SCUOLA DOTTORALE DI INGEGNERIA
INGEGNERIA DELL'ELETTRONICA BIOMEDICA,
DELL'ELETTROMAGNETISMO E DELLE TELECOMUNICAZIONI

XXIII CICLO

Codifica di canale orientata al pacchetto

Ing. Elena Mammi

A.A. 2010/2011

Docente Guida - Relatore: Prof. Alessandro Neri

Coordinatore: Prof. Lucio Vegni

Correlatore: Ing. Paolo Talone

Abstract

Il presente lavoro di tesi di dottorato affronta la problematica del raggiungimento di un alto livello di qualità per un servizio di tipo televisivo trasmesso su reti con perdita, come Internet. Lo studio della qualità per servizi televisivi su IP risulta di grande attualità, in quanto tali servizi necessitano di una qualità dell'esperienza pari a quelli dei servizi televisivi tradizionali per potersi affermare. Nel lavoro di tesi si è focalizzata l'attenzione sulla valutazione dell'impatto delle tecniche di correzione dell'errore a livello applicativo (AL-FEC) sulla qualità del servizio e sulla qualità dell'esperienza, considerando diverse tipologie di perdita di pacchetti in rete. In una prima parte del lavoro si è effettuata un'analisi approfondita delle prestazioni di una tecnica AL-FEC standard a basso costo computazionale, per verificarne le capacità e i limiti, nei confronti di eventi di perdita propri di una reale rete IP. Successivamente, valutati gli eventi di perdita più critici, si è pensato ad una nuova strategia, attuabile nelle reti gestite, che permettesse di potenziare le prestazioni della tecnica precedente, ottenendo allo stesso tempo un basso impatto sui flussi concorrenti in rete. Di seguito si è studiato un sistema di monitoraggio della qualità percepita dal fruitore di un servizio televisivo tramite l'uso di metriche oggettive, al fine di mantenere la qualità percepita costante variando le configurazioni della tecnica di correzione d'errore. Così facendo non si va ad agire sulla rete, ma si lavora in maniera *end-to-end*, rendendo attuabile tale sistema in reti non gestite. In una seconda parte del lavoro si è considerata una nuova strategia che permettesse un multicast efficace in reti MANET con perdita, sfruttando le caratteristiche di codici maggiormente innovativi, ovvero i codici a fontana. Infatti durante lo studio della qualità di un servizio televisivo trasmesso su reti con perdita, si è visto come la parte più difficile sia la realizzazione della consegna di un servizio che sia allo stesso tempo robusto alle

perdite, e quindi con alta qualità, e che sia real-time. Sfruttando il fatto che se si utilizzano dei codici a fontana, allora per la ricostruzione del file sorgente non è più fondamentale sapere quali pacchetti sono stati ricevuti, ma solamente quanti, si è pensato che se il destinatario ricevesse i pacchetti non da una sola sorgente, ma da più sorgenti contemporaneamente, allora potrebbe iniziare a decodificare il flusso, una volta raggiunta la quantità necessaria per la ricostruzione, prima che le sorgenti finiscano di inviarlo completamente, diminuendo così la latenza complessiva. I risultati ottenuti mostrano l'efficacia dei sistemi sviluppati e le potenzialità della codifica di canale orientata al pacchetto per servizi televisivi trasmessi su reti con perdita.

A chi mi è stato sempre accanto.

Ringraziamenti

Il presente lavoro è stato svolto nell'ambito di una collaborazione tra Università Roma Tre e la Fondazione Ugo Bordoni; desidero quindi ringraziare sia il mio docente guida il Prof. Alessandro Neri, che l'Ing. Paolo Talone per aver reso possibile tutto questo, per quel che mi hanno insegnato e per il tempo che mi hanno dedicato.

Un ringraziamento speciale va all'Ing. Giuseppe Russo della Fondazione Ugo Bordoni per avermi sempre sostenuta e consigliata.

Desidero inoltre ringraziare tutto il gruppo di Telecomunicazioni di Roma Tre, ed in particolare l'Ing. Marco Carli che mi ha continuamente incoraggiata ad andare avanti.

Infine, ringrazio tutti gli amici FUB con cui ho condiviso i rigeneranti momenti di pausa ed i momenti di confronto, sia le mie amiche di sempre che quelle nuove e la mia famiglia per avermi supportato e sopportato quotidianamente.

Indice

Elenco delle figure

Elenco delle tabelle

Glossario

AL-FEC Application Layer - Forward Error Correction

ARQ Automatic Repet reQuest

ASI Asynchronous Serial Interface

ATIS Alliance for Telecommunications Industry Solution

AVP Audio-Visual Profile

BF Broadband Forum

CBR Costant Bit Rate

CE Consumer Edge

CoS Class of Service

CPU Central Processing Unit

CRC Cyclic Redundancy Check

DiffServ Differentiated Services

DSL Digital Subscriber Line

DTT Digital Terrestrial Television

DVB Digital Video Broadcasting

EC Error Concealment

EPON Ethernet Passive Optical Networ

FEC Forward Error Correction

HD High Definition

HSPA High Speed Packet Access

IMS IP Multimedia Subsystem

IntServ Integrated Services

IP Internet Protocol

IPTV Internet Protocol TeleVision

ITU-T International Telecommunication Union - Telecommunication

LT Luby Transform

LTE Long Term Evolution

MANET Mobile Ad-Hoc NETwork

MCTP Motion Compensated Temporal Prediction

MDS Minimum Distance Separable

MPEG Moving Picture Experts Group

MPLS Multi Protocol Label Switching

MSD Minimum Distance Separable

MTU Maximum Transfer Unit

NGN Next Generation Network

NORM NACK-Oriented Reliable Multicast Protocol

NTIA National Telecommunications and Information Administration

OIPF Open IPTV Forum

OSPF Open Shortest Path First

PC Personal Computer

PE Provider Edge

PLR Packet Loss Rate

PSNR Peak Signal to Noise Ratio

PUMA Protocol Unified Manet Announcement

QoE Quality of Experience

QoS Quality of Service

RAC Resource and Admission Control

REIN Repetitive Electrical Impulse Noise

RMT Reliable Multicast Transport

RTP Real time Transport Protocol

SACK Selective ACKnowledgement

SD Standard Definition

SLA Service Level Agreement

SMPTE Society of Motion Picture and Television Engineers

STB Set Top Box

TCP Transmission Control Protocol

TS Transport Stream

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

USB Universal Serial Bus

VBR Variable Bit Rate

VPLS Virtual Private LAN Services

VQM Video Quality Metric

WAN Wide Area Network

WiMAX Worldwide Interoperability for Microwave Access

XOR eXclusive OR

1

Introduzione

In questi ultimi anni accanto a classiche modalità di trasmissione dei segnali televisivi, come ad esempio la diffusione in modalità broadcast mediante satellite o su onde terrestri (DTT), si sta affermando una nuova modalità di trasmissione in cui i dati vengono trasportati su Internet. Tali servizi televisivi trasmessi sulla rete Internet si stanno consolidando come la modalità emergente per una nuova fruizione televisiva.

La Televisione su Internet apre infatti prospettive inedite per nuove tipologie di servizio e modalità innovative di fruizione. In estrema sintesi, si tratta dell'opportunità di affiancare ai tradizionali ‘‘Servizi Lineari" (nel senso della direttiva UE ‘‘Media senza frontiere") propri del Broadcast Televisivo, i nuovi servizi ‘‘On demand" (non lineari) che aprono prospettive di mercato innovative e che non si adatterebbero alle tradizionali piattaforme broadcast a radiofrequenza.

La principale criticità della Televisione su Internet è universalmente riconosciuta essere la qualità del servizio.

Qualora la rete utilizzata per il servizio televisivo su IP sia del tipo ‘‘gestito", e quindi sotto il controllo di un operatore Telco, allora si può fornire una qualità di trasporto, sfruttando dei meccanismi di rete che consistono essenzialmente nella

‘‘prioritizzazione” del traffico sui router e/o nell’instradamento su percorsi con banda riservata. In questo caso si parla di *IPTV* e in questo scenario l’operatore Telco non è più solo il fornitore del servizio di trasporto, ma anche il fornitore di servizi, regolando le politiche di accesso alla sua rete ed anche le politiche tariffarie.
Nel caso in cui la rete sia ‘‘non gestita”, Open Internet, allora non è possibile ricorrere a meccanismi di rete per il raggiungimento dei vincoli di QoS, tipici dei servizi di diffusione televisiva tradizionali, e il trasferimento dati avverrà in modalità ‘‘best effort”; si parla in questo caso di *Web-TV*. In questo scenario, la qualità del servizio è un fattore molto critico, tuttavia in questo scenario l’utente può accedere ad un panorama di contenuti di dimensione globale, dialogando con i fornitori dei servizi che si affacciano direttamente su Internet senza la mediazione di Operatori Telco. Infatti la televisione su Open Internet non altera sostanzialmente la catena del valore di un servizio televisivo.

La televisione su Internet possiede anche il vantaggio di rappresentare una delle principali declinazioni del principio di ‘‘neutralità’’ della rete di trasporto rispetto all’erogazione dei servizi operata dai diversi fornitori. Essendo, in questo caso, la rete di trasporto costituita dall’infrastruttura aperta derivante dall’interconnessione delle differenti sezioni di rete sotto il controllo di operatori diversi, che, nel loro insieme costituiscono Internet.

Nell’ambito della televisione su IP, in particolare per la Web-TV, la *codifica di canale a pacchetto* assume una particolare rilevanza e su di essa, negli ultimi anni, la ricerca si è molto concentrata, in quanto questa permetterebbe l’affermazione delle televisione su Internet come vera alternativa, permettendo il raggiungimento

dei livelli di qualità tipici dei servizi televisivi tradizionali. Infatti è necessario che venga raggiunto un livello di qualità percepita dall'utente (Quality of Experience) che possa esser paragonabile a quello proprio delle altre piattaforme di diffusione della TV digitale.

Da un punto di vista sociale, la nascita dell'esigenza di un servizio come la televisione su IP parte dall'evoluzione dello stile di fruizione da parte degli utenti, i quali da una parte sono sempre più disposti a pagare per servizi di tipo simil-televisivo, e dall'altra sono sempre più abituati ad un consumo ‘‘On Demand", che le reti IP permettono meglio di tutte le altre piattaforme di distribuzione.
La possibilità di usufruire di contenuti ‘‘On Demand" da parte dell'utente dà vita ad un fenomeno chiave per i servizi televisivi su IP, che è il fenomeno della coda lunga. Difatti con tali servizi si avranno a disposizione ampi cataloghi di contenuto ‘‘On Demand" da consumare quando e come l'utente vuole e questo potrebbe generare modelli di business basati sul principio della coda lunga. Ci sarà infatti la possibilità di gestire un catalogo virtuale pressochè illimitato e questo permetterà di vendere oltre che migliaia di copie di pochi titoli, anche poche copie di migliaia di titoli.
Attualmente si dedica ‘‘il massimo dello spazio ai prodotti che hanno il massimo di compratori", sacrificando un lunghissimo elenco di contenuti che hanno una richiesta minore. Tuttavia esiste un mercato potenziale dato dalle decine di migliaia di titoli lungo la coda, non disponibili nei tradizionali canali di distribuzione. Con il digitale invece il costo di archiviazione e spedizione viene abbattuto ed è possibile trovare un numero maggiore di contenuti e quindi la coda si allunga. Inoltre maggiore è la disponibilità di contenuti su Internet e maggiore è la crescita del fenomeno della ‘‘esplorazione della coda".

Nell'ambito del tema del raggiungimento di un alto livello di qualità del servizio, e più specificatamente nell'ambito della codifica di canale, si colloca questo lavoro di tesi dottorale. Tale lavoro cerca di esplorare i limiti delle tecniche di codifica di canale, in particolare delle tecniche di Forward Error Correction a livello applicativo, e di valutare delle possibili soluzioni alternative che ne aumentino le prestazioni, fino ad arrivare a sistemi che coinvolgano le più recenti tecniche che potrebbero permettere di raggiungere un'alta affidabilità di un servizio trasmesso su reti IP, non nate per trasmissioni affidabili.

La tesi è organizzata come segue. Dopo un'introduzione che propone lo stato dell'arte in merito alla qualità del servizio, nel capitolo 2 viene proposta l'analisi effettuata dell'efficacia di un sistema di protezione standard per la QoS. La strategia che propone la ‘‘prioritizzazione" del solo flusso di correzione viene descritta nel capitolo 3. Nel capitolo 4, viene descritta la valutazione dell'efficacia di un sistema di protezione per metriche di qualità oggettive. Il sistema di multicast efficace tramite codici a fontana in reti MANET proposto viene illustrato nel capitolo 5. Infine, nel capitolo 6 vengono tratte le conclusioni.

1.1 Il servizio televisivo su IP

La televisione su IP, ovvero la diffusione di servizi televisivi ad un'utenza dotata di accesso alla rete Internet, è il frutto della convergenza del broadcasting televisivo con i paradigmi di comunicazione tipici della rete Open Internet o delle reti IP proprietarie dei diversi Operatori che ad Internet fanno capo. I paradigmi di comunicazione per i servizi televisivi sono definiti dalla Direttiva Europea ‘‘Media

senza frontiere'', [13], approvata dal Parlamento Europeo nel 2007. Seguendo tale direttiva, i servizi di media audiovisivi si possono classificare nel seguente modo:

- *lineari*, i quali forniscono immagini in movimento con associati audio ed eventuali elementi grafici o testuali (sottotitoli), diffusi sulla base di un palinsesto prefissato, per cui l'utente gioca un ruolo passivo;

- *non-lineari*, i quali permettono all'utente di aver pieno controllo sul momento di fruizione. In altri termini un servizio messo a disposizione da un ''media service provider'' per la visione di programmi nel momento scelto dall'utente e su sua richiesta individuale (*on demand*), sulla base di un catalogo di programmi. In questo caso l'utente finale diventa un soggetto attivo.

Inoltre prende particolare importanza una classe di servizi non lineari, basata sul download di contenuti audiovisivi, ovvero di file visualizzabili una volta terminata l'operazione di download sul terminale d'utente. Va rilevato che i paradigmi di comunicazione di Internet rendono possibile interfacciare non solo gli utenti nella loro globalità (Multicast su rete IP) ma anche il singolo utente (Unicast), [32].

