UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea triennale in *Ingegneria Informatica e dell'Automazione*

Studio e configurazione di una VPN site-to-site in ambiente CG-NAT

Study and configuration of a site-to-site VPN in CG-NAT environment

Relatore:
Prof. Ennio Gambi

Correlatore:
Ing. Adelmo De Santis

Tesi di Laurea di:
Alessandro Illuminati
matricola 1078466

Anno accademico 2020-2021

A Franco.

Prefazione

Nell'ambito del mio percorso universitario ho avuto modo di approfondire le tematiche relative al mondo delle reti e del networking, a tal proposito grazie alla possibilità offerta dal Dipartimento di Ingegneria dell'Informazione, dal Prof. Ennio Gambi e dall'Ing. Adelmo De Santis ho conseguito con successo la certificazione "*HUAWEI HCIA Routing and Switching*". Successivamente, grazie alle competenze acquisite, ho collaborato con alcuni miei colleghi per progettare e realizzare una implementazione di una VPN site-to-site attraverso una connessione radiomobile per conto dell'azienda Esse-ti S.r.l.
In questo elaborato verranno esposte le principali fasi del progetto realizzato, ponendo un particolare focus sulle problematiche iniziali affrontate e all'architettura di rete nel cui ambito è stata realizzata la comunicazione tramite un canale sicuro.

Indice

Elenco delle figure

Nella didascalia di ogni immagine vi è il link della pagina web da cui è stata presa, inoltre, sono citate anche accanto ai link nella sitografia.

Capitolo 1

Introduzione

1.1 Scopo e analisi del progetto

Nell'ambito di una convenzione stipulata con il Dipartimento di Ingengeria dell'Informazione, l'azienda **Esse-ti S.r.l.** ha esposto il suo progetto di fornire ad un certo gruppo dei propri clienti un router 4G per permettere di raggiungere e controllare dispositivi domotici, cablati e non, esterni alla rete locale dell'utente. Il gateway fornito al cliente risulta dotato di una batteria e di uno slot SIM quindi in grado di connettersi a *global internet* attraverso una connessione geografica radiomobile, in particolare mediante la connettività 4G garantita da un *Internet Service Provider* nazionale. Questa caratteristica permettere ai dispositivi connessi al gateway di fruire di un accesso ad internet e della possibilità di gestire comunicazione vocali, con il vantaggio di essere indipendenti da eventuali guasti che possono occorrere all'alimentazione elettrica o alla connettività via cavo nel luogo d'installazione dell'apparato.
Il lavoro si è focalizzato sul rendere possibile una comunicazione sicura tra un calcolatore autorizzato del cliente, situato nella rete locale dello stesso, e un dispositivo installato in un'altra sede fisica connesso al gateway fornito, nonostante le reti dei grandi provider mobili adottino nella maggior parte dei casi l'uso del protocollo **CG-NAT**, che impedisce la comunicazione diretta tra i due *end-point* della trasmissione. Per realizzare questa particolare configurazione di rete è stato perciò necessario utilizzare un server esterno dotato di un indirizzo IP pubblico, appoggiandosi ad un servizio *VPS* fornito in particolare dal provider **OVHCloud**. In seguito si è dovuto provvedere alla creazione di una connessione sicura per i dati nel trasito attraverso la rete pubblica dal calcolatore del cliente al cloud server e infine dallo stesso al device domotico finale. Una volta ottenuto l'accesso all'istanza server remota, dopo una verifica delle caratteristiche computazionali e di compatibilità software della stessa, si è deciso di adottare la suite open-suorce **OpenVPN** per l'implementazione dei tunnel cifrati attraverso i quali instaurare la comunicazione; questo software, a configurazione conclusa, garantirà un canale virtualmente diretto tra i due *end-point*, in altre parole essi risulteranno connessi ad una stessa rete locale.

Per la realizzazione e la verifica della configurazione richiesta, la problematica è stata schematizzata in una topologia di rete di test facendo le seguenti semplificazioni ed osservazioni:

- Ogni cliente ha accesso ad un unico gateway remoto;
- Ogni gateway è caratterizzato da più interfaccie, una di queste è connessa a global internet tramite la tecnologia radiomobile;

- Le restanti interfaccie del gateway sono tutte caratterizzata da uno spazio degli indirizzi privato, in generale ad esse possono essere collegati diversi dispositivi;
- La connessione sicura dal calcolatore del cliente al server remoto sarà di facile configurazione tramite l'interfaccia grafica del software OpenVPN;
- Dovrà essere possibile raggiungere dall'elaboratore del cliente, con il comando di debug *ping*, uno dei device connessi al gateway remoto, garantendo la bidirezionalità della comunicazione.

Di seguito (Figura 1.1) viene proposto lo schema di principio che permette di realizzare la comunicazione desiderata con i vincoli posti, adottando le tecniche già citate.

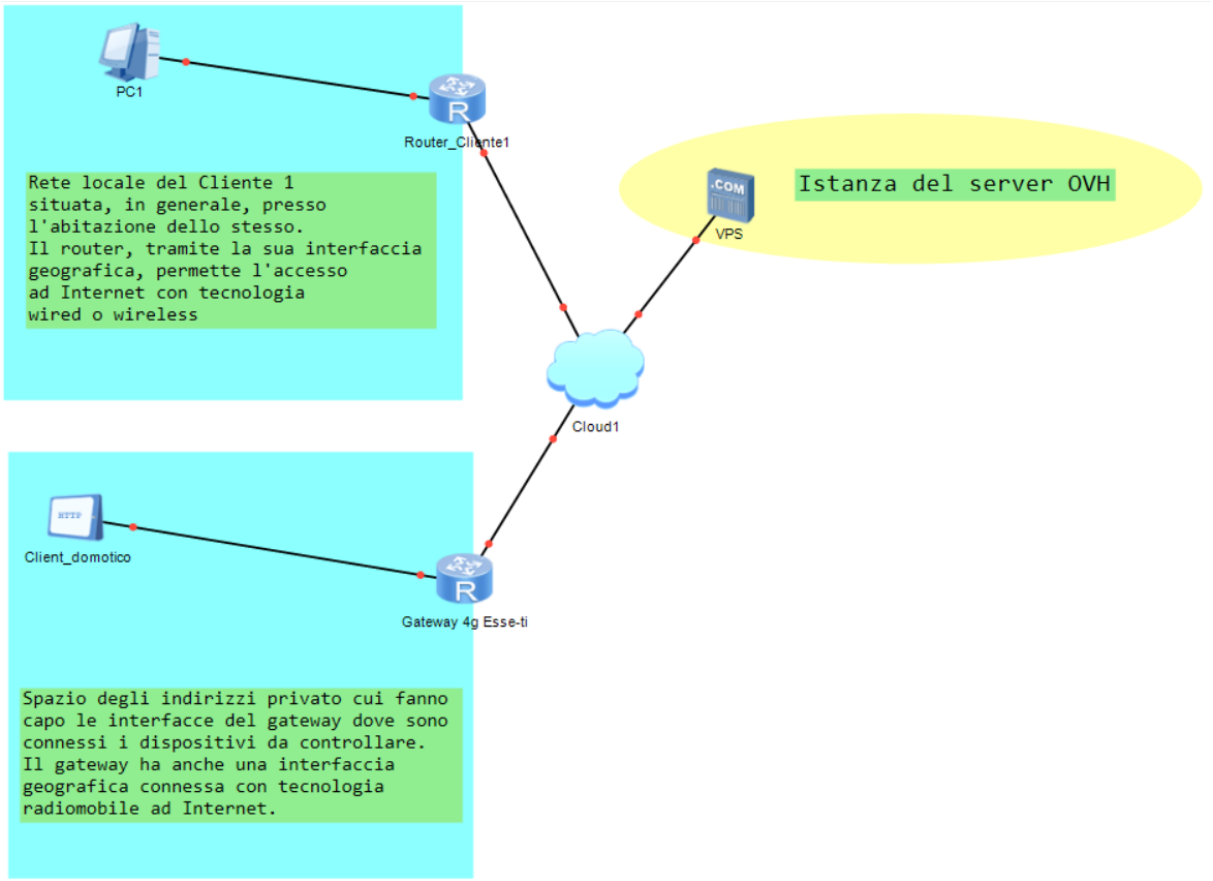

Figura 1.1: Topologia di rete di principio adottata per la comunicazione

1.2 Generalità sui servizi VPS e caratteristiche del particolare servizio scelto

Per realizzare la comunicazione progettata si è evidenziata la necessità di un server remoto, che provveda a veicolare i dati tra i due *end point* della comunicazione. In particolare l'azienda ha

richiesto un servizio che, seppur costoso, sia in grado di garantire un'alta affidabilità, bassa latenza e che in generale ponga pochi vincoli ad ulteriori e futuri sviluppi (quali ad esempio VNC o un servizio di desktop remoto), sia in termini di prestazioni offerte, sia in quanto a banda garantita. Ad una prima analisi, ci sono 3 grandi player che offrono sul mercato servizi cloud altamente configurabili, potenti e granitici: *Amazon AWS*, *Microsoft Azure* e *Google Cloud*. I prezzi proposti variano molto in base al servizio richiesto, abbiamo importi contenuti per macchine che lavorano a "trigger", fino a cifre più importanti per macchine always-on che riservano staticamente delle precise risorse; inoltre è possibile anche configurare dei limiti di budget con i relativi avvisi. In una successiva discussione si è deciso di usare un servizio più economico per le finalità di test preposte, e la scelta si è orientata su due provider in particolare, Aruba ed OVHCloud, che offrono entrambi un ampia gamma di server privati virtuali, con molte opzioni configurabili.
L'adozione di un *Virtual Private Server* mette a disposizione una singola istanza di un sistema operativo che viene eseguito in ambiente virtuale, di conseguenza più VPS possono essere eseguiti contemporaneamente sullo stesso hardware fisico. Questa caratteristica permette di lavorare in maniera del tutto indipendente dai vincoli associati all'hardware (evoluzione o upgrade dei componenti, malfunzionamenti tecnici, monitoraggio dello stato dei Dischi, RAM e CPU ecc...) dato che le singole istanze possono risultare anche migrabili su diverse macchine fisiche. Si ha così il vantaggio di avere un controllo totale sul proprio server per finalizzare al meglio i propri obiettivi, pur mantenendo nella maggior parte delle situazioni le performance di un ambiente dedicato. I costi minori dovuti alla condivisione tra molte istanze di uno stesso hardware fisico permettono l'accesso ai clienti finali a piani che possono anche essere molto economici. In generale le VPS sono adatte alla maggior parte degli utilizzi Web e per progetti di dimensioni contenute, anche in ambienti di produzione dove possono garantire delle prestazioni costanti, bisogna però dimensionare le caratteristiche del servizio scelto in base agli applicativi da eseguire. L'utilizzo di un VPS richiede delle discrete competenze in amministrazione di server, in particolare queste nozioni sono fondamentali per gestire il sistema operativo della macchina virtuale, installare e configurare applicazioni. Nel nostro caso andremo ad utilizzare il software OpenVPN per permettere la comunicazione sicura tra server e client, si richiede perciò una fondamentale esperienza nel networking e del funzionamento dello stack TCP/IP, per la configurazione del firewall e il debug.

Le soluzioni proposte per i server virtuali del provider OVH garantiscono prestazioni elevate, scalabilità, semplicità e la localizzazione presso un Data Center non in territorio Italiano, ma comunque Europeo per una buona latenza della comunicazione (le opzioni consigliate a riguardo sono Francia o Germania), il tutto ad un prezzo ragionevole, perciò si è deciso di appoggiarsi ai servizi proposti da questa azienda.
Tra le opzioni offerte da OVH per le istanze VPS, è stato concordato l'acquisto della VPS di gamma *Essential* con in più l'opzione di backup *Snapshot*: essa risulta essere un ottimo compromesso per l'ambiente di testing da predisporre, oltretutto disporre in anticipo di tutte le risorse non è essenziale: è infatti possibile aggiungerle quando necessario, direttamente dalla DashBoard dello spazio cliente, in questo modo la gestione del budget è più semplice. Ovviamente, in ambiente di produzione, quando si suppone che il traffico veicolato sarà decisamente maggiore con centinaia di host, una VPS con specifiche più generose di certo garantirà un servizio migliore per l'utente finale, ma questi aspetti esulano dal setup progettuale, per cui le risorse a disposizione con la macchina virtuale opzionata sono più che abbondanti.
Dal punto di vista tecnico, tutte le VPS offerte sono basate su architetture Intel di ultima generazione, con storage NVMe ed è possibile opzionare un'ampia scelta di distribuzioni Linux preinstallate, così come di interfaccie di gestione web. Di seguito (Figura 1.2) sono riepilogate le

Processore	1 vCore
Memoria	2 GB
Storage	40 GB SSD NVMe
Banda passante	250 Mbps illimitato*
Anti-DDos	✓
KVM	✓
Accesso root	✓
API	✓
Indirizzo IPv4	1
Indirizzo IPv6	/128
Localizzazione	8
Monitoraggio e interventi	24/7
SLA	99.9%

Opzioni disponibili

Indirizzo IPv4	2 € +IVA/IP cioè 2,44 € IVA incl./IP			
Snapshots	✓	✓	✓	✓
Dischi aggiuntivi	✓	✓	✓	✓
Backup automatizzato	✓	✓	✓	✓

Figura 1.2: Caratteristiche e opzioni agguntive disponibili della soluzione VPS acquistata (link)

principali caratteristiche del servizio acquistato. Tra le feature incluse spicca la banda passante al VPS, che è garantita e si riferisce alla velocità di trasmissione minima assegnata, inoltre è incluso il sistema di protezione *anti-DDoS OVHcloud*.
Una vasta gamma di sistemi operativi sono equipaggiabili per la macchina virtuale, quali Windows Server, Debian, Fedora, CentOS ed Ubuntu, in particolare si è scelto di adottare **Ubuntu 16.04 LTS**. Inoltre è da sottolineare che il *Service Level Agreement* per il servizio scelto è caratterizzato da un tasso di disponibilità mensile pari al 99,9%.
Una volta finalizzato il pagamento, viene fornito l'accesso alla dashboard principale del servizio scelto tramite lo spazio utente (Figura 1.3). Da qui, è possibile accedere al servizio acquistato direttamente con un *click* nella sezione server, ma anche alle ultime fatture pagate, nonchè andare a gestire tutto ciò che riguarda l'account utente con il pannello sulla destra della schermata. A partire dalla dashboard si individuano due aree diverse, una per la gestione del nostro account e le informazioni personali, l'altra per la gestione dei servizi acquistati. La dashboard per la gestione delle informazioni personali dell'utente permette di scaricare le fatture dei pagamenti effettuati i servizi attivi, con lo stato di funzionamento, e la modalità di rinnovo (manuale o automatica), inoltre è possibile modificare, aggiungere e rimuovere i metodi di pagamento ed infine di aprire e gestire eventuali ticket, sia per assistenza tecnica che assistenza commerciale, con i relativi contatti di supporto. Per quanto riguarda invece la gestione dei servizi acquistati come appunto VPS, server dedicati ecc., possiamo accedere, a partire dalla dashboard principale sotto la voce *My product and services*, ad un pannello di controllo generale dove è possibile selezionare quale prodotto, tra quelli acquistati, andare a gestire. Possiamo fare click direttamente

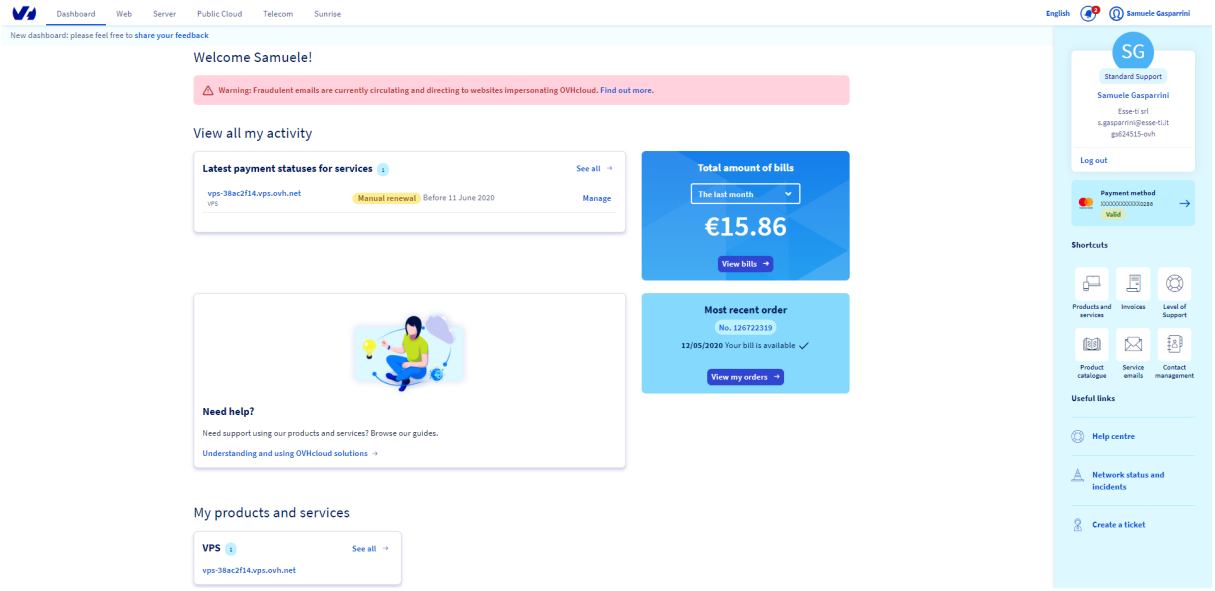

Figura 1.3: Dashboard OVHCloud

sull'hostname della VPS acquistata e verremo reindirizzati al pannello di riepilogo (Figura 1.4).

Esso presenta tutte le informazioni di cui possiamo aver bisogno per la gestione della nostra macchina virtuale, in particolare:

- informazioni sul sistema operativo;
- localizzazione fisica della *server farm* presso cui la VPS è installata;
- l'indirizzo IP attraverso il quale è possibile accede alla macchina remota tramite il protocollo sicuro SSH;
- un riepilogo delle opzioni attivate o disattivate che sono state opzionate all'acquisto della macchina, così come il piano attuale.

Come già evidenziato, con un paio di *click* è possibile fornire più risorse hardware, in termini di RAM o storage, alla nostra VPS, è inoltre possibile aumentare il numero di unità elaborative passando ad una VPS di livello superiore.

1.3 Caratteristiche del gateway adottato dall'azienda

L'azienda Esse-ti S.r.l. ha fornito due gateway con funzionalità di router identici, modello *4G.Router* (Figura 1.5), ognuno corredato di una SIM per la connessione all'operatore radiomobile partner. Il modello in questione offre connettività Internet e consente il telecontrollo dei dispositivi connessi via Wi-Fi, porta LAN o porta seriale. Ad oggi l'azienda impiega questo dispositivo nel panorama dell'IoT applicato al settore dell'elevazione, con funzionalità che spaziano in molti ambiti per garantire la massima flessibilità d'impiego:

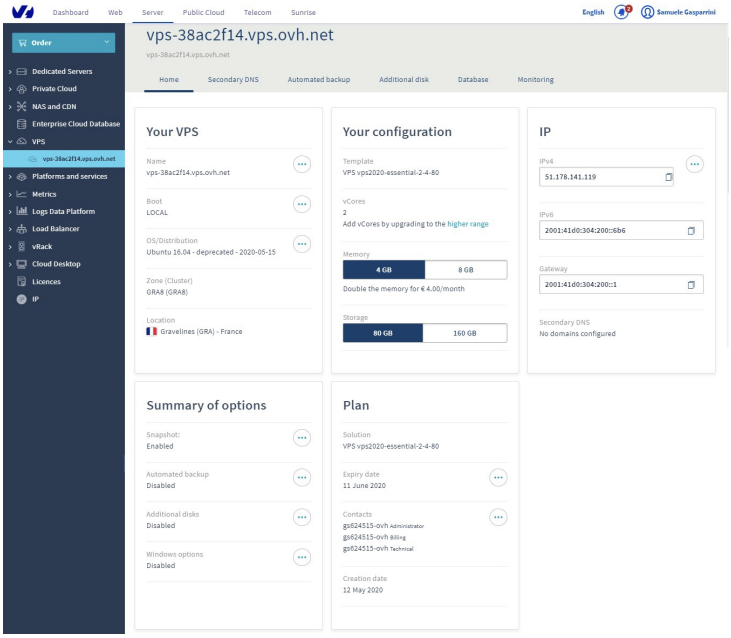

Figura 1.4: Pannello di controllo del servizio VPS acquistato

- Access Point wireless per offrire connettività Internet Wi-Fi a dispositivi wireless;
- Client Dynamic DNS per consentire all'utente di raggiungere da remoto, tramite Internet, il router stesso e tutti i dispositivi connessi via Wi-Fi o porta LAN;
- Trasmissione dati in standard RS-232/RS-485/CAN-bus per consentire all'utente di monitorare da remoto, tramite servizio COMNet, i dispositivi connessi alla porta seriale, oppure per consentire ai dispositivi connessi alla porta seriale di inviare automaticamente dati, segnalazioni, notifiche tramite servizio COMNet;
- Gateway telefonico per consentire l'invio e la ricezione di chiamate attraverso la rete 4G LTE/UMTS/GSM a telefoni fissi, combinatori o altri dispositivi telefonici collegati all'ingresso FXS, con la possibilità di visualizzare l'identificativo del chiamate;
- Gestione servizio roaming;
- Ingressi digitali programmabili, configurabili anche con antifurti tecnologici;
- Uscite relè attivabili localmente o via SMS, possono anche segnalare eventi come la mancata alimentazione e l'assenza di segnale radiomobile;
- Programmazione locale o remota del gateway telefonico tramite telefono (toni DTMF) o via SMS;
- Lettura programmazione via SMS;
- Invio di segnalazioni ed avvisi tecnici tramite SMS per il controllo della scandenza della scheda SIM, stato della batteria e dell'alimentazione esterna;
- Batterie interne di backup per garantire il funzionamento anche in assenza di alimentazione;
- Aggiornamento firmare *over-the-air*.

Figura 1.5: Gateway 4G.Router fornito ai clienti Esse-ti (link)

Le caratteristiche hardware del gateway includono:

- Modulo LTE Cat 1 Penta-Band / UMTS HSPA+ Dual-Band / GSM Dual-Band
- Frequenze LTE (700/800/900/1800/2100 MHz) / UMTS HSPA+ (900/2100 MHz) / GSM (900/1800 MHz);
- Velocità LTE Cat 1, download max. 10,2 Mbps / upload max. 5,2 Mbps;
- Wi-Fi 2.4 GHz - IEEE 802.11b/g/n con supporto ai protocolli di sicurezza WEP, WPA, WPA2, WPA-WPA2, WPA-WPA2-AES;
- Dotazione di ingressi e uscite:
  - Porta LAN con ingresso RJ45 10/100 Mbps;
  - Porta FXS (morsetto);
  - Morsettiera per trasmissione dati in standard RS-232, RS-485 e CAN-bus;
  - Due Uscite relè (NA e NA/NC; 1 A 24 V);
  - Due Ingressi optoisolati per allarmi tecnologici;
  - Porta micro USB A/B per connessione a pc;
  - Alloggiamento SIM Card;
  - Connettore antenne esterne SMA.
- Molteplici LED di segnalazione per stato del dispositivo, trasmissione dati COMNet, stato dell'alimentazione, livello del segnale radiomobile ricevuto, stato della linea telefonica connessa alla porta FXS;
- Alimentazione da 11 a 26 Vdc con apposito morsetto o tramite jack per alimentatore esterno 100-240 Vac;
- Batteria di backup a tecnologia Ni-MH con capacità di 800mAh a 7,2V che garantisce fino a 8 ore di funzionamento in stand-by o 2 ore di funzionamento attivo.

