ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
SEDE DI CESENA

SECONDA FACOLTÀ DI INGEGNERIA CON SEDE A CESENA
CORSO DI LAUREA IN INGEGNERIA ELETTRONICA, INFORMATICA E TELECOMUNICAZIONI

VIRTUALIZZAZIONE DI RISORSE DI RETE E DI SERVIZI NELLE RETI DI ACCESSO

Elaborato in:
Applicazioni e Tecniche di Telecomunicazioni

Relatore:
Prof. Walter Cerroni

Presentata da:
Mattia Bartoli

III Sessione

Anno Accademico 2011/2012

Ai miei genitori che mi hanno permesso
di intraprendere questo percorso
e ai miei amici che mi hanno
accompagnato lungo di esso.

Indice

Introduzione

È sotto gli occhi di tutti come il mondo sia una realtà in costante evoluzione, tutto si evolve con una velocità sorprendente, di anno in anno, di mese in mese, di giorno in giorno. A cambiare non sono solo l'ambiente, le cose e le abitudini delle persone ma il modo stesso di pensare di queste ultime che si evolve così come evolve l'ecosistema in cui esse vivono. Ma che cos'è che fa si che il mondo cambi così velocemente, come mai tutto si modifica con il tempo? Come mai tutto non resta così com'è?

Una delle principali linee guida dell'evoluzione del mondo è senz'altro il progresso tecnologico. Gli uomini con la loro intelligenza si sono da sempre impegnati nel cercare soluzioni sempre più ingegnose ai loro problemi permettendo così alla tecnologia di avanzare sempre di più e al mondo di evolvere. Più il tempo passa più la tecnologia avanza perché sempre di più sono le conoscenze su cui gli uomini si possono fondare grazie al lavoro e alla ricerca dei loro predecessori, in tal modo possono concentrarsi su ricerche sempre più approfondite e fare scoperte sempre più sorprendenti.

Negli ultimi secoli e negli ultimi decenni in modo particolare il progresso tecnologico ha assunto una velocità impressionante, dall'Ottocento in poi il mondo si è trasformato in modo sempre più rapido dietro l'onda delle nuove tecnologie.

Se si volgono gli occhi alla storia recente e ci si ferma a pensare alle innovazioni tecnologiche degli ultimi decenni quello che salta subito alla mente è il mondo dell'elettronica e dell'informazione. Esso è sicuramente l'ambito nel quale più di tutti gli altri nei tempi recenti il progresso tecnologico ha avuto modo di apportare le sue innovazioni tanto che quando si parla di tecnologia al giorno d'oggi quasi tutti corrono con la mente al mondo dei computer e dell'elettronica.

Dalla nascita dei primi computer a oggi il mondo dell'informazione ha continuato a evolversi in modo ininterrotto e rapidissimo raggiungendo livelli di innovazione che qualche tempo fa sarebbero stati impensabili. Con la crescita del mercato dell'informatica ha iniziato a sentirsi sempre più viva la necessità di poter collegare anche molti computer tra loro, anche a grandi distanze. Questo ha portato alla nascita delle reti di calcolatori e in modo particolare di internet.

L'evoluzione dell'elettronica ha permesso la realizzazione di hardware sempre più potenti e prestanti, in particolare in termini di risorse di elaborazione e di capacità di memoria, il che ha permesso la nascita di software sempre più complessi in grado di fornire agli utenti sempre più funzionalità a scapito appunto di un enorme aumento nell'impiego delle risorse. Come conseguenza anche la quantità di dati destinata a essere scambiata in rete è aumentata notevolmente, il che ha spinto anche il mondo delle telecomunicazioni ad aggiornarsi costantemente.

Nonostante i servizi che fanno uso della rete internet siano profondamente cambiati, diventando sempre più numerosi e complessi, e così anche le tecnologie utilizzate all'interno della rete (ad esempio la diffusione dell'ADSL prima e delle fibre ottiche poi) non è mai cambiato il paradigma su cui essa si basa, che è rimasto in sostanza invariato dalla sua nascita a oggi. Il modello su cui si basa la rete internet è il TCP/IP che ha avuto il suo punto di forza nella definizione di uno strato di trasporto in grado di offrire un servizio di comunicazione uniforme alle applicazioni al di sopra di uno strato di rete basato sull'instradamento di pacchetti, astraendo dalle tecnologie utilizzate per i collegamenti fisici sottostanti, e ciò ha permesso facilmente l'interconnessione di segmenti di rete eterogenei andando a creare la rete di reti che è internet.

Nel modello TCP/IP però esiste una rigida separazione tra il software e lo strato di trasporto per cui le applicazioni non sono in grado di avere informazioni sullo stato della rete su cui comunicano mentre la rete non sa nulla a riguardo dei dati applicativi che trasporta. Questa rigida separazione, non era mai stata un

problema ma con l'evolversi delle ultime tecnologie, in particolar modo le tecnologie di telecomunicazioni mobili, la cosa è cambiata ed inizia a farsi sentire sempre di più il bisogno di far dialogare le applicazioni con la rete in modo tale da poter configurare automaticamente e su richiesta i servizi di comunicazione offerti da essa a seconda delle necessità del momento.

Per queste ragioni negli ultimi tempi è iniziato lo sviluppo di alcune nuove tecnologie con lo scopo di permettere alla rete di evolvere in modo tale che diventi possibile per le applicazioni dialogare con essa per configurare *on-demand* i servizi di telecomunicazioni dei quali necessitano; una rete con queste caratteristiche viene denominata *Next Generation Network*. Alcune delle tecnologie che sono attualmente in fase di sperimentazione sono: Cognitive *Transport Service* (CTS), *Software Defined Network* (SDN) e Network *Function Virtualisation* (NFV).

In particolare questa tesi tratta della Network Function Virtualisation con particolare riferimento alle reti di accesso, cioè quella parte delle reti di telecomunicazioni dedicata al collegamento tra gli utenti dei servizi di comunicazione e il provider, che spesso viene indicata anche come *last mile*. Lo scopo è mostrare come sfruttando tecnologie di virtualizzazione già presenti sul mercato e ampiamente utilizzate e collaudate in altri ambiti, quale per esempio quello dei server web e del *cloud computing*, sia possibile sostituire alcuni degli apparati hardware normalmente presenti nelle reti con delle semplici macchine virtuali in grado di erogare le stesse funzionalità, con in più il vantaggio di una maggiore flessibilità e la possibilità di far dialogare le applicazioni con i servizi di rete.

Questo scritto è suddiviso in tre capitoli: nel primo viene fatta una trattazione teorica sulle reti di nuova generazione che è lo scenario in cui si inseriscono gli esperimenti condotti durante questo lavoro di tesi e sulle nuove tecnologie adottate all'interno di esse, nel secondo si descrivono gli strumenti che sono stati

utilizzati durante il lavoro mentre nel terzo sono descritti in dettaglio gli esperimenti pratici condotti, i loro scopi e i loro risultati.

Capitolo 1
Scenario applicativo

Il modello su cui si basa la rete internet cioè il TCP/IP è rimasto sostanzialmente invariato dalla sua prima definizione al giorno d'oggi. Il maggiore vantaggio offerto da questo schema è la definizione di uno strato di trasporto unitario che permette di creare una rete unica unendo segmenti di reti anche molto diversi tra di loro e svincolando le applicazioni dai problemi di gestione della rete stessa.

Ma se la separazione tra applicazioni e servizi di rete finora era stata un vantaggio con l'evolversi delle tecnologie essa inizia a diventare un limite. Infatti con l'avvento di nuove tecnologie in particolare quelle legate alle comunicazioni mobili e ai servizi di cloud computing si inizia a far sentire sempre più forte la necessità di maggiore flessibilità nella configurazione delle risorse di rete. Insomma quello di cui si inizia ad aver bisogno è la necessità di far dialogare le applicazioni con la rete, cosa impossibile con il modello attuale in cui esse sono rigorosamente separate, in modo tale che esse possano interagire per configurare servizi di comunicazione personalizzati e modificabili in tempo reale.

Per queste ragioni è stato definito il concetto di Next Generation Network, cioè una rete di nuova generazione all'interno della quale le applicazioni possono dialogare con la rete per configurare in tempo reale, a seconda delle loro esigenze del momento e delle richieste degli utenti, dei servizi di telecomunicazioni personalizzati. Sono state inoltre presentate alcune nuove tecnologie quali le Software Defined Network e la Network Function Virtualisation con lo scopo di permettere il passaggio dal modello di rete tradizionale a quello di nuova generazione in una maniera più facile e proficua.

In modo particolare questo capitolo ha lo scopo di descrivere in modo dettagliato che cosa si intende per Next Generation Network e quali sono i vantaggi di questo modello rispetto a quello tradizionale, e di presentare alcune delle tecnologie che potranno essere utilizzate all'interno della rete di nuova generazione.

1.1 Next Generation Network

1.1.1 Cosa si intende per Next Generation Network

Con il termine Next Generation Network [1] (NGN) cioè "rete di prossima generazione" si intende secondo la definizione data dall'ITU [2] (International Telecommunication Unit) una rete basata sulla commutazione di pacchetto in grado di fornire servizi, inclusi quelli di comunicazione, e in grado di fare uso di molteplici tecnologie a larga banda con QoS (*Quality of Service*), nella quale le funzionalità legate alla fornitura dei servizi sono indipendenti dalle tecnologie di trasporto utilizzate. Questa rete deve inoltre garantire un accesso illimitato agli utenti e ai fornitori di servizi e deve supportare una mobilità generalizzata consentendo la fornitura dei servizi di telecomunicazioni agli utenti in maniera coerente indipendentemente dalla loro posizione.

1.1.2 I limiti del modello TCP/IP

Il paradigma su cui al momento attuale si basa la rete internet è quello costituito dal TCP/IP che ha avuto il suo punto di forza nella capacità di offrire un servizio di trasporto unificato che permette di far comunicare tra di loro segmenti di reti basati su tecnologie differenti e che si trova al di sopra di uno stato di rete incentrato sull'instradamento di pacchetti.

Il modello TCP/IP sebbene abbia costituito fino ad adesso una solida base per la rete internet a causa della sua staticità sta iniziando a costituire una limitazione allo sviluppo e all'impiego di nuove funzioni di rete, servizi, politiche di gestione o altri elementi che sono necessari per venire incontro alla sempre crescente complessità e dinamicità della rete.

Con i sistemi attuali il lancio di nuovi servizi richiede agli operatori di rete ingenti investimenti in termini di risorse monetarie ma anche in termini di tempo di sviluppo, il che molto spesso costituisce un ostacolo all'evolversi delle nuove tecnologie che invece sono essenziali per fare in modo che la rete rimanga al passo con le nuove necessità del mondo dell'informazione.

Gli operatori di telecomunicazioni sentono inoltre la necessità di poter automatizzare la gestione e la configurazione degli apparati e delle funzionalità di rete in modo da limitare i costi e poter beneficiare di un utilizzo più razionale e ottimizzato delle risorse, cosa difficilmente attuabile mantenendo il modello attuale.

La staticità del modello TCP/IP e la struttura della rete attuale che prevede che la maggior parte dei servizi siano dislocati in nodi intermedi della rete, separati dagli end-point, inoltre contrastano fortemente con il fatto che gli utenti, spinti dal progredire delle nuove tecnologie nel mondo dell'informazione, richiedono sempre di più servizi di rete personalizzati che molto spesso necessitano di riconfigurare le risorse e i servizi di telecomunicazioni in tempo reale, cosa al momento difficilmente attuabile dato che richiederebbe di intervenire direttamente su tutti i nodi di rete interessati.

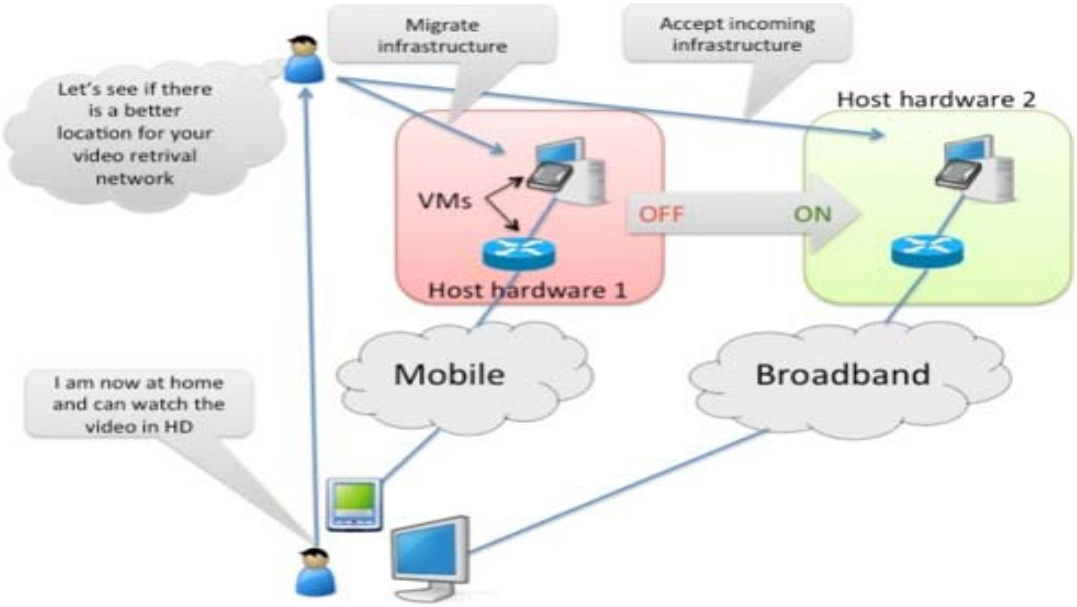

Figura 1.1: *Riconfigurazione automatica della rete in real-time a seconda delle esigenze dell'utente [1]*

1.1.3 Nuove tecnologie all'interno delle reti di prossima Generazione

Alcune tecnologie che si stanno sviluppando in questi ultimi tempi quali le Software Defined Network (SDN) e la Network Function Virtualisation (NFV) possono essere alcuni elementi chiave per la transizione verso modelli di rete che vengano incontro alle necessità del mondo dell'informazione.

L'ambito in cui queste nuove tecnologie avranno un impatto maggiore sarà sicuramente quello delle reti di accesso, in cui sarà possibile sfruttare per questi nuovi scopi tecnologie già esistenti e ampiamente utilizzate in altri ambiti del mondo dell'informazione quali in modo particolare i sistemi di virtualizzazione, che in questi ultimi anni con l'avvento del cloud computing hanno assunto un'importanza rilevante e hanno avuto modo di mostrare tutti i loro vantaggi.

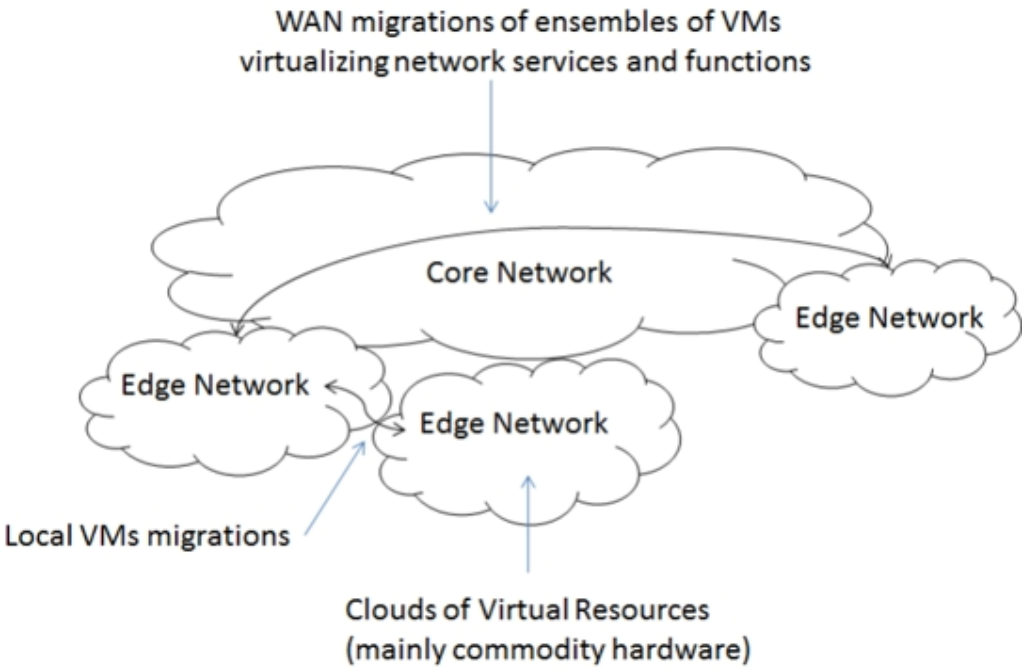

Figura 1.2: *Applicazione degli strumenti di virtualizzazione alle reti di accesso [1]*

Nei prossimi anni si assisterà a una profonda trasformazione delle reti di accesso in un ambiente software distribuito fatto di risorse virtuali in cui alla parte hardware della rete verrà lascito solo il compito di realizzare le interconnessioni tra le varie risorse. In pratica gli apparati di rete saranno sostituiti da componenti hardware standard ed economici, cioè dei semplici server su cui saranno in esecuzione delle macchine virtuali e le relative interconnessioni, che dovranno essere potenti a sufficienza da garantire la fornitura dei servizi di rete virtualizzati.

Una delle caratteristiche più interessanti di questo modello è la possibilità di istanziare, modificare, spostare e gestire in maniera dinamica e automatizzata gruppi di macchine virtuali che realizzano i servizi di rete. Questo permetterà agli operatori di gestire in modo molto flessibile la propria rete e di offrire servizi in grado di adattarsi in tempo reale alle proprie esigenze e a quelle degli utenti.

1.2 Cognitive Transport Service

1.2.1 La necessità di un nuovo Cognitive Transport Service

Il modello su cui si basa la rete internet al momento attuale [3], cioè il modello TCP/IP si basa su di un servizio di trasporto unificato che permette di unire tra di loro reti diverse ed eterogenee al di sopra di uno strato di rete basato sull'instradamento di pacchetti. In un simile modello lo strato di trasporto è completamente trasparente e indipendente rispetto al livello applicativo, quindi le applicazioni non hanno controllo sulla rete e la rete non possiede informazioni a riguardo del tipo di dati che deve trasportare.

Se queste caratteristiche hanno garantito il successo del modello TCP/IP fino ad oggi, negli ultimi tempi stanno emergendo tutte le limitazioni che questo approccio comporta. Negli ultimi anni si è assistito alla moltiplicazione dei servizi di rete offerti e dei protocolli da essi utilizzati oltre ad un'enorme evoluzione degli apparati di telecomunicazioni, e di conseguenza si è cominciata a fare sentire la necessità da parte delle applicazioni di poter interagire con la rete stessa per poterla configurare dinamicamente in modo personalizzato a seconda delle esigenze dell'utente.

La rete internet nel futuro dovrà essere in grado non solo di fornire un semplice servizio di trasporto di dati ma dovrà essere in grado di erogare numerose funzionalità di comunicazione integrate configurabili in tempo reale dalle applicazioni per venire incontro alle necessità degli utenti. Perché ciò accada occorre che le applicazioni siano in grado di ottenere informazioni sulla configurazione di rete (devono essere insomma *network-awared*) mentre la rete deve essere in grado di ricevere disposizioni dalle applicazioni e riconfigurarsi attivamente per conformarsi a esse.

Tutto questo è però impossibile continuando a seguire il modello TCP/IP nel quale lo strato di trasporto e quello di applicazione sono rigorosamente separati.

Occorre allora inserire un nuovo strato che sia intermedio tra quelli sopra nominati e che sia responsabile della fornitura di servizi di rete personalizzati a seconda delle necessità delle applicazioni, questo nuovo strato può essere indicato con il termine Cognitive Transport Service [4] (CTS). L'introduzione di questo nuovo strato permette la realizzazione di applicazioni di nuova generazione capaci di configurare attivamente i servizi di rete di cui necessitano per fornire servizi di comunicazione on-demand.

1.2.2 Come realizzare il Cognitive Transport Service

A differenza di come si potrebbe pensare l'introduzione del CTS non implica necessariamente degli sconvolgimenti radicali alla struttura delle reti di telecomunicazioni attuali, infatti è possibile implementare questo nuovo strato semplicemente sfruttando protocolli già esistenti e ampiamente utilizzati nelle reti attuali per altri scopi ma che possono facilmente essere adattati per ottenere quando sopra auspicato.

E'opportuno che il CTS sia realizzato disaccoppiando le problematiche relative al protocollo di segnalazione che definisce la sintassi usata per la negoziazione dei servizi di comunicazione dal protocollo descrittivo che fornisce la semantica necessaria per descrivere i servizi. Di conseguenza è necessario dividere il CTS in due sotto-strati:

- **Strato di sessione**: specifica il flusso di messaggi necessario per iniziare, modificare e terminare una sessione di comunicazione.
- **Strato di presentazione**: è responsabile della descrizione dei servizi, il suo scopo è quello di rendere le applicazioni in grado di comunicare con la rete in modo da configurare i servizi di comunicazione da essa offerti a seconda delle necessità dell'utente.

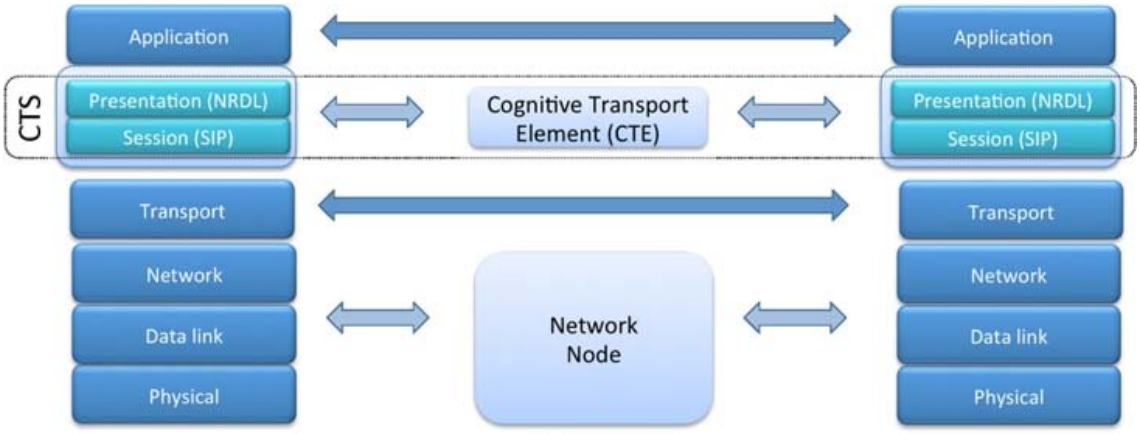

Figura 1.3: *Introduzione del nuovo Cognitive Transport Service [3]*

I protocolli che sono stati proposti per realizzare il CTS sono:

- **SIP** (Session Initiating Protocol): è un protocollo di segnalazione ampiamente utilizzato nell'ambito delle applicazioni VoIP dove è responsabile dell'apertura, della modifica e della conclusione delle chiamate, esso non fornisce direttamente dei servizi di comunicazione ma piuttosto delle primitive per implementare dei servizi.
- **NRDL** (Network Resource Description Language): è un linguaggio descrittivo in grado rappresentare servizi di rete in modo che siano comprensibili alle applicazioni.