Dal punto di vista dei servizi, nel caso di servizi di televisione su IP, non viene stabilito il tipo di rete IP da utilizzare per il trasporto. Tuttavia le diverse reti di trasporto hanno prestazioni differenti che portano a diversi gradi di qualità del servizio. I maggiori enti di standardizzazione prevedono l'utilizzo di reti IP tradizionali in una prima fase dello sviluppo della televisione su IP, in vista della migrazione verso reti di nuova generazione (NGN), quando saranno disponibili. La tradizionale rete IP può essere gestita o non gestita. Una rete IP gestita è un

insieme di segmenti di rete gestiti da un solo Network Provider. Di solito questo operatore è anche fornitore di servizi, che seleziona quindi i contenuti da offrire, le restrizioni di accesso e la qualità del trasporto.
Al contrario, una rete IP non gestita è costruita su segmenti di rete sotto il controllo di diversi operatori: il paradigma di qualità dei trasporti è il classico ''best-effort'', e non ci sono intermediari tra l'utente finale e fornitori di servizi.
Le Next Generation Network sono reti gestite e di solito basate su un trasporto di tipo IP. Le NGN possono poi essere basate su IP Multimedia Subsystem o meno; nel primo caso la coesistenza di servizi televisivi con gli altri servizi di telecomunicazione è semplificata, nel caso secondo caso la televisione su IP deve essere solo integrata in ambiente NGN.
I servizi televisivi su una rete IP gestita, caratterizzata da un'alta qualità, sono di solitamente definiti come servizi ''IPTV''. In questo caso gli operatori Telco, fornendo l'accesso alla rete agli utenti, assumono la maggior parte dei funzioni tradizionalmente svolte dalle emittenti. I servizi televisivi su una rete non gestita IP (tipicamente l'Open Internet) dove la qualità non è garantita, sono solitamente denominati come ''Over the Top TV'' o ''Web-TV''. Questo ultimo scenario lascia intatta il ruolo tradizionale delle emittenti televisive, ma porta con sè maggiori problematiche per quanto riguarda la qualità del servizio.

1.2 La qualità del servizio

Nel caso di stream audio/video trasmessi su reti con perdite, la qualità del servizio è un fattore critico per il raggiungimento della soddisfazione dell'utente. Tuttavia, la trasmissione di servizi con qualità garantita è particolarmente difficile da ottenere su Open Internet, in quanto lavora in modalità best-effort. Dall'altra parte le esigenze dell'utente nel caso di servizi video, in particolare servizi televi-

sivi, sono molto elevate, in quanto l'utente finale è abituato a vedere la televisione in modalità analogica o tramite il digitale terrestre che garantiscono entrambe le modalità un'alta qualità. Ne consegue che il traffico televisivo su IP ha requisiti di banda, ritardo, jitter e rate di perdita di pacchetti molto stringenti; questi parametri non possono essere garantiti da una gestione best-effort. Risulta, quindi, necessario implementare diversi meccanismi addizionali per assicurare il livello di qualità appropriato, [39].
In generale i principali parametri che caratterizzano la qualità del trasporto su una infrastruttura di tipo IP sono le statistiche del trasferimento end-to-end e sono espresse in termini di:

- latenza (ritardo end-to-end),
- jitter (variazioni del ritardo istantaneo),
- affidalibità.

Se si considerano servizi On Demand, allora valori ragionevoli di jitter e ritardo end-to-end possono esser raggiunti grazie ai buffer dei Set Top Box, dato che la dimensione di tali buffer viene stabilita in maniera tale da essere compatibile con le prestazioni degli elementi di rete e video, [12]. Invece, un errore o una sequenza di errori in uno stream video può causare effetti variabili, da un impatto sul video o sull'audio non percepibile per l'utente fino alla perdita completa del segnale audio/video a seconda di quanto che è stato perso in rete e di quali pacchetti sono stati persi. Come già detto precedentemente è molto importante che un servizio televisivo su IP sia disponibile nella migliore qualità possibile, poichè il successo di tale servizio dipenderà dalla soddisfazione dell'utente finale. Esistono due diversi punti di vista delle prestazioni del sistema, legate alla consegna di servizi con qualità assicurata:

- la Quality of Experience,
- la Quality of Service.

La qualità dell'esperienza descrive le prestazioni del sistema nella sua globalità dalla prospettiva dell'utente finale ed è collegata al livello della qualità di servizio e alla capacità di un servizio di soddisfare le aspettative dell'utente.
La qualità del servizio rappresenta una misurazione delle prestazioni dalla prospettiva della rete e coinvolge un insieme di meccanismi che vengono impiegati per gestire le condizioni della rete. La relazione tra queste due qualità riguarda il problema di mappare i requisiti dell'utente finale e del servizio ai meccanismi adottati e ai parametri di trasporto.

1.2.1 Tipologie di perdite

Le perdite di pacchetti possono essere classificati in tre grandi tipologie, burst sparsi, burst continui e perdite random. Più nel dettaglio si definiscono:

- *burst sparsi*, periodi di grossa perdita, dell'ordine dei secondi, con gravi errori; un burst sparso è un periodo che inizia e finisce con un pacchetto perso o scartato, durante il quale si realizzano determinate condizioni, come il vincolo che all'interno di un burst vi siano meno di G_{min} pacchetti consecutivi ricevuti. G_{min} viene scelto in maniera tale da corrispondere al tasso minimo di perdita per il quale si presenta qualche distorsione visibile all'interno dello stream decodificato. Una delle cause principali dei burst sparsi sono le congestioni di rete;
- *burst continui*, periodi durante i quali vengono persi tutti i pacchetti, che possono avvenire ad esempio a causa dei guasti dei link all'interno di una rete IP;

- *perdite isolate*, pacchetti isolati persi tipicamente a causa di bit error nella trasmissione o di collisioni eccessive nelle reti ad area locale.

Solitamente vengono posti sette pacchetti Transport Stream MPEG-2 in un singolo pacchetto UDP per porre più pacchetti TS possibili in una singola trama Ethernet, senza superare la dimensione della Maximum Transfer Unit di 1500 byte, per minimizzare la quantità di overhead. Con questa pacchettizzazione un singolo bit perso conduce alla perdita dell'intera trama Ethernet. Le strutture TS MPEG possono essere sincronizzate alla trama IP, ma le immagini MPEG non sono sincronizzate ai pacchetti IP e in genere ne occupano un certo numero. Se il pacchetto che viene perso include l'informazione di header della trama di immagine, allora può essere persa l'intera trama. L'informazione video MPEG è organizzata in trame I (che sono complete), trame B e trame P (che vengono interpolate tra trame di immagini successive).
Il decodificatore MPEG può avere alcune opzioni per occultare gli errori, ma nel caso peggiore la perdita di un pacchetto può portare alla perdita dell'immagine, finchè non viene ricevuto il quadro I successivo. Ogni quadro I può essere rappresentato indipendentemente dai quadri precedenti, permettendo quindi al ricevitore di recuperare l'effetto di una precedente perdita di pacchetti, ma le trame I sono più grandi di quelle P e B e quindi usano maggiore banda. Di conseguenza, inviare trame I ad un rate più alto riduce il tempo di recupero di una perdita di pacchetti e il ritardo tra il momento in cui un ricevitore si unisce ad uno stream in corso e il momento in cui può iniziare a rappresentare lo stream; tuttavia si paga il prezzo di un maggiore utilizzo della banda. C'è quindi bisogno di un compromesso tra la capacità di recupero degli errori e l'occupazione di banda.

Invece i principali eventi che possono incidere sull'affidabilità di un servizio trasmetto sono:

- scarto di pacchetti da parte di router congestionati;
- scarto di pacchetti da parte di ricevitori in caso di fuori sequenza o di pacchetti duplicati;
- scarto di pacchetti da parte di ricevitori in caso di ricezione di pacchetti con un ritardo superiore al massimo accettabile;
- scarto di pacchetti da parte di router o ricevitori in caso di ricezione di pacchetti danneggiati, i cui errori possono essere individuati, ma non possono essere recuperati da un algoritmo Forward Error Correction (ad esempio, i pacchetti con sbagliato Cyclic Redundancy Check);
- scarto di pacchetti corrotti perchè affetti da bit residui sbagliati non corretti da algoritmi FEC (in genere in presenza di collegamenti radio, come in WiMAX e UMTS HSPA e LTE);
- link failure che implicano un tempo di recupero necessario ai meccanismi di rete per impostare percorsi di rete alternativi.

1.3 La codifica di canale

Claude E. Shannon, nel suo lavoro del 1948 ‘‘The mathematical theory of communication" [36], diede una prima definizione quantitativa di informazione, da tale definizione nacque tutta la teoria dell'informazione. Nella sua opera Shannon dimostrò che la comunicazione affidabile su un canale inaffidabile è possibile. È quindi possibile che, data la trasmissione di un segnale su un canale che introduce errori o distrugge parte del messaggio, il destinatario possa comunque recuperare

il messaggio originale; questo risultato è conosciuto come il secondo teorema di Shannon, ovvero ‘‘teorema della codifica di canale’’.

Più specificatamente tale teorema stabilisce che quando un canale trasmissivo è affetto da rumore, allora è possibile trasmettere dati con probabilità d'errore piccola a piacere se il ritmo di trasmissione dell'informazione R è minore della capacità C. La capacità di Shannon, *limite di Shannon*, di un canale di comunicazione è il massimo tasso di trasferimento di dati che può fornire il canale per un dato livello di rapporto segnale-rumore, con un tasso di errore piccolo a piacere. Il teorema descrive quindi la massima efficienza possibile di un metodo di correzione degli errori in funzione del livello di rumore. La teoria non spiega come costruire un codice ottimo, ma stabilisce solo quali siano le prestazioni del codice ottimo. Quindi con tale teorema si dimostra la possibilità di raggiungere la capacità, ma non il modo di raggiungerla; da qui la nascita della teoria dei codici e la ricerca di tecniche in grado di raggiungere il limite definito dal teorema della capacità di canale.

La prima dimostrazione rigorosa del teorema si deve a Amiel Feinstein nel 1954. Il teorema stabilisce che, dato un canale con capacità C, su cui viene trasmessa informazione ad un tasso R, allora se $R < C$ esiste un codice che consente di rendere la probabilità di errore al ricevitore arbitrariamente piccola. Questo significa che, teoricamente, è possibile trasmettere informazione senza errori a qualunque tasso inferiore a C. Se invece $R > C$, allora non è possibile raggiungere una probabilità di errore piccola a piacere. Non è quindi possibile garantire che l'informazione sia trasmessa in maniera affidabile su un canale ad un tasso superiore alla capacità. Il teorema non considera il caso in cui R e C siano uguali.

1.4 Il canale trasmissivo con perdita

I canali trasmissivi con perdita a pacchetto hanno una grande importanza; se si vuole trasmette un flusso dati su Internet, allora il flusso dovrà essere pacchettizzato e ci sarà un'alta probabilità che dei pacchetti vadano persi e l'utente finale non riceverà il flusso in maniera integra.

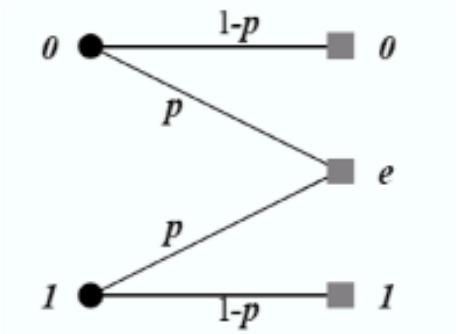


Figura 1.1: Canale con introduzione di errori

Un modello semplice di canale che possa descrivere tale situazione è un canale a cancellazione *q-ario*, che ha una probabilità di trasmettere un simbolo senza errori pari ad $1 - f$, e la probabilità di fornire in uscita un simbolo errato pari a f. La dimensione q dell'alfabeto è pari a 2^l con l pari al numero di bit che compongono il pacchetto. In figura 1.2 è mostrato il canale a cancellazione *q-ario*:

La probabilità condizionata è pari a:

$$p(y|x) = \begin{cases} 1 - f & y = x \\ f & y \neq x \end{cases} \tag{1.1}$$

dove $x, y \in \{1, ..., q\}$ e dove x è il simbolo di ingresso, mentre y è quello di uscita, $1, ..., q$ è l'alfabeto di simboli ed f è la probabilità di cancellazione del simbolo trasmesso.

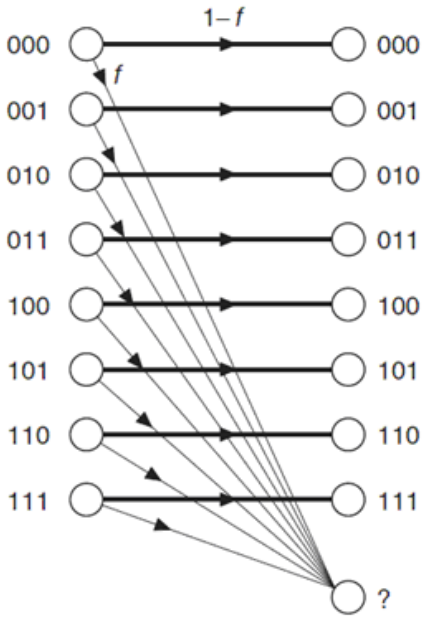


Figura 1.2: Canale a cancellazione *8-ario*

Le metodologie più comuni di comunicazione che vengono sfruttate per questi particolari canali sono quelle che utilizzano un canale di feedback tra il ricevitore e il mittente per controllare la ritrasmissione di pacchetti persi. Queste sono le tecniche di ARQ, dove il ricevitore richiede al trasmettitore la ritrasmissione dei pacchetti cancellati dal canale di comunicazione. Però, in caso di alta probabilità di cancellazione, il canale potrebbe saturare di fronte ad un numero elevato di richieste di ritrasmissione, oppure si potrebbero avere delle copie multiple di pacchetti già ricevuti.

Seguendo il teorema di Shannon per la codifica di canale, non è però necessario avere un canale di ritorno; una comunicazione affidabile dovrebbe essere possibile con un appropriato utilizzo di un codice FEC.

1.5 Tecniche *end to end* per il recupero delle perdite

Per poter recuperare le perdite in maniera *end to end*, si può utilizzare il protocollo ARQ oppure il Forward Error Correction a livello applicativo. Queste due tecniche hanno caratteristiche differenti e quindi sono più o meno adatte al sistema scelto in base alle sue esigenze. Infatti il protocollo ARQ garantisce completa affidabilità, ma è poco scalabile, soprattutto quando il numero di utenti finali è elevato; inoltre necessita del canale di ritorno. Il Forward Error Correction non necessita del canale di feedback ed è vantaggioso rispetto al protocollo ARQ in termini di efficienza di trasmissione e scalabilità. Basti infatti pensare, parlando di efficienza, che un pacchetto di recupero può essere utilizzato per recuperare la perdita di diversi pacchetti in diversi ricevitori. Infine, per quel che concerne la scalabilità, usando i pacchetti di recupero il trasmettitore necessita soltanto di una stima del caso peggiore per quel che riguarda le perdite e non di conoscere il numero esatto di pacchetti da ritrasmettere. Tuttavia l'uso del Forward Error Correction implica un overhead e quindi un'occupazione maggiore di banda.