Capitolo 2

Architettura di rete

2.1 Modello a strati

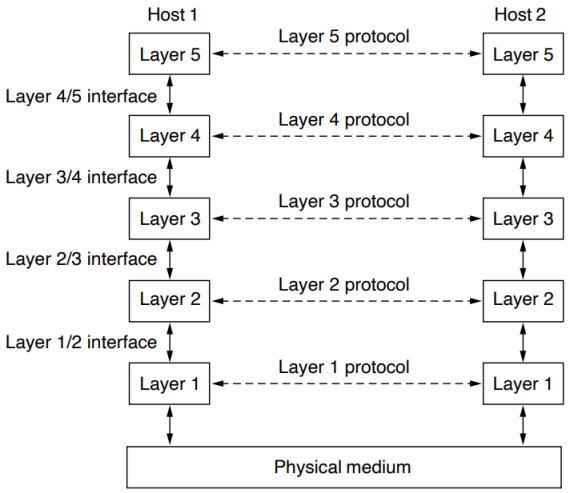

Figura 2.1: Ipotetico modello a strati a 5 livelli con protocolli ed interfacce (link)

A partire dallo topologia di principio che schematizza la comunicazione desiderata in Figura 1.1 è fondamentale comprendere come avviene la trasmissione di dati tra due nodi attraverso una rete. Nel caso particolare di progetto, la comunicazione avviene attraverso la rete globale con dei determinati accorgimenti e protocolli che vengono adottati dagli ISP per le connessioni radiomobili. Si andranno ad evidenziare problemi ed ostacoli che rendono necessari l'adozione del server remoto e del software OpenVPN per veicolare le informazioni in maniera efficace e sicura.
La realizzazione ed il funzionamento di una *"rete di reti"* quale è Internet coinvolge numerose applicazioni e protocolli a livello software ma anche mezzi fisici trasmissivi ed elaboratori hardware, che possono essere gli apparati che gestiscono l'inoltro fisico dell'informazione verso le destinazioni desiderate, oppure i dispositivi degli utenti finali che fruiscono dei dati trasmessi. Affrontare la complessità di tutti questi elementi che devono interagire tra loro, pur garantendo affidabilità e sicurezza, richiede la definizione di una **architettura di rete**, ossia di un insie-

me di specifiche secondo le quali viene realizzata la comunicazione, dettagliando i componenti hardware, software e le funzionalità svolte. In particolare, si è deciso di adottare un'architettura basata su un modello a strati, sfruttando la metodologia *top-down*, dove la complessità vista viene suddivisa in molti sottoproblemi ognuno dedito a svolgere un compito diverso, ciò risulta essere un approccio decisamente più manegevole e modulare rispetto alla realizzazione di un modello di comunicazione monolitico.
Nell'ambito di un'architettura di rete che segue il modello a strati (Figura 2.1), devono essere specificati i protocolli da usare perchè la comunicazione abbia luogo: è necessario definire le modalità di comunicazione tra i vari livelli, specificando la tipologia e la modalità di scambio dei dati. Due host che vogliono comunicare realizzano la stessa architettura a strati, implementano uno o più protocolli ad ogni livello per consentire lo scambio di informazioni tra **peer level**, ossia tra un livello n del mittente e lo stesso livello n del destinatario. Ogni strato è costituito da più *entity*, ossia elementi attivi in grado di inviare e ricevere informazioni, le entità relative a livelli paritari sono chiamate **peer entity**.
In riferimento a tale modello un **protocollo** è un insieme di regole che definiscono le modalità logiche di comunicazione tra due peer entity, specificando le informazioni di controllo da utilizzare per la gestione della comunicazione affinché il trasferimento dei dati vada a buon fine.
Benché la trasmissione a livello logico sia diretta, il messaggio che viene trasmesso da un livello N a partire dall'*Host 1* al suo pari (secondo le regole definite dal protocollo) arriva a destinazione per mezzo della comunicazione con il livello inferiore, che a sua volta realizza una trasmissione diretta con il suo peer entity rivolgendosi al livello inferiori ecc. fino ad arrivare al mezzo fisico, i cui strati relativi effettuano una comunicazione diretta fisica. Con modalità del tutto simili a quanto spiegato, l'informazione passerà dal livello più basso dell'*Host 2* via via allo strato superiore, finchè il messaggio generato inizialmente raggiungerà il livello N a realizzare la comunicazione desiderata in principio. Possiamo notare come, nel procedimento analizzato, sia fondamentale definire l' interazione fisica tra livelli adiacenti:

- ogni strato interagisce solo con i due adiacenti;
- l'interazione avviene per mezzo di un'**interfaccia** che definisce delle regole di comunicazione;
- ogni livello, grazie alle interfacce di comunicazione, fornisce **servizi** al livello superiore e usufruisce di diversi **servizi** dal livello sottostante a realizzare rapporto che ricalca la relazione client-server;
- non è necessario che un livello conosca i dettagli delle implementazioni funzionali degli strati con cui vuole comunicare, basta attenersi alle regole definite dall'interfaccia per la comunicazione.

È importante sottolineare ulteriormente la differenza tra il concetto di servizio e protocollo, il primo riguarda la comunicazione fisica tra strati, il secondo la comunicazione logica tra peer entity (Figura 2.2). I servizi offerti, definiti a livello architetturale, man mano che si procede dal livello più basso al livello più alto risultano essere sempre più astratti e slegati dalla effettiva trasmissione dell'informazione sul mezzo fisico, oltretutto il principio dell'*information hiding* garantito dall'uso delle interfacce tra ogni strato permette modificare il funzionamento di un singolo strato, apportando ad esempio delle ottimizzazione nell'elaborazione, senza compromettere la comunicazione diretta logica tra peer entity.

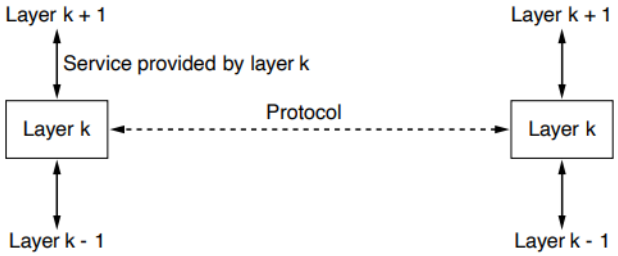

Figura 2.2: Differenza tra servizi e protocolli (link)

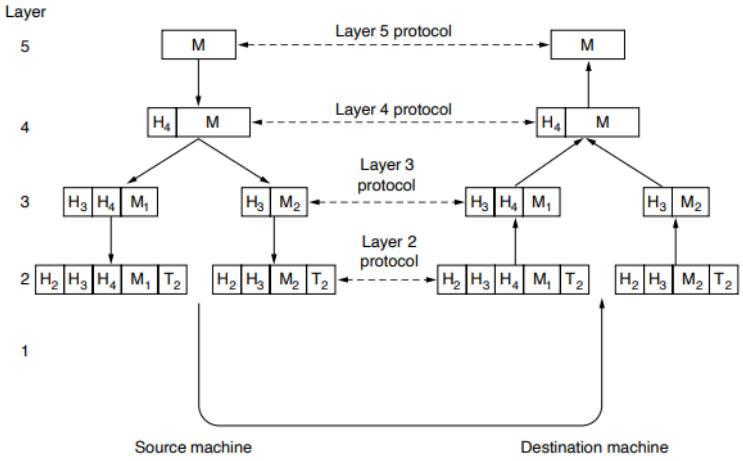

Figura 2.3: Flusso delle informazioni a realizzare una comunicazione virtuale a partire dal *layer 5* (link)

Incapsulamento

Per quanto analizzato finora, si sono individuate due diverse modalità di comunicazione che devono essere implementate nel modello a strati che realizza l'architettura di rete:

- la comunicazione logica diretta tra peer entity;
- la comunicazione fisica tra livelli adiacenti.

L'interazione fisica tra livelli adiacenti permette di realizzare l'astrazione necessaria a creare una comunicazione logica diretta tra peer entity attraverso il meccanismo dell'**incapsulamento**. Supponendo la necessità di inviare un messaggio da una sorgente ad un destinatario, l'informazione attraverserà in trasmissione livelli via via inferiori dello *stack* che realizza il mittente, verrà propagata attraverso un mezzo fisico ed in ricezione attraverserà livelli man mano superiori dello stack del destinatario fino ad essere effettivamente consegnata (Figura 2.3). Considerando un generico strato N dello stack di trasmissione, l'informazione viene ricevuta dal livello superiore, elaborata aggiugendovi un insieme di istruzioni, detto *header*, utili per lo svolgimento delle funzioni relative a quel livello (è ciò che permette la comunicazione diretta logica, le informazioni aggiunte verranno elaborato dal pari livello del destinatario), verrà quindi inoltrato al livello sottostante, che considera il blocco di dati ricevuto, costituito da informazioni e dall'header,

come un unico pacchetto dati che viene elaborato, unito ad un'altro header, ed inoltrato. Nello stack di ricezione, il livello N riceve un blocco dati da cui è in grado di estrarre l'header aggiunto dal pari strato in tramissione, questo verrà elaborato a realizzare la comunicazione diretta logica tra i due peer entity, mentre il resto delle informazioni vengono inoltrate al livello sovrastante.

2.2 Modello di riferimento ISO/OSI

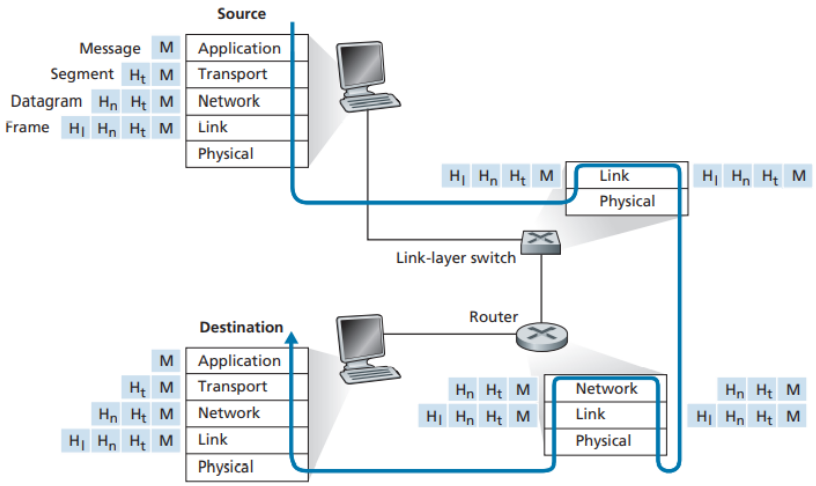

Figura 2.4: Modello di comunicazione tra due end point che evidenzia l'architettura TCP/IP implementata con un modello a strati basato sul principio dell'incapsulamento (link)

La definizione di un'architettura di rete basata su un modello a strati corredato da un insieme di protocolli e servizi risulta un'astrazione fondamentale da adottare nel design delle reti, in particolare bisogna dettagliare e standardizzare le funzionalità di ogni livello e le interazioni tra di loro mediante delle precise interfacce. l'**ISO** è l'ente internazionale di standardizzazione che per primo cercò di definire formalmente una modalità per interconnettere gli elaboratori, così nella seconda metà degli anni *'70* arrivò a specificare un modello, chiamato **OSI** (*Open Systems Interconnected*). Il modello ISO/OSI (Figura 2.5) è una struttura astratta che suddivide in sette livelli le funzionalità necessarie alla realizzazione di reti senza però andare a specificare una suite di protocolli, bensì dettagliando come devono essere organizzati i livelli, le interfacce e i servizi che ogni strato deve offrire a quelli adiacenti. Questa sua particolare caratteristica lo ha reso il modello di riferimento che tutt'ora viene adottato a livello globale. ISO/OSI si distingue per una precisa suddivisione tra i concetti di servizi, interfacce, protocolli, inoltre ogni strato viene individuato in base alla necessità di un differente livello di astrazione ed è dedito ad una definita funzionalità. Verificando brevemente i sette strati del modello, possiamo individuare tre gruppi:

- i primi tre livelli fanno riferimento alla rete, difatti sono detti **network oriented layers**;
- il quarto livello separa l'ambiente rete dall'ambiente applicativo;
- gli ultimi tre livelli fanno riferimento all'applicazione, difatti sono detti **application oriented layers**;

In particolare possiamo descrivere brevemente ogni livello come segue:

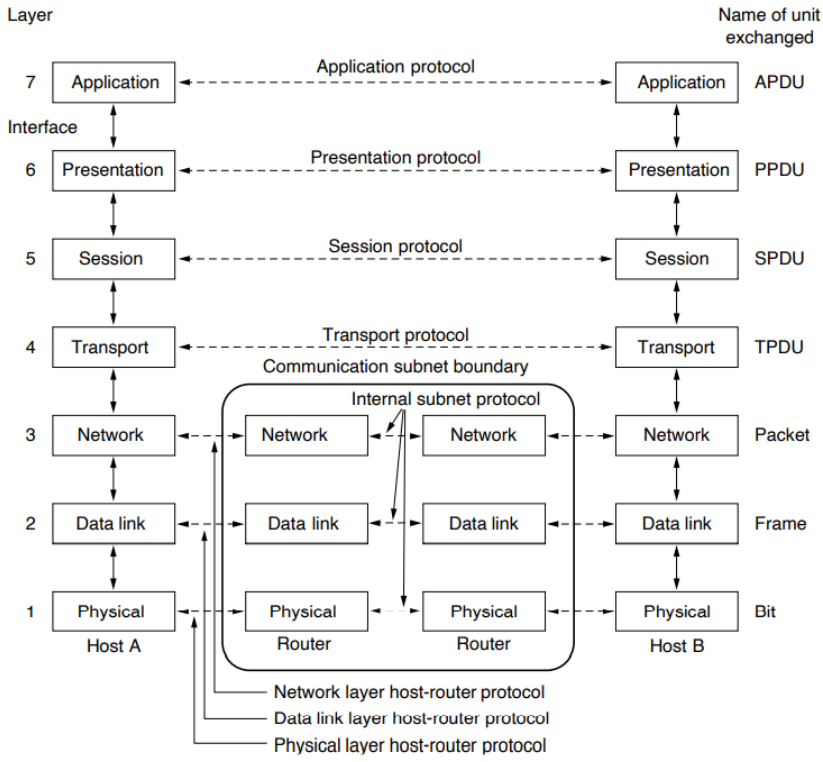

Figure 1-32. The OSI reference model.

Figura 2.5: Comunicazione tra due host utilizzando il modello di riferimento ISO/OSI (link)

- **Layer 7**. **Application Layer**, si occupa delle applicazioni che usano la rete;
- **Layer 6**. **Presentation Layer**, fornisce una rappresentazione standard dei dati delle applicazioni;
- **Layer 5**. **Session Layer**, gestisce le sessioni tra le applicazioni;
- **Layer 4**. **Transport Layer**, fornisce la connessione end-to-end con controllo della congestione;
- **Layer 3**. **Network Layer**, gestisce la connessione alla rete;
- **Layer 2**. **Data Link Layer**, provvede alla trasmissione dei dati sulla rete fisica;
- **Layer 1**. **Physical Layer**, definisce le caratteristiche fisiche della rete;

2.3 Architettura TCP/IP

Mentre il modello ISO/OSI risulta essere l'esito di un successivo sforzo dell'ente di standardizzazione ISO, nelle prime reti realizzate per scopi militari e di ricerca venne inizalmente adottato un modello differente, frutto di uno studio finanziato da un'agenzia governativa degli Stati Uniti: **l'architettura TCP/IP**. Tale modello non distingue chiaramente le funzionalità in riferimento ai layer, in particolar modo quelli dediti alla trasmissione fisica, anzi non vengono nemmeno evidenziati e distinti i concetti di servizi, interfacce e protocolli. Risulta essere un modello poco generale, corredato da protocolli *ad hoc* che difficilmente può essere impiegato come riferimento

in altri ambiti. Nonostante i difetti, con la rapida diffusione delle reti anche nel mondo civile e la nascita di Internet, questa architettura diventò velocemente lo standard *de facto* essendo **ready-to-use**, completa di una suite di protocolli funzionanti e già implementati nell'hardware in commercio, oltretutto la struttura del modello a strati ricalca, al netto dei difetti evidenziati, quanto già visto nel modello ISO/OSI. Quest'ultimo non era comunque esente da difetti, essendo ritenuto da molti troppo complesso e teorico, con strati quali Presentation e Session che risultavano più figurativi che funzionali. Queste problematiche comportarono scarse implementazioni, per di più inefficienti ed incomplete. Per questi motivi, seppur l'architettura TCP/IP fosse ritenuta una soluzione temporanea in vista dell'adozione mondiale di un'implementazione del modello ISO/OSI, ogni tentativo di transizione risultò inutile ed eventualmente tale intenzione venne abbandonata, lasciando la distinzione tra modello di riferimento astratto per il funzionamento delle reti di elaboratori, e l'architettura effettivamente implementata, corredata di una suite di protocolli, nella realizzazione delle stesse.

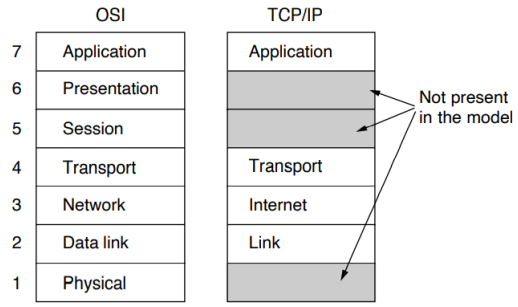

Figura 2.6: Confronto tra ISO-OSI e TCP/IP (link)

L'architettura TCP/IP prende il nome dai suoi due protocolli più importanti: **IP** e **TCP**, spesso si fa riferimento a essa con il nome di *Internet protocol stack* ad indicare la struttura a strati dell'architettura e i protocolli definiti a ogni livello. Come si nota dalla Figura 2.6, questa architettura si basa solo su quattro strati che, con le dovute differenze, corrispondono ai sette livelli del modello OSI, quest'ultimo specifica in modo rigoroso le funzioni che devono essere svolte da ciascuno strato, mentre i livelli del TCP/IP contengono protocolli relativamente indipendenti che possono essere usati a seconda delle necessità. Di seguito vengono riassunte le funzioni e i protocolli che caratterizzano ogni strato.

2.3.1 Layer Application

Fornisce dei servizi di connettività alle applicazioni *user oriented*, ad esempio browser e client e-mail mediante dei protocolli ben definiti, tra cui possiamo evidenziare i seguenti:

- **TELNET**, protocollo client-server per impartire comandi a maccchine remote tramite terminali virtuali;
- **FTP**, permette il trasferimento di file tra due stazioni remote;
- **HTTP**, permette la fruizione di documenti ipertestuali;
- **SMTP**, permette la trasmissione e la gestione di e-mail;
- **DNS**, sistema usato per fornire dei nomi di dominio ai nodi della rete.

Lo strato applicativo racchiude le funzionalità corrispondenti ai tre layer superiori del modello ISO/OSI, in particolare le primitive presenti negli strati Presentation e Session si suppone siano raramente necessarie ed eventualmente incluse negli applicativi finali. La mole di dati prodotta dai protocolli che caratterizzano questo livello è raccolta in un'unico contenitore denominato **messaggio**.

2.3.2 Layer Transport

Fornisce servizi volti al trasporto dei messaggi tra layer applicativi di stazioni remote, garantendo una correzione degli errori. Questo strato è caratterizzato da due protocolli di trasporto, **Transmission Control Protocol** ed **User Datagram Protocol**, che svolgono la stessa funzionalità con modi e obiettivi differenti.
TCP sfrutta la comunicazione logica diretta con il suo peer entity di destinazione per instaurare una certa connessione che possa garantire un determinato livello di qualità, così da soddisfare requisiti di raggiungibilità, garanzia di consegna del messaggio e controllo di flusso, per regolare e portare a termine la trasmissione anche in presenza di reti congestionate. In virtù di tale approccio, il protocollo TCP è un protocollo *connection-oriented*, realizza le funzioni esplorate attraverso un set di primitive, basate sui protocolli dei layer inferiori, che regolano la comunicazione tra pari, l'instaurazione di una connessione prevede una procedura di sincronizzazione tra le due stazioni remote chiamato **three-way handshake** (Figura 2.7). TCP prevede una procedura simile anche per portare a termine la connessione stabilita, una volta che il trasferimento è terminato. Il secondo protocollo in questo strato, UDP, adotta un approccio totalmente

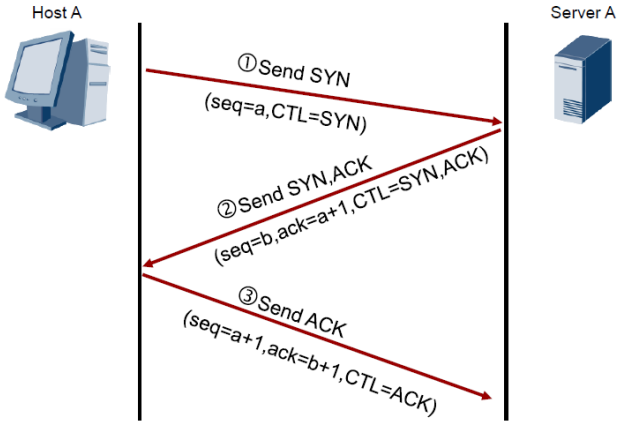

Figura 2.7: Three-way handshake per instaurare una connessione TCP

opposto a quanto visto con TCP, esso infatti risulta fornire un servizio *connectionless* alle applicazioni: si perde tutta la procedura di sincronizzazione che precede la trasmissione dei dati, così come ogni garanzia di consegna dell'informazione o di eventuale controllo di flusso, in favore di una consegna più veloce dell'informazione, difatti le procedure di sincronizzazione nei protocolli connection oriented possono risultare inefficienti per un traffico sensibile ai ritardi, quali ad esempio flussi video in tempo reale. L'approccio minimalista del protocollo UDP, con strutture ed operazioni semplificate, viene ad oggi largamente usato per lo scambio di singoli messaggi nel modello client-server (Figura 2.8). In questo strato è convenzione rifersi al blocco

Figura 2.8: UDP usato per singole interazioni *request-reply*

di dati elaborato con il nome di **segmento** per il protocollo TCP, altrimenti con il nome di **datagramma** per il protocollo UDP.

2.3.3 Layer Internet

Questo livello, comunemente chiamato anche *Network Layer* risulta essere fondamentale, permette difatti agli host di inoltrare **pacchetti** in qualsiasi rete e farli viaggiare in maniera indipendente fino alla loro destinazione, che sia un host della stessa rete o un host su una rete differente. I pacchetti potrebbero essere consegnati al destinatario in un ordine completamente differente rispetto a quello d'invio, se necessario sarà compito dei protocolli del layer superiore riordinarli, cui è lasciata anche l'eventuale gestione di problematiche di congestione. Il protocollo che si occupa di inoltrare i pacchetti alla loro destinazione è detto **IP Protocol**, possiamo riassumerne le principali in due obiettivi:

- Il **routing** dei pacchetti, ossia il meccanismo che permette al traffico di una determinata rete locale di essere trasmesso ad altre reti geograficamente remote attraverso un determinato percorso che coinvolge dei nodi di rete fino al raggiungimento dell'end point finale della comunicazione;
- La **frammentazione**, che si riferisce alla suddivisione dei dati da trasmettere in blocchi più maneggevoli in termini di dimensione e struttura, che saranno poi trasmessi sulla rete.

