POLITECNICO DI MILANO

Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni

Dipartimento di Elettronica e Informazione

Evoluzione dei sistemi peer-to-peer video streaming

Relatore: Prof. Paolo Giacomazzi

Correlatore: Ing. Alessandro Poli

Tesi di Laurea di:

Marco Gasperi, matricola 708482

Anno Accademico 2009-2010

Alle persone a cui voglio bene

"L'unico vero maestro non è in nessuna foresta, in nessuna capanna, in nessuna caverna di ghiaccio dell'Himalaya... È dentro di noi."

Tiziano Terzani

Introduzione

Il lavoro di tesi presentato in questo elaborato esamina gli sviluppi nel campo del peer-to-peer video streaming allo scopo di migliorare la qualità percepita dall'utente. La rassegna di studi proposta analizza la tematica della distribuzione video attraverso reti peer-to-peer in modo globale focalizzando l'attenzione sulle soluzione e sulle proposte più innovative presenti nella letteratura scientifica. Il cuore dello studio proposto valuta, dopo aver esposto i principi genarali del peer-to-peer nell'ambito della distribuzione video, le soluzioni riguardanti le scelte topologiche attraverso l'impiego di soluzioni ibride, l'utilizzo di codifiche scalabili e tecniche di *network coding*, sistemi di rewarding e cooperazione tra gli utenti del sistema e tra i livelli di rete (overlay e underlay) per poi analizzare la tematica della sicurezza e della gestione dei diritti intellettuali. L'elaborato è concluso con la presentazione di un sistema ottimale che raggruppa ed unisce le migliori proposte analizzate all'interno di un prototipo teorico di possibile futura sperimentazione e realizzazione.

Ringraziamenti

Tante sono le persone a cui va un mio sentito ringraziamento per tutto l'affetto ed il sostegno che, giorno dopo giorno, ho ricevuto da loro. Alla mia famiglia, a papà Giacomo e mamma Domenica, alle mitiche sorelle Michela e Maria va il mio primo grande grazie: con tanto amore mi avete sempre sostenuto durante tutti questi anni di studio, mi avete dato la grande possibilità di frequentare l'università e vivere stupende esperienze. Grazie, grazie, grazie. Ai cari nonni, a tutti gli zii ai numerosi cugini un sentito ringraziamento.

Alla mia Simo va un grande bacio per tutto il supporto datomi, per i preziosi consigli, per il bellissimo tempo trascorso assieme e per tutto quello che ci aspetta. Ad una ragazza unica, tenace e molto molto dolce va un grazie di cuore.

Il mio percorso universitario è stato intervallato da due meravigliose esperienze: il viaggio in Australia e l'esperienza lavorativa a L'Aquila. Da entrambe ho imparato molto sia dal punto di vista umano che lavorativo.

Partendo dall'avventura oltreoceano un ringraziamento per tutto l'aiuto offertomi e per il meraviglioso tempo trascorso assieme va a Adrian Canclini e alla sua famiglia: senza di lui l'Australia non sarebbe stata così magnifica. Porterò sempre con me i ricordi di Sydney, della *farm* e di tutte le esperienze vissute assieme. Al *team* Australia si aggiungono il grande Enzo, Dina, Melissa, Maria, Danny, Simon, Henry, Helen, Rebecca, i tanti *backpakers*

incontrati durante il viaggio dei quali non ricordo più i nomi... ed il compagno d'avventure Wouter con il quale ho attraversato in largo e in lungo la terra di Oz a bordo della storica Nissan Pulsar.

Passando al *team* L'Aquila la mia gratitudine va a Renato per la possibilità offertami e per la fiducia ed il rispetto datomi sul lavoro. A Stefano e Simonetta, due persone eccezionali, con le quali ho trascorso bellissimi momenti tra lavoro e cenette in villa è rivolto un affettuoso ringraziamento. Alla Roby, a Fabiola, a Davide, a Teo, a Federico, agli "Stefani", a Sassà, a Diego... a tutto lo staff del Consorzio ForCase un sentito grazie.

E come dimenticare gli amici di sempre: Luca socio d'avventura e fedele conquilino, il super Kek boss dell'informatica sempre disponibile e cortese, Toni dottore dell'audio e compagno di pranzi milanesi, Caspa, Tara, gli ex della quinta liceo di Bormio in ricordo di tante avvenute trascorse assieme, tante risate e tanti tanti piacevoli momenti che ci hanno accompagnato e che ci aspettano ancora per esser vissuti !

Ringrazio i genitori di Simona Giorgio e Loredana per la loro generosa ospitalità ed auguro un grande bocca in lupo a Matteo e Lili.

Lato tesi un ringraziamento va al prof. Paolo Giacomazzi ed Alessandro Poli che durante questi mesi mi ha aiutato nella preparazione e stesura di questo elaborato.

Indice

Sommario **I**

Ringraziamenti **III**

1 Sistemi P2P Video Streaming **1**

1.1 Wikimedia e il P2P 4

1.2 Adobe e il P2P 6

2 Classificazione sistemi P2P: elementi fondamentali e topologie **10**

2.1 Elementi fondamentali 10

2.1.1 Struttura della distribuzione 12

2.1.2 Costruzione della topologia 13

2.1.3 Topologia 14

2.2 Topologia ad albero singolo e foresta 14

2.3 Topologia Mesh 17

2.3.1 GridMedia: esempio di architettura *push-pull* 20

2.4 Topologie ibride-miste 23

2.4.1 Hierarchical Ring Tree (HRT) 24

2.5 Mesh in struttura ad albero 28

2.5.1 mTreebone 28

2.5.2 SmartPeerCast 34
2.5.3 Algoritmo di scheduling del ricevente 38
2.6 MeTree 39
2.6.1 Principi di MeTree 39
2.6.2 Fase di join 41
2.6.3 Peer leaving 42
2.7 StreamComplete 42
2.8 CDN e P2P 46
2.8.1 LiveSky 48

3 Il Video digitale: principi e codifiche 55
3.1 Acquisizione video 56
3.2 Campionamento spaziale 56
3.3 Campionamento temporale 56
3.4 Lo spazio dei colori 57
3.4.1 RGB 58
3.4.2 YCbCr 58
3.5 La compressione video 61
3.5.1 Video Codec 63
3.5.2 GOP, I-Frame, P-Frame e B-Frame 65
3.6 Codifiche scalabili 66
3.6.1 H.264/SVC 66
3.6.2 MDC: Multiple Description Coding 68
3.7 Network coding 78
3.7.1 Aumentare il *throughput* in rete: esempio Butterfly . . 79
3.7.2 Fase di codifica-decodifica attraverso l'utilizzo di combinazioni lineari random 82

3.7.3 Applicazione dei network coding nella distribuzione video live . . . 84
3.7.4 R^2: Random Push con Random Network Coding nella distribuzione P2P live . . . 88

4 Sistemi di incentivo e cooperazione tra peers e livelli di rete 93
4.1 Score-based system . . . 95
4.2 Cooperazione tra livello di trasporto e aplicativo: il progetto NAPA-WINE . . . 96

5 La sicurezza nei sistemi P2P live video streaming 106
5.1 Autenticazione . . . 107
5.2 Confidenzialità . . . 110
5.3 Comuni attacchi . . . 111
5.4 SecureStream: sistema per protezione da attacchi . . . 116
5.5 D.R.M. - *Digital Right Managment* . . . 119
5.6 Considerazioni sulla sicurezza . . . 125

6 Sviluppo futuro e progetti in corso 126
6.1 Proposta di sistema ottimale . . . 126
6.1.1 Architettura generale . . . 127
6.1.2 Scelta topologica . . . 129
6.1.3 Scelta di coding e cifratura . . . 131
6.1.4 Monitoraggio e reward . . . 133
6.1.5 Considerazioni finali sul sistema proposto . . . 134
6.2 Sviluppi e ricerche in ambito accademico e commerciale . . . 135
6.3 Analisi SWOT . . . 139

7 Conclusioni 141

Bibliografia **144**

Capitolo 1

Sistemi P2P Video Streaming

Durante gli ultimi decenni la crescente evoluzione della rete Internet ha portato ad una sempre maggiore richiesta di contenuti multimediali rappresentati, ad esempio, dalla diffusione capillare di *User-Generated-Content (UGC) web site*, sistemi di *Video On Demand (VOD)* ed applicazioni per la trasmissioni in tempo reale di contenuti (*RTVB – Real time video broadcasting*). Tradizionalmente la distribuzione di contenuti video su internet si basa sul classico paradigma Client-Server nel quale il ruolo fondamentale viene svolto dal server centrale che distribuisce agli utenti a lui connessi il media richiesto. Questa visione semplicistica delinea chiaramente come la banda a disposizione del server centrale rappresenti il maggior fattore limitante per la scalabilità del sistema, nel numero di utenti che contemporaneamente possono fare richiesta per partecipare alla visione di un contenuto audio-video. Nel caso di accesso contemporaneo di 10000 utenti richiedenti un flusso video codificato a 512 Kbps sarà necessario predisporre lato server un collegamento con una banda di upload pari ad almeno 5 Gbps, richiesta che prevede un importante investimento per soddisfare una così massiccia richiesta di banda. Il paradigma Client-Server richiede quindi la predisposizione di collegamenti unicast

tra la sorgente distributrice del video ed il fruitore del contenuto: in presenza di un numero crescente di richieste il numero di collegamenti unicast sorgente-destinazione cresce proporzionalmente comportando notevoli consumi di risorse di rete e banda che possono portare ad un congestionamento dei links che trasportano lo stream video. Portare lo stesso contenuto video ad un gruppo di utenti induce a pensare all'utilizzo di un sistema multicast invece di singoli e dispendiosi collegamenti unicast IP. Il multicast a livello IP sembrerebbe essere la soluzione ideale per ottimizzare l'uso dei link nella distribuzione dello stesso contenuto a più *users*. Sfortunatamente tale soluzione non è stata sviluppata in modo adeguato per essere utilizzata nell'ambiente Internet: la mancanza di scalabilità, la difficoltà nello sviluppare tecniche di *congestion e error control* e la violazione del principio *stateless* sono solo alcuni fra i maggiori fattori tecnologici che ne hanno ostacolato lo sviluppo e l'utilizzo. Come analizzato nello studio [1] le cause del scarso utilizzo del IP multicasting (IP v4) sono da attribuire principalmente al fatto che tale soluzione è stata pensata e sviluppata senza avere in mente uno specifico campo di applicazione quale, ad esempio, lo streaming video. Questo ha portato alla mancanza di alcune caratteristiche fondamentali (sicurezza, protezione contro attacchi, gestione dei gruppi) che hanno fortemente limitato la diffusione dell'architettura IP multicast. Ulteriore fattore limitante è rappresentato dal grande *overhead* generato per la gestione dei gruppi e dalla complessità del controllo delle sezioni multicast. Un'altra via è stata intrapresa ed il problema è stato spostato dal livello IP al livello applicativo attraverso l'utilizzo del peer-to-peer (P2P) e la creazione di una rete overlay. Come riportato in [2] una rete overlay è definita come:

> *"Un livello applicativo virtuale che fornisce connettività, instradamento e messaggistica tra gli end-points che ne fanno parte.*

Le reti overlay sono spesso utilizzate per offrire servizi di routing non disponibili nelle sottostanti reti fisiche (underlay network). Molti sistemi peer-to-peer sono reti overlay costruite al di sopra di Internet"

Lo studio della costruzione delle reti P2P ha ricevuto molta attenzione da parte della comunità scientifica in quanto questa soluzione permette l'utilizzo di una efficiente infrastruttura che adopera in modo trasparente le risorse messe a disposizione dalla rete underlay. Le caratteristiche peculiari fornite dalle reti P2P sono riassumibili in:

- Condivisione di risorse: ciascuna entità contribuisce in modo attivo al sistema P2P
- Decentralizzazione: il comportamento del sistema P2P è determinato dalle azioni collettive che vengono eseguite dai singoli peers e non da un'unica entità centrale
- Equità: ogni peer gode dei stessi doveri-privilegi degli altri partecipanti al sistema. Solamente in certi casi si ricorre a peer con "funzioni speciali" per questo detti super-peer.
- Scalabilità: caratteristica fondamentale che permette la partecipazione contemporanea di varie entità al cui aumento il sistema risponde in modo opportuno.
- Auto-organizzazione: capacità da parte dei peers di gestire le operazioni di creazione-gestione della rete ed inoltro del contenuto senza la presenza di un'entità centrale.

Le reti overlay sono considerate come la più promettente infrastruttura per lo sviluppo di applicazioni distribuite ed i successi di svariate applicazione per il *file sharing*, per la distribuzione di voce (Skype), di contenuti audio-video, la crescita esponenziale di servizi di streaming multimediale e Video on Demand (Vod), sono solo alcuni esempi a prova delle enormi potenzialità offerte da tale tecnologia.

1.1 Wikimedia e il P2P

Un esempio molto interessante di utilizzo della tecnologia P2P nella distribuzione di materiale video è rappresentato dal forte interesse mostrato dalla fondazione Wikimedia [3] nell'utilizzo di questa tecnologia. La crescente quantità di materiale video inserito dagli utenti all'interno delle pagine della famosissima enciclopedia libera Wikipedia ha portato ad una richiesta di banda talmente elevata ed onerosa dal punto di vista economico da far rilasciare questa dichiarazione a Eric Moeller, vice-direttore dell'associazione Wikimedia: *"I costi sostenuti per l'acquisto di banda potrebbero in breve tempo prosciugare i fondi della fondazione o ridurre considerevolmente gli investimenti verso gli altri progetti e programmi. Per questo è estremamente importante rivolgere attenzione alle nuove tecnologie di distribuzione dei contenuti "*. La scelta tecnologica è ricaduta sull'utilizzo della piattaforma sviluppata dal consorzio europeo P2P-Next [4] che utilizza l'approccio peer-to-peer per la distribuzione di contenuti sia live sia on-demand attraverso l'impiego di un protocollo basato sui principi di BitTorrent [5]. Il principio fondamentale del P2P è, come già ribadito, la collaborazione attiva nella distribuzione del materiale e tale modo di pensare è alla base della filosofia di Wikipedia. Il P2P rappresenta un ottimo strumento per la diffusione del sapere utilizzando non

soltanto le risorse messe a disposizione di server centrali ma basandosi sulle risorse messe a disposizione da ciascun utente. L'algoritmo di live-streaming alla base di P2P-next è un'estensione del protocollo BitTorrent nel quale sono stati introdotte innovazioni tali da garantirne l'utilizzo in un contesto live. Lo studio [7] evidenzia l'approccio necessario nel trasformare BitTorrent da un protocollo per la distribuzione di contenuti in ambienti non soggetti a vincoli temporali ad uno *live*. Sostanzialmente l'approccio utilizzato consiste nel dare alta priorità ai blocchi video che stanno per esser riprodotti dal *player*, nell'effettuare il download prioritario dei pacchetti rari ed abilitare la ricezione di pacchetti parallelamente. Lo stesso inventore del protocollo BitTorrent Bram Cohen ha dichiarato in un'intervista del 19 gennaio 2011 [7] di esser al lavoro per lo sviluppo di un nuovo protocollo p2p-live in grado di rivoluzionare il mondo della distribuzione live video così come è stato fatto dieci anni fa nel modo del *file-sharing*. Nello specifico caso di Wikimedia la riduzione del costi di distribuzione avviene attraverso l'utilizzo del player Swarmplayer [8] che viene distribuito sotto forma di *plug-in* per browser web e, per la prima volta, un protocollo P2P video live è inglobato all'interno della navigazione web attraverso l'utilizzo di HTLM5. Il player permette di impostare il valore della banda di upload che si è disposti a condividere e, nel caso in cui l'utente rifiuti la condivisione, la rete P2P sarà ugualmente utilizzata solamente per la fase di download. L'utilizzo di tale tecnologia risulta essere ancora in una fase di test ed il passaggio alla sistema P2P avverà in modo regolare dopo aver potuto analizzare i dati provenienti dalla fase di sperimentazione. Per questo i contenuti multimediali presenti in Wikimedia sono ancora attualmente ospitati presso i *web severs* e, nel caso in cui il contenuto non sia disponibile all'interno della *community* P2P esso verrà scaricato in maniera tradizionale. Una volta installato il player alla domanda

di visione del contenuto video viene inviata una richiesta per la ricezione del file .torrent dal server URL2Torrent.net e sarà quindi possibile ricevere il file richiesto dalla rete P2P come mostrato in figura 1.1. Se si ha accesso al vi-

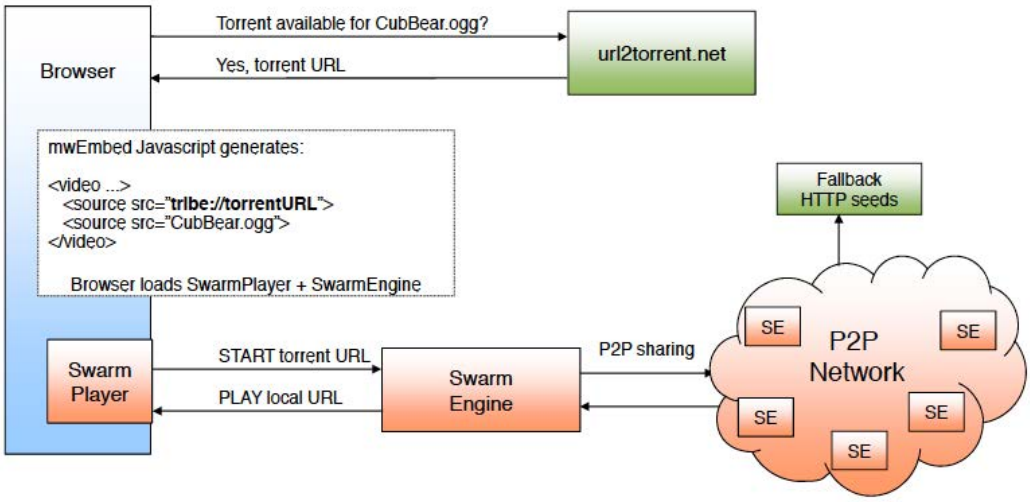

Figura 1.1: Schema di funzionamento scambio messaggi

deo per la prima volta ed il file .torrent non è ancora stato creato si da inizio alla così detta procedura di *auto-torrentization*: il primo visitatore effettua il download del video dal server web mentre i visitatori successivi richiedenti lo stesso contenuto utilizzeranno la procedura precedentemente descritta in quanto il file .torrent sarà ora disponibile. Il caso di WikiMedia rappresenta un interessante scenario: i dati e le informazioni che si raccoglieranno nella fase sperimentale mostreranno quanto il P2P rappresenti uno strumento flessibile, scalabile e molto vantaggioso dal punto di vista economico.

1.2 Adobe e il P2P

L'interesse nell'utilizzo della tecnologia P2P è fortemente evidenziato dalle grandi società software quali Adobe. Kevin Towes, Product Manager di

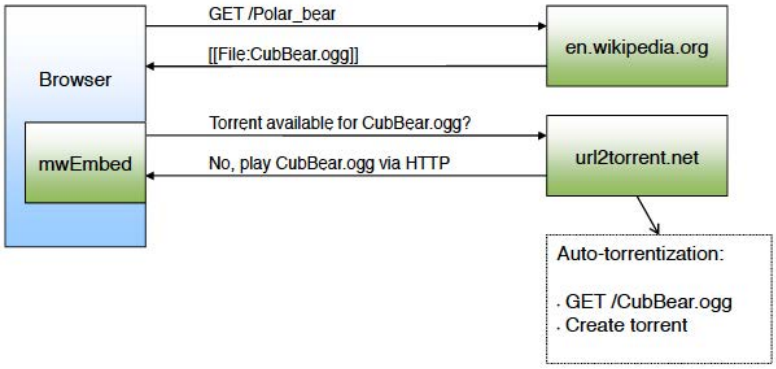

Figura 1.2: Scambio di messaggio con primo utente in Swarm Player

Adobe Flash Media Server, in un'intervista rilasciata a Beet.Tv [9] presenta l'introduzione della tecnologia P2P all'interno del player Flash 10.1 con le seguenti parole: *" L'utilizzo di questa tecnologia cambierà radicalmente il modo di pensare la distribuzione dei contenuti in rete. Il grande vincolo rappresentato dai costi della banda che, in alcuni casi raggiunge il 75% del costo totale della distribuzione del video, potrà considerevolmente essere abbassato se non addirittura eliminato"*. I laboratori di Adobe stanno implementando lo sviluppo della tecnologia P2P all'interno del popolare player Flash attraverso l'utilizzo di Cirrus [10] piattaforma in grado di supportare sia video streaming live sia video on-demand. Il principio del P2P nell'utilizzo della distribuzione di contenuti live non viene limitato al solo video ma utilizzato in un contesto più ampio che raggruppa le varie forme della comunicazione in tempo reale come, ad esempio, il gioco multiplayer on-line. Cirrus svolge il ruolo di gestore delle comunicazioni tra i peers partecipanti svolgendo in sostanza le stesse funzionalità che un *server tracker* svolge per BitTorrent. La privacy ed il controllo sull'utilizzo della banda di upload di ciascun utente viene garantita attraverso l'accettazione di un messaggio da parte del player Flash che informa l'utente dell'utilizzo del P2P come mostrato in figura 1.3

Se l'utente non vuole mettere a disposizione la propria banda di upload il

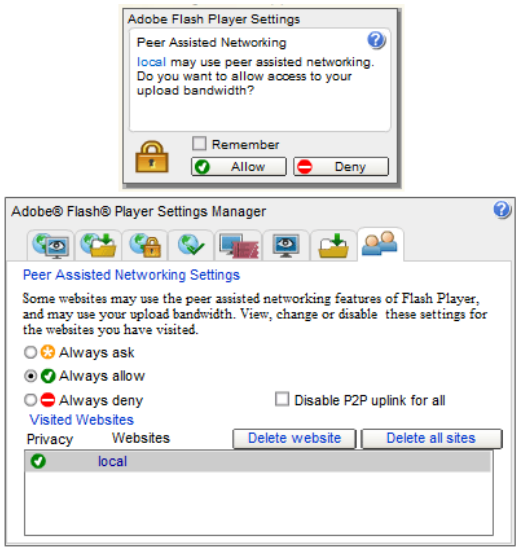

Figura 1.3: Finestre di messaggi di accettazione condivisione banda in Flash Player 10.1

contenuto richiesto potrà essere distribuito in maniera tradizionale, in qualità ridotta o non essere visibile in base alle scelte strategiche decise dal fornitore del video. Il network televisivo CNN per la trasmissione sul proprio portale web del messaggio inaugurale del presidente Obama (25 milioni di utenti connessi al sito cnn.com) ha utilizzato a supporto della distribuzione tradizionale una tecnologia P2P implementata dalla compagnia danese Octoshape sempre utilizzando come player Adobe Flash. L'utilizzo della tecnologia P2P che, secondo [11] ha contribuito a ridurre il carico di traffico sui server centrali del 30%, ha permesso la diffusione di quello che può esser considerato il più grande evento live streaming mai trasmesso.

E' molto interessante osservare come l'impiego del P2P sia uno strumento talmente potente da attirare investimenti ed attenzioni sia da grandi distributori di informazioni sia da *community* indipendenti che, utilizzando soluzioni

opensource come quella impiegata da WikiMedia, sono in grado di distribuire informazioni verso un numero sterminato di utenti cooperanti tra di loro.

In questo capitolo sono stati introdotti i principi fondamentali del peer-to-peer e, attraverso la descrizione dell'impiego di tale tecnologia in due ambiti totalmente diversi (WikiMedia fondazione senza scopi di lucro vs. Adobe impero commerciale) si è voluto dimostrare come l'interesse verso questa forma di distribuzione dei contenuti sia universale e rivesta un importante abito di ricerca e sviluppo.

Capitolo 2

Classificazione sistemi P2P: elementi fondamentali e topologie

2.1 Elementi fondamentali

La rete P2P unisce tra di loro nodi cooperanti chiamati peer che distribuiscono e ricevono dati attraverso l'utilizzo del multicast a livello applicativo. Nelle soluzioni P2P per la distribuzione di contenuti audio-video lo stream, diviso in segmenti di piccole dimensioni chiamati *chunks*, viene inoltrato dai peers verso gli altri peers del sistema utilizzando il livello applicativo costruito al di sopra di un'infrastruttura già esistente che non ha necessitato di cambiamenti strutturali per poter esser impiegata. Il P2P ha guadagnato un'enorme popolarità all'interno del settore del *file sharing* attraverso la diffusione capillare di applicazioni come eMule, BitTorrent, Kazaa. E' comunque molto importante sottolineare che il passaggio del sistema P2P dal *file sharing* alla distribuzione di contenuti video live comparta molte sfide e

problematiche da affrontare. La natura del video stesso impone vincoli stringenti al sistema in termini di ritardo *end-to-end, jitter, startup time* tutti fattori caratteristici di un'applicazione *real-time* dove la perdita di pacchetti o il ritardo nell'arrivo degli stessi comporta una degradazione della qualità video ricevuta dal fruitore.

Per far si che il P2P video streaming diventi una tecnologia sempre più utilizzata per la distribuzione video è necessario fronteggiare i problemi derivanti dalla scarsa disponibilità di banda in uplink dei peers spesso connessi alla rete internet attraverso collegamenti asimmetrici quali l'ADSL. Inoltre i fruitori-distributori del servizio possono in qualsiasi momento, in modo inaspettato abbandonare il sistema troncando la distribuzione del flusso verso i peers con i quali erano connessi. Diversamente dalla struttura client-server molto spesso i pacchetti dati vengono inoltrati attraverso vari link, seguendo percorsi che possono introdurre ritardi addizionali soprattutto in presenza di collegamenti congestionati da forte traffico.

Esistono vari sistemi peer-to-peer classificati in maniera diversa in base a come essi si adoperano a costruire la rete overlay e a come vengono gestiti i peer che ne fanno parte. Le entità logiche che compongono un sistema peer-to-peer possono essere così elencate:

- Peer: elemento fondamentale rappresentato dalla macchina utente. Esso non svolge più soltanto un ruolo passivo (come fatto dal client tradizionale) ma partecipa in modo attivo alla distribuzione del contenuto in rete.

- Storage server: sorgente nella quale è memorizzato il contenuto da distribuire.

- Web server: sever attraverso il quale è possibile consultare l'elenco dei

contenuti richiesti e grazie al quale è possibile ricevere informazioni riguardo allo startup server.

- Startup server: server che si occupa della gestione della rete peer-to-peer. Tali funzioni variano molto in base al sistema utilizzato. Fra le più comuni troviamo: mantenimento lista peer attivi, mantenimento lista contenuti, controllo della topologia.

La classificazione di un sistema peer-to-peer avviene tenendo in considerazione le caratteristiche principali che ne identificano la natura. Di seguito sono elencate e descritte brevemente le principali soffermandosi in particolar modo sulla descrizione delle topologie ibride (albero-anello, albero-mesh) che rappresentano ad oggi uno fra i maggior campi d'interesse in questo contesto. Principali elementi:

- Struttura della distribuzione
- Costruzione della topologia
- Topologia

2.1.1 Struttura della distribuzione

In base al numero di peer che partecipano in modo attivo alla distribuzione del contenuto si individuano due tipologie di distribuzione:

- End-to-end overlay: Ogni client (end–host) gestisce la tipologia della distribuzione e scambia il contenuto con gli altri peer facenti parte della rete.
- Proxy-based overlay: viene creata una struttura gerarchica in cui solo alcuni nodi, i multicast node, vengono utilizzati per la distribuzione e

questi creano tra di loro un *backbone overlay*. La maggioranza dei peer riceve solamente il contenuto senza inoltrarlo, sono quindi delle foglie collegate alla dorsale di distribuzione.

Il vantaggio principale nell'utilizzo di una soluzione proxy-base è rappresentato dal basso traffico di overhead necessario per la gestione topologica poiché essa è limitata solamente al sottoinsieme dei multicast node. Un esempio di questa struttura si trova nella combinazione di reti CDN-P2P come riportato in [12]. Nei sistemi end-to-end è invece necessario uno scambio maggiore di informazioni tra peer poiché la rete è totalmente distribuita, senza una struttura specifica che la controlli dall'alto. Sistemi end-to-end risultano essere adatti in situazioni in cui la topologia della rete varia velocemente.

2.1.2 Costruzione della topologia

In base alla modalità con cui vengono gestiti i peer, organizzate le relazioni tra di loro e costruita la topologia di distribuzione tre classi vengono definite:

- Diretta: il nodo quando entra nel sistema viene assegnato direttamente ad un'entità chiamata *parent* dalla quale riceve il contenuto. Sistema frequentemente utilizzato nelle topologie ad albero.

- Indiretta a maglia: i peer scambiano fra di loro informazioni sulla loro posizione e sulla loro disponibilità di contenuti. Utilizzando queste informazione costruiscono la rete overlay.

- Indiretta a cluster gerarchici: i nodi vengono divisi gerarchicamente, raggruppati in cluster e successivamente vien creato un albero di distribuzione che connette tutti i livelli gerarchici.

2.1.3 Topologia

Il modello preso in considerazione per la distribuzione dei contenuti video è il multicast: una sorgente e un numero non precisato di utenti intenzionati a fruire del contenuto. Tre sono le principali topologie di architetture overlay utilizzate nello streaming peer-to-peer:

- Topologia ad albero/foresta
- Topologia mesh
- Topologie ibride-miste

2.2 Topologia ad albero singolo e foresta

Gli utenti che partecipano ad una sezione di video streaming, in modo analogo a quello che vien fatto a livello di rete da IP multicast, formano a livello applicativo un albero che li connette fra di loro e che distribuisce il contenuto video proveniente dalla sorgente disposta al livello più alto. Ciascun peer entra a far parte dell'albero ad un determinato livello, riceve il contenuto dal nodo gerarchicamente a lui superiore (*parent*)e lo inoltra verso i suoi figli a livello inferiore, contribuendo in modo attivo alla distribuzione del contenuto. Dati un insieme di peer i fattori fondamentali che vengono tenuti in considerazione al momento della creazione dell'albero riguardano la profondità in livelli dell'albero stesso ed il numero di figli che un singolo nodo può avere. Per ridurre il ritardo percepito dai peer che si trovano nei livelli inferiori bisogna cercare di ridurre al minimo la profondità dell'albero condensandolo in pochi livelli. Per far questo bisogna associare il numero massimo di peer figli al padre compatibilmente alle proprie capacità di upload. Un altro fattore

molto importante da tenere in considerazione riguarda il mantenimento dell'albero: gli utenti all'interno di una sezione streaming sono molto dinamici e si assiste a nuovi ingressi (*join*) ed a uscite dal sistema (*leave*). Gli abbandoni di più utenti contemporaneamente possono avvenire in maniera inaspettata e questo provoca l'interruzione del flusso video per tutti quei nodi che ricevevano il contenuto dai nodi usciti di scena. Per minimizzare l'interruzione di flusso devono essere adottati dei sistemi che permettano la ricostruzione dell'albero e la riconnessione di tutti i nodi. Questi processi possono essere gestiti in modo centralizzato da un server che monitora la costruzione e la riparazione dell'albero oppure in modo distribuito attraverso algoritmi distribuiti. Esempi di sistemi basati su topologia ad albero son ESM, Chainsaw, Nice, ZigZag. Dal singolo albero di distribuzione si giunge ad una struttura

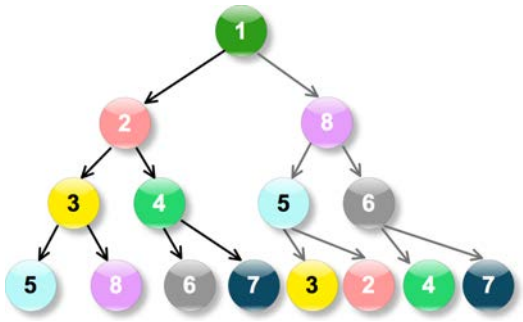

Figura 2.1: Esempio di topologia a foresta: il contenuto originale è diviso in due flussi distribuiti su due alberi indipendenti. I peer sono nodi foglia in un albero e nodi interni nell'altro.

di multi-albero, chiamata topologia a foresta, dove il singolo stream video viene diviso in più sub-stream e ciascuno di essi viene distribuito attraverso un singolo albero. Ciascun peer, per ricevere tutti i sub-stream, deve essere associato a tutti gli alberi componenti la foresta e questa associazione può avvenire a livelli diversi a seconda dell'albero preso in considerazione.
In figura 2.1 e' riportato un esempio nel quale un flusso video viene diviso

in 2 sub-strem e distribuito a 7 peer. La sorgente diffonde i 2 sub-strem rispettivamente nel ramo destro e sinistro: i peer 2 – 3 e 4 ricevono e distribuiscono contenuti nel ramo sinistro mentre sono foglie nel ramo destro. Il peer 7 risulta esser un nodo foglia in ambo i rami e quindi non contribuisce in alcun modo alla diffusione del contenuto. In un approccio basato sulla costruzione di una topologia ad alberi multipli o foresta ciascun peer partecipante viene organizzato all'interno di più alberi multipli in base alle proprie capacità in termini di banda di uplink. Lo scopo fondamentale di questa organizzazione consiste nella capacità del sistema nel resistere ai così detti *peer churn*, situazione nella quale molti peer abbandonano il sistema di distribuzione. L'utilizzo di codec MDC (*Multiple description code*) permette la distribuzione di descrizioni indipendenti attraverso il corrispettivo albero. Adottando un semplice meccanismo di tipi *push* ciascuno nodo inoltra i pacchetti ricevuti verso i nodi figli. Nell'utilizzazione di un sistema ad albero il problema fondamentale è rappresentato dalla costruzione stessa dell'albero che deve esser fatta in maniera tale da garantire alberi multipli bilanciati, stabili e corti. Come riportato nello studio [13] tali caratteristiche possono essere attuate attraverso alcuni accorgimenti: il mantenimento di alberi bilanciati avviene posizionando ciascun peer come nodo interno di un solo albero e come nodo foglia nei restanti alberi. Per mantenere la popolazione di nodi interni equilibrata un nuovo nodo che entra nel sistema è assegnato all'albero che possiede il minor numero di nodi interni ed il vincolo degli alberi corti viene rispettato assegnando i nuovi peer a quei nodi con minore profondità, in altre parole a quei nodi che si trovano maggiormente vicini alla sorgente informativa. Quando un nodo interno abbandona l'albero il sotto-albero formato dai suoi figli resta isolato, il sistema risulta esser frazionato ed i peer orfani devono attivarsi alla ricerca di un nuovo nodo parent.

Il compito di trovare un nuovo nodo fornitore viene preso dal nodo radice del sotto-albero orfano: se questo, entro un arco di tempo prestabilito non è in grado di trovare un fornitore, autorizza i singoli peer orfani ad attuare le procedure di rejoin individualmente.

Esempi di sistemi commerciale che utilizzano la topologia a foresta sono rappresentato da SplitStream, CoopNet e THAG. Per dare stabilità, la struttura è composta da una foresta di alberi che si occupano di parti indipendenti del flusso. Il principio fondamentale sta nel fatto che il video trasportato dispone una codifica tollerante alle perdite, in modo tale che se dovesse verificarsi una situazione in cui un peer si trovasse disconnesso da un numero limitato di alberi, potrebbe comunque visualizzare il video proveniente dagli altri in modo accettabile. Per questa ragione è auspicabile distribuire video codificato con un codec a descrizione multipla (MDC), in modo da distribuire le singole descrizioni su alberi disgiunti, e in caso di disconnessione da un albero permettere la riproduzione del contenuto proveniente dalle restanti descrizioni.