1.5.1 La ritrasmissione

Come accennato in precedenza i protocolli di tipo ARQ necessitano di un canale di ritorno, che viene usato per controllare la ritrasmissione dei pacchetti cancellati. Per esempio, il ricevitore può mandare indietro messaggi di acknowledgement per ogni pacchetto ricevuto correttamente; il mittente ritrasmetterà i pacchetti per cui non abbia ricevuto l'ACK corrispondente.
Una tecnica nota nel caso di trasmissioni unicast è il Transmission Control Protocol (TCP), [2], in cui il *nodo target client* manda un messaggio di acknowl-

edgement per ogni pacchetto ricevuto correttamente. Un vantaggio di questo protocollo è che le ritrasmissioni successive correggono la perdita dei pacchetti. La regola seguita del protocollo è quella di ridurre il rate di trasmissione dei pacchetti finchè gli acknowledgement non confermino l'assenza di pacchetti persi; la tecnica usata è quindi adattativa. L'assunzione alla base di questo approccio è che la perdita dei pacchetti viene causata dalla congestione, che può essere diminuita riducendo il rate di trasmissione. Il grado delle ritrasmissioni scala con il grado della perdita dei pacchetti. Dal momento che il meccanismo di recupero dei pacchetti persi del TCP è posto insieme ad un meccanismo di controllo della congestione (riduzione drastica del rate di trasmissione in caso di perdita di pacchetti), questo approccio funziona abbastanza bene per la trasmissione di piccole quantità di dati sotto vincoli di tempo e di ritardo non stringenti.Si può pensare che questo meccanismo non sia appropriato per la distribuzione su IP di video in broadcast, in cui uno stream ha un alto rate, che non viene ridotto in maniera significativa dal controllo di congestione, deve rimanere relativamente costante per assicurare una riproduzione video continua ad alta qualità.

Il TCP definisce anche un'opzione per gli *acknowledgement* selettivi. Questa tecnica emula il controllo del rate, vantaggioso per i media stream, e riduce il traffico dei messaggi di *acknowledgement*; tale opzione non è pervasiva. I meccanismi di ritrasmissione TCP e quelli di controllo della congestione sono descritti in molti documenti, tra cui la specifica [3], che descrive un meccanismo di *Selective Acknowledgement*, la specifica [7], che propone un meccanismo per recuperare i segmenti persi in caso di finestra di congestione piccola o in caso di perdita di più segmenti in una singola finestra di trasmissione, e la specifica [8], che espone un algoritmo conservatore per l'opzione SACK.

Un'altra tecnica di ritrasmissione, applicabile alle trasmissioni multicast, è stata analizzata nel contesto del trasferimento affidabile in multicast di grandi quantità di dati (ovvero il trasferimento di file) all'interno dell'IETF RMT working group. Il NACK-Oriented Reliable Multicast Protocol (NORM) è un tecnica per la consegna affidabile di file su multicast basata su *acknowledgement* negativi, [6] [11]. A differenza degli approcci ARQ di tipo unicast come il TCP, il protocollo NORM è meno ridondante in quanto non richiede l'invio di messaggi di *acknowledgement* per ogni pacchetto. Sebbene anche il trasporto di stream multicast possa trarre benefici dalle tecniche con *acknowledgement* negativi, lo scenario che ha dato vita al lavoro dell'IETF RMT è quello del trasporto di file.

La ritrasmissione RTP è un'altra tecnica di recupero delle perdite dei pacchetti per applicazioni real-time. I pacchetti RTP ritrasmessi vengono inviati in uno stream separato dallo stream RTP originario. In modo analogo alle ritrasmissioni TCP, si assume che sia disponibile un feedback dai ricevitori ai mittenti, ma l'RTP/UDP non impone che il controllo di congestione riduca il rate di trasmissione dei pacchetti, rendendo questo meccanismo potenzialmente più appropriato per il broadcast video. L'RTCP non utilizza gli *acknowledgement* per ogni singolo pacchetto IP, ma riporta delle statistiche sulla perdita dei pacchetti che il nodo sorgente può valutare per determinare se l'adattamento è adeguato.
Un'estensione dell'RTCP per l'Audio-Visual Profile consente ai ricevitori di fornire feedback più immediati ai mittenti, permettendo di implementare meccanismi di recupero efficienti basati su feedback, come ad esempio la ritrasmissione. Più precisamente, le specifiche dell'IETF legate alla ritrasmissione RTP usano un semplice sistema in cui gli utenti richiedono la ritrasmissione dei pacchetti per-

si specifici inviando *acknowledgement* negativi (RTCP NACK) ad un *feedback target* o ad una sorgente di ritrasmissione su un flusso RTCP associato alla sessione RTP. Un ricevitore può usare un singolo pacchetto NACK per richiedere la ritrasmissione di uno o più pacchetti persi.

1.5.2 Il Forward Error Correction a livello applicativo

Un metodo per garantire l'affidabiltà è l'utilizzo di codici Forward Error Correction. Gli ingressi di un codificatore FEC sono k pacchetti sorgenti di uguale lunghezza. Il codificatore genera n pacchetti codificati della stessa lunghezza con una intestazione che contiene le informazioni necessarie al decoder. Il codice è sistematico se i pacchetti sorgente sono trasmessi insieme a quelli di riparazione. Dal lato del ricevitore il decodificatore utilizza i pacchetti ricevuti per ricostruire i pacchetti sorgente.
Tali tecniche introducono ridondanza, data dal rapporto n/k, ed essa è tanto più grande quanto più grande è il numero dei ricevitori.

Qundo uscì il libro di Shannon, l'unico modo in uso per realizzare la codifica FEC erano i codici a blocco. Con questa tecnica il codificatore inserisce bit di parità nella sequenza di dati usando un particolare algoritmo algebrico. A sua volta, il decodificatore applica l'algoritmo inverso per rivelare e correggere qualsiasi errore provocato dall'attraversamento del canale. In seguito vennero introdotti i codici convoluzionali, che non elaborano i bit in ingresso in blocchi, ma in stream. La codifica di ogni bit è quindi pesantemente influenzata dai bit che lo precedono, ovvero dalla ‘‘memoria" dei bit precedenti. Il decodificatore deve tener conto di questa memoria nel tentare di stimare la sequenza originale che può aver prodotto la sequenza ricevuta. Alcuni tipi di codici a blocco sono molto

efficaci per la correzione di errori a burst, mentre i codici convoluzionali sono in genere più robusti contro errori casuali. Si è quindi pensato di combinare le due tecniche, dando vita così a codici concatenati. In questo modo, eventuali burst di errori in uscita dal codificatore convoluzionale possono essere corretti dal decodificatore a blocco. Le prestazioni di un sistema di codifica sono ulteriormente migliorate dall'utilizzo di un interleaver, che disperde i burst di errori in modo da evitare che siano troppo lunghi per il decodificatore a blocco.

Nel 1993 sono stati sviluppati dei codici a correzione d'errore molto potenti, i Turbo codici. Essenzialmente, un Turbo-codificatore consiste in due codici, i cui ingressi sono connessi in parallelo mediante un interleaver. Quindi, la sequenza che alimenta il secondo codificatore è la stessa di quella che alimenta il primo a meno della permutazione introdotta dall'interleaver. Sebbene ognuno dei due codici componenti possa essere o a blocchi o convoluzionale, il codificatore globale può essere considerato a blocchi, poichè i dati vengono processati in blocchi. La caratteristica principale di questi codici è che permettono di ottenere probabilità di errore molto basse a rapporti segnale a rumore molto bassi. Di più recente introduzione sono i codici a fontana, [27]. Si consideri la trasmissione di informazioni da un mittente a più ricevitori e si ipotizzi che il canale tra il mittente e ogni ricevitore sia un canale a cancellazione, in cui un pacchetto o viene ricevuto senza errore o non viene ricevuto, come la rete Internet, con probabilità di cancellazione non nota. I codici a fontana permettono di ottenere una trasmissione con rate molto prossimo alla capacità del canale su tutti i canali di trasmissione, simultaneamente. Questi codici hanno la caratteristica di essere rateless, ovvero possono generare, potenzialmente, una sequenza illimitata di simboli di codifica da un dato insieme di simboli sorgente. Così facendo i simboli sorgente originali possono essere recuperati con alta probabilità da qualsiasi sottoinsieme di simboli

codificati, di dimensione leggermente superiore al numero dei simboli sorgente. Il numero di simboli codificati generati può essere variato dinamicamente.

1.6 Error Concealment

L'error concealment è una tecnica molto utilizzata per ottenere una qualità migliore delle sequenze video, [39]. Tuttavia, a differenza delle tecniche di ritrasmissione e di Forward Error Correction, l'error concealment è una tecnica di post-processing effettuata a lato destinatario, sul decoder. Lo scopo finale di tale tecnica non è quello di recuperare le perdite bensì di individuare gli errori e scartare l'informazione corrotta; questa tecnica viene molto utilizzata nei sistemi a basso tasso di perdita. Il vantaggio principale è quello di non utilizzare informazioni addizionali, ma aggiunge complessità computazionale al decoder. Tecniche di error concealment possono essere:

- *Motion Compensated Temporal Prediction*, che è un approccio semplice, ma efficace, che per recuperare un macroblocco danneggiato copia il corrispondente macroblocco nel frame decodificato precedentemente;

- *Interpolazione spaziale*, che è un approccio che interpola i pixel in un blocco danneggiato da pixel in blocchi adiacenti ricevuti correttamente. Di solito, poichè tutti i blocchi o macroblocchi nella stessa riga sono messi nello stesso pacchetto, i soli blocchi vicini disponibili sono quelli sopra e sotto. Poichè la maggior parte dei pixel in questi blocchi sono troppo lontani, di solito solo i pixel di confine sono usati per linterpolazione. Invece di interpolare i pixel individuali, un approccio più semplice è stimare i coefficienti DC, cioè il valore medio, di un blocco danneggiato e sostituire il blocco danneggiato

tramite una costante uguale al valore DC stimato. Tale valore DC può essere stimato mediando i valori DC dei blocchi circostanti.

1.7 Tecniche a livello rete per la QoS

Una prima tecnica, valida per tutte le piattaforme, consiste nella locazione geograficamente distribuita e prossima ai bacini d'utenza dei server dei contenuti, questo minimizza il traffico in rete e contribuisce significativamente al raggiungimento degli obiettivi di qualità prefissati. Nelle reti gestite, inoltre, risulta possibile prevedere opportuni meccanismi di gestione delle risorse e pre-scegliere delle configurazioni architetturali del servizio. I meccanismi in questione consentono di amministrare le risorse di trasporto disponibili, facendo sì che i pacchetti del servizio a cui deve essere garantita una determinata QoS ottengano un trattamento privilegiato rispetto agli altri; questo metodo ricade sotto la terminologia molto generale di ‘‘prioritizzazione del traffico”. Ne consegue che per fornire *priorità* ad un determinato traffico, i relativi pacchetti devono essere opportunamente identificati. La gestione del traffico con priorità associato ai diversi servizi fa riferimento a due possibili approcci:

- *Integrated Services*, che prevede la preventiva prenotazione delle risorse di rete richieste dal servizio;
- *Differentiated Services*, che prevede, all'interno della rete, il trattamento differenziato del traffico associato a servizi differenti.

Questa seconda metodologia è spesso utilizzata congiuntamente all'impiego del protocollo MPLS che consente l'allocazione garantita, lungo il percorso di rete, delle risorse trasmissive necessarie per il soddisfacimento dei requisiti di QoS del servizio. Il DVB e l'ATIS prevedono esplicitamente l'uso della tecnica

DiffServ quale tecnica di rete su cui basare la QoS di una rete managed per servizi di diffusione televisiva. L'OIPF, pur facendo riferimento ad architetture di rete tradizionali (non NGN) prevede, nell'architettura funzionale relativa a piattaforme managed, uno specifico blocco RAC che, interagendo con altri blocchi architetturali, tipicamente le funzioni di trasporto, si occupa della gestione della QoS. Tuttavia, come nel caso di architetture su NGN, non sono specificate le tecniche di rete utilizzate allo scopo.

Per quanto riguarda invece le architetture basate su NGN, si deve tenere presente che la tecnologia di trasporto prevista per tale tipo di reti è di tipo IP con gestione della QoS. Tuttavia le normative, che fanno riferimento a tale infrastruttura di rete per il supporto di servizi televisivi, fanno generalmente riferimento solamente alle funzionalità di trasferimento ad alto livello offerte dall'NGN, senza specificare le modalità di gestione della QoS nello strato di trasporto demandata alle relative specifiche.

2

Valutazione dell'efficacia di un sistema di protezione per la QoS

Al fine di valutare se sia effettivamente possibile ottenere un servizio televisivo trasmesso su IP con alta qualità ed affidabilità, sono state valutate le prestazioni di un tecnica di correzione a livello applicativo standardizzata dall'SMPTE per vari eventi di perdita. Successivamente sono state messe a paragone le prestazioni ottenute con le prestazioni di tecniche di rete, che sono però fruibili solamente in una rete di tipo gestito, mentre la tecnica di protezione analizzata, essendo di tipo end-to-end, può esser utilizzata anche per reti non gestite. In questo capitolo verranno riportati i test bed sperimentali che sono stati creati e i risultati che sono stati ottenuti.

2.1 Prestazioni di una tecnica di correzione d'errore a livello applicativo

Per poter ottenere una determinata qualità del servizio su reti con perdita, vengono studiate varie tecniche di codifica di canale e ne viene valuta la capacità del recupero delle perdite. Nella sperimentazione che viene qui esposta è stata valutata la capacità di correzione di una tecnica standard, facente parte dei codici Application Layer - Forward Error Correction, sistematica, a basso costo computazionale, nominata SMPTE 2022-1, [14]. Al fine di effettuare tale analisi sono stati considerati gli eventi di perdita più comuni su una rete quale è l'Open Internet, [30].

2.1.1 Tecnica di correzione d'errore a livello applicativo considerata

Vista la struttura della rete di trasporto, nelle architetture dei servizi di diffusione televisiva su IP, i programmi sono usualmente gestiti e trasferiti come unità singole e non affasciati in multiplex come avviene nelle reti broadcast terrestri e satellitari. Il Single Program Transport Stream (SPTS) MPEG-2 è generalmente il formato di trasporto utilizzato per sincronizzare gli stream elementari (video, audio, dati) componenti il servizio e per inserire le opportune Service Information. Gli stream SPTS sono trasferiti in rete tramite incapsulamento nello stack di protocolli RTP/UDP/IP secondo regole standardizzate da IETF, RFC 2733 [5]. In particolare l'uso del protocollo RTP sopra la pila UDP/IP consente di assegnare una numerazione sequenziale ai pacchetti, circostanza questa che può essere sfruttata dal ricevitore per ristabilire il corretto ordine dei pacchetti, che non risulta garantito dal protocollo di trasporto UDP, [9] [10]. I pacchetti RTP

sono dunque le unità dati maggiormente utilizzate per il trasferimento dalla sorgente a destinazione. Conseguentemente queste sono le unità fondamentali che devono essere protette dalle perdite nell'attraversamento della rete, per esempio con l'applicazione di opportune tecniche di Forwad Error Correction.
Un importante vantaggio di questo schema è che si può usare con qualsiasi trasporto standardizzato, fin tanto che sia incapsulato in un pacchetto RTP. L'uso di RTP è necessario, in quanto fornisce un header standard per i pacchetti.