Ogni nodo che rispetta lo stack TCP/IP deve implementare il protocollo IP e sarà quindi caratterizzato da un certo indirizzo logico univoco, detto *indrizzo IP* (Figura 2.10) nel contesto della rete in cui risiede. Il protocollo IP definisce una precisa struttura dell'header (Figura 2.9) che compone il pacchetto IP, esso definisce come il traffico viene instradato attraverso le reti: elaboratori o apparati di rete, provvederanno a modificare determinati campi così da permettere la trasmissione del pacchetto al successivo nodo della rete individuato da un diverso indirizzo.
Il protocollo IP è fondamentale per lo svolgimento delle funzionalità del layer Network, ma sono presenti altri protocolli funzionali alle operazioni di routing, in particolare si distingue **Internet Control Message Protocol**. ICMP lavora affiancando IP nella gestione delle reti, esso implementa un sistema di messaggistica che va a compensare la limitata affidabilità di IP, difatti viene usato, oltre che a supporto del routing, anche per scopi diagnostici e di *error reporting* nel processo di elaborazione dei pacchetti. Vengono scambiati messaggi di notifica e feedback tra i gateway e gli host sorgente riguardo eventuali problemi di trasmissione nell'ambiente di

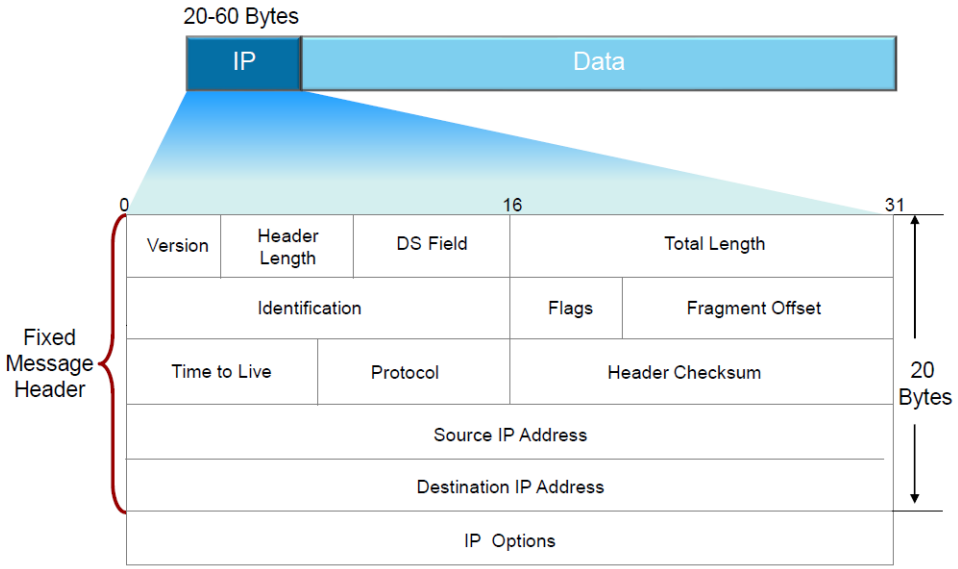

Figura 2.9: Struttura dell'header IP per il protocollo IPv4

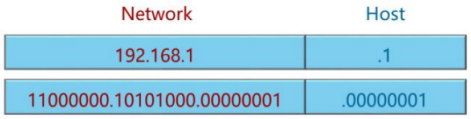

Figura 2.10: Esempio di indirizzo IPv4

comunicazione, pur tenendo presente che queste notifiche non cambiano la natura *best effort* del protocollo IP e non forniscono alcun tipo di garanzia sulla ricezione del pacchetto.

2.3.4 Network Interface

Include tutte le primitive necessarie per l'interfacciamento con i dispositivi fisici, costituisce perciò un'interfaccia tra host e mezzi fisici trasmissivi che inizialmente era stata ignorata nelle prime release dell'architettura TCP/IP. Successivamente i miglioramenti apportati al modello e la necessità di chiarire le funzionalità incluse in questo livello, similarmente a quanto fatto nello stack ISO/OSI, hanno portato ad una modellazione più vicina alla realtà suddividendo tale strato in due ulteriori sottostrati.

Data Link

Questo livello definisce una precisa organizzazione dell'informazione per strutturarla per la comunicazione, intesa come scambio di dati tra nodi che insistono nella stessa rete. I dati vengono raggruppati in un **Frame** (Figura 2.12) di un certo tipo in base allo specifico protocollo adottato, in particolare il frame viene arricchito con delle informazioni hardware che inviduano in maniera univoca nel contesto della rete locale destinatario e ricevente (viene usato il **MAC address**, ossia l'indirizzo fisico univoco e non modificabile delle schede di rete degli elaboratori).

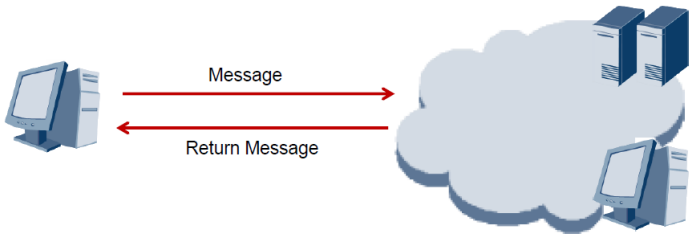

Figura 2.11: Protocollo ICMP basato su messaggi di notifica e feedback

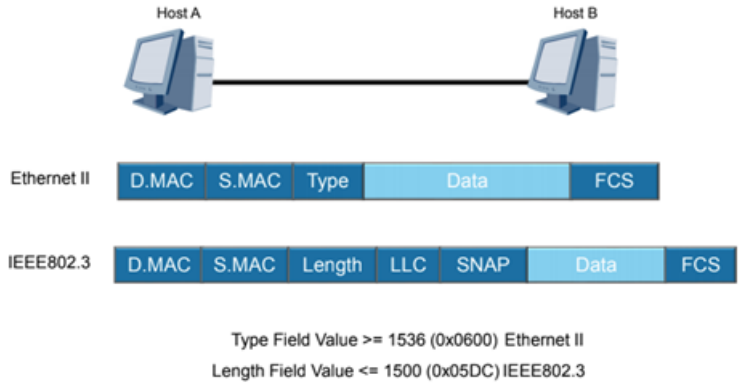

Figura 2.12: Struttura del frame data link che evidenzia le differenze tra l'adozione di incapsulamenti diversi

Alcuni esempi di protocolli impiegati in questo livello sono:

- Ethernet II;
- Ethernet 802.3;
- WiFi, ossia IEEE 802.11 nei suoi vari emendamenti;
- Point to Point Protocol;
- DOCSIS protocol.

I frame possono essere veicolati su una rete locale composta da diversi host secondo tre diverse modalità:

- **Unicast** *uno ad uno*: si realizza una comunicazione diretta da un singolo nodo ad un'altro, una volta che il frame viene inoltrato sull'interfaccia di rete fisica del mittente, viene trasmesso attraverso la rete all'interfaccia del destinatario che provvederà ad elaborarlo. Se ci sono più host in un dominio di collisione unico (Figura 2.13), tutti ricevono il frame, ma solo il destinatario lo processa, gli altri nodi lo scartano senza ulteriori elaborazioni;

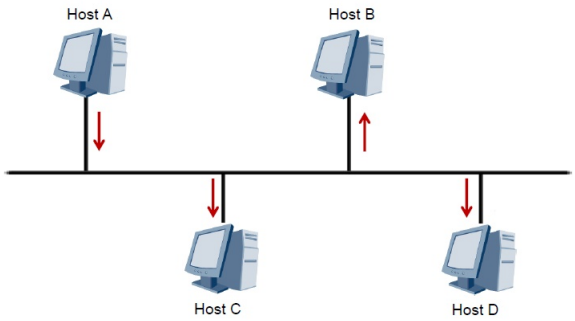

Figura 2.13: Diversi host attestati in un unico dominio di collisione elettrico

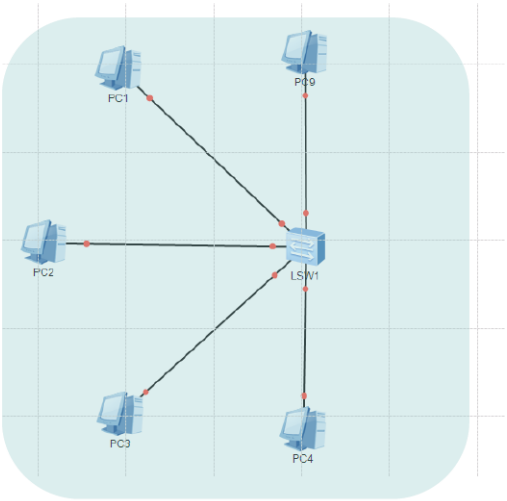

Figura 2.14: Diversi host attestati in un unico dominio di broadcast che possono comunicare direttamente scambiandosi frame, esiste invece un dominio di collisione elettrico separato su ogni porta dello *switch*

- **Broadcast** *uno a tutti*: permette di indirizzare i frame di una singola sorgente a tutti i nodi che insistono su una stessa rete locale, che definisce un dominio di broadcast[1] (Figura 2.14). Tutti i nodi ricevono e processano il frame ricevuto. il traffico di broadcast è usato in numerosi protocolli per scoprire nuovi nodi della rete e attuare operazioni di manutenzione, ma la procedura può risultare molto onerosa specie per estesi domini di broadcast;
- **Multicast** *uno a molti*: permette di realizzare un broadcast selettivo più efficiente tanto da sostituire l'uso del broadcast in alcuni contesti. il frame raggiunge molti nodi ma non tutti.

Si è evidenziato come la comunicazione tra dispositivi nella stesso dominio di broadcast avviene con un frame cui i campi relativi agli indirizzi fisici vengono appositamente popolati dal mittente della trasmissione. L'host di sorgente viene a conoscenza dell'indrizzo fisico del destinatario mediante **ARP**: è un protocollo che caratterizza il layer Data Link che consente di apprendere dinamicamente il Mac Address di un dispositivo che si trova sulla nostra stessa rete locale

[1]Mentre un dominio di collisione elettrico è anche un dominio di broadcast, il viceversa non vale nel caso di una rete con topologia a stella dotata di Switch.

conoscendone l'indrizzo di Livello Network relativo, il protocollo andrà a costruire una tabella di mappatura tra indirizzi IP ed indirizzi fisici.

Physical

Definisce le specifiche di cavi e connettori, così come i livelli dei segnali che transitano su di essi, in generale approfondisce ogni aspetto che afferisce al mezzo tramissivo della comunicazione che sostiene il trasferimento dei singoli bit da un nodo all'altro. I protocolli che caratterizzano questo livello dipendono dal collegamento fisico impiegato e specificano ognuno in maniera diversa rispetto a costruzione e materiali, la trasmissione del flusso binario.

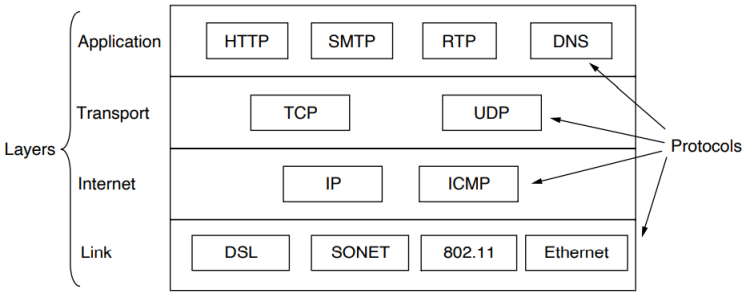

Figura 2.15: Schema riepilogativo dei protocolli principali che sono implementati nel modello TCP/IP (link)

Volendo fornire un breve riepilogo, si è evidenziato come attraverso la suite di protocolli Internet è possibile connettere tra loro vari tipi di reti assumendo solo che siano in grado di trasferire dati, senza entrare nel merito di come tale trasferimento è effettuato. Lo stack TCP/IP è diventato l'architettura ufficiale di Internet e nel corso degli anni è evoluto arrivando alla versione 6:

- versione **4**. È ancora oggi molto diffusa, si basa sulla suddivisione in classi degli indirizzi IP che sono in un formato a 32 bit; la diffusione rapidissima di Internet ha portato, però, alla carenza di indirizzi e alla difficoltà di soddisfare nuove richieste;
- versione **5**. È una proposta basata sul modello OSI che non è mai stata portata avanti come già evidenziato, comportava troppe modifiche da effettuare sugli strati e costi elevati;
- versione **6**. Le modifiche riguardano principalmente lo strato Network, viene introdotto un nuovo sistema d'indirizzamento basato su 128 bit che consentono di gestire un numero elevatissimo di utenti.

Capitolo 3

Data Forwarding scenario

3.1 Approfondimento sull'indirizzamento IP

Al fine di comprendere il funzionamento dell'architettura TCP/IP e capire come avvenga effettivamente la comunicazione tra due end point, è possibile verificare, passo per passo, come i dati viaggiano dal mittente ad destinatario in una topologia semplificata. L'analisi dei dettagli del processo di *data forwarding* evidenzia ogni dettaglio del comportamento dei protocolli, in particolare per l'IP protocol, così come del processo di incapsulamento e decapsulamento, queste sono conoscenze necessarie alla creazione di topologie di rete efficaci ed al *troubleshooting* di eventuali problemi che possono sorgere e sono risultate essere delle capacità fondamentali per completare l'obiettivo del progetto.

Per procedere con l'analisi del forwarding dei dati in una topologia semplificata, è necessario fare alcune precisazioni sull'indirizzamento IP. Come visto nel Capitolo 2, una rete locale è caratterizzata da più dispositivi che insistono nello stesso dominio di broadcast e la comunicazione tra gli elaboratori, basata sugli indirizzi fisici degli stessi, può avvenire attraverso lo scambio di frame che contengono, nel campo data, le informazioni riferite ai livelli superiori dello stack che hanno richiesto la connessione. Al fine di mantenere le dimensioni delle reti locali contenute, dato che le prestazioni decrescono all'aumentare del numero di host (in termini tecnici il protocollo Ethernet risulta essere poco *scalabile*), i domini di broadcast, quindi la rete locale iniziale, vengono segmentati tramite l'uso di apparati che processano i pacchetti, detti **router**, che lavorano al Layer 3 dello stack TCP/IP. Avendo reti locali differenti, si pone ora il problema della raggiungibilità tra host che risultano remoti, ed è qui che entra in gioco l'indirizzamento logico fornito dal Layer Network, esso garantisce un livello di astrazione più alto e permette di dare una **connotazione geografica** al blocco di dati che deve essere inviato. Riprendendo il concetto di rete e focalizzando lo scopo del Layer Network, dal dominio di broadcast si è definito un concetto logico, una rete come insieme di nodi associati ad uno stesso riferimento. Tali gruppi posso essere creati sulla base dell'appartenenza dei nodi ad una stessa organizzazione, piuttosto che sulla loro comune locazione geografica e permettono una diversificazione funzionale. Il funzionamento dei router si basa sull'analisi del campo *Type* che caratterizza i frame che esso riceve all'attenzione di una sua interfaccia fisica, sulla quale è attestata un rete locale. Tale campo indica, attravero un apposito codice esadecimale, il protocollo di Livello Network che dovrà processare il contenuto del campo Data, anche detto *payload*, contenuto frame ricevuto. Nell'esempio posto in Figura 3.1, il codice decimale *0x0800* indica che il *payload* del frame risulta essere un pacchetto IP, caratterizzato da un header (Figura 2.9) che và processato per effettua-

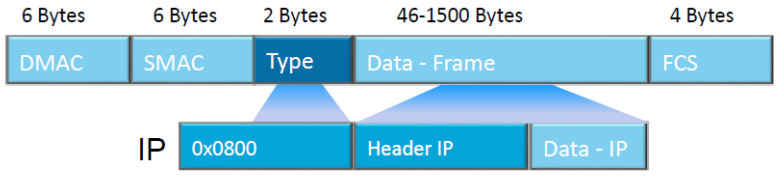

Figura 3.1: Struttura di un frame Ethernet II dove si evidenza il contenuto del campo *Type*

re operazioni di routing e frammentazione. L'header IP definisce perciò come il traffico viene instradato attraverso le reti. È importante notare come il protocollo IP non è l'unico protocollo layer 3, benchè risulta essere sicuramente d'importanza fondamentale, nel campo type potremo trovare specificati altri codici esadecimali che indicano il set d'istruzioni che dovrà gestire i dati di quel pacchetto, ad esempio il codice *0x001* indica il protocollo ICMP. Tra i campi che popolano l'header IP, rivestono sicuramente un ruolo principale gli indirizzi IP di sorgente e destinazione. Facendo riferimento alla quarta release del protocollo, ogni indirizzo rappresenta un valore a 32 bit nel formato *dotted decimal*, quindi risulta essere costituito da quattro gruppi da 8 bit separati ciascuno da un punto. Ogni byte viene poi convertito in formato decimale per una migliore identificazione e lettura, di conseguenza ogni numero ha un range di varazione tra 0 e 255 (Figura 2.10). Un indirizzo IP agisce come un identificatore per gli end system[1] o per altri apparati contenuti all'interno della rete, così da avere device raggiungibili sia localmente all'interno della rete o da sorgenti remote oltre i confini della rete corrente.
Stante la costruzione evidenziata, le combinazioni effettivamente utilizzabili risultano pari a circa 4 miliardi, un numero di nodi di rete indirizzabili che risultava enorme agli albori di Internet, ma che oggi invece appare decisamente ristretto, perciò sono state identificate delle soluzioni:

- l'uso uso di indirizzi IP **privati**;
- In accoppiata all'uso degli indirizzi IP privati, gli host di una rete locale possono comunicare con elaboratori e server remoti attraverso la rete globale con la tecnica **Network Address Translation**, analizzata nel dettaglio nel Capitolo 4;
- tecniche per l'ottimizzazione dell'uso degli indirizzi IP, in particolare il **subnetting** e la tecnica *Variable Length Subnet Mask*;
- la transizione verso la nuova versione del protocollo, **IPv6**, che ristruttura il Layer Network ed l'indirizzamento IP sfruttando 128bit. Questa soluzione ridisegna le attuali topologie di rete dato che viene meno la necessità di adoperare la tecnica NAT ed indirizzi IP privati, ma un cambiamento così radicale necessita di una ragionevole quantità di tempo e di accorgimenti per il periodo di transizione.

Tornando all'analisi degli indirizzi IPv4, questi risultano costituiti da due campi (Figura 2.10):

- il campo **Network** indica la connotazione geografica della rete (può essere visto come l'analogo del CAP per un'indirizzo postale), rappresenta un sistema di identificazione univoco a livello globale. La dimensione delle reti, intesa come numero di nodi indirizzabili, è diversificata proprio sulla base della lunghezza in bit di tale campo;
- il campo **Host** indica il singolo nodo all'interno della rete stessa, identifica un particolare device in quella rete geografica.

[1] l'indirizzo identifica una precisa interfaccia di rete di un nodo, i router hanno naturalmente più interfacce di rete, ma anche gli host possono presentarne più di una.

Ogni rete identifica uno *spazio degli indirizzi*, di questi due risultano non essere assegnabili ai nodi che la costituiscono, in particolare sono l'indirizzo più basso e l'indirizzo più alto per il campo host:

- **l'indirizzo di rete** (Figura 3.2), anche detto *Net-Identifier*, viene ottenuto ponendo tutti i bit della sezione host dell' indirizzo IP al valore zero e rappresenta l'intera rete locale;
- **l'indirizzo di broadcast** (Figura 3.2), ottenuto ponendo tutti i bit della sezione host dell'indirizzo IP al valore uno, viene usato come indirizzo di destinazione per tutti quei dati che devono essere inoltrati a qualsiasi nodo che compone la rete locale.

Posto H il numero di bit che caratterizza il campo Host, è possibile quindi caratterizzare lo spazio degli indirizzi di una certa rete:

- Numero di nodi indirizzabili: $2^H - 2$
- Indirizzo più basso assegnabile: $NetID - 1$
- Indrizzo più alto assegnabile: $BroadcastIP - 1$

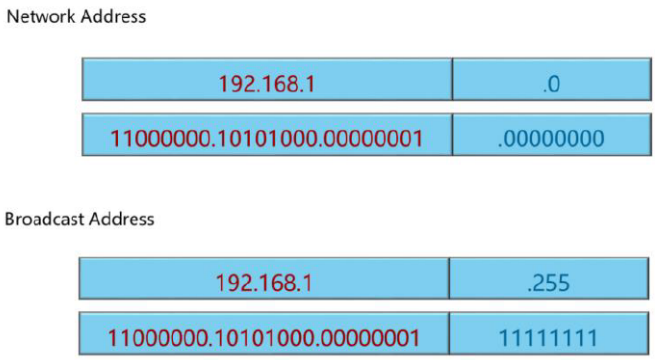

Figura 3.2: Esempio di indirizzi IP riservati per la funzione di *Network Address* e di *Broadcast Address*

Sulla base delle due parti che contraddistinguono un indirizzo IP, è possibile suddividere l'intero gruppo di indirizzi IPv4 univoci esistenti in più classi, in particolare in base al valore dei bit più significativi gli IPv4 verranno suddivise nelle 5 classi di Figura 3.3. Tanti più bit sono asseganti al NetID, in tante più reti potremo organizzare i nostri nodi, al contrario usando pochi bit per il NetID avremo poche reti molto popolose. Le classi **A**, **B**, **C** costituiscono range di indirizzi effettivamente assegnabili usate principalmente per le reti con molti host, in particolare con la classe A potremo avere sino a 126 diverse reti, ognuna che permette di assegnare 16M di indirizzi IP univoci agli host. Reti così estesi potrebbero, in contesi operativi, causare problemi di affidabilità ed elevata latenza a causa dell'elevato traffico broadcast che verrebbe generato a livello Data Link, difatti si preferisce usare le classi B e C, che costituiscono un equilibrio decisamente più adatto alle caratteristiche dei protocolli coinvolti, con la classe B si possono creare fino a 65k reti differenti ognuna che può indidrizzare 65k host, mentre con la classe C abbiamo oltre due milioni di potenziali reti, ognuna che può accogliere fino a 254 host.
Le ultime due classi costituiscono dei segmenti riservati per scopi differenti dall'assegnamento di host: La classe **D** è usata per le comunicazioni multicast, la classe **E** è coinvolta per scopi di

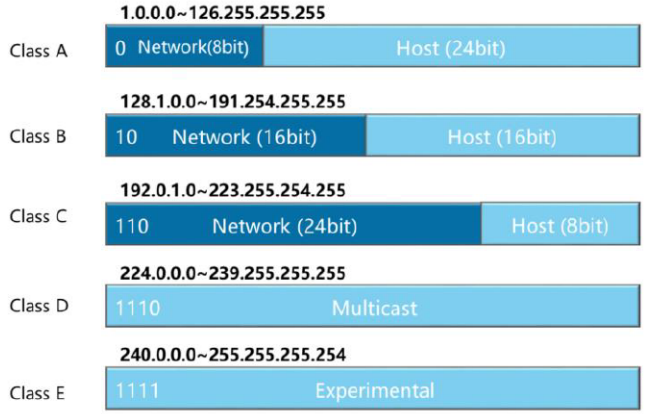

Figura 3.3: Suddivisone Classful dell'intero pool di indirizzi IPv4 univoci

sperimentazioni e testing.
Come già accennato, a causa dell'esaurimento degli indirizzi IP univoci assegnabili a singoli nodi, uno degli accorgimenti adottati ha comportato l'ulteriore suddivisione del range d'indrizzi IPv4 in due gruppi secondo la direttiva **RFC 1918** determinata dall'ente di standardizzazione **IETF**:

- Indirizzi IPv4 **pubblici**: questo gruppo di indirizzi è instradabile diretamente attraverso global Internet dato che risultano univoci a livello globali, essi sono assegnati dall'autorità mondiale *Internet Assigned Number Authority* per mezzo dei referenti regionali denominati *RIR*(per il nostro continente il referente risulta essere l'ente *RIPE NCC*). Per gli indirizzi pubblici permane la suddivisione in classi già vista, ma da ognuna delle prime tre classi sono stati riservati dei range di IP da essere considerati privati;

- Indirizzi IPv4 **privati** (Figura 3.4), non sono instradabili attraverso Global Internet dato che non hanno un vincolo di unicità globale, ma identificano univocamente i dispositivi all'interno di una stessa rete locale per permettere la comunicazione *intra network*. Sono tre range di indirizzi, uno per ogni classe individuata precedentemente.

