UNIVERSITÀ DEGLI STUDI DI PISA
Facoltà di Ingegneria
Corso di Laurea Specialistica in Ingegneria Informatica

Tesi di Laurea Specialistica

PROGETTAZIONE ED IMPLEMENTAZIONE DELL'INFRASTRUTTURA SERVER PER UN SISTEMA DISTRIBUITO DI SCANSIONE E MAPPATURA DELLA RETE INTERNET

Candidato
Alessandro Iaria

Relatori
Luciano Lenzini
Giovanni Stea
Matteo Maria Andreozzi

Anno accademico 2010/2011

INDICE

Indice 2
Tabella delle figure 3
Capitolo 1: Introduzione 4
Capitolo 2: Stato dell'arte 7
Capitolo 3: Struttura del Sistema 9
3.1 Struttura generale 9
3.1.1 Caratteristiche di un Client 9
3.1.2 Protocollo client – server 10
3.2 Struttura del Server 13
3.2.1 Obiettivi e funzionalità 13
3.2.2 Moduli 15
3.2.3 Dalla prima interazione al merge dei nuovi dati 17
Capitolo 4: Validazione del sistema 29
Capitolo 5: Conclusioni 37

Tabella delle figure

Figura 1: Refresh stato 11
Figura 2: Richiesta di un Job 12
Figura 3: Invio dei risultati 12
Figura 4: Struttura del Server 15
Figura 5: Elementi base di un Link 22
Figura 6: Unione di due router comunicanti 26
Figura 7: Scissione di un router 26
Figura 8: Backbone rete GARR 30
Figura 9: Scansione 1 31
Figura 10: Scansione 2 31
Figura 11: Grafo parziale 1 31
Figura 12: Scansione 3 32
Figura 13: Grafo parziale 2 32
Figura 14: Scansione 4 33
Figura 15: Grafo parziale 3 33
Figura 16: Scansione 5 33
Figura 17: Grafo parziale 4 33
Figura 18: Scansione 6 34
Figura 19: Grafo parziale 5 34
Figura 20: Grafo completo 36

Capitolo 1: Introduzione

Data la naturale evoluzione della rete Internet in continua crescita a ritmi sempre più elevati, dovuta all'esplosione del World Wide Web e dei servizi ad esso correlati, è divenuto sempre più fondamentale il bisogno di capire come la rete stesse evolvendo e secondo quali meccanismi.
Il bisogno di capire l'evoluzione della rete nasce da diversi fattori, che possono essere la ricerca di nuovi meccanismi più efficienti di gestione per il transito dei dati, ma anche studi di livello sociale, come ad esempio il livello di diffusione della rete nei vari continenti e secondo quali leggi questa si stia evolvendo.
È infatti di vitale importanza avere dei modelli affidabili della rete per poterne prevedere i suoi sviluppi futuri, che può portare a notevoli vantaggi in fase di progettazione di nuove infrastrutture e quindi ad un deployment di risorse più accurato o ad effettuare stime più accurate sulla scalabilità dei sistemi attuali e quindi prevenire sovraccarichi della rete.
La conoscenza della rete quindi presenta notevoli vantaggi sia dal punto di vista della comunità scientifica sia da quello delle aziende che lavorano nel mondo dell'Internet in quanto fornitori di servizi o di infrastrutture.
Le aziende o enti che detengono le infrastrutture di rete però, e quindi la conoscenza dettagliata delle stesse, raramente pubblicano tali informazioni.
È necessario quindi trovare dei meccanismi che, sfruttando i meccanismi insiti nella rete stessa, producano automaticamente la topologia di rete.
I progetti riguardanti la scoperta e la mappatura della rete Internet sono numerosi e agiscono a diversi livelli di dettaglio e con meccanismi diversi. I metodi principali utilizzati sono l'esame delle tabelle BGP, che permettono di inferire i collegamenti fra AS a livello logico, e l'esame della rete tramite meccanismi di traceroute, cioè l'invio attivo di sonde all'interno della rete stessa e il collezionamento dei dati generati dalle stesse, che permettono di ricostruire la rete a livello router.
L'obiettivo della tesi è quello di basarsi sui meccanismi di traceroute e prevede la progettazione e l'implementazione di un Server per un sistema distribuito che preveda sia meccanismi per scegliere opportunamente come inviare le sonde, in base a fattori opportunistici, sia la raccolta dei dati per ogni singola scansione così generati e la fusione degli stessi in un'unica mappa.

Il lavoro di generazione dei dati verrà rimandato a dei terminali mobili, principalmente smartphone o tablet, che quindi permettono di usufruire delle loro caratteristiche di nomadicità per ottenere vantage point sempre differenti per la generazione dei dati, aumentando quindi le percentuali della mappa effettivamente scoperta.
Questo è possibile grazie alla diffusione e alla facilità di distribuzione delle applicazioni su questi terminali assieme alla diffusione di tariffe di traffico di tipo flat che ne agevolano la connessione costante ad Internet. La potenza di calcolo sempre crescente di questi terminali inoltre consente l'uso di tecniche all'avanguardia per la generazione dei dati e la ripulitura degli stessi ancor prima che questi siano inviati al Server, garantendo quindi un buon livello di confidenza sui dati ricevuti.
La presenza eterogenea e non deterministica dei terminali sulla rete è un concetto chiave del lavoro, in quanto le metodologie di routing applicate all'interno della rete tendono a nascondere collegamenti fra router molto lontani dalla sorgente della scansione.
Spetta al Server quindi coordinare le scansioni sulla rete in maniera da coprire zone della rete non ancora esplorate o di particolare interesse.
Al momento, i fattori principali che determinano le modalità di invio delle sonde sono:

- Posizione geografica
- Rete a cui il terminale mobile è connesso
- Informazioni già raccolte sulla rete

Il Server si occuperà altresì di evitare di ripetere di esaminare le varie rotte, evitando quindi l'accumulo di dati non necessari. Una rotta è costituita da una coppia di AS per cui si è già effettuata una scansione da uno verso l'altro. Questa coppia è formata dall'AS cui si affaccia il terminale e dall'AS verso cui si dirige la scansione; gli AS sono ricavati a partire dagli indirizzi IP con un opportuno algoritmo di risoluzione.
Di notevole importanza all'interno della ricerca di una nuova rotta sono gli AS di tipo "stub", cioè terminali, che sono noti a priori e cui il Server presta particolare attenzione e predilige quando possibile.
Una volta eseguita la scansione, i dati raccolti vengono inviati al Server, il quale si occuperà di raccoglierli e di unirli ai dati già esistenti: questo implica l'associazione fra i dati in arrivo e quelli già esistenti, in modo da arrivare ad una mappa unica, e la risoluzione di eventuali situazioni di discordanza fra i dati.

Per risolvere le discordanze, si utilizza un metodo statistico. basato sull'ipotesi che i dati che corrispondono alla configurazione reale della rete tendono ad apparire più spesso e possono quindi essere identificati in base a scansioni successive.
Si dovrà inoltre fare attenzione ai casi particolari della rete, come i router che non rispondono ai meccanismi di traceroute, reti private, o altri elementi che possono generare delle anomalie nei dati e gestirli appositamente.
L'elaborato è organizzato secondo il seguente schema:

- **Introduzione**: presentazione del lavoro svolto e delle motivazioni
- **Stato dell'arte**: elenco dei principali progetti esistenti e differenze principali
- **Architettura Generale e struttura del Server**: descrizione della progettazione e implementazione del Server all'interno del sistema distribuito
- **Validazione del sistema**: descrizione dei risultati ottenuti tramite test all'interno della rete GARR di cui è nota la struttura fisica reale
- **Conclusioni e sviluppi futuri**

Capitolo 2: Stato dell'arte

Vedremo ora quali sono i principali progetti che si basano su traceroute per la mappatura di Internet con sistemi distribuiti e vedremo quali sono le caratteristiche principali che li distinguono dal sistema proposto nell'elaborato.

- **ROCKETFUEL**
  (*http://www.cs.washington.edu/research/networking/rocketfuel/*)
  L'obiettivo del progetto è quello di realizzare un sistema per la mappatura della rete interna ad un ISP. Il progetto è stato testato su circa 10 ISP, con una base di circa 800 terminali disposti su dei Web Server. Le ultime pubblicazioni riguardo questo progetto risalgono al 2003.
  La differenza principale è proprio nei terminali usati per effettuare le analisi, che sono mobili e non fissi. Inoltre i task assegnati ai terminali non riguardano un singolo ISP ma sono potenzialmente su scala globale.

- **CAIDA – ARK** (*http://www.caida.org/projects/ark/)*
  CAIDA (*Cooperative Association for Internet Data Analysis*) è un'organizzazione che si occupa di raccogliere, analizzare e divulgare informazioni sulla rete Internet a livello globale.
  Il progetto ARK (*Archipelago*) è uno dei progetti gestito da CAIDA e si basa sul deployment di server dedicati che si occupano di eseguire le scansioni della rete, a intervalli regolari, in base a un set pre-determinato di indirizzi IP da esaminare, scelti dividendo l'intero spazio dei possibili indirizzi IP pubblici in base ai terminali. Il numero di terminali è di circa 18 terminali per team e al momento sono presenti 3 team di raccolta. Il progetto è attivo dal 2007.
  Le principali differenze sono ovviamente le caratteristiche dei terminali, mobili e non fissi, e il sistema di scelta dei task da assegnare ad ogni terminale, determinata in base al contesto del terminale stesso. Inoltre il numero di terminali risulta piuttosto limitato, data la necessità del deployment fisico, mentre la distribuzione del software su terminali mobili è sempre più facile e permette accesso a un numero di terminali molto elevato.

- **DIMES** (*http://www.netdimes.org/*)
  Il progetto si basa sulla diffusione di un client, eseguibile su quasi tutti i PC e disponibile per vari SO, che viene distribuito gratuitamente. La scelta dei task viene effettuata in base a delle euristiche basate sulla posizione del client e altri parametri. Il progetto è attivo dal 2004.
  La differenza principale anche qui risiede nei terminali utilizzati, mobili e non fissi.