La tecnica AL-FEC considerata è la tecnica definita dallo standard SMPTE 2022-1. Tale tecnica prevede che a partire da un insieme di pacchetti RTP contenenti dati audio/video nel loro payload (*media packets*) sia generato un corrispondente pacchetto di protezione FEC, anch'esso di tipo RTP. Il payload di tale pacchetto contiene dell'informazione ridondante per il recupero, in ricezione, di eventuali perdite tra i *media packet* ad esso corrispondenti. Lo schema FEC introdotto dall'SMPTE si colloca a livello applicativo e quindi opera in maniera end-to-end (Application Layer - Forward Error Correction).
L'SMPTE 2022-1 è un'estensione di una tecnica ad una dimensione basata sullo XOR, che applica la protezione solo a pacchetti consecutivi, come è possibile vedere in 2.1. Se però viene utilizzato uno schema monodimensionale, due o più pacchetti persi consecutivi o comunque facenti parte dello stesso gruppo a cui fa riferimento un pacchetto FEC, non possono essere recuperati.

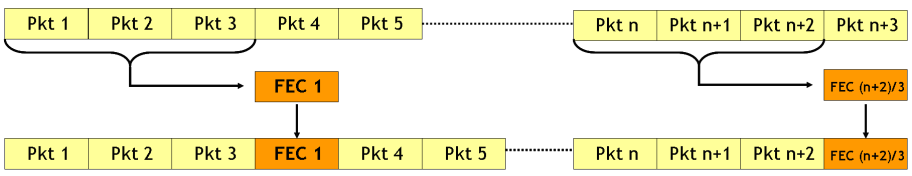


Figura 2.1: Schema di codifica monodimensionale.

La tecnica è stata quindi estesa, per recuperare anche le perdite di pacchetti consecutivi e quindi a burst; ciascun pacchetto FEC è associato a media packet periodicamente selezionati e, quindi, pacchetti RTP consecutivi persi possono essere recuperati da pacchetti FEC consecutivi. La funzione che viene utilizzata per correggere gli errori è l'operazione XOR, che ha la capacità di recuperare ciascun singolo pacchetto perso. Per comprendere meglio il funzionamento di questo schema AL-FEC, i pacchetti vengono organizzati in una matrice con L colonne e D righe, 2.2.

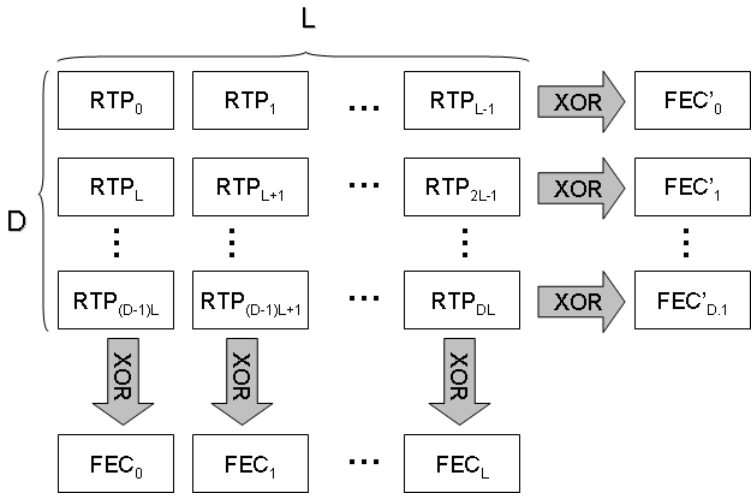


Figura 2.2: Generazione dei pacchetti FEC righe e FEC colonne secondo SMPTE 2022 in modalità 2D.

L'applicazione della tecnica FEC SMPTE 2022-1 ad un singolo flusso informativo consente di generare due distinti flussi di pacchetti di protezione, il primo composto da pacchetti FEC calcolati sulle righe dello schema di generazione riportato in 2.2, il secondo composto da pacchetti FEC calcolati sulle colonne del predetto schema. Dalla figura si evince che il primo flusso, ''FEC righe'', si riferisce all'applicazione del codice FEC a pacchetti RTP audio/video consecutivi mentre il secondo flusso ''FEC colonne" si riferisce all'applicazione del codice FEC a pacchetti RTP audio/video non consecutivi, risultando utile per il recupero di

perdite a burst. I pacchetti etichettati con **FEC** sono il primo stream di pacchetti FEC e i pacchetti etichettati con **FEC'** sono il secondo stream di pacchetti FEC. Il secondo stream FEC è in grado di far fronte a perdite di pacchetto singole e il primo stream FEC è in grado di far fronte a perdite a burst di lunghezza fino a L.
La scelta se generare, per uno stesso flusso di pacchetti dati, entrambi i flussi FEC oppure uno soltanto e la scelta delle dimensioni L e D è operata come compromesso tra la capacità di recupero delle perdite e l'overhead di protezione. La scelta della configurazione FEC ottimale è in ogni caso strettamente correlata alla tipologia delle perdite in rete in quanto differenti configurazioni, presentando caratteristiche strutturali differenti, si prestano a contrastare più efficacemente particolari tipologie di perdite piuttosto che altre.

Lo standard definisce quindi due livelli di protezione, il livello A e il livello B. I dispositivi di livello A devono sostenere uno stream FEC, mentre i dispositivi di livello B devono sostenere contemporaneamente due stream FEC simultanei. Questi due flussi devono essere portati su porte UDP separate, per consentire loro di avere un numero di trattamento di sequenza separato e per mantenere una retrocompatibiltà con le implementazioni che supportano un unico flusso di FEC o che non lo supportano affatto.

Per facilitare l'interoperabilità e la semplificazione dell'implementazione vengono poste delle limitazioni ai valori dei parametri D e L. Al minimo, i mittenti e i destinatari supporteranno tutte le combinazioni di valori di D e di L che soddisfano i seguenti limiti:

$$\begin{aligned} L \times D &\leq 100 \\ 1 \leq L &\leq 20 \\ 4 \leq D &\leq 20 \end{aligned} \tag{2.1}$$

Nel caso in cui $L \geq 4$ un dispositivo dovrà supportare due stream FEC.
Il limite di questo schema FEC sta nell'impossibilità di correggere errori che si presentano in forma matriciale. In questo caso infatti ogni FEC in questione ha due pacchetti persi sulla propria riga o colonna e quindi non è in grado di recuperare le perdite.

Da un punto di vista sintattico i pacchetti FEC sono pacchetti RTP il cui payload è composto a sua volta da un campo ‘‘FEC header" e un campo ‘‘FEC payload". Il campo ‘‘FEC header" comprende le informazioni per associarlo ai pacchetti RTP ai quali si riferisce, il campo ‘‘FEC payload" il risultato dell'operazione XOR sui bit dei relativi pacchetti RTP, [15].
I trasmettitori che adottano tale tecnica usualmente inviano i pacchetti FEC in maniera interallacciata ai pacchetti informativi, in maniera tale da evitare brusche variazioni nel bit-rate trasmesso, ma senza incrementare significativamente nello stesso tempo la latenza del sistema.

2.1.2 Test-bed sperimentali creati

La valutazione delle prestazioni dell'AL-FEC SMPTE 2022-1 in caso di perdite random e di perdite causate dal Repetitive Electrical Impulse Noise (REIN) è stata realizzata utilizzando sia un simulatore di rete IP proprietario, sia un simulatore di rete IP open source, Netem. La catena sperimentale utilizzata è mostrata in 2.3.

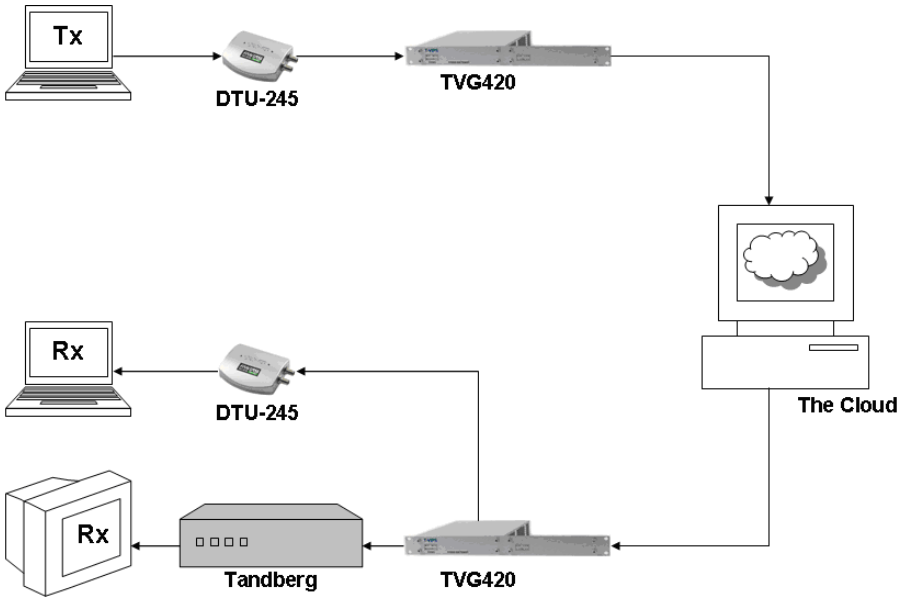


Figura 2.3: Schema di rete con il simulatore di rete utilizzato per le simulazioni.

Il TVG420 di trasmissione viene alimentato da un Transport Stream MPEG-2 Constant Bit Rate pre-registrato e genera il flusso o i flussi di pacchetti IP di protezione. I flussi dati e i flussi di pacchetti di protezione transitano quindi nel simulatore di rete che permette di introdurre delle perdite sul flusso entrante, simulando così i possibili comportamenti di una rete IP reale; si potuto stabilire il tipo di perdite da introdurre e la probabilità di occorrenza (Packet Loss Rate). Il flusso uscente dal simulatore di rete passa quindi nel TVG-420 di ricezione che si occupa di recuperare le perdite introdotte e fornisce in uscita un Transport Stream MPEG-2 ricostruito. A destinazione il decodificatore di TS MPEG-2 Tandberg e il monitor a cui è collegato consentono di avere un riscontro visivo dell'entità delle perdite.

Per valutare le prestazioni dell'AL-FEC SMPTE 2022-1 in caso di perdite dovute o al riempimento delle code di un router, o alla presenza di flussi Variable Bit Rate o che avvengono nel caso di link failure si è fatto ricorso al ‘‘Test Bed

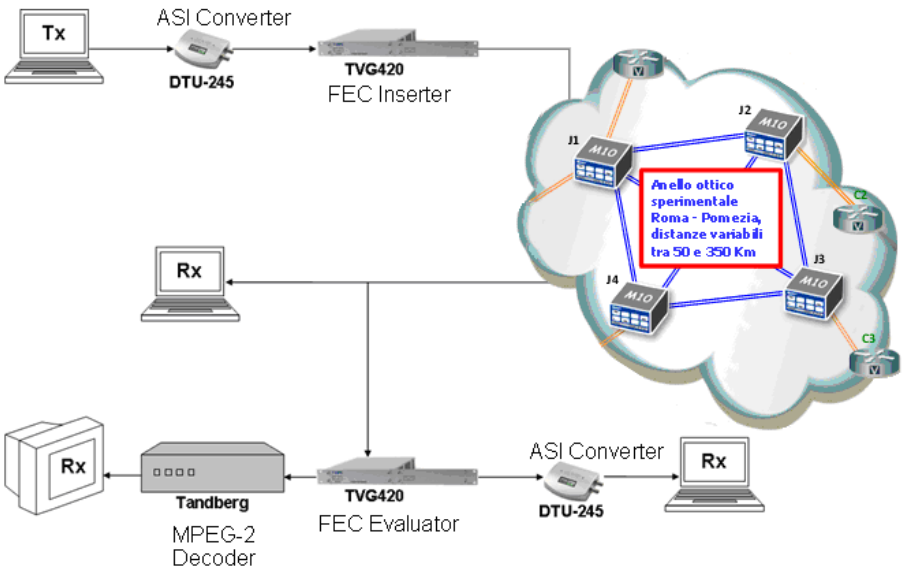


Figura 2.4: Schema di rete con il Test Bed utilizzato per le simulazioni.

di rete IP'' messo disposizione dall'Istituto Superiore delle Comunicazioni e delle Tecnologie dell'Informazione del Dipartimento Comunicazioni del Ministero dello Sviluppo Economico. La catena sperimentale utilizzata è rappresentata in 2.4, dove la rete è costituita dall'anello ottico sperimentale Roma-Pomezia.

2.1.3 Risultati sperimentali

Per testare le prestazioni dell'AL-FEC SMPTE 2022-1 sono state considerate varie tipologie di perdite e varie situazioni che possono avvenire in rete e che comportano la perdita di pacchetti.

2.1.3.1 Perdite random

Inizialmente sono state considerate perdite di tipo random i.i.d. con Packet Loss Rate crescente. I flussi che sono stati esaminati per tale tipologia di perdite sono un flusso a 2Mb/s e un flusso a 20Mb/s. Per il flusso a 2Mb/s sono stati rispettati i vincoli di qualità di una trasmissione Standard Definition, ovvero un

artefatto ogni ora, mentre per il flusso a 20Mb/s sono stati rispettati i vincoli di qualità di una trasmissione High Definition e quindi di un artefatto ogni quattro ore. I vincoli citati sono stati ripresi dalle raccomandazioni del Broadband Forum. Inoltre, i grafici che seguono sono stati ricavati considerando una latenza massima di 400ms, come specificato dal DVB.

è stato quindi analizzato per primo il flusso video a 2Mb/s, per il quale si è ricavato il grafico rappresentato in 2.5. Si tenga presente che l'overhead minimo di un FEC solo colonne è 5% e l'overhead massimo è 25%, mentre l'overhead minimo per un FEC righe e colonne è 20% e l'overhead massimo è 50%. Tuttavia nel caso di un flusso a bit-rate pari a 2Mb/s, come si può vedere dalla 2.5, l'overhead minimo che è stato utilizzato per un FEC solo colonne è il 14.29%, anzichè il 5%. è stata fatta questa scelta in quanto è necessario rispettare il vincolo di latenza di 400ms; infatti un FEC (5,20) con overhead 5% necessita di una latenza di circa 900ms per poter memorizzare due matrici e consentire quindi le correzioni necessarie. Questo non dovuto al costo computazionale della tecnica di correzione scelta, ma al tempo che impiega un flusso di 2Mb/s a trasmettere 200 pacchetti. Per tale overhead la latenza diminuisce se si considerano FEC con $L < 5$; per esempio un FEC (1,20) necessita di una latenza di 100ms. Tuttavia, secondo teoria, un FEC (1,20) non sarebbe in grado di correggere due errori qualora questi si presentassero all'interno di una sequenza di 20 pacchetti. Quindi dalle prove fatte si è notato che il FEC che permette di rimanere entro i vincoli posti dal DVB presenta un overhead minimo del 14.29%.

