrease) modificato rispetto a quello di Reno.

Si possono trovare maggiori informazioni a riguardo qui: <https://www.nsnam.org/docs/models/html/tcp.html>

3. Simulazione

3.1 ns-3

ns-3 è definito nel sito ufficiale come “un simulatore di rete a eventi discreti, destinato principalmente alla ricerca e all'uso didattico. ns-3 è un software gratuito, concesso in licenza con la licenza GNU GPLv2 ed è disponibile pubblicamente per ricerca, sviluppo e utilizzo. L'obiettivo del progetto ns-3 è quello di sviluppare un ambiente di simulazione aperto e per la ricerca sulle reti”.

Inoltre, sempre secondo il sito ufficiale, “l'infrastruttura software di ns-3 permette lo sviluppo di modelli di simulazione sufficientemente realistici da consentire l'utilizzo di ns-3 come emulatore di rete in tempo reale, interconnesso con il mondo reale e che consente di riutilizzare molte implementazioni di protocolli del mondo reale esistenti, all'interno di ns-3.”

3.2 Implementazione

La rete implementata è composta da due nodi, uno che svolge il ruolo di client e uno che svolge il ruolo di server, connessi tramite una connessione punto punto. Al momento dell'installazione della pila protocollare, il tipo di protocollo TCP utilizzato cambia in base alla versione presa in esame. La scelta di questo tipo di rete deriva dagli obiettivi del progetto, avente come focus il confronto del comportamento dei protocolli TCP citati precedentemente di fronte a più perdite. Risulta superfluo o controproducente simulare reti più complesse, poiché la lente d'ingrandimento deve essere posta sulla relazione che c'è tra i pacchetti trasmessi e gli ACK ricevuti da un nodo e il valore della congestion window e della soglia di slow start dello stesso nodo al medesimo istante, nello specifico quando si registrano delle perdite. Un canale punto punto, invece che wireless, garantisce un ambiente più controllato. Per la generazione delle perdite è stato assegnato un modello di errore in ricezione al nodo server, che permette di specificare in percentuale un tasso di perdita.

Per tracciare i loro valori sono state utilizzate delle funzioni di callback connesse alla socket del client, una per tipo di dato, che sono chiamate ogni qual volta avviene l'evento a loro associato: la trasmissione di un pacchetto, la ricezione di un ack, l'aggiornamento della congestion window o della soglia di slow start.

I dati prodotti sono inseriti di volta in volta in una struttura dati dedicata, una per tipo di dato. In seguito questi sono inseriti in diversi file csv: in tal modo vengono prodotti quattro dataset differenti, sempre uno per tipo di dato.

Infine, utilizzando i file csv, viene prodotto un grafico con i dati ottenuti.