Private Address Ranges		N° di reti
Class A	10.0.0.0~10.255.255.255	1
Class B	172.16.0.0~172.31.255.255	31
Class C	192.168.0.0~192.168.255.255	255

Figura 3.4: Range di indirizzi IPv4 privati

Ulteriori range ed indirizzi speciali sono stati definiti (Figura 3.5) per fini diagnostici, migliorare il data forwarding e per permettere di assegnare un indirizzo IP anche a calcolatori direttamente connessi con un cavo *dritto*. Nella progettazione delle reti si fanno uso di alcuni strumenti ed espressioni di indirizzi particolari che facilitano le operazioni più comuni manipolando gli indrizzi IP, è il caso della **Netmask**, o **maschera di rete**, che, dato un certo indirizzo, consente di individuare facilmente il campo Network, dando conto del numero di bit che lo compone, ed il campo Host. La netmask può essere espressa secondo due diverse notazioni:

Special Addresses	
Diagnostic	127.0.0.0 ~ 127.255.255.255
Any Network	0.0.0.0
Network Broadcast	255.255.255.255

Special Addresses	
Link Local	169.254.1.0 – 169.254.255.255

Figura 3.5: IPv4 riservati per particolari fini diagnostici, di routing e per rappresentare un link locale

- mediante la *dotted decimal notation*, come in Figura 3.6;
- mediante la *slash notation* in cui all'indirizzo IP viene appunto fatto seguire uno *slash* ed il *prefix length*, un numero decimale che indica il numero di bit ad uno della maschera.

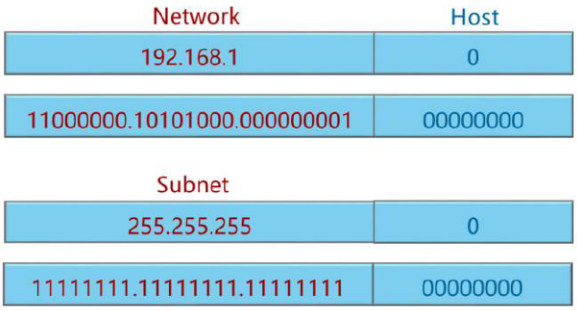

Figura 3.6: Maschera di rete espressa in *Dotted Decimal Notation* ed in *Dotted Binary Notation*

Per ognuna delle classi di IP definite in precedenza viene perciò definita una *maschera di rete di default* 3.7. Dal concetto di rete già definito, risulta parte integrante della progettazione la

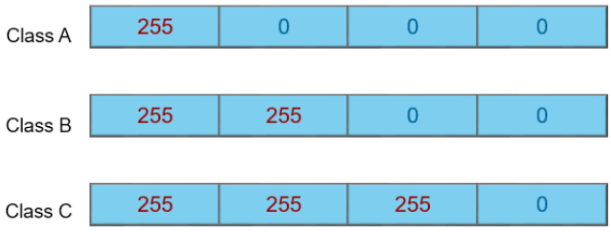

Figura 3.7: Maschere di rete di default per le classi A, B e C

suddivisione dei nodi in più reti, questi gruppi individuati vengono ulteriormente organizzati in sottosegmenti differenti in base a:

- funzionalità dei nodi che compongono una sottorete;
- distribuzione geografica dei nodi;

- ottimizzazione del traffico e riduzione dei pacchetti di broadcast;

L'adozioni di reti classful in contesti operativi presenta un grande difetto, ossia quello di essere costretti ad usare per piccole sottoreti delle classi che, di norma, permettono di indirizzare un numero di nodi decisamente superiore rispetto alle reali necessità, ciò comporta un forte spreco di indirizzi univoci ed anche un alto costo essendo quest'ultimi formalmente esauriti da diversi anni. Lo schema classful è quindi passato in secondo piano dato che, presa una singola rete di una certa classe, questa viene stato quindi ulteriormente segmentata mediante la tecnica di **subnetting**, definita per la prima volta nella direttiva **RFC 950**.
Il subnetting permette di suddividere ulteriormente lo spazio degli indirizzi IPV4 in modo da ottenere dei gruppi più piccoli. Possiamo quindi prendere una rete che appartinere ad una classe (A,B o C) e dividerla in gruppi più piccoli contigui tra loro che aderiscono meglio ai nostri scopi e possono anche essere venduti a soggetti terzi. Questo metodo garantisce perciò una migliore ottimizzazione della risorsa limitata degli IP pubblici al costo di usare alcuni bit che le classi già menzionate dedicano agli host per differenziare ed indirizzare univocamente le sottoreti che si intendono realizzare. Una volta che una classe di indirizzi viene ulteriormente suddivisa in più reti, i device di *subnet* diverse risultano appartenere a delle reti indipendenti a tutti gli effetti, nonostante abbiano una classe di partenza comune. Il calcolatore può effettuare delle operazioni binarie sulla maschera che caratterizza ogni singola sottorete, così da capire se un pacchetto da inviare ad un certo host debba essere inoltrato sulla stessa rete del mittente oppure se debba coinvolgere il router che, come già menzionato, permette la comunicazione tra reti diverse.
All'interno delle singole subnet, ogni host si vedrà assegnato un certo indirizzo di livello Network, in particolare possiamo avere:

- asseganzione **statica** da parte dell'amministratore di rete, che mantiene un controllo completo del piano degli indirizzi, ma può commettere errori di duplicazione;
- assegnazione **dinamica** degli indirizzi previa la configurazione di un **DHCP server**, così non viene richiesto un continuo intervento dell'amministratore di rete.

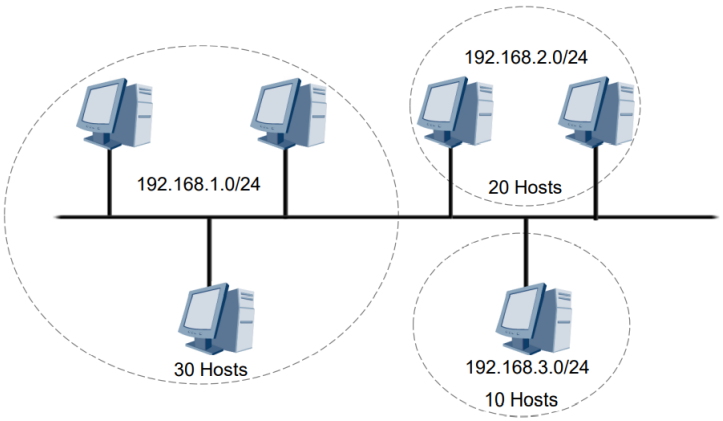

Figura 3.8: Design che include tre sottoreti diverse, ognuna adotta la maschera di rete di default, quindi viene assegnata una rete di classe C ad ogni raggruppamento

In Figura 3.8 è mostrato una necessità di indirizzamento tipica in ambiti enterprise con molte sottoreti composte da pochi nodi. Adottando un approccio classful come in figura, si evidenzia come, nonostante il ridotto numero di host presenti in ogni sottorete, gli indirizzi IP riservati per ogni subnet risultano essere decisamente sovrabbondanti. Questo succede perchè si è adottata

una tecnica di subnetting statico, anche detta a *single mask size*,in cui ogni sottorete è caratterizzata dalla stessa maschera, quindi le subnet risultano tutte capaci di indirizzare potenzialmente fino a 254 host diversi, oltretutto il piano d'indirizzamento così concepito è caratterizzato da una scarsa flessibilità. Adottando la tecnica di subnetting **Variable Length Subnet Mask** è invece possibile superare questa problematica, essa permette di usare una prefix length diversa per ogni subnet, così da segmentare lo spazio degli indirizzi della classe iniziale in maniera disomogenea. la tecnica VLSM permette quindi un approccio più flessibile, **classless**. Con VLSM risulta fondamentale la scelta della maschera di sottorete, in particolare scelta una rete di una certa classe, dovrò considerare alcuni bit che risultano, secondo l'approccio classful, dedicati all'indirizzamento degli host, per identificare la subnet, vado così a definire il *subnet ID* (Figura 3.9). Si noti che dividere in due una rete significa a tutti gli effetti fare uno *shift binario*, si

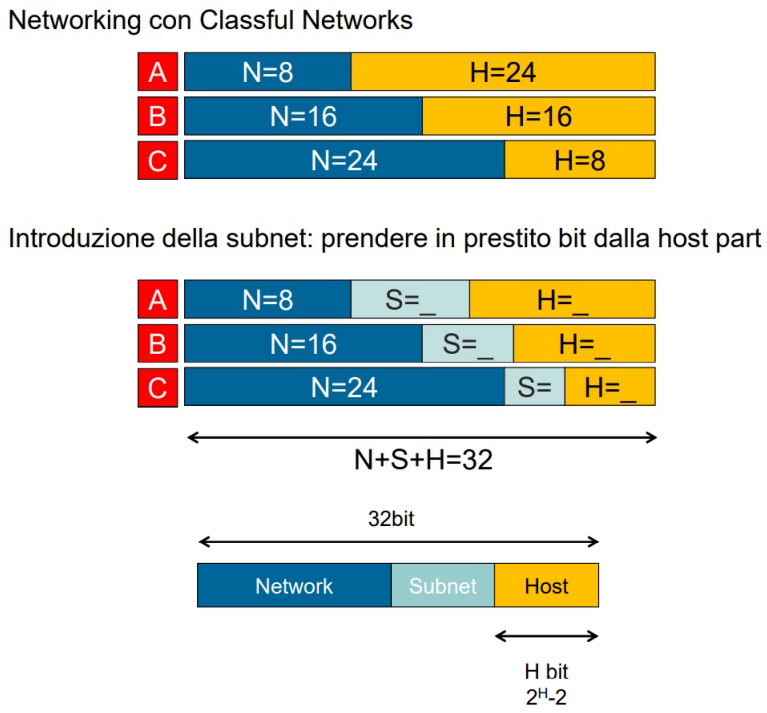

Figura 3.9: Subnetting con subnet ID

transla un bit del campo host al campo network e ciò comporta una diminuzione del numero di host indirizzabili, inoltre le operazioni di subnetting vanno sempre valutate, perchè ogni ulteriore suddivisione della rete più grande in reti più piccola comporta lo spreco di 2 indirizzi IP per ogni subnet, che rappresenteranno NetID e indirizzo di broadcast della sottorete. Per questi motivi, è buona norma predisporre in anticipo un adeguato piano di indirizzamento.

3.2 Comunicazione tra host remoti

Per far sì che un host sia in grado di inoltrare delle informazioni ad una generica destinazione, il mittente deve avere conoscenza della rete in cui risiede il nodo cui il messaggio è rivolto. Due o più dispositivi possono comunicare fra di loro direttamente solo se appartenenti alla stessa rete locale e in questo caso utilizzano dei frame, al contrario se non si ha conoscenza della rete di

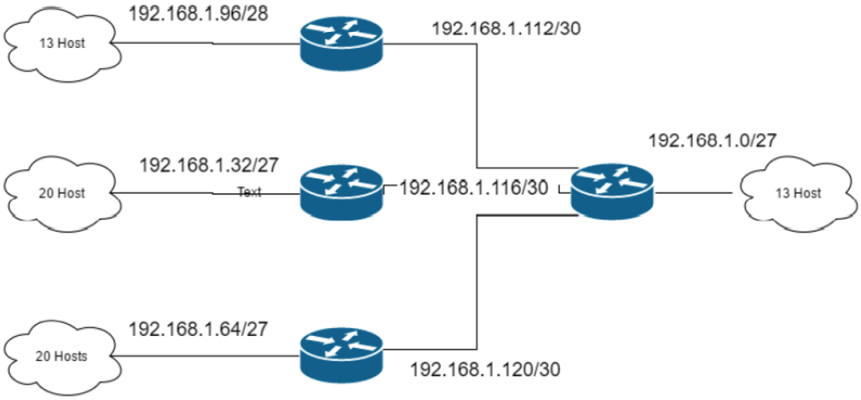

Figura 3.10: Design di rete che adotta la tecnica di subnetting VLSM

destinazione è necessario coinvolgere un router a cui inoltrare i pacchetti IPv4 che costituiscono la richiesta all'host remoto. Ogni nodo ha quindi almeno due indirizzi:

- un indirizzo di livello 2;
- uno (o più, nel caso di router) indirizzi di livello 3.

Perché la comunicazione tra due host sia verificata, bisogna accertarsi innanzitutto se la destinazione è parte della stessa rete, questo avviene comparando gli indirizzo IP di sorgente e destinazione indicati nell'header IP: Il calcolatore può effettuare delle operazioni binarie sulla maschera di ogni singola sottorete, in particolare effettua un processo di *AND bit a bit* per capire se un pacchetto da inviare sia verso un host della stessa rete del mittente.
In caso l'operazione binaria risulti in una corrispondenza, il pacchetto viene inviato ai layer più bassi per il processamento e la comunicazione avviene tramite frame direttamentre tra i due host, altrimenti se il nodo di sorgente non ha conoscenza della rete di destinazione essendo differente da quella d'origine, il pacchetto viene scartato prima che raggiunga il layer DataLink dello stesso mittente.
Per permettere a due host di network differenti, di comunicare tra loro è necessario aggiungere un altro componente, il **Gateway** (anche detto **router**). Questo permette l'instradamento dei pacchetti provenienti e diretti verso reti diverse fra di loro (Figura 3.11). L'instradamento dei pacchetti verso una certa rete di destinazione richiede la conoscenza del percorso che l'informazione deve seguire, in particolare deve risultare nota al mittente l'interfaccia dove inoltrare i pacchetti prima di procedere all'incapsulamento in un frame cosi che il nodo non scarti il pacchetto, bensì proceda l'elaborazione nei layer più bassi.
Il gateway è quindi un apparato di rete in grado di processare pacchetti IP per coprire la necessità di instradare messaggi tra reti differenti. Questo dispositivo deve essere a conoscenza del percorso necessario per raggiungere la rete di destinazione per una certa richiesta, ancora prima che l'inoltro dei pacchetti abbia inizio. Il router è connesso a tutte le reti che deve mettere in comunicazione mediante le sue interfacce, ognuna di esse risulta essere a tutti gli effetti un nodo di una certe rete locale, ed in quanto tale è individuata con un certo indirizzo IP in coerenza alla subnet adottata per la specifica rete di cui sopra. A questo nodo è associato un **indirizzo di gateway** (anche chiamato **default gateway**) che rappresenta la destinazione predefinita della rete corrispondente, difatti tutte le reti non note ad un host potrebbero essere raggiungibili mediante il gateway.
In figura 3.11 l'host A, non avendo conoscenza dell'interfaccia cui inoltrare un pacchetto per

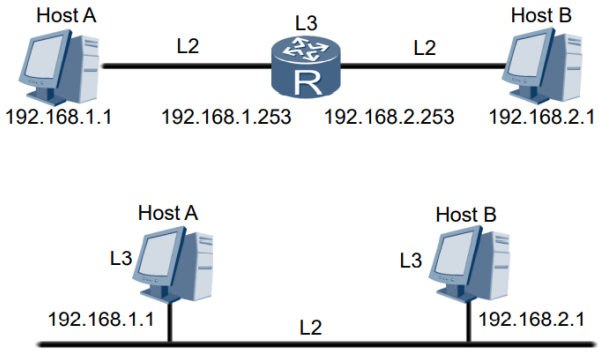

Figura 3.11: Comunicazione tra host appartenenti a reti differenti tramite un nodi che processa pacchetti IP

comunicare con l'host remoto B, si rivolge al gateway per evitare di scartare il pacchetto, che quindi viene incapsulato in un frame inviato all'indirizzo di gateway. Il gateway riceve un frame, lo analizza e ,verificando il campo Type, estrae il pacchetto incapsulato ed analizza i campi dell'header IP. Sulla base del destination address indicato nell'header IP, il gateway, sfruttando una tabella a lui nota che contiene gli indirizzi di rete che fanno capo al router, incapsula il pacchetto in un adeguato frame in cui viene indicato l'indirizzo fisico del destinatario e lo instrada sulla sua interfaccia connessa direttamente nella rete di destinazione. Nel secondo caso in figura, in cui sono presenti due host attestati su reti diverse senza un gateway fisico a dividerle, è responsabilità dell'host comportarsi come gateway, ossia deve essere a conoscenza della rete alla quale i pacchetti devono essere trasmessi e deve specificare il proprio indirizzo IP di interfaccia come indirizzo di gateway, tramite il quale è possibile raggiungere la rete di destinazione.
Il gateway risulta quindi essere un nodo intermediario che conosce e comunica con tutti i nodi della rete locale di partenza e di destinazione, esso è in grado di prendere decisioni riguardo all'instradamento dei pacchetti perchè questi raggiungano i nodi desiderati. La conoscenza del router riguardo alle rotte verso un possibile IP destinazione è rappresentata dalla **tabella di routing** che ha in memoria, essa fornisce una conoscenza a priori al gateway di tutte le possibili reti di destinazione e l'interfaccia cui sono attestate.
La tabella di routing è quindi un database, memorizzato in un router o in un host, che elenca le rotte di destinazione di una data rete ed una **metrica di tale rotta** (in genere legata alla distanza). La tabella può essere popolata in due modi differenti:

- **manualmente**. L'amministratore di rete definisce delle rotte statiche secondo quello che ritiene il miglior percorso per raggiungere ogni destinazione, tali rotte non reagiscono ad eventuali cambiamenti topologici;
- **automaticamente**. la costruzione della tabella è obiettivo dei *protocolli di routing*. Essi provvedono ad un aggiornamento automatico della tabella, ma richiede uno scambio di informazioni tra i router con un apposito protocollo ed un metodo (algoritmo) per determinare il percorso migliore il base al concetto di metrica definito dal protocollo adottato.

Tra i protocolli adottati per popolare una tabella di routing possiamo citare **OSPF**: è un protocollo *link state* che si basa sullo scambio di informazioni sullo stato dei link tra router (tiene

quindi conto delle caratteristiche, in particolare la velocità, dei collegamenti fisici tra i nodi), tali informazioni compongono un database i cui dati vengono elaborati dall'algoritmo *Shortest Path First* che calcola la miglior rotta verso ogni destiazione secondo la specifica metrica definita. Questo protocollo è stato inizialmente pensato per reti con un certo numero di nodi e può risultare anche complesso da gestire, ma risulta molto veloce. In base alle nozioni riportate, è ora

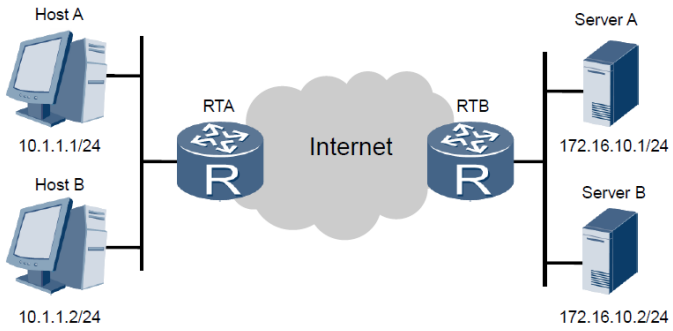

Figura 3.12: Esempio di semplice topologia di rete in cui si analizza il forwarding dei dati

possibile visualizzare nel dettaglio il flusso dei dati nell'instradamento da una generica sorgente ad un generico destinatario, che sia locale o remoto, basandosi sull'esempio di Figura 3.12. Il processo di data forwarding poggia le sue fondamenta nell'incapsulamento garantito dallo stack TCP/IP per realizzare una comunicazione logica end-to-end, la trasmissione effettiva si basa sui protocolli che caratterizzano ogni strato sia del mittente, sia del destinatario. Supponendo che l'host A voglia realizzare una connessione HTTP (che corrisponde ad una certa *porta* di destinazione per il layer Applicativo del destinatario, ossia la porta *80*) con il server A che risiede in una rete locale remota, il nodo deve innanzitutto determinare se è possibile raggiungere tale destinazione desiderata con un processo di *path discovery*.

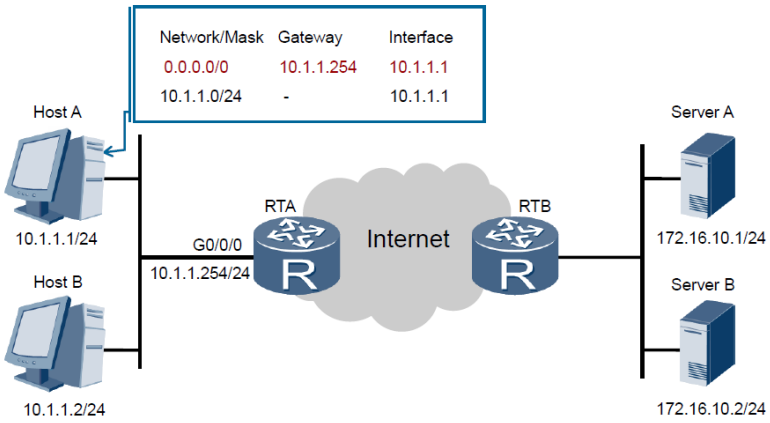

Figura 3.13: Processo di *path discovery*

Tale procedimento coinvolge l'indirizzo IP del destinatario remoto: esso viene confrontato con una tabella di raggiungibilità (Figura 3.13) memorizzata dall'host, in particolare nell'esempio essendo la rete di destinazione non nota, l'unica corrispondenza con la tabella è con l'indirizzo *any-network*. Quest'ultimo ha la particolarità di inoltrare il pacchetto al *default gateway*, quindi la resposabilità di inoltrare il datagramma verso la sua destinazione in una rete remota è lasciata

al router. Il nodo si preoccuperà solamente di inviare il pacchetto al gateway tramite la sua interfaccia con IP *10.1.1.1*.
Una volta determinato il *path logico* per il pacchetto, l'host deve determinare il *path fisico*, ossia l'interfaccia DataLink cui inoltrare il frame in cui andrà poi ad incapsulare il pacchetto originale. Per completare il percorso fisico, viene consultata la tabella ARP memorizzata dall'host, detta *ARP cache table*. Si verifica quindi che esista un'associazione tra l'indirizzo IP di destinazione desiderato, quello del gateway, ed un MAC address che indentifica l'indirizzo fisico dell'interfaccia del router. Nell'esempio, come evidente dalla Figura 3.14, è presente il record cercato, quindi

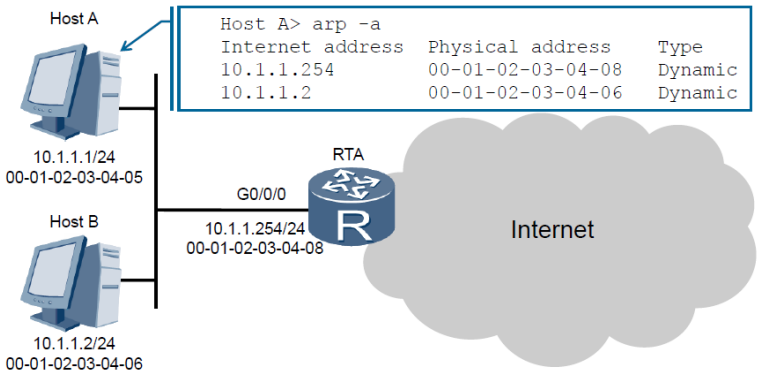

Figura 3.14: Tabella ARP

anche il *path fisico* è stato risolto. Nel caso in cui invece non vi sia alcuna corrispondenza, è necessario eseguire la procedura di *ARP request*, essa sfrutta un apposito frame inviato in broadcast per venire a conoscenza della destinazione, che si presenta al mittente con un frame inviato a lui in unicast che costituisce l'*ARP reply*. Una volta che sono stati risolti sia il path logico che il path fisico, avviene l'incapsulamento effettivo del messaggio da inviare al destinatario. Il layer applicativo, secondo il protocollo TCP, si occuperà di popolare i campi del segmento che realizzeranno l'instaurazione della connessione, inoltre verrà calcolato un *checksum* di integrità per assicurare l'integrità dei dati una volta che il segmento TCP sarà elaborato dal layer applicativo del destinatario (Figura 3.15).