Sempre dalla 2.5 si può ricavare che il FEC solo colonne riesce a correggere perdite random fino ad una probabilità di perdita dei pacchetti di $8 \cdot 10^{-4}$, con un overhead del 25%. Dal grafico, inoltre, si deduce che il passaggio dal FEC solo colonne al FEC righe e colonne porta vantaggi dal punto di vista dell'overhead

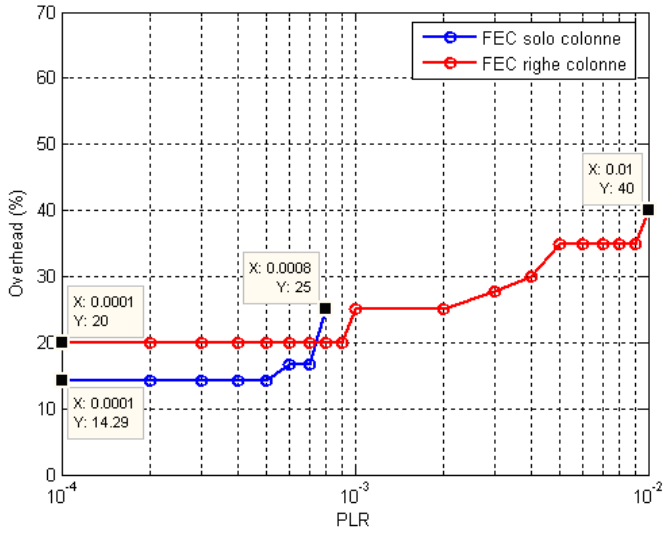


Figura 2.5: Overhead minimo richiesto in caso di perdite random i.i.d.: flusso a 2Mb/s.

utilizzato per probabilità che vanno da $7 \cdot 10^{-4}$ a $9 \cdot 10^{-4}$, mentre per overhead maggiori consente di contrastare probabilità di perdita di pacchetti fino a 10^{-2}. Per tale probabilità, è necessario utilizzare un overhead del 40%. Con un overhead del 50% non si riescono a correggere perdite maggiori, rispettando il vincolo di un massimo di un artefatto visibile ogni ora.

Anche per il FEC righe e colonne una latenza imposta non sufficiente potrebbe incidere sulla capacità di correzione del FEC. Tuttavia, le perdite isolate con una probabilità non troppo elevata possono essere corrette dal solo FEC riga e risulta quindi non incidente, in questo caso, la capacità di bufferizzazione di una sola matrice.

Di seguito è stato analizzato il flusso video a 20Mb/s, per il quale è stato ricavato il grafico rappresentato in 2.6. Per quanto riguarda il flusso a 20Mb/s, come mostrato in 2.6, è possibile utilizzare un FEC solo colonne con overhead minimo, in quanto il vincolo di latenza di 400ms viene sempre rispettato, poichè

il tempo necessario per memorizzare due matrici è in ogni caso inferiore ai 100ms, quindi è possibile effettuare tutte le correzioni possibili. Dal grafico si può vedere come le capacità di correzione del FEC solo colonne si esauriscano per un PLR di $4 \cdot 10^{-4}$, che può essere corretto con un overhead del 16.67%. Inoltre si può vedere come con un FEC righe e colonne si riesca ad avere una capacità di correzione maggiore, a patto di utilizzare overhead più elevati, occupando di conseguenza una percentuale di banda maggiore. Si noti che con un overhead del 20%, è possibile recuperare perdite con un intervallo di PLR che va da $5 \cdot 10^{-4}$ a 10^{-3}. Con un overhead del 40% è possibile recuperare perdite con PLR pari a $8 \cdot 10^{-3}$, aumentando l'overhead non è possibile recuperare perdite con PLR maggiore.

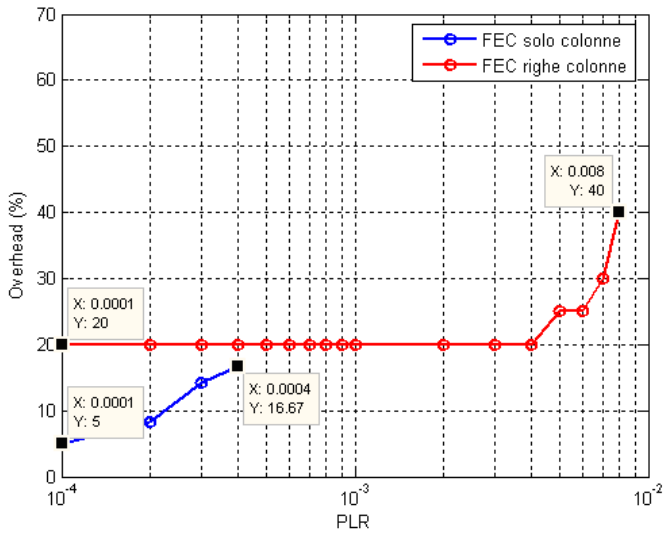


Figura 2.6: Overhead minimo richiesto in caso di perdite random i.i.d.: flusso a 20Mb/s.

2.1.3.2 Perdite causate dal Repetitive Electrical Impulse Noise

Con la medesima catena sperimentale utilizzata precedentemente, sono state effettuate prove per valutare la capacità di correzione considerando il caso di perdite causate dal REIN, tipiche del DSL, che corrispondono ad una perdita di circa

8ms. Il grafico in 2.7 è stato ottenuto per un flusso a 2Mb/s, considerando una latenza di 400ms. Come è possibile osservare dalla 2.7, le capacità di correzione del FEC solo colonne non subiscono forti degradamenti, infatti con l'overhead massimo del 25% si riesce a correggere anche perdite REIN con una probabilità di $8 \cdot 10^{-4}$. Peggiorano notevolmente, invece, le prestazioni del codice FEC righe e colonne, che con l'overhead massimo del 50% permette di correggere una probabilità di perdite REIN di $3 \cdot 10^{-3}$ e non oltre, mentre per perdite isolate si arriva a probabilità del 10^{-2}. Questo risultato segue la teoria, in quanto il FEC calcolato sulle colonne permette di recuperare le perdite di tipo burst, mentre il FEC a protezione delle righe è maggiormente utile nel caso di perdite isolate.

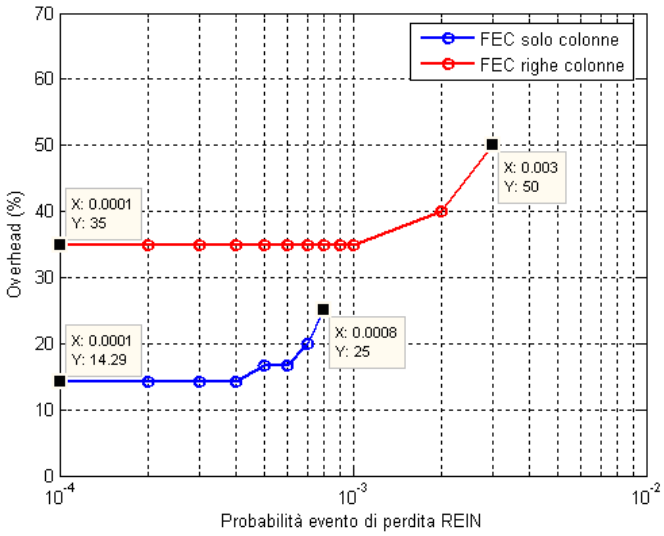


Figura 2.7: Overhead minimo richiesto in caso di perdite REIN: flusso a 2Mb/s.

Nel caso di perdite REIN vi è la necessità di bufferizzare due matrici, in quanto si potrebbe avere bisogno sia del FEC riga che del FEC colonna. Quindi in questo caso se non si ha una latenza sufficiente, non si è in grado di correggere errori consecutivi. Così facendo non è possibile utilizzare l'overhead minimo del 20% senza uscire fuori dalle specifiche di errore del Broadband Forum; con un overhead del 35% si rientra nei limiti di latenza e di conseguenza è possibile re-

cuperare le perdite REIN.

Se si considera il flusso a 20Mb/s non è possibile recuperare perdite di 8ms con le probabilità d'errore considerate per il flusso a 2Mb/s. Tuttavia qualora le perdite a burst fossero con probabilità minore, quindi al di sotto di 10^{-4}, allora il FEC esaminato sarebbe in grado di contrastare anche questo tipo di perdite. Infatti è possibile recuperare perdite di un flusso video a 20Mb/s di durata fino a 10ms, utilizzando un FEC (20,5) solo colonne con un overhead del 20%, qualora queste perdite fossero isolate.

2.1.3.3 Valutazione del caso di eventi di congestione

Altra analisi fatta è stata la valutazione della quantità di pacchetti IP che vengono persi sul flusso dati analizzato, nel momento in cui si saturano le capacità del link collegato all'interfaccia del router attraverso cui passa il flusso dati. Gli esperimenti sono stati eseguiti per un intervallo di tempo di un minuto, utilizzando un flusso con bit-rate di circa $46,8$Mb/s. Così facendo in un minuto si hanno all'incirca 258850 pacchetti IP. Il router considerato riesce a gestire un traffico totale in ingresso sul link considerato di circa 975Mb/s (link fisico a 1Gb/s); insieme al flusso dati sul link viene fatto passare un traffico di background, generato con un bit-rate differente per ogni test (in un intervallo di tempo di un minuto), partendo da 997Mb/s ed arrivando a 928Mb/s. Il flusso video considerato è un flusso codificato MPEG-2 mentre il traffico di background viene iniettato sullo stesso link da un generatore di traffico Smartbits. Si è quindi aggiunto il flusso di protezione FEC SMPTE 2022-1, al fine di valutare quanto incida la scelta di proteggere i dati in termini di occupazione di banda e quindi di numero di flussi contemporanei che è possibile gestire. I dati vengono riportati nella 2.1.

Traffico back-ground (Mb/s)	Flusso video a 46.8 Mb/s con aggiunta del FEC	Bit-rate totale (Mb/s)	N flussi totali	Pacchetti persi sul flusso video (%)	Pacchetti recuperati dai pacchetti persi (%)	Pacchetti non recuperati dai pacchetti persi (%)
FEC (20,5) righe e colonne (25% overhead)						
942	58.55	1000.55	17.1	3.64	63.95	36.05
937	58.55	995.55	17	0.92	100	0
937	58.55	995.55	17	0.89	100	0
FEC (20,5) righe e colonne (20% overhead)						
940	56.2	996.2	17.73	0.32	83.48	16.52
940	56.2	996.2	17.73	0.28	83.91	16.09
938	56.2	994.2	17.7	0.02	100	0
FEC (5,20) solo colonne (5% overhead)						
932	49.2	981.2	19.95	0.18	70.15	20.85
931	49.2	981.2	19.9	0.02	100	0
FEC (4,4) solo colonne (25% overhead)						
937	58.55	995.55	17	0.07	100	0

Tabella 2.1: Percentuali di pacchetti persi sul flusso dati con l'uso di FEC.

Per ogni configurazione di AL-FEC, la percentuale di overhead è differente e quindi il numero di flussi che possono essere trasmessi sul collegamento è differente. In linea teorica, se cresce l'overhead, allora cresce la capacità di correzione, ma contemporaneamente diminuisce il numero di flussi video concorrenti che possono essere inviati.
Come si può vedere dalla 2.1 un codice FEC con un overhead minimo del 5% riduce di un flusso il numero totale di flussi contemporanei, ma permette di recuperare eventuali perdite dovute ad errori della rete che in assenza di FEC non potrebbero essere corretti.

2.1.3.4 Valutazione delle prestazioni del FEC in presenza di flussi Variable Bit Rate

Lo scopo della sperimentazione successiva è stata la valutazione della robustezza del FEC SMPTE 2022-1 in condizioni di traffico simili a quelle della rete Internet, dove i flussi sono a bit-rate variabili e non Costant Bit Rate, mentre il flusso che viene protetto dalla tecnica di FEC considerata è a bit-rate costante. Nella prova seguente è stato creato, tramite un primo generatore di traffico (SmartBits), un traffico di background, al quale è stato aggiunto un traffico variabile, generato tramite un secondo generatore (Anritsu). Questo traffico variabile è stato realizzato inserendo ogni 5 secondi un burst istantaneo variabile tra 0 e 316 Mb/s. Per ogni valore di traffico di background la prova ha avuto una durata di un minuto. Il flusso video sul quale sono state valutate le perdite e i pacchetti recuperati è un flusso CBR a 20Mb/s. I risultati vengono riportati nella 2.2.

Si nota come in caso di variazioni di breve durata il FEC riesca a correggere gli errori. Inoltre si può notare che l'uso del FEC (20,5) righe e colonne porta ad essere maggiormente robusti nei confronti di perdite di tipo burst. D'altro canto il FEC (4,4) solo colonne non è il più indicato per contrastare perdite di questo

Bit-rate base (Mb/s)	Bit-rate max raggiunto (Mb/s)	Tipo di FEC	Percentuale di pacchetti persi sul flusso video	Percentuale di pacchetti recuperati dai pacchetti persi	Percentuale di pacchetti non recuperati dai pacchetti persi
684	1025	(20,5) righe e colonne	0.12%	100%	0%
733	1074	(20,5) righe e colonne	0.22%	97.41%	2.59%
781	1122	(20,5) righe e colonne	0.58%	96.24%	3.76%
830	1171	(20,5) righe e colonne	1.34%	89.32%	10.68%
684	1025	(4,4) solo colonne	0.07%	100%	0%
733	1074	(4,4) solo colonne	0.17%	91.49%	8.51%
781	1122	(4,4) solo colonne	0.68%	85.5%	14.5%
830	1171	(4,4) solo colonne	1.04%	76.61%	23.39%

Tabella 2.2: Perdita di pacchetti su un flusso a bit-rate costante in caso di traffico a bit-rate variabile.

tipo. Si noti che il FEC (20,5) righe e colonne e il FEC (4,4) solo colonne hanno entrambi un overhead del 25%.

2.1.3.5 Capacità di recupero del FEC SMPTE 2022-1 in caso di link failure

è stata di seguito indagata la capacità di correzione della tecnica SMPTE 2022-1 in presenza di un'interruzione del collegamento tra due router. Per realizzare questa situazione è stato fatto passare il flusso dati di interesse in un link del test bed, nel quale era stato preventivamente inserito uno switch gestibile da remoto. Attraverso le istruzioni inviate allo switch tramite linee di comando è stato possibile simulare un link failure, costringendo i router ad un ricalcolo del percorso. Il ripristino in OSPF, [4] [16], ha un tempo medio di 170ms, che viene ridotto se fatto in MPLS mediante i meccanismi di Link Protection e Standby Secondary Path (MPLS LP+SSP) arrivando ad un tempo medio di circa 50ms.
In caso di OSPF non si riesce a recuperare la perdita, mentre nel caso in cui si utilizzi la tecnica MPLS è possibile recuperare la perdita qualora si utilizzasse un flusso di 2Mb/s. Per un flusso a 20Mb/s non è in ogni caso possibile recuperare la perdita provocata dall'interruzione del link, perchè in un tempo di ripristino pari a 50ms vengono trasmessi troppi pacchetti dati. Ciò pone l'accento sul fatto che per servizi ad alta qualità le microinterruzioni da link failure rappresentano un problema non superabile con il solo FEC SMPTE 2022-1. La 2.8 illustra i valori minimi del numero delle colonne per far sì che sia possibile recuperare un'interruzione di 50 ms. Si tenga presente che con un valore minimo di righe, allora l'overhead dei FEC solo colonne considerati è fissato a 25% per tutti i flussi. Per flussi superiori a 4Mb/s, non è più possibile recuperare la perdita utilizzando il codice SMPTE 2022-1.