1.2.3 Cognitive Transport Element

Per essere implementato il CTS necessita di un nuovo tipo di sotto-blocco di rete denominato *Cognitive Transport Element* (CTE) che può essere collocato presso l'utente finale o anche in alcuni nodi di rete. Ciascun CTE deve essere dotato di:

- Un **proxy SIP** responsabile della gestione della sessione di comunicazione che è incaricato di inoltrare i pacchetti SIP a destinazione.
- Un **decoder NRDL** in grado di decifrare e interpretare le direttive contenute all'interno dei messaggi, il quale si basa su di un database

che contiene informazioni sulle applicazioni e sulle risorse di rete disponibili.

- Un **modulo network-dependent** in grado di interagire con il piano di controllo della rete o con le interfacce di configurazione dei nodi per poter configurare in modo appropriato le risorse di rete a seconda delle necessità.

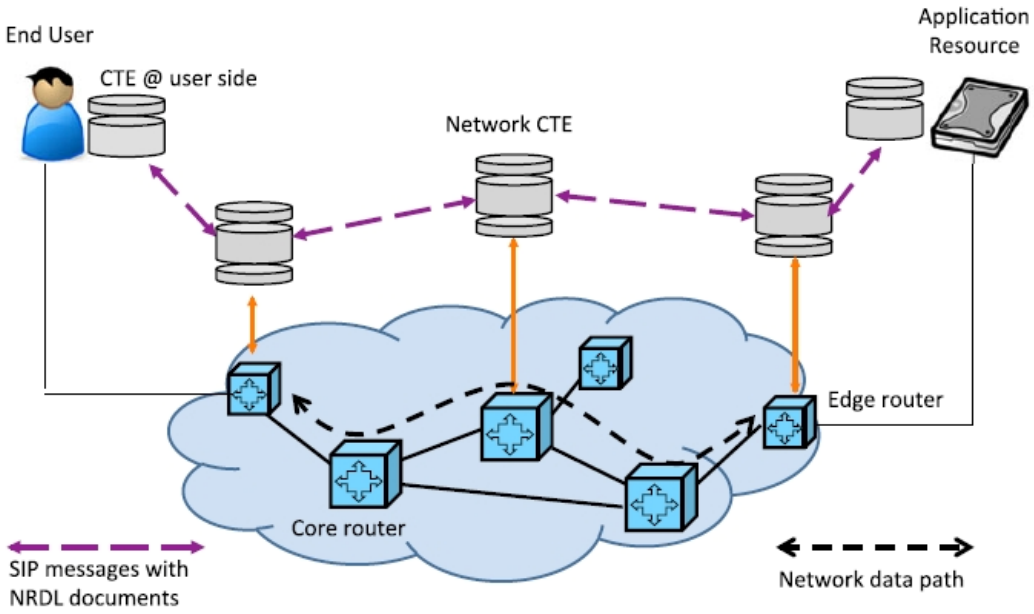

Figura 1.4: *Implementazione del Cognitive Transport Service tramite i Cognitive Transport Element [3]*

Il CTE collocato presso l'utente finale è responsabile della richiesta dei servizi di comunicazione a seconda di quanto richiesto dalle applicazioni, i CTE all'interno della rete invece sono incaricati dell'attivazione dei servizi richiesti. Ogni volta che è necessario attivare una connessione il CTE presso l'utente finale prepara un documento NRDL contenente la configurazione di rete desiderata incapsulato in un pacchetto SIP, e inizia poi una sessione SIP con il CTE ricevitore sfruttando il proxy SIP in esso integrato. Il CTE ricevente conferma l'apertura della sessione e attiva le configurazioni di rete richieste. Ciascun nodo durante il percorso è in grado di leggere e modificare il messaggio a seconda delle necessità.

1.3 Software Defined Network

1.3.1 Cosa si intende per Software Defined Network

Software Defined Network [5] (SDN) è un'architettura per la realizzazione di reti di telecomunicazioni nella quale il piano di controllo della rete (*control plane*) e quello di trasporto dei dati (*data plane*) sono separati logicamente.

Nell'architettura SDN il piano di controllo della rete viene realizzato tramite risorse software e viene astratto dall'hardware sottostante che mantiene unicamente la funzione di inoltrare fisicamente i dati.

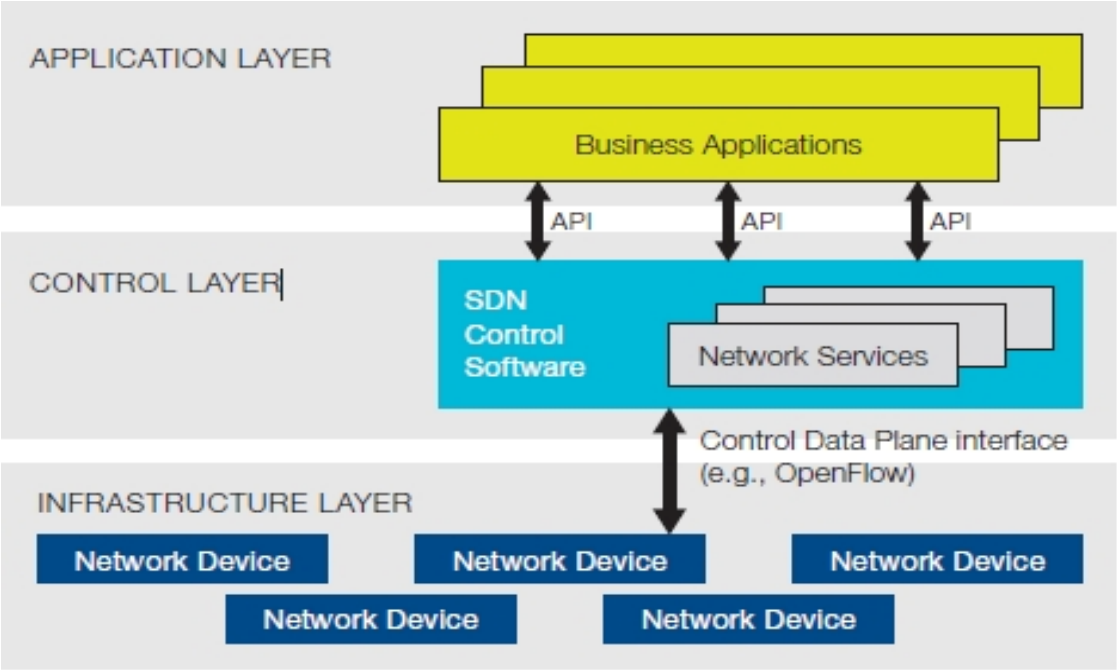

Figura 1.5: *Disaccoppiamento del piano di controllo dal piano di trasporto nelle Software Defined Network [5]*

1.3.2 I vantaggi dell'architettura SDN

Il passaggio da una struttura di rete tradizionale a una di tipo SDN consente di beneficiare di numerosi vantaggi quali:

- Controllo e gestione centralizzata degli apparati di rete anche provenienti da produttori diversi.

- Possibilità di automatizzare molte delle operazioni di gestione sfruttando unicamente funzionalità software essendo il piano di controllo svincolato dalla struttura fisica sottostante della rete.
- Maggiore apertura all'innovazione grazie alla possibilità di aggiungere sempre nuove funzionalità e servizi unicamente aggiornando il software e senza dover intervenire sugli apparati di rete fisici per riconfigurarli.
- Possibilità di programmare le funzionalità di rete unicamente via software sfruttando linguaggi di programmazione di uso comune nel mondo dell'informatica, eliminando la necessità di firmware proprietari diversi da produttore a produttore.
- Maggiore affidabilità e sicurezza della rete grazie alla possibilità di gestire la rete in modo centralizzato; inoltre diminuzione degli errori di configurazione grazie alla gestione completamente automatizzata.
- Possibilità di gestire la rete in modo granulare applicando politiche di gestione variabili a seconda dell'utente, della sessione o del servizio utilizzato.
- Miglioramento dell'esperienza utente grazie alla possibilità di offrire servizi di rete personalizzati adattando le funzionalità in run-time al tipo di servizio richiesto.

1.3.3 La necessità di un nuovo modello di rete

Il passaggio a reti basate sull'architettura SDN si rende necessario dato che l'architettura di rete tradizionale, di tipo gerarchico, normalmente costituita da un insieme di switch ethernet distribuiti su di una struttura ad albero, risulta molto poco flessibile e scalabile e mal si adatta alla necessità sempre più sentita di poter offrire servizi personalizzati agli utenti per i quali occorre la possibilità di modificare a run-time, in modo immediato, le configurazioni della rete a seconda delle necessità.

I principali cambiamenti nel mondo dell'informazione che spingono verso la necessità di reti di tipo SDN sono:

- Cambiamento dei modi in cui gli host comunicano tra di loro: in passato il modello predominante all'interno delle reti di calcolatori è sempre stato quello *client-server* in cui un host server fornisce dei servizi a un host client, mentre ora con le moderne applicazioni si sta facendo largo un modello di comunicazione di tipo *peer-to-peer* in cui gli host sono tutti sullo stesso piano e alternativamente, a seconda della necessità del momento, svolgono la funzione di fornitore o di fruitore dei servizi.
- Diffusione dei dispositivi mobili: l'avvento di un sempre maggiore numero di dispositivi mobili quali smartphone e tablet ha comportato da una parte un aumento spropositato dal numero di connessioni di rete necessarie dall'altra la necessità di riconfigurare i servizi di comunicazione per adattarsi alle esigenze di mobilità degli utenti.
- L'avvento del cloud computing: negli ultimi anni sempre più utenti, sia privati che aziende, hanno iniziato a fare uso sistematico dei servizi cloud, e ciò ha comportato una crescita spropositata di questi servizi, i quali per la loro stessa natura necessitano di connessioni di rete che siano il più possibile dinamiche e riconfigurabili a seconda della necessità del momento.
- Aumento della richiesta di banda: la sempre maggiore necessità di servizi di comunicazione comporta un aumento notevole della quantità di dati scambiati in rete il che ha come conseguenza un inevitabile aumento della richiesta di banda da parte degli utenti, ciò si rispecchia in un aumento della complessità delle reti di telecomunicazioni che diventano difficili da gestire con i metodi tradizionali.

1.3.4 I limiti del modello di rete tradizionale

Venire incontro alle nuove necessità del mercato è materialmente impossibile mantenendo la struttura di rete tradizionale. Fino ad ora, per contenere le spese, i gestori delle infrastrutture di telecomunicazioni hanno fatto tutto il possibile per sfruttare al massimo le risorse delle loro reti spesso ricorrendo a strumenti di

gestione operanti a livello dei singoli device o direttamente tramite processi manuali. Ma al momento attuale in cui il numero di connessioni sta crescendo esponenzialmente così come la necessità di servizi personalizzati questo approccio sta diventando sempre più impraticabile.

I principali punti deboli delle architetture di rete tradizionali possono essere sintetizzati in:

- Complessità che comporta una mancanza di dinamicità: al momento attuale le reti di telecomunicazioni si basano su di un larghissimo numero di protocolli differenti. Negli ultimi decenni, spinte dalla necessità di nuove funzionalità, le aziende hanno definito un gran numero di protocolli, i quali però sono spesso stati definiti in maniera isolata con lo scopo di risolvere determinati problemi e senza un lavoro di astrazione che possa permettere un'unificazione tra di essi. Tutto ciò ha portato a una grande complessità la quale è forse una delle principali limitazioni delle reti attuali, infatti questa complessità viene riflessa in una forte staticità delle reti che poco si adatta alla crescente dinamicità del mondo dei servizi di informazione, sostenuta tra l'altro dall'avvento dei sistemi di virtualizzazione.
- Politiche di gestione inconsistenti: la complessità delle reti di telecomunicazioni attuali comporta l'impossibilità di applicare delle politiche di gestione che siano universali.
- Limitata scalabilità: il forte aumento della domanda di servizi di telecomunicazioni comporta un necessario aumento delle dimensioni delle reti che con il sistema attuale richiede l'aggiunta di sempre nuovi dispositivi, la cui configurazione e gestione diventa sempre più difficoltosa.
- Dipendenza dai fornitori: i produttori di hardware per le telecomunicazioni, incalzati dalla sempre maggiore richiesta di dispositivi, sempre più moderni e performanti sviluppano in continuazione nuove tecnologie, ma molto spesso ciò avviene senza

che sia definito alcuno standard e questo comporta che dispositivi provenienti da fornitori diversi non siano compatibili tra di loro, costringendo i gestori a una dipendenza da uno o pochi costruttori di hardware.

Nell'architettura SDN grazie al disaccoppiamento tra piano di controllo della rete e piano di trasporto dei dati, diventa possibile trattare la rete come un'unica entità logica che può essere facilmente gestita nel suo insieme. Il piano di controllo della rete è centralizzato in controller software che mantengono una visione globale della rete che perciò appare alle applicazioni come un'entità unitaria. Con l'architettura SDN i gestori possono avere il controllo della loro intera rete da un singolo punto, semplificando la struttura stessa della rete e tutte le operazioni di gestione. Si assiste anche a una notevole semplificazione degli apparati stessi i quali non possiedono più funzioni di gestione internamente ma si limitano a ricevere istruzioni dai controller software. Con questo sistema diventa inoltre possibile automatizzare molte delle operazioni di gestione della rete semplicemente programmando in modo adeguato il software.

1.4 Network Function Virtualisation

1.4.1 Cosa si intende per Network Function Virtualisation

Con il termine Network Function Virtualisation [6] (NFV) si intende la realizzazione tramite risorse software di virtualizzazione di funzionalità di rete normalmente svolte da apparati di telecomunicazione fisici.

Il concetto di NFV non deve essere confuso con quello di Software Defined Network (SDN). Questi due concetti sono strettamente legati tra di loro e possono comportare particolari vantaggi se applicati contemporaneamente ma sono di per se indipendenti: si può applicare la virtualizzazione delle risorse di rete pur non disponendo di reti definite via software e vice versa.

1.4.2 Lo scopo della Network Function Virtualisation

Al momento attuale le reti degli operatori di telecomunicazioni sono costituite da una grande varietà di apparati hardware di diverso genere con i quali sono realizzate le varie funzioni che la rete deve offrire per fornire i suoi servizi. Questi apparati di telecomunicazioni possono essere realizzati da molti produttori diversi e molto spesso sono costruiti seguendo standard proprietari dei fornitori piuttosto che specifiche universali, questo comporta che essi possono essere spesso profondamente differenti come funzionamento e soprattutto come modalità di gestione.

Secondo la logica attuale quando un operatore vuole lanciare un nuovo servizio di rete deve acquistare nuove risorse hardware appositamente progettate. Questo significa innanzitutto che dovranno essere sostenuti sicuramente cospicui investimenti per la progettazione e l'acquisto degli apparati necessari, i quali molto spesso necessitano di essere disegnati e costruiti ad hoc per il servizio a cui sono destinati. La necessità di disporre di componenti hardware appositamente progettati comporta numerosi altri problemi oltre sicuramente ai costi sostenuti per ricerca e sviluppo, infatti ci si trova a dover fronteggiare questioni relative al tempo necessario per lo sviluppo dei nuovi apparati prima che essi possano essere finalmente disponibili sul mercato (*time to market*) e alla difficoltà di reperire la manodopera altamente specializzata che risulta necessaria per la progettazione e la costruzione di apparati così altamente tecnologici. L'attivazione di nuovi servizi richiede inoltre la ricerca di nuovi spazi per alloggiare le risorse hardware aggiuntive necessarie e un inevitabile aumento dei costi per l'approvvigionamento dell'energia elettrica per sopperire all'alimentazione di tali apparati. Bisogna inoltre aggiungere che una volta progettati, costruiti e attivati i nuovi apparati occorrerà anche provvedere a una adeguata attività di manutenzione, la quale potrebbe comportare costi non indifferenti sia per quanto riguarda il costo della manodopera necessaria a questo

scopo sia per quanto riguarda eventuali ricambi, che nel caso di apparati non standard possono risultare relativamente costosi.

A peggiorare la situazione sopra descritta si aggiunge il fatto che le tecnologie sono in costante evoluzione, e che specie negli ultimi tempi il ciclo di vita degli apparati hardware si sta sempre di più riducendo a causa del ritmo sempre più incalzante del mercato dell'information tecnology. Infatti da una parte con il diffondersi delle tecnologie gli utenti richiedono servizi sempre più aggiornati ai gestori, dall'altra i produttori di hardware mettono sul mercato apparati sempre più nuovi e performanti. I gestori sono così costretti a un continuo aggiornamento delle risorse hardware a loro disposizione il che comporta una sempre maggiore accelerazione dei cicli di sviluppo e messa sul mercato che implica per essi costi sempre più alti tanto da arrivare a volte ad annullare qualsiasi beneficio per gli investitori. Tutti questi problemi rischiano di inibire lo sviluppo di nuovi servizi di telecomunicazioni e di porre dei freni al progresso delle nuove tecnologie.

Lo scopo della NFV è svincolare la fornitura dei servizi di telecomunicazioni dai problemi legati agli apparati hardware sfruttando le tecnologie di virtualizzazione che il mercato dell'informazione offre e che già sono ampiamente sfruttate in altri ambiti quali ad esempio nel campo dei server web. Con la NFV tutti gli apparati di telecomunicazione fisici adibiti alle varie funzioni di rete possono essere sostituiti da macchine virtuali che vengono messe in esecuzione su server identici a quelli comunemente utilizzati nelle industrie del mondo dell'informazione. Grazie alla NFV è possibile ridurre la necessità di apparecchiature hardware a pochi dispositivi di tipo standard quali server, switch e dispositivi di storage, cancellando il bisogno di ricorrere a componenti hardware appositamente progettati. Con la NFV è inoltre possibile offrire servizi sempre al passo con i tempi senza la necessità di costosi investimenti, infatti è possibile adeguarsi alle nuove tecnologie semplicemente eseguendo degli aggiornamenti software senza bisogno di sostituire l'hardware.

1.4.3 I vantaggi della Network Function Virtualisation

I maggiori benefici offerti dalla NFV possono essere sintetizzati in:

- Riduzione dei costi relativi all'acquisto degli apparati hardware grazie sia al minor numero di apparati necessari sia alla possibilità di utilizzare hardware di tipo standard al posto di dispositivi progettati e costruiti ad hoc beneficiando quindi di economie di scala.
- Riduzione dei consumi energetici che permette da una parte un vantaggio economico grazie alle minori spese per l'approvvigionamento energetico, dall'altra una maggiore sostenibilità in termini di impatto ambientale.
- Semplificazione dell'attività di progettazione legata all'utilizzo di hardware e software di tipo standard con conseguente diminuzione dei costi degli investimenti e contrazione dei tempi di sviluppo delle nuove tecnologie.
- Possibilità di offrire servizi sempre al passo con i tempi senza avere necessità di sostituire i dispositivi hardware, semplicemente tramite aggiornamenti software.
- Minori costi di manutenzione sia in termini di minor costi per ricambi grazie all'utilizzo di hardware standard sia per la ridotta necessità di manodopera specializzata.
- Maggior flessibilità in quanto è possibile rimodulare facilmente l'offerta in base ad un eventuale aumento o calo della domanda.
- Possibilità di offrire servizi personalizzati in base a vari fattori quali ad esempio la dislocazione geografica degli utenti e le loro necessità.
- Ottimizzazione dei servizi grazie alla possibilità di modificare le configurazioni della rete in real-time.
- Svincolamento dagli standard proprietari dei fornitori di hardware quindi possibilità di integrazione tra hardware provenienti da costruttori diversi.

- Possibilità di utilizzare software open source, facilitando quindi la ricerca e lo sviluppo di nuove applicazioni anche grazie al coinvolgimento di enti quali Università e società emergenti.

1.4.4 I presupposti affinché la NFV raggiunga i suoi scopi

Per poter sfruttare appieno i vantaggi apportati dalla NFV occorre che siano verificati alcuni presupposti:

- Compatibilità tra i dispositivi hardware provenienti da fornitori diversi.
- Portabilità tra software di virtualizzazione diversi.
- Possibilità di fare convivere le funzioni di rete virtualizzate con gli apparati di rete standard in modo che sia possibile effettuare in modo graduale la transizione dal vecchio al nuovo sistema senza necessità di cambiamenti immediati e radicali e senza che ci si verifichino interruzioni dei servizi di telecomunicazioni.
- Automatizzazione della gestione delle funzionalità di rete virtualizzate in modo da poterle adattare in modo flessibile a seconda delle necessità.
- Adeguata protezione del software per quanto riguarda attacchi esterni ed errori di configurazione.
- Buona stabilità e disponibilità di affidabili procedure di recupero nel caso di errori del software o guasti dell'hardware.

Bisogna inoltre tener conto che il passaggio da hardware specifici a strumenti di virtualizzazione in esecuzione su server di tipo standard in genere comporta un degrado in termini di prestazione che non può essere trascurato. E' necessario mettere in atto tutti gli accorgimenti possibili in modo da ridurre al minimo il divario prestazionale tra le funzioni di rete virtualizzate e quelle realizzate con apparati fisici appositi.

La NFV può sfruttare le moderne tecnologie che sono state già sviluppate e sono già largamente utilizzate in altri ambiti quali ad esempio il cloud computing. Tecnologie provenienti dall'ambito del cloud possono essere la virtualizzazione di hardware tramite l'utilizzo di hypervisor software e la possibilità di creare switch ethernet virtuali per connettere tra di loro le macchine virtuali e le interfacce di rete fisiche. Inoltre il mondo del cloud fornisce meccanismi di automatizzazione della gestione che permettono di creare nuove istanze software quando si presenta la necessità e di allocare e assegnare alle varie istanze le risorse hardware, come memoria RAM, risorse di calcolo e interfacce di rete a seconda della necessità in modo completamente automatico.

1.4.5 Principali ambiti di applicazione della NFV

Alcune esempi di funzionalità di rete che possono essere virtualizzate sono:

- Instradamento di pacchetti: router e NAT.
- Funzionalità di bridging e switching.
- Servizi di storage: NAS.
- Strumenti di analisi: deep packet inspection (DPI), analizzatori di traffico.
- Strumenti per test e diagnosi.
- Funzioni di sicurezza: firewall, server di autenticazione.
- Distribuzione di contenuti: server web e server video.

