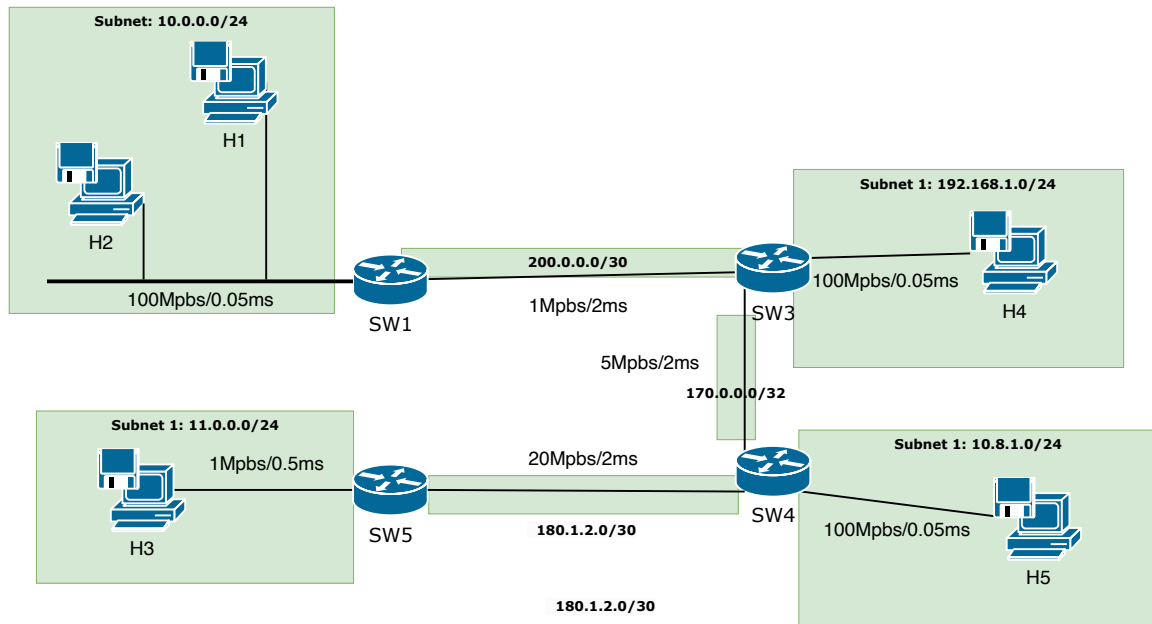


## Progetto

Progettare ed implementare un sistema di comunicazione che emuli una rete composta da switch, router e host. In particolare, si faccia uso di **Mininet** [1] per la realizzazione della rete. La topologia di rete richiesta è composta da **n. 4 nodi che operano come router L3 e n. 5 host**. I nodi di rete sono collegati tra loro come mostrato in figura:



In figura sono inoltre indicate le capacità dei link e i ritardi di propagazione.

Si consiglia di utilizzare un controller SDN per configurare gli indirizzi di rete le tabelle di flusso per ciascuno switch. (esempio: Ryu-Manger)[2] . È possibile configurare i router usando REST [3] .

Occorre sviluppare il progetto rispettando le seguenti specifiche:

1. Tutti gli host devono essere raggiungibili fra loro,
2. Gli indirizzamenti IP devono rispettare le sottoreti indicate nello schema,
3. I link emulati con Mininet devono avere Rate e Ritardo di propagazione come indicato nello schema.
4. Sul nodo H1, avviare un server http usando il framework Flask[4] che implementi un servizio di API REST per l'esecuzione di esperimenti di prestazioni sulla rete.
5. Ogni nodo della rete deve poter raggiungere il servizio su H1 ed inviare i comandi.
6. I comandi forniti nella API consistono nell'attivazione di traffico mediante tool "iperf". In particolare, è necessario che siano definite almeno i seguenti endpoint:
  - a. /start\_iperf, che avvia il traffico iperf e prende come input: *IP\_DEST*, *L4\_proto*, *src\_rate*
  - b. /stop\_iperf, che interrompe ogni traffico iperf istanziato sul client.

7. Il destinatario deve essere validato mediante il campo `src_rate`.
8. Il traffico tra client e server che può essere di tipo sia TCP che UDP validando il campo `L4_proto`.
9. Il rate di invio dei pacchetti può essere modificato impostando il campo `src_rate`.
10. È necessario che in fase di inizializzazione della rete, tutti i nodi devono avviare un server iperf, per tale ragione si suggerisce di avviare i server in fase di avvio della topologia.
11. I risultati degli esperimenti svolti devono essere salvati per ciascun host su file di log separati. Per il salvataggio su file di log sui server si suggerisce di utilizzare il seguente formato di avvio: `iperf -s -y C >> h2_log.csv &`  
Usare il formato già fornito da iperf così strutturato:  
20241128113057,10.1.1.1,5001,192.168.1.2,56178,4,0.0-12.0,1441792,957762

#### Output previsto:

- **Relazione tecnica di progetto** dettagliata che includa la descrizione dell'analisi progettuale, la descrizione e lo schema logico architetturale del sistema, l'analisi e la validazione delle specifiche richieste, le metodologie di raccolta dati e risultati ottenuti. Non includere il codice nella relazione, esso deve essere consegnato su file separati.
- **Codice sorgente del progetto** (script, eventuali file di configurazione, eventuali file di libreria, eventuali log, eventuali dipendenze).

#### Modalità di consegna dei progetti:

- Invio e-mail all'indirizzo [fabrizio.giuliano@unipa.it](mailto:fabrizio.giuliano@unipa.it), si suggerisce di specificare nell'oggetto "PROGETTO RETI DI CALCOLATORI ID-GRUPPO/COGNOME", allegando:
  - o PDF della relazione
  - o ZIP file contenente i codici sorgente.
- Nel caso di tesine di gruppo, tutti i componenti del gruppo devono essere inclusi tra i destinatari della email.

#### Link di riferimento:

- [1] <https://mininet.org/>
- [2] <https://ryu.readthedocs.io/en/latest/>
- [3] [https://osrg.github.io/ryu-book/en/html/rest\\_router.html](https://osrg.github.io/ryu-book/en/html/rest_router.html)
- [4] <https://flask.palletsprojects.com/en/stable/>
- [5] <https://github.com/Rad6/SDN-Routing-Ryu.git>