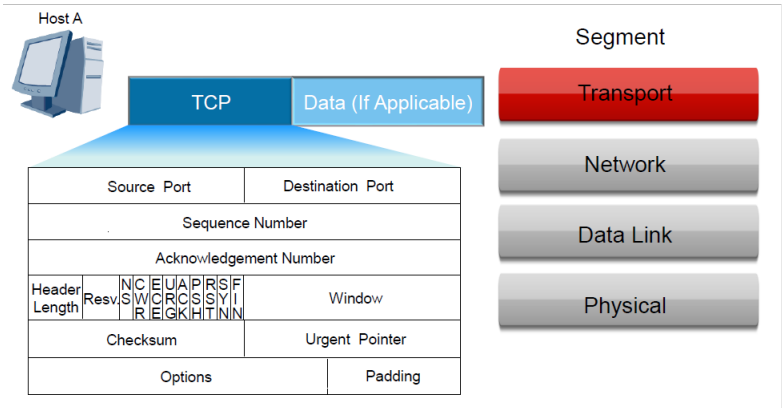

Figura 3.15: Incapsulamento del messaggio HTTP in un segmento TCP

A seguito dell'incapsulamento in un segmento TCP, è necessario passare al livello inferiore dell'architettura per specificare i dettagli che permettono la tramissione dell'informazione tra reti diverse, in particolare evidenziando gli indirizzi IP di sorgente e destinazione (Figura 3.16).
Nella procedura di incapsulamento in un pacchetto IP, nel relativo header vengono inoltre specificiati altri campi, come la natura del payload del pacchetto attraverso il campo protocol (nel caso dell'esempio, avendo incapsulato un segmento TCP, in tale campo sarà indicato il numero esadecimale *0x06*), oppure il campo TTL che determina il tempo il numero di volte cui il pacchetto può transitare in un nodo prima di essere ritenuto non valido.

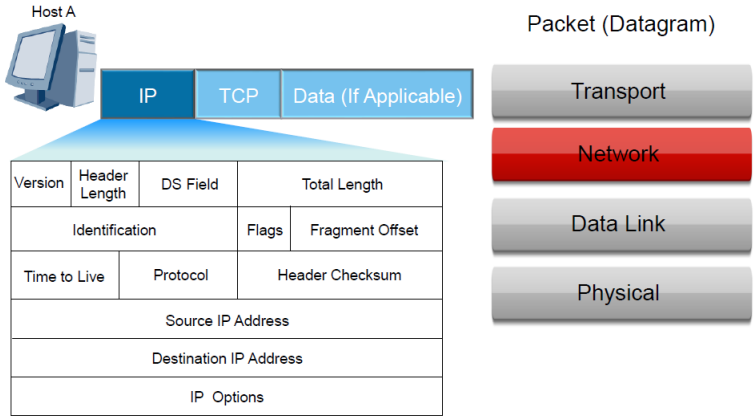

Figura 3.16: Incapsulamento del segmento TCP in un pacchetto IP evidenziando i campi dell'header IP

Arrivando al layer Data Link del mittente, l'incapsulamento segue lo standard Ethernet II dato che viene incapsulato un payload di livello Network che contiene un pacchetto IP, difatti nel campo Type del frame viene specificato il numero esadecimale *0x0800*, inoltre vengono evidenziati gli indirizzi MAC delle interfaccie di invio e destinazione (Figura 3.17).
Il processo di incapsulamento del mittente risulta ora terminato, si coinvolge così il layer fisico

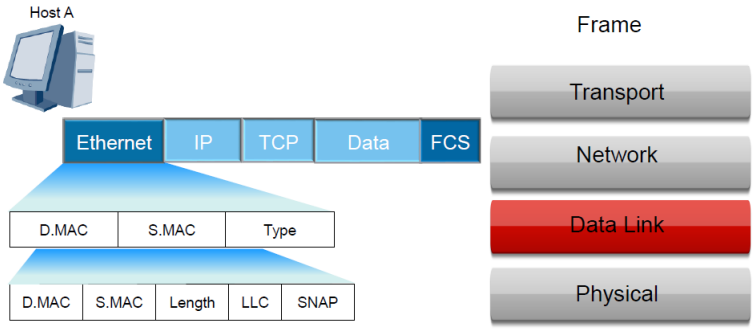

Figura 3.17: Incapsulamento del pacchetto IP in in frame Ethernet II

attraverso il quale avverà la trasmissione verso l'interfaccia fisica del gateway (Figura 3.18). In questa fase è importante assicurarsi che non coesistano altri segnali elettrici nel mezzo di trasmissione che risulta essere un dominio di collisione, si adotterà quindi il protocollo *CSMA/CD*. I 64 bit di preambolo del frame permettono la sincronizzazione della comunicazione, in particolare l'interfaccia di destinazione sarà avvertita di una imminente comunicazione in arrivo tramite un

frame attraverso i primi 56 bit, con l'ultimo ottetto, denominato *Start of Frame* si avvisa l'host dell'imminente arrivo dei bit che rappresentano effettivamente il frame, in particolare il MAC di destinazione.

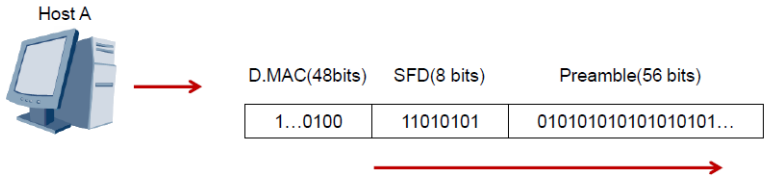

Figura 3.18: Trasimissione fisica dei bit del frame sul mezzo trasmissivo

Tutte le interfacce attinenti allo stesso dominio di collisione elettrico riceveranno il frame, ma solo il destinatario il cui indirizzo fisico corrisponde effettivamente a quello indicato nel campo *Destination MAC* del frame, procederà con l'elaborazione dello stesso, gli altri provvederanno a scartarlo (Figura 3.19). Oltre il controllo di questa corrispondenza, vengono effettuate ulteriori operazioni di *matching* per determinare l'integrità del frame ricevuto sulla base del campo *Frame Check Sequence*. In caso di validità e integrità del frame, il gateway avvierà la procedura di decapsulamento scartando header e trailer del frame. Il funzionamento del router viene

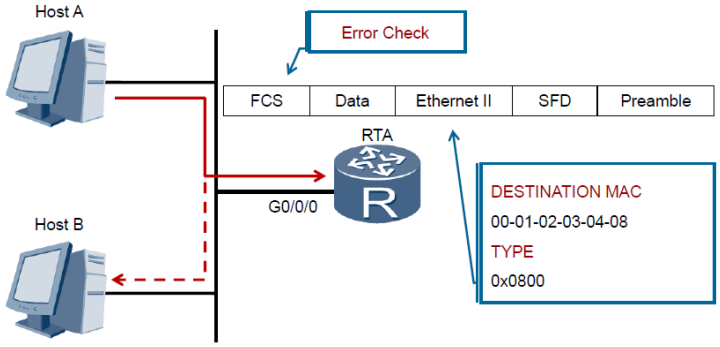

Figura 3.19: Elaborazione del frame da parte del gateway

principalmente espletato a livello Network, difatti una volta che il router ha estratto l'header IP, procede nella sua elaborazione controllandone innanzitutto l'integrità, controllando il valore del TTL (se fosse nullo procederebbe a scartarlo) quindi verifica l'IP di destinazione per accertarsi se il pacchetto abbia raggiunto o meno la sua destinazione. Nell'esempio di Figura 3.20 è evidente come il router dovrà provvedere ad inoltrare il pacchetto su una delle sue interfaccie che permette di raggiungere la rete dove risiede il nodo destinatario, in particolare in questa fase entrerà in gioco la **tabella di routing** come già evidenziato.
A questo punto il pacchetto verrà incapsulato in un nuovo frame con nuovi indirizzi fisici di sorgente e destinazione, quest'ultimo MAC viene ricavato dalla tabella *ARP cache* memorizzata nel router.

Quando il frame verrà inoltrato dal gateway nella rete locale di destinazione, si andrà incontro allo stesso processo già visto per cui solo il reale destinatario elaborarerà il frame (Figura 3.21),

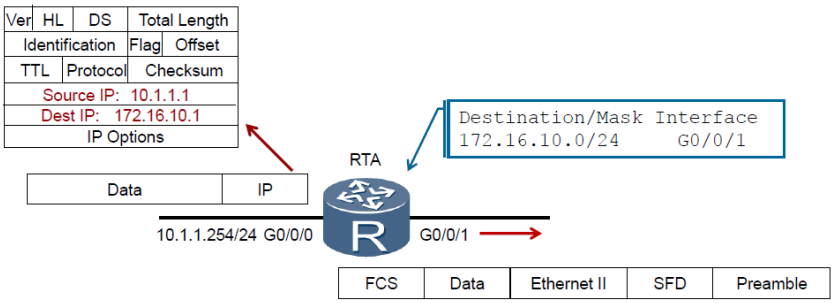

Figura 3.20: Elaborazione del pacchetto IP da parte del router

quindi effettuerà i controlli per verificare l'integrità dello stesso e procederà ricavare il pacchetto IP dal payload. Il nodo destinatario andrà ad elaborare l'header IP in maniera del tutto simile a

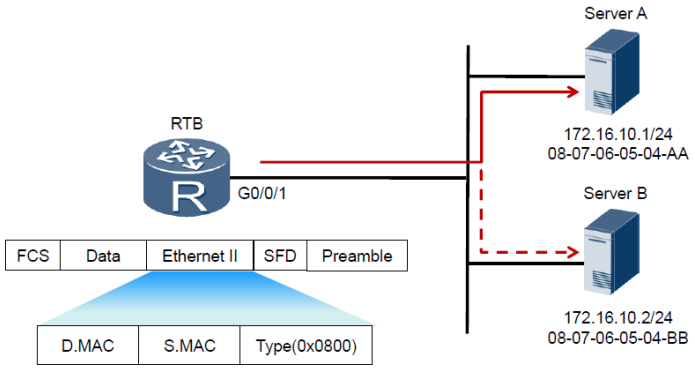

Figura 3.21: Inoltro del frame nella rete locale del destinatario

quanto già visto per il router, ma in questo caso, dato che la destinazione finale è stata raggiunta, l'indirizzo IP di destinazione corrisponde all'indirizzo IP della stazione corrente. A questo punto viene valutato il campo *Protocol* per determinare il protocollo applicativo che dovrà elaborare il payload del pacchetto ed individuare così l'header di livello 4 (Figura 3.22). In questo caso il protocollo sarà TCP per l'esempio visto, quindi scartando l'header IP si ottiene un segmento. Il segmento risultante sarà elaborato dal protocollo TCP nel layer Transport, in figura 3.23 si può notare come la connessione TCP sia già stata stabilita in precedenza e l'header TCP indica l'instaurazione di una connessione con il servizio applicativo *HTTP* sulla porta *80*, con eventuali paramentri contenuti nell'header che regoleranno i paramentri della connessione.

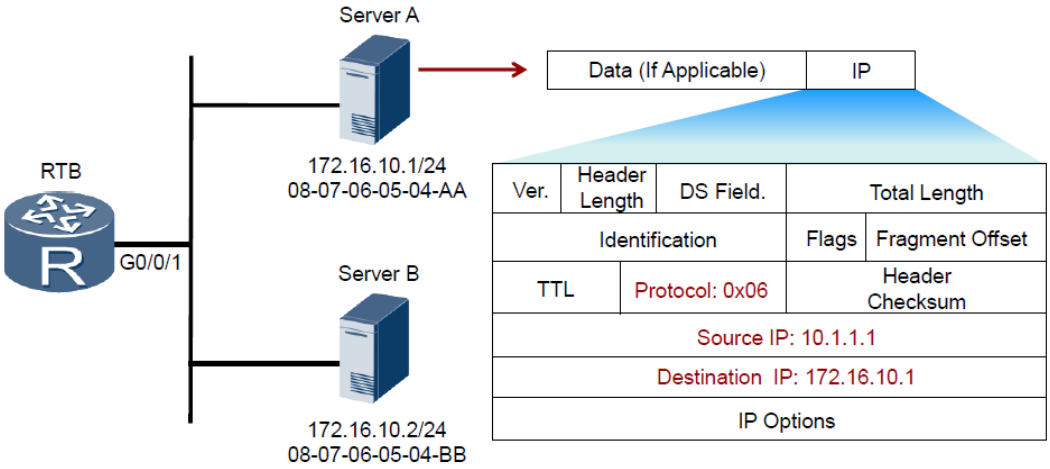

Figura 3.22: Procedura di decapsulamento del frame e analisi dell'header IP

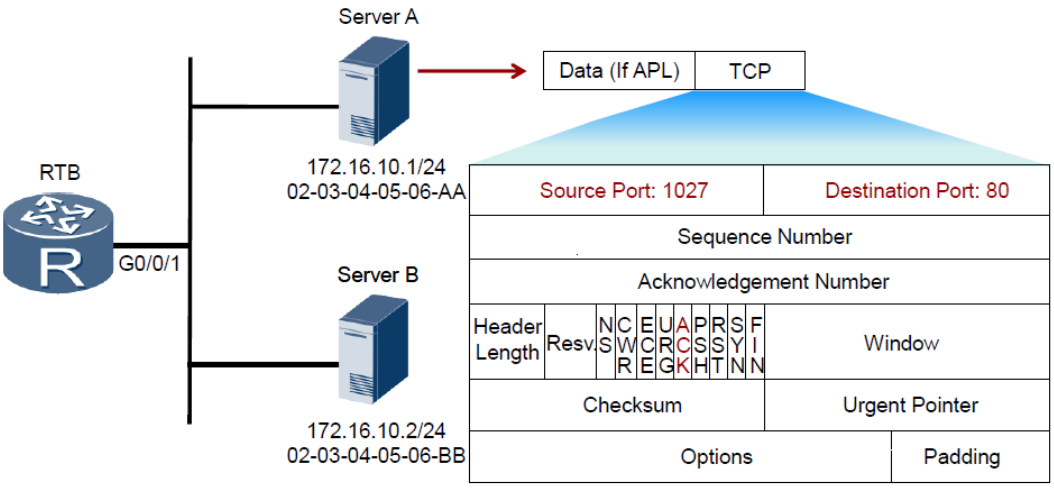

Figura 3.23: Procedura di decapsulamento del pacchetto e analisi dell'header TCP

Capitolo 4

Network Address Translation

4.1 Generalità e principio di funzionamento

Nell'ambito del progetto realizzato l'uso di una connessione radiomobile fornita da un ISP commerciale ha fornito un accesso ad internet al nostro gateway attraverso una rete che sfrutta la tecnica **Network Address Translation** per fornire connettività ai propri utenti. Proprio l'adozione del NAT costituisce l'ostacolo principale alla desiderata comunicazione tra i due host remoti che ha comportato la necessità di dotarsi di un servizio di server remoto VPS.
Questa tecnica di design per le reti ha avuto una diffusione sempre più pervasiva negli ultimi anni perchè, come citato nel capitolo 3 mitiga l'esaurimento degli IPv4 permettendo agli host indirizzati con IP privati nelle reti locali di accedere a global Internet. La politica di assegnazione degli indirizzi IPv4 è stata basata inizialmente sul principio del *first came first served*, dal 2012 la scarsità di indirizzi univoci globalmente ha costretto lo *IANA*, tramite le sue agenzie regionali, a limitare l'assegnazione di nuovi blocchi finchè nel 2019 si è arrivati al completo esaurimento degli spazi (Figura 4.1). L'adozione di IPv6, unica soluzione definitiva alla penuria di indirizzi pubblici, benchè avviata non risulta così rapida, data la mancata definizione di scadenze o vincoli perentori.

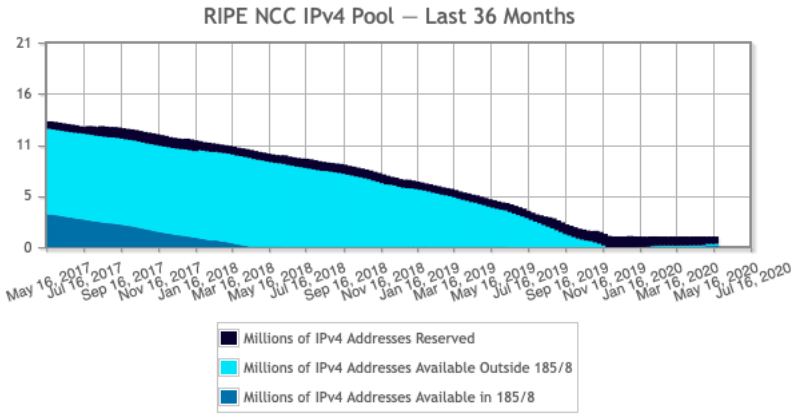

Figura 4.1: Andamento della disponibilità di blocchi IPv4 non assegnati (link)

Definito nella direttiva **RFC 3022**, NAT consente ad uno o più host che non hanno un indirizzo IP valido, registrato e globalmente unico, di comunicare con altri host attraverso la rete internet. Per ottenere questo risultato, NAT utilizza uno o più indirizzi IP validi e pubblici che fanno le veci di una moltitudine di indirizzi privati sulla rete pubblica. Gli indirizzi privati di sorgente risultano effettivamente mascherati da un pool minore di indirizzi pubblici, ciò può essere visto come una buona garanzia di sicurezza, seppur non sia il suo scopo principale.
Il dispositivo che effettua il NAT solitamente è un nodo di transito su un sistema in cui agisce da gateway all'interno di un ambito di comunicazione tra due o più host. Nella maggior parte delle configurazioni perciò se ne occuperà il router, esso agisce da *confine* tra un ambiente interno privato, con indirizzi non instradabili su global Internet, e la rete pubblica, si occupa della translazione degli indirizzi per la comunicazione tra i due amibiti (Figura 4.2).

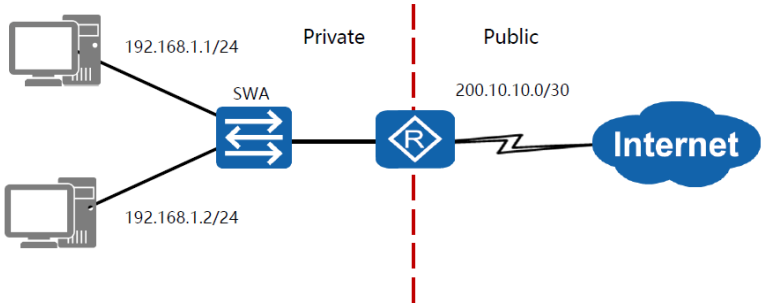

Figura 4.2: Esempio di configurazione domestica con un router ad una cui interfaccia è assegnato un IP pubblico, mentre una rete locale è attestata ad un'altra interfaccia del router

La tecnica si basa essenzialmente su due operazioni di **traduzione degli indirizzi**:

- operazione di *Source NAT*: modifica il contenuto dell'header IP del pacchetto che transita nel nodo per uscire dalla rete locale, in particolare riguardo al campo dell'indirizzo IP di sorgente;
- operazione di *Destination NAT*: modifica il contenuto dell'header IP del pacchetto che transita nel nodo per entrare dalla rete locale, in particolare riguardo al campo dell'indirizzo IP di destinazione.

In dettaglio, quando un host inoltra un pacchetto al default gateway per comunicare con un nodo in una rete esterna, il router lo riceve e lo confronta con la sua tabella di routing, decidendo innanzitutto l'interfaccia d'uscita per il pacchetto, con una procedura del tutto simile a quella vista nel capitolo 3. In genere, essendo il pacchetto rivolto ad una destinazione remota, esso dovrà transitare attraverso global Internet, quindi sicuramente l'interfaccia individuata sarà caratterizzata da un IP pubblico essendo affacciata sulla rete globale. Prima di effetttuare l'icapsulamento per l'inoltro, il gateway si comporta da *Source NAT*, sostituendo l'IP privato presente nell'header IP come sorgente con l'IP pubblico che caratterizza l'interfaccia d'uscita. Tale traduzione viene registrata in un'apposita tabella di *mapping*, dato che quando la destinazione remota invierà una risposta all'host originale, essa sarà rivolta all'IP pubblico del router che effettua una nuova operazione di traduzione (*Destination NAT*) sostituendo però l'indirizzo pubblico con l'indirizzo privato precedentemente memorizzato (Figura 4.3). Sono state sviluppate diverse implementazioni della tecnica NAT per adattarsi ad una varietà di situazioni differenti.

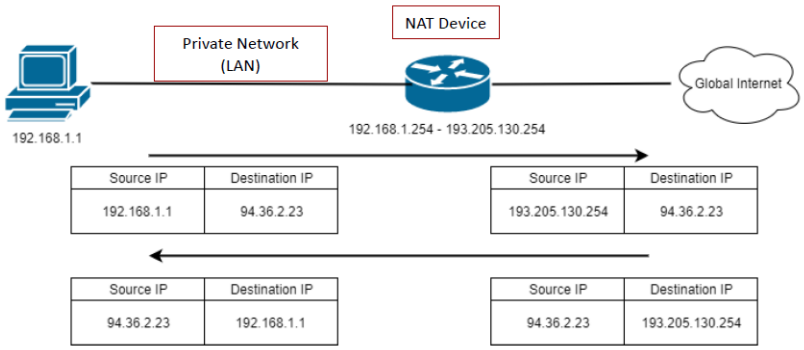

Figura 4.3: Esempio di funzionamento della tecnica NAT con evidenza della tabella di *mapping*

4.2 NAT statico

Con il NAT statico si rappresenta una corrispondenza *1 ad 1* tra IP pubblico e IP privato, permettendo al traffico di un preciso host nella rete locale di essere tradotto ad uno specifico IP pubblico (Figura 4.4). È una modalità che, per come è stata concepita, non permette di

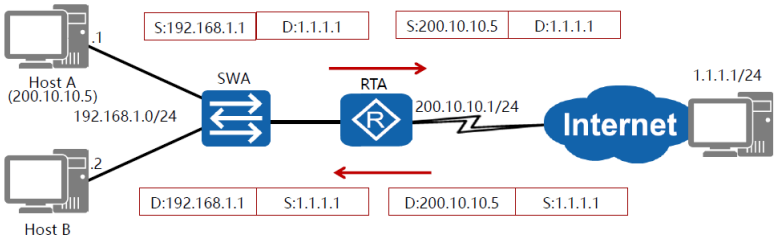

Figura 4.4: Esempio di funzionamento della tecnica NAT statico evidenziando gli indirizzi IP coinvolti

risparmiare indirizzi pubblici, ma può risultare comoda in caso si voglia rendere accessibile un server, residente nella rete locale, a richieste esterne provenienti da global Internet mantenendo un certo *information hiding* riguardo i dettagli implementativi della rete locale aziendale, si usa quindi per motivi di sicurezza. L'operazione di mapping statico effettuata dal router non richiede una vera e propria gestione degli indirizzi assegnati agli utenti, dato che l'assegnazione è fissata dall'amministratore di rete solo al primo *setup*.

