# Analisi delle prestazioni della Rete Wi-Fi del Campus X di Roma Tor Vergata

## Introduzione

Il presente documento si riferisce allo studio fatto come progetto del corso Performance Modelling of Computer Systems and Networks (9 CFU) tentuto dalla professoressa De Nitto Personè. Il team di sviluppo è composto da 3 persone: G. La Delfa, G. Tummolo, D. Verde. L’idea iniziale era quella di sviluppare un qualcosa di utile anche nel mondo reale e proporre delle soluzioni per migliorare. Dato il fatto che uno dei componenti vive al Campus X e le numerose voci sulle presunte scarse prestazioni della rete wifi del campus si è deciso di procedere analizzando tale progetto.

## Traccia del Progetto

****

Blocco 1

Edificio 4D

Access Points (APs)

Il Campus X di Roma Tor Vergata è il più grande centro di residenze universitarie in Italia; esso sorge in un’area adiacente alla maggior parte delle Macroaree dell’Università ed è attualmente sede del CLA. Al suo interno si svolgono regolarmente diversi eventi riguardanti la comunità universitaria, tra cui anche le lezioni di alcuni corsi di laurea. Esso comprende 5 Blocchi di Edifici (nominati rispettivamente Blocco 1, …, Blocco 5) ciascuno dei quali comprende a sua volta 3 o 4 palazzi (nominati con le lettere A,B,C,D – ad esempio, ci si riferisce ad un palazzo con la sigla 4D). Ogni palazzo è costituito da 3 o 4 piani (a cui vi si aggiunge il piano terra). L’accesso ad internet per i residenti nella struttura è garantito attraverso una rete wireless che comprende 4 Access Point (AP) per piano, posizionati agli angoli dei piani di ogni edificio. Ciascun utente quindi si connette all’ AP più vicino utilizzando un dispositivo Wi-Fi e può navigare sul web (non vi sono limiti sul numero di dispositivi che ogni utente può connettere alla rete). Ciascun Access Point è collegato mediante un cavo Ethernet (RJ45) ad uno Switch a 48 porte posto al piano terra di ogni edificio. Lo switch provvede a fondere i flussi provenienti dai diversi piani del palazzo e ad inoltrare correttamente il traffico in entrata ed in uscita nella giusta direzione.

## Obiettivi dello Studio

Il Primo Step del processo di sviluppo, secondo il libro di test, consiste nel definire attentamente gli obiettivi dello studio. Tenuto conto di quanto detto durante il corso e nei file messi a disposizione dal docente, contenenti esempi di progetti degli anni passati e delle linee guida di tutto ciò che il progetto deve contenere, si è deciso di definire come obiettivi dello studio i seguenti:

1. Simulare le prestazioni della rete in un edificio tramite un approccio di tipo Next-Event Simulation ed effettuare un’analisi del Transiente e dello stato Stazionario;

2. Valutare le prestazioni della rete:

2.1. Studiare come varia il tempo medio di risposta E(Ts) di un utente al variare dell’intensità e della distribuzione del traffico in ingresso;

2.2. Identificare eventuali Bottlenecks;

3. Proporre delle possibili soluzioni alternative finalizzate al miglioramento delle prestazioni della rete.

## Il Modello Concettuale (a Reti di Code)

Seguendo quanto riportato nel libro di testo, definiti gli obiettivi è necessario costruire prima un modello concettuale. Analizzando attentamente quanto scritto nella traccia, un possibile modello a reti di code potrebbe essere il seguente:



Chiaramente, essendo la topologia della rete identica per ogni piano, possiamo studiare un solo piano e ricavare per simmetria i valori negli altri piani. È necessario fare attenzione alle variabili di stato a questo punto: Ciascun AP e lo switch possono avere 2 stati, ovvero Libero (Idle) ed Occupato (Busy) e supponiamo che ciascun server abbia infiniti posti in coda.

## Il Modello di Specifica

Ora, occorre costruire un modello di specifica, cercando di capire quali sono i valori e le distribuzioni dei tempi di arrivo/servizio. In base a quanto riportato nel link 3 è possibile assumere:

Si assume = 512 bytes (Medium Packet Size on Internet)

Inoltre, è ragionevole assumere che gli arrivi siano random e, che per simmetria, in ciascun piano ci sia la stessa quantità di traffico:

Distribuzione dei Tempi di Interarrivo: Esponenziale (Processo di Poisson)

Inoltre, come visto durante il corso e riportato in (trova link) esistono diversi modelli di traffico e la distribuzione di pareto (bounded) può essere utile per simulare la dimensione del pacchetto. Pertanto analizzeremo sia i casi in cui i tempi di servizio siano esponenziali (per validare il modello) e poi il caso in cui la distribuzione sia bounded pareto. Nel caso di quest’ultima, occorre calcolare i valori E(S) minimo e massimo per tarare la distribuzione; studieremo il comportamento del sistema sia nel caso di alta variabilità (heavy tail – a < 1) sia nel caso di bassa variabilità del traffico ( 1 < a < 2).