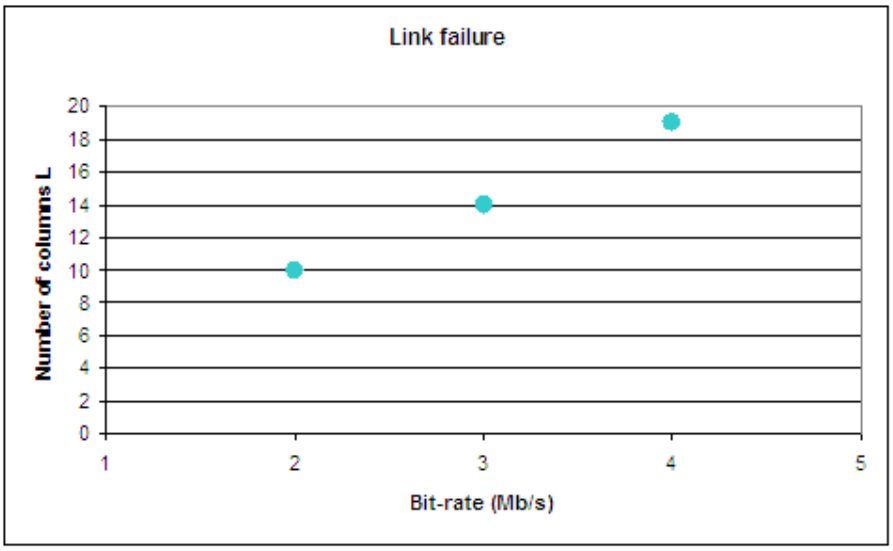


Figura 2.8: Valori minimi del numero di colonne per recuperare un link failure di 50 ms.

Per un flusso a 2Mb/s, il recupero dei pacchetti persi a causa dell'interruzione del link, se si considerano non più di 400ms di latenza, viene effettuato utilizzando al minimo un FEC (10,4) solo colonne con un overhead del 25%. Si noti che, qualora si potesse aumentare la latenza richiesta, allora la perdita sarebbe recuperabile anche con altri FEC con overhead minori. Ad esempio, per una latenza di 600ms, l'interruzione sarebbe recuperabile con un FEC (10,6) solo colonne, che presenta un overhead del 16.67%.

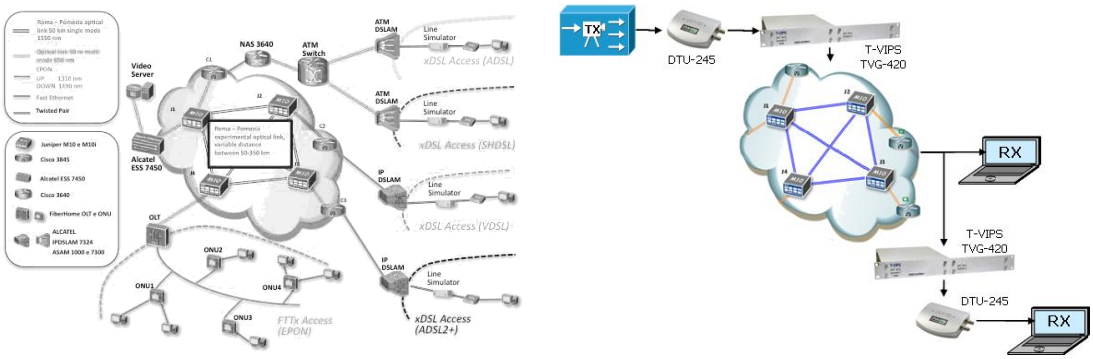
2.2 Confronto tra le prestazioni di tecniche di rete e tecniche end-to-end

Come già detto in precedenza, quando si lavora sui servizi televisivi, e in generale su flussi audio/video, è essenziale considerare aspetti di Qualità del Servizio relativi ai flussi video che verrano trasportati su una rete a pacchetto come quella IP. In particolare, è possibile distinguere due differenti scenari. Nel primo, un opera-

tore televisivo (o più genericamente un Service Provider) sottoscrive un contratto (Service Level Agreement) con il Network Operator, che può attuare dei meccanismi per il controllo della QoS. Si parla in questo caso di rete ‘‘Managed”. Nel secondo scenario, non è previsto alcun accordo tra chi fornisce il contenuto e chi lo trasporta. In questo caso, gli operatori televisivo non sono a conoscenza delle caratteristiche della rete e devono adottare meccanismi per il ‘‘recupero” della qualità. Si parla in questo caso di rete ‘‘Unmanaged”.
Lo scopo di questa sperimentazione è analizzare e paragonare i due differenti scenari. Le prove sono state effettuate trasportando servizi audio/video, come MPEG-2 TS [17], sul protocollo RTP mediante una rete ottica GbE ad estensione geografica, [35].

2.2.1 Test-bed sperimentale generato

Il test bed sperimentale 2.9a utilizzato è composto da una sezione di core e una di edge/accesso. La sezione di core è composta da quattro router Juniper M10 completamente magliati mediante l'anello ottico sperimentale con link (1 Gbit/s) in fibra monomodale. La sezione di edge è composta da tre router Cisco 3845, quella di accesso prevede l'utilizzo di tecniche xDSL e EPON. Inoltre è stato utilizzato lo Smartbits 6000 per generare traffico e gli analizzatori di traffico Anritsu MD1230B e Wireshark per la cattura dei pacchetti.
In 2.9ab è riportata la catena utilizzata per la valutazione delle prestazioni del FEC SMPTE 2022-1. Per la generazione dei pacchetti di correzione, secondo lo standard SMPTE, sono stati utilizzati i T-VIPS TVG-420.



(a) (b)

Figura 2.9: (a) Test-bed sperimentale. - (b) Catena per valutazione dell'AL-FEC.

2.2.2 Risultati sperimentali

In questa sezione, viene valutato l'impatto sui flussi video delle tecniche proposte nei due scenari considerati.

2.2.2.1 Gestione della QoS: VPLS vs FEC

Lo scopo di questa prova è valutare l'impatto sui flussi video di una congestione di rete sia in uno scenario di rete Managed che Unmanaged. Per il primo scenario, vengono comparati il caso in cui viene applicato il controllo della QoS per mezzo del VPLS, [19] e il caso in cui non vi è alcuna gestione della QoS. Per il secondo scenario, viene utilizzata la tecnica SMPTE 2022-1 ed effettuata una comparazione tra diverse configurazioni di AL-FEC.

Considerando una rete managed si è provveduto alla configurazione del VPLS definendo due Classi di Servizio (CoSs):

- ‘‘high priority’’ CoS;
- ‘‘low priority’’ CoS.

Traffico di background (Mb/s)	Bit-rate totale (Mb/s)	N flussi totali	Percentuale pacchetti persi sul flusso video
997	1043.8	22.3	5.4%
997	1023.8	21.876	2.04%
957	1003.8	21.45	3.18%
957	1003.8	21.45	2.6%
937	983.8	21.02	0.43%
937	983.8	21.02	0.28%
932	978.8	20.91	0.017%
932	978.8	20.91	0.013%
931.7	978.5	20.9	0.00155%
930.5	977.3	20.88	0
928	974.8	20.83	0

Tabella 2.3: Percentuali di pacchetti persi sul flusso video senza l'uso di FEC.

Per mezzo del VPLS è possibile creare fino a otto CoS sfruttando tre bit dell'etichetta MPLS, per gestire la QoS tra i Provider Edge router, e i tre bit del campo User Priority, per gestire la QoS tra i Provider Edge router e i Customer Edge router. I test sono stati eseguiti creando una congestione nella sezione core della rete (link a 1Gb/s) tramite un generatore/analizzatore di traffico. In particolare sono state considerate diverse situazioni di carico crescente, a partire da 953Mb/s fino a 1022Mb/s di traffico entrante in un nodo con CoS a bassa priorità (o, equivalentemente, senza priorità). Per ogni situazione di carico, un flusso dati High Definition (circa 20Mb/s) è stato inviato prima con CoS ad alta priorità, e poi con CoS a bassa priorità. Come mostrato in 2.3, per il flusso ad alta priorità non sono state rilevate perdite di pacchetti anche quando in rete vi era un traffico totale di 1042 Mb/s (1022 + 20 Mb/s); al contrario, perdite di dati si sono registrate per il flusso a bassa priorità non appena il link è giunto a saturazione, circa 975Mbit/s comprensivi del flusso sotto test. Si è così analizzato come il VPLS possa essere impiegato come tecnica per il controllo della QoS di servizi televisivi su IP quando si considera una rete di tipo gestito.

Per le reti non gestite si è considerata esclusivamente la tecnica di recupero

pacchetti SMPTE 2022-1 con diverse configurazioni e i risultati sono quelli riportati in 2.1.

Dal confronto delle due tabelle si può notare come l'utilizzo di una configurazione di SMPTE 2022-1 più robusta (25%) e quindi con consistente overhead apportata a tutti i flussi concorrenti possa non essere una buona soluzione in caso di congestione, in quanto la presenza di un alto overhead fa aumentare il carico sul collegamento, riducendo così la sua capacità effettiva in termini di numero di flussi video instradabili nella rete.

Tuttavia va considerato il fatto che la prioritizzazione non protegge da perdite diverse da quelle dovute per congestione, come link failure o perdite a livello fisico, mentre l'AL-FEC fornisce protezione rispetto qualsiasi evento di perdita, non solo al caso della congestione. Ovviamente al fine di non arrivare ad uno stato di congestione è bene diminuire il numero di flusso concorrenti su una medesima interfaccia di un router.

2.2.2.2 Link Failure: AL-FEC/SDH vs AL-FEC/VPLS

In caso di un evento di guasto, si possono verificare delle perdite di pacchetti che portano ad una degradazione della QoS dei flussi. In questa sezione, viene effettuata una comparazione delle prestazioni dell'AL-FEC in caso di guasto, ovvero di link failure; in particolare vengono paragonate le capacità per tempi tipici di fuori-servizio in reti SDH (intorno ai 50ms), 2.1.3.5, con quelli del VPLS, che è stato stimato intorno ai 26ms (con tecnica Fast ReRoute) [29]. L'AL-FEC viene utilizzato per la ricostruzione dei pacchetti persi durante il fuori-servizio, evitando così i relativi effetti sui flussi trasmessi. In questa situazione, il principale parametro di valutazione è il tempo di fuori-servizio, inoltre le prestazioni dell'AL-FEC sono legate anche al bit-rate del flusso video ''protetto''.

In reti SDH, dove il tempo necessario per il ripristino è di circa 50ms, le possibili

configurazioni della tecnica SMPTE 2022-1 permette di recuperare i pacchetti persi per flussi video con bit-rate non superiori a 4Mb/s. Quando si utilizza il VPLS, con tempi di ripristino di circa 26ms, è possibile riparare le perdite di flussi con bit-rate più elevati, anche fino a 8Mb/s, 2.10.

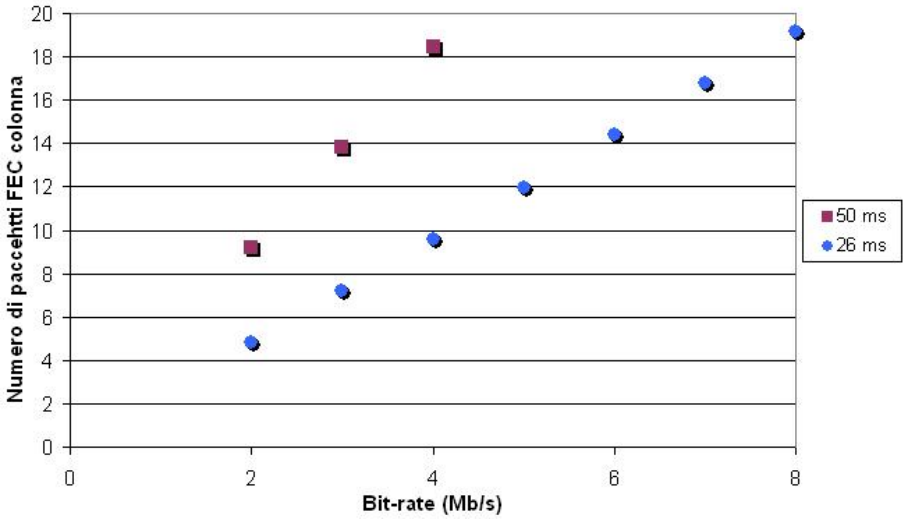


Figura 2.10: Ricostruzione pacchetti con FEC in caso di guasto.

2.2.3 Conclusioni

Dai risultati mostrati in questo lavoro, è possibile concludere che in una rete Managed l'utilizzo del VPLS assicura un controllo della QoS in caso di congestione e che in una rete Unmanaged, l'efficienza di una più robusta configurazione di AL-FEC SMPTE 2022-1 è controbilanciato da un maggiore overhead, che riduce il numero di flussi instradabili nella rete. I risultati mostrano che la tecnica AL-FEC considerata può essere utilizzate sia in reti IP Managed che Unmanaged, ma che da sola non è in grado di garantire la QoS in caso di forte congestione, quindi si può pensare che sia necessario adottare in caso di rete con alto tasso di perdita un AL-FEC più robusto, come i codici a fontana che verrano discussi in seguito. In caso di link failure, invece, si può concludere che, in combinazione

con il VPLS, i benefici delle tecniche AL-FEC sono inversamente proporzionali al bit-rate dei flussi protetti, permettendo di ricostruire tutti i pacchetti persi per flussi video fino a 8Mb/s, che il bit-rate degli attuali flussi in High Definition.

3

Prioritizzazione del solo flusso di correzione

Come già detto in precedenza il fattore probabilmente determinante per l'affermazione di servizi televisivi su IP è costituito dalla qualità dell'esperienza (QoE) che risulta legata in maniera essenziale alla qualità del servizio (QoS) di trasporto. Quindi l'affidabilità del trasporto risulta un fattore chiave da prendere in considerazione in quanto perdite ed errori sui pacchetti di rete possono danneggiare in maniera rilevante gli stream audio/video. Nelle reti IP le perdite di pacchetti possono essere causate da diversi tipi di eventi, quali il guasto del link fisico o errori residui sui bit del pacchetto non corretti dagli algoritmi FEC nello strato di trasporto o in quelli inferiori. In tale ambito eventi critici risultano essere quelli di congestione della rete in quanto possono determinare perdite di pacchetti di considerevole entità non sempre recuperabili e spesso recuperabili in maniera non completa.