2.3 Topologia Mesh

La topologia mesh non impone il confinamento dei peer all'interno di una struttura rigida bensì le connessioni tra i nodi sono stabilite e rilasciate in base alla disponibilità di materiale-banda dei peer stessi. I peers sono connessi ad un insieme di altri peers con i quali periodicamente vengono scambiate informazioni riguardanti la disponibilità dei contenuti e alla loro localizzazione. Il contenuto video viene richiesto da un nodo a quelli vicini che lo possiedono e, grazie al fatto che tale contenuto può essere ricevuto da più sorgenti allo stesso tempo, i sistemi mesh sono molto più robusti alle situazioni di *peer*

churns.

Creare e mantenere una struttura mesh comporta una serie di operazioni che vengono nel seguito brevemente elencate:

- Contattare il *tracker*: il peer comunica le proprie informazioni base come indirizzo IP e *port number*. Il tracker risponderà inviando una lista di nodi attivi partecipanti alla sezione.

- Contattare i vicini: il peer inizia ora a contattare i vicini contenuti nella lista e, se la connessione è accettata dall'altro peer, viene inserito all'interno della lista dei vicini. Una volta raggiunto un numero sufficiente di vicini si avvia lo scambio di informazioni.

- Aggiornamento dei vicini: poiché si è in presenza di una situazione in cui si assiste a svariati *join* e *leave* la lista dei vicini deve esser aggiornata per far si che si ottimizzi al meglio l'invio e la ricezione del video. Tale aggiornamento viene fatto attraverso l'invio da parte dei peer stessi di messaggi *keep-alive*.

- Abbandono della sezione: se il peer esce dal sistema in maniera concordata esso invia un messaggio di notifica al *tracker* che potrà quindi eliminarlo dalla lista di nodi attivi. Se invece l'abbandono avviene in modo improvviso i suoi vicini capiranno dell'abbandono attraverso i messaggi di *keep-alive* e l'informazione della scomparsa del peer dalla rete verrà diramata agli altri nodi attraverso i periodici scambi di *neighbors-list*.

Due sono i modi per distribuire i contenuti attraverso le connessioni create fra i peer:

- *Push:* in un sistema mesh-push il peer appena riceve un segmento lo inoltra ciecamente a tutti i suoi vicini ma questo può provocare inutile

spreco di banda in quanto il frammento di video inoltrato potrebbe già esser stato ricevuto dal peer da un'altra fonte.

- *Pull*: attraverso lo scambio periodico di *buffer-maps* i nodi sono a conoscenza della disponibilità di contenuti presso i loro vicini e con l'invio di un messaggio richiedono il contenuto. Lo scambio di *buffer-maps* ed i messaggi di segnalazione aumentano il traffico all'interno della rete andando a creare situazioni che provocano ritardi nella ricezione dei segmenti video con conseguente deterioramento della qualità percepita.

A questi due classici sistemi di distribuzione se ne affianca un altro denominato *swarming content delivery*: esso unisce il sistema *push* e quello *pull*. La prima metodologia viene utilizzata per la distribuzione di informazione di disponibilità, la seconda per le richieste di materiale. Come dimostrato nello studio [14] la metodologia *push-pull* è molto più efficiente in termini di *overhead* e porta ad una diminuzione considerevole in termini di ritardo di *playback* perché elimina la necessità, imposta dalla metodologia *pull*, dello scambio di *buffer-maps* e traffico di segnalazione. Così come per la precedente topologia la creazione dell'albero è essenziale, di fondamentale importanta è l'impiego di un algoritmo di *sheduling* nella topologia mesh. L'algoritmo di *scheduling* per la gestione della distribuzione dei pacchetti deve esser in grado di rispondere a queste richieste: utilizzare in modo adeguato la banda a disposizione di ciascun *peer parent*, gestire in maniera consona il numero di descrizioni (indice della qualità video trasmessa) ed assicurare una consegna dei pacchetti entro determinati ritardi stabiliti. L'andamento della consegna di un singolo pacchetto all'interno della rete mesh è fortemente determinato dal comportamento globale della rete stessa, da come le altre entità partecipanti sono gestite: la creazione di una topologia overlay performante e

l'utilizzo di uno *scheduling* intelligente sono i punti di forza per una corretta distribuzione del video.

In Prime [15], primo sistema mesh ad implementare un sistema di distribuzione di tipo *swarming*, viene utilizzato un algoritmo di scheduling che agisce nella seguente maniera: ciascun peer mantiene due entità informative nei confronti dei propri fornitori riguardanti la disponibilità di informazioni e la banda a disposizione. Attraverso il monitoraggio della banda ciascun peer è in grado di adattare il numero di richieste nei confronti dei propri *parents*. Lo scheduler individua i pacchetti che da più tempo sono disponibili presso il parent e ne invia le richieste proporzionalmente alla banda a disposizione. Per bilanciare il carico di traffico nella rete in presenza di stesso contenuto presso più fornitori il peer selezionerà come sorgente quello che presenta il minor utilizzo di banda.

2.3.1 GridMedia: esempio di architettura *push-pull*

Lo studio [16] esamina le prestazioni ottenute da GridMedia sistema creato per la distribuzione su larga scala di contenuti video live attraverso l'impiego del metodo *push-pull*. Ciascun nodo all'interno del sistema indipendentemente sceglie dei vicini con cui scambiare i contenuti video e, come mostrato dalla campagna di esperimenti condotti su PlanetLab [17] il metodo push-pull utilizzato in GridMedia porta a considerevoli risultati in termini di miglioramento dei ritardi *end-to-end*, riduzione di traffico *overhead*, pacchetti correttamente ricevuti anche in situazioni complesse, caratterizzate da elevati cambiamenti topologici dovuti ad ingressi-uscite dal sistema. Nel seguito vengono riportati i risultati ottenuti dallo studio. La parte di traffico generata da ciascuno nodo per il controllo e la gestione del traffico nella rete viene definita come *control-overhead*. Il traffico di controllo è generato dalla

necessità di inviare in rete dei pacchetti e dei messaggi che siano in grado di coordinare lo scambio di buffer-maps, i messaggi di richiesta pacchetti, i messaggi di presenza per sondare l'effettiva presenza di un determinato peer. In figura 2.2 è riportato l'andamento della percentuale di traffico di controllo in relazione al numero di peers che compone il sistema. Gli esperimenti sono

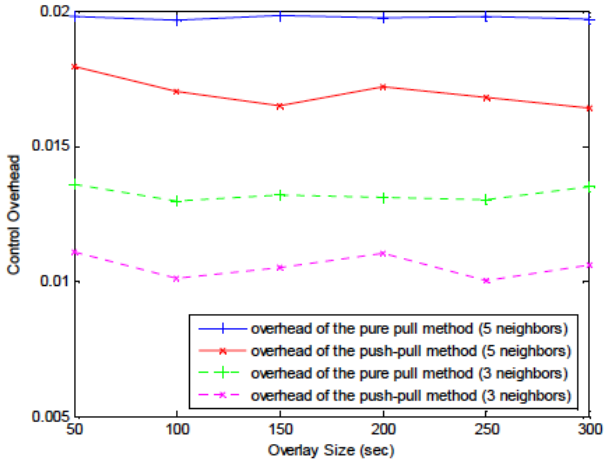

Figura 2.2: Percentuale traffico di controllo in relazione al numero di vicini presenti in GridMedia

stati condotti in uno scenario dove il numero di vicini per ciascun peer è tre e cinque. Le linee tratteggiate si riferiscono al caso di peer avente cinque vicini mentre le due linee solide al caso di tre vicini. Si evince chiaramente come non esista una relazione tra la dimensione della rete overlay e la percentuale di traffico di controllo richiesta. Tale percentuale dipende esclusivamente dal numero di vicini che ogni peer deve gestire: maggiore sarà il numero di vicini maggior traffico di controllo sarà richiesto. Dall'osservanza del grafico si deduce come il metodo *push-pull* riduca il traffico di overhead rispetto al caso di solo metodo pull. In 2.3 è riportato il confronto tra il tradizionale metodo pull ed il push-pull in ambiente statico e dinamico in assenza di limiti di upload nella banda dei peer. Nella situazione di ambiente dinamico 335 nodi

della rete PlanetLab effettuano procedure di join e leave con una distribuzione dei tempi esponenziale a valor medio di 100 e 10 secondi rispettivamente. I risultati ottenuti dimostrano come il metodo push-pull riesca ad ottenere un equivalente livello di α-*playback-time* (rappresenta il ritardo assoluto minimo per il quale la porzione di pacchetti ricevuti supera la soglia imposta dal valore di alfa, con α compreso tra 0 e 1) molto più velocemente del solo metodo pull come schematizzato in tabella 2.3 . Le assunzioni sulla disponi-

	PUSH-PULL	**PULL**
97 % play-back time in ambiente STATICO	4 sec.	11 sec.
95 % play-back time in ambiente DINAMICO	13 sec.	22 sec.

Figura 2.3: Risultati in assenza di limiti di upload in caso statico e dinamico

bilità di una banda di upload infinita vengono rimosse nei dati riportati in figura 2.4 dove si assume una limitazione di 500 kbps nella banda di upload di ciascun peer. Operando con le stesse condizioni del caso precedente in 2.4 sono rappresentati i valori di α-*playback-time*. Il sistema GridMedia è stato

	PUSH-PULL	**PULL**
97 % play-back time in ambiente STATICO	13 sec.	18 sec.
95 % play-back time in ambiente DINAMICO	20 sec.	24 sec.

Figura 2.4: Risultati in presenza di banda di upload pari a 500 kbps in caso statico e dinamico

utilizzato commercialmente da CCTV per la distribuzione del "Festival del-

la Primavera" attraendo un audience di mezzo milione di utenti con 15.000 utenze servite contemporaneamente da un unico server video che immetteva in rete il video al bit-rate di 300 Kbps.

2.4 Topologie ibride-miste

Recenti studi [18], [19], [20], [21] hanno focalizzato l'attenzione sulla ricerca di sistemi che permettessero di unire i vantaggi offerti delle due classiche topologie tree e mesh allo scopo di migliorare le performance del sistema intero. Molte applicazioni P2P live streaming [22], [23] organizzano i peers che fanno parte del sistema all'interno di una struttura ad albero; questa può sembrare la soluzione più naturale ed efficiente per la distribuzione del contenuto video ma soffre di una seria di problemi dovuti all'abbandono dei peer e alla conseguente interruzione del flusso informativo verso i nodi discendenti. Un continuo monitoraggio della struttura ad albero allo scopo di permetterne una rapida riparazione e riaggregazione in caso di abbandono dei partecipanti porta all'introduzione di ulteriori costi e a trasforma quella che pareva essere una soluzione ottima in una da migliorare attraverso, ad esempio, l'impiego di alberi multipli disgiunti [24], [26]. Una soluzione alternativa e completamente diversa sia nell'organizzazione dei peers sia nella modalità di distribuzione del contenuto è rappresentata dai sistemi mesh. A differenza della topologia ad albero risulta esser molto più robusta e versatile in situazioni di variazioni topologiche dove si assiste ad un elevato numero di ingressi ed uscite dei partecipanti nel sistema. Presenta altresì problematiche introdotte dalla necessità di continui scambi informativi per la trasmissione delle informazioni di controllo e di disponibilità. L'approccio intrapreso nel campo dell'ottimizzazione delle topologie overlay è essenzialmente quello di riuscire

ad unire e combinare tra di loro le caratteristiche salienti e fondamentali delle due classiche topologie per riuscire a creare una soluzione ibrida. Vengono di seguito descritti gli approcci utilizzati nella fusione di tali caratteristiche analizzando le innovazioni da essi introdotti.

2.4.1 Hierarchical Ring Tree (HRT)

L'idea che sta alla base di questo sistema consiste nell'unire la topologia ad albero con la topologia ad anello andando a creare una struttura stabile e dinamica con le seguenti caratteristiche:

- Numero di messaggi ridotto per aggiungere-rimuovere nodi
- Tempo di recovery ridotto
- Basso impatto su gli altri peer al verificarsi di eventi di join-leave
- Facile mantenimento e gestione

In figura 2.5 viene rappresentata la topologia HRT nella quale sono visibili le strutture ad anello che connettono i peer nello stesso livello, l'albero che unisce i parent-peer fra di loro ed i link di backup che assicurano il trasporto dei dati dai livelli superiori in caso di abbandono di un di un parent-peer. L'anello rappresenta l'elemento fondamentale di questa topologia: esso serve sia da *data link* che da *backup link*. Gli anelli sono distribuiti ai vari livelli dell'albero e, poiché svolgono due mansioni, sono detti DDR (*distribured/decentralized dual role*). L'unione dei backup link (fra un peer dell'anello ed un vicino d'anello del padre posto a livello superiore) e la struttura DDR permette una notevole riduzione del ritardo introdotto in fase di abbandono dei peer. In una struttura tradizionale ad albero i peer sono a conoscenza soltanto dell'informazione verticale che riguarda padri e figli mentre con

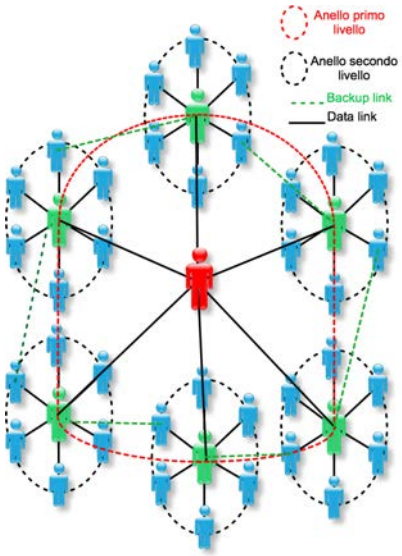

Figura 2.5: Rappresentazione di struttura HRT con anelli congiungenti peers disposti allo stesso livello

l'utilizzo di HRT si aggiunge una conoscenza orizzontale trasmessa ai propri vicini d'anello limitando la diffusione dei messaggi per non andare ad aumentare il traffico in rete. Nel seguito vengono presentate le fasi di join, leave e recovery.

Fase di join

Quando un nuovo peer vuol ricever il contenuto multimediale deve connettersi all'albero di distribuzione e per far questo invia una richiesta alla sorgente dell'albero stesso la quale inoltra la richiesta al nodo foglia con minor lunghezza in modo tale da trovar un padre al nuovo peer. In figura 2.6 viene mostrato l'inserimento di nuovi peers all'interno della struttura nella quale ognuni nodo ha outdegree pari a 6. I primi sei peers vengono posizionati a livello tre ed associati al loro padre e così per i restanti, sempre raggruppati in sei, fino all'esaurimento delle 36 posizioni disponibili al terzo livello. Quando si hanno a disposizione almeno due peer aventi lo stesso parent si crea l'anello

DDR al quale verranno aggiunti i nuovi peer che si uniscono alla rete. Viene quindi creato il link di backup che connette il peer avente numero di posizione minore con un fratello del padre posizionato nell'anello superiore. Se non dovessero esserci fratelli del padre il link di backup viene effettuato con il nonno. Come ultima operazione, nella fase di join, viene inviato un messaggio di update per informare la rete sulla posizione del nuovo peer. Queste informazioni sono essenziali per il mantenimento della struttura HRT, per il monitoraggio dell'occupazione dei sotto-alberi da parte dei peers.

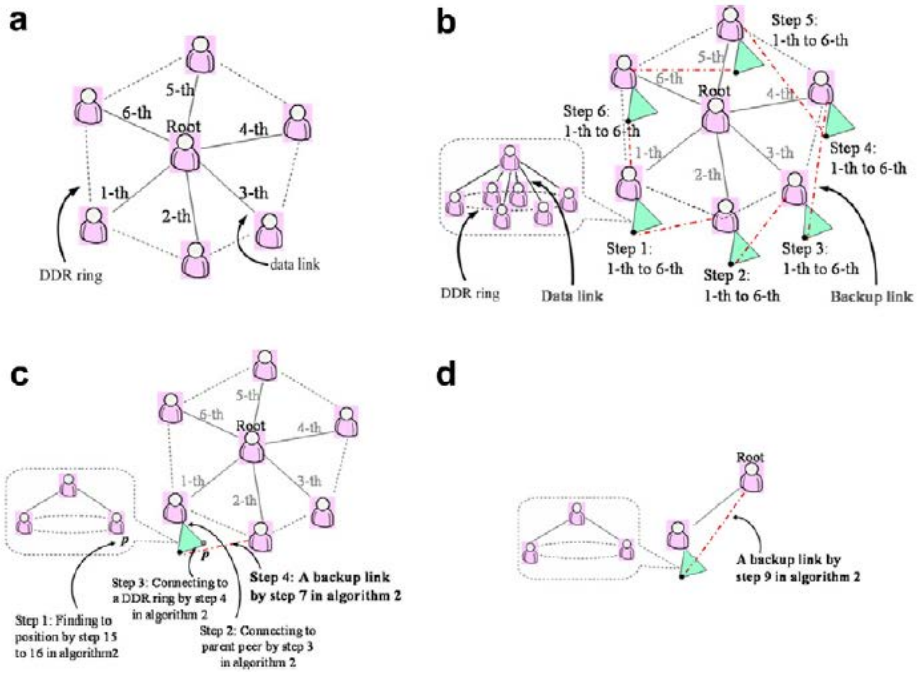

Figura 2.6: Esempio fase di join:a) struttura a due livelli b) struttura a 3 livelli c) backup-link d) back-up link tra figlio e nonno

Fase di leave

Quando un peer abbandona il sistema i dati che provengono dal livello superiore devono continuare ad esser recapitati ai figli orfani e questo avviene

attraverso l'attivazione del backup link e dell'anello DDR che inoltra il flusso informativo ai nodi facenti parte dell'anello stesso. Il peer che ha abbandonato il sistema rompe l'anello di cui faceva parte e per questo i suoi vicini destro e sinistro appena vengono a conoscenza dell'abbandono creano un nuovo anello. Se il *leave-peer* faceva parte di un backup link un nuovo peer viene selezionato per ricreare il link di backup in modo tale da avere un'architettura stabile e pronta ad un nuovo cambiamento.

Fase di recovery

L'uso del backup link e dell'anello DDR rappresenta una soluzione temporanea che deve esser sistemata attraverso la ricollocazione di un nuovo peer nella posizione lasciata dal leave peer. Per far questo viene invocata la procedura di recovery che va a selezionare il nodo da riposizionare fra quelli appartenenti all'anello orfano. Il peer viene eletto padre dell'anello ed il backup link viene ristabilito per preservare la stabilità del sistema. Il mantenimento della struttura HRT richiede l'utilizzo di messaggi di update nelle situazioni di join-leave e tali messaggi sono trasmessi solamente nei suoi immediati livelli superiori e non nell'intera rete. Con questo meccanismo la struttura dell'albero multicast può essere monitorata e comunicata attraverso l'utilizzo di soli messaggi locali. Le simulazioni effettuate nello studio [28] dimostrano che tale topologia mista porta notevoli vantaggi in termini miglior performance nel *ritardo end-to-end* e *loss rate*, in QoS, in un veloce tempo di *recovery* ed in una effettiva riduzione dell'*overhead*.

2.5 Mesh in struttura ad albero

L'idea di unire i vantaggi prodotti dall'impiego di un una topologia ad albero con quelli della topologia mesh ha portato alla creazione di sistemi a topologia ibrida. Recenti studi hanno analizzato in dettaglio tali soluzioni che nel seguito vengono descritte ed illustrate.

2.5.1 mTreebone

Lo studio [80] propone l'interessante fusione della topologia ad albero unita a quella mesh: l'analisi condotta in [91] evidenzia come una struttura di tipo mesh dipenda da un numero di nodi stabil1, nodi che, grazie alla loro presenza continua nel sistema, contribuiscono maggiormente alla diffusione del contenuto. Il sistema mTreebone si basa proprio su questo principio: l'individuazione di nodi stabili che formano un albero (*treebone*) attraverso il quale il flusso video viene distribuito (in modalità *push*) verso gli altri nodi. Tale dorsale è supportata da una rete mesh che unisce i nodi stabili agli altri peers permettendo la creazione di una struttura in grado di reagire alle dinamiche del sistema e di sfruttare pienamente le risorse messe a disposizione degli utenti. La struttura ibrida che si viene a creare è frutto della fusione delle migliori qualità e caratteristiche dell'approccio *tree* e di quello *mesh*. Tale unione porta ad una serie di problemi da affrontare nella crazione della topologia ibrida: in primo luogo la determinazione di quelli che sono i nodi stabili, la costruzione del *treebone* e la coordinazione necessaria tra la modalità *push* tipica della struttura ad albero e quella *mesh-pull*.

Architettura di mTreebone

L'elemento chiave dell'intero sistema è la dorsale *treebone* che unisce tra loro un sottoinsieme di nodi stabili. La stabilità di un nodo, ed il conseguente impiego nell'albero, si basa sulla valutazione del tempo trascorso dal nodo all'interno del sistema. Come esaminato in [96] i nodi con maggior età tendono a rimanere più a lungo all'interno del sistema quindi, superata una determinata soglia di permanenza temporale, tali nodi possono essere considerati stabili e promossi all'interno della struttura ad albero. Indentificati i nodi stabili viene predisposta la creazione di una rete ausiliaria mesh che connette sia i rimanenti nodi sia i nodi che compongono il *treebone* come illustrato in figura 2.7. L'organizzazione all'interno della rete mesh implica l'impiego di

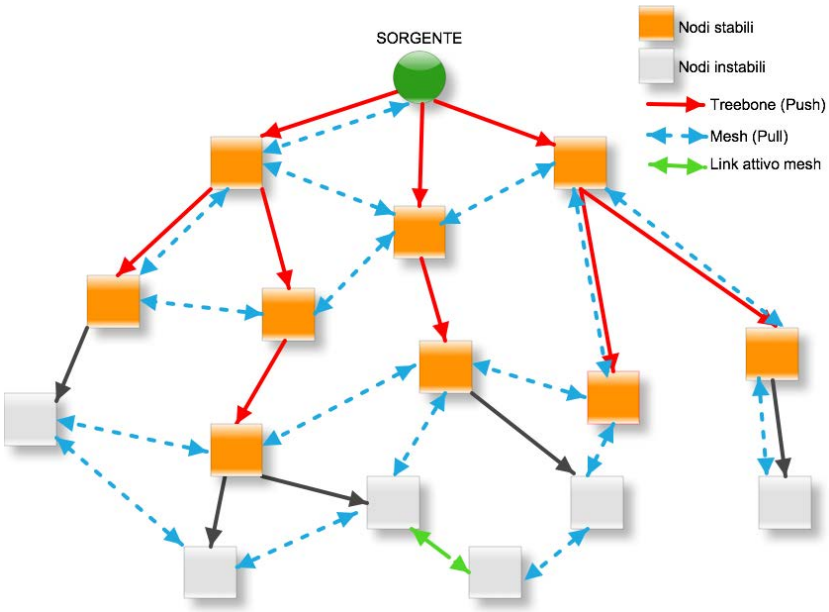

Figura 2.7: Architettura mTreeBone: topologia ad albero e rete mesh

un protocollo di *gossip* che permetta lo scambio d'informazioni rigurdanti lo stato dei nodi vicini ed il materiale da loro posseduto in caso che il flusso video non giunga attraverso la dorsale *treebone*. La miglior efficienza e re-

sistenza offerta dalla rete ausiliare mesh è evidente in presenza di *leave* da parte di un nodo facente parte dell'albero: i figli orfani sono in grado di ricevere il flusso richiedendolo ai propri vicini mitigando quindi l'interruzione di flusso durante la fase di ricerca di un nuovo genitore. La figura 2.8 mostra la fase di leave di 2 nodi: l'abbandono del nodo A (nodo instabile) non provoca cambiamenti all'interno del sistema in quanto i suoi vicini continueranno a ricevere il flusso video dalla struttura *treebone*. Viceversa l'abbandono del peer B (nodo stabile) impone ai propri figli l'utilizzo della rete mesh durante la fase di ricerca del nuovo genitore.

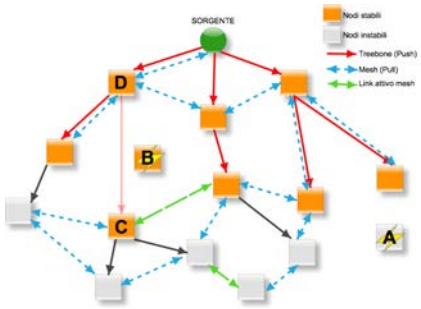

Figura 2.8: Fase di leave per i nodi A e B

La creazione della struttura ad albero avviene in maniera dinamica ed ottimizzata in modo tale da sfruttare al massimo la caratteristica del basso ritardo end-to-end attraverso il mantenimeto di un albero di bassa profondità. L'ottimizzazione della struttura ad albero è attuata per mezzo di due algoritmi di locali:

- *High-degree preemption:* viene impiegato nel caso in cui un nodo possiede più figli rispetto al numero di figli posseduti dal proprio genitore (figura 2.9). L'algoritmo consiste nello scambio periodico di informazioni riguardanti il numero di figli serviti da ciascun nodo: nel caso riportato in figura il nodo X comunica al proprio genitore Y il numero

di figli Z e, l'analisi della situazione porta allo scambio di posizioni tra padre e figlio.

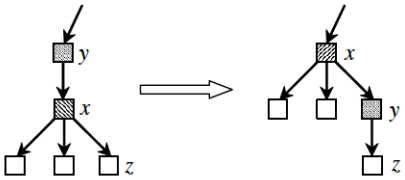

Figura 2.9: Situazione in cui opera algoritmo High-degree preemption

- *Low delay-jump:* rappresenta il caso in cui un nodo al livello superiore, quindi più vicino alla sorgente, dispone ancora di banda inutilizzata per ospitare ulteriori figli (2.10). Tale algoritmo verifica la presenza di un nodo ad un livello superiore rispetto al corrente genitore in grado di ospitarlo. In figura il peer Y dispone di capacità residua inutilizzata e può quindi ospitare il peer X. Tale ottimizzazione permette di ridurre di un livello l'albero di distribuzione e conseguentemente diminuire il ritardo end-to-end tra sorgente e peer destinatario.

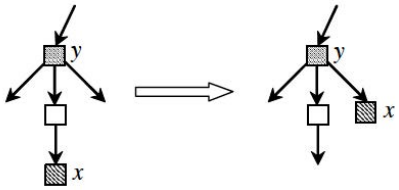

Figura 2.10: Situazione in cui opera algoritmo Low delay-jump

I due algoritmi descritti vengono eseguiti da ciascun nodo fino a quando nessun cambiamento topologico risulta ancora attuabile: tale situazione porta ad ottenere un albero ottimo quanto a profondità di livelli.

Collaborazione tra sistema Pull e Push

L'impiego contemporaneo di due topologie overlay comporta la determinazione di un sistema che permetta la collaborazione tra la modalità di consegna *push* e *pull*. Usualmente i blocchi di dati vengono diffusi attraverso l'albero *treebone* in modalità *push* ma, se per cause dovute al comportamento anomalo della rete oppure all'uscita dal sistema di un nodo stabile, si presenta una interruzione all'interno del playback buffer viene attivata la modalità *pull* e l'invio di richiesta del segmento mancante verso i vicini attraverso la rete mesh. In figura 2.11 è rappresentato il playback buffer di un peer: sono presenti tre puntatori: il *playback pointer* che indica il punto nel quale i blocchi vengono letti e passati al player per essere riprodotti, il *tree-push pointer* indicante l'ultimo blocco ricevuto attraverso *treebone* e una zona collocata tra i due precedenti puntatori denominata *mesh-pull window*. La funzione del puntatore *mesh-pull* è quella di scorrere l'area delimitata dalla finestra alla ricerca di blocchi mancanti e farne richiesta alla rete mesh in caso di necessità. Se un nodo è disconnesso da *treebone* solamente il puntatore *mesh-pull* sarà attivo e l'intera ricezione dei blocchi avverà attraverso la rete mesh fintantoché il nodo troverà una collocazione all'interno dell'albero ed il puntatore *tree-push* sarà riattivato. Il fatto che la finesta *pull-mesh* sia temporalmente disposta all'interno del buffer in un'area che è già stata servita dal puntatore *pull-tree* permette di evitare la ricezione di blocchi duplicati.

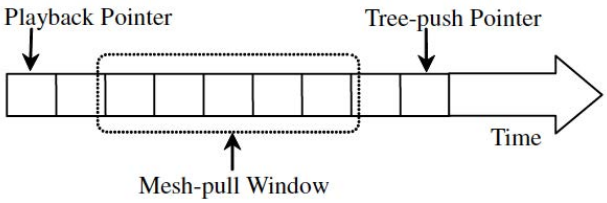

Figura 2.11: Playback buffer

Risulati sperimentali

La campagna di simulazione eseguita per validare i miglioramenti introdotti dalla soluzione mTreebone è stata effettuata comparando il prototipo in esame con i sistemi Coolstreaming (*pull-mesh system*) e ChunkySpread(*multiple tree system*) impiegando come metriche di confronto il ritardo di *start-up* (il tempo che trascorre tra la richiesta di join e la ricezione di sufficienti blocchi per cominciare la riproduzione), il ritardo di trasmissione ed il *rate* di perdita di paccheti. In [80] sono state condotte molte simulazioni e sperimentazioni attraverso l'impiego della piattaforma PlanetLab sia in situazione di ambiente statico sia nel più realistico scenario dinamico. I risultati ottenuti mostrano la superiorità della scelta di topologia ibrida maggiormente performante grazie ai vantaggi offerti dalla fusione della topologia ad albero e mesh. In figura 2.12 sono riportati tre grafici raffiguranti le tre metriche esaminate e confrontate con i sistemi citati: è graficamente possibile vedere la superiorità offerta dalla soluzione ibrida.

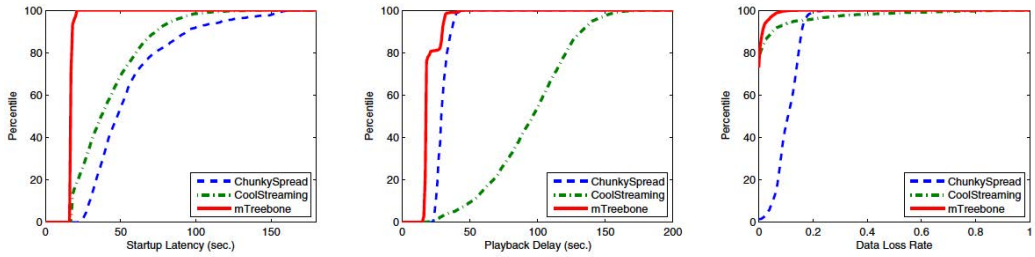

Figura 2.12: Risultati sperimentali in ambiente dinamico. Funzioni di distribuzione del ritardo di start-up, del ritardo di trasmissione e rate di perdita pacchetti confrontate con i sistemi Coolstreaming e ChunkySpread

2.5.2 SmartPeerCast

Quattro sono i principi su qui si basa SmartPeerCast [19] evoluzione di Multipeercast [18] a sua volta derivato da PeerCast [26], [27]

- Raggruppamento in cluster basato su offerta banda di upload dei peer
- Qualità dello *streaming* che varia in modo adattativo tra sorgente e destinazione in base alle caratteristiche della rete
- Peer ricevente monitora il padre e dinamicamente sceglie il migliore
- Uso di un motore di transcodifica all'interno dei singoli peer

La topologia di SmartPeerCast può essere considerata una rivisitazione di ALM (*Application Layer Multicast*) nella quale i nodi allo stesso livello possono connettersi fra di loro e l'albero di distribuzione è utilizzato per unire i cluster. La struttura di SmartPeerCast e' formata da tre elementi:

- Sorgente
- Tracker
- Peer

La sorgente ha il compito di immettere il flusso video in rete ed essa fornisce tre *streams* video (naturalmente dello stesso contenuto) codificati contemporaneamente a livelli di qualità differenti e poi ciascuno di essi viene distribuito attraverso il proprio albero. Il tracker svolge il ruolo di supernodo, gestisce le fasi di join e di leave ed utilizza metodi per premiare e punire i peer che si comportano in modo più o meno virtuoso. I peer sono, come sempre per i sistemi P2P, le unità basilari. Essi sono eterogenei fra di loro, dispongono di diverse capacità di banda e, per evitare la formazione di colli di bottiglia,

dispongono di un motore di transcodifica che permette loro di adattare in modo dinamico il *bit rate* del video da loro trasmesso in base alle condizioni che si creano nella rete. Il peer comunica con il tracker sia il proprio ingresso

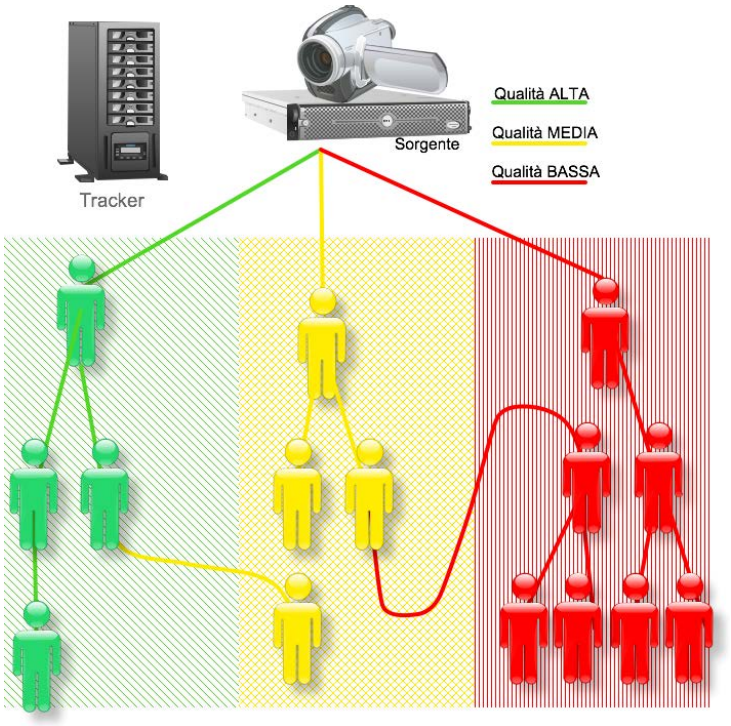

Figura 2.13: Architettura SmartPeerCast

in rete sia l'abbandono. Se l'uscita avviene in modo inaspettato saranno i messaggi di keep-alive ad informare il sistema della scomparsa del peer. Il peer costantemente monitora il QoS che riceve dal proprio padre e con esso scambia messaggi allo scopo di poter migliorare la qualità del video ricevuto.

Fase di Join

Un nuovo peer che vuole partecipare alla sezione video deve seguire le seguenti fasi:

- Registrazione: viene inviato al tracker il messaggio di richiesta <*peer id, uploading bandwidth, max connections, stream id*>. In base al valore di upload dichiarato il peer verrà associato all'albero che più si addice a lui. Se il valore di upload è falso l'algoritmo SmartQoS individuerà tale incongruenza e punirà il "peer bugiardo" associandolo ad un albero con qualità inferiore.