Capitolo 3: Struttura del Sistema

3.1 Struttura generale

La struttura generale si presenta sotto il paradigma **client – server**, dove i Client sono dei **dispositivi mobili** che si occupano di effettuare le mappature della rete (***Job***) a livello locale e inviare al Server le informazioni così ottenute, che si occuperà poi di raccoglierle e unirle. Il Server inoltre si occupa di fornire i Job ai Client in base a condizioni opportunistiche, come locazione del Client o informazioni sulla rete cui il Client stesso si affaccia.

3.1.1 Caratteristiche di un Client

Un Client, come già detto, è un dispositivo mobile con capacità di connessione alla rete, su cui viene installato l'applicativo di mappatura della rete. L'applicazione richiede che il dispositivo possa conoscere la sua posizione geografica in termini di latitudine e longitudine.
L'architettura di un Client è divisa in vari moduli:

- **Ambassador**: implementa il protocollo client – server (discusso in seguito) per ricevere Job dal Server o inviarne i risultati, più le funzioni per la loro gestione.
- **Data Manager**: raccoglie i risultati delle analisi, le impostazioni e altri possibili dati usando i meccanismi di persistenza del dispositivo.
- **Analyses**: insieme di moduli adibiti all'elaborazione dei Job
    - **Tracerouter**: implementa un'analisi di tipo traceroute verso un target specifico in base ai parametri ricevuti dal Server.

Meccanismi di traceroute

Il Client utilizza metodi di traceroute avanzati, in modo da avere dati al più possibile completi e senza errori.

- *Paris Traceroute*[1]: questa metodologia di traceroute permette di avere dati più accurati rispetto alle prime versioni del traceroute. Evita infatti la creazione di anomalie all'interno dei risultati della scansione quali "loop", "cicli", e "diamond". Permette inoltre l'identificazione e la distinzione del load balancing effettuato dai router in:
    - Per-flow load balancing
    - Per-destination load balancing
    - Per-packet load balancing

[1] *Avoiding traceroute anomalies with Paris traceroute* - Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira - 2006
Measuring Loadbalanced Paths in the Internet - Brice Augustin, Timur Friedman, and Renata Teixeira - 2007

- *MDA (Multi-path Detection Algorithm)*[2]: il meccanismo, che rappresenta una naturale estensione del Paris Traceroute, permette di scoprire, con una certezza probabilistica del 95%, tutti i percorsi da una fonte a una determinata destinazione.
- *MIDAR (Monotonic ID-based Alias Resolution)*[3]: permette la risoluzione da interfacce a router, usando parametri ricavati dalla scansione come il RTT (Round-Trip Time) e il campo ID dei pacchetti IP ricevuti.

3.1.2 Protocollo client – server

Formato dei dati

Ogni implementazione del protocollo deve aderire ai seguenti formati:

- *network order* (big endian) deve essere applicato quando possibile;
- valori decimali: sono trasmessi come interi secondo la seguente formula

$$V_i = \lfloor V_f * 10^4 \rfloor$$

In questo documento, verrà utilizzato il bytefield per illustrare la struttura del protocollo: le zone colorate in grigio sono usate per migliorare la visibilità del protocollo stesso e vanno ignorate.

Identificatori

- **CID**: Client ID
  Il Client ID, o CID, identifica un particolare Client.
  Un CID è rappresentato come un intero senza segnosu 4 byte.
- **OID**: Operation ID
  L' Operation ID, o OID, rappresenta una particolare operazione che il Client vorrebbe intraprendere o un particolare servizio che il Client richiede al Server.
  Un OID è rappresentato come un intero senza segno su un byte.
- **AID**: Analysis ID
  L' Analysis ID, o AID, discrimina fra i possibili tipi di analisi che il Client supporta.
  Un AID è rappresentato come un intero senza segno su un byte.
- **JID**: Job ID
  Un Job ID, o JID, è un identificatore collegato a un particolare Job

[2] *Failure Control in Multipath Route Tracing* - Darryl Veitch, Brice Augustin, Renata Teixeira, and Timur Friedman – 2009

[3] *Internet-Scale IPv4 Alias Resolution with MIDAR:System Architecture* - Ken Keys, Young Hyun, Matthew Luckie, and k claffy - 2011

assegnato ad un Client. Un Job è un'istanza di un'analisi assegnata a un particolare Client.
Un JID è rappresentato come un intero senza segno su 8 byte.

Stabilire una sessione del protocollo

Un Client, quando si connette al Server, deve seguire una procedura per stabilire una sessione valida:

- Il Client invia il suo CID al Server e attende una risposta
- Il Server risponde con un messaggio di acknowledgement su un byte (1 accettato, 0 rifiutato)

Operazioni

Ogni operazione rappresenza una specifica richiesta di un Client verso il Server; un'operazione può avere uno o più parametri.

OID	Operazione
0	Refresh stato
1	Richiesta Job
2	Invio risultati

- **Refresh stato**

Un Client esegue questa operazione per aggiornare il suo stato sul Server, inviando la sua longitudine e latitudine. Non c'è altra interazione fra Client e Server.

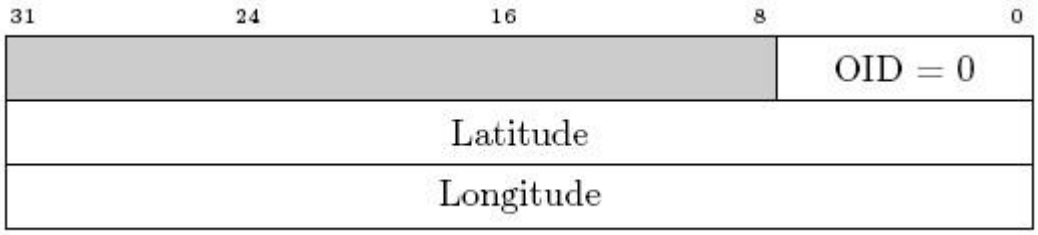


Figura 1: Refresh stato

- **Richiesta di un Job**

Con questa operazione il Client richiede un Job al Server. Questa operazione richiede l'invio della propria latitudine e longitudine.
Il Server a sua volta risponderà inviando un particolare AID, un JID e altri parametri (mostrato in seguito).

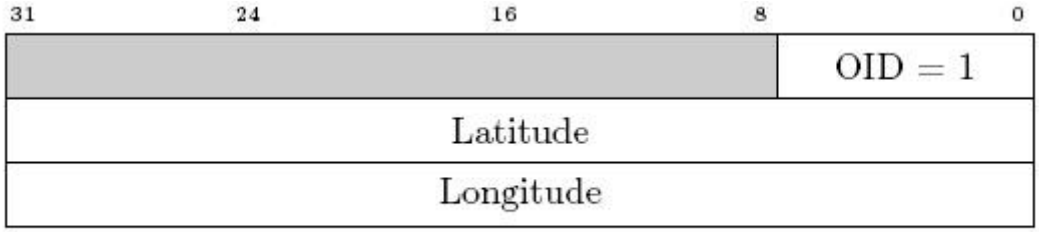


Figura 2: Richiesta di un Job

- **Invio dei risultati**

Un Client esegue questa operazione quando vuole inviare dei risultati al Server.
L'operazione è divisa in vari step:

- Il Client invia al Server il proprio CID assieme al JID assegnato
- Il Server invia al Client un messaggio di acknowledgement su un byte per validare l'invio dei dati (1 accettato, 0 rifiutato)
- Se l'invio dei dati viene accettato, il Client invia i dati secondo il formato di serializzazione previsto (discusso in seguito)
- Il Server invia al Client un messaggio di acknowledgement su un con l'esito della trasmissione (0 dati invalidi, 1 dati validi, 2 ritrasmissione richiesta)

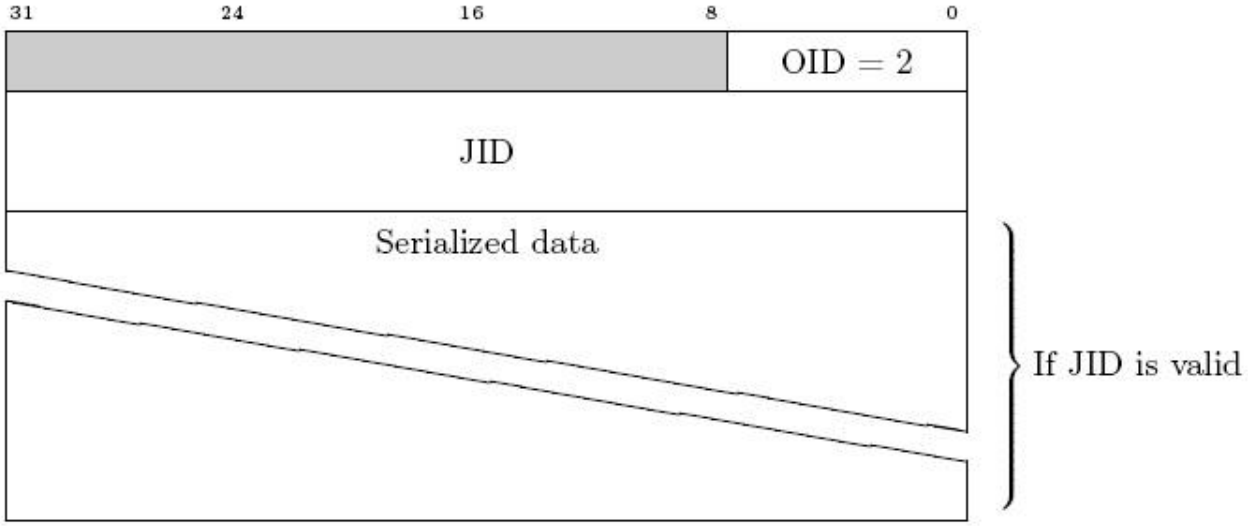


Figura 3: Invio dei risultati

Tipi di analisi

I parametri inviati per le analisi, per essere più generali possibile, vengono inviati tramite delle terne TLV (Tipo-Lunghezza-Valore), con il seguente formato per i campi fissi:

- Tipo, 1 byte senza segno

- Lunghezza, 1 byte senza segno

AID	Analisi
0	Nessun analisi
1	Traceroute

➢ **Traceroute**

Questa analisi, consiste nell'esplorazione della rete tramite traceroute verso un particolare indirizzo IP target.
Elenco dei parametri possibili:

- Target: Tipo 0, specifica l'indirizzo IP target:
  - IPv4 Lunghezza 4
  - IPv6 Lunghezza 16
- Tipo di sonda: Tipo 1, specifica il tipo di sonde da utilizzare:
  - ICMP (default): Lunghezza 1, valore 0
  - UDP: Lunghezza 1+4 / 1+16, valore 1, richiede l'invio dell'indirizzo IP del Client visto dal Server per lo svolgimento dell'analisi
- Lista di esclusione: Tipo 2, contiene una lista di indirizzi IP oltre il quale il Client non deve proseguire l'esplorazione, se incontrati lungo la scansione
- Max TTL: Tipo 3, Lunghezza 1, il massimo valore di TTL da usare per la scansione
- Modalità di esplorazione: Tipo 4, Lunghezza 1+1+1 descrive il comportamento durante la scansione
  - Modalità: 1 MDA modifica il destination address, 0 lo mantiene
  - Limite: Numero massimo di sonde inviate per ogni nodo (default 96)
  - Incremento: Numero massimo di sonde inviate per ogni nuova interfaccia scoperta in uscita da un nodo (default 6)

3.2 Struttura del Server

3.2.1 Obiettivi e funzionalità

L'obiettivo principale preposto alla realizzazione del Server è quello duplice di organizzare il lavoro dei Client in modo da poter raccogliere dati più rilevanti possibili per la stesura della mappatura della rete Internet e di raccogliere i dati stessi, di gestirli e di unirli. Nello specifico il Server deve scegliere qual è il Job migliore da assegnare ad ogni Client, in base alle informazioni che possiede o che gli sono fornite dal Client stesso; una volta

ricevuti i risultati, dovrà inoltre assicurarsi che siano compatibili con le informazioni già accumulate in precedenza e in caso vi siano dei conflitti, trovare il modo di risolverli.

Tipi di dati

I dati da gestire sono di vario tipo:

- Informazioni note sulla rete Internet: posizione geografica delle reti IP, informazioni sugli AS, informazioni per la conversione da indirizzo IP in ASN, etc.
- Informazioni sullo stato dei Client: ultima posizione nota, Job assegnato, etc.
- Dati sulla rete Internet raccolti

I dati sulla rete vengono inviati dal Client secondo una visione a livello router, cioè come un grafo in cui i vertici sono i router scoperti e gli edge sono i loro collegamenti, che viene costruito tramite procedure di dealiasing a partire dal livello interfacce.

Il Server può quindi, analizzando questi dati, ricostruire la mappa della rete Internet a livello router, da cui poi si potrà eventualmente ricostruire mappe a livelli più alti di astrazione logica, come per esempio la mappa dei collegamenti a livello AS, che è di particolare importanza.

I problemi principali in questo tipo di approccio sono i seguenti:

1) Associazione dei router inviati dal Client con quelli memorizzati sul Server
2) Risoluzione di eventuali disambiguità

Il primo problema nasce dal fatto che il Client per identificare i router e distinguerli, utilizza degli ID numerici, che però hanno significato solo locale. Il Server dovrà quindi occuparsi di convertire gli ID e di identificarli secondo il proprio sistema, prima di poter unire i dati in un'unica mappa.

Il secondo problema invece nasce quando la configurazione dei router ricevuta e quella memorizzata differiscono in maniera critica, per ragioni dovute a cambiamenti reali all'interno della rete o a errori nella mappatura: nasce quindi un'ambiguità che deve essere risolta.

Dato che, sia in caso di errori sporadici, sia in caso di cambiamenti reali, la configurazione corretta è quella che presumibilmente tenderà naturalmente a ripresentarsi più spesso, il Server memorizza i dati per cui ci sono le disambiguità e, una volta raggiunta una certa soglia di confidenza calcolata in base ad una euristica, procede alla risoluzione del conflitto.

3.2.2 Moduli

Il Server è composto da vari moduli o blocchi funzionali, come esposto in figura.

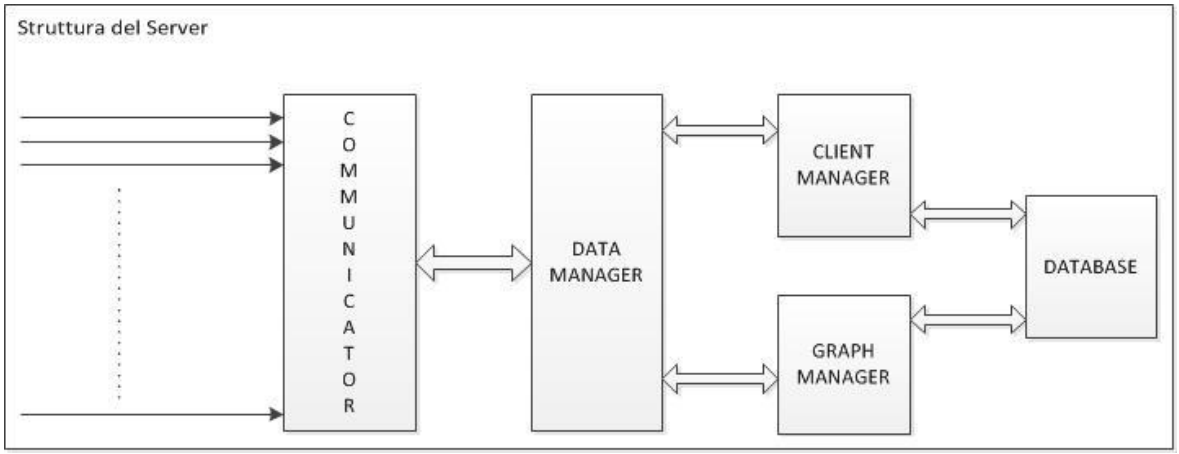


Figura 4: Struttura del Server

Communicator

L'obiettivo principale di questo modulo è la comunicazione con i Client, che avviene secondo il *protocollo Client – Server*; inoltre si occupa di gestire la parallelizzazione e l'handling dei segnali per la terminazione in modo che i dati rimangano consistenti.

Il modulo gestisce tutte le fasi della connessione e gestisce le situazioni di errore dovute alla connessione stessa o agli errori di comunicazione ove possibile.Il modulo fa uso delle interfacce fornite dal Data Manager, che si occuperanno dell'elaborazione dei dati ricevuti durante la comunicazione, delegando loro ogni potere decisionale sull'andamento della comunicazione.

Data Manager

Questo modulo offre un'interfaccia fra il *communicator* e i moduli per la gestione dei dati sottostanti: il communicator quindi non interagirà mai direttamente con i moduli ai livelli inferiori, permettendo quindi eventuali modifiche che risulteranno completamente trasparenti. Si occupa inoltre di tradurre eventuali dati in arrivo dai Client convertendoli in strutture adeguate e effettuare alcuni semplici controlli su di essi (vedere paragrafo sull'iter dei dati per maggiori dettagli sui controlli effettuati): infatti i dati in arrivo dal Client sono serializzati secondo il protocollo Client – Server, che potrebbe essere soggetto a modifiche nel tempo. Ancora una volta, il mascheramento operato dal Data Manager permette di effettuare le modifiche necessarie senza compromettere gli altri moduli.

Client Manager

Qui è dove vengono memorizzati i dati sui Client e lo stato delle loro interazioni con il Server.
Questo modulo inoltre comunica con il database per garantire la persistenza delle informazioni. Le transizioni di stato inoltre vengono stampate su file in modo da poter tener traccia dello storico.

Dati e stato associati a un Client

Ad ogni Client, identificato tramite un ID numerico, vengono associate diverse informazioni:

- **Posizione geografica**: latitudine e longitudine
- **Rete in cui si trova il Client**: indirizzo IP della rete e ASN associato
- **Job assegnato**: indirizzo IP e ASN del Job assegnato, se ve n'è uno
- **Job ID**: identificativo del Job assegnato

È da mettere in evidenza che l'unico parametro che effettivamente distingue un Client da un altro è il Client ID, il quale viene inviato dal Client stesso durante la comunicazione: se il Client cambia il proprio Client ID, il Server non ha modo di verificare l'accaduto.
Il modulo memorizza anche se un Client è già connesso al Server, informazione che sarà poi utilizzata dal communicator il quale potrà eventualmente rifiutare delle connessioni.
Si occupa inoltre di effettuare un log dei dati inviati dai Client; il log verrà poi memorizzato nel database assieme alle informazioni rilevanti, quali la posizione del Client che ha generato quei risultati, la rete in cui si trova, il Job ID e il Job effettivamente assegnato.
Gli stati in cui si può trovare un Client, oltre quello transitorio già accennato di connesso o non-connesso, sono tre:

- Client non ancora registrato: il Client ID non è mai stato incontrato dal Server
- Client registrato ma senza Job assegnato: ogni nuovo Client che si registra si trova in questo stato, ma anche qualsiasi Client che ha finito di inviare i dati rispetto all'ultimo Job
- Client registrato e con Job assegnato: il Client è effettivamente conosciuto dal Server, il quale gli ha assegnato un Job, per il quale il Client è abilitato a inviare dei risultati

Graph manager

Questo modulo rappresenta il nucleo del Server: è a lui che sono delegate le

principali mansioni e si occupa della memorizzazione in database degli stati riguardanti il grafo e l'esplorazione.
I compiti di cui si occupa sono:

- Conversione da indirizzo IP ad AS number
- Memorizzazione di informazioni di localizzazione degli indirizzi IP
- Scelta e assegnazione del Job
- Gestione del grafo dell'Internet prodotto
- Controllo, validazione e fusione dei risultati ottenuti dai Client
- Salvataggio stato dell'esplorazione

3.2.3 Dalla prima interazione al merge dei nuovi dati

Vediamo ora cosa accade durante le interazioni lato Server e quali sono gli step decisionali che si attraversano.