4.3 NAT dinamico

In questa tecnica si dispone di un certo *pool* di IP pubblici disponibili. Nel caso in cui un host interno alla rete desideri instradare del traffico attraverso la rete pubblica, ad esso, tramite il suo indirizzo privato, viene mappato un certo IP pubblico tra quelli disponibili nel pool, il mapping avviene in base alle necessità dei singoli host e alla disponibilità di risorse assegnabili (Figura 4.5). Il particolare IP pubblico assegnato ad un certo host che desidera comunicare non è importante, ma è fondamentale che questo riesca a cominicare con global Internet. Una

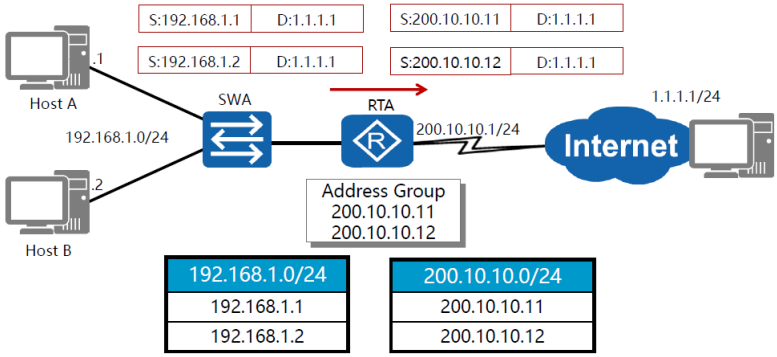

Figura 4.5: Esempio di funzionamento della tecnica NAT dinamico evidenziando gli indirizzi IP coinvolti

volta terminanata la comunicazione, l'assegnazione viene rimossa dalla tabella di mapping per poter riutilizzare l'indirizzo pubblico con altri host. In genere, il pool di risorse diponibili è decisamente minore rispetto al numero di host nella rete locale per risparmiare IP pubblici, quindi c'è la possibilità che a qualche calcolatore venga negata la comunicazione verso l'esterno.

4.4 NAT con Port Address Translation

Anche detto *IP masquerading*, a differenza del NAT dinamico esso permette un mapping multiplo di più indirizzi privati in un'unico IP pubblico, non ci saranno così richieste non evase. Il suo funzionamento si basa sul tracciamento delle connessioni, intese come flusso bidirezionale di pacchetti tra due host identificati da caratteristiche a livelli superiori a quello Network, le connessioni quindi verranno differenziate sulla base delle porta che identifica il servizio richiesto, definita nell'header di Livello Transport. Anche in questa tecnica abbiamo una moltitudine di host cui sono assegnati degli IP privati ed un pool, molto minore, di indirizzi instradabili globalmente. Come evidente in Figura 4.6, nella tabella di mapping vengono gestite indivi-

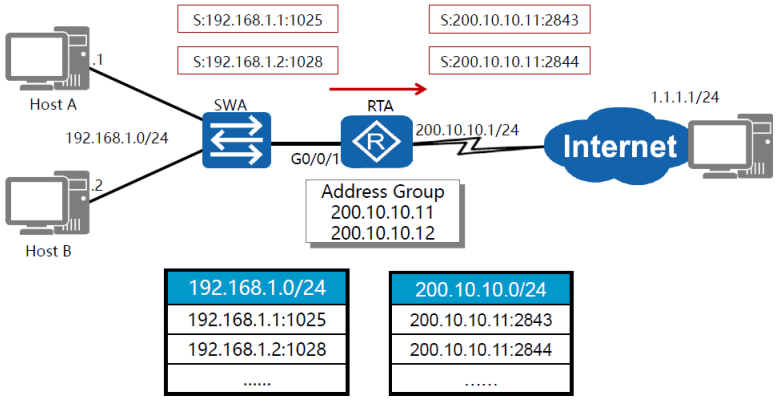

Figura 4.6: Esempio di funzionamento della tecnica NAPT evidenziando gli accoppiamenti *indirizzi IP: porta* coinvolti

dualmente le connessione di livello Transport: la coppia *indirizzo IP : porta* che identifica un

singolo pacchetto inviato da un host della rete locale, anche detta **socket** viene riportata nella tabella di mapping e viene associata ad una socket caratterizzata dall'indirizzo IP pubblico e da una porta scelta spesso in maniera casuale. Ciascuna corrispondenza tra porta dell'indirizzo esterno e socket interna individua una connessione. quindi un altro elaboratore caratterizzato da una diversa socket, sarà mappato sullo stesso IP pubblico ma con una diversa porta. Alla ricezione della risposta rispetto richiesta inviata, il router consulta la tabella per sapere a quale host interno e su quale porta inviarlo. La flessibilità garantita da questa tecnica è notevole, è possibile gestire senza problemi socket identiche originate da macchine differenti, così come sequenze di socket richieste dallo stesso elaboratore, il mapping delle singole richieste su porte diverse effettuato dal NAPT permetterà di distinguerle. Con questo meccanismo l'elaborazione della tabella risulterà più lenta e la stessa dovrà essere periodicamente ripulita dalle connessioni non più attive, ma sarà possibile distinguere potenzialmente sino a 65k utenti privati per ogni risorsa pubblica disponibile.
L'implementazione tipica della tecnica NAPT in ambienti privati (o comunque di piccole aziende) prevede un pool di risorse disponibili ridotto ad un'**unico IP pubblico** su cui vengono mappate tramite diverse porte tutte le socket interne (Figura 4.7).

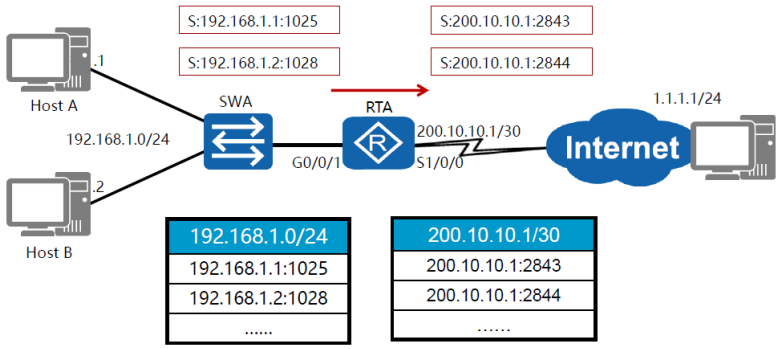

Figura 4.7: Tipica implementazione della tecnica easyIP, il router è caratterizzano da una singola porta WAN con un IP pubblico

4.4.1 NAT Internal Server

L'adozione del NAT è stata criticata da molti esperti, ritenendo si stesse violando il fondamento della connettività *end-to-end* su cui si basa global Internet, difatti il principio che ha guidato gli enti di studio e standardizzazione è quello di far sì che qualsiasi host possa comunicare con qualsiasi altro nodo della rete in ogni momento. Con il NAPT, la più versatile tra le tecniche di traduzione visti, la corrispondenza viene registrata solo in presenza di un un pacchetto inviato da un host verso l'esterno, di conseguenza eventuali comunicazioni in entrata non saranno accettate finchè viene non avviene una connessione verso l'esterno della rete locale. Da un punto di vista pratico, un utente può instaurare connessioni TCP/IP ad un server Web remoto, ma se questo volesse ospitare un server di gioco nella propria rete locale, nessun esterno potrebbe accedervi. Per mitigare questo problema sono stati adottati alcuni accorgimenti che però non risultano sempre risolutivi, oltretutto spesso queste tecniche non risultano pienamente standardizzate, anzi adottano procedimenti proprietari e mal documentati.
Nel caso specifico in cui un nodo di una rete locale voglia accettare delle connessioni dall'esterno per fornire servizi Web HTTP, il NAT può essere configurato per apportare una specifica traduzione: il router, tramite la sua interfaccia pubblica, risponde ad eventuali connessioni in

ingresso su una specifica porta, ad esempio la porta 80, standard nel caso in cui si voglia esporre un servizio HTTP ad utenti esterni. Nella tabella di mapping del router un apposito *record statico preconfigurato* traduce la socket costituita dall'IP pubblico e dalla porta 80 nella socket privata su cui il server espone il servizio (Figura 4.8). L'accorgimento esposto, configurato dall'amministratore di rete, è anche denominato *port forwarding*.

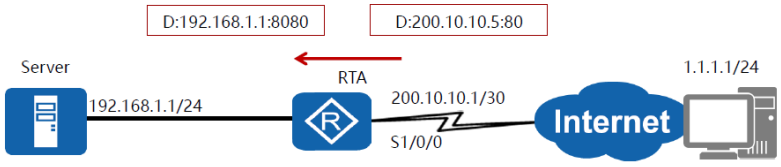

Figura 4.8: Design di rete che accomoda un NAT internal server

A livello pratico, perchè la fruizione del servizio avvenga, è necessario che l'host che desidera fare richieste HTTP (perciò sulla porta 80) al server abbia conoscenza dell'IP pubblico del router su cui è attestata la rete locale del servizio Web, a riguardo nella direttiva *RFC 6886* viene descritto un protocollo che include un metodo per ricavare l'indirizzo pubblico di un NAT gateway, così da permettere al client che desidera accettare connessioni dall'esterno di rendere noto ai propri potenziali peer la socket *IP pubblico:porta* cui risulta raggiungibile.
In ambiti domestici l'interfaccia rivolta alla rete globale del router è caratterizzata da un IP pubblico assegnato **dinamicamente** dall'ISP, l'adozione di un servizio *Dynamic DNS*, di norma configurabile direttamente nel gateway, può facilitare le connessioni di eventuali peer esterni, che possono così individuare il servizio desiderato attraverso un certo *URL*, piuttosto che una socket di volta in volta diversa.

4.5 CG-NAT

Il **Carrier Grade NAT**, noto anche come *NAT su larga scala*, è un approccio alla progettazione delle reti IPv4 in cui i nodi finali, in particolare le reti residenziali, sono configurati con indirizzi di rete privata che vengono tradotti in indirizzi IPv4 pubblici, così da permettere la condivisione di subnet di indirizzi pubblici tra molti clienti. Regolamentato dalla specifica *RFC 6598*, sfrutta l'utilizzo di particolari strumenti di NAT, e si differenzia tra reti di accesso via cavo e reti radiomobili, in queste ultime la tecnica risulta particolarmente diffusa. In questo ambito, un tipico cliente residenziale accederà a global Internet attraverso tre diversi ambiti di indirizzamento IPv4: la rete privata del cliente, la rete privata dell'ISP e l'indirizzo pubblico di Internet.

Come evidenziato in figura 4.9, il router non ha un'interfaccia direttamente esposta alla rete globale, difatti ad essa non è assegnato alcun indrizzo pubblico, bensì sarà caratterizzata da un'IP privato della rete locale del provider. Il router domestico svolge ancora la funzione di NAT e risulta direttamente accessibile all'utente, ma la tabella di mapping presenta corrispondenze tra la socket privata del cliente ed una socket che presenta un indirizzo di un'altra rete locale, dedicata dall'ISP ai propri clienti per raccogliere il traffico e convogliarlo verso dei nodi intermedi che effettuano un secondo grado di traduzione degli indirizzi. Fino a 64 singoli utenti risultano

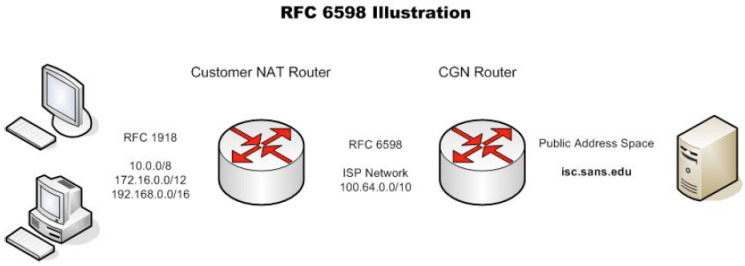

Figura 4.9: Topologia di rete che coinvolge un CG-NAT (link)

mappati sullo stesso indirizzo pubblico che caratterizza l'interfaccia d'*outbound* del nodo di traduzione, in questo modo ogni utente potrà realizzare fino a ad 1k connessioni verso reti esterne. L'adozione di questa tecnica si traduce in un notevole risparmio di indirizzi pubblici: nel design delle reti radiomobili è diventata uno standard *de facto* per consentire a milioni di singoli clienti, autenticati tramite la *SIM* personale, di accedere ad internet, nonostante la risorsa di indirizzi pubblici in possesso all'ISP risulti notevolmente inferiore.
Con il CG-NAT i problemi già evidenziati per il NAT risultano ancora più pronunciati, in particolare non sarà più possibile attuare la tecnica del *port forwading* in quanto la configurazione del nodo che realizza il secondo step di traduzione degli indirizzi non è direttamente accessibile all'utente che desidera fornire un certo servizio a partire dalla propria rete locale. In questo caso la comunicazione *end-to-end* risulta essenzialmente possibile solo con l'ausilio di un **relay server**.
Quanto appena introdotto giustifica perciò l'acquisto del server VPS nell'ambito del progetto svolto per conto dell'azienda Esse-ti, possiamo visualizzarlo nella figura 1.1.

Capitolo 5

VPN

5.1 Generalità ed ambiti d'uso

Le prime release del protocollo TCP/IP non prestavano particolare attenzione alla sicurezza della comunicazione tra dispositivi, ma con l'espansione di internet, la consapevolezza del concetto di privacy online e le necessità di grandi aziende dislocate in sedi geograficamente remote sono state implementate delle soluzioni per la protezione dei dati, delle vere e proprie architetture che garantiscono:

- **confidenzialità**. Impedisce che il contenuto della comunicazione possa essere rilevato da altri, proteggendo anche mittente e destinatario, verificando la lunghezza del messaggio e la frequenza di comunicazione;
- **integrità dei dati**. Può essere garantita con un servizio *Connectionless* che rileva la modifica di un pacchetto IP senza tenere conto dell'ordine dei datagrammi nel flusso di comunicazione, oppure con un servizio *connection oriented* che impone dei vincoli anche sulla sequenza dei pacchetti e può rilevare tentative di riordino degli stessi;
- **autenticazione dell'origine dei dati**. Protegge mittente e destinatario da attacchi che possono degradare la comunicazione.

Si supponga di voler accentrare diversi dipartimenti dislocati fisicamente sotto la stessa rete locale, che quindi verrebbe estesa oltre i limiti fisici dei singoli uffici, per motivi organizzativi, di sicurezza e per permettere lo svolgimento dello *smart working*. Un approccio sicuramente affidabile, in termini di prestazioni e sicurezza, risulta quello di dotarsi di un canale dedicato tra le diverse sedi, ma l'affitto esclusivo di infrastrutture di comunicazione in genere risulta estremamente costoso e non sempre fattibile, ad esempio potrebbe essere necessario commissionare delle opere di scavo o posa di fibra ottica, con oneri in termini di costi e permessi difficilmente ammortizzabili. Per questi motivi, moltissime aziende, specie di piccole e medie dimensioni, impiegano una soluzione differente: la creazione di una rete privata, detta **Virtual Private Network** all'interno di una infrastruttura di rete pubblica, ossia Internet. La creazione di una VPN richiede l'implementazione un insieme di tecnologie che consentono a qualsiasi sede o dipendente di trovarsi virtualmente collegati alla stessa rete locale aziendale indipendentemente dalla loro posizione fisica (Figura 5.1). Le VPN risultano inoltre estremamente configurabili, scalabili ed in generale offrono un ottimo rapporto tra costi e funzionalità, bisogna però dotarsi di un buon accesso alla rete globale, che garantisca delle adeguate prestazioni in termini di velocità,

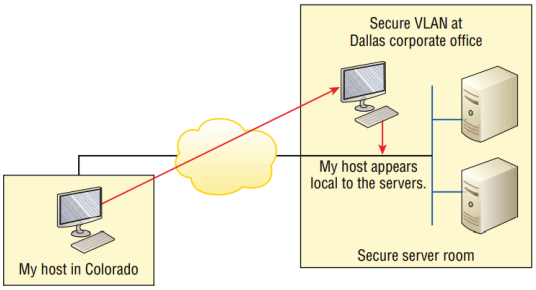

Figura 5.1: schema di principio che si realizza adotando una VPN

latenza e jitter. I singoli utenti che si vorranno connettere alla rete aziendale dovranno essere autenticati secondo un apposito controllo degli accessi e la comunicazione tra i vari nodi della rete virtuale avverrà tramite appositi canali cifrati per garantire la sicurezza delle informazioni. Le VPN quindi affrontano tali obiettivi basando il loro funzionamento su tre diversi fattori:

- **autenticazione**, definisce un processo con cui un nodo verifica la corretta identità di un altro nodo che vuole comunicare attraverso una connessione, per poi concedergli l'autorizzazione a usufruire dei relativi servizi associati mantenendo però traccia delle risorse concesse durante l'accesso;
- **cifratura**, è possibile usare un'ampia gamma di algoritmi di crittografia con delle chiavi segrete appositamente concordate scambiate tra gli *end point* della comunicazione per cifrare il traffico in rete. Nello specifico delle reti VPN viene sopratutto utilizzato il protocollo **Internet Key Exchange**;
- **tunneling**, aggiunge un livello di sicurezza per proteggere ogni pacchetto nel suo viaggio atttraverso la rete pubblica. In maniera del tutto trasparente per l'utente finale, le VPN posso essere configurate in modalità **trasporto**, in cui la protezione è garantita dal software impiegato nei calcolatori finali, oppure in modalità **tunnel**, che offre più garanzie in termini di sicurezza, in cui gli apparati si preoccuperanno di proteggere il traffico tra i due endpoint tramite un opportuno incapsulamento ad entrambe le estremità del tunnel, con un procedimento che risulta del tutto trasparente per l'utente finale.

Esistono due principali tipi di soluzioni VPN adottate in ambito aziendale.

Remote-Access VPN

È una configurazione adata a singoli dipendenti, ad esempio degli agenti, che vogliono accedere alle risorse della sede principale (Figura 5.2). Consente ai singoli utenti di stabilire connessioni

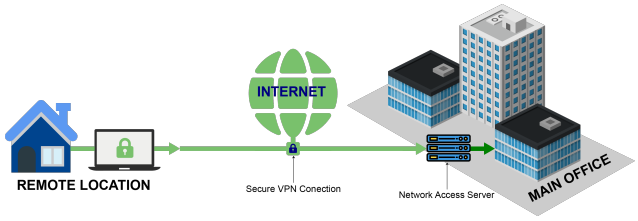

Figura 5.2: VPN ad accesso remoto

sicure con la rete aziendare remota, emulando applicazioni dati, voce e video dalla sede principale all'elaboratore remoto. Gli utenti possono perciò accedere alle risorse protette come se fossero direttamente collegati alla rete aziendale, in quest'ultima dovrà essere presente un server di accesso alla rete, anche detto *NAS* che si preoccuperà di confermare l'identità del dipendente remoto (operazione di *Authentication*), individuerà le risorse cui l'utente può accedere (operazione di *Authorization*) e terrà traccia delle azioni fatte su di esse (operazione di Accounting). Risulta inoltre indispensabile che l'utente abbia debitamente configurato ed installato sul proprio elaboratore, da cui accedere alla rete aziendare, un software VPN Client.

Site-to-Site VPN

Permette di stabilire connessioni sicure attraverso una rete pubblica interconnettendo più sedi aziendali con un quartiere principale. Si creano dei collegamenti tra reti locali per permettere l'uso delle risorse allocate nella sede principale anche alle filiali remote (Figura 5.3). Risulta la configurazione VPN più usata. In base alle esigenze, questo tipo di VPN può essere implementato

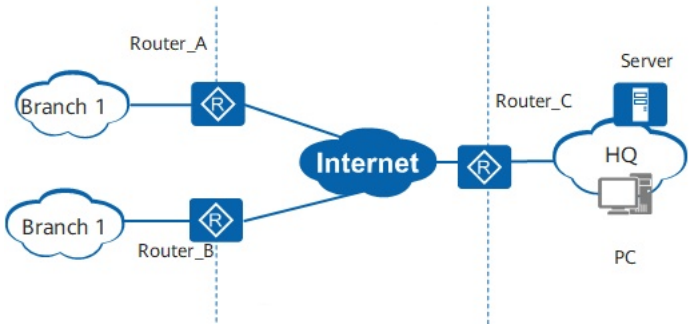

Figura 5.3: Site-to-site VPN

in due diversi modi:

- **Intranet-based** nel caso in cui una società desideri unire le reti locali delle sedi remote in un'unica rete privata;
- **Extranet-based** quando diverse società affiliate vogliono interconnettere le proprie reti locali per lavorare in un'ambiente sicuro accedendo ad alcune risorse comuni, ma i diversi partner mantengono ognuno una propria intranet privata.

In ambito aziendale enterprise, entrambe le soluzioni architetturali viste per le VPN possono essere adottate, come nell'esempio in figura 5.4.

5.2 IPSec VPN

Internet Protocol Security, comunemente denominata **IPSec** è una suite di protocolli definita dall'*IETF* che permette di mettere in sicurezza le comunicazioni che sfruttano IP, in particolare IPSec agisce al livello Network dell'architettura TCP/IP, autenticando e cifrando ogni pacchetto di una sessione di comunicazione tra due *end point*. Risulta essere la soluzione adottata più frequentemente dalle aziende, in quanto consente di realizzare entrambi i tipi di VPN visti, particolarmente comodo come già visto in figura 5.4. IPSec stabilisce una delle convenzioni

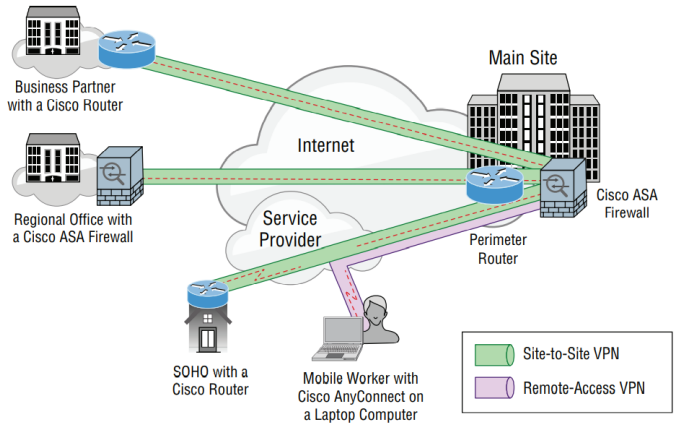

Figura 5.4: Esempio d'uso delle tecnologie VPN in ambito enterprice con soluzioni Cisco Systems

di sicurezza tra tutti i peer della comunicazione, inoltra il flusso di dati da proteggere nel tunnel sicuro, infine usa dei protocolli di sicurezza per cifrare ed autenticare i dati che viaggiano attraverso il tunnel.