- Calcolo della qualità *stream* che può ricevere: in SmartPeer un utente può ricevere un flusso video che abbia *bit rate* minore delle propria capacità di upload.

- Raggruppamento: il peer riceve una lista di nodi prescelti e questi vengono raggruppati in base al valore di upload del candidato. Nella lista sono presenti anche peer con capacità di upload maggiori in quanto questi possono adattare il flusso per i livelli più bassi attraverso l'utilizzo del motore di transcodifica.

- Trovare il parent node: ricerca all'interno dei tre gruppi i parent dal cui ricevere il flusso. Prima si ricerca all'interno del gruppo più simile e, solo se non trovo partner disponibili, ricerco nei livelli superiori che implicano però un dispendio di energie poiché il transcodificatore deve esser usato.

- Ricezione e riproduzione dello stream: una volta eseguita la procedura di *handshake* con il parent il peer inizia la ricezione e conseguente riproduzione del video.

Algoritmo di Smart QoS

L'algoritmo di SmartQoS rappresenta il cuore del sistema per quanto riguarda l'adattamento del *bit rate* del flusso in base alle situazioni in cui si trova

la rete. Attraverso il suo impiego si evita la creazione di colli di bottiglia e si mantiene basso il *jitter* del ricevente. Questo algoritmo si basa sull'analisi di indicatori di posizione all'interno del buffer del peer ricevente. Il buffer viene suddiviso in tre aree così denominate: *low water*, *hight water* e *normal area* e, gli eventi che riguardano la variazione dell'indicatore di posizione all'interno di queste aree chiamati *QoS event* vengono adoperati come metro di misura per comprendere la qualità ricevuta dal trasmettitore P_j e per adattare il *bit rate* impiegando il motore di transcodifica. Due sono gli eventi di QoS presi in considerazione in questo algoritmo:

- *Bad Qos event:* la posizione corrente arretra dal livello di buffer alto verso la zona normale-bassa. Tale evento indica la presenza di un collo di bottiglia fra P_j (trasmittente) e P_i (ricevente) poiché il *rate* con cui si riceve è molto minore rispetto al rate con cui si va a leggere dal buffer. Questo comporta uno svuotamento repentino del buffer con conseguente situazione di *freezing*. Per evitare questo il peer P_i invia alla sorgente il messaggio di *bad QoS event* e questo attua la procedura di utilizzo del motore di transcodifica ed abbassa il livello di rate con cui inoltra il video.

- *Good QoS event:* si verifica quando la posizione corrente di lettura nel buffer passa dal livello normale al livello alto. Questo indica che la velocità di lettura ha raggiunto il livello di quella di trasmissione evidenziando che il throughput tra i e j diventa maggiore del rate di trasmissione. Un messaggio di *good QoS event* viene quindi inviato alla sorgente che provvede ad aumentare il rate di trasmissione.

All'interno del buffer deve esser scelto un livello di soglia iniziale superato il quale la riproduzione del video possa iniziare. Tale scelta risulta complessa

poiché deve essere pensata in modo tale da garantire una riproduzione fluida e tempi di *start-up* bassi. In SmartCast tale soglia è stata scelta pari alla dimensione del HWM (*high water marker*) pari a 80% della dimensione del buffer intero.

2.5.3 Algoritmo di scheduling del ricevente

Questo algoritmo è eseguito ad intervalli regolari Δt dal peer ricevente in modo da monitorare il QoS offerto in upload dal trasmittente P_j e, in caso questo non soddisfi le esigenze del peer, serve a selezionare un nuovo parent più performante. SmartPeer conteggia il contributo offerto da ciascun peer e questa misurazione avviene attraverso il parametro T_{lwm} che registra il tempo nel quale il puntatore di posizione corrente si trova in una posizione inferiore alla soglia di *low water*. Da questo parametro viene ricavato l'indice BSI *(bandwith sharing index)* calcolato come: $BSI = T_{lwm}/T_{tot}$. Esso definisce la qualità dell'inoltro da parte di P_j attraverso il rapporto fra il tempo in cui P_i riproduce il video stando nella zona bassa del buffer ed il tempo totale di trasmissione dello stream. BSI indica la probabilità di *jitter* in i ricevendo il flusso da j. La variazione di BSI indica un cambiamento della qualità del link: se aumenta indica un deterioramento e se invece scende al di sotto di una determinata soglia significa che P_j non è più in grado di gestire la trasmissione. Si contatta quindi il tracker e si ricomincia l'algoritmo di join per la ricerca di un nuovo parent.

L'impiego del raggruppamento di peers in cluster, l'utilizzo di un motore di transcodifica, la politica di premio-punizione hanno portato ad avere una soluzione scalabile e performante.

2.6 MeTree

MeTree [20] è un sistema che propone l'utilizzo di una topologia ibrida costituita da reti mesh inserite all'interno di una struttura ad albero andando a sfruttare i vantaggi delle due soluzioni considerando al contempo anche il posizionamento ed il contributo offerto da ciascun utente. I peer vengono collocati all'interno di cluster sulla base della capacità di upload offerta e connessi tra loro con rete mesh. Tali gruppi di peer formano delle sottoreti che vengono connesse alla struttura intera grazie alla giunzione all'albero. Il raggruppamento dei cluster permette una differenziazione del servizio e sprona contemporaneamente i peer a contribuire maggiormente.

2.6.1 Principi di MeTree

Anche in questo sistema gli elementi fondamentali sono il *media source*, il t*racking server* ed il peer. La sorgente, utilizzando il paradigma peer-to-peer, distribuisce il contenuto ad alcuni peer che a loro volta lo diffondono attraverso le loro connessioni. La banda generale del sistema quindi aumenta all'aumentare dei peer che ve ne fanno parte.

La suddivisione dei peer è così ripartita:

- Nodo radice: sorgente del video.
- Nodi ISP: nodi che appartengono al secondo livello di distribuzione.
- Nodi della sottorete: tutti i restanti nodi.

I nodi vengono posizionati ai vari livelli in base al contenuto che sono in grado di offrire: più si è vicini alla sorgente minore è il ritardo con cui si riceve il contenuto e maggior è lo sforzo che viene chiesto al nodo per inoltrare il contenuto e diffonderlo ai livelli più bassi. Dalla sorgente il contenuto

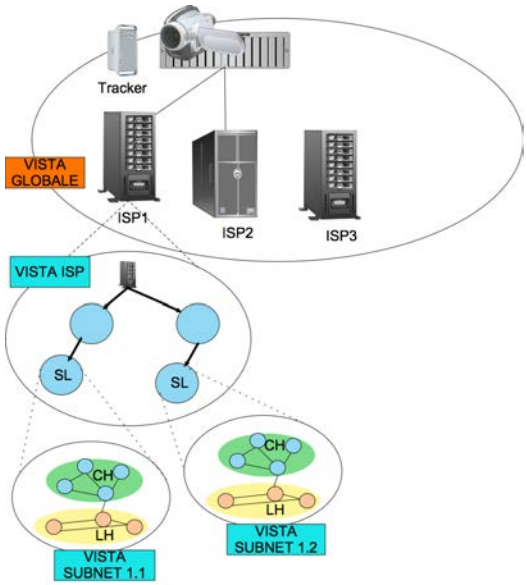

Figura 2.14: Architettura di MeTree

raggiunge i nodi ISP da dove si diramano gli alberi ISP che hanno come radice l'ISP leader (IS) che rappresenta il nodo più performante in termini di banda. Gli altri nodi degli alberi ISP sono i *subnet leader* (Sls) che sono gli elementi di giunzione verso la sottorete mesh. I peer vengono quindi divisi in due classi: CH (*hight contibution*)e LH (*low contibution*). La divisione tra LH e CH viene fatta basandosi sulla soglia Tc, soglia calcolata come n (indice di contribuzione) volte il *rate* dello stream:

$$T_c = nR \tag{2.1}$$

Il tracker calcola il contributo di upload offerto dai peer CH appartenenti alla sottorete i come:

$$TUC_i^H = \sum_{j \in CH_i} BU_j \tag{2.2}$$

con BU_j parametro indicante capacità di upload del peer j che a sua volta appartiene alla sottorete i. Per garantire un miglior QoE ai peer CH viene loro riservata una parte di banda quindi la banda disponibile ai peer CL vien

data da

$$TUC_i^{H2L} = TUC_i^H - (R \times CH_i - R_i^b) \tag{2.3}$$

Ricevuto il contenuto dal nodo SL questi lo dirama al gruppo dei CH che a sua volta lo invia al gruppo dei CL. Per poter formare una sottorete è necessario raggiungere una soglia di peer minima. Se tale soglia non è raggiunta i peer sono collocati temporaneamente in una zona comune dove creano una rete mesh fra loro e dove il più virtuoso viene connesso direttamente alla sorgente per ricevere il contenuto.

2.6.2 Fase di join

La prima fase che un peer si presta ad effettuare è quella di contattare il tracker comunicandogli il proprio indirizzo ed il contributo di banda in upload che si offre. Il tracker esaminando l'indirizzo IP, la banda offerta ed il proprio database verifica l'esistenza di una sottorete i che possa ospitare il peer p. Se il numero di peer facenti parte della sottorete i supera la soglia Ts al peer p viene inviata una lista di peer da contattare per stabilire la connessione. Contemporaneamente il tracker verifica se l'aggiunta del peer p ha portato ad una situazione tale per cui la sottorete i debba essere ricollocata all'interno dell'albero. Nel caso in cui la sottorete i non esista si possono avere due situazioni:

- Nell'area comune sono presenti peer aspiranti a far parte della sottorete i e con l'arrivo del peer p viene raggiunta la soglia minima di partecipanti. I peer vengono quindi spostati dall'area comune, ricreano una nuova rete mesh in cui partecipa anche il nuovo peer p, eleggono il leader SL e connettono la sottorete all'albero.

- La soglia non viene raggiunta ed il peer p resta nell'area comune in attesa ed inizia la ricezione del contenuto tramite le connessioni con i vicini dell'area comune.

La struttura dell'albero in caso di ingresso di una nuova sottorete oppure, in caso di variazioni nelle sottoreti già presenti, deve esser in grado di evolversi e ricollocare i gruppi di peer in modo tale da mantenere l'ordinamento secondo il quale le sottoreti con TUC maggiore sono ai livelli più alti e quelle posizionate allo stesso livello sono disposte in ordine decrescente da sinistra verso destra.

2.6.3 Peer leaving

Un peer intenzionato all'abbandono di MeTree invia un messaggio di leave al tracker il quale deciderà a sua volta se eliminare o meno dall'albero il nodo SL corrispondente al gruppo di cui il peer faceva parte. Se il tracker decide l'eliminazione una lista di peer dell'area comune viene inviata ai peer della sottorete che vengono quindi spostati. Se diversamente la sottorete viene mantenuta si procede alla rivalutazione del TUC ed all'eventuale riposizionamento all'interno dell'albero. Simulazione effettuate nello studio [20] dimostrano che gli incentivi verso i peer più meritevoli, l'architettura combinata albero-mesh, il raggruppamento in cluster porta notevoli vantaggi sotto vari aspetti: robustezza, scalabilità e QoE.

2.7 StreamComplete

Streamcomplete [29] rappresenta un'innovazione nel campo della distribuzione dei contenuti *live* attraverso architetture peer-to-peer in quanto non

solamente unisce i vantaggi delle consolidate topologie mesh e tree in un'unica soluzione ma sviluppa altresì un nuovo algoritmo per la gestione ed ottimizzazione della rete overlay. Il parametro fondamentale è rappresentato dalla cosiddetta "funzione di salute" H(nodo) che viene calcolata da ciascuno nodo sulla base dell'utilizzo della banda in ingresso (X), sull'utilizzo della banda in uscita (W), sul valore nominale della banda a disposizione (Y), sul numero di fornitori (Z) e sulla distanza dalla sorgente video (T). Ciascun valore, pesato con opportuni coefficienti, contrinuisce alla determinazione delle funzione H del nodo *i-esimo* definita come:

$$H(nodo_{i-esimo}) = \alpha X - \beta W + \chi Y + \delta Z - \epsilon T \tag{2.4}$$

L'informazione riguardo lo stato di salute viene distribuita in rete durante ogni scambio di dati ed i peers che risultano godere di una miglior salute sono posizionati nella parte centrale della rete, andando così a formare un backbone di distribuzione "sano" e robusto mentre i peer in possesso di bassi valori dell'indicatore H sono rilegati ai bordi periferici della rete. Per la costruzione della gestione e manutenzione della rete StreamComplete implementa quattro algoritmi distribuiti che di seguito vengono descritti.

- *Join:* la fase di join, come usuale, consiste nella ricerca di patner a cui collegarsi per poter ricevere il contenuto video. Il nuovo peer invia una richiesta di join ad una lista di peers (ricevuta a priori attraverso un tracker) ed attende il messaggio di risposta. Il peer che si vede giungere la richiesta esamina la propria banda a disposizione e, nel caso che questa non sia sufficientemente elevata per gestire la nuova connessione, inoltra tale richiesta ad uno dei suoi parent, più precisamente a quello con maggiore salute e maggiormente distante dalla sorgente video. Questa scelta permette di distribuire il carico sulla rete e di

non inoltrare tutte le richieste rifiutate dai peers verso la sorgente. Se invece il peer che ha ricevuto la richiesta dispone di banda diventerà uno dei fornitori dei peer richiedente.

- *Membership:* attraverso l'utilizzo di questa funzione il peer è in grado di ampliare la conoscenza della rete aggiornando la lista dei propri vicini attraverso l'invio di messaggi di *LocalViewRequest*.

- *Scheduling:* la procedura di scheduling viene invocata in presenza di abbandono del peer o di inutilizzo della banda in ingresso. Tale procedura ricerca nuovi fornitori in maniera tale da poter utilizzare al meglio le risorse a disposizione del peer ed aumentare il valore della propria funzione di salute.

- *Departure:* all'abbondono di un peer il sistema deve essere in grado di riorganizzare la propria struttura topologica. Questo avviene attraverso l'invio di messaggi di *Departure* verso i vicini che saranno quindi in grado di interompere la connessione in corso, cambiare la visione locale dei propri vicini e liberare risorse per far fronte a nuove richieste di join.

L'ottimizzazione della rete overlay in StreamCompleate avviene attraverso l'impiego di due innovative procedure chiamate *Fast-Top Procedure* e *Loop-Check*. La prima viene impiegata nel caso in cui il generico peer x disponga di fornitori con un stato di salute troppo basso, inferiore allo stato posseduto del peer x. In presenza di questa situazione il peer x, che precedentemente svolgeva la funzione di figlio del fornitore, inverte i ruoli e diventa lui stesso fornitore. Il peer x esegue le procedure per ricevere il flusso video dai fornitori del sul suo stesso ex-fornitore trasformatosi in figlio e spostato in posizione periferica della rete mesh causa l'incapacità di svolgere adeguatamente la

funzione di distribuzione del materiale a lui richiesta. L'intera procedura di "scambio ruoli" avviene in sei fasi come mostrato in figura 2.15

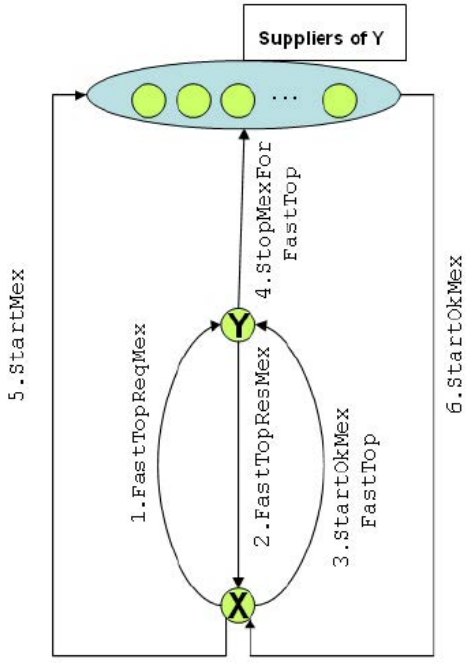

Figura 2.15: Scambi di messaggi per ottimizzazione topologica in StreamComplete

1. La prima fase consiste nella ricerca del fornitore del peer X con il minor grado di salute. Si procede quindi con l'invio del messaggio ReqMex verso il parent Y.

2. Y all'arrivo del messaggio verifica il proprio stato di salute risponde ad X attraverso il ResMex dove conferma o smentisce il proprio stato.

3. In caso di invio da parte di Y di un ack positivo due operazioni sono intraprese: (i) Y invia un messaggio di ai propri fornitori ed interrompe l'inoltro dei flussi video verso il figlio; (ii) invia la lista dei propri fornitori ad X

4. In caso risposta negativa il peer X interrompe la procedura e continua ad accettare il video trasmessogli da Y. In caso di risposta affermativa X contatta i fornitori di Y per istaurare una nuova connessione.

5. Ricevuto l'Ack dai suoi nuovi fornitori X è di nuovo in grado di ricevere il flusso video

6. X procederà all'invio del flusso verso Y divenuto ora suo figlio.

La procedura di *Loop-Check* consiste nel verificare la presenza di anelli all'interno della rete che connettono tra di loro genitori-figli-genitori. Per avere una procedura veloce che non appesantisca ulteriormente il traffico in rete il controllo degli anelli viene limitato alla dimensione della vista locale di cui il peer dispone. Tale verifica viene fatta dal peer X inviando un messaggio di Loop-Check verso i figli: se X riceverà lo stesso messaggio da uno dei suoi fornitori significa che è presente un anello nella topologia overlay. Al verificarsi di questa situazione si risponde attraverso l'interruzione del collegamento verso il fornitore e si invocano le procedure di *membership* e *scheduling*.

I risultati esposti nello studio [29] sono il frutto di una campagna di test effettuata attraverso l'utilizzo della rete di PlanetLab. Tali risultati dimostrano come la procedura di join e la conseguente ricezione di video venga risolta nell'ordine temporale di pochi secondi; l'impiego delle procedure di Fast Top e Loop Check permettono una continua ottimizzazione della struttura overlay che porta ad un conseguente miglioramento dell'intero sistema.

2.8 CDN e P2P

La tecnologia CDN (*Content Delivery Network*) è un sistema alternativo al P2P per la distribuzione di contenuti attraverso la rete Internet. Le reti CDN,

come ad esempio Akamai [65] e Limelight [66], sono costituite da una miriade di server dedicati disposti in posizioni strategiche all'interno della rete in vari ISP. Le richieste di contenuto fatte dagli utenti vengono indirizzate verso i server CDN in base a criteri di località, carico della rete ed altri fattori gestiti dall'operatore di rete che, attraverso un controllo centralizzato, è in grado di amministrare e gestire la propria rete dedicata alla distribuzione. La soluzione CDN rispecchia fortemente il tradizionale paradigra client-server con la differenza che, l'utilizzo di server strategicamente posizionati in rete, l'impiego di sistemi di ottimizzazione e l'intelligente utilizzo delle risorse, permette una gestione di volumi di traffico molto elevati. I servizi delle reti CDN offrono elevati standard qualitativi in termini di QoS offerto, ritardi nella trasmissione e qualità offerta all'utente finale. Il raggiungimento di questi obbiettivi qualitativi richiede forti inversimenti economici: secondo uno studio del Credit Suisse [63] YouTube, che si basa sui servizi offerti da Limelight CDN, pagherebbe una cifra attorno al milione di dollari/giorno in costi di gestione della rete. Inoltre le reti CDN non godono della scalabilità offerta dai sistemi P2P: i server devono essere configurati e predisposti a far fronte ad un traffico stimato e, nel caso in cui il numero di utenti richiedenti il servizio sia superiore a quello previsto, si assiste ad un degrado o, nei casi peggiori, ad una interruzione del servizio causata dal sovracarico di traficco. Vari siti informativi hanno subito disservizi durante la trasmissione in streaming del discorso inaugurale del presidente Obama [64] quando, secondo i dati raccolti da Akamai, si è assistito ad un incremento di traffico pari al 60%.

Sia la soluzione CDN che quella P2P presentano rispettivamente svantaggi ed opportunità: in letteratura esistono vari studi che focalizzano l'attenzione sulla creazione di un'architettura ibrida CDN-P2P che ha portato all'impiego commerciale di tali soluzioni da parte di provider CDN quali Velocix [68],

Octoshape [67] e *LiveSky*, famoso sistema ibrido CDN-P2P sviluppato da ChinaCache [69] per la distribuzione di contenuti live streaming.

2.8.1 LiveSky

Lo studio condotto sulla piattaforma LiveSky [70] è estremamente interessante poichè basato sulla raccolta e sull'analisi di dati provenienti da misurazioni di traffico reale.

Architettura

Come mostrato in figura 2.16 tre sono i componenti principali che compongono l'architettura LiveSky:

- Centro di controllo: composto dal server DNS, GSLB (*Global Server Load Balance*) e sistemi per monitoraggio e pagamento. Questo elemento dell'architettura svolge il compito di indirizzare le richieste dell'utente verso server appropriati e monitorare la distribuzione del carico sulla rete.
- Cache server: sono i server proprietari della rete di distribuzione che servono a traspostare il contenuto dalla sorgente verso l'utente finale. Sono classificati in due tipologie:
  - Core SN: sono i server che formano l'ossatura della distribuzione e che ricevono il contenuto direttamente dalla sorgente.
  - Edge SN: sono i server disposti nel livello distributivo più basso, a confine delle rete direttamente collegati agli utenti finali da un lato ed ai *core server* dall'altro.

L'impiego dei server dedicati garantisce una efficace distribuzione del contenuto attraverso una topologia ad albero. Per garantire ridondanza e stabilità in presenza di problemi sui collegamenti tra server sono mantenute connessioni P2P anche nei livelli alti della rete.

- Utenti finali: *client* che possono ricevere il flusso video in maniera tradizionale oppure partecipare alla distribuzione P2P.

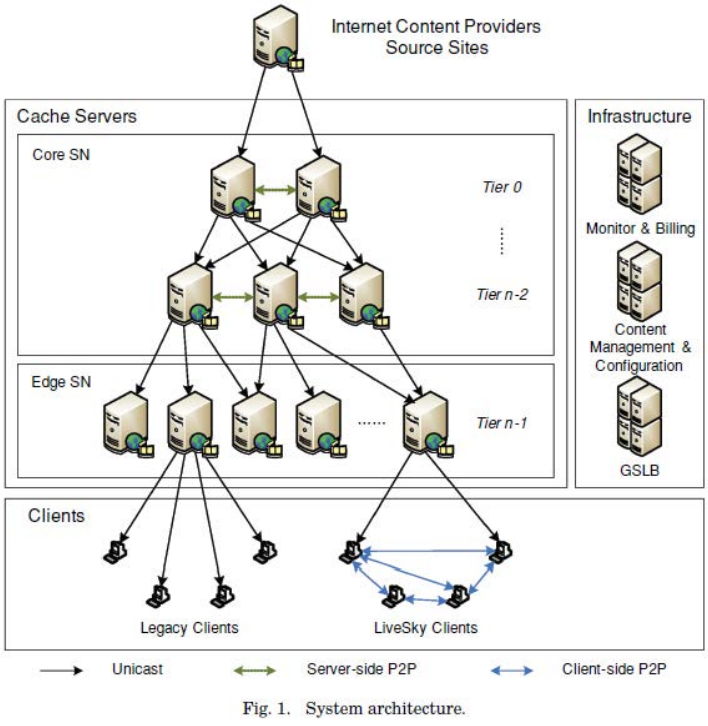

Fig. 1. System architecture.

Figura 2.16: Architettura del sistema LiveSky: collegamenti P2P e unicast sia a livello peer che a livello server

L'utente interessato alla ricezione del flusso video contatta il centro di controllo che interpella il server GSLB il quale, tenendo in considerazione il posizionamento geografico dell'utente ed il carico a cui in quel momento sono sottoposti i server SN, fornisce al client le indicazioni per connettersi al Edge

SN designato. Ciascun server Edge SN svolge il duplice ruolo di *server tradizionale* del mondo CDN e *tracker* per il mondo P2P svolgendo le operazioni di *bootstrap* per i nuovi peers. Le decisioni sulla modalità di distribuzione da adoperare (CDN-P2P) vengono prese dal server in base a delle metriche preconfigurate che verranno discusse in 2.8.1 . La topologia della rete overlay adoperata da LiveSky è anch'essa un ibrido basata su alberi multipli e mesh: il flusso video è suddiviso in vari substreams ciascuno distribuito attraverso un albero. Ogni peer ha quindi un genitore per substream e, per garantire una robustezza maggiore, una rete overlay mesh è mantenuta tra i peers per lo scambio di frame eventualmente non ricevute dalla struttura ad albero.

Modello analitico per scelta utilizzo CDN-P2P

Come detto nel paragrafo precedente i servers Edge SN svolgono il duplice ruolo di tradizionali server CDN ed entità della rete P2P: la chiave nei sistemi ibridi CDN-P2P risiede nella capacità del sistema nel regolamentare l'utilizzo delle due tecnologie in base al carico di lavoro a cui i server sono sottoposti mantenendo alto il livello qualitativo che la rete CDN è in grado di offrire e sfruttare al contempo la scalabilità offerta dal P2P. Per capire come sia possibile trovare un compromesso fra queste due realtà nello studio [70] viene introdotto un modello analitico che permette di controllare il numero di *utenti CDN* che vengono serviti direttamente dai servers di confine e definire allo stesso tempo la soglia oltre la quale si avviano le procedure per la creazione e l'utilizzo della rete P2P. Il modello analizzato assume delle semplificazioni: la rete overlay P2P viene considerata come un singolo albero e non una topologia mista foresta-albero. I ritardi tra la sorgente e la destinazione vengono quindi conteggiati in termine di *livelli*. Il flusso video è diviso in fragmenti di egual lunghezza. Lo studio analitico porta alla rappresentazione

di figura 2.17. In ciascun edge SN è preconfigurato un algoritmo che permette

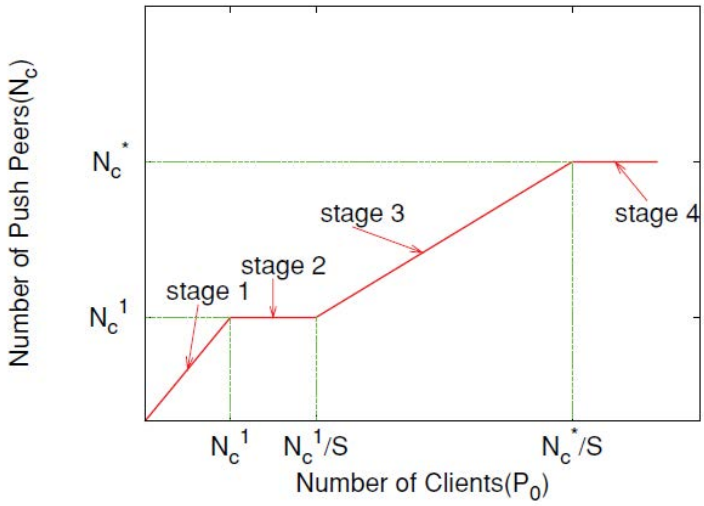

Figura 2.17: Le quattro diverse modalità operazionali adottate dal server SN in base alla numerosità degli utenti da servire

di determinare N_c (numero di utenti direttamente serviti dal server) in base al carico corrente della rete P_0. Per poter determinare tali valori si ipotizza di conoscere, da studi sul comportamento degli utenti e sulle capacità in termini di banda degli stessi, i pramentri ρ, λ e σ indicanti rispettivamente la *banda in upload* messa a disposizione dai peer, il tasso di *join* ed il tasso di *leave*. Definito un valore di ritardo sorgente-destinatario accettabile dal parte del sistema (schematizzato dal modello nel valore K_0, numero di livelli dell'albero) quattro sono le situazioni in cui il serve edge SN si può trovare ad operare:

- Stage 1: $P_0 \leq N_c^1$ Tutti gli utenti ricevono il contenuto dal server CDN in quanto poche risorse sono disponibili per la creazione della rete P2P.
- Stage 2: $N_c^1 \leq P_0 \leq \frac{N_c^1}{S}$ dove S, frazione di utenti serviti dalla rete

CDN, derivanta dal modello analitico è definita come:

$$S = \frac{N_c}{P_0} = \begin{cases} \frac{(1-\rho)(1+\lambda-\sigma)^{K_0}}{(1-\rho^{K_0})(1-\sigma)^{K_0-1}} & \text{se } \rho \neq 0 \\ \frac{(1+\lambda-\sigma)^{K_0}}{K_0(1-\sigma)^{K_0}} & \text{se } \rho = 1 \end{cases} \tag{2.5}$$

I nuovi utenti sono connessi al primo peer libero disponibile al livello 1. Nello stage 2 si passa quindi ad alzare l'albero a 2 livelli.

- Stage 3: $\frac{N_c^1}{S} \leq P_0 \leq \frac{N_c^*}{S}$ Metà utenza è servita della rete CDN e metà dalla rete P2P

- Stage 4: $P_0 \geq \frac{N_c^*}{S}$ Il server edge SN raggiunge la propria capacià massima ed i nuovi utenti sono reindirizzati verso un nuovo server se anche l'albero P2P risulta completo in ogni livello.

Validazione del modello analitico tramite raccolta dati

Il modello analitico descritto nel paragrafo precedente è stato confrontanto con i dati raccolti dall'analisi del sistema commerciale ibrido CDN-P2P sviluppato da ChinaCache. In figura 2.18 è mostrata la disposizione geografica dei server CDN utilizzati dal sistema: 400 server edge SN e 50 *core servers* sono stati impiegati per la distribuzione di uno stream video di 400 kbps. La banda aggregata totale offerta del sistema CDN è stata misurata in 34 Gbps ed è stata usata dal 58.6% degli utenti mentre i restanti peer sono stati serviti dai 17 Gbps messi a disposizioni dagli utenti P2P. Questi dati evidenziano come la soluzione ibrida abbia permesso un risparmio di oltre il 40% nei costi di banda grazie all'integrazione del sistema P2P. Interessante è il caso dei server collocati in regioni ad alta richiesta di contenuti: lo studio riporta che il l'Edge Server della regione di Beijing nella situazione di picco ha servito per una quota superiore al 60% i propri utenti attraverso la rete

Figura 2.18: Disposizione geografica nelle regioni cinesi dei server CDN

P2P rispetto alla percentale media del 40% delle altre aree. In figura 2.19 viene rappresentato il comportamento del server CDN all'aumentare delle richieste di connessioni: la capacità di SN è pari a 200Mbps e in presenza di un flusso video con bit-rate medio di 400 kbps significa esser in grado di servire, nella maniera tradizionale, cinquecento utenze. Dal grafico è evidente come tale limitazione sia fortemente superata raggiungendo il valore di 1200 utenze grazie alla creazione della rete P2P. I dati raccolti dalle misurazioni del sistema commerciale e dalla qualità sperimentata dagli utenti dimostrano come questa soluzione ibrida rappresenti un ottimo miglioramento del tradizionale paradigma CDN: l'impiego della tecnologia P2P permette di aggiungere scalabilità e riduzione dei costi mantenendo i servizi garantiti da un sistema dedicato e centralizzato come quello offerto dalle *content delivery network* e dalla loro sofisticata architettura.

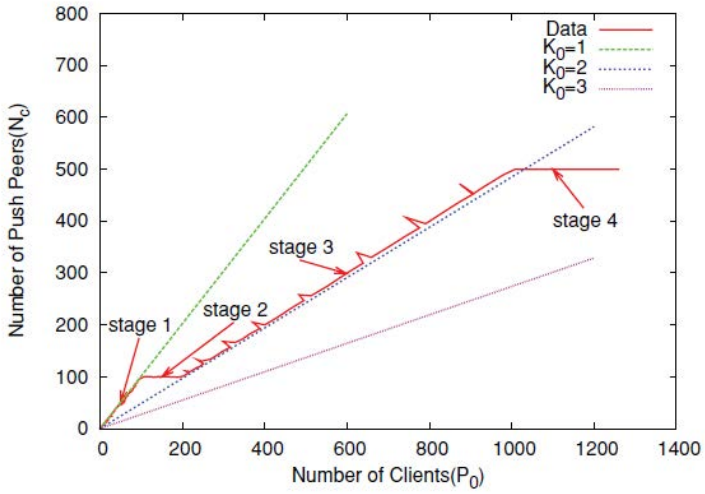

Figura 2.19: Stages di funzionamento del server della regione di Beijing: in ascissa il numero totale di clienti serviti, in ordinata il numero di utenti direttamente connessi al server edge SN

Capitolo 3

Il Video digitale: principi e codifiche

Con il termine *video coding* si identifica il processo di codifica e decodifica di un segnale video digitale che ritrae la rappresentazione di una scena naturale composta da oggetti multipli ciascuno dei quali possiede proprie caratteristiche quali luminosità, illuminazione, ombre e colori. La rappresentazione di una scena naturale, spazialmente e temporalmente continua, viene rappresentata in forma digitale attraverso il campionamento della scena stessa in due domini: quello spaziale e quello temporale come mostrato graficamente in figura 3.1. Il campionamento spaziale viene solitamente ottenuto attraver-

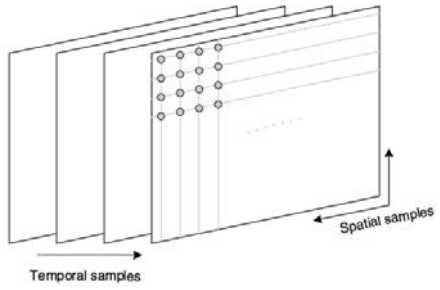

Figura 3.1: Il campionamento spazio-temporale

so l'impiego di una griglia rettangolare nel piano dell'immagine mentre quello temporale avviene acquisendo una serie di immagini (*frames*) ad intervalli di tempo regoli. Ciascun campione spazio-temporale (elemento d'immagine o *pixel*) rappresenta un numero o una serie di numeri che identificano determinate caratteristiche dell'immagine quali la luminosità ed il colore.

3.1 Acquisizione video

Il video viene acquisito attraverso l'utilizzo di telecamere che "intrappolano" le sequenze di immagini attraverso, ad esempio, un sensore CCD (*Charge Coupled Devices*) che converte la scena in segnali elettrici. In presenza di una rappresentazione a colori ciascun componente di colore è separato attraverso un filtraggio ed il valore corrispondente convertito in un valore binario e memorizzato.

3.2 Campionamento spaziale

La griglia rettangolare è il formato più comunemente utilizzato per la rappresentazione dell'immagine campionata. La qualità visiva della dell'immagine è fortemente proporzionata al numero di campioni presenti nella griglia: un numero maggiore di campioni fornisce una rappresentazione più accurata ma allo stesso tempo richiede una maggiore capacità di storage.

