

Tecnologie di Infrastrutture di Reti proposte per Gruppi di Laboratorio

Prof: Maurizio Casoni

April 17, 2018

Contatti

Professor Maurizio Casoni: maurizio.casoni@unimore.it

Assistant Professor Carlo Augusto Grazia: carloaugusto.grazia@unimore.it

Post Doc Martin Klapez: martin.klapez@unimore.it

Proposte:

1 Test prestazionale del controller SDN Ryu (Python)

Lo scopo di questo lavoro è quello di creare un testbed capace di valutare le prestazioni del controller SDN Ryu, in particolare latenza e throughput.

Ai candidati verrà chiesto di creare il suddetto testbed in modo che sia replicabile, con quindi le istruzioni di installazione e replicazione dei risultati.

Verranno fornite le specifiche di una semplice topologia di rete da emulare con Mininet, e del funzionamento atteso del controller SDN.

Workflow proposto:

1. Breve studio di SDN, Mininet e Ryu (Python).
2. Installazione e configurazione di un SO Linux su macchina virtuale o fisica.
3. Creazione della rete virtuale attraverso Mininet.
4. Programmazione del controller SDN con Ryu (Python).
5. Test latenza.

6. Test throughput.

2 Realizzazione prototipo di trasmissione concorrente con P4 e Mininet

Prendendo spunto da una soluzione già esistente sviluppata con Mininet (emulatore di rete) e Floodlight (SDN), ai candidati verrà chiesto di replicarla con Mininet e P4 (linguaggio ad alto livello che permette via software di definire in tutto e per tutto il comportamento di uno switch compatibile).

Workflow proposto:

1. Breve studio (panoramica) di Software-Defined Networking (SDN) e linguaggi di programmazione relativi
2. Comprensione del testbed già esistente
3. Studio di P4
4. Realizzazione del testbed con Mininet e P4
5. Test concorrenza (con grafico risultati)

3 Test del tcp BBR su Linux

Ai candidati viene chiesto di studiare il nuovo algoritmo di controllo della congestione implementato da Google, chiamato Bottleneck Bandwidth Round-trip time. Alla fase di studio seguirà una fase di testing su una macchina (dei candidati o messa a disposizione in laboratorio) in cui comunicazioni di trasferimento TCP verranno istanziate con BBR e valutate sulla base di diversi parametri configurabili di una rete

Workflow proposto:

1. Buono studio di BBR e stack di rete TCP/IP su Linux
2. Configurazione di un semplice ambiente di testing in laboratorio
3. Valutazione dei risultati conseguiti

4 Studio del protocollo IEEE 802.11p su Linux

Ai candidati viene chiesto di studiare un testbed pronto che implementa una tecnologia 802.11p. Verra' presa dimestichezza con un tool per valutare le performance di una rete qualsivoglia tramite clonessioni TCP/UDP standard.

Workflow proposto:

1. Buono studio di un testbed fisico fornito
2. Creazione di script per la creazione di test
3. Creazione di script per l'elaborazione dell'output (moduli linux)
4. Valutazione dei dati

5 Studio delle tecniche di Deep Packet Inspection su Linux

Ai candidati viene proposto di studiare le tecniche attualmente esistenti su Linux per realizzare modifiche all'interno di pacchetti ricevuti su un router. Esempio di modifica in seguito a DPI il cambiamento della receiver window (rWnd) di un segmento TCP, e di ricalcolarne il checksum.

Workflow proposto:

1. Studio delle tecniche DPI esistenti, sia proprietarie che non
2. Analisi delle caratteristiche, pro e contro
3. Provare ad implementare una tecnica di rewrite della rWnd (esempio: rWnd limitata a 10 Kbyte per tutti i flussi backlogged) in uno dei tool studiati
4. Presentazione dell'analisi delle varie tecniche e, se possibile, dell'implementazione della tecnica sopra citata in un elaborato.

6 Studio del simulatore Flent: The FLExible Network Tester

Ai candidati viene proposto di studiare un tool chiamato Flent, disponibile su varie piattaforme (OSx e diverse distro Linux). Tale strumento e' usato per fare veloci test su algoritmi TCP, AQM e Scheduler. L'obiettivo sar quello di studiare, installare e testare questo tool.

Workflow proposto:

1. Studio e installazione di Flent
2. Analisi di alcuni algoritmi a scelta (TCP, AQM, Scheduler)
3. Realizzazione di test tramite Flent sugli algoritmi scelti
4. Elaborazione dei risultati ottenuti (post-processing)

7 Analisi delle TSQ, TCP Small Queues di Linux

Ai candidati viene proposto di studiare il meccanismo TSQ. Si tratta di una coda "di sistema" che Linux utilizza per limitare il numero di segmenti TCP in volo all'interno della macchina. Questo meccanismo e' stato introdotto da poco per evitare che diversi segmenti TCP vengano inviati e accumulati nella NIC del sender, evitando cosi' la formazione di Bufferbloat gia' nel primo stadio della comunicazione.

Workflow proposto:

1. Studio di TSQ
2. Analisi di alcuni test per modificare TSQ
3. Realizzazione di test di connessione TCP per isolare l'effetto di TSQ
4. Elaborazione dei risultati ottenuti (post-processing)