Prima interazione di un Client con il Server

Durante la prima connessione di un Client, questo viene identificato e memorizzato lato Server; le informazioni necessarie a memorizzare il nuovo Client vengono recuperate usando diversi metodi:

- Posizione geografica e Client ID: dal Client stesso tramite il protocollo di interazione
- Rete in cui si trova il Client, caratterizzata da:
  - Indirizzo IP: ricavato dalle informazioni della connessione stessa
  - AS number: ricavato tramite un'apposita procedura di conversione a partire dall'indirizzo IP (discussa in seguito)
- Job: ancora non assegnato

Nel caso in cui la rete risulti non convertibile in un ASN, la comunicazione con il Client viene interrotta: questo perché le politiche per l'assegnazione dei Job (discusse più avanti), si basano tutte almeno in parte sulla conoscenza di questo dato.

Procedura di conversione da indirizzo IP ad AS number

Prima di operare un tentativo di conversione, vengono effettuati dei controlli sull'indirizzo IP in modo tale da essere certi che sia valido secondo gli standard IPv4 (indirizzi privati, indirizzi di loopback, multicast, etc.).

L'operazione di conversione può essere divisa in due fase distinte: la prima cerca di trovare un match per l'indirizzo IP nei dati che si hanno già a disposizione all'interno della memoria, se viene trovato, si applica un

controllo di “freschezza” all’informazione. Qualora il dato risultasse valido l’operazione termina e viene restituita l’informazione.
Nel caso in cui invece la prima fase dovesse fallire, viene fatto scattare un meccanismo che si occupa di prelevare il dato desiderato direttamente dai registri per le informazioni di routing (nel caso specifico, si è scelto di utilizzare Merit RADb http://www.radb.net/), cui si ha accesso tramite il comando “whois”. Se l’interrogazione ha successo, vengono aggiornate le informazioni contenute sul Server dopodiché il dato è pronto per essere utilizzato.

Assegnazione di un Job: politiche e procedure
Superata la fase di identifazione per il Client, il Server procede alla ricerca di un Job adatto. Al momento l’unico Job supportato è quello di un **traceroute** verso un determinato **target**. Ciononostante, la scelta del target può avvenire in base a diverse politiche.

Nomenclatura: Reti e Rotte
Una *rete* è determinata da una coppia di indirizzi IP che delimitano gli estremi della rete stessa. Ad una rete possono essere attribuite altre informazioni:
- *Geografiche*: una coppia latitudine e longitudine e una nazione di appartenenza
- *AS*: l’ASN a cui appartiene la rete

I due insiemi non vanno di pari passo, infatti una rete con informazioni geografiche può contenere indirizzi IP appartenenti a diversi ASN o viceversa.

Una *rotta* è una coppia formata da due ASN per cui si è già tentata una esplorazione: vengono considerati proprio gli ASN in quanto sono i collegamenti fra questi ad avere particolare importanza all’interno della topologia Internet. Ad ogni rotta inoltre viene associato un timestamp, che indica il momento in cui la rotta il Server ha ricevuto dei dati riguardanti la rotta, e un contatore, che indica, in caso i dati non siano stati considerati validi (vedere iter dei dati in seguito), quante volte si è già esaminata la rotta.

Politiche basate su distanza geografica
Queste politiche effettuano la loro decisione prendendo in considerazione:
- Posizione del Client

- Reti per cui è conosciuta la posizione geografica
- Rotte già esplorate

Le politiche implementate basate su questo paradigma sono due:
- Farthest Network Around the World
- Farthest Network in Country

Le due politiche sono in realtà molto simili e la procedura su cui si basano è quasi identica in entrambi i casi.

1. Dalle informazioni sulle posizioni geografiche delle reti, viene creato un punto medio per ogni nazione
2. Le nazioni vengono quindi ordinate in base alla distanza del Client da questo punto medio (in ordine decrescente o crescente rispettivamente per la prima politica o la seconda)
3. Per ogni nazione (seguendo l'ordine)
    a. Vengono ordinate le reti appartenenti alla nazione in base alla loro distanza dal Client in ordine decrescente
    b. Per ogni rete (seguendo l'ordine)
        i. Viene estratto un indirizzo IP campione in maniera casuale appartenente alla rete
        ii. Viene tentata una conversione per ottenere l'ASN
        iii. Se la conversione ha avuto successo e la rotta così formata assieme all'ASN di partenza del Client ancora non ha ottenuto dei dati validi da nessuna scansione o ha un timestamp troppo vecchio, l'indirizzo IP è effettivamente il target desiderato

Politiche basate sulla rete in cui si trova il Client

Le politiche di questo tipo prendono in considerazione:
- ASN associato all'indirizzo IP pubblico del Client
- Una lista di target noti a priori

Questo tipo di politiche è efficace nei casi in cui si vogliano svolgere delle campagne mirate con dei particolari obiettivi.

L'unica politica attualmente implementata che segue questo paradigma prende il nome di:
- Static List

Questa politica è stata pensata per rilevare i percorsi che portano, a partire da un AS di partenza, ad una determinata lista di AS di arrivo, passando per un dato AS intermediario (es. IXP).

Vediamo come viene scelto un target in base a questa politica:

1. A partire da una lista di ASN nota, viene preparata una lista di target formata da tutte le reti appartenenti a quegli AS
2. Se l'AS della rete cui è connesso un Client è l'AS di partenza definito, viene scelto un target fra quelli della lista esclusi quelli per cui è già stato trovato almeno una volta il percorso tramite l'AS intermedio definito

Il target viene poi escluso dalla lista in quanto già esaminato e, per garantire una minore starvation, l'AS di destinazione viene scelto in maniera ciclica fra quelli possibili.

Politiche basate sulla rete di accesso e posizione geografica

Queste politiche, per prendere la decisione sul target, considerano:

- ASN associato all'indirizzo IP pubblico del Client
- Posizione del Client
- Reti per cui è conosciuta la conversione in ASN
- Reti per cui è conosciuta la posizione geografica
- Rotte già esplorate

Al momento, esiste una sola politica basata su questo paradigma:

- Nearest ASN Stub

Uno **Stub** è un ASN che, all'interno di quella che è la gerarchia degli AS della topologia Internet, si colloca ai livelli più bassi, è cioè una foglia nell'albero degli AS. L'elenco degli AS di tipo Stub è noto a priori.
È di particolare interesse esaminare le rotte fra gli stub in quanto fra di essi si vanno a formare dei collegamenti diretti, (spesso tramite IXP) che non sono facilmente tracciabili. È necessario infatti che la sorgente e la destinazione siano vicine fra loro in modo che il link diretto sia quello preferenziale per il routing.

Vediamo la procedura per la scelta del target:

1. Dalle informazioni sulle posizioni geografiche delle reti, viene creato un punto medio per ogni nazione
2. Viene verificato che l'AS cui il Client appartiene sia effettivamente uno Stub
3. Le nazioni vengono quindi ordinate in base alla distanza del Client da questo punto medio in ordine crescente
4. Per ogni nazione (seguendo l'ordinamento)
    a. Vengono ordinate le reti appartenenti alla nazione in base alla loro distanza dal Client in ordine crescente

b. Per ogni rete (seguendo l'ordinamento)
    i. Viene cercato all'interno della rete un indirizzo IP di un altro AS di tipo stub
    ii. Se la rotta così formata assieme all'ASN di partenza del Client ancora non ha ottenuto dei dati validi da nessuna scansione o ha un timestamp troppo vecchio, l'indirizzo IP è effettivamente il target desiderato

Comunicazione del target al Client e Job ID

Una volta scelto, il target viene comunicato al Client, assieme al timestamp in cui è stato generato. Il timestamp prende il ruolo di Job ID, cioè, di identificatore del Job all'interno del sistema. Il Job ID inoltre è collegato al particolare Client e quindi al Client ID. Lo stato del Client viene quindi aggiornato nella memoria del Server, in quanto ha ricevuto un Job.

Ricezione dei risultati e risposta verso il Client

Il Server per prima cosa si assicura che il Client che vuole comunicare dei risultati, abbia effettivamente un Job già assegnato e che sia quello che il Server aveva assegnato al Client per ultimo. Non è possibile quindi l'invio di risultati per dei target che non siano stati assegnati dal Server, o di cui il Server non ha traccia. Il Server potrà rifiutare i dati in arrivo anche in base al tempo trascorso dall'assegnamento del Job, che potrebbe risultare troppo vecchio.

Una volta collezionati i dati, il Server risponderà al Client con un messaggio di *acknowledgement* come già visto dal protocollo in base ai seguenti casi:

- Ricezione dei dati parziale: richiesta di ritrasmissione dei dati
- Dati non validi o corrotti: acknowledgement negativo
- Dati validi: acknowledgement positivo

Iter dei dati

Il primo dato utile che viene recuperato dall'interazione con il Client è il timestamp del momento in cui la scansione è stata effettivamente realizzata: questo viene generato sommando al timestamp dell'istante in cui il Job è stato assegnato, che coincide con il Job ID e che è noto sia al Client che al Server, con una quantità delta che viene inviata dal Client stesso. La scansione infatti può avere avuto luogo in momenti diversi da quelli in cui il target è stato generato, o da quelli in cui i risultati sono effettivamente raccolti dal Server.

Vediamo ora come i risultati, una volta collezionati vengono trattati e gestiti dal Server.