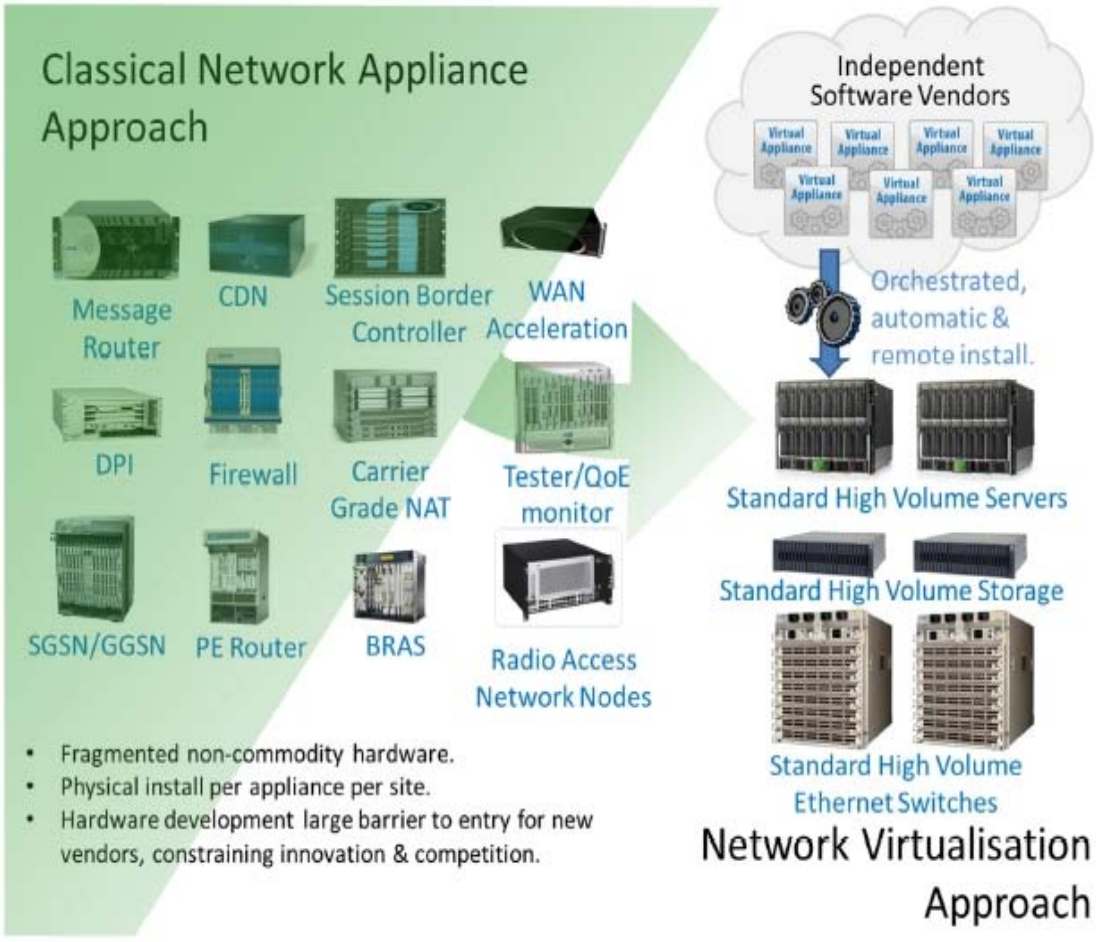

Figura 1.6: *Sostituzione degli apparati di rete fisici con risorse virtuali [6]*

L'ambito in cui si prevede che la NFV potrebbe apportare maggiori benefici è quello delle reti di accesso.

Capitolo 2
Risorse e tecnologie utilizzate

Gli esperimenti condotti durante questo lavoro di tesi riguardano tutti la virtualizzazione dei servizi di comunicazione all'interno di una rete di accesso. In modo particolare le risorse di rete virtualizzate sono state realizzate durante il corso di essi sfruttando delle macchine virtuali create tramite l'utilizzo di software disponibile sul mercato e hardware di tipo standard.

Questo capitolo ha lo scopo di presentare in dettaglio tutti gli strumenti software, con riferimento in particolare alle macchine virtuali e alle funzionalità di virtualizzazione, e le risorse hardware che sono state utilizzate durante il corso degli esperimenti condotti, i quali invece saranno descritti in dettaglio nel terzo capitolo.

2.1 Il sistema operativo CentOS

CentOS [7] (Community Enterprise Operating System) è una distribuzione di Linux ideata per fornire un sistema operativo di classe *enterprise*, cioè un sistema adatto a usi professionali. Essendo quindi un sistema operativo professionale CentOS si adatta sia all'utilizzo su computer standard utilizzati in ambito professionale sia all'utilizzo su veri e propri server.

CentOS così come le altre distribuzioni di Linux include un *kernel* GNU/Linux oltre ad un insieme di applicazioni software sotto forma di pacchetti, molto spesso rilasciati sotto la licenza GNU General Public License così come il sistema operativo stesso. Il gestore di pacchetti utilizzato in CentOS è YUM (Yellow Dog Updater Modified) che utilizza i pacchetti in formato .rpm.

Il sistema operativo è dotato normalmente dell'interfaccia grafica GNOME (GNU Network Object Model Enviroment) ed è gestibile anche tramite riga di comando attraverso il terminale, i comandi supportati sono in generale quelli standard di Linux. E' in genere anche possibile effettuare l'accesso tramite terminale remoto utilizzando il protocollo SSH (Secure Shell).

CentOS deriva da RedHat Enterprise Linux, una distribuzione di Linux molto diffusa in ambito professionale che ha però lo svantaggio di essere resa disponibile solo a pagamento; nonostante ciò così come per le altre distribuzioni di Linux il codice sorgente di RedHat è disponibile liberamente sotto la licenza GNU. CentOS nasce proprio dal codice sorgente di RedHat con lo scopo di offrire un sistema operativo professionale completamente compatibile con la distribuzione da cui deriva ma a differenza di essa completamente gratuito.

Come in tutti i sistemi Linux e più in genere in tutti i sistemi basati su Unix, in CentOS esiste il concetto di utente amministratore o *superuser*. L'utente amministratore è identificato in CentOS, e in genere in Linux, con il nome di *root*. L'utente root dispone del massimo controllo sul sistema e può effettuare operazioni in genere non permesse ai normali utenti ad esempio l'installazione di taluni pacchetti software e la modifica delle configurazioni di sistema (ad esempio la configurazione delle interfacce di rete, della tabella di routing o del firewall). E' in genere sconsigliabile per motivi di sicurezza eseguire il sistema operativo come root nelle normali sessioni di lavoro e limitarsi a utilizzarlo solamente per le operazioni di gestione. Certi utenti possono eseguire da terminale comandi normalmente riservati solo all'utente root utilizzando il comando *sudo* (super user do), ma solo se esplicitamente autorizzati dall'amministratore di sistema tramite la modifica dell'apposito file *sudoers*.

L'ultima versione disponibile di CentOS è al momento la release 6.3. Sono disponibili versioni di CentOS sia per i sistemi più recenti a 64 bit (x86-64) sia per i più vecchi a 32 bit (i386).

Essendo CentOS un sistema operativo libero e gratuito per poterlo ottenere è sufficiente accedere al sito ufficiale della community e recarsi nella sezione download, qui bisogna selezionare un *mirror* cioè un server da cui scaricare

l’immagine ISO di installazione del sistema operativo. Esistono vari tipi di immagini del sistema operativo che possono essere scaricate tra cui la live cd che serve per avere un sistema avviabile live da cd rom, la versione minimale che costituisce una versione del sistema essenziale per utilizzi in cui si ha a che fare con sistemi a bassissime prestazioni, la versione DVD completa con tutti i pacchetti software opzionali che possono essere selezionati o no durante l’installazione, e infine la *net-install* che si tratta di un’immagine che comprende solo i file essenziali per avviare l’installazione e che necessita durante essa di una connessione di rete dalla quale scaricare i pacchetti software da installare. Se si vuole installare CentOS su di un PC o un server e si dispone di una connessione veloce la net-install è sicuramente la scelta più conveniente. Quando si effettua la net-install durante l’installazione viene chiesta l’URL da cui prelevare i pacchetti per l’installazione.

CentOS è il sistema operativo che è installato sui server utilizzati per condurre i lavori di sperimentazioni descritti in questo lavoro di tesi. Inoltre CentOS è stato utilizzato anche come sistema all’interno di molte delle macchine virtuali che sono state create.

2.2 Macchine Virtuali

2.2.1 Cos’è una macchina virtuale

Nel mondo dell’informatica una macchina virtuale è un software che attraverso un sistema di virtualizzazione è in grado di emulare il funzionamento di un determinato sistema hardware, solitamente un computer, e di tutte le sue periferiche. Insomma una macchina virtuale può essere vista come una sorta di contenitore software nel quale può essere eseguito un sistema operativo con le relative applicazioni come se fosse un computer reale. Il sistema operativo e le applicazioni non rivelano alcuna differenza se sono eseguiti su di una macchina virtuale invece che su di una macchina fisica.

Le macchine virtuali per poter funzionare necessitano della presenza di un software di virtualizzazione cioè di un programma che deve essere installato su

di un PC o server fisico il quale costituisce la cosiddetta macchina ospite. Su di uno stesso computer e con lo stesso software di virtualizzazione possono essere create e messe in esecuzione contemporaneamente numerose macchine virtuali che sono indipendenti una dall'altra.

2.2.2 Vantaggi delle macchine virtuali e loro applicazioni

I vantaggi delle macchine virtuali possono essere molteplici:

- Migliore sfruttamento delle risorse: è possibile eseguire su di uno stesso computer molte macchine virtuali per sfruttarne appieno le risorse, e questo è molto utile se si vogliono ridurre costi e consumi.
- Flessibilità: creare una nuova macchina virtuale è praticamente immediato e senza costi aggiuntivi, invece non si può dire la stessa cosa dell'acquisto di un nuovo PC; inoltre è anche molto facile modificare le macchine virtuali cambiando le periferiche a loro disposizioni senza dover fare alcuna modifica all'hardware fisico.
- Isolamento: eventuali problemi occorsi a una macchina virtuale non influiscono in nessun modo sulle altre macchine virtuali e sulla macchina ospite.
- Resistenza ai guasti: è possibile clonare, esportare e importare le macchine virtuali, è sempre possibile quindi fare dei backup da utilizzare in caso di problemi.
- Mobilità: quasi tutti i software di virtualizzazione offrono la possibilità di migrare le macchine virtuali da una macchina ospite a un'altra senza fermarne l'esecuzione, questo può tornare molto utile sia in termini di ottimizzazione, ad esempio spostando delle macchine virtuali da un server sovraccaricato a uno con molte risorse disponibili, sia in termini di facilità di manutenzione, ad esempio nel caso si debba spegnere un server per una riparazione le macchine virtuali che sono su di esso possono essere migrate su di un'altra macchina senza che i servizi da esse forniti siano soggetti a discontinuità.

Le macchine virtuali sono molto diffuse e utilizzate in modo particolare nell'ambito dei server web e del cloud computing ma in virtù dei vantaggi che esse comportano possono prestarsi all'utilizzo in molti altri ambiti come ad esempio la virtualizzazione di servizi di rete.

2.3 Software di virtualizzazione

2.3.1 Cosa sono i software di virtualizzazione

Un software di virtualizzazione, spesso indicato anche come hypervisor, è un'applicazione che permette la creazione e l'esecuzione di macchine virtuali su di una macchina fisica ospite. Esistono numerosi hypervisor, alcuni dei quali sono disponibili sotto licenza open-source come QEMU/KVM che è il software di virtualizzazione integrato nel kernel Linux e VirtualBox di Oracle, mentre altri sono rilasciati sotto licenze di tipo commerciale quali Xen di Citrix e VMWare. Esistono software di virtualizzazione in grado di funzionare su tutti i principali sistemi operativi e in modo particolare su Windows e su varie distribuzioni di Linux. Ogni hypervisor offre una certa scelta di scelta di componenti hardware virtuali che possono essere scelti per configurare le macchine virtuali.

I software di virtualizzazione che sono stati utilizzati durante gli esperimenti condotti in questo lavoro di tesi sono QEMU/KVM e Oracle VirtualBox, entrambi installati su macchine dotate del sistema operativo Linux CentOS.

2.3.2 Un software di virtualizzazione commerciale: Virtualbox

VirtualBox [8] è un software di virtualizzazione commerciale distribuito dalla società Oracle e precedentemente dalla Sun Microsystems sotto licenza open-source, è disponibile gratuitamente per tutti i principali sistemi operativi tra cui Windows, Mac OS e Linux e offre la possibilità di virtualizzare una grande

varietà di hardware e di creare macchine virtuali che possono eseguire molti sistemi operativi diversi.

L'ultima versione di VirtualBox disponibile per Linux è la 4.2 che introduce alcune nuove funzionalità quale la clonazione delle macchine virtuali ma purtroppo possiede alcuni bug, in modo particolare per quanto riguarda la funzionalità di migrazione delle macchine virtuali, quindi è consigliabile utilizzare la versione 4.0 che è più stabile e collaudata. VirtualBox in Linux per essere utilizzato necessita dei permessi di amministratore ed è possibile gestirlo in modalità grafica oppure tramite terminale grazie al comando VBoxManage che permette anche di usufruire di opzioni e funzionalità non disponibili in modalità grafica quale ad esempio la migrazione delle macchine virtuali.

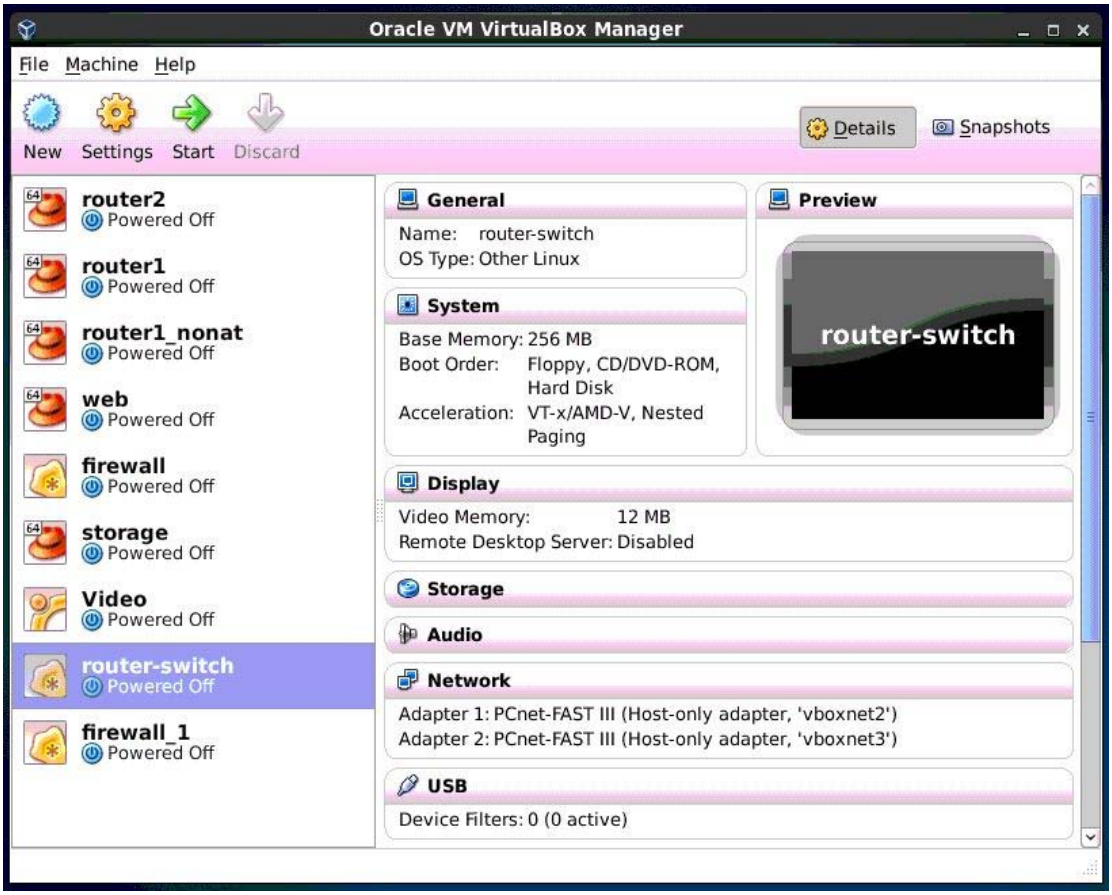

Figura 2.1: *VirtualBox 4.0 per Linux in esecuzione su CentOS*

2.3.3 QEMU/KVM, la virtualizzazione nel kernel Linux

QEMU/KVM [9] è una suite software di virtualizzazione integrata nel kernel Linux, quindi completamente gratuita e open-source. In particolare la suite si compone di QEMU (Quick EMUlator) che è un emulatore ed è responsabile dell'emulazione dell'hardware fisico e di KVM (Kernel-based Virtual Machine) che costituisce l'infrastruttura di virtualizzazione vera e propria. Un altro software ad essi correlato è LibVirt che è un software di gestione che permette di controllare le macchine virtuali in esecuzione sia basate su QEMU/KVM sia su altri hypervisor, ad esempio Xen.

2.4 Migrazione di macchine virtuali

2.4.1 Cos'è la migrazione e qual è il suo scopo

La migrazione [10] o *live migration* di una macchina virtuale è un processo che consiste nel trasferire una macchina virtuale che è in esecuzione su di un certo host fisico su di un secondo host senza che vi sia un'interruzione nel funzionamento della macchina. Affinché si possa eseguire una migrazione di una macchina virtuale occorre disporre sull'host sorgente e sull'host destinazione due macchine virtuali identiche per quanto riguarda la configurazione hardware e anche l'hardware delle macchine fisiche deve essere il più simile possibile altrimenti si ha il rischio che la migrazione fallisca, ad esempio non è in genere possibile migrare macchine virtuali tra un host con processore Intel e uno con processore AMD.

L'obiettivo della live migration è quello di eseguire il trasferimento della macchina virtuale in modo tale che non ci sia alcuna interruzione nel suo funzionamento. In realtà per quanto il trasferimento possa essere veloce, è sempre presente un breve tempo di inattività tra il momento in cui l'esecuzione della macchina virtuale viene interrotta sulla macchina di origine e quello in cui avviene il riavvio sulla seconda, questo tempo viene comunemente indicato come

downtime. I parametri indicativi dell'efficienza di una migrazione sono il downtime e il tempo totale di migrazione cioè il tempo trascorso tra quando viene dato il comando di migrazione e quando la macchina riprende l'esecuzione sul secondo host.

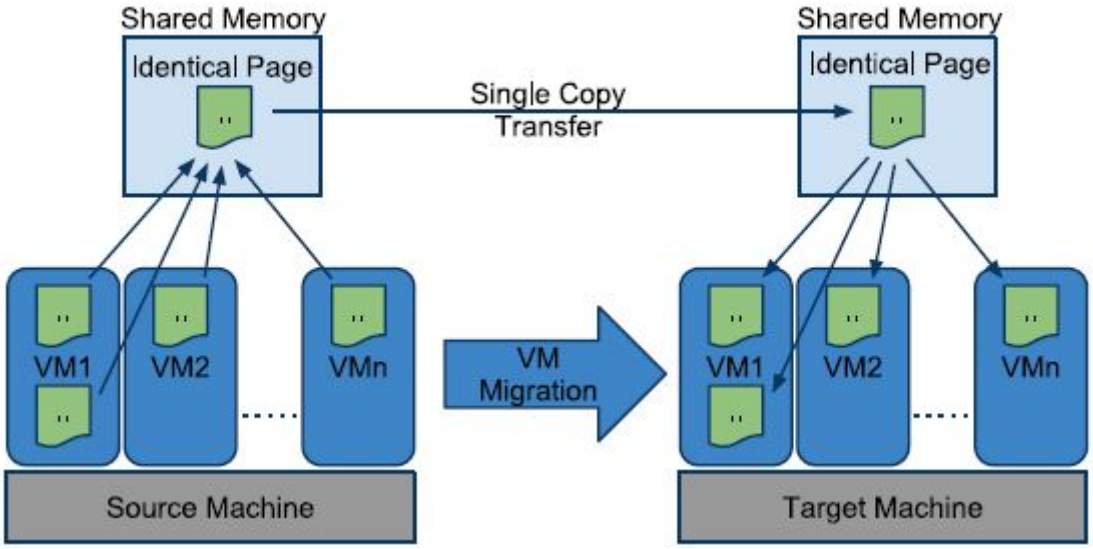

Figura 2.2: *Migrazione di macchine virtuali da una macchina ospite ad un'altra [10]*

2.4.2 Come funziona la migrazione

Esistono due differenti tecniche che possono essere utilizzate per la migrazione di macchine virtuali che sono denominate *pre-copy migration* [11] e *post-copy migration*, le quali differiscono tra di loro per il modo in cui viene copiata la memoria volatile (RAM) della macchina virtuale che viene migrata.

Il metodo più diffuso per la migrazione delle macchine virtuali, che è anche quello che è utilizzato in praticamente tutti i software di migrazione di uso comune, è quello della pre-copy migration. In questo metodo quando viene dato il comando di migrazione, prima di effettuare il trasferimento vero e proprio della macchina in esecuzione, vengono controllate le *dirty-pages* cioè quelle locazioni di memoria RAM che sono state modificate durante il funzionamento della macchina virtuale, ed esse vengono copiate sulla macchina di destinazione. Successivamente alla prima copia si ricontrolla se nel frattempo altre locazioni di memoria sono state modificate e, nel caso, si esegue di nuovo la copia. Si continua poi in questo modo fino a quando la quantità di dati modificati in

memoria diventa sufficientemente piccola da non giustificare più l'operazione o finché non si raggiunge un tempo limite nel caso in cui esso sia stato stabilito. Solo dopo che sono stati trasferiti i dati della memoria l'esecuzione della macchina virtuale viene interrotta sull'host sorgente e ripresa sull'host di destinazione. Il metodo della pre-copy è quello che garantisce una maggiore affidabilità nelle migrazioni ma a causa dei controlli ripetuti sulle dirty-pages è caratterizzato da un downtime e da un tempo totale di migrazione relativamente elevati.

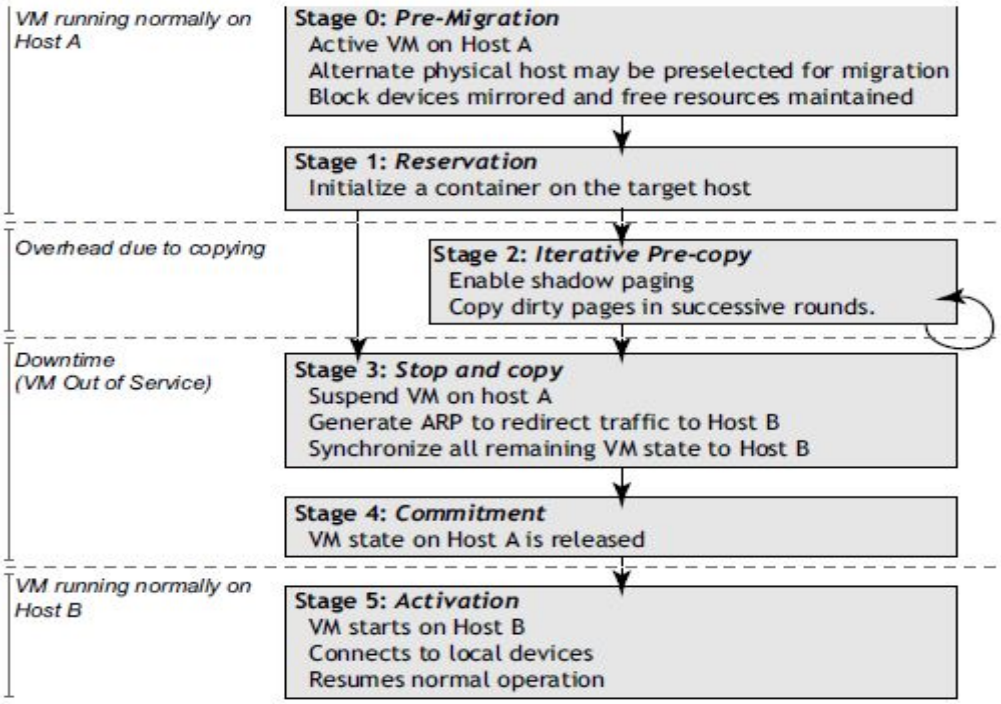

Figura 2.3: *Funzionamento della pre-copy migration [11]*

La post-copy migration è un metodo alternativo, al momento scarsamente utilizzato. In questo metodo quando viene dato il comando di migrazione la macchina virtuale viene immediatamente trasferita copiando sull'host di destinazione solo i dati essenziali, cioè sostanzialmente il contenuto dei registri della CPU, mentre i dati della RAM vengono copiati solo successivamente se necessari. Con questo metodo non essendoci il problema del controllo delle dirty-pages prima della migrazione, il downtime e il tempo totale di migrazione vengono ridotti al minimo. D'altra parte però non venendo copiato il contenuto della RAM, al momento della ripresa dell'esecuzione potrebbero sorgere dei problemi o se non altro la macchina virtuale potrebbe subire un calo di

prestazioni, questo rende il metodo della post-copy meno affidabile. Attualmente non esistono software di uso comune che sfruttino questa procedura, se si escludono alcuni *plug-in* aggiuntivi, disponibili in via ancora sperimentale, per QEMU/KVM.