3.3 Campionamento temporale

L'illusione di movimento viene data nel video dalla riproduzione in sequenza di ciascuna *frame*. Una frequenza di campionamento temporale alta permette

di avere un numero di fotogrammi al secondo (*fps*) elevato ed una fluidità del video ottenuto sempre maggiore come riportato in tabella 3.2 :

Numero di fotogrammi al secondo (fps)	Fluidità del video digitale percepito
< 10 fps	Riproduzione del movimento "a scatti"
10≤ fps < 20	Movimenti lenti fluidi, movimenti veloci innaturali
20 ≤ fps < 30	Riproduzione omogenea, qualità da standard televisivo
50 ≤ fps < 60	Video ad alta qualità

Figura 3.2: Fluidità del video in base a numero di fps

3.4 Lo spazio dei colori

Le applicazioni video si basano sul bisogno di trasmettere l'informazione riguardante il colore che rappresenta la scena video. In presenza di un'immagine monocromatica è sufficiente disporre di un valore che identifica la luminanza del singolo campione spaziale preso in esame: convenzionalmente un valore elevato rappresenta un campione più luminoso, un valore basso ne rappresenta uno più scuro. Se si utilizzano n bit per la rappresentazione del colore, il valore 0 indica il nero, il valore 2^{n-1} il bianco e l'intervallo di valori compresi fra i due estremi le sfumature dei grigi. Tipicamente la luminanza è rappresentata usando 8 bit per campione in modo tale da disporre di un set di 255 valori. In un'immagine a colori sono richiesti più valori per poter rappresentare accuratamente il colore di ciascun pixel. Esistono numerosi sistemi per la rappresentazione del colore indicati con il termine *Colour Space* (Spazio dei colori). I più adoperati sono: RGB (*red-green-blue*) e YCrCb (*luminance–red cronimance-blue crominance*).

3.4.1 RGB

Nello spazio dei colori RGB il singolo campione d'immagine viene rappresentato impiegando tre valori che identificano rispettivamente le porzioni di rosso, verde e blu. Questi sono i tre colori primari della luce e qualsiasi altro colore viene ricavato da una combinazione di queste tre componenti base. Usando 8 bit per ciascun colore primario si ottiene una rappresentazione a 24 bit per ciascun pixel. La figura 3.3 mostra un'immagine a colori (a) e la

Figura 3.3: Immagine a colori e rappresentazione delle componenti rosso, verde e blu

luminanza delle componenti dei colori rosso, verde e blu. Si nota chiaramente come le fragole rosse appaiano molto luminose nell'immagine (b) mentre risultano molto scure nello spazio dei blu (d) segno evidente dell'assenza di questa componente nella porzione di immagine analizzata.

3.4.2 YCbCr

Il sistema visivo umano è meno sensibile al colore rispetto alla luminosità: la soluzione RGB non trae vantaggi da questa caratteristica poiché gestisce in maniera equivalente le tre componenti di colore assegnando loro la stessa importanza. E' possibile rappresentare un'immagine a colori in maniere molto più efficace separando la luminanza delle informazioni sul colore, ovvero dalla crominanza. Nello spazio YCbCr la componente di luminanza Y (3.1)

viene calcolata come una somma pesata delle componenti rosso(R), verde(G) e blue(B) dove k rappresentano i fattori di peso.

$$Y = k_r R + k_g G + k_b B \tag{3.1}$$

L'informazione del colore viene rappresentata come differenza di colori dove ciascuna componente di crominanza dei colori R,G,B è pari alla differenza tra il valore della componente in esame ed il valore di luminanza Y:

$$Cr = R - Y \tag{3.2}$$

$$Cb = B - Y \tag{3.3}$$

$$Cg = G - Y \tag{3.4}$$

La descrizione completa dell'immagine è data dal valore di Y e dalle differenze di colore Cr, Cb e Cg che rappresentano la variazione fra l'intensità del colore e la luminanza di sfondo dell'immagine. A differenza del sistema RGB ora si è in presenza di 4 valori da trasmettere ma questo inconveniente viene risolto poiché la somma Cr+Cb+Cb è costante quindi solo due delle tre componenti di crominanza vengono trasmesse mentre la terza viene ricavata come differenza. Nel sistema YCbCr vengono quindi trasmesse la luminanza (Y) ed i valori di crominanza del blu e del rosso (Cb e Cr). La figura 3.4 riporta la rappresentazione della luminanza e della crominanza dell'immagine a colore (a). Il vantaggio fondamentale fornito dallo spazio dei colori YCbCr è dato dal fatto che le componenti Cr e Cb possono essere rappresentate con minor risoluzione rispetto alla componente Y poiché l'occhio umano è meno sensibile al colore rispetto alla luminanza. Questo permette di ridurre considerevolmente la quantità d'informazione necessaria.

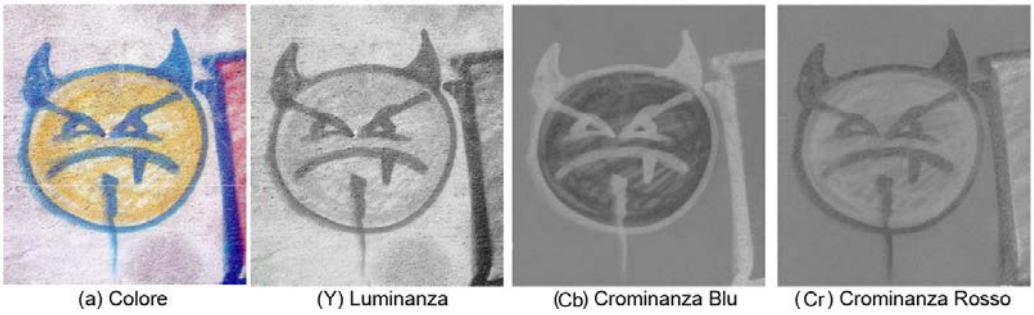

Figura 3.4: Immagine a colori e rappresentazione della luminanza e crominanza del blu e del rosso

Sotto-campionamento delle componenti Cb e Cr

Tre sono le principali metodologie adoperate per il sotto-campionamento delle componenti Cb e Cr:

- 4:4:4 le tre componenti Y,Cb e Cr hanno la stessa risoluzione (figura 3.5-a)
- 4:2:2 le componenti della crominanza hanno la stessa risoluzione verticale ma la metà della risoluzione orizzontale. 4:2:2 significa che per ogni 4 campioni di luminanza nella direzione orizzontale sono presenti due campioni Cb e due campioni Cr (figura 3.5-b)
- 4:2:0 le componenti della crominanza hanno la metà della risoluzione sia in orizzontale che in verticale (figura 3.5-c)

Per comprendere meglio i vantaggi del sotto-campionamento in tabella 3.6 viene rappresentato un esempio impiegando la sotto-campionatura 4:2:0. Attraverso l'impiego del sotto-campionamento la quantità di informazione viene ridotta notevolmente senza perdita di qualità percepibile dall'occhio umano. Un'immagine nello spazio dei colori RGB può essere convertita in quello YCbCr utilizzando le equazioni 3.7. Non è necessario specificare il valore del

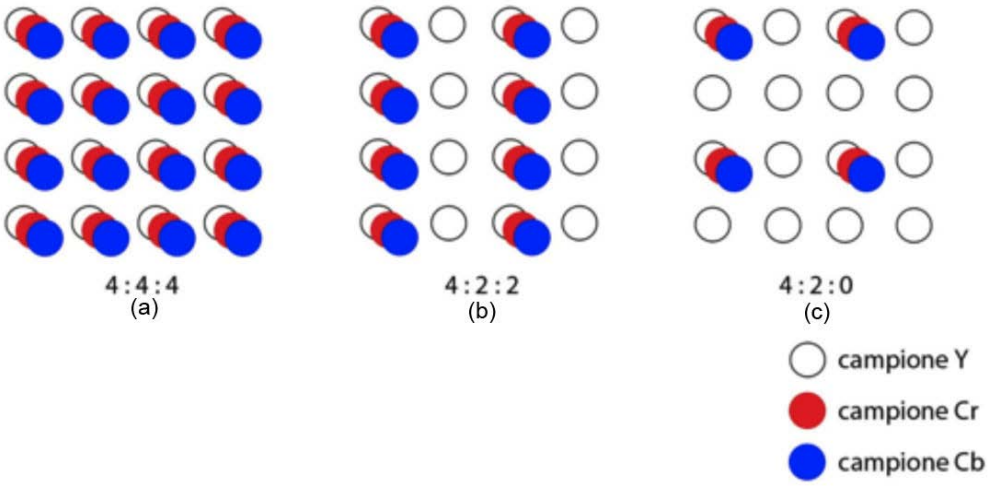

Figura 3.5: Sottocampionamento attraverso le varie risoluzioni

Metodo	Risoluzione immagine	Dimensione
4:4:4	720 x 576 pixel	720x576x8x3= **9953280** bit
4:2:0	8 bit per ciascuna componente	(720x576x8)+(360x288x8x2)= **4976640** bit

Figura 3.6: Confronto delle dimensioni di un'immagine applicando 2 sottocampionamenti differenti

coefficiente k_g poiché è ricavabile come differenza (k_b+k_r+k_g=1) e similmente il valore di G può esser ricavato dalla rappresentazione YCbCr sottraendo i valori di Cr e Cb da Y.

3.5 La compressione video

Nel campo video con il termine compressione ci si riferisce a quel processo che permette di ridurre e compattare una sequenza video grezza in una che occupi meno spazio, condensare l'informazione in un minor numero di *bits*. Questo processo viene attuato attraverso l'impiego di due soggetti:

- *Encoder:* responsabile della conversione del video originale in una rap-

$$
\begin{aligned}
Y &= k_r R + (1 - k_b - k_r)G + k_b B \\
Cb &= \frac{0.5}{1 - k_b}(B - Y) \\
Cr &= \frac{0.5}{1 - k_r}(R - Y)
\end{aligned}
$$

$$
\begin{aligned}
R &= Y + \frac{1 - k_r}{0.5} Cr \\
G &= Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} Cb - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} Cr \\
B &= Y + \frac{1 - k_b}{0.5} Cb
\end{aligned}
$$

Figura 3.7: Equazioni di conversione da spazio dei colori RGB a YCrCb

presentazione più parsimoniosa.

- *Decoder:* soggetto che decodifica l'informazione compressa per poter poi riprodurre il video originale.

La compressione dati viene effettuata attraverso l'eliminazione dell'informazione ridondante: molti tipi di dati contengono informazione statisticamente ridondante e la rimozione di questa permette di avere una compressione *lossless*, senza perdita di informazione. L'utilizzo di questa tecnica di compressione permette di ricostruire l'informazione esattamente come l'originale con lo svantaggio che l'impiego di questa tecnica per la compressione video porta ad una compressione debole, con un fattore di compressione nell'ordine di tre-quattro. Si utilizzano quindi sistemi di compressione *lossy* (con perdita) che forniscono fattori di compressione molto più elevati a discapito del segnale che viene ricostruito a valle: questo non sarà più una coppia fedele dell'originale ma sarà un qualcosa di simile con una qualità percepita dall'utilizzatore proporzionale al grado di compressione impiegato. Questo tipo di compressione basa il proprio principio sulla rimozione di elementi dell'immagine o del video che non vanno ad impattare la percezione visiva dell'utente

finale, sfrutta quindi i difetti del sistema visivo umano per poter eliminare informazione dal segnale originale.

3.5.1 Video Codec

Il segnale video consiste in un sequenza di fotogrammi: singole immagini che si ripetono ad intervalli di tempo regolari. Un'idea per comprimere l'informazione è quella di utilizzare un Codec per immagini e codificare separatamente ciascun frame. Questo processo, chiamato *Intra-frame-coding*, trascura completamente l'informazione data dalla ridondanza temporale presente all'interno di una sequenza video. L'utilizzo di CODEC permette di rappresentare la sequenza video originale attraverso l'impiego di un modello che, idealmente, dovrebbe essere in grado di rappresentare la sequenza d'immagini con il minor numero di bit possibile garantendo il maggior grado di fedeltà possibile. Un video encoder è costituito fondamentalmente da tre unità:

- Blocco temporale: riceve in ingresso il video grezzo e ne analizza la ridondanza temporale allo scopo di ridurlo di dimensioni. Il suo output è rappresentato dalla frame residua che va ad alimentare il blocco spaziale.

- Blocco spaziale: ricerca le somiglianze tra i pixel vicini per ridurre la ridondanza spaziale. Questo viene effettuato attraverso l'applicazione di trasformate e attraverso un processo di quantizzazione del risultato. Le trasformate convertono i campioni spaziali in un dominio diverso nel quale sono rappresentati attraverso i coefficienti della trasformata che vengono poi quantizzati.

- Codificatore entropico: provvede alla compressione degli output dei due blocchi precedenti: i vettori di moto provenienti dal blocco temporale ed i coefficienti provenienti da quello spaziale. Questo codificatore rimuove la ridondanza presente nei dati attraverso, ad esempio, l'utilizzo di codici binari corti per rappresentare le sequenze maggiormente presenti.

L'output di quest'ultimo blocco è una sequenza di bit compressi inviata al destinatario che potrà ricomporla per mezzo del decodificatore. Il compito del video decoder è quello di prendere in ingresso il flusso, estrarne i valori dei coefficienti spaziali unitamente ai vettori di moto per giungere alla ricostruzione della frame. La riduzione della ridondanza temporale avviene

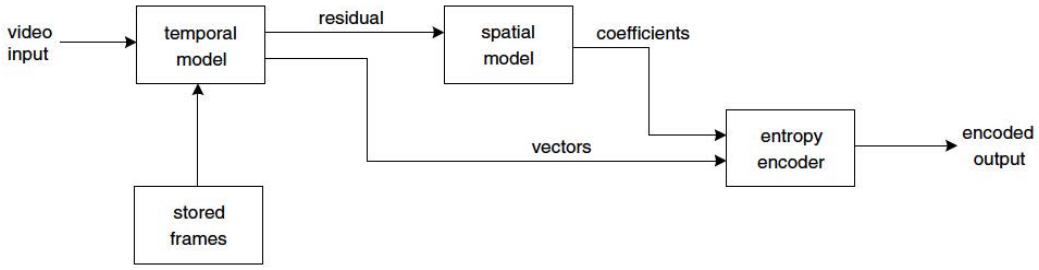

Figura 3.8: Schema a blocco rappresentante processo di codifica

attraverso un processo di predizione del *frame* corrente che viene poi sottratta al frame corrente creando una nuova frame nominata *frame residua*. Il nodo fondamentale di questo processo è rappresentato dalla funzione di predizione: maggiore è l'accuratezza di tale funzione minore sarà l'energia contenuta nel frame residuo e questa potrà essere compressa ed avere dimensioni ridotte. In ricezione, per poter decodificare il frame, il decoder deve necessariamente effettuare il processo inverso aggiungendo la predizione al fotogramma residuo decodificato. La figura 3.9 mostra la frame residua (c) creata dalla differenza fra i fotogrammi (a) e (b).

Figura 3.9: Frame residua (c) creata dalla differenza delle frames (a) e (b)

3.5.2 GOP, I-Frame, P-Frame e B-Frame

In presenza di perdita di una frame il decoder non è più in grado di predire in maniera corretta la frame successiva andando ad impattare negativamente sulla qualità della traccia video. Per evitare che un errore ad inizio video si propaghi per l'intera sequenza vengono poste INTRA FRAME ad intervalli regolari creando dei blocchi di immagini chiamati GOP (*Group Of Pictures*). Ciascuna frame può essere codificata in maniera differente:

- I-FRAME: fotogramma INTRA utilizzato come riferimento per la predizione delle frame successive (*reference frame*). E' codificato senza alcuna predizione e compensazione.

- P-FRAME: fotogramma INTER codificato utilizzando predizione e compensazione. Sfrutta una sola reference frame, I o P, che precede la P-frame corrente.

- B-FRAME: fotogramma INTER codificato utilizzando predizione e compensazione. Sfrutta due reference frames: una Frame I o P ed una B-Frame che precedono e seguono rispettivamente la corrente B-frame.

La presenza dalle I-Frame all'interno del *bit-stream* rappresenta un utilissimo punto di ri-sincronizzazione: esse possono essere decodificate in maniera indipendente rispetto alle altre poiché codificate senza predizione.

3.6 Codifiche scalabili

L'utilizzo delle reti P2P overlay per la distribuzione di contenuti video deve affrontare molte problematiche derivanti dalle necessità imposte dal flusso multimediale in *real-time*: necessità di bassi tempi di start-up, bassi ritardi end-to-end, flusso costante tra i peers partecipanti rappresentano solo alcuni dei temi che vanno affrontati. L'utilizzo dei codec video permette la riduzione di banda necessaria alla trasmissione del flusso video. Data la disomogeneità dei collegamenti che interconnettono i peers all'interno della rete P2P di distribuzione, la codifica del flusso video dovrebbe esser tale da fornire un *bit-rate* e una qualità diversa in base alle condizioni offerte dalla rete. Questo richiederebbe varie codifiche dello stesso flusso video in base alle richieste provenienti dai peers: è evidente che il procedimento diventa molto oneroso ed impraticabile all'aumentare delle diversità tra gli utenti. La soluzione consiste nel codificare il flusso video solamente una volta attraverso l'impiego di un codec intelligente che sia in grado di adattarsi alle disponibilità e alle richieste degli utenti. SVC (*Scalable video coding*) e MDC (*Multiple Description Coding*) sono due tecnologie adoperate per rispondere all'esigenza di scalabilità in ambienti eterogenei.

3.6.1 H.264/SVC

Il recente standard H.264/SVC (*Scalable Video Coding*)[30] aggiunge scalabilità al codec H.264/AVC [31] permettendo la codifica del segnale video in diverse qualità grazie all'utilizzo di diversi livelli di codifica. Attraverso l'impiego della variazione del *rate* di trasmissione del video la soluzione SVC permette un miglior utilizzo delle risorse della rete P2P fornendo la capacità ai peer di ricevere un flusso video con qualità proporzionale alle risorse messe

a disposizione. La codifica scalabile di un video permette al decoder di decidere selettivamente quali parti dello stream video decodificare. Il flusso video è composto da una serie di *layers*: un *base layer* e una serie di *enhancement layer* che forniscono informazione supplementare al fine di migliorare la qualità del video. In figura 3.10 il decoder A decide di decodificare solamente il *base layer* ricevendo quindi una qualità "base" del video: questa decisione può esser dovuta alla presenza di un segmento di rete a capacità limitata in grado di sostenere solamente un video a basso bit-rate. Il decoder B invece riceve tutti i layers e decodifica quindi un video di qualità superiore.

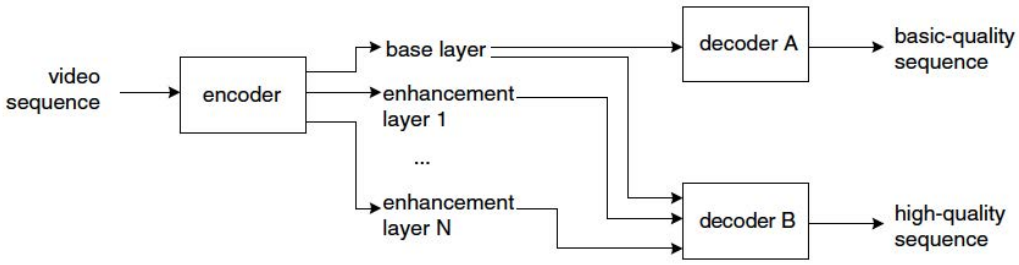

Figura 3.10: Schema rappresentante la suddivisione del flusso video in livelli

La scalabilità viene applicata in tre settori:

- Scalabilità temporale: si riferisce alla possibilità di decodificare il flusso video a diversi *frame-rate* in accordo con le capacità lato ricevitore. Questo offre la possibilità di offrire un basso frame rate attraverso la decodifica del base layer e di incrementarlo attraverso l'impiego di uno o più enhancement layers.

- Scalabilità spaziale: permette la codifica del video a varie risoluzioni spaziali. In questo modo è offerta la possibilità di passare da una risoluzione, ad esempio QVGA offerta dal livello base, ad una HD aggiungendo livelli in decodifica. I dati forniti dai livelli a risoluzione

minore sono utilizzati per predire quelli delle risoluzioni maggiori in maniera tale da ridurre il bit-rate impiegato nel processo di codifica di quest'ultimi.

- Scalabilità SNR, qualità e fedeltà: vengono impiegate per migliorare la qualità visiva del flusso video ricevuto. Un semplice impiego commerciale di questa caratteristica sta nel fornire un'anteprima gratuita del video attraverso la decodifica del livello base mentre l'accesso ai livelli superiori avviene solamente dietro pagamento del servizio.

I diversi layers possono essere trasmessi in differenti flussi video chiamati *sub-streams* oppure esser inglobati nello stesso flusso: in questo caso si parla di *embedded bit streams*.

Figura 3.11: Esempio di scalabilità spaziale e scalabilità SNR

3.6.2 MDC: Multiple Description Coding

MDC rappresenta un strumento in grado di fornire la codifica del flusso video attraverso l'impiego di più rappresentazioni chiamate descrizioni. Le descrizioni create da MDC sono tra di loro indipendenti e caratterizzano in

maniera diversa il video: ciascuna di essa può possedere risoluzioni temporali, spaziali, SNR diversi. Esse possono avere la stessa importanza all'interno del video (schema MDC bilanciato) oppure aver importanza diversa (schema MDC sbilanciato). La ricezione di tutte le descrizioni permette la ricostruzione ottimale del video mentre, in presenza solo di alcune descrizioni, la qualità degrada. La scalabilità viene offerta grazie all'indipendenza delle descrizioni: in presenza di banda insufficiente per la trasmissione dell'intero flusso il trasmettitore può decidere di inviare solamente il numero di descrizione accettate della rete e di scartare le restanti. Diversamente da SVC l'utilizzo di *Multiple Description Codec* non necessita la ricezione di un livello-descrizione base: in MDC non esistono gerarchie tra le descrizioni, ogni descrizione è utile per aumentare la qualità. I benefici offerti dalla totale indipendenza delle descrizioni sono pagate dal punto di vista dell'efficienza della compressione del codec: l'esistenza di ridondanza d'informazione all'interno delle varie descrizioni richiede un impiego di risorse maggiore rispetto ad un sistema tradizionale a singola descrizione. Il grande vantaggio offerto dall'utilizzo dei sistema MDC è la robustezza offerta dal sistema: in presenza di perdite di pacchetti e variazioni topologiche della rete l'impiego di MDC permette la ricostruzione del video. La ricezione di una descrizione sottoposta a perdita d'informazione in un caso a singola descrizione comprometterebbe la qualità dell'immagine percepita mentre, grazie alle informazioni presenti nelle altre descrizioni, l'immagine può essere ricostruita e portare ad una buona qualità anche in presenza di un elevato rate di perdita dei pacchetti in ambito MDC. In figura 3.12 viene mostrato il caso di impiego di MDC con 4 descrizioni in presenza di un tasso di packet loss rate pari al 30%. La ricostruzione dell'immagine originale avviene unendo le descrizioni attraverso un processo di *interleaving* dei pixel. I pixel mancanti (figura centrale) vengono stimati

utilizzando tecniche che analizzano i pixel vicini. La qualità percepita in ricezione è rappresentata nella figura di destra. L'assunzione di essere in presenza di errori indipendenti tra di loro è verificata se le singole descrizioni sono trasmesse su canali e percorsi indipendenti: MDC viene prevalentemente utilizzato in presenza di topologie P2P a foresta dove le descrizioni vengono assegnate ai molteplici alberi.

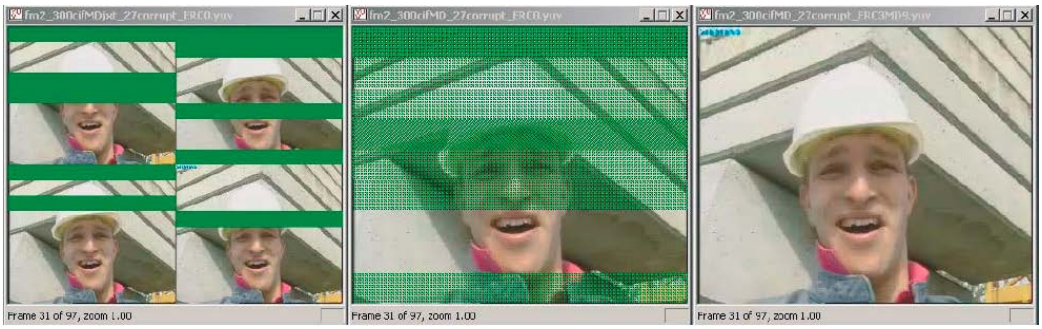

Figura 3.12: Utilizzo di codifica MDC in presenza di 4 descrizioni in presenza di tasso di packet loss pari al 30%

LayerP2P: utilizzo di layer codec in un sistema P2P live streaming

LayerP2P [32] è un sistema P2P per la distribuzione di video live streaming che adotta l'utilizzo dei codec layered in grado di fornire un efficiente servizio differenziato fra i peer. In LayerP2P quando il sistema dispone di banda "in eccesso", cioè quando la banda media disponibile in upload è maggiore del rate del video trasmesso, ciascun peer può ricevere tutti i layer che compongono il video ed usufruire della miglior qualità video offerta dal loro collegamento. Diversamente in una situazione di carenza di banda, dovuta ad esempio da un'eccessiva presenza di peers con limitate risorse di upload o da presenza di *free-rides* (utenti che partecipano alla sezione video senza condividere la banda di upload) il sistema avvia un procedimento che per-

mette di attuare una diversificazione nel trattamento dei peers: coloro i quali contribuiscono maggiormente alla diffusione del video ricevono una qualità migliore mentre gli altri ricevono una qualità commisurata alle risorse impiegate. Le caratteristiche chiave del sistema LayerP2P sono riassumibili in questi tre punti:

- Adattamento dinamico alla banda disponibile: la qualità del video trasmesso viene adattato alla caratteristiche individuali di ciascun peer.
- Degradazione della qualità video graduale: la perdita di pacchetti nei livelli migliorativi non influisce sul processo di decodifica del livello base. Esiste un sistema di prioritizzazione che da maggior importanza al livello base rispetto a quelli superiori.
- Tecniche di incentivo per la distribuzione: vengono forniti incentivi per motivare i peers ad utilizzare la banda in upload. Basandosi sulla cosiddetta strategia del *tit-for-tat* i peers che contribuiscono in maniera maggiore ricevono la qualità video migliore.

In LayerP2P la sorgente codifica il video adoperando L livelli ciascuno dei quali è poi suddiviso in unità denominate *layer chunks* (LCs). Queste sono distribuite attraverso una rete P2P-mesh dove , da parte dei peers, vengono attuate le usuali procedure di join, richiesta di buffer-maps, scambi di pacchetti e leave. Diversamente dai sistemi che impiegano una singola descrizione, in LayerP2P ciascun peer dispone di L buffers, uno per ogni layer, nei quali immagazzina i pacchetti LCs corrispondenti alla descrizione in esame. Periodicamente vengono scambiate tra i vicini le buffer-maps per poter informare i peers sui segmenti video a disposizione e procedere con la segnalazione di richiesta dei LCs mancanti: i peers che ricevono richieste di LCs servono con maggior priorità i peers da cui ricevono più contenuti rispettando

quindi il principio del tit-for-tat che permette l'applicazione di una gestione differenziata del servizio. Sostanzialmente un nodo che fornisce maggior banda verrà ricompensato dai propri vicini tramite l'invio prioritario di LCs.

Implementazione del sistema LayerP2P e risultati sperimentali

La validità della soluzione offerta da LayerP2P è sostenuta dai risultati derivanti da una serie di esperimenti condotti sia attraverso l'utilizzo di PlanetLab sia attraverso un'intensa campagna di simulazioni. Le prestazioni offerte da LayerP2P sono confrontate con due sistemi comparabili che impiegano però un singolo livello di codifica.

Implementazione e metriche di misurazione

Il cuore del sistema LayerP2P sta nell'impiego dei codec layer: sebbene il sistema possa gestire qualsiasi tipo di codec per l'implementazione corrente è stato deciso di utilizzare il codec H.264/SVC tempo scalabile. La codifica avviene utilizzando tre livelli: layer 1 contiene le frame I e P, layer 2 contiene le Bs ed il layer 3 le frame B. Un GOP è composto da 64 frames. L'encoder utilizzato è JM11 H.264 applicato ad una sequenza video di risoluzione 704x576 a 30 fps. I bit rates risultanti sono variabili e riportati in tabella 3.13 Le simulazioni si sono sviluppate in due scenari: in ambiente *underloaded* e *overloaded*: nel primo la banda fornita dal sistema è superiore a quella richiesta dal video, nel secondo caso questa condizione non è verificata. Tre

Livello	Bit-rate
Sequenza video completa	620 Kbps
Layer 1	290 Kbps
Layer 2	230 Kbps
Layer 3	100 Kbps

Figura 3.13: Valori di bit-rate nei layer coficati

sono le tipologie di peers che variano in base alla banda di upload di cui dispongono: peer istituzionali, peer residenziali e free-ride peers. La distribuzione dei peer e la loro capacità d'upload sono riportate in tabella 3.14 La

Peers		Free-ride	Residenziali	Istituzionali
Scenario underloaded in assenza di free-ride peers	Distribuzione (%)	-	40 %	60 %
	Upload rate (kbps)	-	400	1000
Scenario overloaded in presenza di free-ride peers	Distribuzione (%)	10 %	30 %	60 %
	Upload rate (kbps)	0	300	700

Figura 3.14: Distribuzione dei peers nei due ambienti di simulazione

studio comparativo è stato fatto impiegando due diverse tipologie di sistemi a singola descrizione:

- Single-Layer: ciascun peers viene gestito in maniera identica senza differenziazioni dovute ai contributi di upload.
- Single-Incent: viene adoperato il principio del *tit-for-tat* e differenziato il trattamento dei peers.

I dati raccolti durante le simulazioni sono valutati su queste tre metriche:

- Playback rate (R): indice che identifica la frazione di LCs ricevuti nell'intervallo di validità (entro la playback deadlines)
- Received chunck ratio (α): indica la frazione di segmenti ricevuti per ciascun layer
- Average PSNR (Q): per ciascuna frame decodificata viene calcolato il valore medio di PSNR (*peak signal-to-noise ratio*) che fornisce un indice di qualità dell'immagine. Q1 indica il valore per le frames P, Q2 per le Bs, Q3 per le B. Q è il valore medio di PSNR su tutte le frames componenti il video. Data l'immagine originale I e l'immagine compressa

K entrambe di dimensioni MxN il PSNR è definito come: $PSNR = 20log_{10}\frac{(Max(I))}{\sqrt{MSE}}$ con $MSE = \frac{1}{MN}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$

Di seguito sono riportati i risultati illustrati nello studio in esame e brevemente commentati.

Scenario underloaded senza free-ride peers

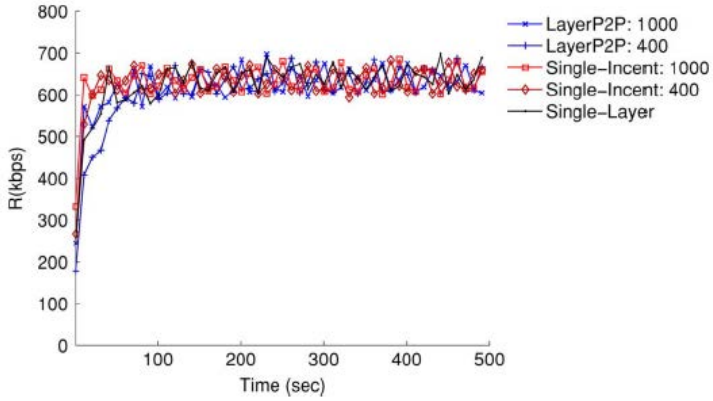

Figura 3.15: Playback rate di peer istituzionale e residenziale in ambiente underloaded

In figura 3.15 e 3.16 è rappresentato il *playback rate* di due peer scelti a caso: uno istituzionale (1 Mbps bada di upload a disposizione) e l'altro residenziale (400 kbps banda di upload a disposizione). Entrambi i peers raggiungono uno stato stabile dopo 100 secondi. Anche i due sistemi a singola descrizione hanno un comportamento del tutto simile a quello dato da LayerP2P. In figura 3.16 si osserva come per le varie tipologie di sistema e per le classi di peers il valore di playback assume valori molto alti: LayerP2P utilizza efficacemente la banda in *surplus* offerta dai peer istituzionali per andare in aiuto dei peer bisognosi. In figura 3.17 vengono mostrati i valori di PSNR: grazie al trattamento differenziato dei peer i valori di PSNR in

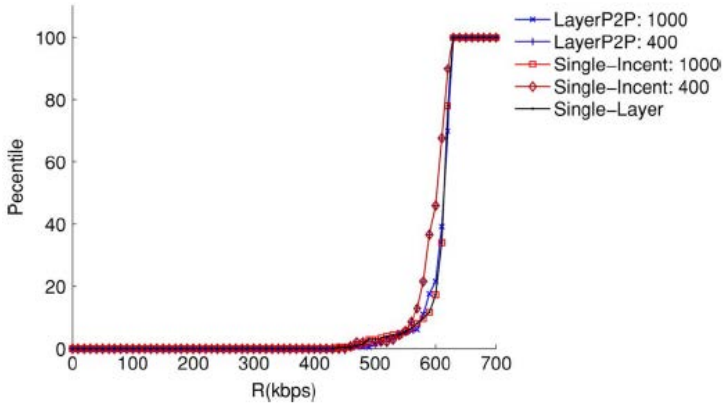

Figura 3.16: Distribuzione cumulativa del rate di download medio in ambiente underloaded

LayerP2P sono più elevati rispetto a quelli di entrambi i sistemi a singola descrizione. La differenza raggiunge valori di 2 dB.