Nomenclatura: Link, Router, Router Falsi e Router Privati

Un *link* è l'informazione base che compone il nostro grafo e raccoglie numerosi dati:

- Router di uscita
- Router di ingresso
- Indirizzo IP dell'interfaccia del router di ingresso
- Indirizzo IP dell'interfaccia precedente lungo il percorso di scansione (predecessore)
- Delay stimato

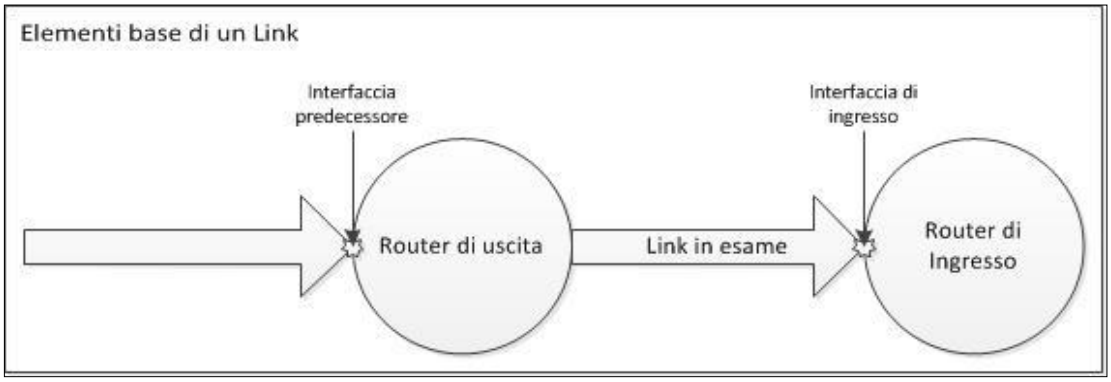


Figura 5: Elementi base di un Link

Un *router* è invece nient'altro che una raccolta di link: se un link è in entrata in un router, significa che l'indirizzo IP dell'interfaccia associata a quel link appartiene al router in questione. Ogni router è identificato da un ID numerico unico, con significato solo locale alla mappa in esame.

Un *router falso* è un router per cui non si hanno informazioni: la sua presenza può essere inferita da anomalie riscontrate durante la scansione della rete. Gli esempi principali di queste anomalie sono router che non inviano i messaggi ICMP Time Exceeded, necessari per la rilevazione tramite traceroute, o che effettuano quello che è definito come "Zero-Forwarding", cioè la propagazione di pacchetti IP anche con TTL ormai esaurito.
Il router è in questo caso è quindi niente più che un'astrazione, ad indicare che dovrebbe essere presente un router reale ma di cui non si ha traccia: infatti i link in entrata in un router falso non presentano un indirizzo IP, mentre di quelli in uscita non è possibile ricavare l'indirizzo IP del predecessore lungo

la scansione. Inoltre in entrambi i casi non è possibile ottenere una stima corretta del delay.

Un *router privato* è invece un router le cui interfacce a noi note sono tutte appartenenti a reti private, cioè caratterizzate da indirizzi IP designati per reti private secondo lo standard IPv4.

Prima fase: lettura e validazione dei dati

Il primo controllo che viene effettuato è sulla percentuale di router falsi all'interno dei risultati; se la percentuale supera un determinato valore, i dati vengono considerati invalidi e quindi:

a) I dati vengono segnalati come invalidi
b) I dati vengono scartati

Se si supera il controllo si passa quindi al parsing dei dati veri e propri. I dati ricevuti vengono esaminati in modo da rilevare:

- Indirizzi IP invalidi (interfacce di loopback, indirizzi multicast, etc.)
- Incongruenze nella struttura dei dati ricevuti

I dati relativi a router privati vengono inoltre scartati, in quanto non sono di particolare interesse nella topologia dell'Internet.

Se il target era stato scelto in base alla politica di Static List, durante il parsing, gli indirizzi IP vengono controllati per verificare se si è avuto un passaggio all'interno dell'AS intermedio ricercato e, in caso di riscontro e se i dati sono stati considerati validi, viene memorizzato che si è trovato il percorso per l'AS di destinazione associato al target nella Static List.

Seconda fase: Ricerca dei conflitti e Quarantena

I router, per come li abbiamo visti finora e per come sono inviati dal Client, sono caratterizzati da degli ID numerici progressivi.
Il significato di questi ID numerici è però collegato solamente alla scansione in cui sono stati creati: una volta sul Server infatti, vanno associati ai router che sono già presenti sul Server e di cui il Client non ha informazione.
Le regole di associazione sono solo due e sono le seguenti:

- Due router sono associabili se entrambi posseggono la stessa interfaccia (lo stesso indirizzo IP)
- Due router sono associabili se entrambi possiedono un link in uscita verso la stessa interfaccia

Queste regole nascono da due ipotesi fondamentali, e cioè:

- Gli indirizzi IP pubblici sono unici nel mondo dell'Internet (sempre vera)
- I link sono tutti di tipo punto – punto (in alcuni casi può portare a degli errori, ad es. gli IXP usano delle ethernet che sono link punto – multipunto)

Da queste regole esulano i router falsi, che formano un caso a parte e che verranno discussi in seguito.
Durante l'associazione possono capitare dei casi in cui le informazioni presenti sul Server e quelle arrivate dal Client non coincidano, generando così dei conflitti.
Tenendo a mente che le informazioni contenute sul Server possono essere considerate un'unica scansione, i casi per cui esiste un conflitto sono i seguenti:

- Un router associato ad una scansione raccoglie link di più router differenti associati ad un'altra singola scansione
- Esiste già un conflitto non risolto riguardante un router preso in esame nell'associazione

I conflitti sono comunque sempre derivati da casi in cui i link sono raggruppati in maniera errata e quindi formano router non esistenti (errore nel dealiasing dal livello interfaccia al livello router del meccanismo di traceroute) o da errori di associazione (link punto – multipunto).
Se esiste un conflitto e non è possibile risolverlo nelle fasi successive, i router che hanno causato il conflitto vengono memorizzati appositamente e posti in *quarantena* in modo da poter essere risolti grazie a scansioni future.
Le scansioni in quarantena vengono identificate in base al timestamp loro associato.

Terza fase: Calcolo Confidenza ed Eleggibilità

Il calcolo della *confidenza* riguarda la risoluzione dei conflitti.
Ad ogni router, di ogni scansione, per cui è stato rilevato un conflitto viene assegnato un valore di confidenza, inizialmente pari ad 1 (uno). Il valore di confidenza viene poi incrementato all'arrivo di ogni nuova scansione che interessa quel router ma che di per sè non creerebbe conflitto, che ricade quindi nel secondo caso di quelli elencati. Il valore di confidenza così calcolato può essere quindi memorizzato in attesa di altre scansioni, senza

rendere necessario ricalcolarlo ogni volta. Inoltre viene tenuto, tramite un apposito contatore, il conto delle scansioni in conflitto considerate per il calcolo. Questo viene fatto per tutte le scansioni in conflitto sullo stesso router.

Una volta calcolati i nuovi valori di confidenza, viene calcolata la *percentuale di confidenza* per le scansioni. La percentuale di confidenza viene calcolata dividendo il valore calcolato di confidenza per il numero di scansioni in conflitto che hanno contribuito al calcolo di quel valore. Nel caso in cui più scansioni arrivino a superare il limite prefissato della percentuale di confidenza, la precedenza viene data prima al grafo principale (che quindi non comporta modifiche) e poi alle altre scansioni in quarantena a partire da quelle più recenti.

Se la percentuale supera un certo valore prefissato, la scansione viene *eletta* a prendere posto nel grafo principale al posto di quella presente al momento contenuto nella memoria del Server. Questo significa che la configurazione dei router, o meglio, la loro ripartizione dei link presente nella scansione in conflitto viene scambiata con quella del grafo principale.
La procedura è effettuata tramite due tipi di operazioni:

- Unione di due router, in cui il router risultante raccoglie tutti i link dei due router
- Scissione di un router

In entrambe, un elemento particolarmente utile da considerare è quello dei predecessori dei link in uscita, che ci danno informazioni certe sulla topologia reale di Internet e che aiutano nei casi particolari di indecisione.

Durante l'operazione di unione di due router, va controllato che non esistano link fra i due: infatti questo porterebbe alla creazione di un loop, che sappiamo non può esistere. In questi casi l'operazione va effettuata solo in parte ove possibile, prestando particolare attenzione a quali interfacce vengono tralasciate: i link in uscita che posseggono queste interfacce in quanto precedessori, devono essere tralasciati anch'essi. I link in uscita a loro volta possono avere altre interfacce come predecessori, che vanno tralasciate anch'esse, e così via.

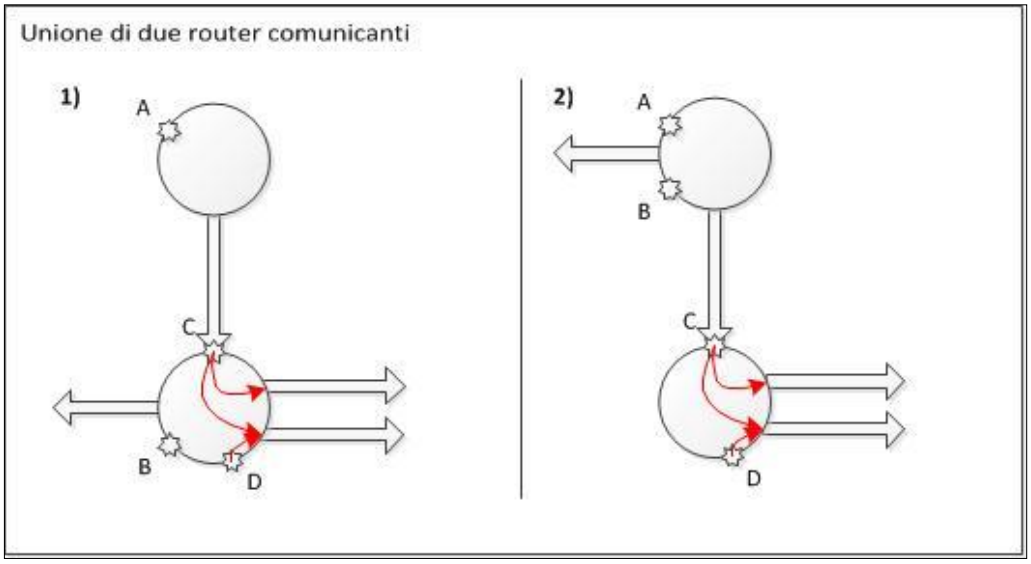


Figura 6: Unione di due router comunicanti

Anche per l'operazione di scissione bisogna prestare particolare attenzione, in quanto, considerando il router come un elenco di indirizzi IP che corrispondono alle sue interfacce più un elenco di link di uscita verso altri router, quando questo viene diviso restano due problemi da risolvere:

- come suddividere le interfacce di cui non si hanno informazioni nella scansione eletta o che non è possibile associare altrimenti in quanto predecessori di link in uscita già assegnati: queste andranno a formare un nuovo router ulteriore, che verrà poi eventualmente riassorbito successivamente tramite unioni e/o scissioni derivanti da altri conflitti
- come suddividere i link in uscita dal router originario: questi link vengono ripartiti in base alle interfacce del router, grazie all'informazione sugli indirizzi IP immediatamente precedenti lungo il percorso di scansione che ha portato alla loro scansione

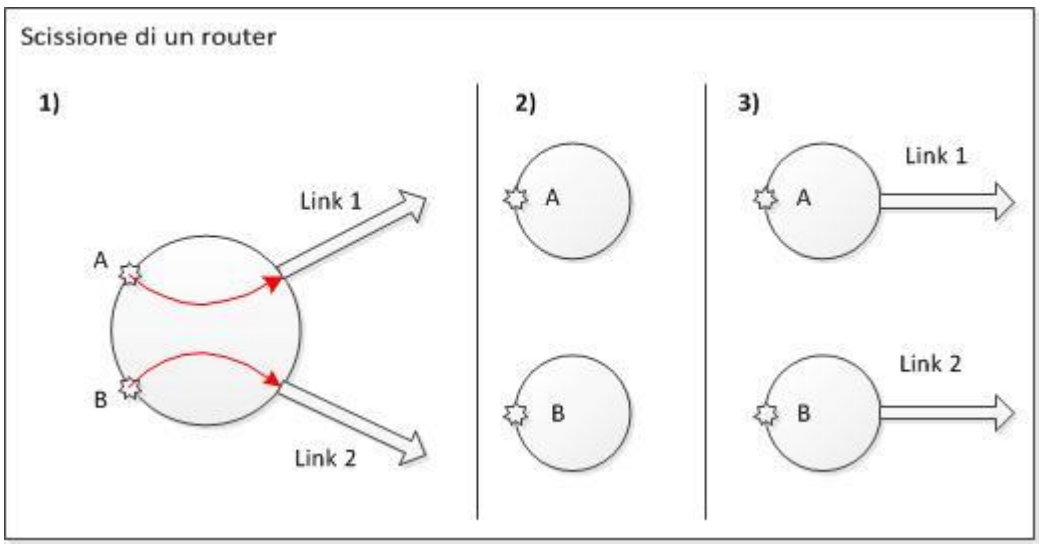


Figura 7: Scissione di un router

Una volta che una scansione viene eletta, le altre vengono considerate come invalide, e vengono quindi cancellate dalla memoria. Da qui l'importanza del contatore per il numero di scansioni, in quanto se una scansione viene cancellata, se non si tenesse traccia della sua esistenza nel calcolo di confidenza per gli altri conflitti, il meccanismo produrrebbe delle percentuali di confidenza errate.

Quarta fase: Fusione dei Dati

In questa fase vengono esaminati i dati che non hanno prodotto dei conflitti, e ne viene effettuata la fusione con i dati già presenti nel grafo principale nella memoria del Server.

Per ogni router per cui non si è riusciti a trovare nessuna associazione viene creato un nuovo router, identificato da un nuovo router ID.

Fino a questo punto abbiamo tralasciato il caso dei router falsi e la loro associazione con i router già presenti nella memoria del Server: questa viene fatta in base ai router a cui loro sono direttamente collegati. Questo nasce dall'ipotesi che questi router siano effettivamente dei casi isolati nella topologia dell'Internet e quindi è facile pensare che se un router presenta già un collegamento ad un router falso, quello sia proprio il router falso che stavamo cercando. La ricerca delle associazioni per i router falsi viene fatta in maniera iterativa, fino ad esaurimento o fino a quando non si riesce a trovare altre associazioni. Inoltre, va evitato che lo stesso router sul Server venga associato a più router falsi, in modo da rispettare le sequenze dei dati in arrivo dal Client, altrimenti si avrebbero degli errori in caso di catene di router falsi.

Una volta finite le associazioni e creati i nuovi router, i link vengono memorizzati all'interno degli stessi o, in caso di link già presenti, vengono aggiornati con le nuove informazioni.

Salvataggio risultati e ripristino dello stato iniziale

Una volta che i dati sono stati ricevuti, e se ne è verificata l'integrità, il Server si occupa di effettuare il salvataggio delle informazioni riguardanti i dati ricevuti:

- Dati validi -> Rotta esplorata con successo
- Dati non validi -> Contatore per le ricezioni invalide per la rotta incrementato

In entrambi i casi viene memorizzato anche il timestamp collegato ai dati che hanno prodotto questi risultati. I dati stessi ricevuti dal Client vengono memorizzati in un log e collegati alle informazioni sul Job (Job ID e target assegnato) e sul Client (posizione geografica e rete), in modo da poter essere eventualmente recuperati in seguito.

Finite le operazioni di memorizzazione dei dati ricevuti, vengono cancellate le informazioni sul Client riguardo al Job assegnato e viene riportato allo stato iniziale, in modo che alla prossima connessione gli possa essere assegnato un nuovo target.

Capitolo 4: Validazione del sistema

Dopo aver illustrato l'architettura generale e spiegato come il Server svolga le proprie funzioni, in questo capitolo si andrà ad illustrare un esempio di funzionamento del sistema; per fare ciò ci baseremo sulla rete GARR[4], la rete italiana dell'università e della ricerca, la cui struttura fisica reale è pubblica ed è quindi un ottimo campo di prova per verificare i risultati che si ottengono.

Usando come punto di partenza l'Università di Pisa, che si affaccia direttamente sul POP (Point-Of-Presence) denominato "PI1" della rete GARR, si è individuato degli indirizzi IP target da assegnare al client per la scansione tramite traceroute. I target sono stati scelti agli edge della mappa, in modo che i punti interni delle rotte venissero scoperti automaticamente durante le scansioni. Per le scansioni si è usato il traceroute in modalità ICMP.

[4] *http://www.garr.it*

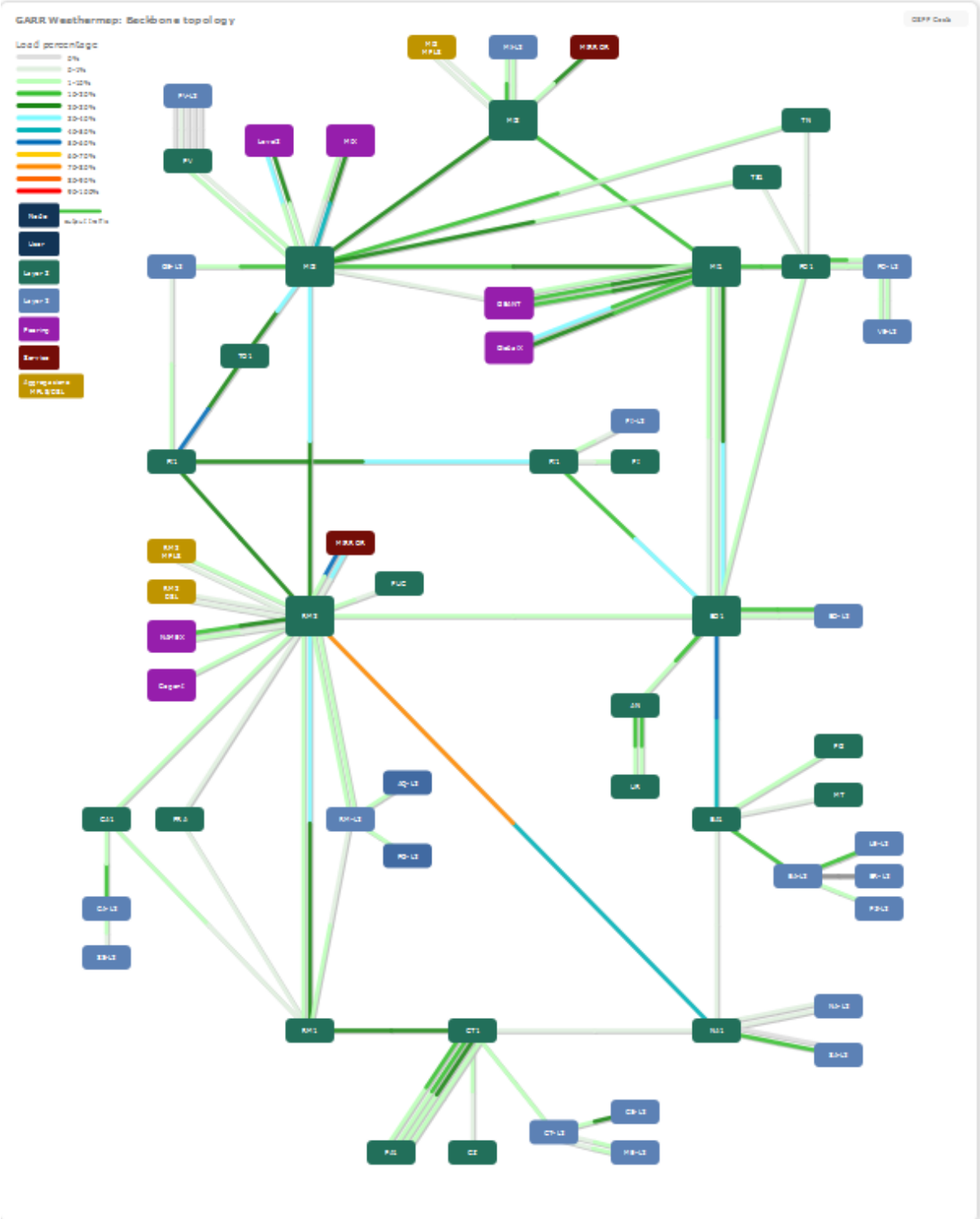


Figura 8: Backbone rete GARR

Si vedranno ora i risultati delle singole scansioni, mostrati per semplicità di illustrazione, come grafi[5], i cui nodi rappresentano i singoli router scoperti e gli edge mostrano a lato l'interfaccia su cui si affacciano; oltre a mostrare i risultati delle singole scansioni, si vedrà anche come il Server operi su di esse per arrivare al grafo finale.