2.5 Protocollo SIP

2.5.1 Il protocollo SIP

Il protocollo SIP [12] (Session Initiation Protocol) è un protocollo applicativo basato su IP utilizzato principalmente nell'ambito delle applicazioni di telefonia su IP (VoIP). Il SIP svolge la funzione di protocollo di segnalazione responsabile dell'instaurazione, della modifica e della conclusione di una sessione di comunicazione.

Il SIP nasce nel 1996 con lo scopo di unificare due precedenti protocolli, il Session Invitation Protocol (SIP) e il Simple Conference Initiating Protocol (SCIP).

Il SIP è un protocollo di strato 7 (livello applicazione), si basa su linee di testo (come l'http) ed è indipendente dallo strato di trasporto. E' quindi in grado di funzionare con un vasto numero di protocolli di trasporto, sia dotati di controllo dell'affidabilità come il TCP (Trasmission Control Protocol) che no come UDP (User Datagram Protocol), in questo ultimo caso per garantire l'affidabilità viene utilizzato il metodo della ritrasmissione. Il SIP non offre di per se nessun servizio ma solamente primitive elementari utilizzabili per la creazione di servizi.

Il SIP è un protocollo *transaction-oriented*, è basato cioè su sequenze di domanda-risposta, opera tra *end-point* denominati *User Agent* (UA) e come metodo di indirizzamento utilizza il protocollo URI (Uniform Resource Indicator).

2.5.2 Le funzioni del protocollo SIP

Il SIP supporta varie funzioni per stabilire e modificare sessioni di comunicazione multimediali:

- **User location**: determinazione degli end system da utilizzare per la comunicazione.
- **User availability**: determinazione della disponibilità dei partecipanti a impegnarsi in una comunicazione.
- **User capabilities**: determinazione dei media e dei parametri da utilizzare nella sessione.
- **Session setup**: "*ringing*", instaurazione dei parametri di sessione tra il chiamante e il chiamato.
- **Session management**: trasferimento e terminazione delle sessioni, modifica dei parametri della sessione e invoco di servizi.

Il protocollo SIP ha essenzialmente le funzioni di:

- Localizzare gli utenti e acquisire delle loro preferenze.
- Invitare gli utenti a partecipare a una sessione, negoziare le *capability* e trasportare una descrizione della sessione.
- Instaurare le connessioni necessarie alla sessione.
- Gestire eventuali modifiche ai parametri della sessione.
- Rilasciare le parti.
- Cancellare la sessione.

2.5.3 Le entità di una rete SIP

Le entità essenziali di una rete SIP sono:

- **SIP User Agent** (UA): è un end-point che può funzionare da server o da client passando dall'una all'altra funzione in maniera dinamica all'interno della stessa sessione. Quando funge da client genera le

richieste dando inizio alla comunicazione, quando funge da server invece rimane in ascolto e riceve le richieste, soddisfacendole se possibile.

- **Registrar Server**: può essere un server dedicato o può essere collocato presso un Proxy Server, ha lo scopo di raccogliere le registrazioni degli utenti connessi. Quando un utente entra a far parte di un dominio deve inviare la propria registrazione a un Registrar Server.
- **Proxy Server**: è un server intermedio che può, a seconda della necessità, rispondere alle richieste degli utenti oppure inoltrarle agli User Agent o ad altri Proxy Server, nascondendo il destinatario del messaggio. Un Proxy Server può essere *stateful* o *stateless*.

2.5.4 Il messaggio SIP

Un pacchetto SIP è composto da:

- **Start Line**: include il tipo di messaggio, l'indirizzo URI e la versione del protocollo.
- **Header**: specifica le intestazioni del messaggio.
- **Body**: contiene il messaggio SIP, spesso può essere omesso.

I principali tipi di messaggio SIP sono:

- **Invite**: avvia una chiamata.
- **ACK**: conferma la ricezione di un messaggio.
- **BYE**: termina o trasferisce una chiamata.
- **Cancel**: annulla una richiesta di chiamata.
- **Option**: elenca le caratteristiche supportate dal chiamante.
- **Register**: registra uno User Agent presso un Registrar Server.
- **Update**: aggiorna un dialogo non stabilito.

2.5.5 La sessione SIP

Una sessione SIP inizia sempre con l'invio da parte dell'User Agent client di un messaggio INVITE, se il destinatario accetta la connessione risponde con un messaggio di tipo 200 OK, infine il mittente conferma con un ACK; questo sistema viene chiamato *Three-way Handshake*. La conclusione della sessione invece prevede l'invio da parte del chiamante di un messaggio BYE e la conferma da parte del chiamato con un messaggio 200 OK.

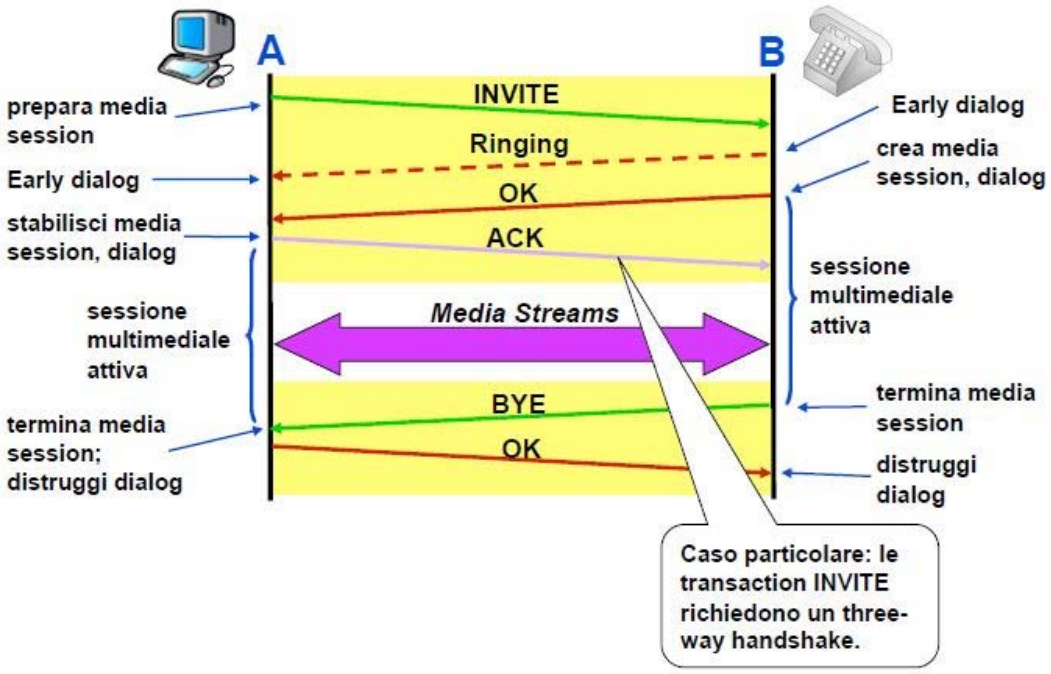

Figura 2.4: *Chiamata diretta tra due User Agent [12]*

Oltre che direttamente la chiamata può avvenire anche tramite l'intermediazione di uno o più proxy server, in tal caso sono possibili due alternative:

- **Redirected Call**: il mittente contatta il proxy server il quale gli indica l'indirizzo del destinatario al quale ridirigere la chiamata.
- **Proxied Call**: il mittente contatta il proxy il quale fa da intermediario reindirizzando i messaggi verso il destinatario.

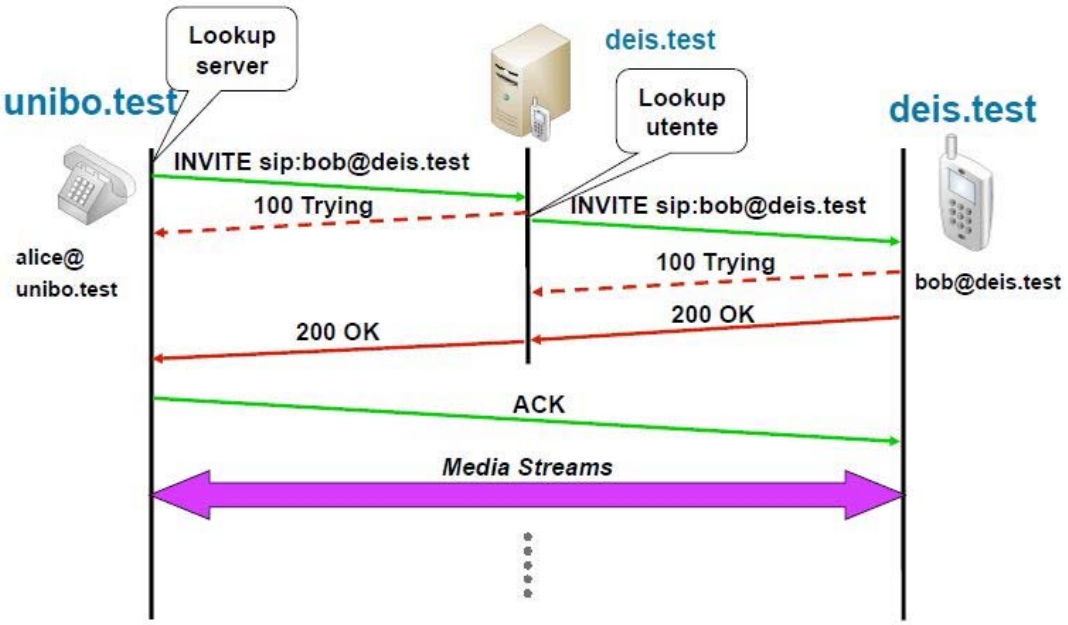

Figura 2.5: *Proxied Call per tramite di un Proxy Server [12]*

2.6 Risorse hardware

2.6.1 Banco di prova per esperimenti con VirtualBox

Per quanto riguarda gli esperimenti condotti con VirtualBox la dotazione hardware utilizzata consiste di due server di tipo *rack* di marca Fujitsu-Siemens denominati VE66 e VE67. In dettaglio l'hardware in dotazione ai due server è il seguente:

- Processore quad-core Intel Xeon E5410 con frequenza di clock pari a 2,33 GHz;
- 4 GB di memoria RAM;
- Hard Disk da 130 GB;
- 3 interfacce di rete ethernet.

Entrambi i server eseguono il sistema operativo Linux CentOS, in particolare su VE66 è installata la versione 5.2 mentre su VE67 è presente la più recente versione 6.2. Delle tre interfacce di rete di ciascun server una è connessa alla rete internet tramite IP pubblico (indirizzi 137.204.107.66 e 137.204.107.67 rispettivamente), una è dedicata al collegamento punto-punto tra i due server per

il quale è riservata la rete IP privata 10.10.10.0/24 (indirizzi 10.10.10.66 e 10.10.10.67 rispettivamente) mentre l'ultima è dedicata al collegamento con uno switch esterno, a questo collegamento è assegnata la rete 172.16.7.0/24 (indirizzi 172.16.7.66 e 172.16.7.67). Entrambi i server supportano l'accesso da remoto tramite terminale SSH e desktop remoto.

I due server in oggetto nel corso degli esperimenti sono stati utilizzati per simulare due ipotetici data center di un service provider tra cui vengono eseguite le migrazioni delle macchine virtuali. Nel corso di alcuni di questi esperimenti è stato necessario ricorrere anche ad una macchina esterna che simulasse un ipotetico utente dei servizi offerti dai data center, questa funzione è stata realizzata grazie ad una macchina virtuale CentOS ospitata su di un server esterno collegato allo switch in precedenza citato, all'interfaccia di questa macchina virtuale è assegnato l'indirizzo 172.16.7.65.

2.6.2 Banco di prova per esperimenti con QEMU/KVM

Gli esperimenti di migrazione con QEMU/KVM sono stati eseguiti invece, dato che per motivi di spazio su disco non era possibile utilizzare i server già utilizzati per VirtualBox, sfruttando una rete composta da un server, sette PC e uno switch, creata precedentemente per un progetto di ricerca sul cloud computing e sugli switch virtuali.

La dotazione hardware del server, di marca DELL, è composta da:

- Processore quad-core Intel Xeon E31230 con frequenza di clock 3,20 GHz;
- 32 GB di RAM;
- 4 schede ethernet.

La dotazione dei sette PC, di marca HP, prevede:

- Processore dual-core Intel Pentium G620 con frequenza di clock 2,60 GHz;
- 4 GB di RAM;
- 1 scheda di rete.

Sia i PC che il server eseguono il sistema operativo CentOS nella versione 6.3, attualmente l'ultima disponibile. Solo il server tramite una delle schede di rete è connesso ad internet mentre i PC sono collegati grazie allo switch tramite una rete privata ed accedono ad internet utilizzando il server come gateway.

Durante gli esperimenti condotti con QEMU/KVM in particolare sono stati utilizzati il server e uno dei computer i quali sono serviti, come precedentemente fatto nel caso di Virtualbox, per simulare i due data center tra cui avvengono le migrazioni delle macchine virtuali.

2.7 Distribuzioni Linux

2.7.1 Che cos'è una distribuzione

Quando si parla di Linux piuttosto che a un sistema operativo unico ci si riferisce a una famiglia di sistemi operativi di tipo Unix-like che sono rilasciati sotto forma di distribuzioni. Tutte le distribuzioni hanno la caratteristica comune di essere basate sul kernel Linux, che è rilasciato sotto licenza open-source.

Grazie alla portabilità e alla flessibilità del kernel Linux, sono state realizzate numerosissime distribuzioni, spesso denominate *distro*, in grado di funzionare su di un gran numero di hardware differenti. Esistono distribuzioni di Linux sia per sistemi basati su architetture di tipo CISC (tipo x86 e x86-64) come PC e Server con processori Intel o AMD, sia per sistemi basati su architetture RISC (ad esempio ARM) quali tablet e smartphone.

Molte distribuzioni di Linux, come ad esempio CentOS di cui si è già parlato in un paragrafo precedente, sono pensate per un utilizzo di tipo generico (sistemi *general purpose*) e sono quindi in grado di eseguire una grande varietà di software differenti, mentre altre sono pensate per ambienti di tipo *embedded* quindi per sistemi dedicati unicamente ad una determinata applicazione e non riprogrammabili dall'utente per altri scopi.

2.7.2 Scelta delle distribuzioni da utilizzare

Per creare le macchine virtuali utilizzate per realizzare le varie funzioni di rete virtuali si sarebbe potuta usare un'unica distribuzione di Linux completa, come ad esempio CentOS, Ubuntu o Debian che sarebbe stata in grado di fornire, tramite funzionalità integrate o software aggiuntivi, tutte le funzionalità necessarie. Però una distribuzione completa ha d'altra parte lo svantaggio di essere abbastanza complessa e di necessitare di risorse hardware (nel caso delle macchine virtuali si tratta di hardware virtualizzato ovviamente) abbastanza elevate per poter garantire prestazioni sufficienti. Per esempio, nel caso del sistema CentOS, è necessario almeno 1 GB di RAM per l'installazione che comprende l'interfaccia grafica, che si riducono a 512 MB se si sceglie di utilizzare solo il terminale.

Esistono numerose distribuzioni di Linux, spesso molto leggere, che sono state create con lo scopo di essere utilizzate solamente per realizzare determinate funzionalità specifiche, quali ad esempio router, firewall o simili. Ove è possibile utilizzare una distribuzione specifica per realizzare una determinata funzione è consigliabile sfruttarla in modo tale da massimizzare l'efficienza e minimizzare l'utilizzo delle risorse hardware, cosa molto importante specie nel caso in cui ci si trovi, come nel caso degli esperimenti condotti durante questo lavoro di tesi, a lavorare con molte macchine virtuali contemporaneamente. La maggior compattezza delle distribuzioni specifiche comporta inoltre ovvi vantaggi anche durante le operazioni di migrazione delle macchine virtuali, in quanto essendo minore la quantità di memoria utilizzata i trasferimenti saranno più veloci. Per le funzioni per le quali invece non esistono distribuzioni pensate appositamente

bisogna invece ricorrere a distribuzioni di tipo generale magari accompagnate dall'utilizzo di software specifici.

Per la realizzazione delle varie funzioni di rete da virtualizzare, negli esperimenti condotti, sono stati scelte determinate distribuzioni, che possono essere a seconda del caso distribuzioni specifiche o di uso generale:

- Storage: CentOS;
- Server video: Ubuntu;
- Server Web: CentOS;
- Routing, bridging e switching: ZeroShell;
- Firewall e security: IpCop.

2.7.3 CentOS: server web e storage

Per quanto riguarda CentOS è già stata data una descrizione sufficientemente completa nel primo paragrafo di questo capitolo, quindi ci limiteremo a ricordare che si tratta di un sistema operativo pensato per l'utilizzo in ambiti professionali e che si tratta di un sistema completo, in grado di svolgere numerose funzionalità, grazie anche alla grande disponibilità di software installabili. In particolare è stato scelto CentOS per l'utilizzo come server Web, in quanto è possibile installare facilmente su di esso la piattaforma LAMP, pensata appositamente per lo sviluppo di applicazioni web, la quale prende il nome dai suoi componenti di base: Linux (il sistema operativo su cui si basa), Apache (il server web vero e proprio), MySLQ (sistema di gestione di database) e Perl, PHP o Pyton (linguaggi di scripting). E' stato scelto di utilizzare CentOS anche per i servizi di storage in quanto non è stato possibile trovare una distribuzione specifica più adatta. Sia per il server web che lo storage è stata fatta una installazione senza interfaccia grafica che permette l'interazione solo tramite terminale, questa installazione necessità di 512 MB di RAM.

2.7.4 Ubuntu: server video

Ubuntu è un sistema operativo GNU/Linux basato su Debian pensato principalmente per sistemi di tipo desktop, ma ne esiste anche una versione apposita per server. Anche Ubuntu è come CentOS un sistema operativo completo in grado di supportare un gran numero di software molti dei quali provenienti da Debian con il quale condivide il sistema di gestione di pacchetti che è APT. Ubuntu è una delle distribuzioni di Linux più diffuse, in virtù della semplicità di utilizzo e della grande attenzione posta dagli sviluppatori sul supporto hardware. L'interazione dell'utente è possibile tramite interfaccia grafica grazie all'ambiente desktop Gnome oppure ovviamente, tramite terminale. Ubuntu è stato scelto per la funzione di server web, in quanto con soli 512 MB di RAM, a differenza che in CentOS, è possibile comunque installare l'interfaccia grafica che rende più comoda la gestione dei flussi video.

2.7.5 ZeroShell: routing, bridging e switching

ZeroShell è una distribuzione di Linux molto leggera creata per l'utilizzo in ambienti embedded e server. Si tratta di una distribuzione pensata in maniera specifica con lo scopo di fornire i principali servizi necessari in una rete locale quali le funzioni di router, bridge e switch. Il nome è dovuto alla disponibilità di un'interfaccia web per la configurazione, che permette quindi di evitare l'utilizzo della shell, cioè del terminale, che rimane comunque disponibile. ZeroShell è la scelta migliore per realizzare macchine virtuali che svolgano le funzioni di router, bridge e switch, essendo un sistema operativo molto leggero e creato appositamente per questi scopi. ZeroShell viene reso per di più disponibile, oltre che in versione Live CD, anche direttamente come immagine per i principali software di virtualizzazione tra cui Virtualbox.

2.7.6 IpCop: firewall e security

IpCop è una distribuzione di Linux molto leggera creata con lo scopo specifico di fornire un firewall semplice e configurabile. E' possibile scegliere in maniera personalizzata le politiche di sicurezza oppure sono disponibili delle

impostazioni standard, secondo queste ultime le interfacce vengono divise in quattro colori (rosso, arancio, verde e blu) a seconda del livello di sicurezza che si intende applicare. In virtù della sua leggerezza IpCop necessita per il suo funzionamento di risorse hardware molto limitate. IpCop è la scelta ideale per realizzare macchine virtuali che abbiano la funzione di firewall o più in generale che offrano servizi legati alla sicurezza di rete.

Capitolo 3
Implementazione e sperimentazione

Gli esperimenti condotti durante questo lavoro di tesi riguardano principalmente la Network Function Virtualisation e la sua applicazione ai servizi offerti dalle reti di accesso. In particolare durante questi esperimenti si è analizzata la possibilità di realizzare la virtualizzazione dei servizi di rete utilizzando unicamente hardware di tipo standard e software di virtualizzazione di tipo commerciale, già presenti sul mercato e ampiamente collaudati e utilizzati in altri ambiti quali principalmente il mondo del cloud computing.

Come descritto nel primo capitolo il passaggio da apparati di rete fisici a risorse virtualizzate comporta numerosi vantaggi e nuove possibilità. In modo particolare l'aspetto su cui più di ogni altro ci si è concentrati in questa tesi è la possibilità di migrare le macchine virtuali che realizzano le funzioni di rete da un host fisico a un altro. La migrazione, infatti, rappresenta una risorsa che permette di moltiplicare ancora di più i vantaggi offerti dalla virtualizzazione in quanto permette di amplificare le caratteristiche di flessibilità e versatilità delle funzioni di rete virtualizzate. La possibilità di migrare le risorse virtuali infatti permette di poter spostare le stesse attraverso la rete in modo tale da soddisfare al meglio, in ogni istante, le necessità sia dei gestori di telecomunicazioni che dei loro utenti.

Nel corso degli esperimenti si sono analizzati in modo particolare gli effetti che la migrazione delle risorse di rete virtuali ha sull'esperienza degli utenti e si sono indagati gli accorgimenti da mettere in atto affinché l'esecuzione delle migrazioni non abbia ripercussioni negative sulla qualità dei servizi di telecomunicazioni, o che per lo meno esse siano ridotte al minimo.