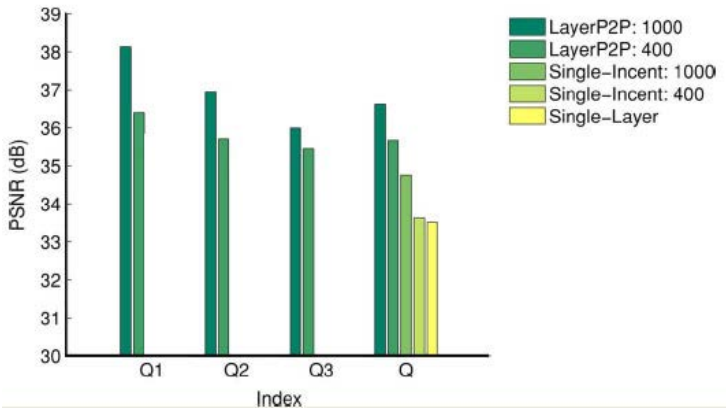

Figura 3.17: PSNR medio per i diversi layer in scenario underloaded

Scenario overloaded con free-ride peers

Vengono ora riportati i dati raccolti della sperimentazione nello scenario overloaded dove la banda media a disposizione del sistema è minore di quella richiesta dallo stream video completo. E' interessante notare dall'osservazione

dei grafici riportati come LayerP2P sia in grado di adattarsi a questo contesto gestendo in maniera differente le tipologie di peers presenti. Dalla figura 3.18 si nota come la tipologia di peer istituzionali abbia un andamento simile allo scenario underloaded; diversamente avviene per i peer residenziali che ricevono un rate video minore rispetto a quello riservato a chi contribuisce maggiormente. Ai free-ride peers viene riservato un download rate molto inferiore a quello richiesto anche solo per la ricezione del layer 1. Dall'osservazione della

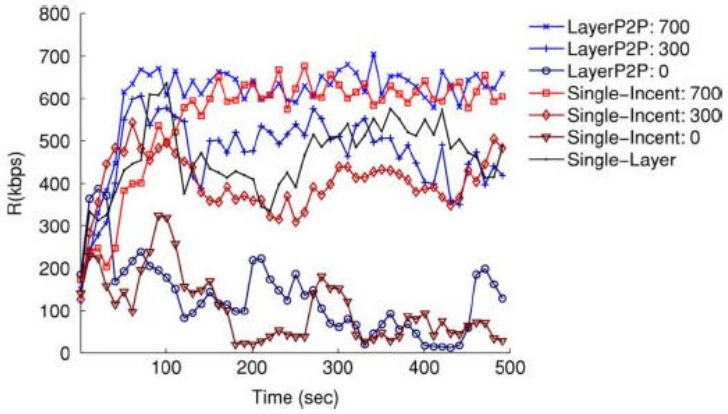

Figura 3.18: Playback rate di peer istituzionale e residenziale in ambiente overloaded

figura 3.19 si capisce chiaramente il comportamento differenziato verso i peer attuato da LayerP2P: più del 70% dei peer istituzionali riceve un rate superiore ai 550 kbps, più del 70% di quelli residenziali supera valori di 400 kbps ma, fattore molto importante, più del 90% dei peers free-ride dispone di un rate inferiore ai 300 kbps. Il trattamento privilegiato dei peers istituzionali è evidente anche dall'osservazione della figura 3.20 : il valori di PSNR attribuito a questa categoria (30.8 dB) è nettamente superiore rispetto alle altre tipologie. I peer residenziali sono in grado di ricostruire una buona qualità video grazie all'elevato valore di PSNR dato da Q1 (32.3 dB). Risulta invece inaccettabile la qualità offerta ai free-ride peer: 15.8 dB. Nello scenario a

singola descrizione con trattamento indistinto dei peer la qualità ricevuta è di soli 18.2 dB. I risultati forniti da questi esperimenti e dalla campagna di

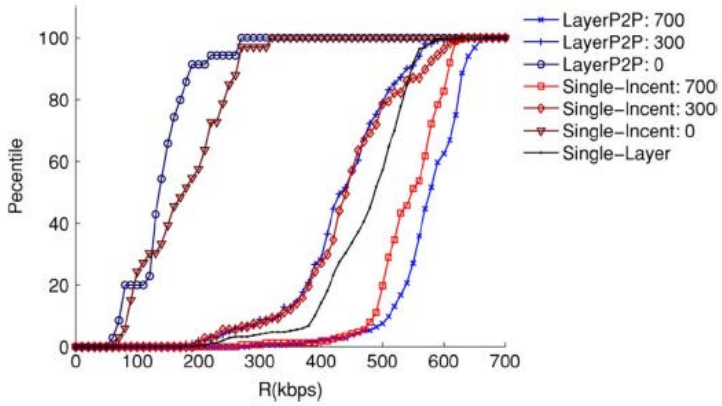

Figura 3.19: Distribuzione cumulativa del rate di download medio in ambiente overrloaded

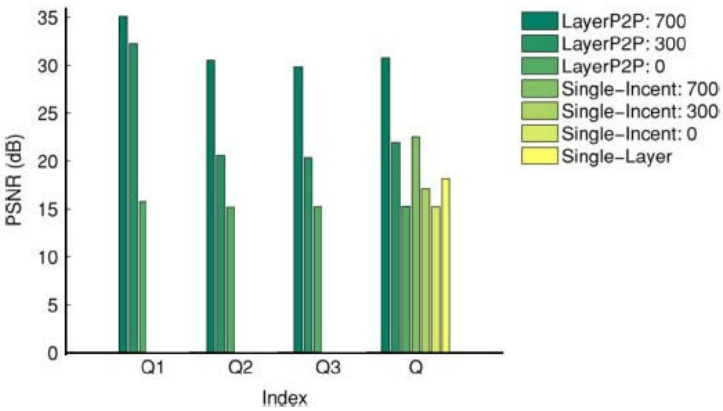

Figura 3.20: PSNR medio per i diversi layer in scenario overloaded

simulazioni mostrano come l'utilizzo di codec layered all'interno di un sistema di distribuzione live peer-to-peer rappresenti un'ottima scelta in grado di migliorare le prestazioni in termini di qualità video ricevuta dall'utente finale.

Lo studio dei codec, della compressione dell'informazione ha come scopo fondamentale la riduzione dello spazio necessario per archiviare oppure trasmettere attraverso una rete l'informazione. Nei paragrafi precedenti sono stati analizzati i principi fondamentali che agiscono sul contenuto informativo mentre ora viene illustrato un approccio diverso consistente nel massimizzare il flusso informativo in rete attraverso un concetto innovativo denominato *network coding.*

3.7 Network coding

La distribuzione di contenuti attraverso sistemi P2P rappresenta, ad oggi, una delle più importanti applicazioni di Internet. Tipicamente *files* di grandi dimensioni o flussi informativi devono essere trasmessi da una sorgente verso molti utenti che ne richiedono il contenuto. Questo problema può essere modellato attraverso la rappresentazione di un grafo nel quale i nodi del sistema sono rappresentati dai vertici mentre le connessioni tra i nodi sono identificate dagli archi rappresentati sia la connessione tra i nodi sia i parametri di capacità del link per la determinata connessione. Risulta evidente che il *throughput* del sistema intero è limitato dal minimo valore di capacità del link impiegato nella consegna dell'informazione dalla sorgente alla destinazione i-esima. Il miglioramento del *throughput* della rete può essere aumentato introducendo nella rete i principi della teoria del network coding. Con il termine network coding si fa rifermento ad un metodo che si propone di massimizzare il flusso informativo trasmettibile in rete. L'innovazione introdotta dal network coding è rappresentata dalla rottura con il tradizionale principio di routing (chiamato anche *store-and forward*)consistente nel trasferire pac-

chetti dati da una sorgente ad una destinazione attraverso nodi intermedi che hanno il solo scopo di instradare il pacchetto nella rete senza però poterlo modificare. Con l'utilizzo del network coding questa operazione di modifica è attuata: l'idea base consiste nell'inviare informazione codificata lungo gli archi del grafo. I nodi della rete sono i responsabili di questo processo: ricevono e producono dei pacchetti codificati che vengono poi inoltrati verso i propri vicini sino a raggiungere i destinatari che procederanno alla decodifica del materiale. Lo studio del network coding viene introdotto per la prima volta da Ahlswede in [33] dove viene analizzato il problema della massimizzazione del flusso in un contesto multicast dimostrando come l'approccio classico consistente nella creazione di alberi multicast non sia ottimale mentre con l'utilizzo dei codec è possibile raggiungere il massimo flusso compatibilmente alla topologia della rete stessa.

3.7.1 Aumentare il *throughput* in rete: esempio Butterfly

Attraverso l'illustrazione di un semplice esempio in questo paragrafo verrà mostrato come l'impiego di network coding possa portare ad un aumento considerevole del throughput. La rete che viene utilizzata in questo esempio è mostrata in figura 3.21. La rete è composta da una sorgente s, da due destinazioni chiamate *t1* e *t2* e da 4 nodi intermedi chiamati 1,2,3 e 4. La sorgente s vuole trasmettere due distinti blocchi informativi *b1* e *b2* verso *t1* e *t2*. Tutti i link di connessione tra i nodi presentano una capacità pari uno: sono in grado di trasferire una singola unità informativa per unità di tempo. Quello che si chiede al sistema è di distribuire i due pacchetti informativi in maniera tale che entrambi siano ricevuti contemporaneamente dalle destinazioni *t1* e *t2*. La destinazione *t1* può ricevere il contenuto *b1* ad esempio

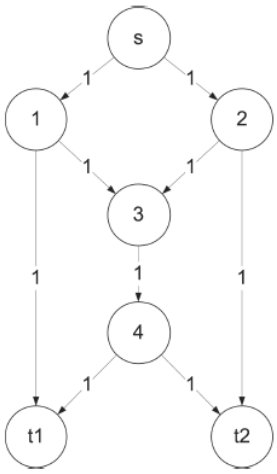

Figura 3.21: Schema di rete per esempio Butterfly

attraverso il nodo 1 (*s-1-t1*). L'unica capacità restante per trasmettere il pacchetto *b2* è quella offerta dal nodo 4: *b2* giungerà quindi a *t1* attraverso il percorso *s-2-3-4-t1*. La prima destinazione ha quindi ricevuto le due unità informative contemporaneamente. Analizziamo ora la seconda destinazione *t2*: entrambi i nodi 2 e 4 posseggono capacità inutilizzata per trasmettere informazione verso *t2* ma entrambi sono in possesso del pacchetto *b2* ! Poiché non esistono altri link la destinazione *t2* non è in grado di ricevere il pacchetto *b1*. I percorsi utilizzati per l'istradamento dei pacchetti sono solo una fra le possibili vie percorribili: altre soluzioni simili esistono ma nessuna permette la distribuzione di entrambi i pacchetti verso le due destinazione nella stessa unità di tempo. Per risolvere questo problema si utilizza il principio del network coding introducendo capacità elaborativa all'interno dei nodi intermedi permettendo lo svolgimento delle fasi di endocoding e decoding. Nella fase di encoding due o più pacchetti sono combinati fra di loro per crearne un terzo, avente le stesse dimensioni degli originali, contenente le informazioni di tutti i pacchetti che lo compongono. Per poter ottenere

le informazioni sui pacchetti originali, lato ricevitore occorre attuare la fase di decoding. Nell'esempio illustrato precedentemente queste due operazioni avvengono nella seguente maniera: i nodi 1 e 2 inoltrano i contenuti da loro posseduti (rispettivamente *b1* e *b2*) verso il nodo 3. Il nodo 3 codifica il contenuto in un unico pacchetto *b1+b2* le due informazioni da lui possedute e procede poi con l'invio del pacchetto codificato verso il nodo 4 che inoltrerà a sua volta tale informazione verso le destinazioni. A questo punto *t1* e *t2* sono in possesso di [*b1, b1+b2*] e [*b2, b1+b2*] rispettivamente. Attraverso una semplice operazione di decodifica le due destinazioni sono i grado di ricavare i contenuti originali dalle informazioni ricevute. Nella maniera più semplice l'utilizzo del network coding può essere implementato attraverso l'impiego dell'operatore logico XOR molto semplice da adoperare su sequenze di bit. Definendo *b1*=1011 e *b2*=1101 l'operazione di codifica effettuata dal nodo 3 consiste nell'eseguire l'operazione di XOR:

$$b1 \oplus b2 = 1011 \oplus 1101 = 0110 \tag{3.5}$$

La fase di decoding eseguita dalle destinazioni consiste nell'esecuzione di un'ulteriore operazione di XOR:

$$b1 \oplus (b1 \oplus b2) = 1011 \oplus 0110 = 1101 (= b2) \tag{3.6}$$

$$b2 \oplus (b1 \oplus b2) = 1101 \oplus 0110 = 1011 (= b1) \tag{3.7}$$

Questo semplice esempio dimostra chiaramente come sia possibile incrementare il throughput attraverso un utilizzo più intelligente delle risorse messe a disposizione dalla rete. Tuttavia il miglioramento di prestazioni avviene solamente attraverso un'accurata scelta delle tecniche di codifica - decodifica contestualmente alla rete in cui queste verranno applicate. La soluzione maggiormente utilizzata consiste nell'impego di *combinazioni lineari random*

[34], [35] che portano ad ottenere una soluzione asintoticamente ottima in qualsiasi tipo di rete.

3.7.2 Fase di codifica-decodifica attraverso l'utilizzo di combinazioni lineari random

Consideriamo il trasferimento di un file che viene diviso in un numero stabilito di blocchi come mostrato in figura 3.22 dove con p_k si denota il *k-esimo* blocco. Il pacchetto codificato, indicato con c_i, è una combinazione lineare

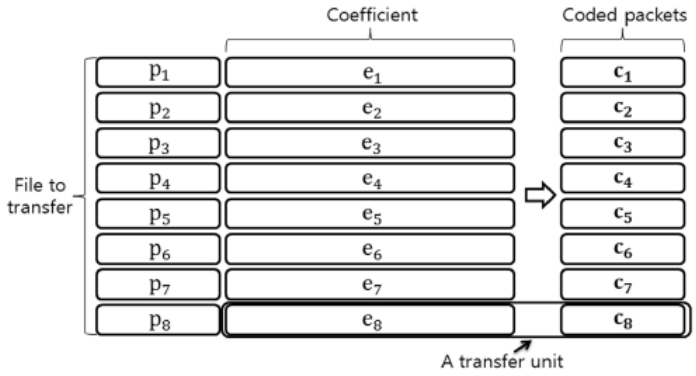

Figura 3.22: Matrice dei blocchi e dei coefficienti

dei blocchi che costituiscono il file: $c_i = \sum_{k=1}^{n} e_k p_k$ con n numero dei blocchi ed e_k vettore di elementi estratti in modo casuale all'interno di un prestabilito campo finito F nel quale vengono svolte tutte le operazioni aritmetiche; in molti casi pratici le dimensioni del campo finito utilizzato sono di 2^{16} . Il pacchetto codificato c_i è quindi inviato in rete congiuntamente al vettore dei coefficienti $[e_1, e_2, \ldots e_n]$. La figura 3.23 rappresenta graficamente questo blocco informativo. I nodi intermedi presenti nel percorso che unisce la sorgente alla destinazione una volta ricevuto il pacchetto non si limitano al mero inoltro ma lo analizzano e ricodificano aggiungendo le informazioni in loro

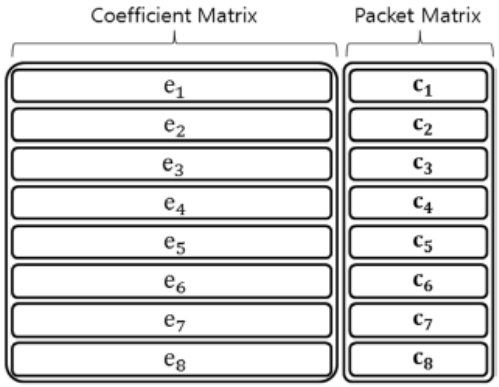

Figura 3.23: Blocco dei coefficienti codificati

possesso. Una volta giunto a destinazione il pacchetto codificato viene memorizzato e, per poter procedere alla decodifica, è necessario disporre di n unità informative aventi vettori dei coefficienti indipendenti. Più precisamente il nodo destinazione sarà in possesso dei vettori trasposti $E^T = [e_1, e_2, \dots e_n]^T$ e $C^T = [c_1, c_2, \dots c_n]^T$. Poiché i pacchetti codificati sono stati calcolati come $C = EP$ il ritorno al file originale P avviene attraverso la semplice operazione inversa: $P = E^{-1}C$. Questa operazione è possibile solamente se la matrice E è invertibile, quindi i coefficienti e_k devono necessariamente essere indipendenti tra di loro. La scelta di quale combinazione lineare ciascun nodo della rete debba utilizzare affinché il sistema risulti risolvibile rappresenta una scelta molto importante nella costruzione del sistema. In [37] viene analizzato l'utilizzo della selezione random dei coefficienti da parte di ciascun nodo in maniera indipendente e completamente decentralizzata. L'utilizzo di questa tecnica risulta molto semplice da implementare ma, con una certa possibilità, è possibile imbattersi nella selezione di combinazioni linearmente dipendenti. Questo fattore è determinato dalla dimensione del campo finito utilizzato per lo svolgimento delle operazione algebriche. La probabilità che due nodi indipendenti in possesso dello stesso blocco estraggano gli stessi coefficienti e

producano lo stesso output può accadere con una probabilità proporzionale alla dimensione del campo finito: in molti casi pratici una dimensione pari a 2^{16} risulta essere sufficientemente elevata per impedire eventi di "collisione". Alternativamente, come proposto in [36] , possono essere adoperati algoritmi deterministici come, ad esempio, l'algoritmo *polynomial-time* che esamina il comportamento di ciascun nodo presente in rete prima di determinare quale combinazione lineare impiegare. Questi algoritmi deterministici non sono però impiegabili in uno scenario P2P in quanto richiedono la conoscenza a priori dell'intera topologia della rete. Lo studio [38] con la proposta dell'utilizzo del random network coding ha aperto le porte all'impiego di questa tecnologia nel campo P2P.

3.7.3 Applicazione dei network coding nella distribuzione video live

L'applicazione della tecnologia del network coding alla distribuzione di contenuti video attraverso l'impiego di reti P2P deve tenere in forte considerazione le specificità imposte dal contenuto multimediale come, ad esempio, i stringenti vincoli in termini di ritardi, l'elevata necessità di banda, la necessità da parte del sistema a reagire a situazioni in cui si assiste a perdite di pacchetti ed eventi che mutano la topologia della rete. Tutti questi parametri devono essere tenuti in considerazione quando si pensa allo sviluppo di un sistema che , oltre ad assicurare il mantenimento di queste garanzie, possa risultare non eccessivamente complesso ed oneroso. L'utilizzo del network coding, se adeguatamente progettato, è in grado di fornire una soluzione che trae vantaggio nell'esser utilizzata in ambito P2P dove il miglioramento del throughput, la riduzione dei ritardi end-to-end sono elementi fondamentali che portano al miglioramento della qualità percepita dall'utente finale. Di

seguito vengono presentati studi recenti che focalizzano la propria attenzione sull'impiego di network coding all'interno di scenari P2P live streaming.

Network coding in sistemi P2P live streaming

Lo studio [39] propone l'analisi di un sistema P2P live streaming basato sull'utilizzo del principio del network coding. L'idea alla base di questo sistema è molto semplice e rispecchia i principi della codifica-decodifica attraverso l'uso di combinazioni lineari scelte casualmente illustrato nel paragrafo precedente. La sorgente video divide lo stream prodotto in segmenti di *group of pictures*(GOP) e durante la trasmissione del contenuto dalla sorgente alla destinazione ciascun nodo intermedio utilizzerà il network coding per migliorare le performance del sistema in termini di velocità di download dei segmenti e miglioramento del ritardo di playback oltre a fornire una maggior robustezza al sistema a fronte di cambiamenti topologici repentini causati dall'abbandono o arrivo di peers. Il sistema in analisi utilizza un semplicissimo algoritmo di scheduling che basa le proprie decisioni sulla conoscenza locale della rete appresa attraverso la conoscenza di un numero limitato di vicini scelti in maniera casuale. La figura 3.24 rappresenta l'architettura di un classico peer: particolare attenzione viene rivolta ai due blocchi network coding responsabili della codifica e decodifica dell'informazione presente nel buffer del peer. Il blocco di network coding disposto tra il buffer e l'interfaccia di rete svolge il ruolo di codifica del materiale presente nel buffer e precedentemente ricevuto attraverso l'interfaccia di rete dai propri vicini; il blocco di network decoding collocato tra il buffer ed il media player svolge invece il ruolo di decodificatore fornendo al player il flusso video originale in modo tale da essere riprodotto. Lo studio condotto compara il sistema testé descritto con CoolStreaming [40] evidenziando i vantaggi di un sistema che utilizza network codec verso uno

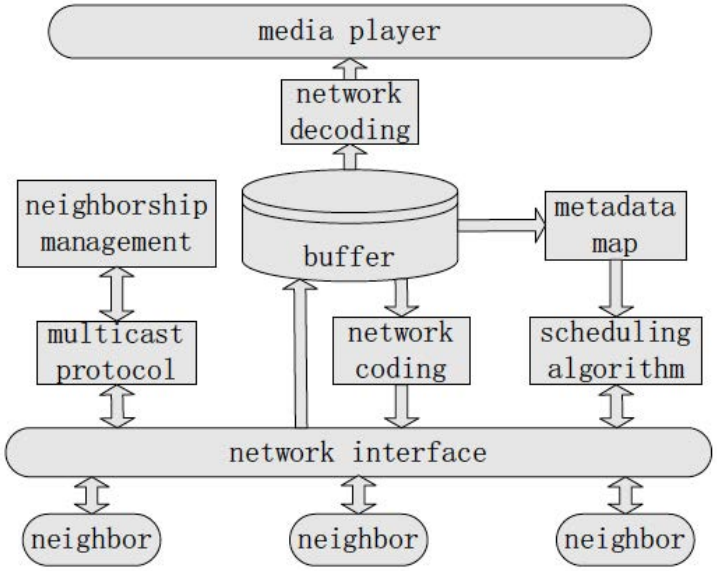

Figura 3.24: Architettura del peer

che non né utilizza affatto. Lo studio comparativo è avvenuto attraverso la simulazione in uno scenario dove le capacità di upload e download dei singoli nodi sono misurate come quantità di blocchi informativi trasmettibili in una unità di tempo, il tempo di playback (tempo di attesa tra l'istante di join e l'istante di riproduzione del video) è anch'esso misurato in unità di tempo definite round. All'inizio di ciascun round i peers contattano i propri vicini e ricevono le informazioni sul materiale da loro posseduto ed avviano le procedure per la distribuzione del contenuto. Il primo fattore comparato nello studio è il cosiddetto *continuity index*: esso rappresenta il numero di blocchi che giungono a destinazione prima della fine del periodo di loro validità, è l'indice che quantifica percentualmente il numero di blocchi utilizzabili per la riproduzione pesati sul numero di blocchi totali ricevuti. In tabella 3.25 sono riportati i parametri impiegati nella simulazione. I grafici in figura 3.26 riportano sull'asse delle x la scansione in 5 periodi temporali della durata

Join rate	1 peer/round
Numero di blocchi per segmento	18
Capacità di upload e download per ciascuno peer	24 blocchi
Ritardo di playback	15 rounds
Streaming rate	18 blocchi/round
Banda di upload server	74 blocchi
Ricerca di nuovi vicini	Ogni 4 rounds
Durata di ciascuna simulazione	300 round ripetuti 10 volte

Figura 3.25: Parametri utilizzati nella campagna di simulazione

della simulazione e sull'asse delle y il valore del *continuity index* variabile da 0 a 1 al variare del numero di vicini. Il *continuity index* assume sempre valori maggiori nel caso di impiego di network codec e miglior all'aumentare del numero di vicini questo perché la probabilità di ricevere il blocco desiderato aumenta all'aumentare del numero di fornitori. Questi risultati sono fortemente legati ai dati riportati in figura 3.27 dove viene illustrato il valore del *buffer filled ratio* che sta ad indicare la percentuale di riempimento dei buffer e quindi è un ottimo indicatore della velocità con cui i blocchi sono ricevuti. Dall'osservazione di questi grafici si evince come l'utilizzo dei network coding incrementi la velocità di download. Il grafico 3.28 riporta l'andamento del continuity index al variare al ritardo di playback: questo migliora all'aumentare del ritardo poiché maggiore è il tempo a disposizione per ricevere i dati e successivamente riprodurli. Quello che risulta evidente è come l'utilizzo di network coding permetta di ottenere migliori prestazioni rispetto al non utilizzo e porta ad avere un buone *performances* anche in situazioni in cui il tempo di playback è ridotto. L'ultima analisi condotta nello studio in esame valuta l'andamento dell'indice di continuità al variare del *rate* di streaming: quando lo streaming *rate* è relativamente basso i due modelli si equiparano mentre all'aumentae del *rate* le prestazioni offerte

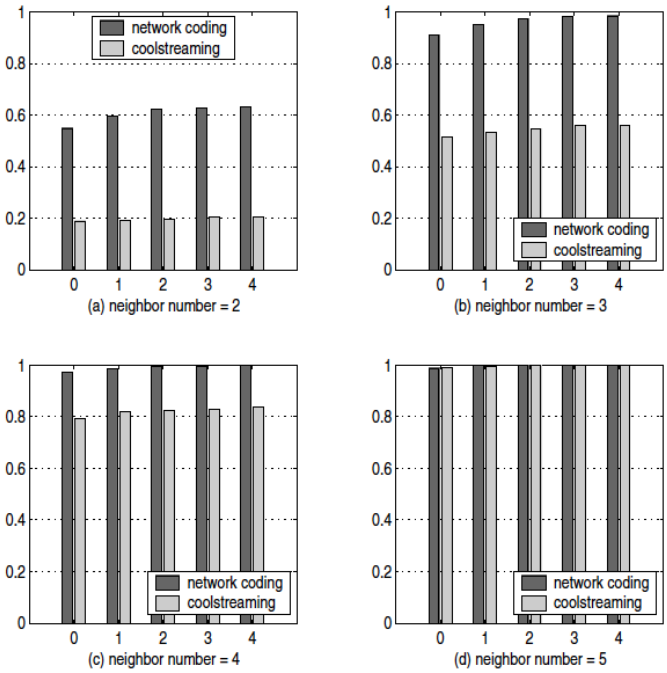

Figura 3.26: Continuity index al variare del numero di vicini in sistema in presenza-assenza di network coding

dal sistema che utilizza network coding sono nettamente superiori: questo evidenzia la capacità del sistema ad utilizzare tutte le risorse a disposizione con un conseguente aumento del throughput e della velocità effettiva con cui vengono reperiti i blocchi informativi. Andamento riportato in figura 3.29

3.7.4 R^2: Random Push con Random Network Coding nella distribuzione P2P live

Usualmente nei sistemi P2P live streaming il flusso video da trasmettere viene suddiviso in segmenti che vengono scambiati tra i peers solitamente in modalità *pull* in presenza di reti overlay di tipo mesh. In R^2 [79] ogni segmento video è ulteriormente suddiviso in n blocchi $[b_1, b_2...b_n]$ ciascuno di dimensioni fisse pari a k bytes. L'impiego del network coding avviene attraverso l'utilizzo

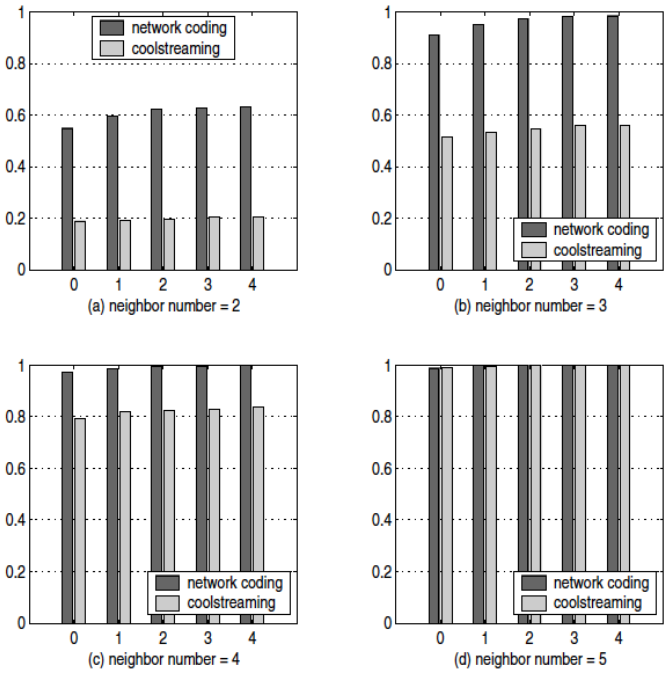

Figura 3.27: Buffer filled ratio al variare del numero di vicini

di random linear coding applicati, da parte del trasmettitore, sui blocchi che compongono il segmento come illustrato in figura 3.30 riducendo in questa modo la complessità delle operazioni ad una parte limitata dello stream. La scelta dei blocchi da codificare viene operata in maniera totalmente casuale: il sistema R^2 adopera un sistema di distribuzione *random push* che permette di sfruttare al meglio le potenzialità offerte dall'impiego dei network codec. Il generico nodo, una volta codificato il blocco, lo invia verso i propri vicini senza aver ricevuto nessuna esplicita richiesta. La diffusione del flusso video fa si che i peers accumulino nei propri *playback buffers* i blocchi codificati: il peer p ricodificherà quindi i blocchi a sua disposizione scegliendo casualmente nuovi coefficienti di codifica ed altrettanto casualmente trasmetterà le nuove informazioni. Il fatto che più entità trasmettano in totale autonomia ed indipendenza rispetto agli altri peers informazione verso i riceventi ha

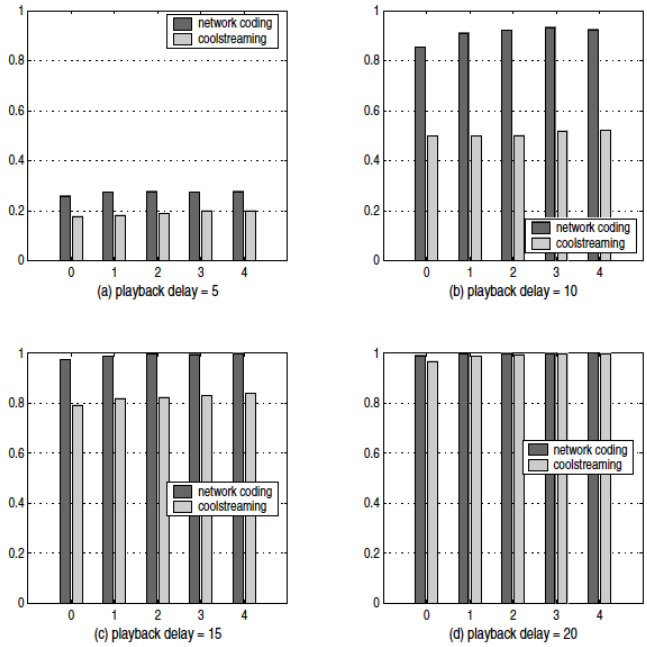

Figura 3.28: Andamento del continuity index al variare al ritardo di playback

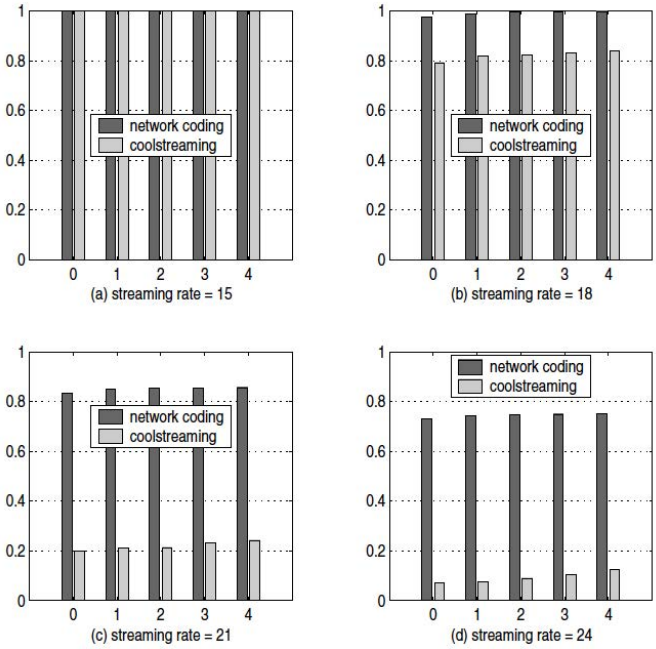

Figura 3.29: Andamento dell'indice di continuità al variare del rate di streaming

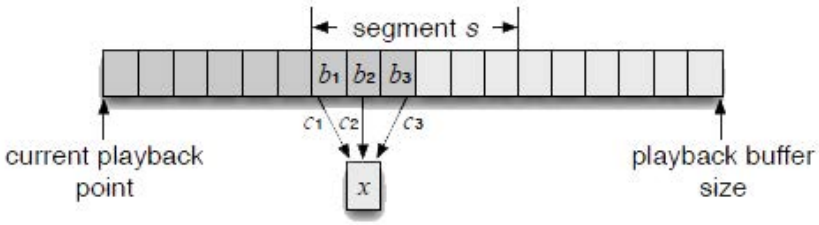

Figura 3.30: Esempio di operazioni di coding operate dal peer p su 3 blocchi facenti parte del segmento s composto da 6 blocchi totali

portato alla definizione del termine *the perfect collaboration*. Il processo di decodifica da parte del ricevitore viene attuato atraverso l'utilizzo del processo *Gaussian-Jordan elimination* che permette il computo progressivo dei blocchi senza la necessità di dover aspettare la ricezione completa di tutti gli n blocchi codificati. L'impiego dell'eliminazione Gaussian-Jordan permette inoltre di determinare la dipendenza lineare di un blocco da uno già posseduto. Se tale dipendenza viene riscontrata il blocco è elimato poichè inutile in quanto non contiene informazione aggiuntiva. Le caratteristiche stesse della distribuzione live impongono che i segmenti più vicini alla soglia di playback siano più importanti, abbiano priorità rispetto a quelli che si trovano temporalmente distanti dalla soglia di riproduzione. Per tale motivo in R^2 nel playback buffer è presente una zona chiamata *priority region* (fig 3.31): il peer p che si trova ad aver blocchi mancanti in questa regione invia verso i suoi fornitori una particolare richiesta, indicando l'urgenza dei blocchi mancanti nella regione prioritaria. I fornitori a questo punto procederanno sempre all'estrazione casuale dei blocchi ma limitando l'intervallo di scelta all'interno della *priority region*: tale operazione ha senso poichè in R^2 è implementato un meccanismo che permette la sincronizzazione dei buffers degli utenti. La scelta dei peers verso cui inoltrare i blocchi avviene in maniera casuale ma limitata ad un determinato numero di utenze tra quei peer che si

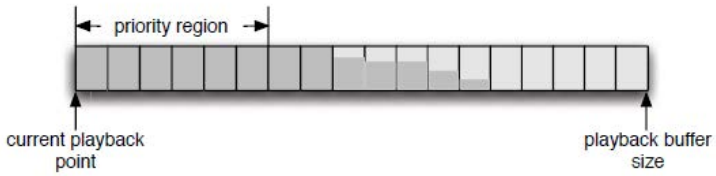

Figura 3.31: Playback buffer e priority region

sono maggiormente distinti nello scambio reciproco di informazioni. La campagna di simulazioni condotta in [79] dimostra come l'impiego di Network Codec abbinato al sistema random push porti a notevoli vantaggi in termini di:

- Ritardo di buffer iniziale: gli utenti beneficiano di un ritardo iniziale inferiore rispetto agli usuali sistemi grazie alla *perfect collaboration* frutto dei network coding e del processo di sincornzzazione.
- Riduzione delle banda: l'utilizzo di sorgenti multiple, il non impiego di messaggi di richiesta materiale, e l'utilizzo di network coding fanno si che la banda a disposizione dei peers sia maggiore rispetto a quella nei sistemi *pull-based*
- Miglior resistenza a cambiamenti topologici grazie alla presenza di fornitori multipli
- Miglior playback grazie all'implementazione della zona prioritaria all'interno del buffer.