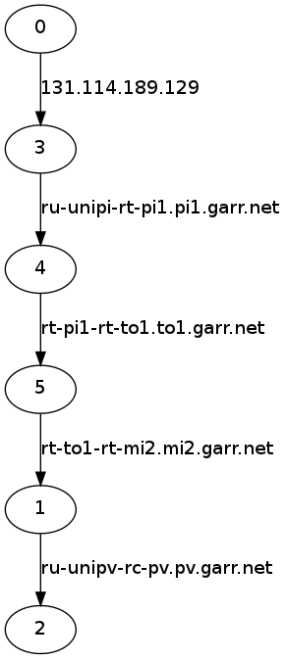


Figura 9: Scansione 1

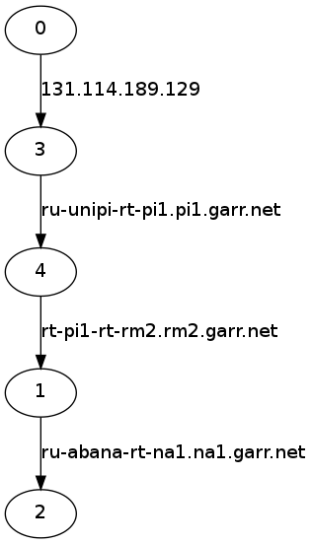


Figura 10: Scansione 2

Come si può vedere, le due scansioni hanno prodotto dei grafi che coincidono dalla radice, fino al nodo 4, che rappresenta il POP PI1, dopodichè divergono. Ci aspettiamo quindi che il Server riconosca i router già incontrati e che operi una fusione dei due grafi.

Primo risultato parziale

Si può notare infatti come il Server abbia effettivamente riconosciuto i router, ne abbia fuso i contenuti e abbia creato dei nuovi nodi per i nuovi router. Da notare come, il router 4 della scansione 1 e il router 4 della scansione 2 siano stati uniti, andando a formare un unico router con due archi di uscita, che era il risultato che ci aspettavamo.

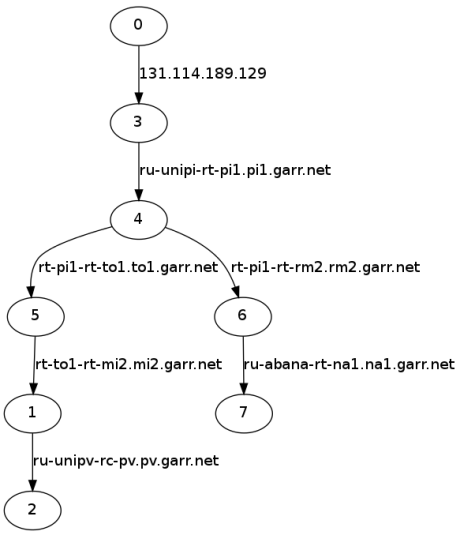


Figura 11: Grafo parziale 1

[5] I grafi sono generati tramite il software "dot" del pacchetto "graphviz" (*http://www.graphviz.org/*)

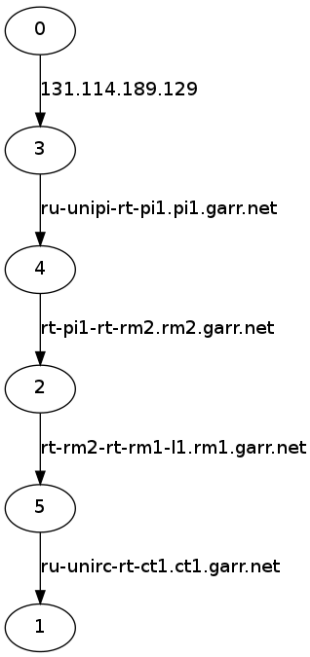


Figura 12: Scansione 3

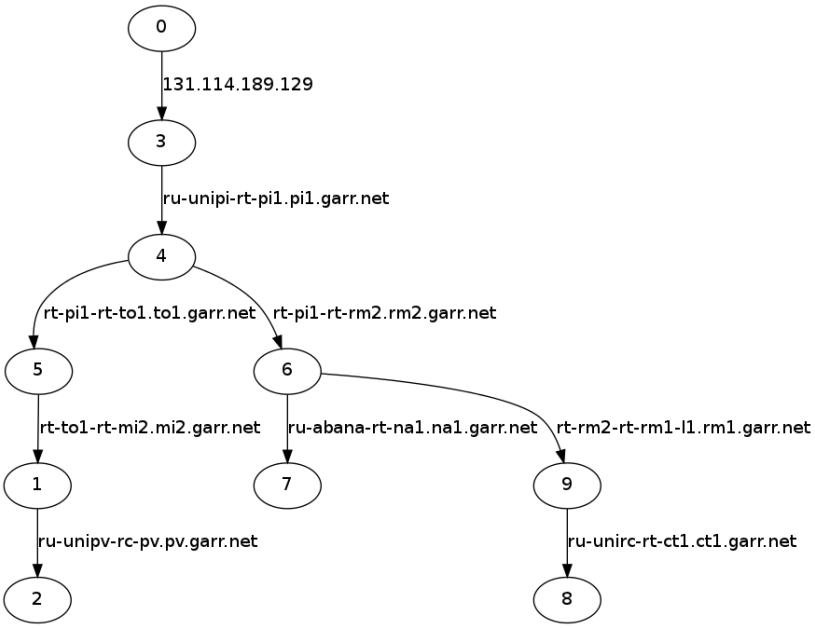


Figura 13: Grafo parziale 2

Anche in questo caso si vede chiaramente come il Server abbia effettuato la fusione della scansione nella mappa globale, identificando i router della scansione 3 e creandone di nuovi per quelli non ancora conosciuti. Le stesse operazioni verranno poi ripetute anche per le scansioni in arrivo successivamente, che verranno illustrate brevemente.

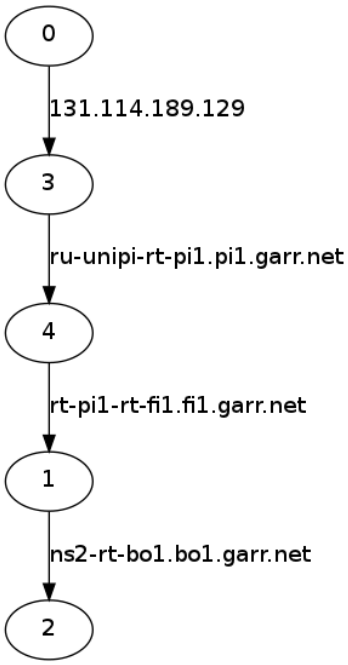


Figura 14: Scansione 4

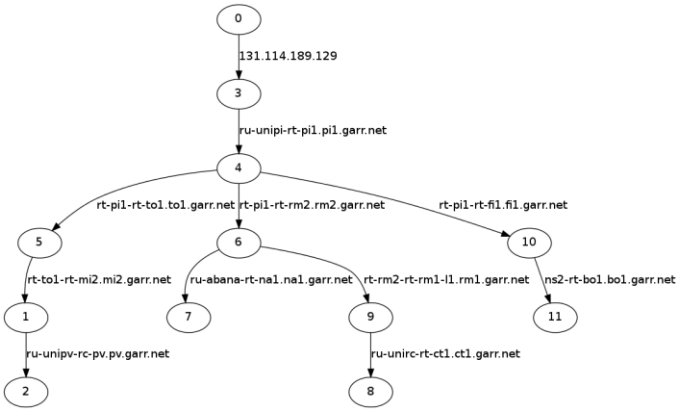


Figura 15: Grafo parziale 3

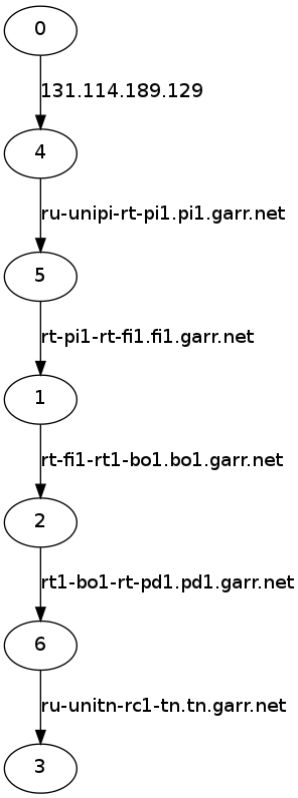


Figura 16: Scansione 5

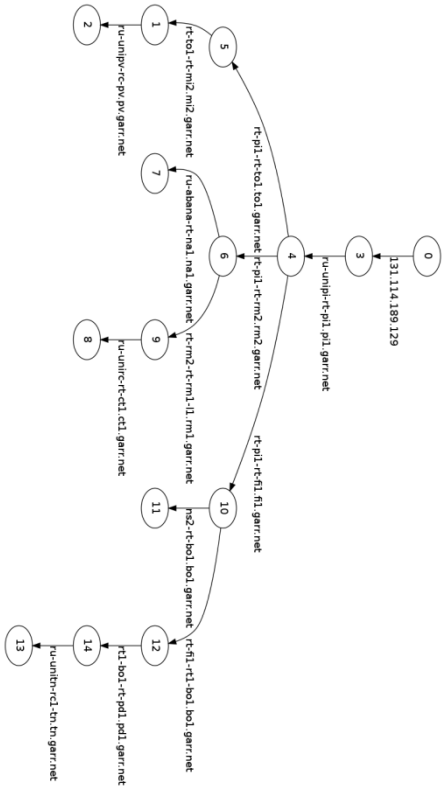
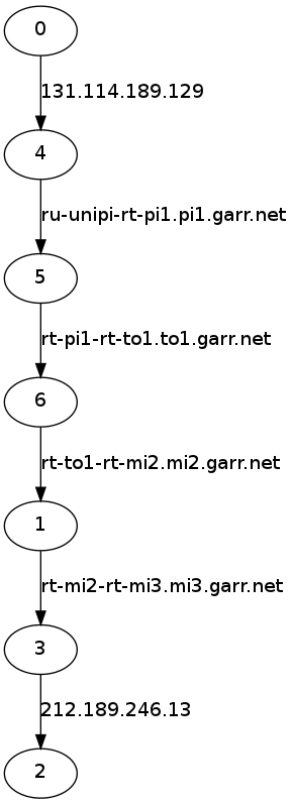