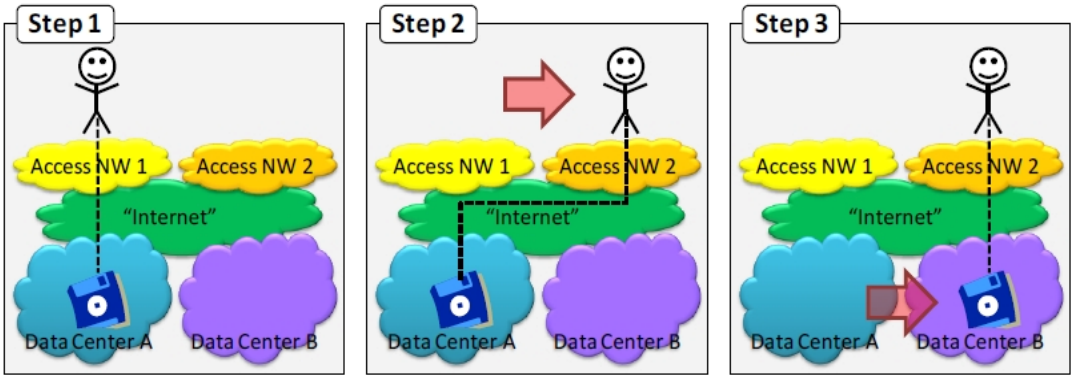

Figura 3.1: *Migrazione delle risorse di rete virtuali da un data center a un altro per venire incontro alle necessità degli utenti [13]*

Lo scenario di base su cui si basano tutti gli esperimenti prevede la presenza di due ipotetici *data center* di un certo fornitore di servizi di telecomunicazioni che si trovano all'interno della rete di quest'ultimo in due generiche posizioni, su di essi si trovano delle macchine virtuali con le quali sono realizzate vari servizi di rete virtualizzati. A questi data center, all'esterno, è collegato un ipotetico utente che vuole fruire delle funzionalità da essi offerte richiedendo servizi personalizzati e variabili in tempo reale. Si ipotizza poi che in un certo momento per venire incontro alle richieste dell'utente e alle proprie esigente l'operatore di telecomunicazioni voglia poter migrare i servizi di rete virtuali sfruttando le funzionalità di migrazione offerte dall'hypervisor utilizzato. Come già anticipato nel capitolo due i data center sono stati simulati nel corso degli esperimenti con due semplici server con sistema operativo Linux sui quali sono state create delle macchine virtuali che simulino le varie funzioni di rete, la connettività tra i data center è simulata tramite la connessione gigabit ethernet presente tra i due server mentre l'utente esterno è stato realizzato tramite un computer esterno.

Gli esperimenti che sono stati oggetto di studio sono in tutto tre. Nel primo esperimento è presente una piccola rete virtuale costituita da due macchine virtuali che realizzano dei router, i quali devono poter essere migrati contemporaneamente da un server all'altro qualora sia necessario. Nel secondo esperimento sono presenti un server virtuale in grado di erogare un flusso video ed un router virtuale che ne permette la connettività all'esterno, essi devono poter essere migrati insieme affinché il flusso video possa raggiungere un utente

esterno. Nel terzo esperimento infine viene simulata una piccola rete di accesso di un ipotetico provider la quale fornisce un certo numero di servizi virtualizzati a un utente e deve poter essere migrata nel suo insieme da un server all'altro senza che vi siano ripercussioni sull'utente.

3.1 Esperimento 1: due router

3.1.1 Descrizione e scopo dell'esperimento

Questo primo esperimento si pone come obiettivo la simulazione di una rete composta unicamente da nodi di commutazione virtuali, in sostanza un insieme di macchine virtuali configurate per funzionare come router. Questa rete deve poter essere migrata in tempo reale da un data center ad un altro a seconda delle necessità di gestione dell'operatore o delle richieste dell'utente dei servizi di telecomunicazioni senza che ciò si ripercuota sulla qualità del servizio

Per questioni di semplicità la rete ricreata durante l'esperimento prevede solamente due nodi virtuali, cioè due macchine virtuali configurate per funzionare come router, in grado di dialogare tra di loro e con la macchina ospite su cui sono in esecuzione. Questa semplificazione non comporta una perdita di generalità in quanto il caso di studio è facilmente estendibile a casi generici che prevedano un numero maggiore di nodi.

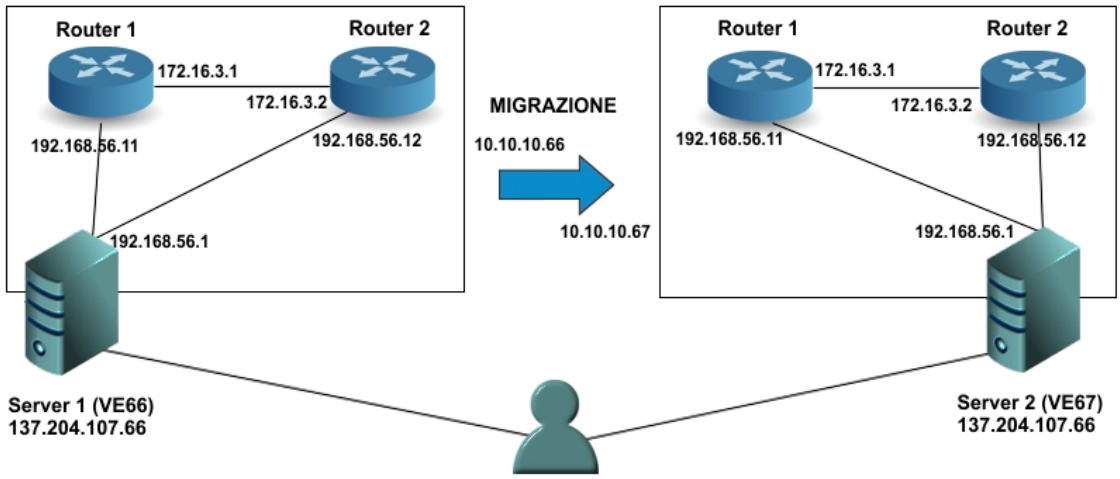

Figura 3.2: *Scenario del primo esperimento*

Per poter effettuare le migrazioni dei nodi di rete virtuali tra i due server che simulano i data center occorre che le macchine virtuali che li realizzano siano replicati identiche sui due server, sia per quanto riguarda le configurazioni software che hardware (ovviamente virtuale).

3.1.2 Creazione e configurazione delle macchine virtuali

Per la realizzazione delle macchine virtuali da utilizzare come router in questo esperimento è stata scelta una installazione minimale del sistema operativo CentOS 6.3 che non prevede l'interfaccia grafica ma solamente l'accesso tramite linea di comando.

Dato che, come già anticipato, per poter permettere le migrazioni è necessario disporre di macchine virtuali identiche sui due server, è stato sufficiente creare le macchine sul primo server (VE66) per poi ricopiarle identiche sul secondo (VE67) sfruttando la funzionalità di esportazione e importazione presente in VirtualBox.

Per creare il primo router è stato necessario creare una nuova macchina virtuale denominata router1 attraverso l'apposita procedura prevista da VirtualBox, è stato innanzitutto necessario scegliere il tipo di sistema operativo cioè Linux, poi è stata fatta la scelta dell'hardware virtuale da utilizzare. Per quanto riguarda quest'ultima questione si è deciso, al fine di ottenere una macchina virtuale più leggera e quindi poi anche meno problematica da migrare, di assegnare alla macchina le risorse hardware minime in grado di fare funzionare il sistema operativo, cioè in particolare un processore con un singolo core, 512 MB di RAM e un hard disk virtuale dinamico con capacità massima di 8 GB. Scelta importante è stata poi quella delle interfacce di rete da utilizzare, ne sono state create tre di cui due di tipo *host-only* che realizzano tramite bridge virtuali appositamente creati rispettivamente il collegamento tra le due macchine virtuali (ETH2) e quello con la macchina ospite (ETH0) e una di tipo bridge (ETH1)

collegata ad una della interfacce fisiche del server per scopi di manutenzione. Al primo avvio della macchina è stato poi necessario provvedere all'installazione del sistema operativo che è avvenuta con il metodo della net install.

Per creare la seconda macchina virtuale, denominata router2, dato che deve essere in tutto e per tutto identica a quella già creata è stato sufficiente replicare la prima con la funzione esporta/importa, con l'accortezza di andare poi a rigenerare gli indirizzi MAC fisici delle schede di rete per evitare conflitti.

Una volta realizzati i due router sono state configurate le interfacce di rete, tramite il tool di configurazione per CentOS system-config-network-tui. Alla rete virtuale (denominata vboxnet0) che collega i router alla macchina ospite è stata assegnata la rete IP privata 192.168.56.0/24 dove al server fisico è stato assegnato l'indirizzo .1, al primo router l'indirizzo .11 e al secondo l'indirizzo .12. Per il collegamento tra i due router (denominato vboxnet1) è stata dedicata la rete 172.16.3.0/24 dove alle interfacce dei due router sono stati assegnati rispettivamente gli indirizzi .1 e .2. Infine per i collegamenti di tipo bridge per la gestione sono state assegnate le reti 172.16.1.0/24 per quello del primo router e 172.16.2.0/24 per quello del secondo, alle interfacce dei router collegate a queste reti è assegnato in entrambi i casi l'indirizzo .1. Dato che queste due macchine devono funzionare da router è stato opportuno abilitare l'IP forwarding su di esse. Inoltre per poter permettere ai router di parlare con l'esterno è stato necessario configurare un NAT sulla macchina ospite che faccia la traslazione di indirizzo IP dall'indirizzo di rete privato dei router a quello pubblico del server.

Terminata la configurazione delle macchine virtuali sul primo server è stato necessario replicarle sul secondo server. Per fare ciò si sono dovute innanzitutto esportare le macchine, poi sono stati creati dei file di esportazione che sono stati copiati sul secondo server tramite il comando SCP e infine su di esso questi sono stati importati in VirtualBox. Sul secondo server è stato inoltre necessario ricreare i bridge virtuali vboxnet0 e vboxnet1 e si è dovuto, come già fatto prima, configurare il NAT per permettere ai router di comunicare con l'esterno, infine è

stato necessario controllare che l'interfaccia dei router collegata in bridge fosse connessa alla scheda di rete fisica giusta.

3.1.3 Esecuzione delle migrazioni

Per poter effettuare le migrazioni in VirtualBox è necessario operare da terminale attraverso il comando VBoxManage. La prima cosa da fare quando si vogliono fare delle migrazioni è configurare le macchine virtuali sul server di destinazione (VE67 nel nostro caso) per poter ricevere un teleporting specificando la porta TCP (diversa per ogni macchina da migrare, in questo caso sono state utilizzate le porte 1234 e 1235) da utilizzare per la migrazione, l'indirizzo su cui ricevere la migrazione ed una password per l'operazione.

Nel nostro caso i comandi da lanciare per predisporre i due router sul secondo server per il teleporting sono:

```
[root@i2-ve067] VBoxManage modifyvm router1 --teleporter on
--teleporterport 1234 --teleporteraddress 10.10.10.67 --
teleporterpassword prova

[root@i2-ve067] VBoxManage modifyvm router2 --teleporter on
-- teleporterport 1235 --teleporteraddress 10.10.10.67 --
teleporterpassword prova
```

Quando sono configurate per il teleporting le macchine virtuali al loro avvio invece di caricare il sistema operativo si mettono in ascolto in attesa di una migrazione.

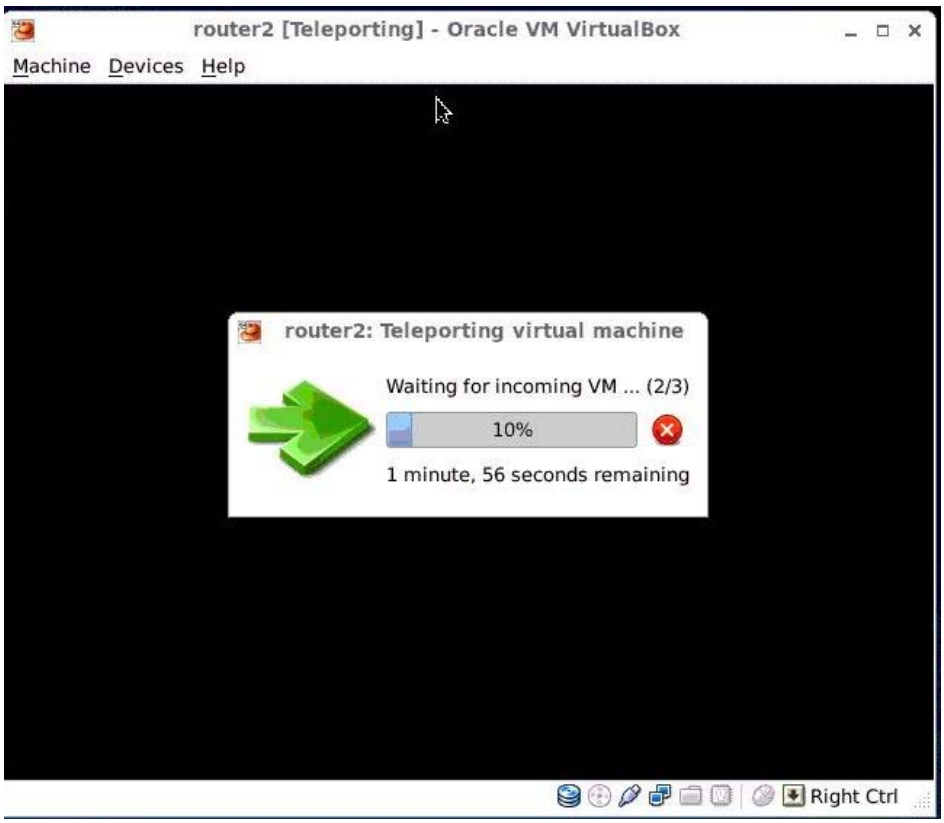

Figura 3.3: *Macchina virtuale sul'host di destinazione in attesa del teleporting*

Sul server di origine invece innanzitutto occorre che le macchine virtuali da migrare siano accese e funzionanti per poter procedere con la migrazione, anche in questo caso bisogna agire da terminale con VBoxManage per poter dare alle macchine il comando di migrazione. Nel comando vanno specificati l'host di destinazione, la porta da utilizzare e la password per la procedura, ovviamente porta e password devono essere gli stessi impostati sulla macchina di destinazione. E' opportuno evidenziare che quando viene dato il comando di migrazione ad una macchina virtuale è necessario affinché tutto vada a buon fine che sul server di destinazione ci sia una macchina identica in ascolto.

Nel caso in oggetto i comandi da lanciare per avviare la migrazione dei due router sono:

```
[root@i2-ve066] VBoxManage controlvm router1 teleport --
host 10.10.10.67 --port 1234 --password prova

[root@i2-ve066] VBoxManage controlvm router2 teleport --
host 10.10.10.67 --port 1235 --password prova
```

Una volta lanciato il commando di migrazione, le macchine virtuali iniziano la procedura di migrazione e quando questa è terminata riprendono la loro esecuzione sul secondo server, il tempo totale di questa operazione, come mostrato più precisamente nel paragrafo sulle macchine virtuali nel secondo capitolo, dipende dalla quantità di dati in memoria da trasferire. E' importante notare che VirtualBox è in grado di eseguire una sola migrazione alla volta, quindi se vengono lanciati più comandi di migrazione contemporaneamente il programma migrerà prima una macchina, poi una volta terminato il processo, provvederà a migrare la seconda.

3.1.4 Misure

Per poter valutare le prestazioni delle migrazioni i due parametri fondamentali sono il tempo totale di migrazione cioè quanto trascorre tra quando viene dato il comando di migrazione e la ripresa dell'esecuzione sul secondo host e soprattutto il downtime cioè il tempo in cui la macchina virtuale rimane non operativa tra quando viene interrotta sul primo host e quando l'esecuzione riprende sul secondo.

Per quanto riguarda la migrazione di una singola macchina virtuale sia il tempo totale di migrazione che il downtime sono molto limitati tanto che si può considerare l'operazione quasi immediata. Quando però ci si trova a che fare con migrazioni multiple di più macchine virtuali contemporaneamente come in questo caso la situazione si complica, infatti come già anticipato anche se sono lanciate contemporaneamente le migrazioni vengono eseguite solo una alla volta. Per questo motivo sicuramente più sono le macchine virtuali da migrare più il tempo necessario per migrarle tutte aumenterà.

Un buon modo per valutare il tempo necessario a migrare i due router virtuali oggetto dell'esperimento è quello di lanciare un ping tra uno e l'altro e vedere quanti pacchetti ICMP vanno persi.

Se si lancia un ping tra i due router con l'intervallo di tempo standard di un secondo e si esegue la migrazione, ripetendo la prova più volte si nota che a volte viene perso un pacchetto, mentre altre volte no, quindi è possibile affermare che il downtime è quantificabile in meno di un secondo. Per poter avere una stima più accurata è possibile ripetere la prova scegliendo un intervallo di mezzo secondo tra due pacchetti, in questo caso si nota, ripetendo l'esperimento che si ha sempre la perdita di un pacchetto, questo permette di quantificare il downtime in un valore prossimo al mezzo secondo. Invertendo l'ordine in cui avviene la migrazione non si notano differenze, essendo le due macchine virtuali identiche.

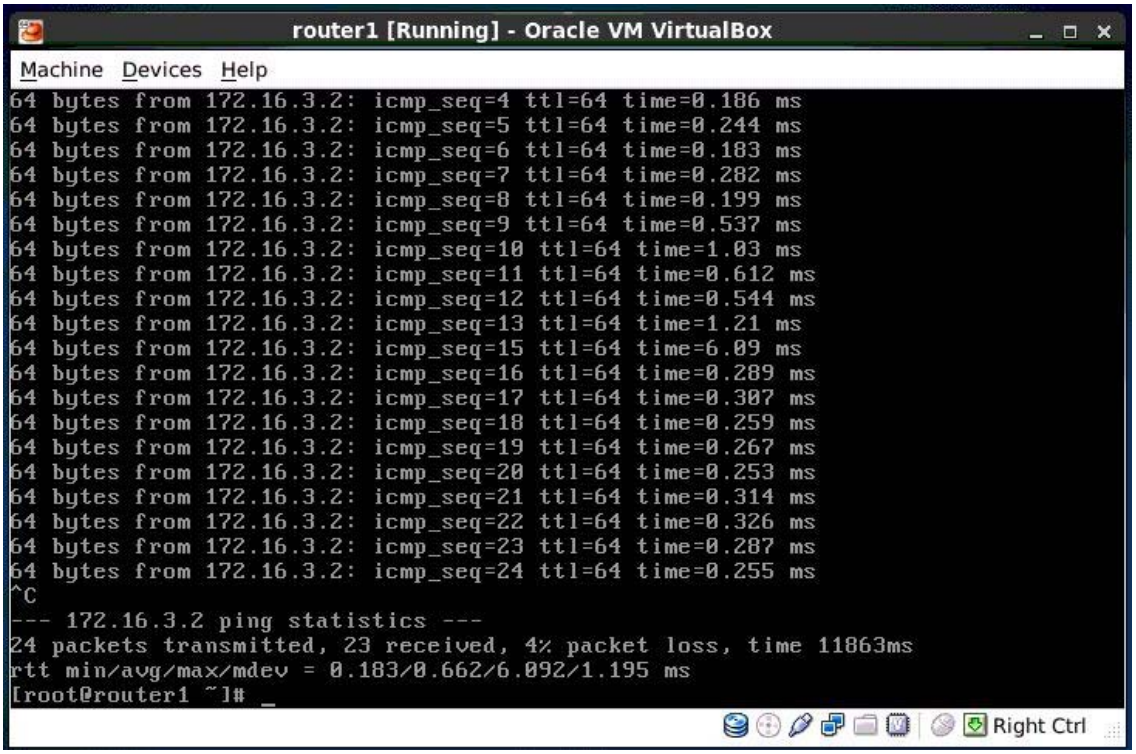

Figura 3.4*: Ping da router1 a router2 con intervallo di 0,5 secondi, si nota la perdita del pacchetto numero 14*

Un'altra misura interessante che può essere fatta facilmente consiste nell'analizzare il traffico di migrazione mettendo un analizzatore di protocollo in ascolto sulla interfaccia di uno dei due server dedicata al collegamento punto-punto su cui avviene la migrazione, quello che si può vedere è un flusso TCP dal primo al secondo host.

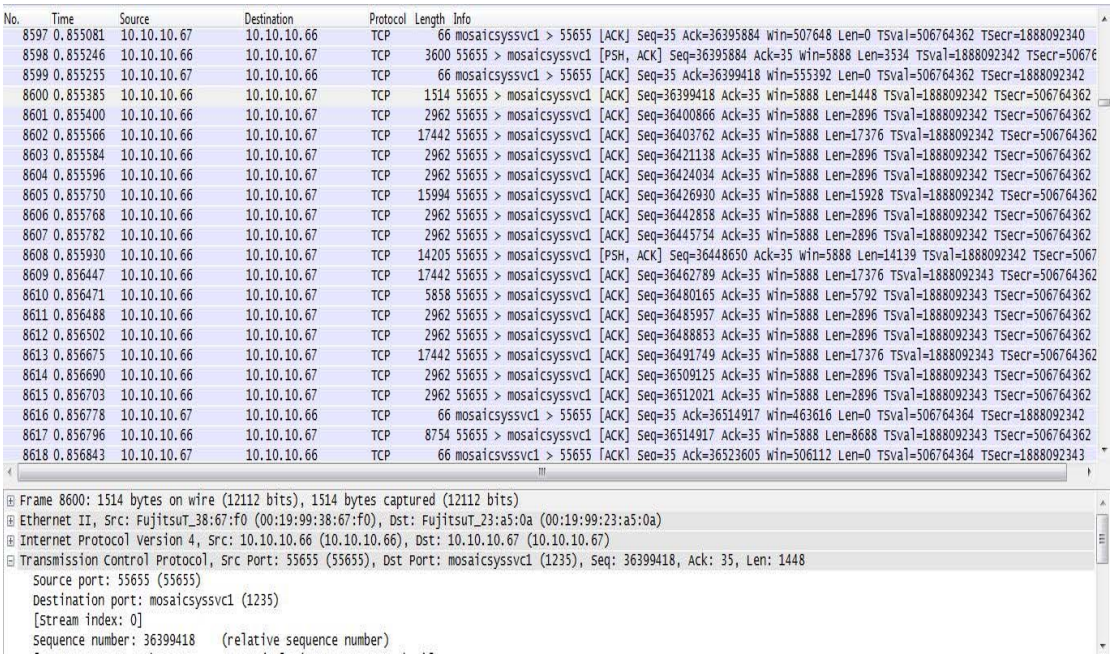

Figura 3.5: *Cattura del traffico di migrazione tramite analizzatore di protocollo*

3.1.5 Conclusioni

Questo esperimento mostra come, utilizzando tecnologie di virtualizzazione già disponibili sul mercato, sia possibile creare una rete composta unicamente da nodi di commutazione virtuali in grado di instradare pacchetti e di comunicare tra di loro e con l'esterno, e come, attraverso i meccanismi di migrazione, si possano migrare facilmente queste risorse da una macchina fisica ad un'altra per venire incontro alle necessità dell'operatore e degli utenti.