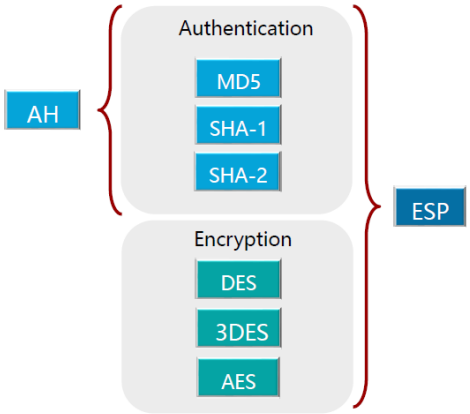

Figura 5.5: Protocolli AH ed ESP con evidenza degli algoritmi adottati

I protocolli principali che costituiscono IPSec sono tre:

- **Authentication Header (AH)**: garantisce l'autenticazione per l'origine dei dati e l'integrità del messaggio (con un servizio *connectionless*) ma non offre la confidenzialità (Figura 5.5);
- **Encapsulating Security Payload (ESP)**: può fornire autenticazione (in maniera parziale rispetto all'analogo servizio offerto da AH), confidenzialità, integrità del messaggio e un limitato controllo del flusso dati. In particolare la bontà del servizio di confidenzialità offerto dipende anche dall'algoritmo crittografico adottato (Figura 5.5);
- **Internet Key Exchange (IKE)**: implementa lo scambio delle chiavi per realizzare il flusso crittografico, in particolare permette di automatizzare il processo di scambio. Risulta essere un protocollo di livello Application, a differenza dei precedenti.

Quando un host o un router invia un datagramma IPsec, di norma lo fa usando AH oppure ESP, in alcuni contensti però vengono usati entrambi i protocolli di sicurezza, ognuno individua

appositi header aggiuntivi per il datagramma.
Nel momento in cui due peer devono scambiare dati tramite la VPN, è necessario instaurare *prima* una **connessione logica** tra loro, così da condividere le **policy** di sicurezza da utilizzare per proteggere il traffico, quali ad esempio l'algoritmo di crittografia, le chiavi, la modalità di verifica dell'integrità dei dati trasmessi, ecc. (Figura 5.6). Tale connessione, creata a livello Network, è detta **Security Association (SA)** e può essere stabilita manualmente o mediante il protocollo IKE. Le SA sono unidirezionali, ossia definiscono i parametri per stabilire la connes-

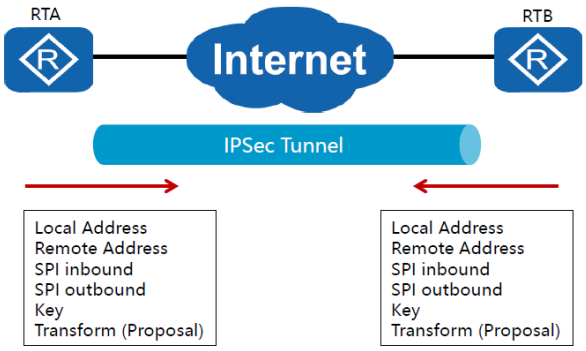

Figura 5.6: Security Association stabilite tra due peer IPSec

sione sicura in un'unica direzione. Nell'esempio di figura 5.6 sarà quindi necessario stabilire una SA tra il router A ed il router B, così come un'associazione tra il router B ed il router A. Oltre ad essere monodirezionali, è possibile attribuirvi solo un unico servizio di sicurezza, perciò nel caso si vogliano applicare sia AH che ESP al traffico dati tra due peer, devono essere stabilite almeno due SA per ogni endpoint della connessione. Le Security Association sono in genere definite automaticamente attraverso IKE, che si può servire di protocolli, quali ad esempio *ISAKMP* e *OAKLEY*, per proteggere lo scambio delle SA ed autenticare il peer. Per rafforzare i vincoli di sicurezza, le SA necessitano di essere rinegoziate dopo un certo tempo o dopo una certa mole di traffico protetto, IKE si preoccuperà di negoziare nuove associazioni al momento opportuno.

Sia AH che ESP possono essere utilizzati in modalità **trasporto** oppure in modalità **tunnel**. Nel primo caso (Figura 5.7) verranno aggiunti gli header dei protocolli visti tra l'header IP e l'header del protocollo di trasporto (TCP o UDP), così da fornire protezione principalmente ai protocolli applicativi, in particolare il protocollo ESP fornisce servizi di sicurezza esclusivamente al segmento TCP ed il suo payload informativo tramite un header ed un trailer, al contrario AH fornisce protezione a selezionate porzioni ed opzioni dell'header IP. Nel caso la VPN sia configurata in modalità tunnel, il pacchetto IP originario, dove sono specificati gli indirizzi di sorgente e destinazione effettivi, viene interamente incapsulato in un nuovo pacchetto prima di essere inviato, in questo modo il datagramma esterno, caratterizzato dagli indirizzi IP dei due lati del tunnel, si comporta da *carrier protocol*, proteggendo il contenuto del pacchetto originale (*passenger protocol*) che rimarrà invariato tranne per un decremento del campo *TTL*, come se si muovesse come all'interno di un tunnel virtuale nel transito attraverso Global Internet. I dispositivi di rete su entrambe le estremità del tunnel provvedereanno ad incapsulare i pacchetti in uscita e riaprirli in entrata, in particolare gli header relativi al protocollo IPsec saranno inseriti prima dell'header IP originale del pacchetto e il datagramma che otteniamo sarà il payload di un nuovo pacchetto IP realizzato aggiungendo un ulteriore Header IP prima dell'header AH e/o ESP (Figura 5.8).

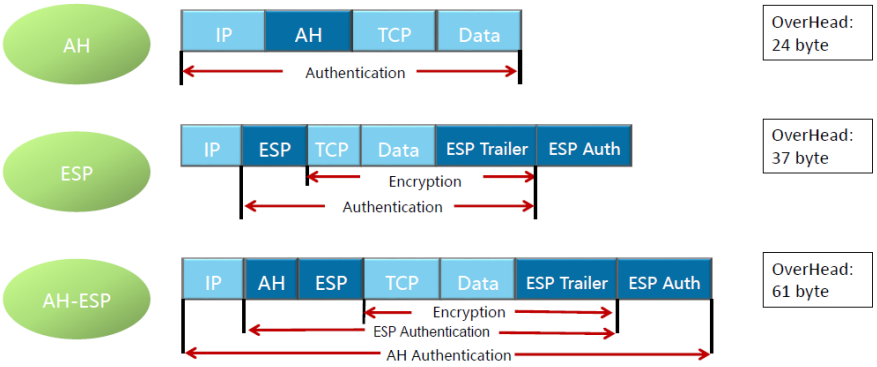

Figura 5.7: IPSec utilizzato in modalità trasporto applicando il protocollo AH e/o ESP

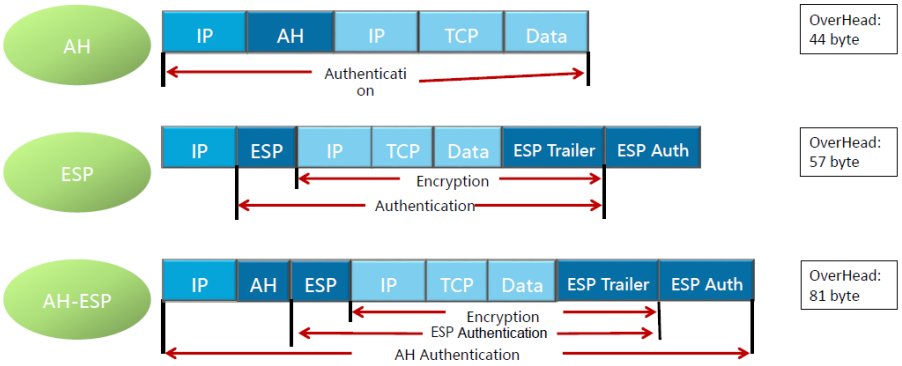

Figura 5.8: IPSec utilizzato in modalità tunnel applicando il protocollo AH e/o ESP

Dalle figure 5.7 e 5.8 è possibile notare come in entrambe le modalità il protocollo AH fornisca autenticazione, calcolando un *checksum* con una certo algoritmo, all'intero pacchetto, incluso l'header IP esterno in modalità tunnel. Al contrario, per quanto riguarda il protocollo ESP, ESP non sottopone a controlli di autenticità ed integrità l'header IP esterno, che nella modalità trasporto sarà l'header originale del pacchetto, mentre nella modalità tunnel sarà l'header aggiunto per realizzare un incapsulamento. Collegandoci a quanto visto nel capitolo 4, è lecito chiedersi se, visto che il NAT modifica i pacchetti che provvede ad instradare, sia possibile stabilire una IPSec VPN dove uno o più host risultano inseriti in una design di rete che prevede la traduzione degli indirizzi. Viste le considerazioni fatte, l'impiego del NAT nell'ambito di una IPSec VPN che implementa il protocollo AH risulta incompatibile in ogni caso, dato che la traduzione degli indirizzi viola l'integrità del pacchetto e perciò comporta un checksum differente. Al contrario, nel caso in cui per la VPN venga adottato il protocollo ESP la convivenza con il NAT è possibile grazie ad alcune tecniche *ad-hoc*, come il **NAT Traversal**, che si basano sull'incapsulamento in una datagramma UDP, in particolare l'header di livello Transport viene inserito tra l'header IP esterno e l'header ESP. Per capire quando attuare questo accorgimento, nelle fasi iniziali in cui si condividono le Security Association durante le negoziazione IKE, i peer determinano se sono in grado di supportare NAT-T, quindi verificano chi dei due subisce il NAT, o al peggio entrambi.

Capitolo 6

Caratteristiche del design di rete adottato

Nei capitoli precedenti sono stati evidenziati diversi aspetti che sono risultati fondamentali nel comprendere come avviene la comunicazione e quali tecnologie bisogna adottare per realizzare lo scopo del progetto posto dall'azienda *Esse-ti*.

A causa del coinvolgimento di almeno un CG-NAT data la connessione radiomobile del router 4G, l'acquisto dell'istanza VPS si è reso necessario per farne l'uso di **relay server**. Questa risulta la soluzione più affidabile, per quanto non la più efficiente dal punto di vista della latenza, che permette la comunicazione *end-to-end*.
Si supponga che un generico host A voglia comunicare con un destinatario remoto B residente in una rete locale connessa ad Internet tramite CG-NAT per mezzo del gateway 4G. Si evidenzia inoltre la presenza di un NAT presso la rete locale dell'host A ma anche presso la rete locale del destinatario B, dove la traduzione degli indirizzi è effettuata proprio dal router 4G. Un eventuale tentativo di instaurare una connessione diretta risulterebbe fallimentare anche nel caso di una configurazione di port forwarding applicata ai NAT presenti ad entrambi gli estremi della comunicazione, questo perchè nel CG-NAT dell'ISP non risulta registrata alcuna corrispondenza relativa all'host B per la porta richiesta dal mittente. Oltretutto non è possibile accedere al CG-NAT per apporre evenutali soluzioni, l'accesso a questi nodi risulta essere a totale appannaggio dell'ISP. Coinvolgendo il server di relay i due client andranno a stabilire singolarmente una connessione TCP con il servizio remoto, definendo nelle tabelle di mapping, sia nel NAT locale che nel CG-NAT, le corripondenze che permettono di stabilire un canale di comunicazione bidirezionale tra ogni singolo host ed il server. A questo punto, mantenedo le connessioni TCP attive, il server provvederà ad inoltrare i messaggi che riceve da un host all'altro, e viceversa se necessario.
Finchè i due nodi potranno stabilire una connessione con il server, la comunicazione sarà garantita a prescindere dai dettagli del design di rete, difatti questo approccio funziona anche nel caso entrambi gli *end point* siano mascherati da un doppio nat, o anche nel caso non siano coinvolti dei CG-NAT ma l'utente risulta impossibilitato a configurare il port forwarding sul proprio gateway (spesso l'opzione risulta non disponibile o bloccata negli apparati forniti dai provider). Ovviamente la necessità di un server per portare a compimento una comunicazione comporta dei costi aggiuntivi che, specie in ambito consumer, non risultano indifferenti, così come un consumo aggiuntivo di banda. Nel nostro specifico scenario, l'applicazione cui è volto il progetto giustifica la spesa.

Un altro importante obiettivo delineato con l'azienda sono stati i requisti di sicurezza, difatti le applicazioni domotiche dei clienti raggiungibili da remoto possono coinvolgere funzioni sensibili di interesse pubblico o privato, come stazioni di energia, sistemi di allarme e rilevazione incendi, inquinanti ecc. Risulta perciò dirimente garantire una comunicazione sicura, quindi si è deciso di adottare una configurazione che comprende una VPN site-to-site, nel dettaglio implementando una particolare declinazione software offerta da **OpenVPN**, una società leader nel mercato della sicurezza che si distingue per fornire una suite software completamente *open source* estremamente duttile che si pone quindi allo scopo progettuale.
OpenVPN fornisce una suite VPN basata sui protocolli **SSL/TLS**, così da implementare le tecniche di sicurezza già viste nel capitolo 5 basandosi su protocolli di livello Session, ciò permette un ampio campo d'uso nelle situazioni più differenti: il canale sicuro è creato tra due applicazioni, viene protetto il traffico TCP fino alla consegna allo strato interessato del mittente, al contrario IPSec protegge tutte le informazioni sopra al livello Network e termina il suo compito quando i dati vengono consegnati al mittente, risulta perciò più indicato per blindare il traffico tra intere sottoreti. Stante la natura di SSL/TLS, vi è la necessità di usare un protocollo di trasporto affidabile come TCP. Essendo SSL/TLS un protocollo Client/Server che ha lo scopo di autenticare il server da parte del client e anche il client da parte del server, OpenVPN permette l'autenticazione basandosi su certificati digitali riconosciuti da una *Certification Authority*, le due parti della comunicazione autenticano così singolarmente il loro interlocutore confrontando la *firma digitale* del certificato ricevuto con quella della CA che lo ha validato. In questa fase della verifica delle CA vengono scambiate le chiavi di cifratura che permettono la creazione di un canale sicuro su cui avviare la comunicazione. Riassumendo, OpenVPN procede ad autenticare gli estremi del tunnel mediante dei certificati (ma è supportato anche l'uso di credenziali e di supporti smartcard) ed alla creazione delle chiavi attraverso l'uso di SSL/TLS al posto del protocollo IKE già visto per l'IPSec.
OpenVPN nella sua ultima major release, che è stata adottata nel progetto, permette anche di essere configurato per offrire una modalità Client/Server scalabile, in cui una moltitudine di clienti, tramite il software fornito, possono connettersi ad una singola istanza server OpenVPN mediante una singola porta TCP.

Nell'ambito del progetto, a partire dalla topologia di rete effettiva rappresentata in figura 1.1, si è provveduto ad implementare la suite OpenVPN in modalità site-to-site, in particolare provvedendo a configurare l'istanza server sul sistema operativo equipaggiante la VPS, ossia Ubuntu, in modo che avesse un comportamento simile a quanto descritto all'inizio per il relay server, ma facendo sì che la comunicazione risultasse sicura e che i singoli host non avessero conto che i dati transitassero attraverso global Internet.

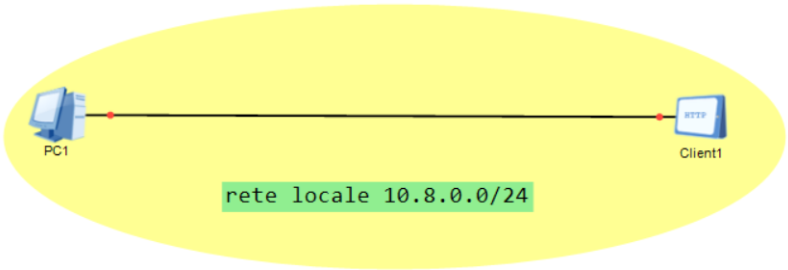

Figura 6.1: Topologia logica che viene vista dai due end-point in comunicazione

A livello logico quindi ogni nodo ha la percezione di comunicare con un suo pari attinente alla stessa rete locale, difatti la sua tabella di raggiungibilità presenta una rotta statica che lo informa

della presenza dell'host di destinazione nello stesso ambito locale. In Figura 6.1 è possibile notare la topologia logica, i due peer della comunicazione sono caratterizzati ognuno da un indirizzo privato univoco che è stato loro assegnato dal server OpenVPN, in particolare si è scelto di usare la subnet *10.8.0.0/24*.

Considerando la configurazione di rete, perchè i due host siano in grado di effettuare tale comunicazione diretta, il server OpenVPN, raggiungibile tramite l'ip pubblico della VPS, deve stabilire due singoli tunnel sicuri con entrambi gli host, in particolare con entrambe le applicazioni OpenVPN client installate sugli host coinvolti (come evidente in figura 6.2), quindi dovrà comportarsi da router, provvedendo al forwarding delle richieste tra un nodo e l'altro. Ogni host è caratterizzato da un certo indirizzo privato caratteristico della rete locale virtuale che crea la VPN, il server invece a livello logico non sarà individuato da alcun indirizzo, in modo da risultare invisibile ai client nella loro comunicazione diretta.

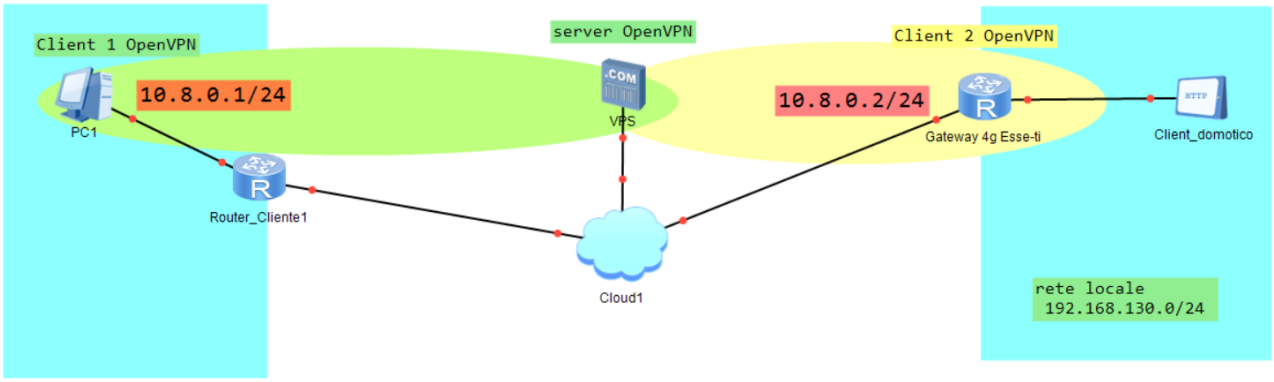

Figura 6.2: Topologia di rete che evidenzia i due tunnel stabiliti con OpenVPN tra ogni client con il server

Questa particolare realizzazione và quindi a sfruttare le capacità che il software OpenVPN offre per instradare pacchetti in maniera sicura tra sedi diverse, ma a differenza delle comuni implementazioni non si individuano delle filiali che devono accedere a risorse fisicamente poste in una sede centrale, dove risiede il server OpenVPN che permette la comunicazione site-to-site, l'obiettivo è garantire la comunicazione *client-to-client* attraverso il server.

In questa prima fase è fondamentale porsi come obiettivo di stabilire **singolarmente** i tunnel tra il server OpenVPN, sempre rappresentato dallo stesso processo sull'istanza VPS, e i due client. Di questi, il primo sarà un calcolatore, che magari potrebbe essere attinente alla rete locale dell'azienda Esse-ti, il secondo sarà il gateway 4G cui è collegato, ad esempio, un dispositivo domotico con cui realizzare la comunicazione. La scelta della subnet caratterizzante la rete locale virtuale creata dalla VPN non è banale, è necessario evitare sovrapposizioni con le differenti reti locali coinvolte nella VPN che potrebbero essere configurate per usare la stessa sottorete privata. Ciò potrebbe comportare un problema di *routing conflict* perchè l'host non è in grado di discernere, nella sua tabella raggiungibilità, il record che individua il traffico diretto nella VPN da quello che identifica il traffico volto alla rete effettiva locale in cui si trova l'host. In questo caso, la miglior soluzione possibile è configurare il server VPN perchè adotti una subnet che di norma non viene usata in altri ambiti, seppur sempre attinente ad una classe di indirizzi privati. Nel caso di analisi, si è scelto di usare, come già evidenziato, la sottorete *10.8.0.0/24*.

Gli step iniziali per costruire la configurazione OpenVPN includono la definizione di una *master Certificate Authority* con una relativa chiave, quest'ultima necessaria per firmare i certificati da rilasciare al client ed al server stesso. La CA viene creata di norma a partire dal modello

easy-rsa messo a disposizione da OpenVPN stessa.
Sempre nell'ambito del server vengono definiti certificati, anche detti **chiavi pubbliche** e **chiavi private** dedicati al server stesso ed ai due client previsti nella configurazione. È interessante notare che il server non ha bisogno di avere conoscenza dei singoli certificati che i client possono presentare a lui, difatti dopo aver creato le chiavi pubbliche dedicate ai due client, queste verranno loro trasferite. Il server OpenVPN accetterà connessioni solo dai client i cui certificati saranno firmati dalla *master Certification Authority* con un confronto che non necessita accesso alla chiave privata relativa alla CA, che quindi può essere custodita in una macchina sicura, anche disconnessa dalla rete. L'insieme di questo processo di creazione di chiavi e certificati viene garantisce l'istaurazione di una **infrastruttura a chiave pubblica**. Al termine di questa fase,in un'apposita directory del server saranno presenti una moltitudine certificati e chiavi private, in particolare sarà necessario, come evidente nella tabella di figura 6.3.

Filename	Needed By	Purpose	Secret
ca.crt	server + all clients	Root CA certificate	NO
ca.key	key signing machine only	Root CA key	YES
dh{n}.pem	server only	Diffie Hellman parameters	NO
server.crt	server only	Server Certificate	NO
server.key	server only	Server Key	YES
client1.crt	client1 only	Client1 Certificate	NO
client1.key	client1 only	Client1 Key	YES
client2.crt	client2 only	Client2 Certificate	NO
client2.key	client2 only	Client2 Key	YES

Figura 6.3: Tabella che mostra i file creati in sede al server, ne discerne la natura di chiave privata o pubbica e mostra dove vanno trasferiti (link)

Chiavi e certificati relativi ad ogni client possono essere condensati in un unico script di configurazione, quest'ultimo viene compilato a partire da una base di partenza fornita da OpenVPN stessa andando ad impostare tutte le direttive necessarie al corretto funzionamento del client OpenVPN nel caso specifico. Il risultato della compilazione sarà un file unico caratterizzato dall'estensione *ovpn*. Nel caso sia necessario aggiungere più client alla configurazione, sarà necessario provvedere alla creazione di un certificato ed una chiave privata a esso dedicati, che saranno inclusi nel file di configurazione che viene trasferito all'host remoto attraverso un protocollo di comunicazione sicuro, come *SFTP*, concludendo così il processo di **Private Key Infrastructure**. Si può notare come questo passaggio sia critico, quindi il trasferimento delle informazioni deve sempre essere effettuato sfruttando un protocollo che fornisce rigide certezze in termini di sicurezza.