Capitolo 4

Sistemi di incentivo e cooperazione tra peers e livelli di rete

Meccanismi d'incentivo all'interno di sistemi P2P rappresentano componenti essenziali per il rafforzamento della cooperazione tra i peers. I sistemi di distribuzione di contenuti multimediali live traggono ispirazione dai meccanismi di incentivazione adottati nel P2P *file-sharing*. Tuttavia, la natura del flusso multimediale live impone forti limitazioni temporali insite nella natura stessa del live video: diversamente dai *files*, i flussi video live vengono utilizzati in real-time, i blocchi di cui sono composti devono essere ricevuti entro il periodo di loro validità affinché siano ancora utilizzabili nel processo di visualizzazione del video. I principali meccanismi alla base della cooperazione tra peers si fondano sulla valutazione del grado di affidabilità fornito dal peer, dalla banda offerta e dal livello di partecipazione al processo di distribuzione. L'utilizzo d'incentivi all'interno dei sistemi P2P è un metodo per poter gestire in maniera più intelligente le risorse presenti in rete all'interno di un

ambiente molto eterogeneo e in continua evoluzione dove le risorse variano a causa delle innumerevoli variazioni topologiche dovute ai *join* e *leave* a cui il sistema è costantemente sottoposto. Una maggior cooperazione tra i peers ed un miglior criterio di scelta dei vicini con i quali interagire porta ad un miglioramento generale della qualità che il sistema è in grado di offrire. In letteratura esistono vari studi che esaminano i sistemi di incentivo da adottare in un sistema P2P: le principali tecniche che permettono una differenziazione nel trattamento rivolto ai peers possono essere così classificate:

- *Currency-payment method o score method:* lo scambio di risorse all'interno della rete viene gestito attraverso un sistema basato sul principio del pagare-guadagnare. Un generico peer aumenterà il denaro in suo possesso collaborando alla diffusione della risorsa mentre spenderà denaro all'atto di richiedere servizi ai suoi vicini. Questo metodo incentiva la collaborazione poiché spinge ad ottenere maggior guadagni e permettersi quindi "vicini più costosi" in grado di fornire un servizio migliore. In [41] viene proposto un sistema di incentivi basato sul paradigma del *payment method* applicato ad un sistema peer-to-peer live streaming. Il cuore di questo sistema risiede nella competizione che nasce fra i peers nel guadagnare punti e questo spinge tutti i partecipanti a collaborare per aver una miglior qualità: i peers in possesso di un capitale maggiore sono liberi di scegliere i propri fornitori. Nello studio viene realizzato un algoritmo distribuito in grado di regolare la competizione tra i peers e massimizzare la qualità che ciascun utente può ricevere.

- *Taxation model:* data l'eterogeneità in cui i sistemi P2P si trovano ad operare, questo modello propone un sistema di tassazione che induce i peers "più ricchi" (che dispongono di maggior risorse) ad aiutare i peers che si trovano in una situazione più sfavorevole della loro in maniera

tale da ridistribuire la ricchezza in maniera equilibrata all'interno della rete.

Come più volte ribadito la forza dei sistemi P2P sta nella capacità di creare una rete overlay al di sopra della rete fisica underlay: una migliore cooperazione tra questi due livelli porta a nuove metodologie per la selezione dei peers. Questo processo collaborativo ha lo scopo di ridurre il carico di rete a cui gli ISPs (*Internet Service Providers*) sono costantemente sottoposti se le fasi di creazione della rete overlay non tengono in considerazione la distribuzione geografica. Nel capitolo 2 si è focalizzata l'attenzione sulla creazione di topologie ibride che uniscono i vantaggi offerti dai classici paradigmi tree e mesh. Ora in questo capitolo vengono presentati i sistemi in grado di fornire incentivi alla collaborazione fra i peers ed architetture in grado di fra comunicare i due layers (underlay e overlay) allo scopo di ottenere, attraverso scelte più coordinate ed intelligenti, una migliore qualità offerta all'utente finale.

4.1 Score-based system

In [42] viene proposto un sistema d'incentivo score-based in grado di fornire un servizio di differenziazione nel processo di selezione dei peers. Gli utenti che partecipano in maniera attiva al sistema vengono ricompensati attraverso la scelta dei peer migliori e maggiormente performanti diversamente, chi non contribuisce al processo di distribuzione, è limitato nella scelta dei propri vicini. Il sistema in esame è denominato PALM [42] e ciascun utente che ne fa parte decide in maniera autonoma il grado di contributo che vuole offrire proporzionalmente al livello di beneficio che ne vuole trarre in termini di qualità. Il livello di contributo x_i dell'utente i-esimo viene convertito in un valore S_i che indica un punteggio (*Score*): la funzione che assegna i punteggi

ai vari utenti è basata sul calcolo della percentuale di byte trasmessi su quelli ricevuti. La selezione da parte di un peer richiedente del proprio fornitore avviene valutando il valore fornito dalla funzione punteggio: per esempio un peer potrà avere come fornitore un peer in possesso del suo stesso score o di un valore inferiore. Il risultato della scelta dei peer fornitori viene rilevato dalla funzione Q così definita:

$$Q = \sum_{k=1}^{T} \frac{V_i}{T} \tag{4.1}$$

Questo valore identifica la qualità ottenuta durante una sessione di streaming composta da T pacchetti. V_i è una variabile che assume valore 1 nel caso in cui il pacchetto sia ricevuto all'interno del suo periodo di validità, 0 altrimenti. Il valore Q può essere espresso come una funzione del fattore di contribuzione oppure del punteggio ottenuto. Quando un peer entra per la prima volta nel sistema il punteggio è zero e riceve un servizio di tipo best-effort $Q_{BEST-EFFORT} = Q(Si = 0) = 0$. La qualità ricevuta in questa particolare situazione varia in base alle condizioni di carico del sistema e dalle politiche di scheduling adottate. L'unica maniera per aumentare la qualità ricevuta è quella di partecipare attivamente alla distribuzione del contenuto video ed aumentare il proprio punteggio.

4.2 Cooperazione tra livello di trasporto e aplicativo: il progetto NAPA-WINE

Le applicazioni P2P live video streaming stanno avendo sempre maggior successo e diffusione: questo aumenta in maniera esponenziale il traffico generato e lo stress a cui sono sottoposte le infrastrutture che creano la rete underlay, il basamento su cui poggia la rete overlay. I *content provider* e gli operatori

di rete vedono nel P2P un potente mezzo per raggiungere un ampio bacino d'utenza ma, al contempo, sono fortemente preoccupati riguardo all'enorme traffico generato. Serve quindi una forma di cooperazione tra questi due livelli, occorre sviluppare un sistema che faccia da interfaccia e permetta una migliore gestione della disposizione degli utenti, una gestione che tenga in considerazione ciò che sta al disotto della topologia overlay. Questa è la mission del progetto NAPA-WINE (*Network Aware P2P-Tv Application over WIse NEtworks*) [43] . La cooperazione tra ISPs e applicazioni P2P è già stata intrapresa [44] , sono nati gruppi di ricerca come il P4P Project (*Provider Portal for P2P applications*) ma queste ricerche sono principalmente sviluppate nel campo del campo del *file-sharing* e non in realtà più complesse come il P2P live streaming. Come ben noto i peers che costruiscono la topologia overlay a livello applicativo scambiano il contenuto informativo con i loro vicini attraverso la rete IP underlay; entrambi i livelli di rete eseguono operazioni di routing ed inoltro dei pacchetti: quella underlay agisce sui pacchetti IP mentre la rete overlay sui segmenti di video comunemente denominati *chuncks*. Quello che il progetto NAPA-WINE presenta è un'innovativa architettura che si propone di ottimizzare la qualità percepita dall'utente finale minimizzando l'impatto che una scorretta dislocazione dei peers e la non conoscenza delle condizioni della rete di trasporto può causare. Un semplice esempio è rappresentato dalla riduzione del traffico generato tra AS diversi: una disposizione intelligente degli utenti può evitare lo spreco di risorse semplicemente creando delle connessioni overlay che tengano presente la configurazione underlay. Quest'architettura non impone l'utilizzo di nessuna topologia specifica ed ha come scopo ultimo il miglioramento della qualità e l'implementazione di una struttura in grado di raggiungere gli standard offerti dalle trasmissioni HD. Come mostrato in figura 4.1 la coope-

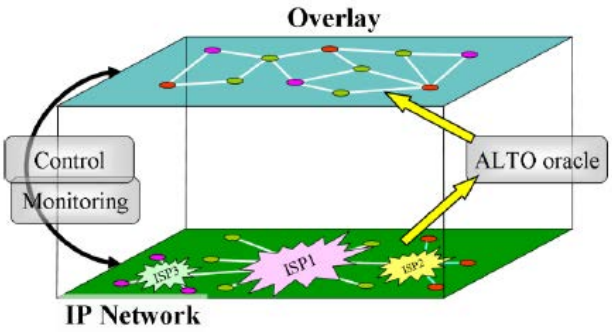

Figura 4.1: Livello underlay (trasporto) e overlay (applicativo) che comunicano tra di loro attraverso scambio di messaggi.

razione tra i due livelli avviene attraverso lo scambio di messaggio di controllo e monitoraggio e l'impiego, non strettamente necessario ma molto utile, di ALTO (Application Layer Traffic Optimization) [45] che fornisce una utile interfaccia di comunicazione tra il mondo underlay e le applicazioni P2P. I blocchi che costituiscono l'architettura NAPA-WINE sono illustrati in figura 4.2 e la loro funzione descritta nei paragrafi sottostanti. Un elemento fondamentale è rappresentato dal Monitoring Layer che permette all'applicazione di avere una costante visuale sia sulle condizioni della rete IP sia della qualità percepita dall'utente.

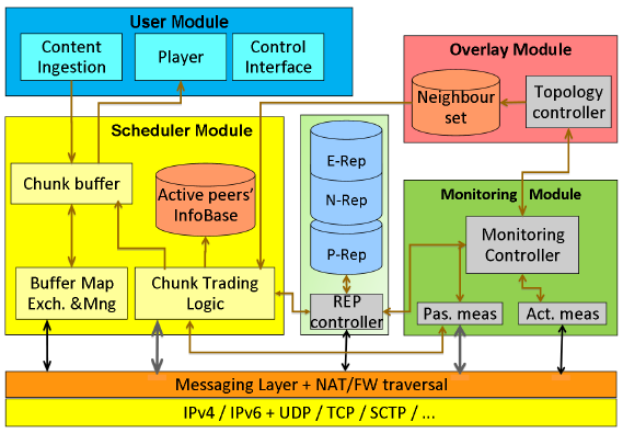

Figura 4.2: Blocchi logici componenti l'architettura NAPA-WINE

User Module

Il "modulo utente" implementa le funzioni che interfacciano l'utente all'applicazione: è l'entità responsabile del processo di codifica-decodifica, della conversione del video iniziale in chuncks e, grazie all'utilizzo di un'interfaccia standard, è in grado di impiegare qualsiasi CODEC. Una volta ricevuti i chunks lo *user model* assembla i flussi video e audio, li sincronizza, li decodifica e li mostra all'utente finale. Un algoritmo per la misura del QoE è costantemente attivo e monitora l'andamento della qualità del flusso video.

Scheduler Module

Questo modulo rappresenta il cuore del sistema di distribuzione: è il responsabile dell'inoltro e ricezione dei segmenti video da e verso la rete da un lato e, al contempo, fornisce il materiale al modulo utente. All'interno di questo modulo sono ospitati algoritmi di scheduling e chunk selection che agiscono avendo conoscenza della situazione della rete grazie alle informazioni provenienti dai moduli *repositories* e dal database dei peers attivi.

Overlay Module

Il modulo overlay è il componente responsabile nella scelta, gestione ed aggiornamento delle relazioni tra vicini. Il difficile compito della selezione dei vicini avviene con l'ausilio dei *repositories* nei quali sono contenute informazioni riguardanti i peers e la rete.

Repositories

I *repositories* sono dei database nei quali l'informazione di ciascun peer è condivisa con il resto degli utenti. L'informazione contenuta proviene dal

modulo di monitoring ed è condivisa ed accessibile da tutti in maniera tale da permettere la conoscenza non solo localmente ma a grande scala. I *respositories* sono l'elemento chiave che aiuta il sistema del processo di selezione dei peer. In base all'informazione che contengono tre tipologie sono identificate:

- P-REP (Peer repository): contiene informazioni riguardanti lo stato dei peer e le misurazioni end-to-end
- N-REP (Network repositories): archivia le informazioni riguardanti lo stato della rete congiuntamente alle informazioni contenute nel P-REP. In letteratura l'utilizzo di coordinate virtuali [46] e tomografie della rete [47] hanno permesso di mappare il posizionamento dei peer all'interno di spazi virtuali le cui distanze sono misura della distanza a livello di rete interposta tra la coppia di peers. In N-REP sono contenute le informazioni generate da questi tipi di sistemi che vengono poi utilizzate per la gestione topologica.
- E-REP (External repository): è impiegato per il mantenimento delle informazioni provenienti dall'esterno: informazioni provenienti dall'operatore di rete quali qualità dei collegamenti fra routers, routing tables, topologia della rete. E-REP è inoltre collegato ai servizi forniti da ALTO: informazioni riguardanti il posizionamento geografico delle reti, i costi di trasmissione da sostenere e gli strumenti necessari ad avere una vista dettagliata sul mondo underlay.

Messaging layer

Rappresenta il modulo che fornisce gli strumenti necessari all'invio e ricezione di dati verso gli altri peers. Offre inoltre uno strumento per la gestione del

NAT (Network Address Translation).

Monitoring module

La conoscenza di quello che avviene in rete, della situazioni in cui si trovano i peers, della banda disponibile e molto altro è fondamentale per le scelte prese dal sistema. Il modulo di monitoring serve all'osservazione e alla misurazione dell'andamento del sistema stesso. Le misurazioni avvengono in due modi: uno passivo, consistente nell'osservazione dei messaggi scambiati fra gli utenti, e uno attivo attraverso l'invio di messaggi sonda. I dati ottenuti da queste misurazioni forniscono un'informazione utile e preziosa sia per la costruzione e per il mantenimento della topologia overlay sia per le decisioni nel processo di scheduling. Le misure che vengono offerte al sistema riguardano:

- Ritardi tra i peers (RTT, ritardi di Jitter)
- Loss probability
- Capacità del collegamento e disponibilità
- Numero di collegamenti da attraversare

Il fine dell'ottimizzazione consiste nel permettere una comunicazione efficiente tra i peers che permetta loro una redistribuzione del contenuto live in real-time, nel minor tempo possibile. Per poter raggiungere questo scopo è necessario l'utilizzo di accurati algoritmi che decidano come costruire il vicinato, verso quali peer inoltrare il contenuto video, quali chunks inviare e quando. Le prime due tematiche riguardano la gestione della topologia mentre le altre si riferiscono a tipiche problematiche di scheduling. La struttura della rete overlay è data dal risultato degli algoritmi distribuiti che gestiscono e mantengono le relazioni di vicinato tra i peers: quando un peer accede per

Index	Accuracy	Usage / Criticalities
Hop Count	High	Topology optimization
Packet Loss	High	Triggering scheduling & topology update and QoE monitoring
Chunk Loss	High	Triggering scheduling & topology update and QoE monitoring
Delay Measurements		
Latency	> 10ms	Source and stream absolute time alignment / Clock synchronization below NTP precision is hard to obtain
RTT	< 1ms	Chunk and peer scheduling / Timestamping accuracy limits precision
Jitter	< 1ms	Delay jitter may be an indication of congestion
Bandwidth Measurements		
Mid term Throughput	High	Neighborhood selection and chunk/peer scheduling
Capacity	Low	Pacing chunk scheduling for altruistic peers to avoid clogging a PHY bottleneck (e.g., ADSL links beyond a LAN)
Available Bandwidth	Low	Fundamental to avoid congestion in the network / Difficult to achieve high accuracy due to correlations and the lack of efficient algorithms to perform the measure

Figura 4.3: Misurazioni offerte dal monitoring module e grado d'accuratezza

la prima volta al servizio di distribuzione seleziona un gruppo iniziale di vicini che ottimizzerà poi dinamicamente nel tempo. Questa fase iniziale (detta di bootstrapping) viene supportata dalle funzionalità offerte da ALTO. L'esempio riportato in figura 4.4 mostra il caso in cui il client 2 voule entrare a far parte del sistema: il protocollo di gossip fornisce una lista di possibili vicini in grado di fornirgli ciò che desidera. Nello specifico la scelta è tra il client 1 e il client 3. A questo punto della scelta interviene il servizio ALTO: il client 2 si rivolge ad ALTO per conoscere la situazione dei due candidati a livello di rete di trasporto. La scelta del patner verrà quindi effettuata tenendo in considerazione le informazioni ricevute lato ISP: queste possono riguardare la dislocazione geografica dei peers, lo stato dei link di trasporto, i costi operazionali o altre politiche generiche. Nello specifico esempio la scelta ricade sul peer 3 in quanto fisicamente collocato nella stessa rete del peer 2: questa scelta ha risparmiato inutile traffico tra AS diversi. Attraverso la con-

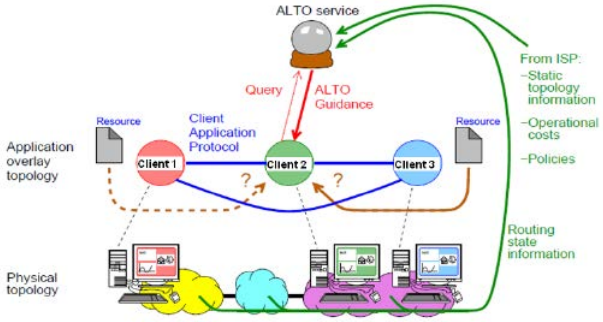

Figura 4.4: Fase di bootstrapping attraverso l'utilizzo dei servizi forniti da ALTO

sultazione dei *repositories* è sempre disponibile un'informazione aggiornata sullo stato della rete che permette la creazione ed il mantenimento di una struttura overlay molto efficiente. L'impiego di diverse metodologie per la selezione dei vicini è illustrato in figura 4.5 : nel caso (A) viene utilizzato un criterio di selezione casuale in (B) i peers vengono selezionati in base all'AS (Autonomous System) di apparteneza (informazione contenuta in E-REP); in (C) la selezione utilizza come informazione la banda disponibili in upload (informazione depositata in P-REP); in (D) i peers sono selezionati tenendo in considerazione sia l'informazione su AS sia quella sulla banda. Creata la topologia ciascun partecipante avvia la procedura di scambio di chunks: la cooperazione di ciascun peer nel processo distributivo e corrette procedure di scheduling basate sulle informazioni a disposizione del sistema permettono la ricezione di pacchetti in tempo e conseguentemente la visione di un flusso video fluido. In un sistema *unstructured* la programmazione della distribuzione dei contenuti implica la trasmissione di messaggi di segnalazione tra i peers: queste operazioni sono svolte in NAPA-WINE dal blocco Information Signaling Strategy block. Il suo compito è quello di distribuire le buffer-maps verso gli altri utenti, riceverle ed archiviarle all'interno dei buffer locali del peer e decidere se diffondere, attraverso l'utilizzo di protocolli di gossip, l'informa-

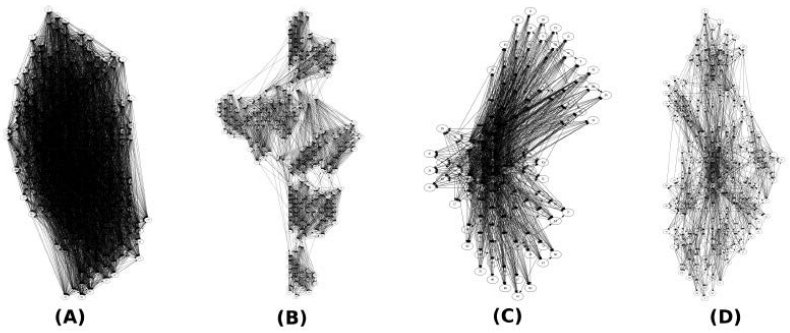

Figure 5: A) completely randomly chosen neighbors, B) neighbors chosen based on locality information provided by E-REP, C) neighbors chosen based on upload bandwidth peer information, D) combination of B) and C).

Figura 4.5: Diverse configurazioni di rete overlay in base al criterio di scelta dei vicini.

zione da lui in possesso. In NAPA-WINE sono stati utilizzati vari algoritmi di scheduling che spaziano dalla pura selezione random ad algoritmi più sofisticati che tengono in considerazione la durata residua del segmento video e le condizioni della rete e dei peers. L'utilizzo delle informazioni provenienti dai repositories nelle scelte prese dallo scheduler porta ad un notevole miglioramento delle prestazioni come illustrato in figura 4.6 : il grafico riporta le prestazioni ottenute in termini di ritardo di consegna in una rete overlay composta da 1000 peers ciascuno dei quali ha 20 vicini. Ogni linea colorata rappresenta una tipologia di scheduling:

- RUc/RUp: selezione causale del chuck ed inoltro in maniera casuale
- RUc/BAp: un chunk selezionato casualmente è inviato verso un peer richiedente preferendo quelli che dispongono di maggior banda.
- LuC-RUp: il segmento più giovane è inviato in maniera random.
- DLc-ELp: algoritmo basato sulla valutazione della deadline [48] . Il chunk che possiede la minor deadline (significa che da più tempo è pre-

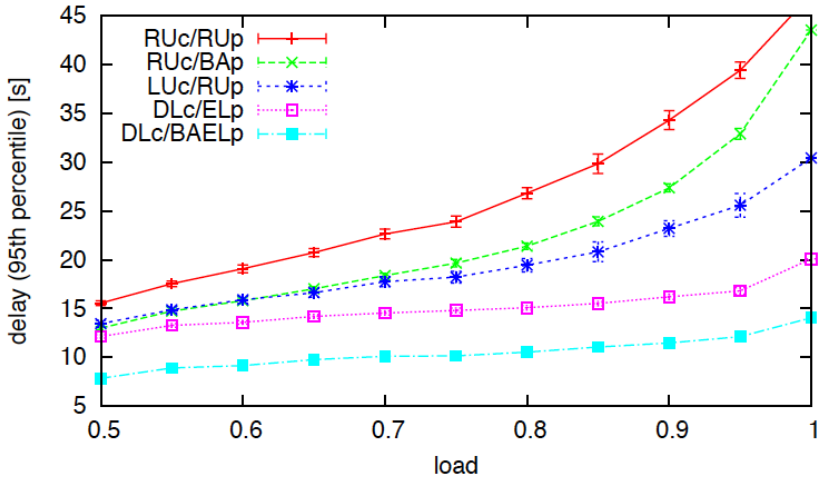

Figura 4.6: Variazioni di prestazioni in base al modello di scheduling adoperato

sente nel sistema ma non è stato ancora diffuso ad un numero sufficiente di peers) viene inviato verso il peer che in quel momento è nel miglior stato per poterlo ricevere e diffondere.

- DLc/BAELp: come caso precedente ma nel processo di selezione del peer si tiene in considerazione la banda a disposizione.

Dall'osservazione dei risultati illustrati è facilmente comprensibile come gli scheduler più complessi siano in grado di fornire prestazioni superiori rispetto agli altri ed incrementare le *performance* di un fattore tre in presenza di scenari in cui il carico della rete è elevato. La cooperazione tra il livello applicativo e la rete di trasporto abbinata ad una intelligente diffusione dell'informazione permette la creazione, gestione e manutenzione di topologie overlay in grado di fornire un'elevata qualità video all'utente finale. L'informazione sulle condizioni dei peer e dello stato della rete permette inoltre migliori scelte di scheduling tutto a favore della qualità finale che si è in grado di fornire.

Capitolo 5

La sicurezza nei sistemi P2P live video streaming

Introduzione alla problematica della sicurezza

Nella distribuzione dei contenuti video live attraverso l'utilizzo delle infrastrutture P2P la presenza di molti attori convolti nello scambio di informazioni e dati porta ad analizzare il problema della sicurezza. Tale tematica riguarda variegati aspetti: autenticazione dei peers coinvolti, integrità dell'informazione ricevuta, rispetto della normative in campo di *copyright* e tutela del diritto d'autore. Molto spesso le reti P2P sono state associate alla pirateria e alla diffusione di materiale illegale. L'impiego delle reti P2P nella distribuzione video deve essere in grado di fronteggiare anche queste problematiche per far sì che questa tecnologia possa esser utilizzata da più attori nel pieno rispetto delle normative vigenti e possa rappresentare al contempo un modello di business. Gli aspetti di fondamentale importanza nella trattazione della sicurezza nelle reti P2P riguardano:

- Autenticità ed integrità: deve esser garantita la provenienza dei dati e

la loro integrità

- Confidenzialità: il contenuto trasmesso deve esser ricevuto ed utilizzato soltanto da entità autorizzate

5.1 Autenticazione

Nello streaming live enormi quantità di dati sono create e distribuite ad un ampio numero di partecipanti. L'utilizzo di una architettura P2P fa si che i blocchi di informazione viaggino attraverso percorsi multipli, vengano inoltrati da vari peers e per questo una delle caratteristiche basilari per la garanzia della sicurezza è rappresentata dalla capacità di autenticare l'origine di ogni blocco informativo a garanzia di una corretta trasmissione di contenuti. Nell'ambiente live la sfida è rappresentata dall'autenticazione di materiale appena creato che viene distribuito attraverso peers intermedi i quali potrebbero diffondere informazioni fraudolente, occupare inutilmente risorse di rete e provocare il collasso del sistema. I sistemi di cifratura e firma comunemente adottati nelle applicazioni tradizionali risultano essere computazionalmente troppo onerosi poiché basati su schemi matematici che prevedono operazioni quali la fattorizzazione di numeri primi o il computo di logaritmi discreti solo per citare degli esempi. Tali schemi non possono essere utilizzati per autenticare i piccoli blocchi di dati formanti uno stream video. La creazione di un sistema di autenticazione pensato per l'utilizzo in ambito broadcast deve tenere in considerazione aspetti fondamentali quali:

- Efficiente generazione e verifica: l'overhead generato dal processo di autenticazione deve necessariamente essere ridotto in quanto un largo numero di entità devono effettuare la verifica dell'informazione di au-

tenticazione ed il tempo a disposizione per eseguire tale operazione è molto limitato trattandosi di un contesto *live*.

- Autenticazione istantanea in real-time
- Robustezza ad eventi di perdita di pacchetti: poiché nelle applicazioni *live* la ritrasmissione di pacchetti andati persi non è applicata il protocollo di autenticazione deve essere in grado di lavorare anche in situazioni nelle quali un alto tasso di perdita di pacchetti si presenta
- Scalabilità: il protocollo di autenticazione deve esser indipendente dal numero di entità presenti nel sistema

Molti studi hanno affrontato il problema dell'autenticazione attraverso l'utilizzo di schemi di firma digitale. L'utilizzo di sistemi basati sulla condivisione di una chiave segreta fra il trasmettitore ed il ricevitore è poco impiegato nell'ambito del P2P poiché tale scelta permette al ricevitore, di sua natura inaffidabile, la firma di pacchetti dati che non sono verificati. L'utilizzo di tecnologie basate sulla crittografia a chiave pubblica rappresentano una valida alternativa ma presentano problematiche nel loro utilizzo derivanti dalla complessità computazionale richiesta da algoritmi quali RSA, DSA e ECDSA inadatti nell'impiego dell'autenticazione di molti piccoli blocchi di dati usualmente della dimensione massima di 1500 byte oltre al fatto che una considerevole mole di risorse è richiesta per la verifica dell'autenticità della firma stessa da parte del ricevitore. In studi come McEliece [49] e Quartz [50] vengono proposte soluzioni di firma alternative facilmente verificabili da lato ricevitore con la forte limitazione rappresentata dalla quantità di tempo necessaria per applicare la firma. Un paio di secondi necessari a compiere tale operazione rappresentano un lasso di tempo troppo prolungato, non accettabile in applicazioni P2P *live* dove il ritardo end-to-end è influenzato da

tanti altri fattori in aggiunta a quelli introdotti dalla crittografia. La riduzione dei costi richiesta per la firma di blocchi di dati ha interessato studi come [51], [52], [53] nei quali però la necessità di lavorare in presenza di blocchi di dati di ridotte dimensioni e la distribuzione di liste hash firmate (un valore hash per ciascun nuovo blocco di dati) ha portato all'abbandono di tali soluzioni poichè troppo onerose. Nello studio [54] viene proposto uno schema che utilizza codici MAC multipli (*Message Autentication Code*) che evita l'impiego di complessi sistemi di firma. L'idea base sta nell'utilizzare tutti i MACs per firmare il blocco di dati mentre viene utilizzato soltanto un sottoinsieme dei corrispettivi segreti lato ricevitore. Tale soluzione si rivela però inefficiente all'aumentare dei peers. BiBa [78] è un sistema che impiega un sistema di firma che adopera una funzione *one-way* senza *trapdoor*. Le sue caratteristiche peculiari sono rappresentate dal fatto che la dimensione della firma è nettamente inferiore rispetto agli approcci tradizionali e la fase della verifica della firma stessa risulta essere molto veloce. Basato sul paradosso del compleanno BiBa rappresenta un sistema di autenticazione in grado di rispettare i vincoli della trasmissione real-time e dell'efficace autenticazione di tutti i peers coinvolti. In ALPS[55] viene analizzato uno schema di firme che estende le potenzialità del sistema di firma one-way per poter esser utilizzato in applicazioni peer-to-peer live. I punti di forza di tale sistema sono rappresentati dalla ridotta dimensione della firma (20 – 40 bytes), dalla chiave pubblica di dimensioni non superiori a qualche kilobytes e dalla velocità nel processo di firma e verifica (da 10 a 300 μs). Questi accorgimenti permettono di avere un sistema all'incirca due volte più performante grazie al quale i stringenti vincoli posti dal sistema *live* possono essere rispettati e la sicurezza di una corretta autenticazione garantita. Entrando in maggiori dettagli lo schema di firma proposto da ALPS è composto da tre parti: un

algoritmo responsabile per la generazione della chiave, uno per la firma ed un ultimo per la verifica.

5.2 Confidenzialità

La confidenzialità del contenuto distribuito attraverso la rete può essere ottenuta attraverso l'utilizzo dei potenti mezzi messi a disposizione della crittografia. Un semplice modo consiste nel cifrare il contenuto da trasmettere con una chiave segreta (*SK Session Key*) posseduta soltanto dagli utenti autorizzati che, grazie ad essa, riusciranno a decifrare lo stream video. La gestione e la distribuzione delle chiavi diventa una questione di fondamentale importanza per la garanzia della sicurezza in un sistema nel quale gli utenti effettuano svariati join e leave durante la sessione. Per questo è necessario rinnovare la chiave SK e ridistribuirla tra i peer, tra coloro i quali sono autorizzati a riceverla ed usufruire del contenuto. Approcci tradizionali per la distribuzione di chiavi in ambienti IP multicast sono esaminati in [56], [57] ma essi risultano essere inadeguati in situazioni con numerosi partecipanti a causa di una elevata richiesta sia elaborativa sia di overhead richiesto. In [58] le DHT (*Distributed Hash Table*) sono state adoperate per la distribuzione delle chiavi ma tale soluzione si rivela inappropriata all'aumentare del numero di peers nel sistema e porta alla necessità di una rete di distribuzione overlay riservata esclusivamente allo scambio delle chiavi andando ad aggiungere ulteriore traffico in rete. Un'interessante alternativa a tale problema viene fornita da [59] EKMD, sistema che fornisce una gestione delle chiavi per applicazioni P2P live streaming garantendo scalabilità ed affidabilità. Nelle applicazioni P2P gli utenti svolgono la doppia funzione di ricevitori-trasmettitori di informazioni e per questo quando un nodo della rete riceve la chiave segreta da KDC

(*Key Distribution Centre*) dovrà provvedere a trasmetterla in maniera sicura a propri vicini. Gli aspetti esaminati da EKDM riguardano: user join, user leave e il processo di distribuzione della chiave SK. Il generico peer i intenzionato a partecipare alla sezione video invia una richiesta di login all'entità KDC nella quale comunica la propria chiave pubblica precedentemente creata. Dopo esser stato accettato ed autenticato il peer i riceve il certificato da KDC che viene scambiato con i vicini per verificarne l'identità e, a processo terminato, inizia lo scambio di informazioni che porta il nuovo utente i a ricevere il materiale necessario per la conoscenza della chiave SK. La chiave di sessione viene generata e periodicamente rinnovata dal KDC che provvede a distribuirla ai vari peers attraverso l'inserimento della stessa all'interno del blocco di dati. A blocco informativo ricevuto l'utente i è in grado di calcolare la nuova chiave di sessione ed inoltrarla ai suoi figli avendola prima cifrata utilizzando la propria chiave segreta. Tale soluzione permette la garanzia dello scambio di contenuti solamente fra utenti autorizzati in possesso di un corretto certificato digitale. La distribuzione delle chiavi aggiornate e distribuite periodicamente permette al sistema l'indipendenza dai comportamenti di join e leave attuati dai peers e l'impiego dei blocchi di dati come contenitori sicuri delle chiavi di sessioni permette il mantenimento di un overhead limitato ed un costo di distribuzione delle chiavi esiguo.

5.3 Comuni attacchi

La maggior parte degli attacchi rivolta verso i sistemi P2P streaming proviene dall'interno del sistema stesso, dai nodi che ne fanno parte. Una volta entrato all'interno del sistema, se non controllato e monitorato, un nodo può trasformarsi in un *malicious node* e contaminare le altre entità che contribui-

ranno a loro volta alla diffusione del problema introdotto. Di seguito vengono elencati i principali tipi di attacchi:

- Forgery attack : intacca il principio di confidenzialità ed integrità. Viene definito *forgery* (falso) ogni blocco di dato immesso fraudolentemente nel sistema attraverso la diffusione di materiale diverso dall'originale. Tali attacchi possono essere risolti attraverso l'utilizzo di sistemi crittografici quali cifrature e firme digitali.

- Eclipse attack: questo tipo di attacco colpisce il funzionamento del protocollo di *membership* e le regole di istradamento dei pacchetti informativi. Un *malicious node* è in grado di prendere il controllo della rete overlay attraverso la diffusione di false informazioni sul posizionamento dei vicini: maggiore è il numero di peers che ricevono queste informazioni maggiore è la forza con con cui l'attaccante riesce ad imporre la propria struttura overlay. L'attacco *Eclpise* può essere considerato una generalizzazione di *Sybil Attac* [81]: il controllo da parte del nodo malitenzionato anche di una sola piccola parte di nodi della rete può portare in breve tempo all'oscuramento della rete overlay originale e alla formazione di quella voluta dal *malicios node*. In [77] viene proposto un efficiente algoritmo di *auditing* che permette il monitoraggio dell'efettivo grado di *outdeegre* e *indegree* a disposizione dei nodi: il mentire riguardo alla risorse a disposizione dei peer influisce sulla scelta dei vicini in quanto la ricerca predilige i nodi che mettono a disposizione più risorse. Fornire false caratteristiche permette al nodo malintenzionato di creare relazioni con un numero elevato di entità e poter quindi distribuire informazioni false che verrano poi diramate ad altri peers contaminando la struttura overlay. Le tecniche di *auditing* e l'imposizione di limiti sul numero massimo di connessioni accettabili

attraverso l'imposizione di una soglia, sono una buona difesa contro l'*eclipse attack*

- Sybil attack: si è in presenza di questo tipo di attacco quando un peer è in grado di creare più identità, più pseudonimi allo scopo di incrementare la propria reputazione e la propria influenza all'interno della rete. Strumenti per la prevenzione di questo attacco consistono nell'impiego di un'entità terza che sia in grado di certificare l'identità dell'entità conivolta nello scambio.

- Pollution attack: si verifica in presenza di un peer malintenzionato che dissemina all'interno della rete blocchi di dati non appartenenti allo stream originale al solo scopo di degradare la qualità del sistema attraverso azioni di spam.