Figura 17: Grafo parziale 4



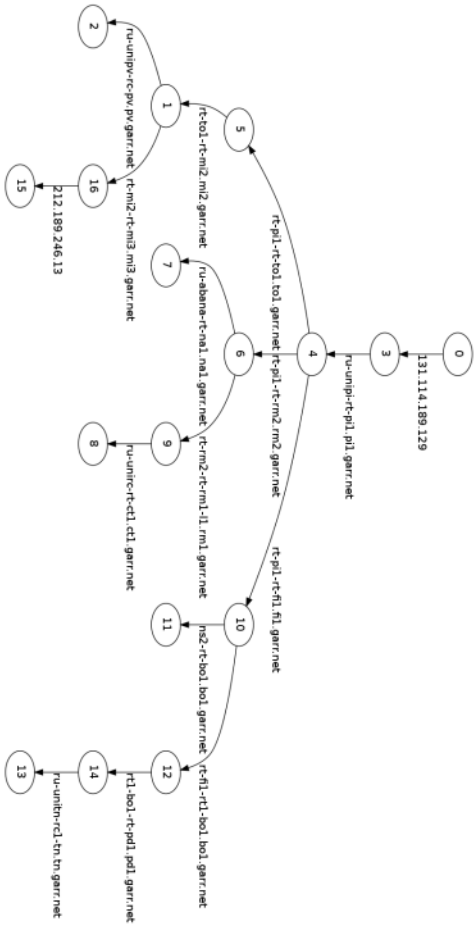


Figura 18: Scansione 6

Figura 19: Grafo parziale 5

Visto il funzionamento del meccanismo sino a questo momento, si andrà ora ad analizzare il grafo complessivo generato durante l'esperimento.

Per l'analisi sono stati considerati 23 target, secondo le modalità già definite, ognuno dei quali ha generato un end point differente. Durante l'analisi delle scansioni lato Server non sono stati rilevati conflitti fra i dati in arrivo e quelli già presenti.
La topologia effettivattivamente ricavata dal Server rispecchia fedelmente la mappa della rete fornita dal GARR, sia per quanto riguarda i collegamenti fra i POP, sia per quanto riguarda gli end point della rete, con l'unica eccezione di un nodo, identificato con il numero 12, corrispondente ad un router che secondo la mappa del GARR dovrebbe collegarsi al POP "MI3" di Milano, mentre dalla mappa generata dalle scansioni risulta collegato al POP "MI2", sempre di Milano; tuttavia non c'è motivo di ritenere che il nodo in questione

sia un errore di rilevazione, dato che il resto dei dati risulta congruente con la topologia reale, ma è più probabile che il router abbia recentemente cambiato il POP a cui effettivamente si collega, ma che tale informazione non sia stata ancora aggiornata nella mappa pubblica del GARR.
In particolare si può vedere come i vari POP siano collegati fra di loro, e come questi collegamenti rispecchino esattamente la disposizione mostrata nella mappa della backbone della rete GARR.

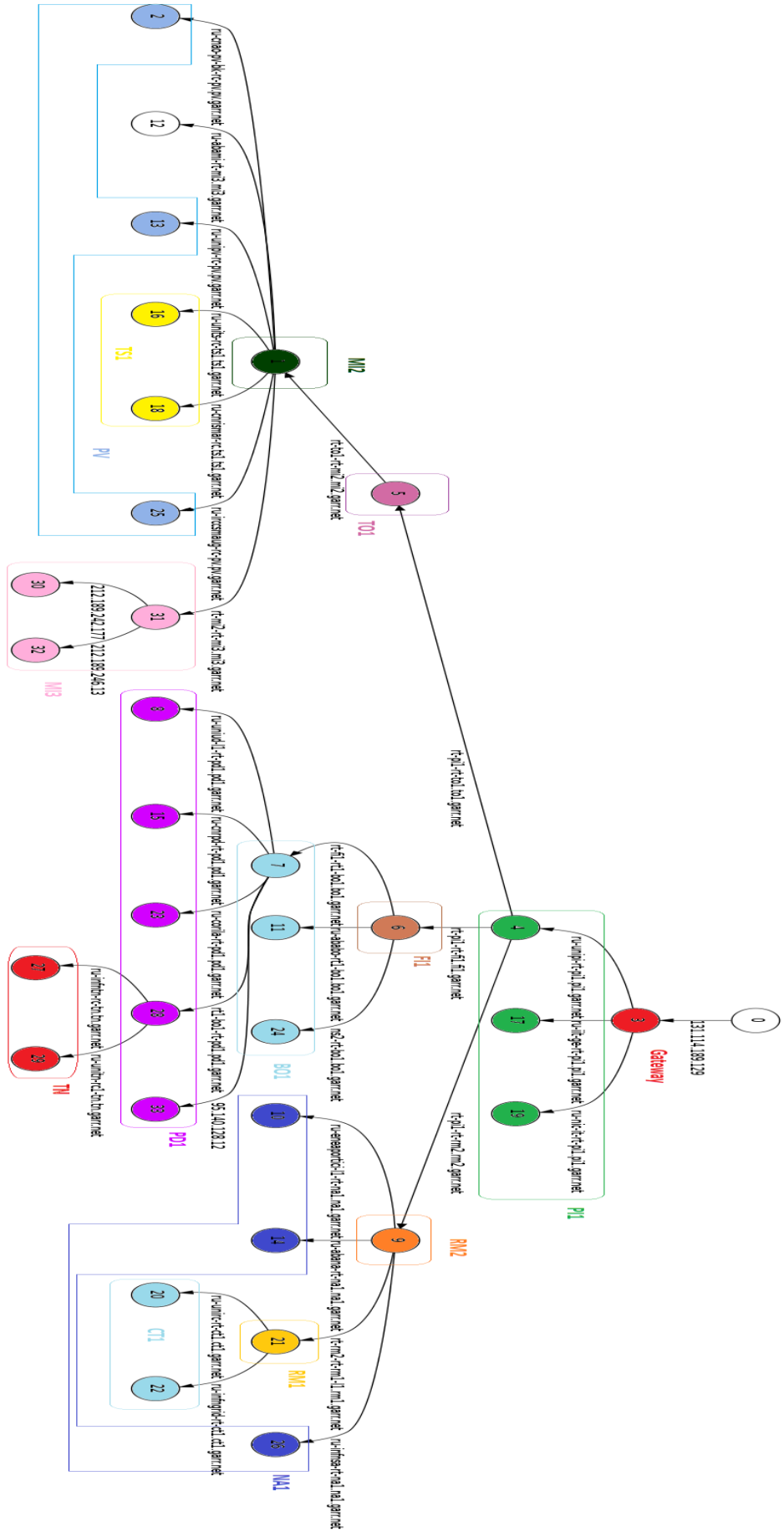


Figura 20: Grafo completo

Capitolo 5: Conclusioni

In questo elaborato si è illustrato quali siano le motivazioni per un sistema distribuito di scansione e mappatura della rete Internet e sulle sfide che esso comporta.
L'attenzione è stata incentrata sulla parte Server dell'architettura, e sui compiti ad esso affidati.
Sono stati elencati quali sono i moduli principali che formano il Server, quali funzioni abbiano e come questi siano collegati fra di loro.
Successivamente si è visto come il Server interagisca con i Client e quali siano le politiche che utilizza per scegliere un task da affidagli, che si basano su parametri dipendenti dalla posizione geografica del Client e dalla rete a cui si affaccia, e quali sono le motivazioni principali che hanno portato alla nascita di ognuna di esse.
È stata posta l'attenzione su quali sono e che formato hanno i dati che il Server raccoglie e sono stati messi in evidenza quali siano i problemi principali nella raccolta dei dati generati dai Client, come questi possano risultare in conflitto con gli elementi già collezionati dal Server e come questo possa risolverli, basandosi su delle euristiche, andando a fondere o scindere elementi già presenti pur mantenendo la coerenza dei dati, risolvendo quindi le ambiguità.
Si è anche visto quali siano le principali anomalie riscontrabili all'interno della rete e come il Server gestisca questi casi particolari sotto forma di router falsi e come questi vengano trattati.
La validazione del sistema è avvenuta tramite esame di una rete nota, la rete GARR, che ha rivelato come il sistema funzioni e non produca errori generando una mappa che segue fedelmente quella pubblica di riferimento.

Per concludere, vedremo quali sono le problematiche ancora da affrontare e quali sono i possibili sviluppi futuri del sistema:

- *Associazione dei router falsi*: la tecnica al momento utilizzata dal Server, seppur immediata, non garantisce la completa affidabilità dei risultati nel caso in cui queste anomalie siano molto frequenti in zone ristrette della rete, si potrebbe quindi voler passare ad euristiche più efficienti e con più garanzie;

- *Raccolta di statistiche sulla rete*: sebbene il sistema ne abbia la capacità, al momento non prevede la raccolta di statistiche come percentuale di pacchetti persi, saturazione della rete o molti altri elementi che potrebbero essere di interesse da monitorare, ma che per il momento vengono tralasciate;

- *Metodologie di analisi:* al momento l'unica analisi supportata dal sistema è l'analisi di tipo traceroute a partire dal terminale Client verso una destinazione scelta dal Server. Potrebbero essere invece definite delle analisi diverse ad esempio di tipo peer-to-peer tra dei Client, in modo da verificare quali siano i comportamenti della rete in base ai diversi tipi di pacchetti e ai requisiti di QoS che essi richiedono (ad es. i pacchetti VoIP), oppure per verificare l'adempimento degli ISP rispetto agli SLA (Service Level Agreement) definiti da contratto;

- *Stima dei parametri ottimi*: sebbene il protocollo supporti diversi tipi di parametri di configurazione, al momento non si fanno assunzioni sugli stessi, lasciandoli ai valori di default; si dovrebbero raccogliere diverse statistiche in base a test ripetuti con parametri differenti e identificare se e in quali casi sia necessario specificare parametri diversi.