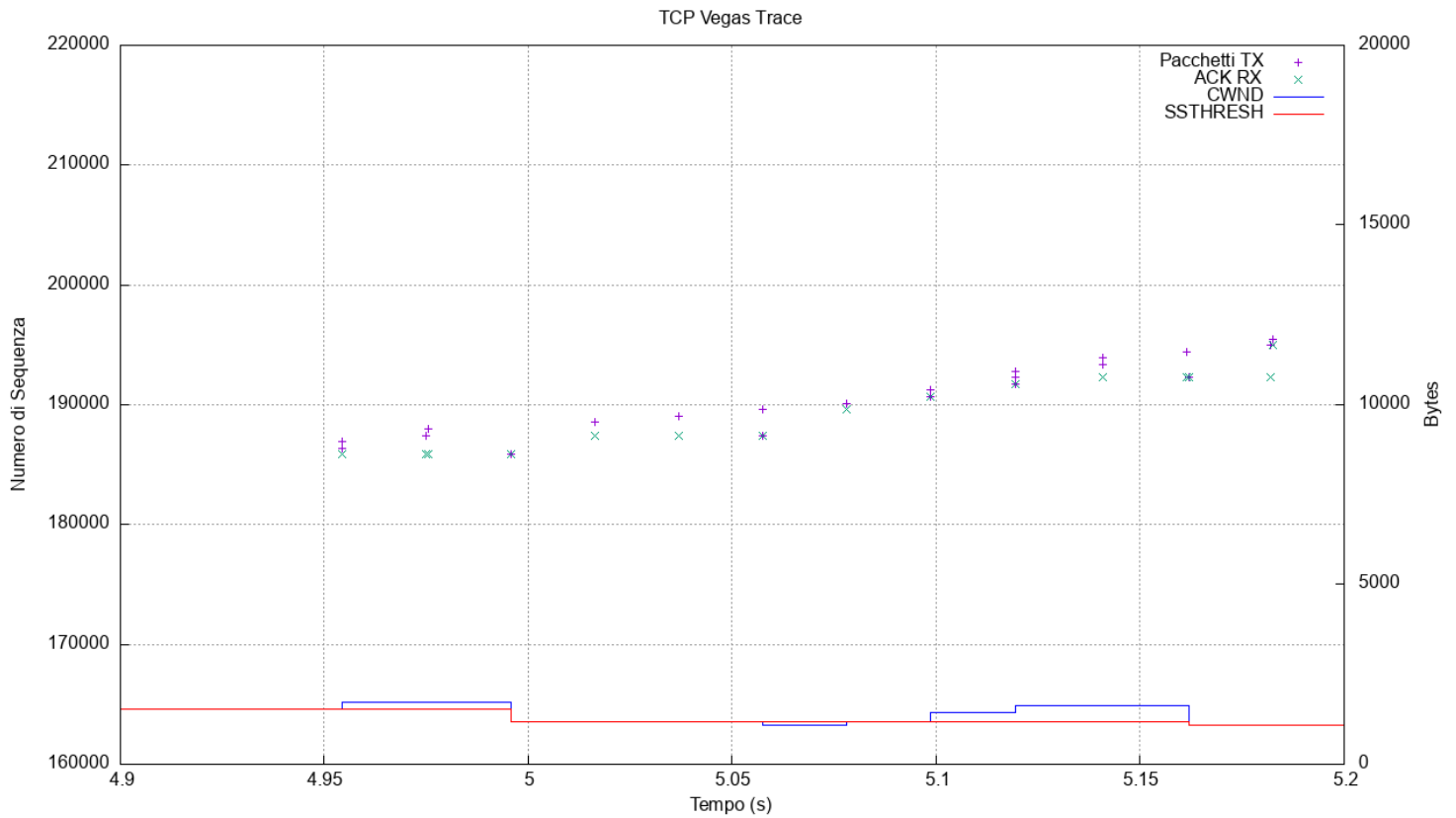
Per il plotting dei dati è stato utilizzato gnuplot.

Per la realizzazione del grafico risulta rilevante la scelta di utilizzare assi multipli. In particolare, viene utilizzato un solo asse x e due assi y: i pacchetti trasmessi e gli ack ricevuti hanno come riferimento l'asse x e l'asse y1 (sinistra); il valore della congestion window e della soglia di slow start hanno come riferimento l'asse x e l'asse y2 (destra). Per rappresentare al meglio tutti i cambiamenti della congestion window e della soglia di slow start, anche i più piccoli, il loro valore viene rappresentato in bytes.

Di seguito la tabella dei valori utilizzati per la realizzazione della simulazione, per comprensione e riproducibilità dell'esperimento:

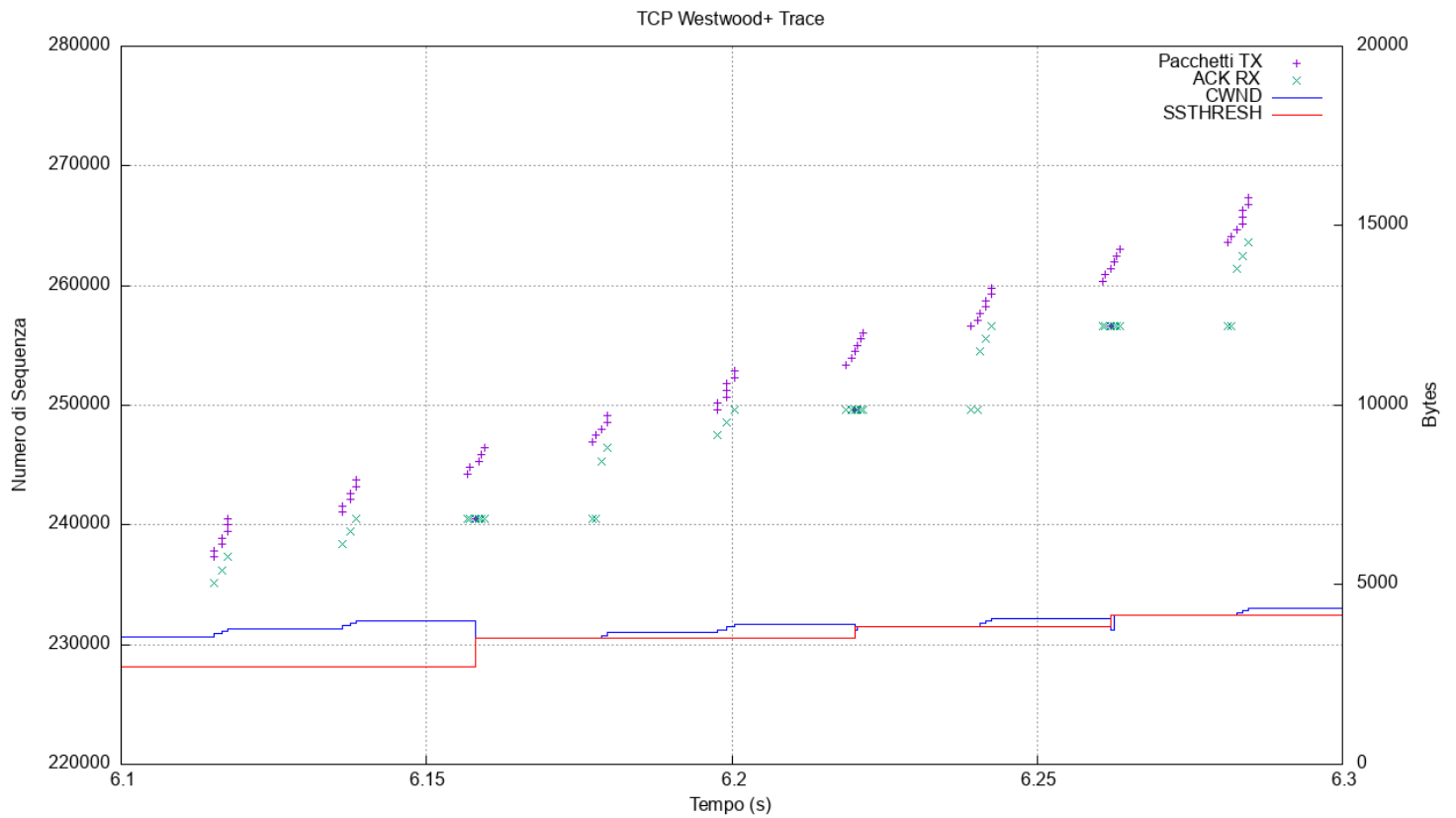
Numero di nodi	2
Tipo di canale	Punto-punto
Data Rate del canale	10Mbps
Delay del canale	10ms
Tasso di perdita del server in ricezione	1%
Data Rate del client	500Kbps (Default)
Grandezza dei pacchetti	512 Bytes (Default)
Durata totale della simulazione	20s
Tempo d'avvio del client	1s
Tempo di stop del client	20s
Tempo d'avvio del server	0s
Tempo di stop del server	20s
Tempo di avvio del tracciamento	1.001s