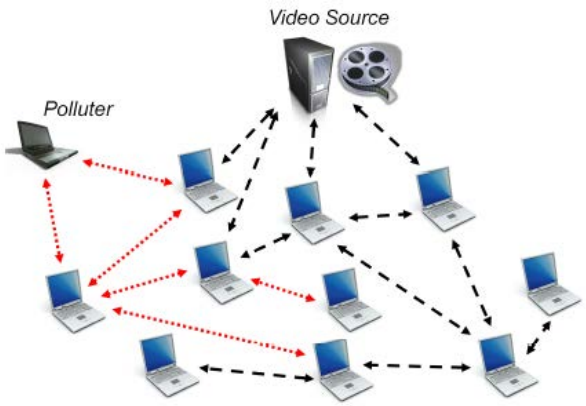

Figura 5.1: Esempio di pollution attack: il malicious node immette in rete blocchi spam allo scopo di degradare la qualità video

- Neighbor selection attack: questo attacco consiste nel controllare i meccanismi di selezione dei vicini per ciascun nodo. Questo permette di

influenzare la creazione della rete overlay andando a sovvertirne l'intero funzionamento.

- DoS attack: tale attacco può esser perpetrato attraverso varie forme quali l'invio di pacchetti duplicati, invio di ripetuti messaggi di richieste allo scopo di disseminare informazione superflua ed inutile che porta alla congestione del sistema.

- Ommission attack: contrariamente all'attacco DoS questa tecnica consiste nel non inviare il contenuto di cui si dispone, nel non trasmetterlo volutamente agli altri peer intaccando quindi l'intero sistema di distribuzione.

I devastanti effetti provocati dal *pollution attack* sono riportati all'interno dello studio [60] nel quale viene dimostrato come un singolo peer malintenzionato sia in grado di degradare notevolmente la qualità del video ricevuta dagli altri utenti attraverso l'immissione in rete di segmenti corrotti. L'esperimento è stato condotto monitorando l'andamento di due peers normali in ascolto di un popolare canale live a 342 kbsp trasmesso da PPLive prima e dopo l'immissione in rete di pacchetti corrotti da parte del malicious peer. Il grafico 5.2 riporta l'andamento del numero dei peers presenti nel sistema: al minuto 34 il malicious peer comincia a diffondere il materiale corrotto inviando messaggi di disponibilità e di elevata capacità sia in termini di segmenti video sia in termini di banda disponibile. Questo comportamento fa si che lui diventi una delle maggiori fonti da cui reperire il video favorendo un rapido divulgarsi tra i peers di materiale corrotto che porta alla degradazione della qualità ricevuta ed al conseguente abbandono da parte dei peers come evidenziato nel grafico. Per impedire l'insorgere di simili situazione sono stati analizzati vari meccanismi di difesa che comprendono: creazione di *blacklist*

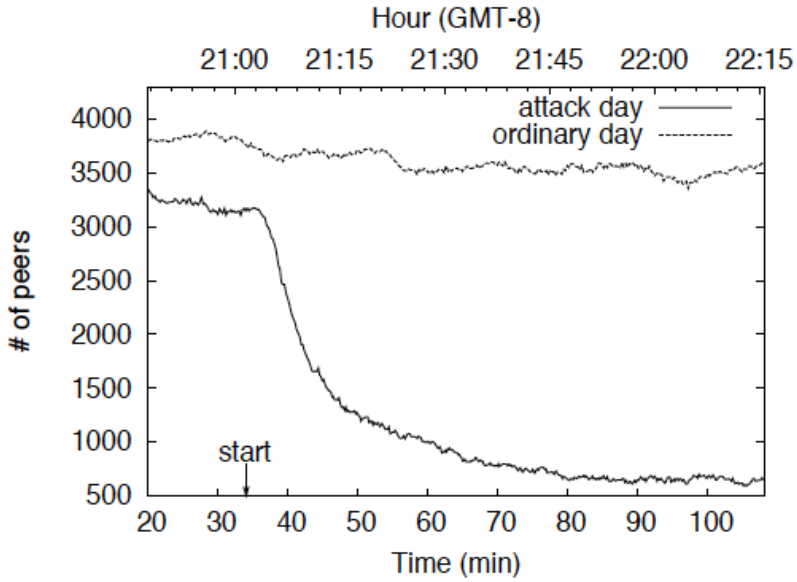

Figura 5.2: Pollution Attack

per la segnalazione di peer sospetti, utilizzo di tecniche crittografiche per la protezione del traffico, utilizzo di sistemi a firma digitale e sistemi di hash per la verifica dei blocchi di dati. Particolare attenzione viene riposta nei vari sistemi di firma digitale dei blocchi informativi:

- *Sign-All:* ciascun blocco informativo è individualmente firmato dalla sorgente e la firma (rappresentante dell'autenticità dell'informazione) è trasportata assieme ai dati verso i ricevitore il quale dovrà verificare ciascun blocco e, in caso di veridicità degli stessi, procedere alla riproduzione. Tale soluzione non può essere impiegata in presenza di trasmissioni ad alto bit-rate dove le operazioni di firma e verifa da effettuare su ciascun segmento generato dal flusso video risultano essere troppo onerose.

- *Start chaining:* per alleggerire il carico di overhead a livello computazionale i singoli blocchi di dati sono raggruppati tra di loro in super-

blocchi. La sorgente procede al calcolo delle funzione di hash di ciascun blocco di dati, li raggruppa tra d loro e firma il super-blocco. L'informazione dell'autenticazione consiste nella firma del super-blocco, nella posizione del blocco dati all'interno di esso e dalle funzioni di hash degli altri blocchi presenti.

- *Merkle-Tree Chaining:* questo approccio richiede la costruzione di un albero di autenticazione per ciascun super-blocco. I nodi figli degli alberi corrispondono con i valori hash dei singoli blocchi contenuti nel super-blocco. La firma del nodo sorgente diventa la firma del super-blocco. L'informazione di autenticazione per ciascun blocco è contenuta nella firma, nella posizione del blocco stesso e nei nodi presenti nel percorso che unisce le foglie alla sorgente dell'albero.

- *Sign-and-correct:* la sorgente calcola la funzione di hash di ciascun blocco separatamente e procede poi alla firma della concatenazione dei blocchi hash. Alla firma ed i blocchi di hash viene poi applicato il codice di correzione RS (*Reed-Solomon*).

5.4 SecureStream: sistema per protezione da attacchi

SecureStream [61] rappresenta un sistema sviluppato per ridurre al minimo la possibilità di attacchi: l'utilizzo di tecniche basate sul calcolo di digest, l'utilizzo di protocolli Fireflies per la prevenzione di attacchi alla struttura overlay, l'utilizzo di tecniche di auditing per il monitoraggio del comportamento dei peers rivolte alla ricerca ed isolamento di peers potenzialmente dannosi sono i fattori che fanno di SecureStream una soluzione adatta nel-

l'utilizzo dei sistemi di distribuzione peer-to-peer. L'impiego della tecnica *pull-based* ed una visione completa sugli scambi che avvengono tra i peer evita potenziali attacchi di tipo DoS ad alto livello. La conoscenza del comportamento dei peers che prendono parte alla sessione avviene attraverso l'utilizzo del protocollo *Fireflies* composto da tre nuclei principali: un protocollo di *ping* per individuare la presenza attiva del peer, un protocollo di *gossip* per diffondere le informazioni tra i membri autorizzati ed un protocollo di *membership* per la gestione della messaggistica del protocollo stesso. La realizzazione delle caratteristiche offerte dal protocollo avviene attraverso l'organizzazione dei membri all'interno di anelli e la posizione che ciascuno di esso assume nello stesso determina la gerarchia all'interno dei nodi, stabilisce quali nodi debbano essere controllati e le azioni che possono essere intraprese. In ciascun anello la generica entità i monitora un sottoinsieme di altre entità j che sono ritenute essere attive. Se i individua un comportamento anomalo del membro j invia un messaggio accusatorio al gruppo di nodi da lui controllati. Questi, una volta ricevuto il messaggio circa il possibile comportamento insolito del nodo j, attendono Δt prima di rimuovere il nodo j dalla propria lista di vicini. Durante il lasso di tempo Δt il peer accusato può rispondere a tale incriminazione inviando un nuovo messaggio di discolpa per riaffermare la propria presenza ed integrità. Per evitare il proliferare di messaggi d'accusa inviati di proposito da un malicious peer i nodi possono invalidare tali messaggi e limitarne il raggio d'azione ad n anelli, circoscriverne quindi la diffusione. L'integrità dei dati trasmessi viene attuata attraverso l'impiego di firme digitali: per evitare la firma e la conseguente verifica di ciascun pacchetto SecureSteam adotta la tecnica di raggruppamento degli hash di n pacchetti all'interno di un speciale messaggio che viene poi firmato dalla sorgente (tecnica del *linear digest*). La distribuzione dei pacchetti

avviene utilizzando un approccio di tipo pull simile a quello offerto dai protocolli usati da Coolstreming e Chainsaw. La possibilità di poter scegliere la sorgente informativa tra più candidati permette di svincolare il sistema dalla dipendenza provocata dalla presenza di un'unica fonte. Tale indipendenza dei peer porta ad avere una capacità immediata nel rispondere a situazioni di attacchi verso il sistema rendendo l'attacco stesso meno dannoso e circoscritto. Un ulteriore livello di protezione contro gli attacchi DoS viene effettuato selezionando dagli stessi anelli utilizzati da Fireflies un insieme prestabilito di peers vicini con i quali effettuare lo scambio di informazioni attraverso canali autenticati. L'inoltro di pacchetti da parte della sorgente avviene attraverso il consueto invio di messaggi di disponibilità verso i propri vicini a cui corrisponde un messaggio di richiesta. Quello che SecureStream aggiunge a tale procedura è l'utilizzo di due finestre temporali: ciascun membro mantiene e distribuisce i pacchetti in suo possesso all'interno del lasso di tempo chiamato "*finestra di disponibilità*". Gestisce allo stesso tempo un ulteriore *timer*, minore rispetto alla finestra di disponibilità, denominato "*finestra di interesse*" che indica i pacchetti per i quali il peer è interessato alla ricezione. Esiste un limite di richieste che possono essere rivolte verso un vicino impedendo ad un malintenzionato la diffusione di un quantitativo abnorme di richieste. Il peer, all'interno del proprio buffer di richieste, provvederà a mantenere solamente i messaggi più recenti eliminando le restanti evitando l'immissione in rete di inutili pacchetti duplicati. Per rispettare il principio della non ripudiabilità un pacchetto viene inoltrato solo dopo esser stato verificato: per far questo l'entità verificatrice deve necessariamente possedere il pacchetto di *digest*. Per questo, quando un peer richiede un pacchetto alla sorgente, comunica di essere o meno in possesso del pacchetto di diget; la sorgente provvederà all'invio dello stesso incorporandolo nel messaggio di risposta al

peer richiedente. L'impiego di tecniche di *auditing* permette l'implementazione di sistemi di punizione verso quei peers che assumono comportamenti più o meno virtuosi. Il processo di *auditing* avviene in maniera distribuita: periodicamente vengono eletti dei nodi ascoltatori che valutano il contributo fornito da ciascun suo vicino e permettendo l'individuazione di attacchi quali *ommission attack*. Lo studio [61] riporta una raccolta di dati ottenuta attraverso una ricca campagna sia di simulazioni sia di esperimenti condotti attraverso la piattaforma Emulab.

5.5 D.R.M. - *Digital Right Managment*

Le reti P2P sono un efficacie strumento per la distribuzione di contenuti ma, allo stesso tempo, possono essere un potente strumento a disposizione della pirateria per la diffusione di materiale protetto da *copyright* . E' di fondamentale importanza riuscire a costruire una piattaforma di distribuzione che integri i vantaggi offerti dal P2P e che garantisca una distribuzione legale del materiale, nel rispetto delle normative vigenti in campo di copyright. La tecnologia DRM (*Digital Right Managment*) [71] rappresenta uno strumento in grado di proteggere, identificare e rendere tracciabili i contenuti digitali che si vogliono distribuire e diffondere. Molti studi in questo ambito hanno portato alla creazione di soluzioni commerciali come ad esempio Windows Media DRM di Microsoft [72], Helix DRM di RealNetwork [73] che si basano però sul tradizionale paradigma client-server. In letteratura vengono identificati tre modelli di archiettura D.R.M.:

- Architettura D.R.M. Client-Server
- Architettura D.R.M. P2P Distribuita

- Architettura D.R.M. P2P Semi-distribuita

In figura 5.3 è illustrata la tipica architettura modello client-server per la gestione D.R.M.: tutte le funzioni per garantire la protezione del contenuto, l'accesso alle sole entità autorizzate, la distribuzione delle chiavi di decifratura e dei certificati sono svolte dal server centrale D.R.M. La rete P2P sottostante viene utilizzata solamente per la distribuzione del contenuto video ma non partecipa in alcuna maniera alla gestione dei diritti digitali. In figu-

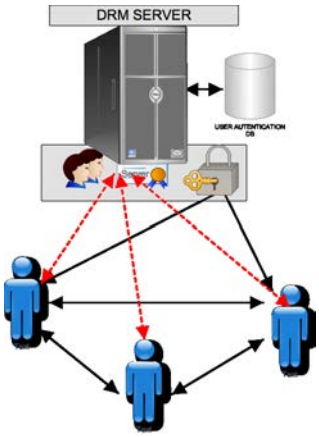

Figura 5.3: Classica architettura Client-Server per la gestione D.R.M.

ra 5.4 viene invece rappresentata l'architettura opposta rispetto al modello Client-Server: nella soluzione P2P distribuita i moduli per l'autenticazione dei peers, per la cifratura e protezione del contenuto e la gestione dei certificati vengono svolti dai nodi stessi che collaborano in maniera attiva non solo alla distribuzione del materiale ma anche alla gestione dei diritti e sicurezza ad essa connessi. Per la gestione di alcune situzioni legate, ad esempio, all'uscita dalla rete di nodi adibiti alla distribuzione dei certificati, un server D.R.M. centrale è presente e supervisiona l'andamento del sistema pronto ad intervenre in caso di necessità. Una soluzione intermedia alle due proposte descritte in precedenza è raffigurata in 5.5: si tratta di un'architettura P2P

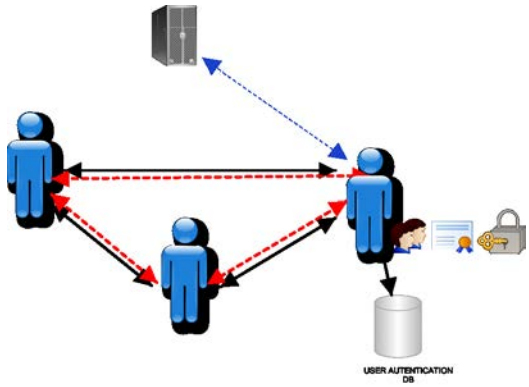

Figura 5.4: Architettura P2P distribuita per la gestione D.R.M.

semi-distribuita dove parte delle funzioni sono ancora svolte dai peers mentre al server centrale viene affidata la gestione dell'autenticazione dei peers. Questo tipo di architettura è in grado di apportare miglioramenti nell'ambito della sicurezza grazie all'impiego di un server dedicato che però utilizza la rete P2P e le caratteristiche dei peer per alleviare il carico di operazioni a lui richieste e permettere una miglior scalabilità e crescita del sistema rispetto al rigido modello client-server.

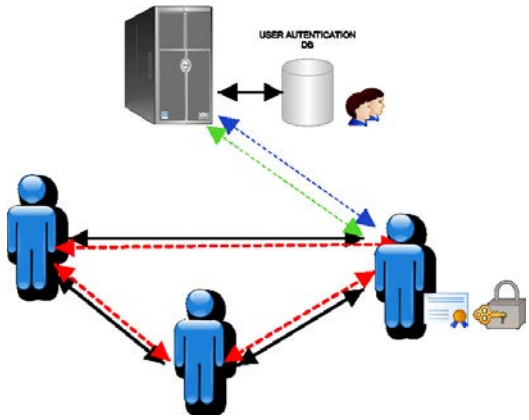

Figura 5.5: Architettura P2P semi-distribuita per la gestione D.R.M.

La presenza di un'entità centrale adetta alla sicurezza potrebbe rappresentare una forta limitazione alla natura cooperativa e distribuita tipica dei

sistemi P2P: in [74] viene proposto un sistema nel quale i peers appartenenti ad uno stesso gruppo sono in grado di distribuire in maniera cooperativa sia il flusso informativo sia il materiale digitale contenente le informazioni per l'accesso, il controllo e la distribuzione delle chiavi di sessione tra gli utenti autorizzati. Lo schema proposto consiste nell'affidare i controlli di sicurezza ad entità amministratrici terze rispetto ai nodi P2P e contemporaneamente affidare parte delle operazioni di sicurezza ad i singli nodi in modo tale da alleviare il carico verso i servers centrali, rendere il sistema scalabile e velocizzare i processi di autorizzazione. La costruzione di un modello di sicurezza che utilizza i paradigmi della distribuzione peer-to-peer affiancati dalla presenza di un'autorità centrale rende possibile l'implementazione di un sistema DRM in ambito P2P. Il modello matematico analizzato dimostra come il sistema DRM-P2P sia superiore in termini di minor overhead generato, scalabilità del sistema, minor carico sui server centrali rispetto alla tradizionale struttura centralizzata. Inoltre la soluzione proposta, applicabile nel campo del *file sharing, e-business, VoD e livestreaming*, è facilmente integrabile in una struttura P2P già esistente senza la necessità grandi cambiamenti infrastrutturali.

Nello studio [82] viene presentata un'architettura che invece di impiegare l'utilizzo di un server centrale adopera dei super-nodi eletti tra i peers partecipanti alla sezione, creando una rete overlay a due livelli. In figura 5.6 è rappresentato il sistema in esame. I Super-Nodi sono i responsabili della gestione e distribuzione dei certificati fra gli utenti: il peer A ha a disposizione il contenuto video che protegge attraverso cifratura e l'impiego di una chiave CEK (*Contents Encryption Key*). Il peer A contatta il suo Supep-Nodo e comunica la lista dei segmenti che sono in suo possesso e dichiara di averli cifrati e di possedere il certificato che ne permette la decifratura.

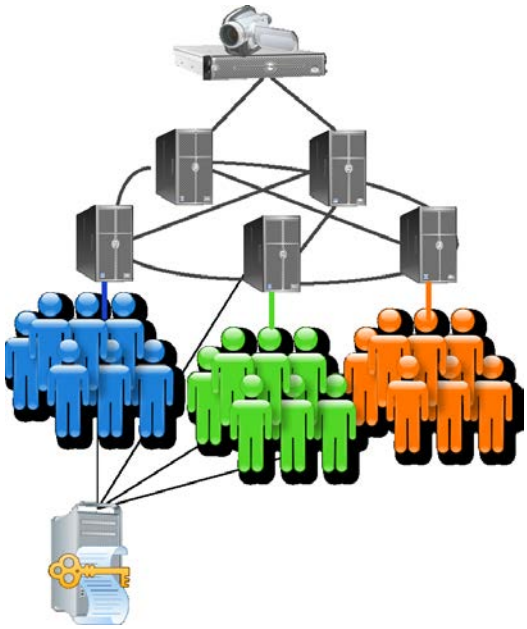

Figura 5.6: Architettura D.R.M. con impiego di Super-Nodi

Un nuovo peer B che vuole entrare a far parte del sistema contatta il livello di Super-Nodi che, dopo aver verificato le credenziali del peer B e verificato l'autorizzazione per l'accesso alla sezione, fornisce la lista dei peers che dispongono del contenuto da lui ricercato. A questo punto il peer B è in grado di contattare il peer A e, attraverso l'impiego di PKI e certificati per l'autenticazione delle entità in gioco, instaurare la connessione che permetta lo scambio informativo. L'impiego di Super-Nodi eletti tra i peers permette al sistema di raggiungere una scalabilità non concessa in presenza di un'unica struttura centralizzata andando a sfruttare le risorse offerte dalla rete P2P.

Nello studio [75] viene esaminata la tematica della protezione dei contenuti digitali nell'ambito della distrubuzione P2P video live attraverso l'impiego di un'architettura composta da quattro entità (figura 5.7)

- Register server: autorizza la partecipazione dei peer alla sezione video fornendo la chiave di decodifica e la lista contenente gli indirizzi dei

server list.

- Index servers: entità preposta al mantenimento delle informazioni riguardanti lo stato dei peer e dei propri buffers.
- Peers: mantengono la connessione con un solo index server durante tutta la loro attività
- Supernodi: sono dei peer speciali preposti a funzioni particolari.

Il processo di join di un nuovo peer avviene contattando il *register server* che, dopo aver accettato il nuovo ingresso, assegna un id univoco *PeerId* e trasmette gli indirizzi dei Index servers e la chiave per poter decifrare il flusso. Il peer a questo punto contatta uno fra gli n servers elencati, comunica il proprio ID e riceve una lista di peer vicini con i quali scambiare il flusso video cifrato. La cifratura del flusso video viene applicata in tempo reale e quindi i tradizionali metodi impiegati nel mondo off-line non sono applicabili in questo contesto. Viene quindi impiegato un algoritmo di selezione caotica per la cifratura del flusso video: la seguente mappa caotica viene impiegataa

$$x_{n+1} = \mu \cdot x_n(1 - x_n) \tag{5.1}$$

dove $x_n \in (0, 1)$ e $\mu \in (0, 4)$. I valori x_0 e μ rappresentano le chiavi segrete [76]. L'espressione 5.1 in XOR con il flusso video crea il flusso cifrato: la ragione fondamentale dell'impiego di questa tipologia di cifratura risiede nel fatto che è un procedimento veloce da implementare per ambienti real-time. Le chiavi segrete sono gestite dal server centrale e distribuite solamente agli utenti autorizzati.

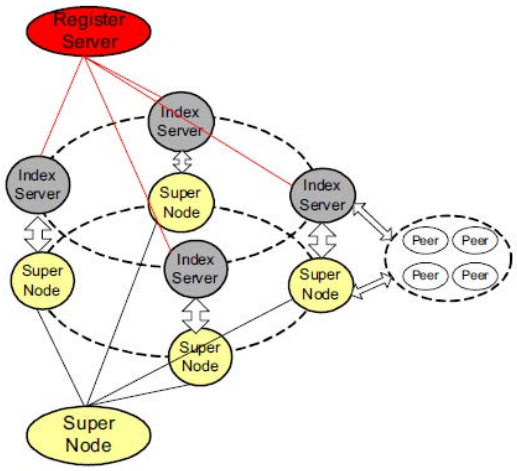

Figura 5.7: Elementi componenti l'architettura del sistema

5.6 Considerazioni sulla sicurezza

Nel seguente capitolo sono state analizzate le problematiche riguardanti la sicurezza nell'ambito di un sistema P2P streaming. Esaminati i vari tipi di attacchi e le maggiori vulnerabilità presentate da tali sistemi. Soluzioni proposte da recenti studi sono stati esposti ed è stata evidenziata la necessità di affrontare la questione della sicurezza sotto ogni punto di vista per far si che lo sviluppo della tecnologia P2P non sia frenato dall'inaffidabilità derivante dalle lacune in ambito di sicurezza.

Capitolo 6

Sviluppo futuro e progetti in corso

6.1 Proposta di sistema ottimale

L'analisi e lo studio delle varie soluzioni proposte in letteratura nell'ambito della ricerca delle soluzioni per il miglioramento dei sistemi di distribuzione video tramite l'utilizzo delle reti peer-to-peer ha portato alla luce la presenza di svariati sistemi sia nel campo accademico sia in quello commerciale. L'analisi di tali studi ha evidenziato come ciascun sistema focalizzi l'attenzione sull'implementazione di un determinato ambito della distribuzione, su l'ottimizzazione di una specifica procedura talvolta dimenticando la globalità del sistema, le problematiche connesse che incorrono nel passaggio da una soluzione teorico-simulata all'utilizzo della stessa nell'ambiente reale, nel mondo di Internet. A mio avviso l'ostacolo che frena l'espansione dell'utilizzo del sistema P2P da parte del mondo della distribuzione video commerciale riguarda fondamentalmente la gestione della sicurezza e dei diritti che garantiscano il rispetto delle normative in termini di copyright e proprietà intellettuale.

Lo sviluppo commerciale ed i molteplici modelli di business che possono derivare dall'impiego di questa tecnologia e dai vantaggi da essa offerta sono fortemente legati alla capacità del sistema di offrire una trasmissione affidabile, sicura, scalabile nella quale gli attori coinvolti possano fruire di servizi, qualità e sicurezza pari o superiore a quella offerta dai sistemi tradizionali.Nel paragrafo successivo verrà presentato un modello di distribuzione che, tenendo in considerazione gli aspetti più innovativi ed interessanti proposti nella recente letteratura, unisce la conoscenza in un sistema omogeneo il più possibile adattabile all'eterogeneità dell'ambiente nel quale potrà essere impiegato. L'analisi del sistema comprende gli ambiti della scelta della topologica da impiegare, i sistemi di rewarding da adottare, le soluzioni in materia di coding da utilizzare e la gestione della sicurezza.

6.1.1 Architettura generale

Il sistema in analisi ha lo scopo di distribuire in modalità live il contenuto video generato dalla sorgente che alimenta il server video responsabile del trattamento del flusso e dell'immissione dello stesso nella rete p2p. Come illustrato in figura 6.1 il sistema è composto inoltre dal modulo Sicurezza addetto alla cifratura del flusso, dal modulo Autenticazione e distribuzione delle chiavi. La gestione degli utenti, i processi di join e leave vengono gestiti dal tracker mentre il server di auditing monitora e registra nel proprio database i punteggi attribuiti ai peers relativamente al proprio comportamento. Il flusso video viene codificato adoperando il codec H.264/SVC con *base layer* cifrato attraverso algoritmo di cifratura [98] mentre gli *enhancement layers* sono trasmessi senza essere sottoposti al processo di protezione. I peer facenti parte del sistema sono raggruppati in *cluster* secondo metriche che tengono in considerazione il posizionamento degli stessi all'interno

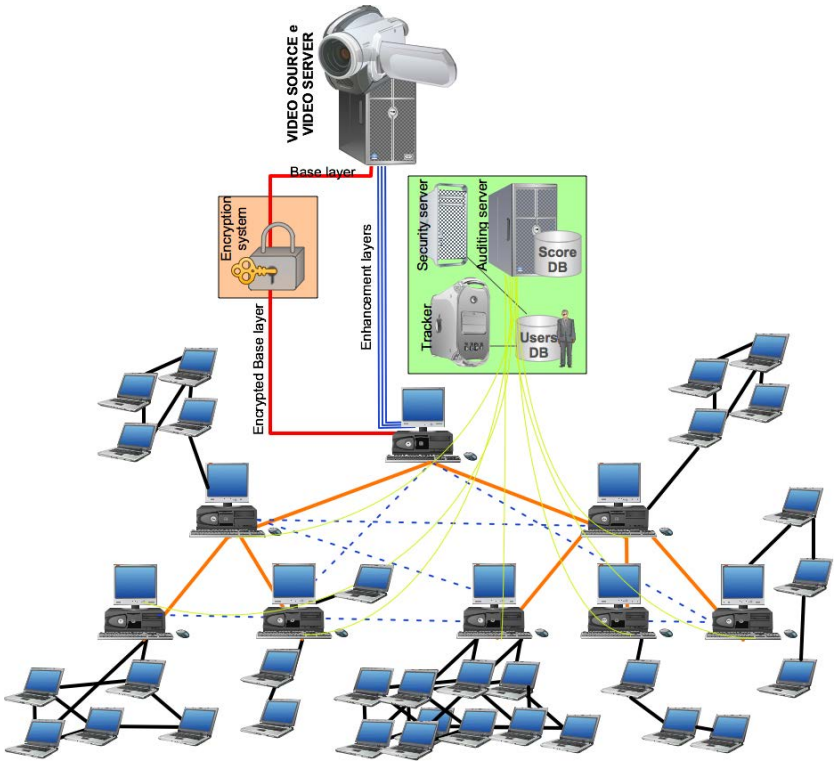

Figura 6.1: Architettura proposta per lo sviluppo di un sistema ideale

della rete underlay favorendo l'aggregazione e lo scambio di informazione tra quelli più vicini. Come riportato in figura nel sistema sono presenti dei peer speciali che svolgono la funzione di super-nodi. Questi svolgono il compito di gestione degli utenti collocati all'interno del cluster sia attraverso la distribuzione delle chiavi di cifratura sia attraverso un controllo di monitoraggio attuato attraverso tecniche di auditing allo scopo di individuare il comportamento anomalo di possibili *malicious peers* che potrebbero compromettere il sistema. Ogni super nodo comunica, dopo aver verificato attraverso l'invio di messaggi di test, il comportamento anomalo o virtuoso dei peers sotto esame al modulo centrale Auditing Monitor che archivia in un database condiviso tra i supernodi le valutazioni degli utenti. Se la valutazione di un peer è inferiore ad una determinata soglia significa che questo sta avendo un

comportamento scorretto, non cooperativo (*free-riders*) oppure si sta comportando come un *malicious peer*. In quest'ultimo caso il *tracker*, una volta verificata la malignità dell'utente, provvederà ad interrompere la connessione del peer accusato ed isolarlo dal sistema confinando e limitando la diffusione del materiale contaminato. Le tecniche di *auditing* sono impiegate non solo per punire i comportamenti scorretti degli utenti ma anche per premiare i peers virtuosi attraverso l'attribuzione di un punteggio positivo che verrà utilizzato nelle fasi di selezione dei super-nodi e dei nodi stabili che compongono l'albero di distribuzione centrale. Tenere un comportamento altruistico attraverso la condivisione delle proprie risorse porta vantaggio al peer che vede aumentare la propria reputazione e la possibilità di avvicinarsi alla sorgente diminuendo il ritardo *end-to-end*.

6.1.2 Scelta topologica

La scelta topologica adottata nell'impiego del sistema in esame si basa su una struttura ibrida che unisce i vantaggi offerti dalla distribuzione ad albero e quelli della tecnologia mesh. Come evidenziato in letteratura nei sistemi mesh è stato osservato che durante il processo di distribuzione alcuni nodi si distinguono per il comportamento che hanno nel creare una sorta di dorsale distributiva, un *backbone* stabile al quale gli altri utenti sono connessi. Questa osservazione ha portato all'idea dell'elezione di nodi stabili per la creazione dell'albero nel quale il contenuto video viene inoltrato in modalità push da padre verso figlio riducendo notevolmente l'overhead presente negli scambi mesh. L'elezione dei nodi stabili avviene ricercando i peers più virtuosi presenti nella rete mesh: questo avviene attraverso la consultazione dei punteggi contenuti nell'Auditing database e valutando il tempo di presenza dei peers all'interno del sistema. Nelle fase iniziali di distribuzione del

video, quando i punteggi di valutazione non sono ancora presenti, l'albero di distribuzione viene creato scegliendo in maniera casuale fra gli utenti dei vari cluster: con il passare del tempo e con la ricezione dei primi punteggi la topologia subirà delle variazioni che porteranno ad una migliore organizzazione topologica grazie all'impiego degli algoritmi di ottimizzazione impiegati durante la fase di join.

Fase di Join

Un nuovo utente che desidera partecipare alla sezione video contatta il tracker server che a sua volta verifica in associazione con il server Access Control le credenziali dell'utente per la ricezione del materiale. In caso affermativo il peer riceverà una lista di possibili vicini in base alle informazioni di località disponibili e al contempo sarà fornito della chiave necessaria per decifrare il flusso. L'ingresso di un nuovo utente all'interno del cluster mesh può portare al superamento delle dimensioni massime del cluster ed alla elezione di un super-nodo.

Fase di Leave

Un nodo che vuole abbandonare in maniera *gracefully* il sistema contatta il tracker ed annuncia la propria uscita. Il tracker procederà quindi all'elezione di un sostituto come descritto in precedenza se il nodo in questione è un super-nodo. Se l'uscita avviene invece in modo inaspettato saranno le procedure di *monitoring* ad avvisare il sistema.

Modalità distribuzione contenuti

La distribuzione del contenuto avviene in modalità *push* per quanto riguarda l'albero ed in modalità *pull* per la parte di cluster mesh: il comportamento

pull-push viene gestito da due puntatori disposti all'interno del *playback-buffer* di ciascun peer. La presenza del puntotatore mesh disposto tra il puntatore di play-back e quello tree permette di individuare, all'interno di una finestra temporale limitata, la mancanza di blocchi informativi che vengono quindi richiesti ai vicini attraverso le connessioni mesh. Gli utenti disposti nei cluster mesh hanno il puntatore tree disattivato e ricevono quindi il flusso video tramite l'esplicita richiesta di contenuti verso i vicini. In presenza di una leave di un super-nodo anche all'interno della dorsale di distribuzione viene impiegata la modalità di distribuzione pull: i figli del padre scomparso attiveranno le procedure per l'esplicita richiesta di contenuti dai propri vicini in attesa che un nuovo padre ritorni a fornir loro i contenuti in modalità push.

6.1.3 Scelta di coding e cifratura

L'impiego delle tecniche di codifiche scalabili offerte da H.264/SVC all'interno del sistema ricopre un duplice ruolo: la scalabilità offerta dal codec in esame permette al flusso video di esser ricevuto ed utilizzato da apparati eterogenei e, al contempo, la necessità di ricevere il layer base per poter decodificare i livelli migliorativi permette di agire in maniera intelligente sulla cifratura dello stream video: non è necessario codificare tutto il flusso video (operazione molto onerosa sia in termini di tempo che di risorse necessarie) ma è sufficiente codificare il livello base senza il quale tutto il resto del video è inutilizzabile. Il processo di codifica in un'architettura SVC è fondato sulla codifica *inter coding* che opera sui vettori di movimento (MVs) per la ricerca di dipendenze statistiche temporali tra le differenti frames, sulla codifica *intra-coding* (IPMs) nella ricerca di dipendenze spaziali nella singola immagine e nella predizione della frame residua traverso l'impiego di trasformate

DCT. In [99] viene proposto un algoritmo di cifratura che va ad operare sulle tre componenti precedentemente citate e rappresentate nello schema 6.2. La peculiarità di questo tipologia di algoritmo è l'utilizzo di tecniche di XOR del flusso video con parte della Master Key (dalla quale sono ricavate le tre chiavi per i distinti tre domini di cifratura), l'impiego di codici *Exp-Golomb* e la ricerca nella creazione di un algoritmo che non generi ulteriore overhead e che non vada ad intaccare l'efficienza di compressione del codec stesso. Le simulazioni condotte nello studio in esame evidenziano come l'overhead introdotto nelle fasi di cifratura per MVs e IPMs sia nullo mentre quello generato nella fase di predizione della frame residua con DCT porta ad un incremento del 0.05% rispetto allo strem non protetto. La percentuale di tempo richiesta nella fase di cifratura non supera lo 0.82% mentre quella di decifratura è attorno allo 0.20%. Questi risultati dimostrano come tale processo sia estremamente veloce ed adatto nell'impiego real-time. In figura 6.2 è riportato lo

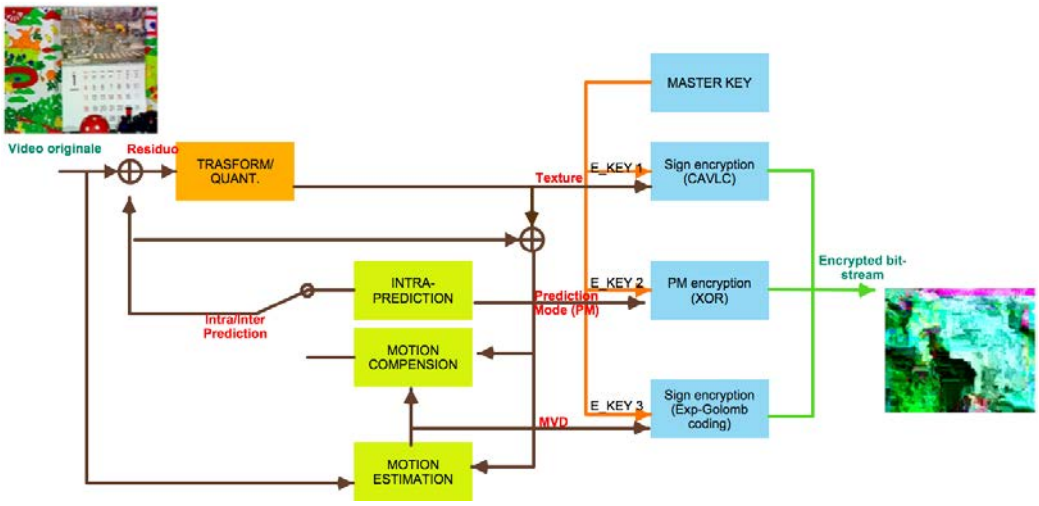

Figura 6.2: Schema a blocchi per il processo di cifratura del base layer con esempio di frame originale e cifrata

schema a blocchi raffigutante il processo di cifratura a cui viene sottoposto il

base layer del flusso video. In assenza della chiave di decifratura il base layer codificato risulta inutilizzabile e quindi anche i livelli superiori che, seppur non protetti, non possono essere impiegati.