A seguito del processo esposto, sarà necessario provvedere al *tuning* del file di configurazione del server per regolare tutte le opzioni che permetteranno la comunicazione così come desiderata,

ma prima bisognerà permettere il forwarding dei pacchetti a livello dell'intera macchina OVH ed installare l'istanza firewall *ufw*, consentendo il traffico della VPN attraverso *ufw* abilitando la porta *1194*. Tra le direttive impostate nel file di configurazione del server, riveste particolare importanza l'abilitazione dell'opzione *client-to-client*. È questa impostazione che effettivamente permette al nostro server OpenVPN di effettuare operazioni di routing per interconnettere virtualmente i due client (Figura 6.4), senza di essa la comunicazione sarebbe limitata tra l'host che implementa l'istanza client OpenVPN, che si trova ad una estremità del tunnel, e un eventuale host connesso nella rete locale ove si trova il server OpenVPN, quindi dall'altra estremità del tunnel. Nel caso tale opzione risultasse disabilitata, i pacchetti diretti da un client all'altro

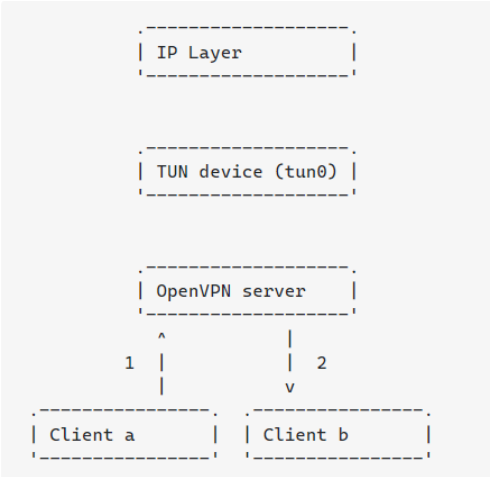

Figura 6.4: Effetto dell'opzione *client-to-client* sulla comunicazione visuallizzata sull'architettura del'istanza VPS (link)

attraverso i tunnel instaurati saranno consegnati solo a seguito di una corretta configurazione delle tabelle di routing implementate dal livello Network della macchina che ospita il server OVH (Figura 6.5).

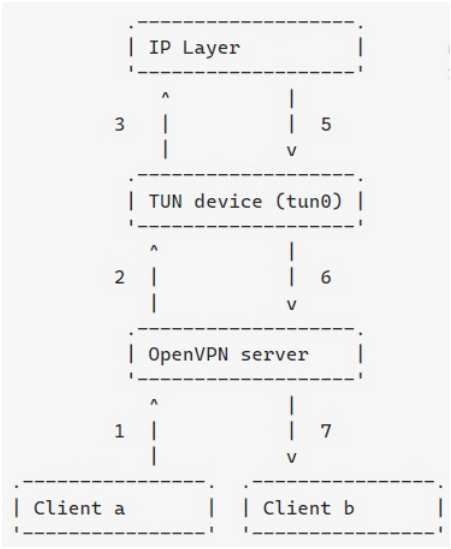

Figura 6.5: inoltro dei pacchetti attraverso le tabelle di routing del'istanza VPS (link)

A tal riguardo tale comportamento non viene escluso nel caso di progetto, il forwarding dei

pacchetti è stato abilitato per realizzare altri scopi secondo le istruzioni dettagliate da OpenVPN, ma la documentazione fornita non chiarisce il comportamento del software nel caso entrambe le opzioni citate siano abilitate, seppur per realizzare scopi differenti. Nelle prove effettuate, l'opzione *client-to-client* è risultata necessaria per portare a termine la comunicazione, ma anche l'abilitazione dell'ip forwarding.

Per le configurazioni apportate finora, è stato possibile mettere in comunicazione un host, supposto ad esempio nella rete aziendale Esse-ti, con il router 4G cui fa capo la rete locale dove risiede il dispositivo domotico da raggiungere, si è quindi realizzato un modello che permette la comunicazione *point-to-point* tra i due client, individuati ognuno da un indirizzo locale di una subnet virtuale che caratterizza l'istanza VPN definita. Per completare l'obiettivo di progetto risulta necessario far sì che, previa un opportuna configurazione delle tabelle di routing del gateway 4G mediante le impostazioni del firewall e del port forwarding, un qualsiasi device nella rete *192.168.130.0/24* risulti raggiungibile dall'altro client della VPN. In particolare è necessario che tale subnet risulti univoca nel contesto di rete, ossia non ci devono essere altri client OpenVPN che usano la stessa sottorete, inoltre il router 4G deve avere un *Common Name* unico indicato nel suo certificato, supponiamo questo sia *client2*.
In termini di operazioni sul server OpenVPN, è necessario specificare un preciso file di configurazione con un certo *Common Name* indicato che verrà adottato nel caso si autentichi un client presentante lo stesso identificativo. All'interno del file viene specificata, con il comando *iroute*, un'entry per la tabella di routing del server che informa come la sottorete *192.168.130.0/24* debba essere instradata al gateway 4G, ossia l'host caratterizzato dal nome *client2*. Questa istruzione quindi permette ad OpenVPN di gestire un evenutale pacchetto diretto ad un client caratterizzato da una subnet di destinazione a lui non nota, dando conoscenza al server quale sottorete risiede dietro ad un certo client OpenVPN. È anche necessario specificare alcune direttive nel file principale di configurazione del server, in particolare:

- un'istruzione *route* che, facendo riferimento alla figura 6.5, regola l'instradamento del pacchetto diretto alla subnet *192.168.130.0/24* in modo che esso venga gestito dal layer applicativo che implementa il server OpenVPN, attraverso quindi l'estremità del tunnel;
- un'istruzione *push* che provvede ad informare della presenza della sottorete *192.168.130.0/24* dell'host *client2* gli altri client OpenVPN, andando quindi ad aggiungere un record nelle loro tabelle di raggiungibilità.

La configurazione del server OpenVPN risulta essenzialmente terminata, al contrario i client in genere non necessitano di ulteriori impostazioni oltre quelle che vengono loro recapitate attraverso il file di configurazione, ma è doveroso notare come per il gateway 4G vi sia la necessità di effettuare delle modifiche al firewall. Nello specifico, come mostrato in figura 6.6, bisogna creare una nuova *zona* denominata *vpn* che permetta ai pacchetti consegnati al router da altri client OpenVPN, e quindi caratterizzati da un indirizzo della subnet della VPN, di essere inoltrati nella zona che indentifica la rete locale *192.168.130.0/24*, chiamata *lan*. In modo del tutto speculare, per la zona creata dovrà essere consentito l'ingresso di eventuali datagrammi originati dal client domotico diretti ai client VPN e quindi provenienti dalla zona della rete locale.
A seguito della nuova zona creata, è necessario creare una relativa *regola di traffico* che definisce il flusso dei pacchetti transitanti attraverso il firewall. La procedura guidata che viene presentata attraverso l'interfaccia grafica è mostrata in figura 6.7.
In particolare si andrà a selezionare come zona di sorgente la *vpn*, e come destinazione la zona *lan*. Il relativo indirizzo di sorgente comprenderà ogni nodo delle sottorete con cui è stata configurata la VPN, quindi *10.8.0.0/24*, mentre come IP di destinazione sono permessi tutti

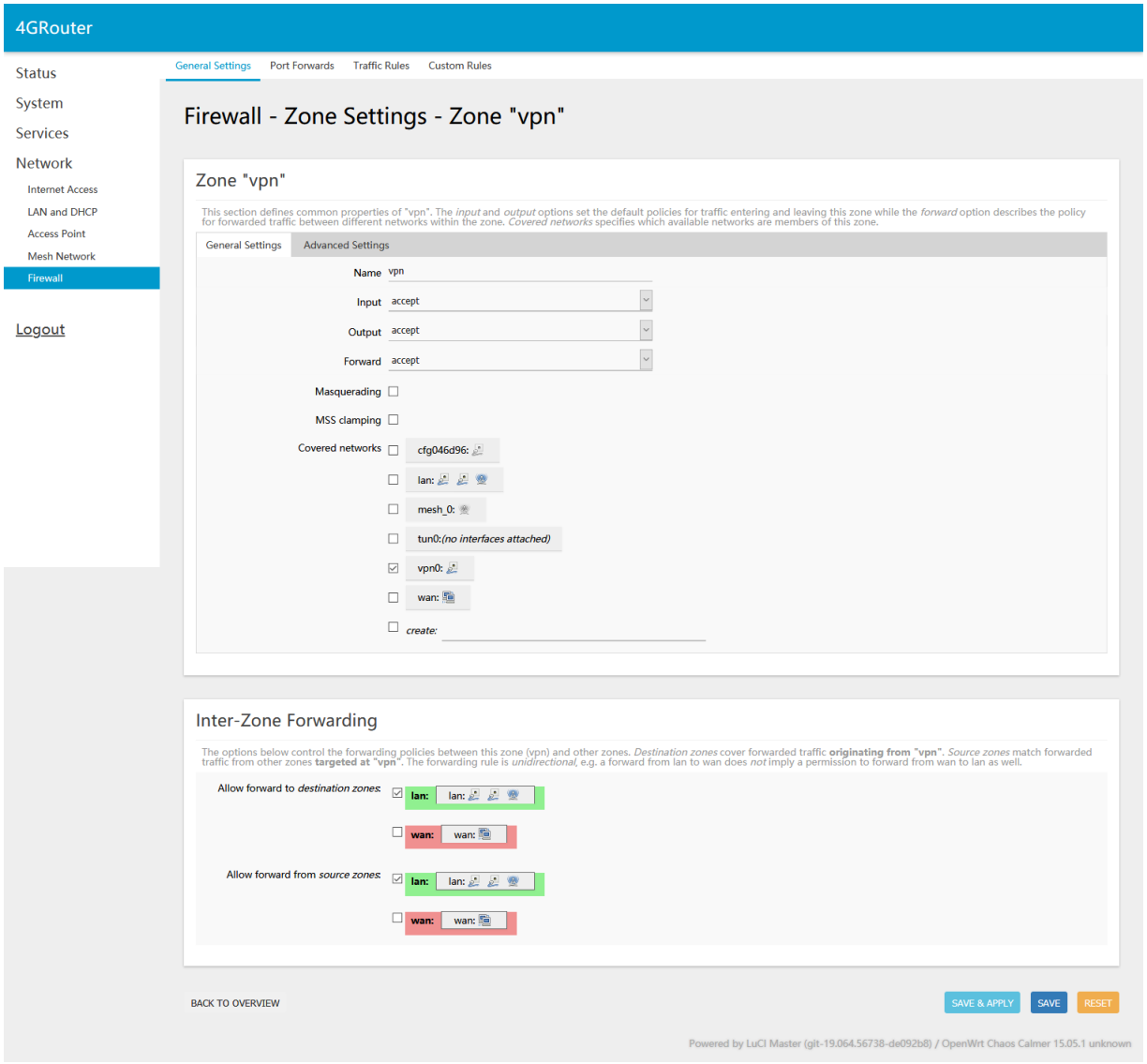

Figura 6.6: Interfaccia grafica del router 4G che permette la creazione di una nuova zona per il firewall

gli indirizzi possibili della rete locale dove risiede l'eventuale (o gli eventuali) end-point della comunicazione desiderata.

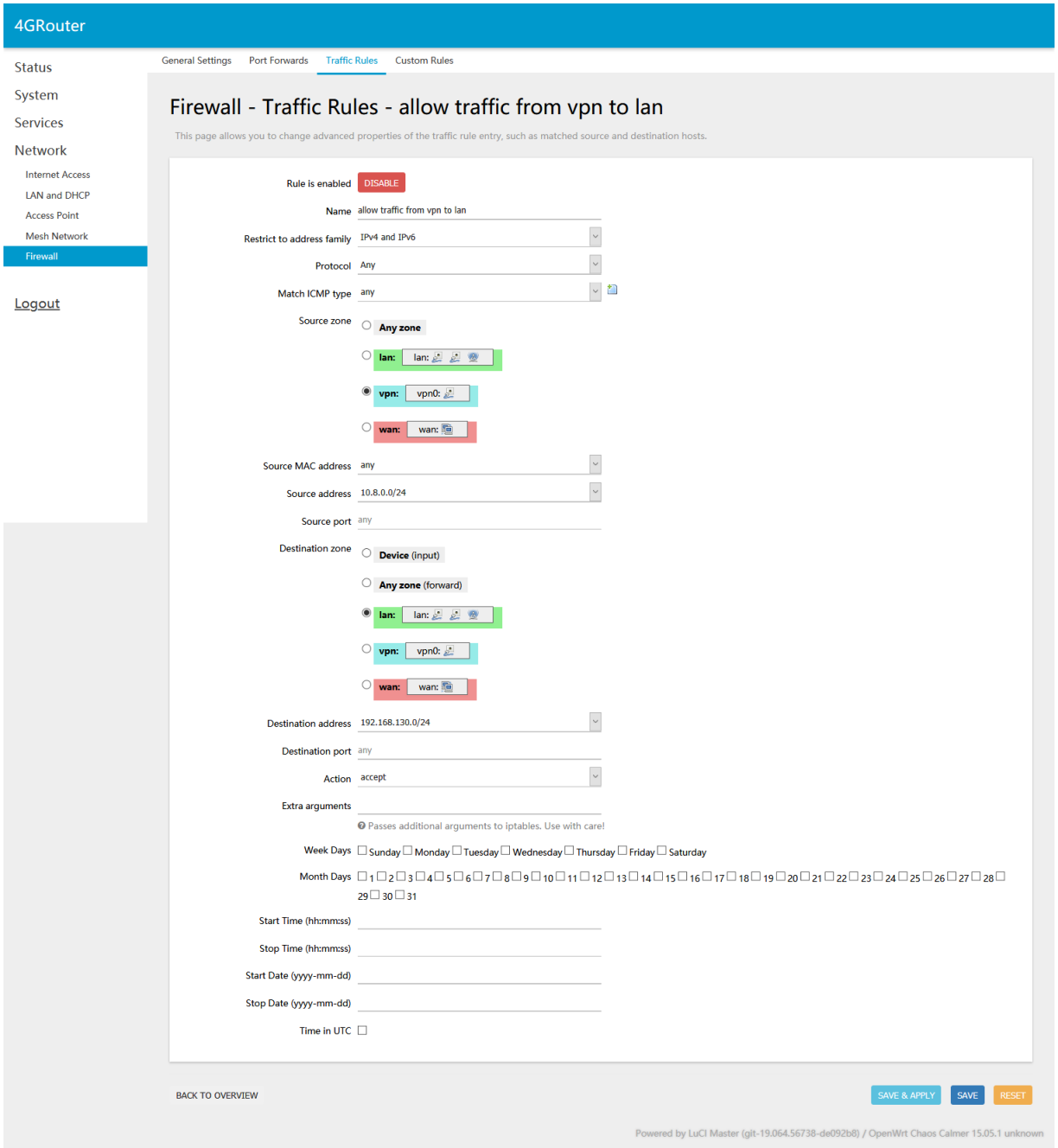

Figura 6.7: Interfaccia grafica del router 4G che permette la creazione di una nuova regola di traffico

Capitolo 7

Conclusione

In questa tesi è stato possibile arricchire in ambito applicativo le conoscenze acquisite sulle reti e sul networking nell'ambito della certificazione "*HUAWEI HCIA Routing and Switching*".

L'occasione di confrontarsi con un'azienda del territorio per chiarire le necessità in cui le nozioni teoriche devono essere declinate, nonchè il coinvolgimento di dinamiche quali limiti di budget, confronto ed analisi sui requisiti, modifiche aggiuntive ed anche le difficoltà determinate dalla pandemia sono sicuramente state fonte di esperienza e motivazione.

Il lavoro in team con i miei colleghi si è rivelato determinante per superare alcune situazioni complesse ed anche la coordinazione del lavoro, la definizione di obiettivi e la verifica dei passi svolti è stata parte integrante dell'esperienza formativa.

Le soluzioni adottate per realizzare il progetto si sono rivelate adeguate e funzionanti, benchè la suite OpenVPN sia risultata documentata in maniera poco approfondita.

L'implementazione di ulteriori dettagli per la configurazione della VPN, in particolare nel riguardo del firewall, verranno trattati in maniera scrupolosa dai miei colleghi, così come l'analisi dei vincoli e delle proposte aziendali che hanno determinato la scelta di acquistare il servizio VPS offerto da *OVHCloud*.

Per poter realizzare ulteriori necessità emerse in fase di confronto con il responsabile aziendale, un possibile sviluppo del progetto comprende l'implementazione di un'architettura ad istanze multiple, in modo da permettere, con l'unico servizio acquistato, la gestione di connessioni indipendenti tra i clienti ed i propri dispositivi domotici. Questo requisito richiederà la virtualizzazione di più processi openVPN sull'istanza remota VPS, così da realizzare differenti istanze VPN, totalmente indipendenti tra di loro per mezzo di uno specifico set di impostazioni.
In figura 7.1 è apprezzabile un prototipo della topologia di rete necessaria a realizzare tale obiettivo.

Per il resto, questo progetto è risulta didatticamente completo, dato che ha fornito l'occasione di fare ulteriori approfondimenti riguardo al funzionamento logico delle reti, così come alle soluzioni implementate dagli *ISP* strutturati. Sono stati esplorati ulteriori tematiche anche riguardo alle soluzioni software, in particolare sul sistema operativo Linux e la gestione del suo firewall, così come riguardo ai software dedicati agli apparati coinvolti, in particolare il router 4G.

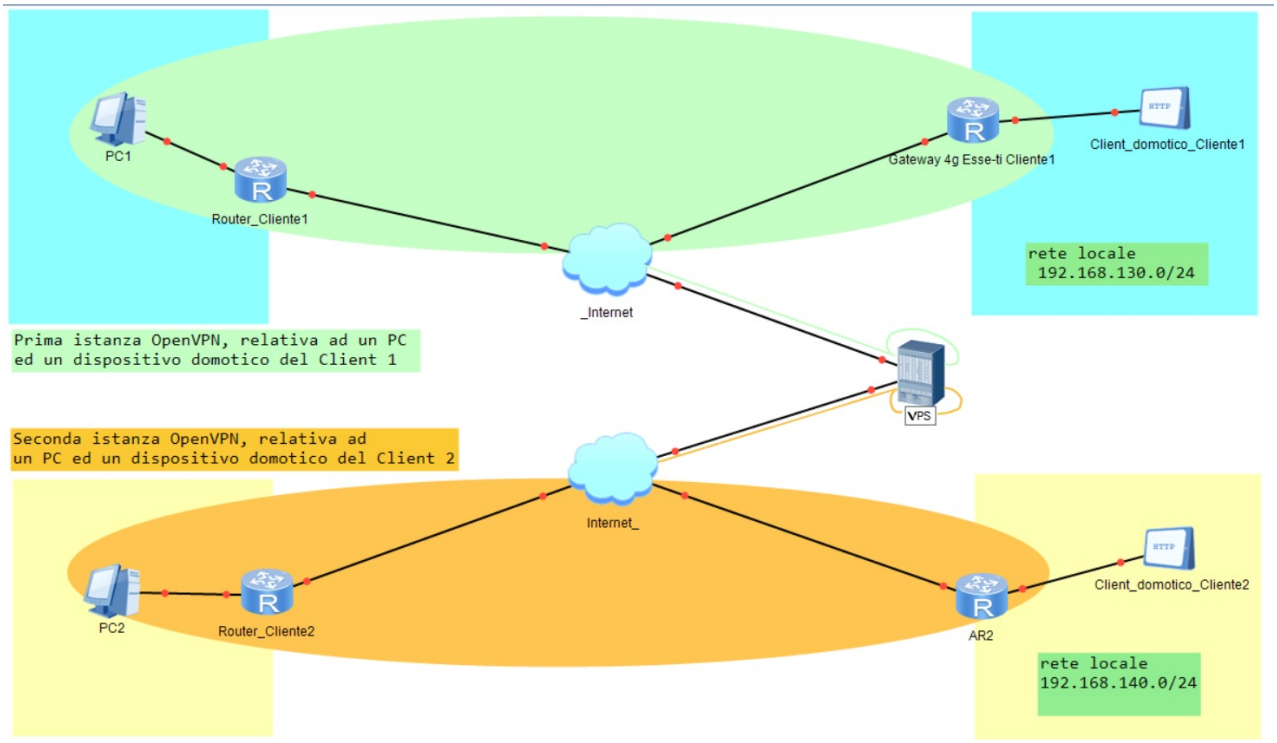

Figura 7.1: Prototipo di topologia di rete a supporto dell'archiettura multistanza

Bibliografia

[1] Slide e dispense di preparazione alla certificazione *HCIA Huawei* versione *2.2* e versione *2.5*

[2] Slide *Elementi di Networking* del corso *Sistemi di Telecomunicazione* tenuto dall'Ing. De Santis nell'Anno Accademico 2019/2020

[3] Andrew S. Tanenbaum, Nick Feamster, David J. Wetherall (2021) *Computer Networks, Global edition*, Pearson Education Limited, 6th ed.

[4] James F. Kurose, Keith Ross (2021) *Computer Networks: A Systems Approach*, Morgan Kaufmann Publishers, 6th ed.

[5] Larry L. Peterson, Bruce S. Davie (2021) *Computer Networking: A Top-Down Approach, Global Edition*, Pearson Education Limited, 8th ed.

[6] Todd Lammle (2020) *Understanding Cisco Networking Technologies: Exam 200-301*, Sybex Inc, John Wiley & Sons Inc, 1th ed.

[7] Todd Lammle (2020) *CCNA Certification Study Guide: Exam 200-301*, Sybex, John Wiley & Sons, Inc, 1th ed.

[8] https://www.esse-ti.it/4g-router (Figura 1.5)

[9] http://www.csc.uvic.ca/~wkui/Courses/networks/0-Introduction.ppt (Figure 2.1, 2.2, 2.3, 2.5, 2.6, 2.15)

[10] https://blog.vayu.it/wp/index.php/2020/05/25/cgn-carrier-grade-nat-con-routeros/ (Figura 4.9)

[11] https://www.ovhcloud.com/it/vps/

[12] https://www.esse-ti.it/images/router/pdf/5IG-600_IT_1.pdf

[13] https://it.wikipedia.org/wiki/Dominio_di_broadcast

[14] https://it.wikipedia.org/wiki/Virtual_private_server

[15] https://en.wikipedia.org/wiki/Routing_loop

[16] https://it.wikipedia.org/wiki/Tabella_di_routing

[17] https://it.wikipedia.org/wiki/Time_to_live

[18] https://en.wikipedia.org/wiki/Network_address_translation

[19] https://en.wikipedia.org/wiki/NAT_traversal

[20] https://en.wikipedia.org/wiki/Carrier-grade_NAT

[21] https://datatracker.ietf.org/doc/html/rfc6886

[22] https://cybertecz.in/what-is-a-vpn-what-are-the-different-types-of-virtual-private-network/ (Figura 5.2)

[23] https://it.wikipedia.org/wiki/IPsec

[24] https://support.huawei.com/enterprise/en/doc/EDOC1000177805/abb422ea/optional-configuring-nat-traversal (Figure 6.4, 6.5)

[25] https://openvpn.net/community-resources/how-to/

[26] https://serverfault.com/questions/736274/openvpn-client-to-client

La data di ultima apertura di tutti i link presenti nella bibliografia è: 3 dicembre 2021

Ringraziamenti:

Innanzitutto devo ringraziare la mia famiglia, specialmente i miei genitori per avermi sostenuto sia economicamente che emotivamente, trasmettendomi tutti i valori e la forza che mi hanno portato a raggiungere questo traguardo.

Un ringraziamento speciale va anche alla mia ragazza, per esserci sempre nei momenti di bisogno, per sopportarmi anche dopo mesi di lontananza.

Un dovuto ringraziamento va anche ai miei colleghi con cui ho svolto questa tesi e con cui ho condiviso tutte le avventure trascorse in questi anni.

Grazie ai professori che ci hanno aiutato e consigliato per realizzare al meglio questo progetto e a tutti gli altri docenti che hanno arricchito la mia esperienza formativa universitaria.

Grazie anche ai miei amici, per i momenti di confronto e di divertimento passati assieme sin dall'adolescenza.