Le prove e le misure effettuate indicano che sfruttando le tecnologie di migrazione rese disponibili dall'hypervisor utilizzato è possibile effettuare le migrazioni delle risorse virtuali in tempi relativamente brevi, dipendentemente dalla quantità di memoria RAM utilizzata. Un'importante osservazione riguarda il fatto che non è possibile migrare più di una macchina virtuale alla volta, il che comporta che se ci si trovasse a dover migrare delle reti composte da molte risorse virtuali, sebbene le singole migrazioni siano relativamente veloci, i tempi necessari per il trasferimento crescerebbero necessariamente rischiando di causare interruzioni del servizio anche considerevoli.

3.2 Esperimento 2: server video e router

3.2.1 Descrizione e scopo dell’esperimento

Al centro di questo secondo esperimento c’è una piccola rete virtuale in cui sono presenti un router virtuale e un server virtuale in grado di erogare un flusso video. Il server video è collegato unicamente con il router tramite un bridge virtuale, mentre il router è collegato anche con la macchina ospite fisica ed è responsabile della connettività del server con l’esterno. Dato che il server video dialoga con l’esterno solo tramite il router ha bisogno necessariamente di esso per poter funzionare erogando il suo flusso video ad un utente esterno.

Il server video deve poter essere migrato in tempo reale a seconda della necessità da un data center ad un altro, questo comporta necessariamente, affinché esso continui a comunicare con l’esterno, che venga migrato insieme ad esso anche il router. Lo scopo dell’esperimento è verificare l’effetto che le migrazioni hanno su di un ipotetico utente del servizio offerto dalle due macchine.

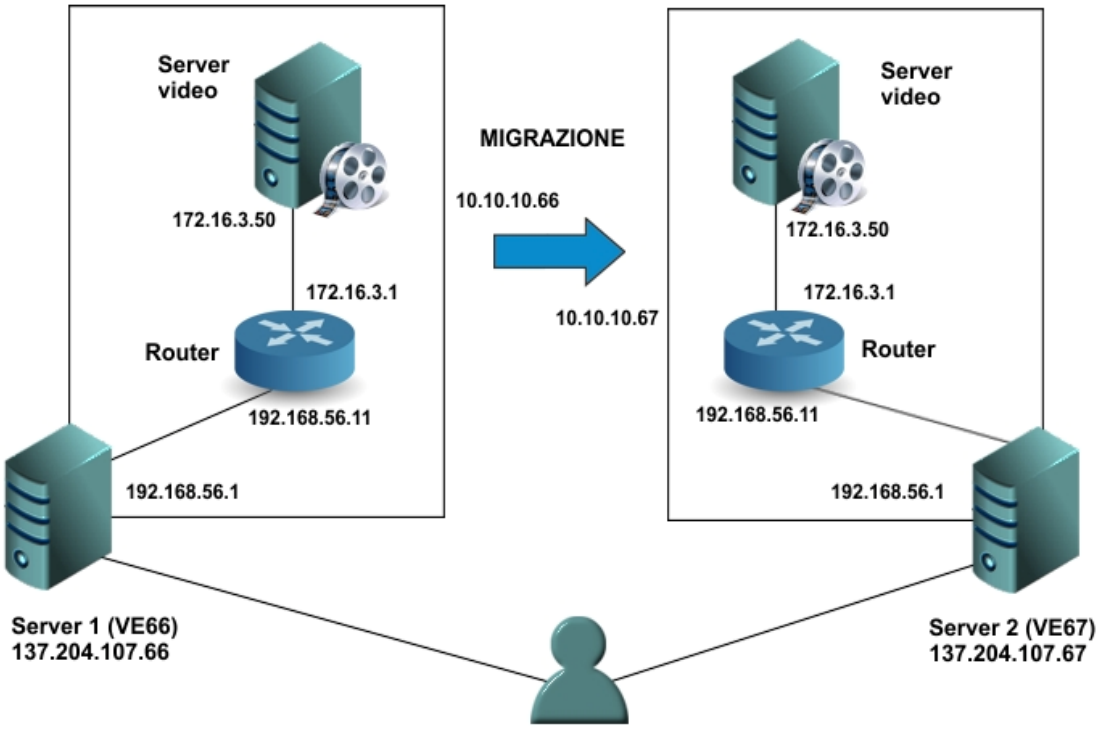

Figura 3.6: *Scenario del secondo esperimento*

Come nel primo esperimento anche in questo caso perché siano possibili le migrazioni le macchine virtuali che realizzano il server e il router devono esistere identiche sui due server che simulano i due data center.

3.2.2 Creazione e configurazione delle macchine virtuali

Per quanto riguarda il router è stato possibile riutilizzare una delle macchine virtuali già create per l'esperimento precedente, è stato scelto di utilizzare router1 che era già presente su entrambi i server. Tutto l'hardware ed in modo particolare le interfacce di rete sono rimaste inalterate rispetto al caso precedente. Più specificatamente ETH0 è collegata al bridge virtuale vboxnet0 e permette la comunicazione con la macchina ospite, ETH1 è predisposta per la gestione ed è collegata in bridge ad una delle interfacce di rete fisiche, mentre ETH2 è dedicata al collegamento tra le macchine virtuali realizzato con il bridge virtuale vboxnet1, in particolare tale collegamento mentre prima era usato per fare

dialogare i due router ora serve per collegare il router al server video. Anche gli indirizzi IP assegnati alle interfacce sono rimasti gli stessi.

Per il server video è stato invece scelto di utilizzare una macchina virtuale con in esecuzione il sistema Ubuntu nella versione dotata di interfaccia grafica, la quale era già presente sui server in quanto utilizzata per precedenti esperimenti sulla migrazione dei server video virtuali. Per quanto riguarda le risorse hardware virtuali questa macchina è configurata con un processore single core, 512 MB di RAM, hard disk virtuale dinamico con dimensione massima di 8 GB e una sola interfaccia di rete (ETH0). Questa interfaccia è stata impostata in modo da essere collegata tramite il bridge virtuale vboxnet1 alla ETH2 del router, è stato poi assegnato ad essa l'indirizzo IP 172.16.3.100.

3.2.3 Esecuzione delle migrazioni

Per eseguire le migrazioni occorre procedere come nel caso precedente. Per prima cosa occorre impostare le macchine virtuali sul server destinazione (VE67) per ricevere il teleporting, specificando porte e indirizzo, tramite i comandi da terminale:

```
[root@i2-ve067] VBoxManage modifyvm router1 --teleporter on
--teleporterport 3000 --teleporteraddress 10.10.10.67 --
teleporterpassword prova

[root@i2-ve067] VBoxManage modifyvm Video --teleporter on -
-teleporterport 4000 --teleporteraddress 10.10.10.67 --
teleporterpassword prova
```

Avviando queste macchine esse si metteranno in ascolto in attesa del teleporting. Una volta che le macchine sull'host di destinazione sono state messe in ascolto è possibile lanciare dal server di origine (VE66) la migrazione delle due macchine virtuali che devono essere già in esecuzione su di esso. Tenendo conto dell'indirizzo dell'host di destinazione e delle porte scelte sopra i comandi da lanciare da terminale per avviare le migrazioni sono:

```
[root@i2-ve066] VBoxManage controlvm router1 --host
10.10.10.67 --port 3000 --password prova
```

```
[root@i2-ve066] VBoxManage controlvm Video --host
10.10.10.67 --port 4000 --password prova
```

Una volta lanciati i comandi, in modo identico a come avveniva nel primo esperimento, vengono portate a termine le due migrazioni che vengono eseguite ancora una alla volta. In questo caso il tempo totale necessario alla migrazione è maggiore che nel caso precedente e questo è dovuto al fatto che la macchina virtuale che svolge la funzione di server video essendo più complessa ha una quantità maggiore di dati in memoria da trasferire.

3.2.4 Automatizzazione delle migrazioni con il protocollo SIP

Come mostrato poco fa e come visto già anche nel primo esperimento quando si vogliono eseguire delle migrazioni di macchine virtuali sono necessarie una serie di azioni che devono essere eseguite a mano dall'operatore. In particolare la procedura da seguire per effettuare delle migrazioni prevede per prima cosa che si impostino tramite terminale le macchine virtuali sull'host di destinazione per ricevere il teleporting, poi esse devono essere avviate in modo che si mettano in ascolto, infine occorre lanciare la migrazione delle macchine virtuali tramite terminale dall'host sorgente. Nel caso si vogliano migrare molte macchine virtuali, come ad esempio nel caso di una rete virtuale che comprende svariati servizi, come quella che si vedrà nel terzo esperimento, questa procedura può risultare scomoda e dispendiosa in termini di tempo, oltre al fatto che essendo abbastanza complessa potrebbe lasciare spazio ad errori che potrebbero compromettere l'esito della migrazione. Per queste ragioni può essere utile disporre di un software che automatizzi tutta la procedura lasciando all'operatore unicamente la scelta di quali macchine virtuali migrare, in quale momento e tra quali host.

Per poter automatizzare le procedure di migrazione con riferimento alla rete presentata durante questo esperimento si è scelto di ricorrere ad un software in PHP scritto precedentemente da uno studente della facoltà [14]. Questo software

si basa sul protocollo SIP e prevede due moduli che devono essere collocati uno sull'host sorgente e uno sull'host destinazione, e che sono accessibili tramite browser web. Il modulo sull'host di destinazione ha solo lo scopo di mettersi in ascolto in attesa di una connessione mentre quello sull'host sorgente presenta una interfaccia grafica con cui l'operatore può interagire per impostare i parametri della migrazione.

Quando si intende procedere con una migrazione bisogna collegarsi tramite browser web ai due moduli, quello sull'host destinazione si mette in ascolto mentre con quello sull'host sorgente si può interagire per comandare la migrazione. Per procedere ad una migrazione la prima cosa da fare è instaurare un dialogo SIP tramite il pulsante *call* sul modulo presso l'host sorgente, occorre poi scegliere quale macchina virtuale migrare e verso quale host e premere il pulsante *migrazione*, a questo punto inizia automaticamente la migrazione, infine una volta terminato il processo è necessario rilasciare la connessione SIP tramite il pulsante *bye*.

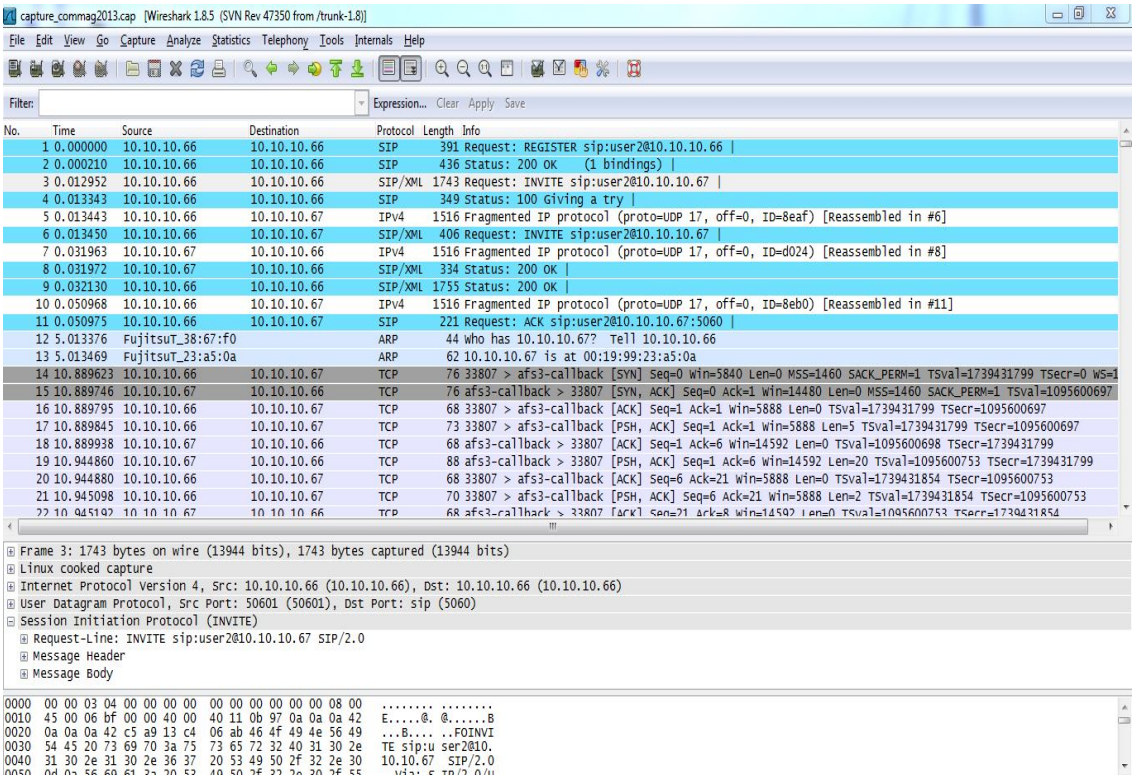

Figura 3.7: *Cattura con analizzatore di protocollo del traffico generato dalla migrazione con in evidenza il messaggio SIP di tipo INVITE che apre la sessione*

Inizialmente questo software era in grado di migrare una sola macchina virtuale per sessione quindi è stato necessario apportargli alcune piccole modifiche per estenderlo in modo da permettere la migrazione di entrambe le macchine virtuali oggetto dell'esperimento, cioè il router e il server video, le quali devono sempre essere in esecuzione insieme per svolgere le proprie funzioni.

3.2.5 Cambiamento di piattaforma: da VirtualBox a QEMU/KVM

Finora tutte le prove che sono state presentate sono state svolte sfruttando come software di virtualizzazione VirtualBox. Può però essere interessante analizzare cosa succede se si usa un differente hypervisor, in particolare notare quali differenze e quali similitudini si riscontrano. A questo scopo è stato scelto di ripetere l'esperimento di migrazione del sistema composto dal server video e dal router sfruttando come hypervisor QEMU/KVM invece che VirtualBox. Come anticipato nel secondo capitolo per motivi di spazio è stato necessario per questa prova lasciare da parte i server utilizzati precedentemente e ricorrere ad altre due macchine.

Nel passaggio non è stato necessario realizzare delle nuove macchine virtuali da zero su QEMU/KVM ma è stato possibile sfruttare gli hard disk virtuali delle macchine già create su Virtualbox. Infatti in QEMU è disponibile una funzionalità di conversione che è in grado di trasformare gli hard disk virtuali dal formato .vdi proprietario di Virtualbox al formato .qcow di KVM. In pochi passi è stato quindi possibile ricreare con il nuovo hypervisor uno scenario identico a quello mostrato precedentemente.

QEMU/KVM sfrutta lo stesso metodo di migrazione di Virtualbox ed offre in più la possibilità di effettuare le migrazioni tramite interfaccia grafica. Anche con questo hypervisor rimane come nel caso precedente il limite massimo di una sola migrazione alla volta. La versione di KVM/QEMU a nostra disposizione presenta purtroppo un limite che non permette di effettuare le migrazioni allo stesso modo di come era possibile con Virtualbox, infatti esso necessita per poter effettuare le

migrazioni che la macchina virtuale di origine e di destinazione condividano un unico hard disk che si deve trovare su di una locazione di rete comune.

3.2.6 Misure

Le misure presentate in questo paragrafo si riferiscono al caso dell'esecuzione delle migrazioni in maniera automatizzata tramite il protocollo SIP, per condurre queste misure è stato utilizzato un PC esterno (avente indirizzo IP pubblico 137.204.107.197) per simulare un utente esterno del servizio.

La prima prova che è stata condotta è nuovamente quella del ping. E' stato quindi lanciato un ping dal PC esterno alla macchina virtuale video e contemporaneamente è stata effettuata la migrazione di essa e del relativo router. In particolare selezionando come intervallo temporale tra un pacchetto e l'altro 0,1 secondi è stato possibile notare che vengono persi all'incirca trenta pacchetti, il che significa una perdita di connettività di circa tre secondi.

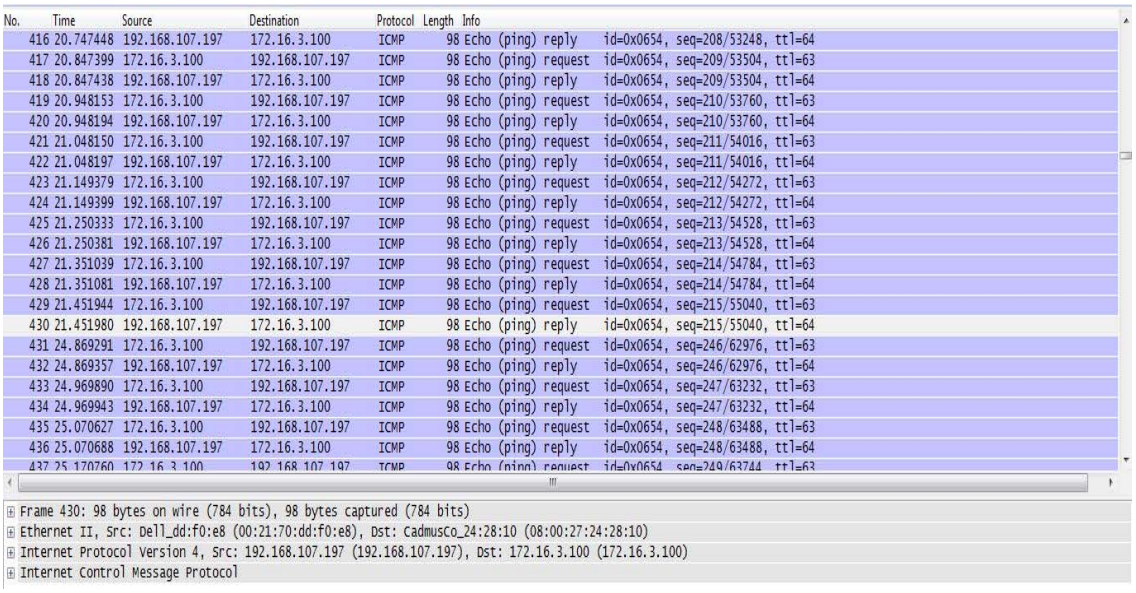

Figura 3.8: *cattura della prova ping con intervallo di 0.1 secondi, in questo caso son stati persi 31 pacchetti, infatti si salta dal numero 215 al 246*

La seconda prova, che probabilmente è ancora più significativa, consiste sostanzialmente in una misura di banda.

Lo scenario ipotetico su cui si basa questa prova prevede che ci sia un utente che stia fruendo di un contenuto video in qualità standard presente su di un server video virtuale in esecuzione in un data center, il quale è collegato all'esterno tramite un router virtuale che permette ad esso la connettività con l'utente. Ad un certo punto per motivi di gestione o per venire incontro alle necessità dell'utente l'operatore decide di migrare il server video e il router verso un secondo data center, questo comporterà inevitabilmente che il flusso video e quindi il servizio di cui l'utente sta fruendo si interromperà per un certo tempo nel momento in cui le due macchine virtuali vengono migrate. Una volta ripresa la trasmissione inoltre l'utente decide di passare dal flusso video a definizione standard che stava guardando finora ad un video in alta definizione comportando necessariamente un aumento della banda utilizzata.

Per realizzare i contenuti video sono stati caricati sulla macchina virtuale Video due filmati identici ma uno in definizione standard e uno in alta definizione, tramite il programma VLC è possibile trasmettere questi video verso un indirizzo di rete, in questo caso quello del PC esterno che simula l'utente, utilizzando dei comandi da terminale, che possono anche essere messi in uno script:

```
#!/sbin/bash

cvlc Hero_chap8_HQ.avi --sout
"#std{access=udp,mux=ts,dst=137.204.107.197:1234}"&
cvlc Hero_chap8_LQ.avi --sout
"#std{access=udp,mux=ts,dst=137.204.107.197:1235}"
```

Per selezionare quale dei due flussi video lasciar passare, sfruttando essi due porte UDP diverse, basta bloccare uno o l'altro aggiungendo delle regole al firewall di iptables sul router. Inizialmente essendo richiesto il video in qualità standard il flusso che deve essere bloccato è quello ad alta definizione (che sfrutta la porta UDP 1234):

```
[root@router1] iptables -I FORWARD 1 -p  udp --dport 1234 -
j DROP
```

Poi quando viene richiesto invece il video in qualità superiore bisogna eliminare la regola che blocca il video in alta definizione e aggiungerne una che blocchi quello in qualità standard (che sfrutta la porta UDP 1235):

```
[root@router1] iptables -D FORWARD 1

[root@router1] iptables -I FORWARD 1 -p udp --dport 1235 -j
DROP
```

Per analizzare l'utilizzo di banda durante la migrazione e durante la riconfigurazione della qualità del video è possibile utilizzare un apposito script che sfrutta per la misurazione il *timestamp* di iptables e che deve essere lanciato sul PC che simula l'utente esterno:

```
#! /bin/bash

if [ $# -lt 3 ]
then
  echo "Usage: $0 <sampling period in seconds> <number of
delta estimation samples> <correction>">&2
  exit -1
fi

period=$1
deltasamp=$2
correct=$3
bwscale=$(echo "scale=6; 1/$period" | bc)

echo "Estimating delta with $deltasamp samples...">&2
delta=0.0
s=0
B1prev=0
B2prev=0
B3prev=0
B4prev=0
while [ $s -lt $deltasamp ]
do
  t=`./gettimestamp`
  B1=`iptables -nvx -L INPUT |grep .107.66 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B2=`iptables -nvx -L INPUT |grep .107.66 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B3=`iptables -nvx -L INPUT |grep .107.67 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B4=`iptables -nvx -L INPUT |grep .107.67 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  b1=$(echo "scale=6; ($B1-$B1prev)*8*$bwscale" | bc)
  b2=$(echo "scale=6; ($B2-$B2prev)*8*$bwscale" | bc)
  b3=$(echo "scale=6; ($B3-$B3prev)*8*$bwscale" | bc)
  b4=$(echo "scale=6; ($B4-$B4prev)*8*$bwscale" | bc)
```

```
  echo "$t  $b1 $b2 $b3 $b4">&2
  B1prev=$B1
  B2prev=$B2
  B3prev=$B3
  B4prev=$B4
  t1=`./gettimestamp`
  delta=$(echo "scale=6; ($t1-$t)+$delta" | bc)
  s=$(($s+1))
  #echo "$s: $delta"
done

echo "Done!">&2
delta=$(echo "scale=6; $delta/$s" | bc)
tsleep=$(echo "scale=6; $period-$delta-$correct" | bc)
echo "--- DELTA = $delta ----">&2
echo "--- SLEEP = $tsleep ----">&2
echo>&2

echo "# Sampling bandwidth every $period seconds">&2
echo "# Bandwidth scale factor = $bwscale">&2
echo>&2

echo "# time    UDP-1234-66 UDP-1235-66 UDP-1234-67 UDP-
1235-67"
echo

B1prev=`iptables -nvx -L INPUT |grep .107.66 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
B2prev=`iptables -nvx -L INPUT |grep .107.66 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
B3prev=`iptables -nvx -L INPUT |grep .107.67 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
B4prev=`iptables -nvx -L INPUT |grep .107.67 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
while true
do
  t=`./gettimestamp`
  B1=`iptables -nvx -L INPUT |grep .107.66 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B2=`iptables -nvx -L INPUT |grep .107.66 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B3=`iptables -nvx -L INPUT |grep .107.67 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B4=`iptables -nvx -L INPUT |grep .107.67 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  b1=$(echo "scale=6; ($B1-$B1prev)*8*$bwscale" | bc)
  b2=$(echo "scale=6; ($B2-$B2prev)*8*$bwscale" | bc)
  b3=$(echo "scale=6; ($B3-$B3prev)*8*$bwscale" | bc)
  b4=$(echo "scale=6; ($B4-$B4prev)*8*$bwscale" | bc)
  echo "$t  $b1 $b2 $b3 $b4"
  B1prev=$B1
  B2prev=$B2
  B3prev=$B3
```

```
  B4prev=$B4
  sleep $tsleep
done
```

Lo script per funzionare necessita della creazione di apposite regole di iptables:

```
iptables -I INPUT 1 -s 137.204.107.66 -p udp --dport 1234 -
j ACCEPT

iptables -I INPUT 2 -s 137.204.107.66 -p udp --dport 1235 -
j ACCEPT

iptables -I INPUT 3 -s 137.204.107.67 -p udp --dport 1234 -
j ACCEPT

iptables -I INPUT 4 -s 137.204.107.67 -p udp --dport 1235 -
j ACCEPT

iptables -I INPUT 5 -s 137.204.107.0/24 -j ACCEPT
```

E’inoltre necessaria la presenza di un file eseguibile nominato timestamp ottenuto dalla compilazione di questo codice:

```
# gettimestamp.c

#include <sys/time.h>
#include <stdio.h>

main (void) {
  struct timeval currentTime;
  gettimeofday(&currentTime,NULL);
  printf("%.6f\n",(double)currentTime.tv_sec+currentTime.tv
_usec/1000000.0);
}
```

Nel lanciare lo script devono essere specificati l’intervallo temporale di misurazione, il numero di campioni desiderati ed un fattore di correzione, è inoltre possibile specificare un file in cui salvare i risultati della misura:

```
[root@137.204.107.197] ./bwmeasure_video.sh 0.5 200 0 >
migra.dat
```

Nel caso specifico lo script è stato lanciato specificando un intervallo temporale di mezzo secondo e duecento campioni, il fattore di correzione è stato posto a zero. Analizzando i risultati della misura si nota che la migrazione ha comportato una interruzione del servizio di sei secondi e mezzo, infatti sono tredici gli istanti, spaziati di 0,5 secondi in cui la banda utilizzata si è mantenuta a zero.