4. Risultati



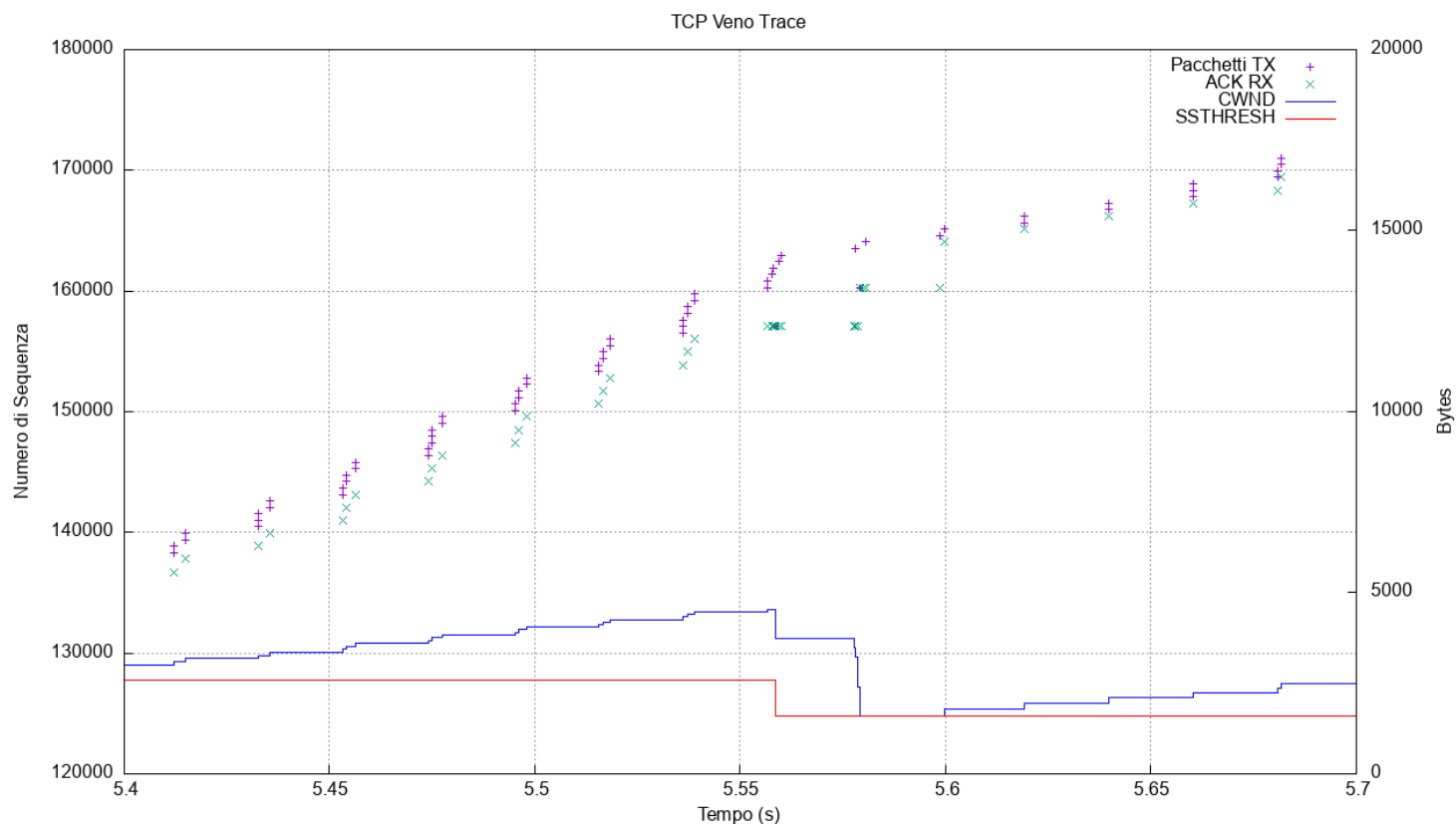
Didascalìa

Come si può notare, intorno ai 4.99s e intorno ai 5.16s si verifica una perdita dopo la registrazione di 3 ack duplicati e il segmento viene immediatamente ritrasmesso. TCP Vegas agisce abbassando il valore della cwnd e reimpostando il valore della ssthresh. Al tempo 5.06 invece abbassa il valore della cwnd ma mantiene stabile quello della ssthresh.



Didascalia

Qui si può osservare il comportamento di TCP Westwood+: ogni volta che si verifica una perdita (intorno ai 6.16s, 6.23s e 6.26s) ricalcola sia il valore della cwnd sia della ssthresh, basandosi sulla stima della banda disponibile, superando il dimezzamento della cwnd.



Didascalia

In questo grafico è possibile esaminare il comportamento di TCP Veno, dunque del suo AIMD. Consideriamo la perdita che si registra al tempo 5.56: riconosce che si tratta di una perdita random e non dovuta a congestione, così riduce il valore della cwnd di solo 1/5.

5. Conclusioni

Il progetto consisteva nel produrre grafici che permettessero di analizzare il comportamento di tre versioni del protocollo TCP, cioè TCP Vegas, TCP Westwood+ e TCP Veno mostrando il cambiamento del valore della congestion window e della soglia di slow start in relazione alle perdite di pacchetti durante una trasmissione tra un client ed un server. È stata così sviluppata una simulazione usando il simulatore di reti ns3 con cui sono stati prodotti i dati necessari per la realizzazione dei grafici, con un livello di dettaglio tale da poter distinguere i singoli pacchetti trasmessi e gli ack ricevuti dal client.

I risultati ottenuti in questo progetto, pur presentando alcune limitazioni dovute alle semplificazioni del modello di rete adottato, hanno permesso di evidenziare chiaramente le differenze fondamentali nel comportamento dei protocolli TCP Vegas, Westwood+ e Veno nel momento in cui si verificano delle perdite di pacchetti.

In particolare, è emerso come TCP Vegas, (utilizzando i valori predefiniti di α e β) mostri una maggiore cautela nella gestione della congestion window, garantendo un throughput più basso ma più stabile rispetto alle altre due versioni del protocollo. TCP Westwood+ reagisce in modo ottimale alle perdite e, considerando le capacità del canale e dei nodi, riesce a garantire un throughput maggiore. Il comportamento di TCP Veno invece presenta delle caratteristiche che lo rendono il più “equilibrato”: garantisce un throughput maggiore rispetto a TCP Vegas ma reagisce più bruscamente alle perdite risultando meno stabile, anche rispetto a TCP Westwood+.

Anche se TCP Westwood+ sembra essere il protocollo migliore, questi risultati, pur necessitando di ulteriori approfondimenti attraverso simulazioni più complesse e l'inclusione di altri fattori come la latenza e la larghezza di banda variabile, suggeriscono come la scelta del protocollo TCP più adatto dipenda strettamente dalle caratteristiche specifiche della rete e dalle esigenze dell'applicazione.