Per ottenere che le piattaforme per la fornitura di servizi televisivi siano caratterizzate da perdite di pacchetti estremamente basse, è possibile prevedere opportune contromisure per contrastare le diverse cause di perdita implementando

meccanismi di prevenzione e protezione sia all'interno della rete di trasporto, [18], possibili solo nel caso di reti gestite e quindi non su Open Internet, sia di tipo end-to-end, cioè tra il server e il terminale dell'utente.

Nell'ambito delle reti gestite si è pensato di unire un meccanismo di protezione a livello di trasporto con un meccanismo di protezione end-to-end per contrastare le perdite avvenute per congestione. è stato fatto ciò in quanto nelle analisi riportate nel capitolo precedente il caso più critico è proprio quello di congestione .

Nelle reti IP con gestione del traffico è possibile realizzare strategie di prioritizzazione del traffico in modo da assicurare che i flussi televisivi siano trasferiti con un elevato grado di affidabilità, maggiore del rimanente traffico concorrenziale. Per realizzare ciò si possono adottare opportuni meccanismi che privilegino i pacchetti IP relativi a servizi ad ''alta qualità'' nell'assegnazione delle risorse di trasporto disponibili (prioritizzazione del traffico). L'implementazione di tali tecniche implica ovviamente la capacità di classificare e marcare opportunamente i pacchetti IP.
Dare una priorità alta ad un servizio comporta che i flussi concorrenti sullo stesso router, se il ruoter è in congestione, se hanno priorità inferiore, allora saranno più danneggiati dal router che non può scartare i pacchetti del flusso con alta priorità. Maggiore sono le dimensioni del flusso a cui viene data priorità e maggiore sarà la percentuale di pacchetti che viene scartata degli altri flussi.
Da questa ultima considerazione è stata pensata la strategia di proteggere il flusso di interesse con una tecnica AL-FEC e di dare priorità al solo flusso di protezione; ciò è possibile se il flusso di protezione è un flusso distinto dal flusso dati, come

nel caso della tecnica SMPTE 2022-1, [14] [15], che è una tecnica di protezione sistematica.
Con questa strategia si è pensato di poter migliorare le capacità di correzione dell'AL-FEC e di diminuire la percentuale di pacchetti che viene scartata degli altri flussi concorrenti del flusso di interesse sulla stessa interfaccia del router, se questo è in congestione, [33].

3.1 Soluzione proposta

L'uso delle tecniche FEC per proteggere gli stream di servizi multimediali comporta l'introduzione di un overhead sui dati trasmessi che generalmente è rappresentato da una percentuale calcolata sui pacchetti dati originali. Tuttavia se nella rete non si prevede nessun meccanismo di gestione del traffico, i pacchetti di protezione sono soggetti, nell'attraversamento della rete, alle stesse perdite ed errori dei pacchetti del flusso televisivo. Per tale ragione, le prestazioni del FEC si riducono rispetto a quelle teoriche, basate sull'assunzione che i pacchetti FEC relativi a pacchetti dati persi siano tutti disponibili.
Nelle reti IP con gestione del traffico è invece prefigurabile una strategia di prioritizzazione del traffico che consenta di ovviare a questo problema. In tale ambito è stato studiata una strategia che combinando la tecnica AL-FEC con un'opportuna prioritizzazione del traffico consenta di assicurare un'elevata qualità dei servizi televisivi minimizzando nello stesso tempo l'impatto della prioritizzazione sugli altri flussi di traffico nella rete.

La strategia proposta prevede di assegnare una priorità di trasferimento più elevata ai pacchetti FEC rispetto ai pacchetti dati, trasmessi in modalità best effort. In questa maniera si riduce, nel caso di router con interfaccia in congestione,

l'ammontare delle perdite dei pacchetti FEC. In altre parole, quando necessario, i router scartano preferibilmente i pacchetti dati rispetto ai pacchetti FEC.
Inoltre poichè gli stream FEC costituiscono usualmente una frazione ridotta dei relativi flussi informativi, nelle reti gestite la prioritizzazione dei soli flussi FEC minimizza l'impatto della prioritizzazione del traffico televisivo sugli altri tipi di traffico.
L'efficienza della strategia esposta è stata verificata attraverso simulazioni adoperando dei simulatori di rete, quindi i risultati sono stati successivamente validati sul test bed IP del Ministero dello Sviluppo Economico, al fine di avere un ambiente sperimentale più prossimo possibile ad infrastrutture di rete reali.

3.2 Test-bed sperimentale generato

Per dimostrare l'efficacia della strategia proposta è stata è costruita una catena sperimentale costituita da un sezione trasmissiva, un percorso di rete ed una doppia sezione di ricezione come illustrato in figura 3.2 e in figura 3.3.
L'architettura del test-bed, usato in alternativa al simulatore di rete descritto precedentemente, viene riportata in 3.1.

Lato trasmissione

Nel lato della trasmissione sono presenti un server video che rappresenta la sorgente, un convertitore USB-ASI ed un gateway ASI-IP. Il server video trasmette un programma televisivo a bit-rate costante sull'interfaccia USB.
Dopo la conversione da USB ad ASI il flusso è posto in ingresso al gateway ASI-IP che provvede ad incapsulare il TS nella pila di protocolli RTP/UDP/IP, come definito dal DVB, genera i pacchetti FEC relativi e trasmette sia i pacchetti informativi che i pacchetti FEC su un'interfaccia Ethernet verso un router IP.

Percorso di rete

Nel caso di simulazioni con assegnazione di priorità ai pacchetti FEC (3.3), il router separa i pacchetti informativi da quelli FEC che presentano lo stesso indirizzo IP di destinazione ma porte differenti. Il flusso dei pacchetti informativi viene inviato verso un simulatore di rete IP mentre il flusso (o i flussi) dei pacchetti FEC è inviato direttamente allo switch d'uscita. Nel caso di assenza di priorità, il router non effettua alcuna operazione e trasmette inalterati pacchetti al simulatore di rete. La connessione di rete punto-punto è stata simulata mediante un software di simulazione di rete eseguito su un PC dotato di doppia interfaccia Ethernet. Il software di simulazione opera secondo i più usuali modelli statistici per modellare il comportamento di una rete WAN di tipo IP relativamente alla latenza e alla perdita dei pacchetti.

Il test-bed a cui si è fatto ricorso è composto da una parte di ''accesso-metro'' ed una di ''core''. In particolare, nella rete di accesso sono presenti apparati di tipo xDSL e EPON, nella rete ''metro'' sono presenti tre router Cisco 3845 mentre la rete ''core'' è costituita da quattro router Juniper M10i. I router sono collegati attraverso interfacce ottiche ZX GbE e nella parte ''core'' i collegamenti in fibra ottica hanno lunghezze di 50km eccetto per una tratta di 300km che impiega amplificatori ottici. La sezione del test bed considerata per il test della strategia di prioritizzazione dei pacchetti FEC è la parte ''core''. Inoltre per la generazione delle congestioni e per il monitoraggio del traffico sono stati utilizzati gli apparati Spirent SmartBit 6000c e Antritsu MD1230B.

Lato ricezione

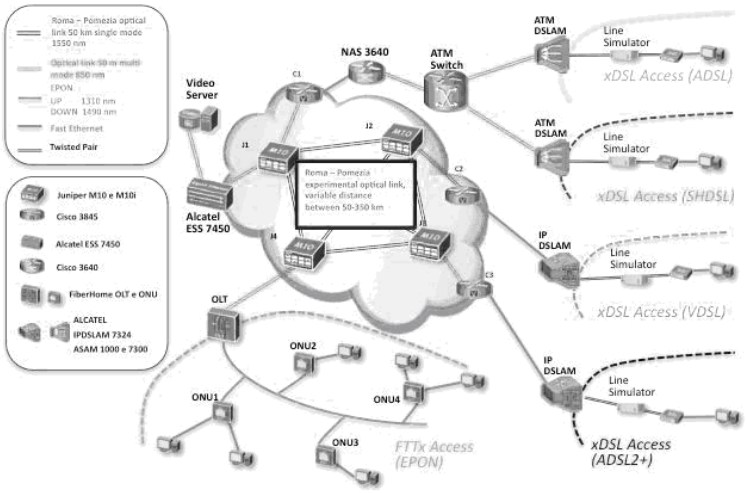


Figura 3.1: Test Bed di rete IP utilizzato.

La sezione di ricezione assume due configurazioni distinte in presenza e in assenza della priorità sui pacchetti FEC.

Nel caso di assenza di priorità (3.2) il gateway IP-ASI riceve i pacchetti dati e i pacchetti FEC e cerca di ricostruire le perdite sui pacchetti dati sulla base dei pacchetti FEC ricevuti. Il TS così ricostruito è posto in uscita su un'interfaccia ASI e tramite un convertitore ASI-USB viene ricevuto da un PC per una successiva analisi.

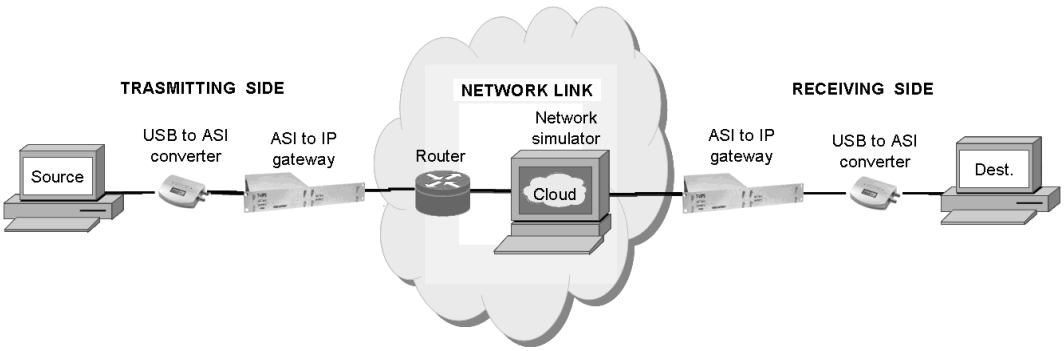


Figura 3.2: Catena di simulazione nel caso di assenza di prioritizzazione.

Nel caso di priorità dei pacchetti FEC (3.3), si osserva che poichè i flussi di pacchetti dati e quelli dei pacchetti FEC possono seguire percorsi differenti all'interno della rete (con caratteristiche di trasporto diverse) generalmente sono rice-

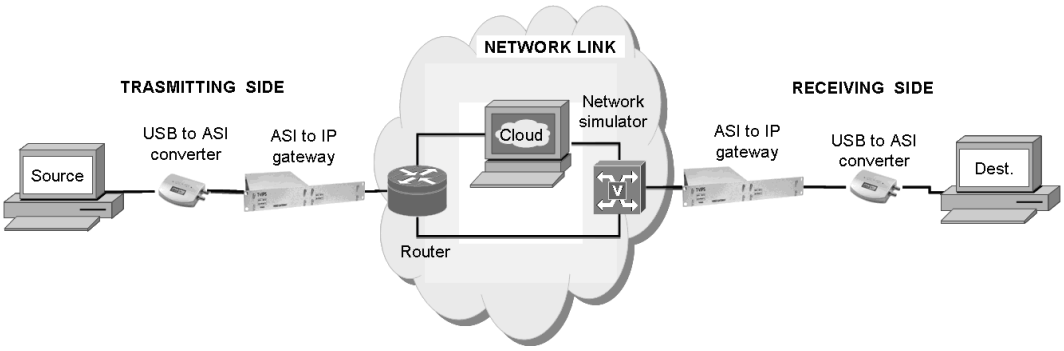


Figura 3.3: Catena di simulazione nel caso di prioritizzazione dei flussi FEC.

vuti dal decoder con una sequenza di interleaving differente da quella con cui sono trasmessi. Gli apparati gateway usati nelle simulazioni (T-VIPS) si sono rivelati particolarmente sensibili a tale ordinamento dimostrando, per un funzionamento corretto, una tolleranza limitata ad alterazioni dell'ordine di ricezione rispetto a quello di trasmissione.

Per ricostruire l'ordinamento di interleaving che il gateway di ricezione si aspetta per un corretto funzionamento, le simulazioni sono state condotte in due fasi distinte. In una prima fase, in ricezione, i flussi di pacchetti dati e FEC sono registrati su PC su file distinti secondo l'ordine di ricezione senza subire alterazioni. In una seconda fase, utilizzando un opportuno software, i pacchetti RTP sono letti dai file e trasmessi, ricostruendo l'ordine di trasmissione, al gateway di ricezione che opera, ove possibile, il recupero dei pacchetti dati persi sulla base dei pacchetti FEC ricevuti. Il TS così ricostruito è posto in uscita su un'interfaccia ASI e tramite un convertitore ASI-USB viene ricevuto da un PC per una successiva analisi.

3.3 Risultati sperimentali

Per verificare le prestazioni della strategia descritta in precedenza sono state esaminate sia perdite di tipo casuale, sia perdite determinate dalla congestione

alla interfacce dei router. La strategia proposta, infatti, ben si adatta a contrastare tali tipologie di perdite, in quanto le perdite casuali possono esser create ad un router con interfaccia che è prossima alla congestione. Mentre perdite di altro tipo quali quelle dovute a REIN (Repetitive Electrical Impulse Noise) ed ad interruzioni del collegamento (link failure) sono tali da non poter essere contrastate per questa via.
La valutazione delle prestazioni del FEC SMPTE 2022-1 e della strategia di prioritizzazione proposta è stata condotta valutando, per diversi casi di interesse, il minimo ammontare di overhead necessario al rispetto dei vincoli di qualità raccomandati dal Broadband Forum [12], in presenza di perdite di pacchetto aventi probabilità di occorrenza crescente (PLR). Come rappresentativi di servizi televisivi SD e HD sono stati considerati flussi TS MPEG-2 aventi bit-rate rispettivamente di 2 e 8 Mb/s (CBR), [17]. In tutti i casi esaminati si è fatto riferimento alla latenza minima per la corretta operatività del sistema. La validazione dei risultati delle simulazioni è stata effettuata mediante opportune sperimentazioni sul test bed IP.

La 3.4 riporta le prestazioni relative a configurazioni ‘‘solo colonne’’ e ‘‘righe e colonne’’ del FEC SMPTE 2022-1 nel caso in cui si utilizzi la strategia proposta di prioritizzazione del solo traffico FEC. Il bit rate di riferimento è pari a 2Mb/s ed il vincolo di qualità è quello di al massimo un artefatto visibile ogni ora di trasmissione.