E’ stato possibile utilizzare il file con i risultati della misura per creare un grafico che mostra l’utilizzo della banda nel tempo, nel quale sono evidenti l’intervallo di tempo in cui avviene la migrazione e l’istante in cui si ha il passaggio dal flusso video a definizione standard a quello ad alta definizione. Le due cose più interessanti che si possono notare dal grafico sono la durata dell’interruzione dovuta alla migrazione delle macchine virtuali che è di circa sei secondi e mezzo come già visto con il ping e il fatto che la banda utilizzata passa da un valore che si aggira attorno a 1,5 MB/s durante la trasmissione del flusso a bassa qualità ad un valore che varia tra i 2 e i 6 MB/s quando si passa al flusso in alta definizione.

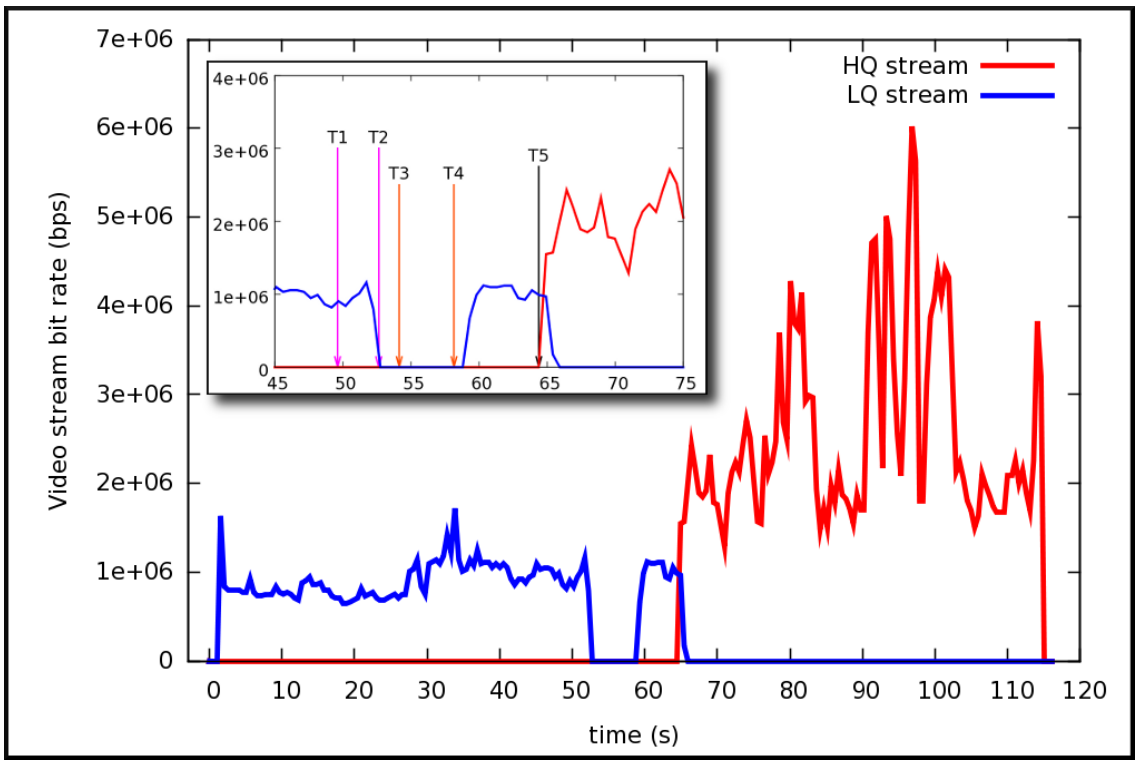

Figura 3.9: *Grafico rappresentante l’andamento dell’utilizzo della banda nel tempo, nella figura in piccolo sono evidenziati gli istanti di inizio e fine della migrazione del server video (T1 e T2) e di quella del router (T3 e T4) e l’istante in cui si ha il cambio di qualità del flusso video (T5)*

3.2.7 Conclusioni

Questo esperimento mostra come sia possibile realizzare una rete all’interno della quale non sono virtualizzati solo i nodi di commutazione ma anche gli altri servizi compresi quelli che prevedono l’erogazione di contenuti multimediali agli utenti e come anche in questo caso si possano sfruttare le funzionalità di

migrazione per offrire dei servizi che si adattino alle esigenze dell'operatore e degli utenti.

Le prove di automatizzazione della migrazione tramite il protocollo SIP evidenziano come sia possibile con pochi strumenti aggiuntivi eliminare la necessità di ricorrere a numerosi comandi manuali da parte dell'operatore per eseguire i trasferimenti e di automatizzare le procedure di migrazione in modo che possano essere eseguite automaticamente qualora se ne presenti la necessità.

Il passaggio a QEMU/KVM mostra come per realizzare la virtualizzazione delle risorse di rete si possa ricorrere a strumenti software di produttori diversi, tra cui anche programmi open-source, con la possibilità di passare abbastanza agevolmente da un hypervisor all'altro tramite appositi sistemi di conversione.

3.3 Esperimento 3: rete di servizi cloud

3.3.1 Descrizione e scopo dell'esperimento

In quest'ultimo esperimento lo scenario che è stato ricreato è quello di una rete di accesso virtualizzata all'interno della quale i vari servizi sia di commutazione che di fornitura di contenuti sono realizzati tramite macchine virtuali appositamente configurate.

In particolare la rete in questione prevede che un suo ipotetico utente possa connettersi direttamente ad un server di storage virtuale e ad un firewall. Attraverso quest'ultimo è possibile accedere ad sorta di *demilitarized zone* (DMZ) all'interno della quale si trovano un server web virtuale e un router virtuale che svolge anche le funzioni di NAT e di switch. Attraverso il router l'utente è infine in grado di accedere alla rete internet rappresentata in questo caso dalla presenza di un server video virtuale.

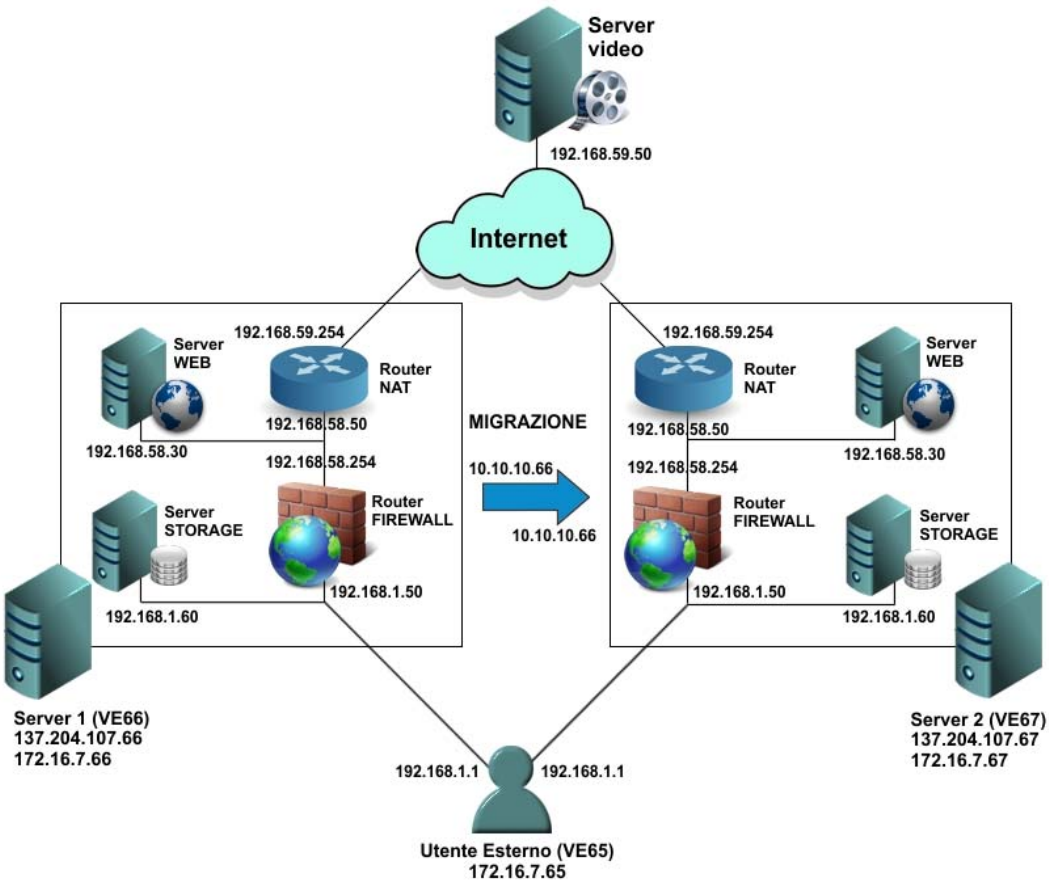

Figura 3.10: *Scenario del terzo esperimento*

L'intera rete di accesso virtuale deve poter essere migrata interamente a seconda delle necessità dell'operatore e dell'utente, senza che quest'ultimo percepisca alcuna interruzione nei servizi offerti dalla rete e nella connettività verso la rete internet, o almeno essendo questo praticamente impossibile che la discontinuità sia il più possibile contenuta

3.3.2 Creazione e configurazione delle macchine virtuali

Al fine di ottimizzare le prestazione della rete d'accesso virtuale che è stata creata prima di procedere con la realizzazione è stata fatta una scelta accurata delle distribuzioni di Linux da utilizzare per le varie funzionalità. Come anticipato nel secondo capitolo per le funzioni di server web e di server di storage è stato scelto CentOS, come firewall è stato utilizzato IpCop mentre per il router la distribuzione prescelta è stata ZeroShell.

In questo esperimento l'ipotetico utente esterno dei servizi forniti dalla rete è stato realizzato con una macchina virtuale su di un server esterno ai due utilizzati per simulare i due data center tra cui avvengono le migrazioni. Questa macchina è connessa tramite uno switch ad una delle interfacce di rete di ciascuno dei due server, a tale collegamento è dedicata la rete IP privata 172.16.3.0/24. Essa è accessibile tramite SSH dai server all'indirizzo 172.16.3.65.

Ovviamente anche in questo caso come negli altri è stato sufficiente creare e configurare le macchine virtuali solo sul primo server mentre quelle sul secondo server, che devono essere identiche per permettere le migrazioni, sono state ottenute ricopiando le stesse con la funzionalità importa/esporta.

Il server di storage è stato realizzato con una macchina virtuale eseguente una versione minimale di CentOS così come era stato fatto con i router degli esperimenti precedenti. A questa macchina come dotazione di hardware virtuale sono stati assegnati un processore single core, 512 MB di RAM, un hard disk virtuale dinamico ed una sola interfaccia di rete. Siccome tale server deve essere accessibile direttamente dal'utente esterno la sua interfaccia di rete è stata collegata in bridge alla scheda di rete fisica del server connessa con esso.

Il firewall è costituito da una macchina virtuale su cui è installato IpCop, sistema operativo specializzato per le funzioni di security. Come dotazione hardware possiede un processore single core, 256 MB di RAM, un hard disk virtuale dinamico e due interfacce di rete. Di queste ultime una è di tipo bridge e collega il firewall con l'utente esterno tramite l'interfaccia fisica del server connessa allo switch, l'altra è connessa ad un bridge virtuale (vboxnet2) che realizza la connettività con il server web e il router/NAT.

Il server web è una macchina virtuale basata su CentOS. La sua configurazione hardware è identica a quella del server di storage. L'unica differenza riguarda l'unica interfaccia di rete che invece di essere connessa con una delle schede di

rete fisiche del server è collegata al bridge virtuale vboxnet2 che connette il server web al firewall.

Il router che può svolgere anche le funzioni di NAT e di switch è realizzato con una macchina virtuale basata su ZeroShell, una distribuzione di Linux molto leggera pensata appositamente per le funzionalità di networking. La dotazione di hardware virtuale prevede un processore single core, 256 MB di RAM, un hard disk virtuale dinamico e due schede di rete. Di queste ultime una è connessa al firewall tramite il bridge virtuale vboxnet2 mentre l'altra è connessa ad un altro bridge virtuale (vboxnet3) con il quale il router si connette al server video che simula la connessione della rete di accesso ad internet.

Per simulare la connessione del router alla rete internet è stato scelto per motivi di semplicità di collegare ad esso un server virtuale il quale sia in grado di erogare un flusso video verso l'utente esterno. In particolare è stato scelto di riutilizzare per questo scopo il server video già creato per il secondo esperimento. L'unica modifica che è stato necessario apportare è stata il collegamento dell'interfaccia di rete della macchina al bridge virtuale vboxnet3 in modo da permetterne il dialogo con il router.

Per quanto riguarda gli indirizzi IP, al collegamento tra l'utente esterno, il server di storage e il firewall è stata assegnata la rete IP 192.168.1.0/24, in particolare all'utente esterno è assegnato l'indirizzo .1, al firewall l'indirizzo .50 mentre al server di storage l'indirizzo .60, il firewall è configurato come default gateway per gli altri due. Al bridge virtuale vboxnet2 tra il firewall, il router ed il server web è assegnata la rete 192.168.58.0/24, in particolare il firewall ha l'indirizzo .254, il router l'indirizzo .50 ed il server web l'indirizzo .40, il router ed il server web hanno come default gateway il firewall mentre quest'ultimo ha come default gateway il router. Infine al bridge virtuale vboxnet3 che realizza il collegamento tra il router e il server web è attribuita la rete IP 192.168.59.0/24 dove il router ha l'indirizzo .254 mentre al server video è assegnato l'indirizzo .50, il router costituisce il default gateway per il server video.

3.3.3 Esecuzione delle migrazioni

Per procedere con le migrazioni occorre come negli altri esperimenti innanzitutto configurare le macchine virtuali sul server di destinazione (VE67) per ricevere i teleporting:

```
[root@i2-ve067] VBoxManage modifyvm firewall --teleporter
on --teleporterport 2000 --teleporteraddress 10.10.10.67 --
teleporterpassword prova

[root@i2-ve067] VBoxManage modifyvm router-switch --
teleporter on --teleporterport 3000 --teleporteraddress
10.10.10.67 --teleporterpassword prova

[root@i2-ve067] VBoxManage modifyvm storage --teleporter on
--teleporterport 4000 --teleporteraddress 10.10.10.67 --
teleporterpassword prova

[root@i2-ve067] VBoxManage modifyvm web --teleporter on --
teleporterport 5000 --teleporteraddress 10.10.10.67 --
teleporterpassword prova
```

Una volta configurate, al loro avvio, queste macchine virtuali si mettono in attesa di ricevere le migrazioni le quali devono essere lanciate tramite gli appositi comandi dal server di origine (VE66):

```
[root@i2-ve066] VBoxManage controlvm firewall teleport --
host 10.10.10.67 --port 2000 --password prova

[root@i2-ve066] VBoxManage controlvm router-switch teleport
--host 10.10.10.67 --port 3000 --password prova

[root@i2-ve066] VBoxManage controlvm storage teleport --
host 10.10.10.67 --port 4000 --password prova

[root@i2-ve066] VBoxManage controlvm web teleport --host
10.10.10.67 --port 5000 --password prova
```

E’ importante notare che il server video non deve essere migrato in quanto, a differenza che nel caso precedentemente analizzato, in questo esperimento è utilizzato unicamente per simulare la presenza della rete internet a cui il router deve essere connesso. Il server video a tale scopo deve essere lasciato in esecuzione contemporaneamente su entrambi i server per assicurare la continuità della connessione del router con la rete esterna che esso simula.

3.3.4 Misure

Similmente a quello che è stato fatto nel secondo esperimento anche in questo caso le prove effettuate consistono nella verifica della connettività tramite ping e nella misurazione della banda utilizzata durante la trasmissione di un flusso video. In questo caso tali misure sono state condotte prendendo come punto di riferimento la macchina virtuale esterna che simula l'utente dei servizi della rete.

Innanzitutto è stata verificata la raggiungibilità durante la migrazione del server web lanciando ad esso dalla macchina virtuale esterna un ping con intervallo standard di un secondo, quello che ne è risultato è stata la perdita di un pacchetto.

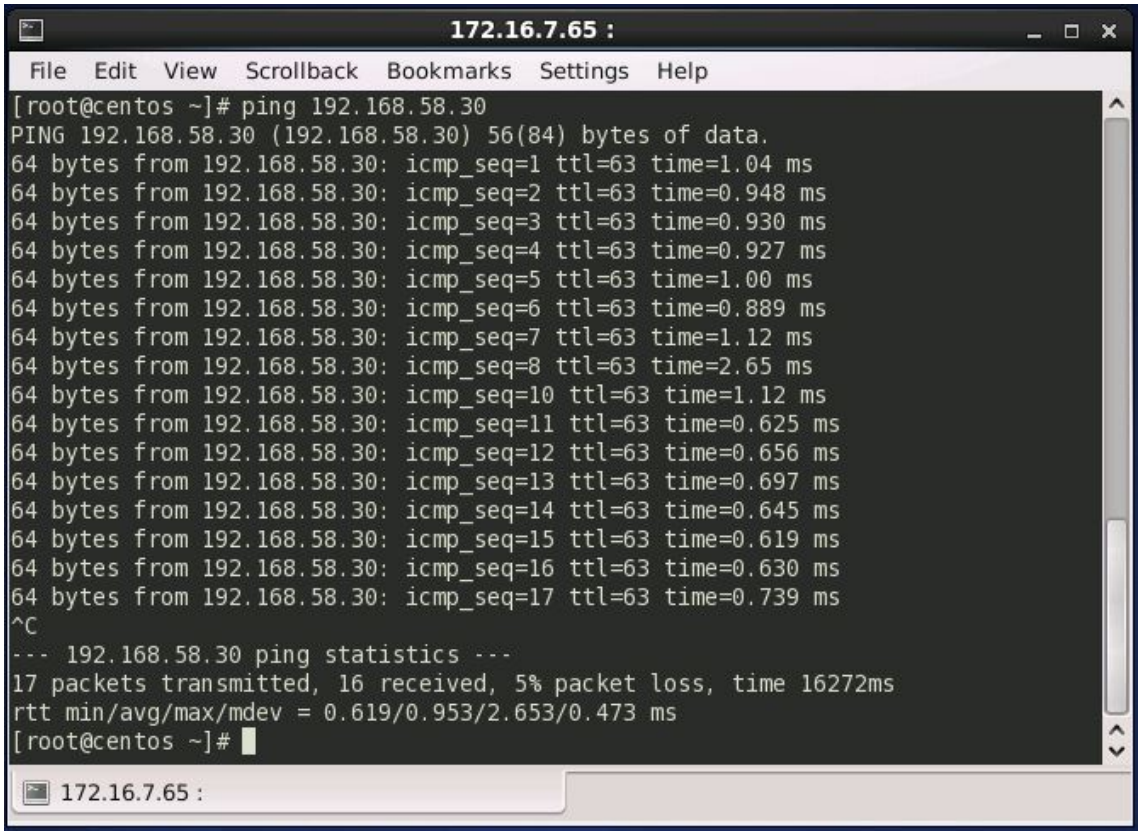

Figura 3.11: *Inviando dei ping dall'utente esterno al server web con intervallo di un secondo si nota la perdita di un pacchetto, in questo caso è il numero 10*

Successivamente si è voluto verificare la raggiungibilità durante la migrazione della rete, sempre tramite ping, del server video rappresentante la rete internet da parte dell'utente esterno. A tale scopo è stata per prima cosa fatta una prova lanciando un ping con intervallo standard di un secondo ed è stato possibile

notare la perdita di un solo pacchetto, il che ha permesso di concludere che la perdita di connettività è di circa un secondo.