6.1.4 Monitoraggio e reward

Nell'ambiente aperto di Internet il comportamento tenuto dai nodi facenti parte del sistema può non esser sempre corretto e cooperativo: alcuni nodi potrebbero ricevere il contenuto video senza reinoltrarlo, ritardare volutamente l'invio dello stesso al solo scopo di danneggiare il sistema di distribuzione e degradare la qualità del video ricevuta dagli utenti. Per supervisionare al corretto comportamento degli utenti nel sistema in esame è stato introdotto un sistema di controllo e monitoraggio basato sulla creazione di una rete overlay dedicata a tale esigenza creata tra gli utenti, i super-nodi ed il server centrale di Monitoring. I super-nodi dell'albero di distribuzione svolgono la funzione di monitorare, ricevere, analizzare ed inoltrare attraverso canali sicuri le misurazioni ottenute verso il Monitor server che provvederà ad archiviare i risultati, convertiti in punteggio, all'interno del Score DB ed avviserà il tracker in caso di azioni da intraprendere contro i malicious node. Il controllo del comportamento di ciascun nodo viene fatto attraverso la scambio tra nodi vicini che si auto-controllano e che inviano i risultati al proprio super-nodo. Ciascun super-nodo sorveglia e gestisce solamente una parte limitata della rete in modo tale da limitare il propagarsi di messaggi, limitare il traffico di overhead dovuto alla segnalazione e facilitare contemporaneamente l'individuazione di *malicious node* poiché operanti in una zona ristretta dove le entità in gioco sono note e conosciute al super-nodo che vigila. L'impiego del protocollo Fireflies [97] permette ai membri del sistema di monitorare il comportamento che i propri vicini tengono attraverso tecniche di *pinging* ed

invio di messaggi accusatori e attraverso tali azioni di controllo permettere sia l'individuazione di comportamenti anomali che l'implementazione di un sistema di gratifica verso gli utenti meritevoli.

6.1.5 Considerazioni finali sul sistema proposto

Il sistema proposto esamina gli aspetti salienti della distribuzione video P2P tenendo in considerazione le più recenti tecnologie e proposte enunciate dalla letteratura scientifica. Tali soluzioni sono state esaminate e combinate tra loro per ottenere un modello che sia in grado di fronteggiare le svariate richieste prestazionali richieste nell'ambito della distribuzione *live*. La scelta è stata quella di implementare un sistema composto da un nucleo di server centrali ai quali sono affidati la gestione degli utenti e la sicurezza. L'impiego di codifiche scalabili SVC porta un duplice vantaggio consistente in primo luogo nell'adattabilità del flusso flusso video all'eterogeneità della rete e dei dispositivi che fluiscono del servizio ed in secondo luogo introduce una notevole semplificazione nel processo di cifratura del flusso grazie al solo impiego del *base layer*. La suddivisione dei peer in cluster basati sul posizionamento geografico degli stessi e la supervisione operata da parte del super-nodo permettono il mantenimento di una struttura più controllata nella quale è più semplice esaminare il comportamento degli utenti e provvedere attraverso azioni o di punizione o di elezione a stadi superiori. Infine la scelta dell'impiego di una topologia ibrida albero-mesh risulta essere una buona soluzione per offrire robustezza e scalabilità sfruttando i vantaggi offerti dalle singole soluzioni; l'impiego di tecniche di monitoring e rewarding incentivano il comportamento corretto ed altruistico dei peers a beneficio dell'intero sistema.

6.2 Sviluppi e ricerche in ambito accademico e commerciale

La distribuzione di materiale audio-video attraverso l'utilizzo di Internet e del paradigma P2P rappresenta un'importante area di ricerca nella quale ingenti investimenti vengono fatti per lo sviluppo di nuove soluzioni e standardizzazioni. L'analisi condotta da Cisco in [62] prevede entro che entro il 2014 il traffico generato per la distribuzione di video attraverso Internet supererà la soglia del 90% su un traffico IP totale previsto di 64 exabytes/mese. La percentuale a cui fa riferimento lo studio include il traffico generato per la distribuzione video in tutte le sue forme: Internet TV, Video On Demand, streaming ed anche P2P. Considerando la crescente diffusione del video live che ha portato, sempre a quanto riportato in [62] , ad un traffico di 280 petabytes/mese nel 2010, la tecnologia P2P rappresenta sicuramente uno strumento in grado di aiutare e sostenere l'enorme proliferazione di contenuti video a cui Internet è sottoposta in questi anni. Il consolidato fallimento dell'approccio client-server spinge allo sviluppo di soluzione decentralizzate ed architetture di rete distribuite: il P2P rappresenta un'opportunità molto attraente come più volte evidenziato in questo lavoro di tesi. Un elevato numero di fornitori di servizio, portali d'informazione ed intrattenimento stanno conducendo intensivi studi e sperimentazioni nell'utilizzo di questa tecnologia: risulta evidente la necessità della creazione di una serie di standard e direttive comuni che permettano uno sviluppo di soluzioni adottabili da un ampio numero di entità e mercati. Gli strumenti messi a disposizione del P2P devono costantemente essere potenziati attraverso processi che li rendano "intelligenti" che permettano un'ottimizzazione delle risorse per giungere ad un miglior QoS e QoE; in questo campo si stanno muovendo progetti fi-

nanziati dalla comunità Europea quali P2P-Next(FP7 IP 14 milioni di euro da CE periodo 2008-2011) e NAPA-WINE (3,73 milioni di euro da CE). In tabella 6.3 vengono elencati i principali sistemi P2P impiegati nella distribuzione del video indicando il tipo di servizio in grado di erogare (Live o Video on demand VoD) il target di dispositivi a cui è rivolto(PC, TV o dispositivi d'elettronica di consumo CE), la topologia impiegata, i principi caratteristici del sistema in esame abbinati al tipo di incentivo utilizzato tra i peer, i sistemi impiegati per la garanzia della sicurezza, i codec adoperati e l'ambito da cui i sistemi in esame provengono (progetti di ricerca accademici o industriali)

Il quadro offerto dalla tabella evidenzia come esistano vari sistemi sviluppati sia in ambito commerciale che in ambito di ricerca e con un target che non si riferisce esclusivamente al mondo del PC ma anche al mondo della TV (attraverso l'utilizzo di STB *Set-Top-Boxes*) e dei dispositivi elettronici quali, ad esempio, *smartphones*. Allo scopo di permettere una diffusione della tecnologia P2P a livello globale è sempre più forte la necessità dell'introduzione di standard in grado di uniformare il variegato mondo del P2P dando delle linee guida comuni allo sviluppo ed implementazione di tale tecnologia. Una forte mancanza nei sistemi in esame e soprattutto in quelli in ambito di ricerca-sperimentale è rappresentato dalla non considerazione delle difficili e complesse tematiche che riguardano la sicurezza. Nei sistemi commerciali che hanno maggior successo sono impiegati ed utilizzati sistemi proprietari che garantiscono una corretta autenticazione dei peer, una gestione dei D.R.M ed una protezione dai più comuni attacchi perpetrati ai sistemi P2P. L'ambito della sicurezza risulta a tutt'oggi il più carente quanto a studi e soluzioni per la garanzia di uno sviluppo di soluzioni affidabili ed impiegabili come reale alternativa ai tradizionali vettori di distribuzione video.

TECNOLOGIA	S.D.	S.O.	TARGET	TOPOLOGIA	REWARDING-PECULIARITA'	SECURITY	CODIFICA	PROV.
Anysee [83]	P2P	LIVE	PC	Mesh e multipleoverlay	Località: utilizzo del peer più vicino	-	Single stream	Accademico Impiegato
Prime [15]	P2P	LIVE	PC	Tree-Mesh	Ricerca ed eliminazione dei colli di bottiglia dovuti alla banda e mancanza contenuto	-	MDC	Ricerca commerciale Simulazione
SecureStream[61]	P2P	LIVE	PC	Mesh	Costruito per tollerare i più comuni attacchi quali *pollution attack* e DoS	Digest, auditing, protocolli Fireflies, firma dei blocchi	H.264 AVC	Accademico Simulazione
R^2 [79]	P2P	LIVE	PC	Mesh	Adattabilità al cambiamento topologico della rete. Push e NC	-	Random network coding	Accademico Simulazione
PULSE [84]	P2P	LIVE	PC	Mesh	Tit-for-tat per selezione dei vicini	-	H.264 AVC	Ricerca – simulazioni
Poems [85]	P2P	LIVE	PC	Tree	Object-based audio-visual quality adaptive mechanism media-aware, network-aware,	-	MPEG4	Ricerca-simulazioni
Dag-stream [86]	P2P	LIVE	PC-CE-Mobile	Tree	Adattabilità dello stream multimediale a terminali eterogenei	-	Mpeg21 gBSD coding	Ricerca commerciale
SmartPeerCast [19]	P2P	LIVE	PC	Tree-Mesh	Tit-for-tat – raggruppamento in cluster	-	Adaptive single stream	Accademico Prototipo
Chunky Spread [87]	P2P	LIVE	PC	Multi - Tree	Tit-for-tat	-	MDC	Accademico simulazione
StreamComplete [29]	P2P	LIVE	PC	Tree-Mesh	Ottimizzazione in base alla funzione salute dei peers	-	H.264 AVC	
SplitStream [24]	P2P	LIVE	PC	Multi Tree	Credit-based		MDC	Proprietario
Zig-Zag [88]	P2P	LIVE	PC	Tree	Suddivisione in cluster permette riduzione ritardi end-to-end e facile gestione join	-	Mpeg-4 WMV	Ricerca-Simulazione
Coolstreaming [40]	P2P	LIVE	PC	Mesh	Peer-adaptation	-	Mpeg-4 WMV	Proprietario
Bullet [89]	P2P	LIVE	PC	Mesh-Tree	Creazione di un singolo albero al disopra della rete mesh – TCP friendly rate control	-	MDC – LT codec – Redundant Tornado Codec	Ricerca-Simulazioni
mTreebone [80]	P2P	LIVE	PC	Tree-Mesh	Algoritmi di ottimizzazione della struttura dell'albero di distribuzione	-	Mpeg-4	Accademico Simulazione
Samsung-P2P	P2P	LIVE, VoD	TVs	-	-	Autenticazione esplicita richiesta	-	Proprietario
NextShare [8]	P2P	LIVE, , VoD	PC, CE	Mesh	Collaborazione con ISP – NAPA-WINE	SHA1 – RSA digital signature	Speex, G722 PCMA, PCMU, iLBC and GSM	Ricerca commerciale
Abacast [90]	CDN+ P2P	LIVE, , VoD	PC	CDN+P2P	Network awarness	D.R.M.	H.264 AVC/SVC	Proprietario

Figura 6.3: Tabella 1/2

TECNOLOGIA	S.D.	S.O.	TARGET	TOPOLOGIA	REWARDING-PECULIARITA'	SECURITY	CODIFICA	PROV.
HRT [95]	P2P	LIVE	PC	Tree+ring	Impiego di anelli DDR, tempo di recovery ridotto	-	Mpeg-4	Accademico Simulazione
Velocix [68]	CDN+ P2P	LIVE, , VoD	PC, CE	CDN+P2P	Network awarness	Microsoft D.R.M. SHA-1 checksums	H.264 AVC/SVC Flash Video Streaming Service (FVSS)	Proprietario
PPLive [91]	P2P	LIVE, VoD	PC-Set top Box - Mobile	Mesh	Collaborazione con ISP attraverso posizionamento strategico peers.	D.R.M. – Message data encripted	RealVideo, (RMVB) Windows Media Video(WMV)	Proprietario
SopCast [92]	P2P	LIVE	PC STB	Mesh	Peer uploading bandwith, premiati i peers che contribuiscono maggiormente	D.R.M. – Autentication server; End-to-End security, encrypted messages.	H.264, WMV, WMA, ASF, RM,RMVB	Proprietario
TVAnts [93]	P2P	LIVE	PC	Mesh	Favorisce scambi tra peer vicini e appartenenti stesso AS	Server D.R.M. centralizzato	H. 264, WMV	Proprietario
PPStream [94]	P2P	LIVE	PC	Mesh	Rate-based peer selection Raggruppamento geografico	Message data encripted		Proprietario
Octoshape [67]	CDN-P2P	LIVE VOD	PC	CDN-P2P	Posizionamento geografico	D.R.M. client-server	H.264 AVC/SVC	Proprietario
Zatoo [94]	P2P	LIVE	PC	Tree Mesh. Hybrid CDN-P2P con impiego di Repeater node	Peer-division multiplexing – Topology-aware overlay permette riduzione del link, minimizza traffico tra ISP diversi	Media encription dedicato e Autentication server- Uso certificati	H.264/AAC Reed-Solomon (RS) error correttin code (ECC)	Proprietario

Figura 6.4: Tabella 2/2

6.3 Analisi SWOT

Di seguito viene presenta l'analisi SWOT riguardante l'impiego della tecnologia P2P dal punto di vista di un emittente evidenziando le criticità ed i vantaggi ottenibili dall'utilizzo di questa tecnologia.

Punti di forza

- Molto efficiente per la diffusione di contenuti verso un'utenza elevata.
- Utilizzo di infrastruttura già esistente.
- Nessuna necessità di nuovo hardware: soluzione software.
- Supporto sia live-streaming che VoD
- Scalabilità
- Possibilità di integrare il sistema in *smartphone* e apparecchiature portatili
- Integrazione con sistema CDN

Punti di debolezza

- Basato su una rete best-effort
- Fondato sul comportamento altruistico ed onesto dei partecipanti
- Possibile strumento per diffusione di materiale non autorizzato
- Difficile controllo del processo di distribuzione nei sistemi completamente decentralizzati
- Ritardi d'accesso ai contenuti *live* elevati rispetto a sistemi *unicast*

Opportunità

- Nuovo canale di distribuzione accessibile e di facile utilizzazione
- Possibilità di distribuire *user-generated-content* ad un vasto pubblico
- Distribuzione verso mercati specifici e community di utenti.
- Trasmissioni in alta definizione e 3D

Minacce

- Grande competizione nel mercato
- ISP potrebbero rallentare o bloccare il traffico P2P
- Possibilità di creare mercati di nicchia e non raggiunger più la grande distribuzione
- Possibili problemi di sicurezza

Dall'osservazione di quest'analisi si evince come la scelta dell'impiego della tecnologia P2P ponga molteplici sfide da affrontare e al contempo offra delle opportunità uniche in grado di cambiare radicalmente il tradizionale metodo utilizzato per la distribuzione video.

Capitolo 7

Conclusioni

Nel presente lavoro di tesi sono stati presentati ed esaminati i recenti studi che la comunità scientifica ha proposo riguardo le tematiche della distribuzione di materiale audio-video *live streaming* attraverso l'impiego di reti peer-to-peer. La ricerca condotta ha abbracciato in modo completo gli ambiti riguardanti i fondamenti della tecnologia peer-to-peer nell'analisi delle classiche topologie *tree* e *mesh* per poi presentare le innovative ricerche di soluzioni ibride in grado di fondere i vantaggi offerti dalle singole soluzioni. Il frutto di queste ricerche ha portato alla modellizzazione di sistemi che impiegano topologia ad anello-albero, albero-mesh e all'integrazione delle reti CDN (*Content Delivery Network*) con le reti peer-to-peer. Si è osservato come ciascuna di queste soluzioni ibride porti un notevole vantaggio rispetto ai modelli tradizionali. Lo studio ha altresì evidenziato come il problema del *live-video streaming* non riguarda esclusivamente le tematiche concernenti la topologia impiegata ma si espande ad altri campi di ricerca come quello dei sistemi d'incentivo e cooperazione tra peers e livelli di rete, le codifiche video e la sicurezza. Nell'ambito della cooperazione particolare attenzione è stata rivolta alle tecniche impiegate per la gestione del comportamento tenuto dai

peers durante la sezione video e all'analisi dei sistemi che permettono una comunicazione tra il livello applicativo e il livello di trasporto favorendo una scelta intelligente durante la creazione delle rete overlay allo scopo di ottimizzare l'impiego di risorse richieste alla rete underlay. Scopo del sistema di distribuzione è la diffusione verso più utenti di contenuti audio-video attraverso la rete Internet: i fondamenti della codifica video sono stati proposti rivolgendo particolare attenzione alle tecniche di codifica attraverso l'impiego di descrizioni multiple (MDC) e alle codifiche scalabili (SVC) che risultano essere un'ottima soluzione da applicare in ambienti eterogenei grazie all'adattabilità che offrono. L'innovativo approccio offerto dalle tecniche di *network coding* introduce capacità elaborativa all'interno dei singoli nodi che sono quindi in grado di rielaborare l'informazione da loro posseduta ed ottimizzare le risorse a loro disposizione attraverso le tecniche offerte dall'impiego di *linear network coding*. La presenza di molte entità convolte ed il trasferimento di materiale molto spesso coperto da *copyright* ho portato all'esame delle problematiche riguardanti la sicurezza e la gestione dei D.R.M. (*Digital Right Managment*). Il lavoro di tesi si è concluso con la presentazione di un sistema ottimale che raggruppa in sé tutte le tematiche precedentemente descritte allo scopo di presentare un'architettura in grado di far fronte alle sfide proposte dalla distribuzione *peer-to-peer video streaming*. A livello topologico il sistema proposto adotta una soluzione ibrida albero-mesh attraverso l'impiego di super-nodi che svolgono il ruolo di supervisori nei confronti dei peers presenti nel proprio cluster geografico di controllo. Il sistema adotta l'impiego di tecniche di codifica scalabili SVC che offrono un doppio vantaggio: la possibilità di adattamento alle condizioni eterogenee delle rete e la riduzione delle operazioni computazionali richiesta in fase di cifratura del contenuto video. La ricezione del *base layer* è indispensabile per la decodifica

del flusso video quindi si sfrutta tale bisogno andando a cifrare solamente il livello base lasciando gli *enhancement layers* privi di cifratura. Tale scelta implementativa permette un notevole risparmio in termini di capacità elaborativa e tempo richiesto per eseguire le procedure di protezione del flusso video. Un sistema di *monitor e reward* è stato implementato per permettere al sistema il controllo verso utenti malintenzionati e proteggersi contro i più comuni attacchi rivolti alle reti peer-to-peer. La presenza di servers dedicati all'autenticazione, al controllo e al monitoraggio del sistema ha permesso lo sviluppo di una soluzione affidabile, sicura e scalabile pronta ad esser impiegata nella distribuzione di materiale audio-video *live streaming* attraverso le reti peer-to-peer.

Bibliografia

[1] Christophe Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, Doug Balensiefen , *"Deployment Issues for the IP Multicast Service and Architecture"*

[2] J. Buford, H. Yu, and E. K. Lua. , *"P2P Networking and Applications"*, Morgan Kaufmann ,2008

[3] http://wikimediafoundation.org/wiki/Home

[4] http://www.p2p-next.org/

[5] http://www.bittorrent.com/

[6] Vlavianos, A.; Iliofotou, M.; Faloutsos, M. , *"BiToS: Enhancing BitTorrent for Supporting Streaming Applications"*, INFOCOM 2006. 25th IEEE International Conference on Computer Communications, April 2006

[7] http://torrentfreak.com/bittorrent-p2p-live-streaming-110119/

[8] http://swarmplayer.p2p-next.org/

[9] http://www.youtube.com/watch?v=QCFm_V9xmOc

[10] http://labs.adobe.com/technologies/cirrus/

[11] http://torrentfreak.com/

[12] Jie Wu Zhihui lu Bisheng Liu Shiyong Zhang, *"PeerCDN: A Novel P2P Network Assisted Streaming Content Delivery Network Scheme"*

[13] Magharei, N.; Rejaie, R.; Yang Guo; , *"Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches"*, INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE , vol., no., pp.1424-1432, 6-12 May 2007

[14] Ouali, A.; Kerherve, B.; Jaumard, B.; ,*"Toward new peering strategies for push-pull based P2P streaming systems"*, Ultra Modern Telecommunications and Workshops, 2009. ICUMT '09. International Conference on , vol., no., pp.1-8, 12-14 Oct. 2009

[15] Magharei, N.; Rejaie, R. *"PRIME: Peer-to-Peer Receiver-Driven Mesh-Based Streaming"*, Networking, IEEE/ACM Transactions on , vol.17, no.4, pp.1052-1065, Aug. 2009

[16] Meng Zhang, Li Zhao, Yun Tang, Jian-Guang Luo,Shi-Qing Yang, *"Large-scale live media streaming over peer-to-peer networks through global internet"*, P2PMMS'05 Proceedings of the ACM workshop on Advances in peer-to-peer multimedia streaming

[17] http://www.planet-lab.org/

[18] ZhiHui Lu, You Li, Jie Wu, ShiYong Zhang, YiPing Zhong, *"Multi-PeerCast: A Tree-mesh-hybrid P2P Live Streaming Scheme Design and Implementation based on PeerCast"*

[19] Wenyi Wang and Yaowu Chen, *"SmartPeerCast: a Smart QoS driven P2P live streaming framework"*

[20] Huey-Ing Liu and I-Feng Wu , *" MeTree: A contribution and Localyti-Aware P2P Live Streaming Architectute"*

[21] Nen-Fu Huang, Yih-Jou Tzang, Hong-Yi Chang, Chia-Wen Ho , *"Enhancing P2P overlay network architecture for live multimedia streaming"*

[22] Y. Chu, S. G. Rao, and H. Zhang, *"A case for end system multicast"* ACM SIGMETRICS, 2000

[23] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, *"Scalable application layer multicast"* ACM SIGMETRICS, 2000

[24] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron and A. Singh, *"Splitstream: High-bandwidth multicast in cooperative environments"* ACM SOSP, 2003.

[25] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanid-kulchai, *"Distributed streaming media content using cooperative networking"*, ACM NOSSDAV, 2002.

[26] D. A. Tran, K. A. Hua, T. T. DOA, *"A peer-to-peer architecture for media streaming"*, IEEE Journal on Selected Areas in Communications, Vol. 22, n. 1, pp: 121-133, 2004.

[27] http://www.PeerCast.org.

[28] Nen-Fu Huang, Yih-Jou Tzang, Hong-Yi Chang, Chia-Wen Ho, *"Enhancing P2P overlay network architecture for live multimedia streaming"*, 2010

[29] Covino, F.; Mecella, M., *"StreamComplete: An Architecture for Mesh-Based Peer-to-Peer Live Video Streaming"*, Consumer Communications

and Networking Conference, 2009. CCNC 2009. 6th IEEE , vol., no., pp.1-5, 10-13 Jan. 2009

[30] H. Schwarz, D. Marpe, and T. Wiegand, *"Overview of the scalable video coding extension of the H.264/AVC standard."* IEEE Transactions on Circuits and Systems for Video Technology, 17(9):1103–1120, September 2007.

[31] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, *"Overview of the h.264/avc video coding standard."* IEEE Transactions on Circuits and Systems for Video Technology, 13(7):560–576, July 2003.

[32] Zhengye Liu; Yanming Shen; Ross, K.W.; Panwar, S.S.; Yao Wang, *"LayerP2P: Using Layered Video Chunks in P2P Live Streaming Multimedia"* IEEE Transactions on , vol.11, no.7, pp.1340-1352, Nov. 2009

[33] Ahlswede, R.; Ning Cai; Li, S.-Y.R.; Yeung, R.W., *"Network information flow"*, Information Theory, IEEE Transactions on , vol.46, no.4, pp.1204-1216, Jul 2000

[34] Ho, T.; Medard, M.; Koetter, R.; Karger, D.R.; Effros, M.; Jun Shi; Leong, B.; ,, *"A Random Linear Network Coding Approach to Multicast"*, Information Theory, IEEE Transactions on , vol.52, no.10, pp.4413-4430, Oct. 2006

[35] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, *"The benefits of coding over routing in a randomized setting"*,In ISIT, July 2003.

[36] S. Jaggi, P. Sander, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, *"Polynomial time algorithms for network code construction"*,IEEE Transaction on Information Theory,2001.

[37] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, *"The benefits of coding over routing in a randomized setting"*,In ISIT, July 2003.

[38] T. Ho, M. Medard, J. Shi, M. Effros, and D. Karger, *"On Randomized Network Coding"*,Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing, Oct. 2003.

[39] Yajie Liu; Yuxing Peng; Wenhua Dou; Bo Guo, *"Network Coding for Peer-to-Peer Live Media Streaming"*,Grid and Cooperative Computing, Fifth International Conference , Oct. 2006

[40] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum., *"Coolstreaming/ donet: a data-driven overlay network for live media streaming."*,In Proceedings of IEEE INFOCO, pages 28–39,Miami,FL,USA,March 2005.

[41] Guang Tan; Jarvis, S.A., *"A Payment-Based Incentive and Service Differentiation Scheme for Peer-to-Peer Streaming Broadcast."* IEEE Transactions on , vol.19, no.7, pp.940-953, July 2008

[42] Hoong, P.K. Matsuo, H., *"PALMS: A Reliable and Incentive-Based P2P Live Media Streaming."*, 2008.

[43] http://napa-wine.eu/cgi-bin/twiki/view/Public

[44] V. Aggarwal, A. Feldmann, C. Scheideler, *"Can ISPS and P2P users cooperate for improved performance?"*,SIGCOMM Comput. Commun.Rev. Vol.37, N.3, pp.29-40, Jul. 2007.

[45] J.Seedorf, S.Kiesel, M.Stiemerling, *"Traffic Localization for P2P Applications: The ALTO Approach."*,IEEE P2P 2009, Seattle, WA, Sept. 2009.

[46] T. S. E. Ng, H. Zhang, *"Predicting internet network distance with coordinates-based approaches."*,In Proc. IEEE Infocom, 2001, Vol. 1, pp.170–179, New York, NY, USA, June 2002.

[47] Y. Yardi, *"Network Tomography: estimating source-destination traffic intensities from link data."*, Journal American Statistics Association, pp.365-377,1996.

[48] L. Abeni, C. Kiraly, R. Lo Cigno, *"On the Optimal Scheduling of Streaming Applications in Unstructured Meshes."*,In Proc. IFIP Networking 2009, Aachen, Germany, May 11–15, 2009

[49] N. Courtois, M. Finiasz, and N. Sendrier, *" How to achieve a McEliece-based digital signature scheme"*,Cryptology - ASIACRYPT 2001.

[50] N. Courtois, L. Goubin, and J. Patarin, *" Quartz, 128-bit long digital signatures."*,Rsa Conference, 2001.

[51] J. Park, E. Chong, and H. Siegel, *" Effcient multicast packet authentication using signature amortization"*, IEEE Symposium on Security and Privacy, 2002.

[52] S. Miner and J. Staddon, *"Graph-based authentication of digital streams."*, IEEE Symposium on Research in Security and Privacy, 2001.

[53] C. Wong and S. Lam, *"Digital signatures for flows and multicasts."*, IEEE/ACM Transactions on Networking, 7, 1999.

[54] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. *"Multicast security: A taxonomy and some efficient constructions."*,IEEE Infocom, 1999.

[55] Meier, R.; Wattenhofer, R.; , *"ALPS: Authenticating Live Peer-to-Peer Live Streams"*, Reliable Distributed Systems 2008. SRDS '08. IEEE Symposium on , vol., no., pp.45-52, 6-8 Oct. 2008

[56] S. Rafaeli, and D. Hutchison, *"A Survey of Key Management for Secure Group Communication"*, ACM Computing Surveys,35(3), pp.309-329, Sep 2003.

[57] W. Trappe, J. Song, and K.J.R. Liu, *"Key Management and Distribution for Secure Multimedia Multicast"*, IEEE Transactions on Multimedia, 5(4), pp.544-557, Dec 2003.

[58] F. Qiu, C. Lin, H. Yin, *" EKM: An Efficient Key Management Scheme for Large-scale Peer-to-Peer Media Streaming"*, in Proc of PCM 2006, LNCS, Nov 2006.

[59] Xuening Liu; Hao Yin; Chuang Lin; Yiping Deng, *"Efficient key management and distribution for peer-to-peer live streaming system"*, Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007 Nov 2007-Dec. 1 2007

[60] Prithula Dhungel, Xiaojun Hei, Keith W.Ross, Nitesh Saxena *"The pollution attack in P2P live video streaming: measurement results and defenses"*

[61] Maya Haridasan, Robbert van Renesse *"SecureStream: An intrusion-tolerant protocol for live-streaming dissemination"* Computer Communications, 2008

[62] *"Cisco Visual Networking Index: Forecast and Methodology, 2009–2014"*

[63] http://www.multichannel.com/_article/191223_YouTubeAnalysts.php

[64] http://bits.blogs.nytimes.com/_2009/01/20/news-sites-struggle-to-stream-obamas-innaguration-speech/?apage=1

[65] http://www.akamai.com/

[66] http://uk.limelightnetworks.com/

[67] http://www.octoshape.com

[68] http://www.velocix.com/

[69] http://www.chinacache.com

[70] H Yin, X Liu, T Zhan, V Sekar, F Qiu, *"Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky"* Proceedings of the seventeen ACM international conference on Multimedia, 2009

[71] Subramanya, S.R., Yi, B.K. *"Digital Rights Management"*, IEEE Potentials 25(2), 2006

[72] http://www.microsoft.com/windows/windowsmedia/licensing/drmlicensing.aspx

[73] http://www.realnetworks.com/helixplatform.aspx

[74] Lie Liu and Chun Yuan, *"Broadcast Encryption-Based P2P DRM without Central License Server"*,Advances in Multimedia Information Processing - PCM 2009

[75] Xuguang Lan; Jianru Xue; Lihua Tian; Wei Hu; Tao Xu; Nanning Zheng; , *"A Peer-to-Peer Architecture for Live Streaming with DRM"*, Consumer Communications and Networking Conference, 2009.

[76] Filippini, A.; Bergamo, P.; Mazzini, G., *"Security issues based on chaotic systems"* Global Telecommunications Conference,2002

[77] A. Singh, M. Castro, P. Druschel, and A. Rowstron, *"Defending against eclipse attacks on overlay networks"*, Proceedings of the 11th workshop on ACM SIGOPS European workshop, 2004

[78] Adrian Perrig *"The BiBa one-time signature and broadcast authentication protocol"*,Proceedings of the 8th ACM conference on Computer and Communications Security 2001

[79] Mea Wang, Baochun Li, *"R2: Random Push with Random Network Codingin Live Peer-to-Peer Streaming"*, IEEE Journal on selected areas in communivation, VOL. 25, NO. 9, DECEMBER 2007

[80] Feng Wang; Yongqiang Xiong; Jiangchuan Liu; *"mTreebone: A Collaborative Tree-Mesh Overlay Network for Multicast Video Streaming"* ,Parallel and Distributed Systems, March 2010

[81] J. R. Douceur, *"The Sybil Attack"*, In Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002.

[82] Jae-Youn Sung; Jeong-Yeon Jeong; Ki-Song Yoon; *"DRM Enabled P2P Architecture"*, Advanced Communication Technology, 2006.

[83] X. Liao, H. Jin, Y. Liu, L. M. Ni, D. Deng, *"AnySee: Peer-to-Peer Live Streaming"*, ,NFOCOM 2006. Proceedings In INFOCOM 2006.

25th IEEE International Conference on Computer Communications. Proceedings 2006

[84] Fabio Pianese, Diego Perino, Joaquín Keller, and Ernst W. Biersack, *" PULSE: an Adaptive, Incentive-based, Unstructured P2P Live Streaming System"*, IEEE TRANSACTIONS ON MULTIMEDIA

[85] Poems: Ahmed T, Mushtaq M, *"P2P Object-based adaptivE Multimedia Streaming (POEMS)"*, 2007

[86] Razib Iqbal, Shervin Shirmohammadi, *"DAg-stream: Distributed video adaptation for overlay streaming to heterogeneous devices"*, Springer Science, 2009

[87] Vidhyashankar Venkataraman, Paul Francisy, John Calandrino, *"Chunkyspread: Multitree Unstructured PeertoPeer Multicast"*, 2006

[88] Tran, D.A.; Hua, K.A.; Do, T. , *"ZIGZAG: an efficient peer-to-peer scheme for media streaming"*, INFOCOM 2003.

[89] Kostic D, Rodrigues A, Albrecht J, and Vahdat A. *"Bullet: high bandwidth data dissemination using an overlay mesh"*, SOSP 2003

[90] http://www.abacast.com

[91] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, *"Insights into PPLive: A Measurement Study of a Large-Scale P2P IPTV System"*, Proc. Workshop IPTV Services over World Wide Web, 2006.

[92] http://www.sopcast.org

[93] Jinkang Jia; Chunxi Li; Changjia Chen; , *"Characterizing PPStream across Internet"*, Network and Parallel Computing Workshops, 2007.

[94] Hyunseok Chang, Sugih Jamin , Ann Arbor, Wenjie Wang, *" Live streaming performance of the Zattoo network"*, IMC '09 Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, 2009

[95] Nen-Fu Huang; Yih-Jou Tzang; Hong-Yi Chang; Chih-Shun Ma, *"Construction of an efficient ring-tree-based Peer-to-Peer streaming platform"*, Networked Computing and Advanced Information Management (NCM), 2010

[96] M. Bishop, S. Rao, and K. Sripanidkulchai, *"Considering Priority in Overlay Multicast Protocols under Heterogeneous Environments"*,Proc. IEEE INFOCOM, pp. 1-13, Apr. 2006

[97] Havard Johansen, André Allavena, Robbert van Renesse, *"Fireflies: scalable support for intrusion-tolerant network overlays"*,EuroSys '06 Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006

[98] Thomas Stutz and Andreas Uhl, *"A Survey of H.264 AVC/SVC Encryption"*, Technical Report Deparment of computer science Salzburg University, 2009

[99] Su-Wan Park and Sang-Uk Shin, *"Combined Scheme of Encryption and Watermarking in H.264/Scalable Video Coding (SVC)"*, Studies in Computational Intelligence, 2008