Distribuzione dei Tempi di Servizio: Esponenziale, Pareto.

Infine, analizzando i Datasheets delle apparecchiature presenti nella rete, abbiamo ricavato:

1 – Caso Esponenziale

* APs UniFi UAP-AC-PRO [Max data-rate 1300 Mbps @ 2.4 GHz e 5 GHz using 3x3 MIMO]:

C = 1300 Mbps = 162,5 MB/s

Mp/s (Millions of Packets per Second)

Mp/s (Millions of Packets per Second)

* Switch Cisco MS250-48-HW [Switching Capacity 176 Gbps]

C = 176 Gbps = 22 GB/s

(Millions of Packets per Second)

Mp/s (Millions of Packets per Second)

2 – Caso Bounded Pareto:

Si assumono pacchetti con dimensioni:

minima = 64 bytes

massima = 1500 bytes

* APs:

C = 1300 Mbps = 162,5 MB/s

E(S)min = E(Z) min / C = 0.3756009615

E(S)max = E(Z) max / C = 8.756197416

* Switch:

C = 176 Gbps = 22 GB/s

E(S)min = E(Z) min / C = 0.002709302035

E(S)max = E(Z) max / C = 0.0631606037

Infine, il modello finale da simulare risulta il seguente (abbiamo eliminato i vari piani e suddiviso il traffico totale):

## Il Modello Computazionale

Si è scelto di utilizzare come linguaggio di programmazione il linguaggio C, per avere maggiore efficienza e per continuità con quanto visto durante il corso. Si è deciso di dotare il programma di diverse strutture quali clock, sum e statistiche di output. File nsssn.c. Descrizioni delle funzioni.

## V&V – Verifica e Validazione

Verifica: Testing Estensivo di funzioni

Validazione: Rete di Jackson e valori teorici.

## Analisi Transiente

Qui va la parte dell’analisi del transiente (Replicazioni)

## Analisi Stazionaria

Qui va la parte dell’analisi dello stato stazionario (Batch Means)

## Identificazione dei Bottlenecks

L’identificazione dei bottlenecks (o colli di bottiglia) avviene osservando se all’aumentare del traffico in ingresso al sistema λ esistono alcuni dei devices nella rete la cui utilizzazione tende ad 1 (ovvero al 100%). In tal caso, il valore del traffico per cui questo fenomeno avviene è detto “di saturazione” e si indica con , ed il devices si dice che “è un collo di bottiglia” nel sistema. Ovviamente, se tutti i devices della rete raggiungono un’utilizzazione pari a 1 contemporaneamente, allora la rete è perfettamente bilanciata e non esistono colli di bottiglia nel sistema. Nel nostro caso, aumentando l’intensità del traffico in ingresso fino al valore (misurato in pacchetti al microsecondo), osserviamo che tutti gli Access Points raggiungono un’utilizzazione che è circa del 100%:

Quindi, si può concludere dicendo che gli APs costituiscono dei colli di bottiglia e pertanto, terremo conto di questo risultato nella progettazione dell’algoritmo migliorativo per il sistema. Inoltre, è bene osservare che la variabilità nella distribuzione dei tempi di servizio influenza fortemente il valore del traffico di saturazione. Tale variabilità è modellata attraverso il parametro α della distribuzione BoundedPareto: valori di indicano un’alta variabilità, mentre valori di indicano una bassa variabilità. Infatti, eseguendo diverse volte il simulatore con un valore di α = 1.5 (bassa variabilità):

Osserviamo facilmente che, in caso di bassa variabilità della distribuzione dei tempi di servizio ed a parità di traffico in ingresso, l’utilizzazione dei devices si riduce fortemente. In altre parole, meno è variabile la distribuzione dei tempi di servizio e più performante è il comportamento della rete (ovvero maggiore è il traffico che la rete riesce a sopportare). Tuttavia, tale variabilità non influenza lo studio fatto per l’identificazione dei bottlenecks, che continuano a rimanere gli APs in ogni caso. I valori mostrati nei grafici precedenti sono riportati rispettivamente nei files bottleneck-high.txt e bottlenecks-low.txt.

## Algoritmi Migliorativi

1. Sostituire gli AP con quelli più veloci
2. Aumentare il numero di AP per piano
3. AP collegati in anello (unica coda – Multiserver)
4. Meccanismo di controllo degli accessi.

Bibliografia e Link Utili

1. Capacity Calculus (MIMO): <https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Wireless_Throughput_Calculations_and_Limitations>
2. Switching Capacity: <https://forum.huawei.com/enterprise/en/forwarding-performance-and-switching-capacity/thread/570609-861>
3. Packet Best Size: <https://www.networkworld.com/article/2300175/picking-the--best--packet-size.html>
4. Pareto Distribution: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1045.7&rep=rep1&type=pdf>