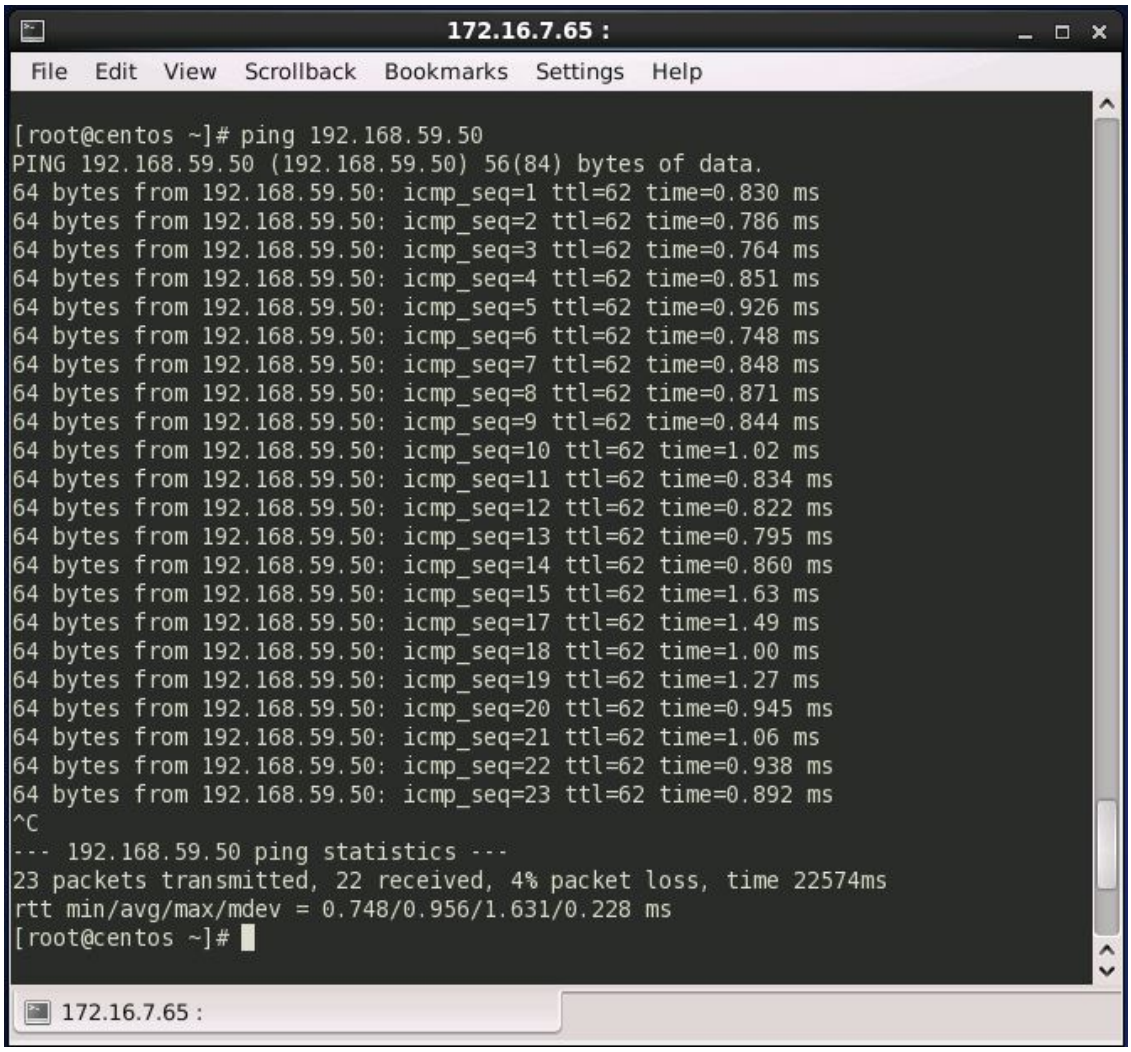

Figura 3.12: *Eseguendo il ping tra l'utente esterno e il server video con intervallo di un secondo si nota la perdita di un pacchetto, in questo caso il numero 16*

Per effettuare una stima più precisa si è deciso di lanciare nuovamente il ping tra le due macchine virtuali, questa volta specificando un intervallo di mezzo secondo tra i pacchetti. In questo caso si nota la perdita di tre pacchetti, quindi visto l'intervallo temporale scelto, si può concludere che la perdita di connettività è di circa un secondo e mezzo.

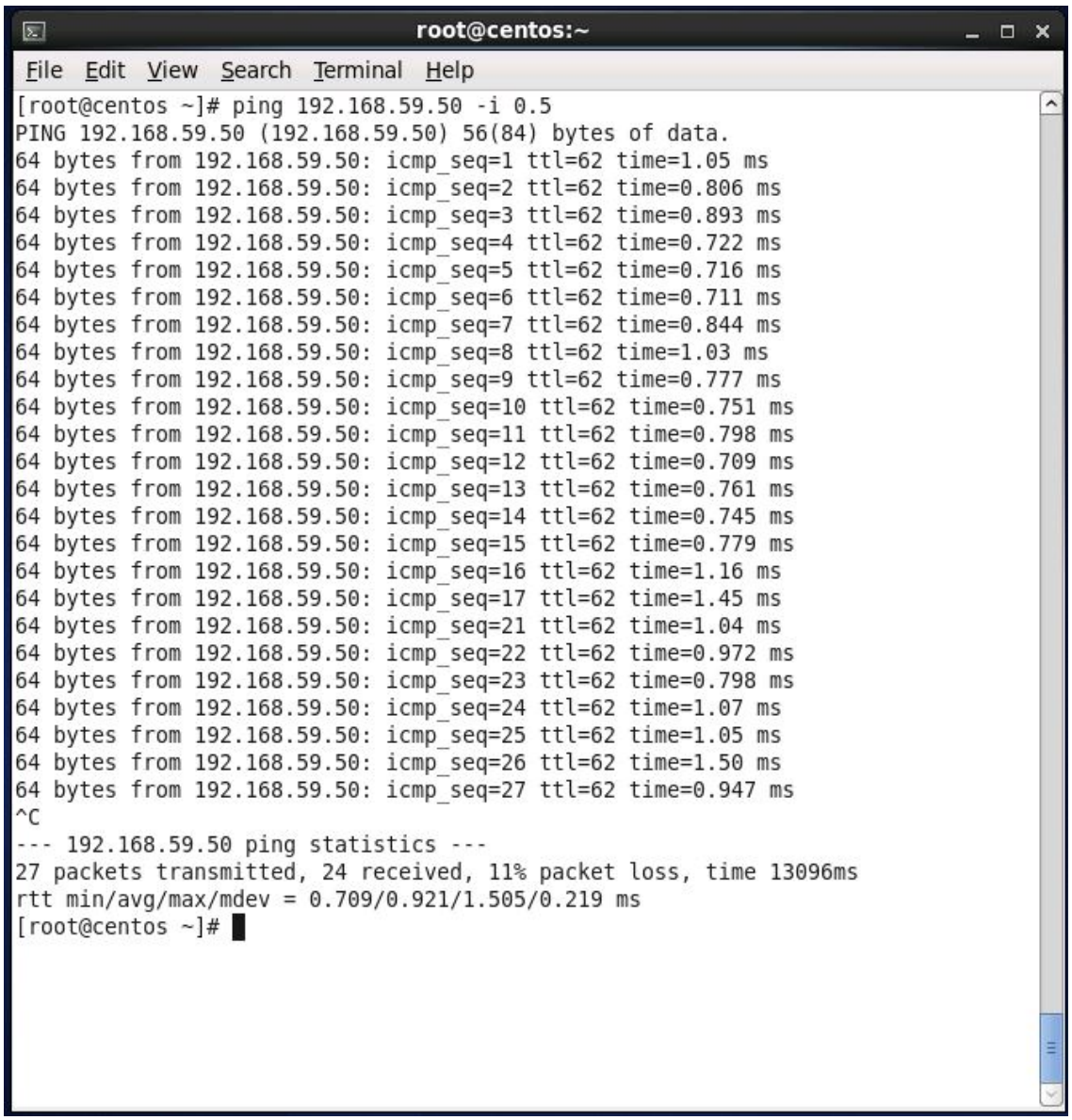

Figura 3.13: *Lanciando il ping tra utente esterno e server video con intervallo di mezzo secondo si nota la perdita di tre pacchetti, nel caso specifico sono i numeri 18,19 e 20*

Come è stato fatto precedentemente anche in questo caso per avere una misura più significativa si è effettuata una misurazione della banda utilizzata. Per fare ciò sul server video, su entrambi gli host contemporaneamente, è stato avviato un flusso video verso l'utente esterno. La scelta di tenere in esecuzione entrambi i server video su entrambi gli host è dovuto al fatto che esso, come già detto, in questo esperimento non fa parte delle risorse da migrare ma deve simulare la presenza della rete internet esterna la quale non deve variare nel corso delle migrazioni. In quanto in questo caso non era necessario analizzare il cambiamento di qualità del video è stato sufficiente lanciare un solo flusso video, tramite uno script:

```
#! /bin/bash
cvlc Hero_chap08_HQ.avi --sout
"#std{access=udp,mux=ts,dst=192.168.1.1:1234}"
```

Per effettuare la misura è stato possibile sfruttare lo script utilizzato nel secondo esperimento al quale sono state apportare le opportune modifiche:

```
#! /bin/bash

if [ $# -lt 3 ]
then
  echo "Usage: $0 <sampling period in seconds> <number of
delta estimation samples> <correction>">&2
  exit -1
fi

period=$1
deltasamp=$2
correct=$3
bwscale=$(echo "scale=6; 1/$period" | bc)

echo "Estimating delta with $deltasamp samples...">&2
delta=0.0
s=0
B1prev=0
B2prev=0
while [ $s -lt $deltasamp ]
do
  t=`./gettimestamp`
  B1=`iptables -nvx -L INPUT |grep .59.50 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B2=`iptables -nvx -L INPUT |grep .59.50 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  b1=$(echo "scale=6; ($B1-$B1prev)*8*$bwscale" | bc)
  b2=$(echo "scale=6; ($B2-$B2prev)*8*$bwscale" | bc)
  echo "$t  $b1 $b2">&2
  B1prev=$B1
  B2prev=$B2
  t1=`./gettimestamp`
  delta=$(echo "scale=6; ($t1-$t)+$delta" | bc)
  s=$(($s+1))
  #echo "$s: $delta"
done

echo "Done!">&2
delta=$(echo "scale=6; $delta/$s" | bc)
tsleep=$(echo "scale=6; $period-$delta-$correct" | bc)
echo "--- DELTA = $delta ----">&2
echo "--- SLEEP = $tsleep ----">&2
echo>&2

echo "# Sampling bandwidth every $period seconds">&2
```

```
echo "# Bandwidth scale factor = $bwscale">&2
echo>&2

echo "# time    UDP-1234-50 UDP-1235-50
echo

B1prev=`iptables -nvx -L INPUT |grep .59.50 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
B2prev=`iptables -nvx -L INPUT |grep .59.50 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
while true
do
  t=`./gettimestamp`
  B1=`iptables -nvx -L INPUT |grep .59.50 |grep "udp
dpt:1234" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  B2=`iptables -nvx -L INPUT |grep .59.50 |grep "udp
dpt:1235" |sed -r 's/[[:space:]]+/_/g'|cut -d_ -f3`
  b1=$(echo "scale=6; ($B1-$B1prev)*8*$bwscale" | bc)
  b2=$(echo "scale=6; ($B2-$B2prev)*8*$bwscale" | bc)
  echo "$t  $b1 $b2"
  B1prev=$B1
  B2prev=$B2
  sleep $tsleep
done
```

Il file timestamp necessario allo script è rimasto identico a quello del caso precedente mentre le regole di iptables che è stato necessario applicare perché lo script funzionasse sono in questo caso leggermente diverse in quanto cambia l'origine del flusso video:

```
iptables -I INPUT 1 -s 192.168.59.50 -p udp --dport 1234 -j
ACCEPT

iptables -I INPUT 2 -s 192.168.59.50 -p udp --dport 1235 -j
ACCEPT

iptables -I INPUT 3 -s 192.168.59.0/24 -j ACCEPT
```

Lo script è stato lanciato specificando un intervallo di mezzo secondo tra i campioni, un numero di campioni pari a duecento e un fattore di correzione nullo ed è stato indicato di salvare i risultati su un file:

```
./bwmeasurevideonew.sh 0.5 200 0 > cattura.dat
```

Dai risultati della misura si vede che gli istanti, spaziati di mezzo secondo ciascuno, in cui il valore della banda utilizzata rimane a zero sono tre, questo conferma il risultato che era stato ottenuto con il ping, cioè che la perdita di

connettività dovuta alla migrazione dura circa un secondo e mezzo. Dal file con i risultati della misura è stato ottenuto un grafico che mostra l'andamento della banda utilizzata nel tempo, in esso è possibile notare con chiarezza il momento in cui si ha la perdita di connettività a causa della migrazione e che durante il tempo rimanente la banda varia tra i 2 e i 4 MB/s.

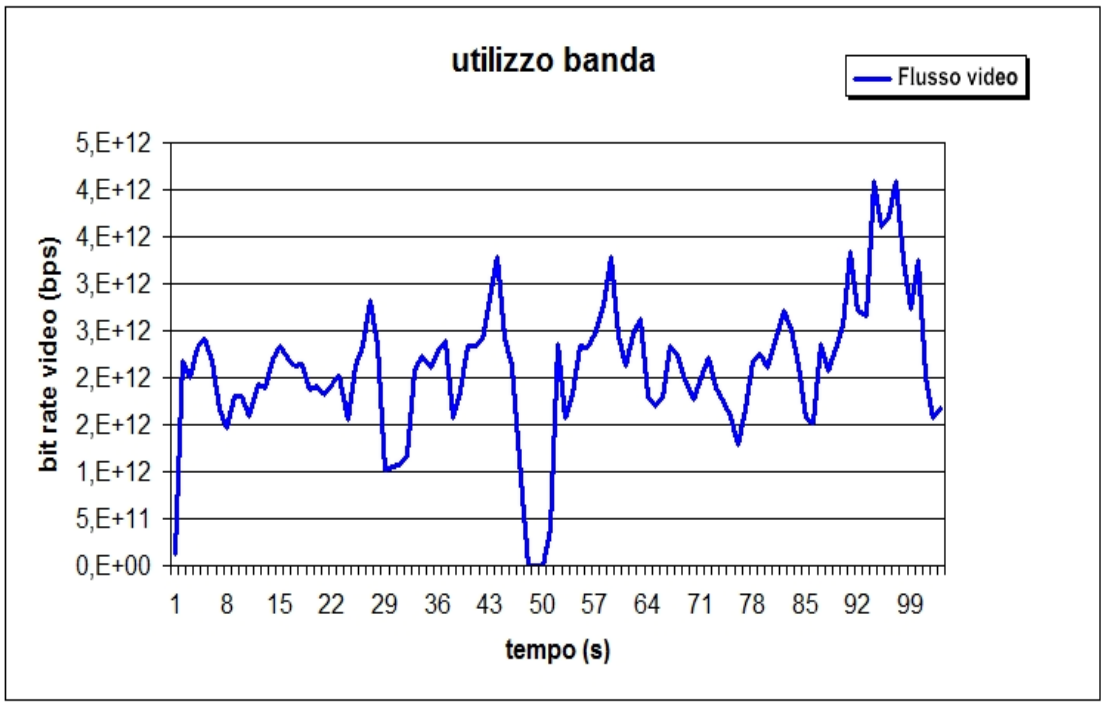

Figura 3.14: *Grafico rappresentante l'andamento della banda utilizzata nel tempo, è evidente nella zona centrale il momento in cui si ha la perdita di connettività dovuta alla migrazione nel quale il valore della banda utilizzata si annulla*

3.3.5 Conclusioni

Questo esperimento evidenzia come sia possibile realizzare una rete di accesso unicamente attraverso risorse virtualizzate, sia per quanto riguarda le funzioni di comunicazione come router, switch e firewall che per quanto riguarda i servizi di fornitura di contenuti come server web e di storage. Esso mostra inoltre come sia possibile migrare per intero questa rete virtuale tra un data center e un altro per seguire le necessità dell'operatore o degli utenti.

In particolare è possibile concludere che sebbene le singole migrazioni avvengano molto velocemente, si è sempre vincolati al fatto che esse possono avvenire sempre solo una alla volta, il che comporta nel caso ci si trovi ad avere a

che fare con una rete virtuale composta da molti elementi che il tempo necessario per trasferirla completamente aumenta in modo inevitabile. I problemi maggiori si hanno soprattutto quando più servizi virtuali necessitano uno dell'altro per funzionare, come ad esempio nel caso analizzato dove il router e il firewall sono essenziali perché le altre macchine possano erogare i loro servizi. Per questa ragione quando ci si trova a migrare una rete composta da molti elementi virtuali interconnessi conviene sicuramente migrare per primi quei componenti i quali sono necessari non solo per le funzioni che svolgono loro stessi ma anche per permettere il funzionamento degli altri elementi della rete. In poche parole le macchine virtuali che conviene migrare per prime sono quelle che realizzano le funzionalità di comunicazione come i router e gli switch.

3.4 Osservazioni e futuri sviluppi

3.4.1 Risultati finali

I tre esperimenti condotti durante questo lavoro di tesi mostrano come sia possibile sfruttare strumenti software già presenti sul mercato e già ampiamente collaudati in altri ambiti per realizzare delle reti di accesso nelle quali gli apparati fisici siano completamente sostituiti da risorse virtuali in esecuzione su server di tipo commerciale con tutti i vantaggi che ne derivano.

In modo particolare le prove eseguite mostrano come sia possibile sfruttare il meccanismo della migrazione delle macchine virtuali con riferimento alle risorse di reti virtualizzate e come questa funzionalità contribuisca ad accentuare i benefici già introdotti da questo nuovo approccio al mondo delle comunicazioni.

Le misure condotte durante gli esperimenti hanno mostrato come le funzionalità di migrazione già presenti nei software attuali permettano di eseguire dei trasferimenti in maniera molto rapida quando si tratta della migrazione di una sola macchina virtuale o di reti molto piccole, con perdite di connettività molto limitate, mente la situazione si complica quando ad essere migrate sono reti più estese che comprendono molte risorse virtuali.

La prima considerazione che è possibile trarre è che specie nel caso in cui si stia lavorando con delle reti virtuali piuttosto estese è consigliabile realizzare le risorse di rete attraverso delle macchine virtuali che siano il più snelle possibili, specie in termini di memoria RAM che risulta il fattore determinante nella velocità delle migrazioni. In particolare, quando possibile, è sempre consigliabile realizzare le funzionalità di rete con macchine virtuali che eseguano dei sistemi operativi creati appositamente per quei determinati scopi che risultano normalmente maggiormente ottimizzati e più leggeri di sistemi creati per scopi più generici.

Un altro punto importante da sottolineare è il fatto che quando ci si trova a dover migrare reti composte da molte macchine virtuali interconnesse tra di loro è bene analizzare in che ordine è più opportuno trasferirle in modo da minimizzare i disservizi. In modo particolare risulta innanzitutto preferibile trasferire tutte quelle risorse che non forniscono solo servizi fini a se stessi ma che sono necessarie anche al funzionamento delle altre, a questa categoria appartengono sicuramente i router virtuali, responsabili della connettività tra le varie macchine virtuali.

3.4.2 Ulteriori sviluppi

La continuazione naturale degli esperimenti condotti durante questo lavoro di tesi è senz'altro la creazione e l'analisi di reti virtuali più complesse composte da un numero di risorse maggiori che si avvicini di più a quella che potrebbe essere la realtà di una vera rete di accesso di un operatore di telecomunicazioni.

Bisogna poi osservare che gli esperimenti che sono stati descritti in questo capitolo sono stati realizzati sfruttando, per simulare i due data center tra cui avvengono le migrazione delle macchine virtuali, due server connessi tra di loro tramite un collegamento gigabit ethernet ad alta velocità con banda interamente libera e disponibile, questa rappresenta una situazione quasi ideale che difficilmente si avvicina ad uno scenario reale in cui le reti virtualizzate devono

essere trasferite tra data center diversi situati in posizioni remote all'interno della rete di un operatore di telecomunicazioni. Ciò che dovrebbe essere fatto in futuro per ottenere dei risultati che più si avvicinino al caso reale è la realizzazione di esperimenti di migrazione, simili a quelli che sono stati mostrati, ma nei quali si impongano delle limitazioni alla banda della connessione disponibile per le migrazioni in modo da analizzare quanto effettivamente questo influisca negativamente sulle prestazioni dei trasferimenti e più in generale di quanto si degradi la qualità dei servizi percepita dagli utenti della rete virtuale.

Altra importante questione che è stata evidenziata in modo particolare nel secondo esperimento e che meriterebbe di essere analizzata con più dettaglio è quella dell'automatizzazione delle operazioni di migrazione. Un primo passo in questa direzione potrebbe essere fatto estendendo il programma basato sul protocollo SIP utilizzato nell'esperimento appena ricordato in modo tale da poter automatizzare per tramite di esso non solo il trasferimento di macchine virtuali singole ma anche la migrazione di intere reti di accesso virtuali.

Conclusioni

La nascita e lo sviluppo di un gran numero di nuove tecnologie nel mondo dell'informazione, in particolar modo quelle legate ai servizi di comunicazione mobile e al cloud computing, hanno iniziato a mettere sempre di più in discussione il modello su cui si basa la rete internet, cioè il TCP/IP, evidenziandone tutti i limiti che esso comporta. In modo particolare quello che manca più di ogni altra cosa al modello attuale è la possibilità di fare dialogare le applicazioni con i servizi di rete, caratteristica indispensabile per poter fornire servizi di telecomunicazioni in grado di adattarsi in maniera automatica e in real time alle esigente degli operatori di rete e degli utenti.

La rete per queste ragioni negli ultimi tempi ha intrapreso un processo di cambiamento che la sta portando a modificarsi in maniera profonda. In particolare quello che sta avvenendo è il progressivo passaggio da uno scenario in cui i servizi di telecomunicazioni, sia quelli legati alle funzionalità di commutazione che quelli di fornitura dei contenuti, sono realizzati tramite appositi apparati hardware fisici ad un altro in cui essi sono costituiti interamente da risorse virtuali.

Gli esperimenti condotti nel corso di questo lavoro di tesi mostrano come già oggi sia possibile realizzare la virtualizzazione dei servizi di comunicazione offerti da una rete di accesso ricorrendo unicamente a strumenti software già presenti sul mercato e collaudati in altri ambiti e a risorse hardware di tipo standard. Le prove evidenziano anche i limiti principali dei software attualmente disponibili i quali dovranno essere risolti affinché la virtualizzazione degli apparati di rete possa essere sfruttata al massimo delle sue potenzialità.

Bibliografia

Testi e materiale di riferimento

[1] Antonio Manzalini, Roberto Minerva, Franco Callegati, Walter Cerroni, *Distributed Clouds at the Edge Networks,* Telecom Italia, UNIBO Dipartimento DEI, Cesena 2013

[2] ITU International Telecomunication Unit, *Definition of Next Generation Network,* http://www.itu.int

[3] Franco Callegati, Walter Cerroni, Aldo Campi, *Automated transport service management in the future internet: concepts and operations*, UNIBO Dipartimento DEIS, Cesena 2011

[4] Franco Callegati, Walter Cerroni, Aldo Campi, *Applications Scenarios for Cognitive Transport Service in Next-Generation Networks*, UNIBO Dipartimento DEI, CIRI-ICT, Cesena 2012

[5] *Software Defined Networking: The New Norm for Networks*, white paper, Open Network Fundation, Aprile 2012

[6] *Network Function Virtualisation*, white paper, ETSI-IGT, Ottobre 2012

[7] Sito ufficiale CentOS, http://www.centos.org

[8] *Oracle VM VirtualBox User Manual*, 2012 Oracle Corporation

[9] Sito ufficiale KVM, http://www.linux-kvm.org

[10] Umesh Deshpande, Xiaoshuang Wang, Kartik Gopalan, *Live Gang Migration of Virtual Machines*, Binghamton University, California 2011

[11] Christopher Clark, Keir Fraser et alii, Live *Migration of Virtual Machines*, University of Cambrige, University of Copenhagen, 2005

[12] Aldo Campi, SIP*: Session Initiation Protocol*, UNIBO Dipartimento DEIS, 2008

[13] Dr. Peer Hasselmeyer, *Follow-Me Cloud*, NEC Laboratories Europe, Germany Ottobre 2012

[14] Jordi Somoza de la Osa, *Cognitive Network Scenario through a SIP Proxy Server*, Unibo Diparimento DEIS, Cesena 2012