Risulta che le capacità di correzione del FEC con la tecnica proposta si estendono fino al limite di $5 * 10^{-2}$. Inoltre, a parità di overhead è possibile si incrementa il valore massimo di PLR che è possibile tollerare; infatti, ad esempio, per un overhead del 20% il PLR tollerabile aumenta da $8 * 10^{-3}$ a $2 * 10^{-2}$).

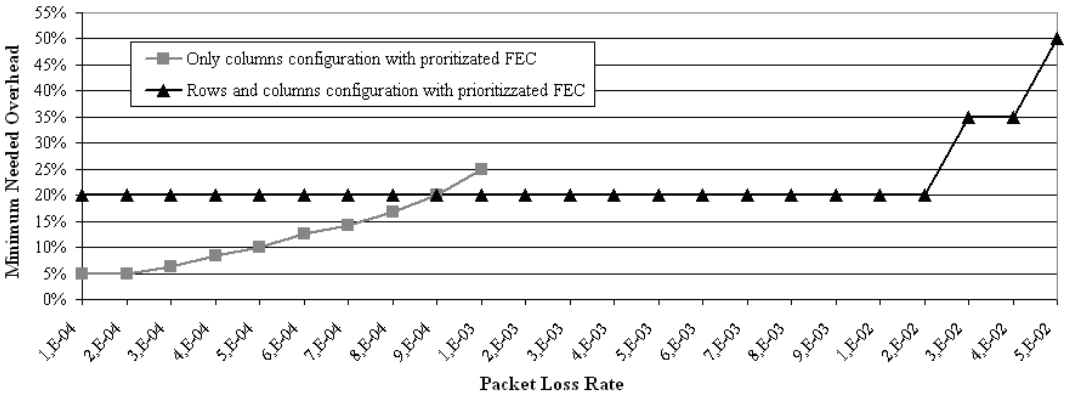


Figura 3.4: Prestazioni del FEC SMPTE 2022-1 per perdite casuali con prioritizzazione dei flussi FEC (flussi TV a 2Mb/s).

In entrambi i casi, con o senza priorità dei pacchetti FEC, i risultati illustrati evidenziano che le configurazioni di tipo ‘‘righe e colonne”, per perdite casuali sono più efficienti in termini di numero di pacchetti recuperati. Tuttavia utilizzando configurazioni di questo tipo il minimo overhead risulta piuttosto elevato essendo pari al 20%.

La 3.5 si riferisce allo stesso caso delle 3.4 ma per un flusso televisivo avente un bit-rate pari a 8Mb/s ed il vincolo di qualità è quello di al massimo un artefatto visibile ogni quattro ore di trasmissione. Tale vincolo è estremamente impegnativo per il FEC considerato, tuttavia la strategia proposta consente di recuperare perdite con PLR dell'ordine di 10^{-2}.

Anche per quanto riguarda le perdite di pacchetti causate da eventi di congestione dei router, sono stati ottenuti risultati significativi. In tal caso il flusso televisivo è stato inviato all'ingresso di un router sovrapposto ad altri flussi aventi la stessa bassa priorità bassa, quindi trasmessi in modalità best-effort. Tutti questi flussi insistono sulla stessa interfaccia fisica del router. Quando tale interfaccia è sovraccaricata il router inizia a scartare pacchetti appartenenti a tutti i flussi e quindi anche al flusso televisivo considerato. Invece nel caso in cui al

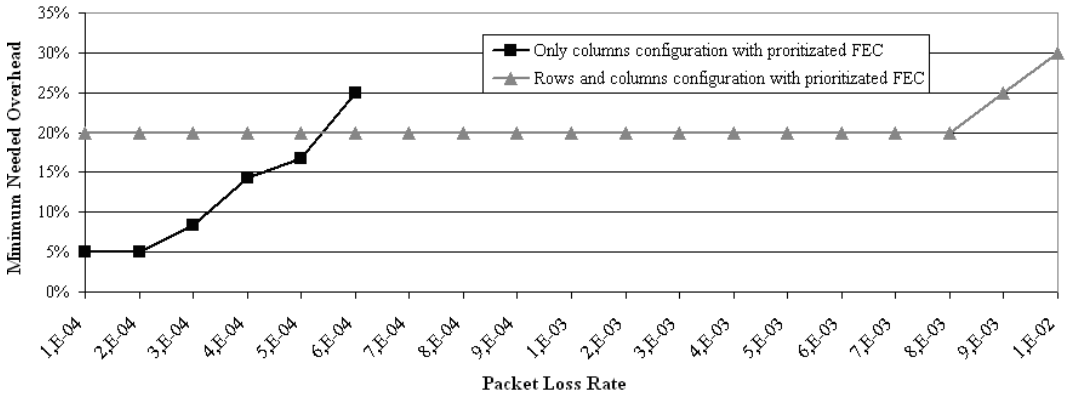


Figura 3.5: Prestazioni del FEC SMPTE2022-1 per perdite casuali con prioritizzazione dei flussi FEC (flussi TV a 8Mb/s).

flusso FEC associato a quello televisivo considerato viene assegnata una priorità più elevata, questo non sarà oggetto di scarto di pacchetti da parte del router in congestione. Per ogni test effettuato l'interfaccia presa in considerazione è stata sovraccaricata per 30 secondi. Per ogni test la configurazione del FEC è stata considerata adeguata se a lato destinatario non risultano pacchetti persi. I risultati ottenuti sono simili per flussi televisivi a 2Mb/s (caso dello Standard Definition) e a 8Mb/s (caso dell'High Definition).
In tale situazione, utilizzando lo stesso overhead, il guadagno della strategia proposta rispetto al caso di assenza di priorità è pari a circa un intervallo di graduazione sull'asse dei valori PLR. Per esempio, con il 5% di overhead, il valore di PLR tollerato passa da $9 * 10^{-4}$ a 10^{-3}, mentre con il 20% di overhead il valore di PLR tollerato passa da $2 * 10^{-2}$ a $4 * 10^{-2}$.

In 3.8 è illustrato un esempio visivo degli effetti di perdite di pacchetti su un quadro decodificato nei casi di assenza di FEC, FEC senza priorità e FEC con priorità per un flusso SD codificato in MPEG-2. Nell'esempio riportato in 3.8 l'uso del FEC con priorità consente il recupero completo delle perdite.
Si noti che il 20% di overhead per un flusso FEC corrisponde a un bit-rate di 400Kb/s per un flusso televisivo a 2Mb/s e a 1.6Mb/s per un flusso televisivo a

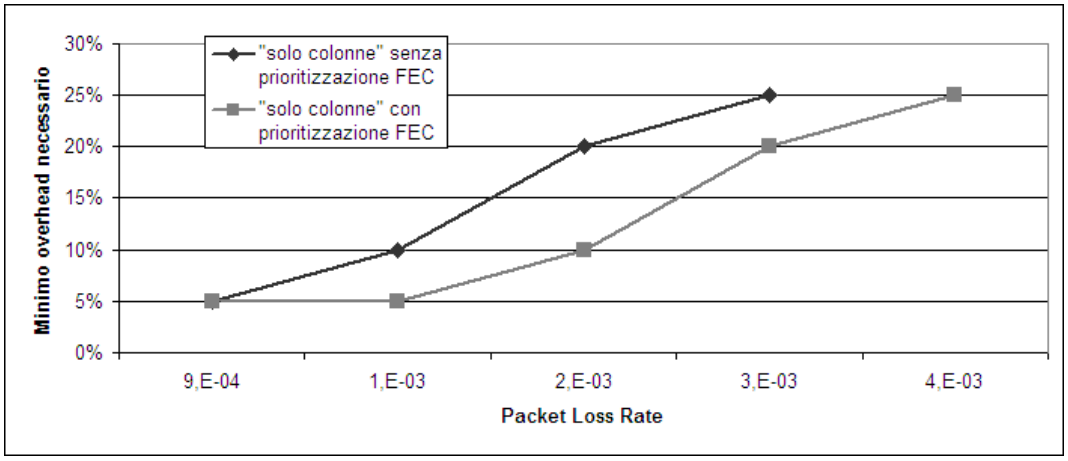


Figura 3.6: Prestazioni della configurazione ‘‘solo colonne’’ dellAL-FEC SMPTE 2022-1 nel caso di evento di congestione.

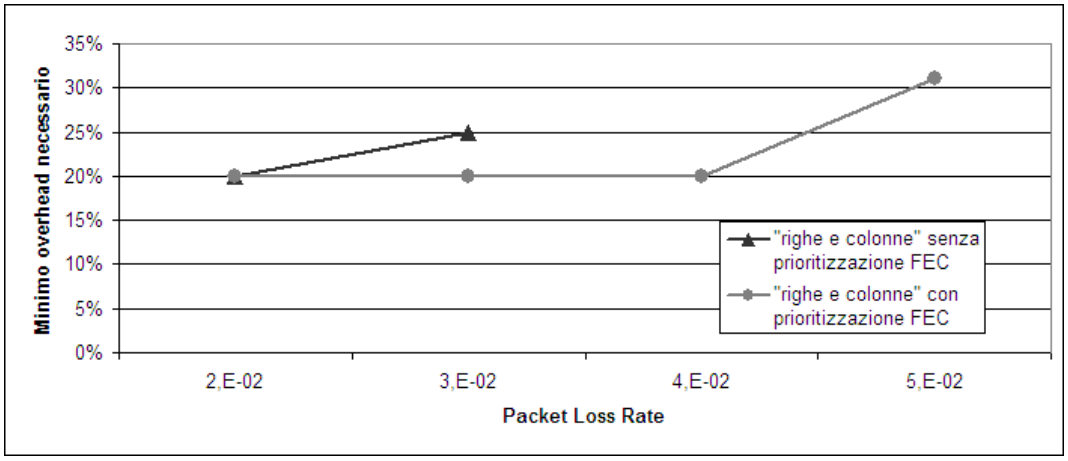


Figura 3.7: Prestazioni della configurazione ‘‘righe e colonne” dell’AL-FEC SMPTE 2022-1 nel caso di evento di congestione.

8Mb/s. Dalla considerazione di tali valori si può evincere che la strategia proposta consente di garantire un'elevata qualità di servizio senza danneggiare gli altri flussi concorrenti poichè la prioritizzazione di un flusso FEC su un'interfaccia non determina un incremento rilevante nella quantità di pacchetti scartati sugli altri flussi.

(a) (b)

(c) (d)

Figura 3.8: Esempio degli effetti della congestione di rete su un quadro decodificato (originale - senza FEC - con FEC senza priorità - con FEC prioritizzato).

4

Valutazione dell'efficacia di un sistema di protezione per metriche di qualità oggettive

Dopo aver valutato le prestazioni della tecnica di correzione SMPTE 2022-1 in merito alla qualità del servizio, rispettando i vincoli di qualità indicati dal Broadband Forum, si è pensato di valutare le prestazione dell'AL-FEC su un canale non affidabile, considerando in questo caso le metriche di qualità oggettive.
L'obiettivo dell'analisi è quindi quello di verificare l'efficacia della tecnica AL-FEC, in termini di qualità percepita, valutando così il compromesso tra l'overhead (ridondanza) dovuto all'utilizzo dell'AL-FEC e la qualità del video degradato da una certo numero di pacchetti persi.
Durante questa analisi quindi ci si svincola dal dover avere al massimo una perdita ogni ora per servizi SD e una perdita al massimo ogni quattro ore per servizi HD, in quanto all'aumentare delle perdite, fissato un determinato overhead, si vogliono valutare i diversi gradi di qualità percepita.
A tale scopo viene considerato come modello di errore di canale la congestione

di rete, vengono considerate quindi sia perdite casuali i.i.d. che piccole perdite a burst. Sono state utilizzate diverse sequenze video di test al fine di ottenere un'analisi il più possibile completa, considerando video con contenuti differenti. In particolare sono state analizzate sequenze video di cartoni animati e di telegiornali per quanto riguarda lo studio di video standard definition ed eventi sportivi, come una partita di calcio e la presentazioni delle Olimpiadi invernali per quanto riguarda lo studio di video high definition.

La qualità percepita a lato destinatario, e quindi dall'utente finale, è stata valutata tramite la metrica *full-reference* standard NTIA-VQM [31] e tramite i risultati ottenuti si sono valutate le prestazione dell'AL-FEC su un canale non affidabile.

Dopo aver valutato l'efficacia di un sistema di protezione di errore end-to-end dal punto di vista della qualità percepita dall'utente finale si è pensato di creare un sistema di monitoraggio della qualità tramite una nuova metrica oggettiva *no-reference*, [20].
L'intento finale della sperimentazione è proprio quello di individuare un sistema che monitori la qualità a lato destinatario, in maniera tale da non far scendere questa ultima oltre una soglia prestabilita. Nel momento in cui il livello di qualità scende sotto la soglia, si aumenta l'overhead della tecnica di correzione. In tal maniera è possibile mantenere un determinato livello di qualità senza conoscere lo stato della rete e senza avere la necessità di disporre di una rete gestita. Infatti unendo la metriche di qualità no-reference con l'AL-FEC è possibile effettuare tale controllo.

4.1 Metriche oggettive analizzate

Nell'ultimi anni la ricerca in merito alla qualità di sequenze video si è molto concentrata sullo sviluppo di metriche di qualità video *oggettive* che simulassero il comportamento *soggettivo*, basandosi sia sulla sequenza video originale che sulla sequenza video ricevuta.

Se la metrica si basa sia sulla sequenza originale, che su quella ricevuta, allora la metrica viene detta *full-reference*; tale metrica non può però essere utilizzata nella applicazioni, come quelle real-time, dove non si ha a disposizione la sequenza video originale a lato destinatario.

Oltre alle metriche *full-reference* sono nate le metriche *reduced-reference* che si basano sull'estrazione di caratteristiche del video originale, che racchiudono il contenuto del video e che vengono quindi trasmesse al lato ricevente, incrementando però in questa maniera l'occupazione della banda.

Le uniche che possono esser utilizzate per le applicazioni real-time senza l'incremento dell'occupazione di banda, sono le metriche *no-reference* che si basano unicamente sulle sequenze video ricevute, senza esser per questo meno accurate delle metriche *full-reference*.

4.1.1 Metrica full-reference considerata

La metrica VQM-NTIA, ideata dal National Telecommunications and Information Administration/Information Technology Services (NTIA/ITS) del Ministero del Commercio degli USA, [31], stabilisce la qualità di un video tramite un indicatore della fedeltà di ricostruzione VQM. I valori che può assumere il risultato della metrica sono compresi nell'intervallo $[0, 1]$. In particolare, una valutazione, che restituisce un indicatore di valutazione di qualità oggettiva pari a zero, indica che i video analizzati sono perfettamente identici. Al contrario, il valore uno

indica il numero massimo di differenze. Tale indicatore viene valutato tramite la combinazione di:

- indicatori basati sui gradienti spaziali, che caratterizzano il grado di distorsione dei bordi di un'immagine effettivamente percepibile;
- indicatori delle distorsioni di cui sono affette le componenti di crominanza di ciascun quadro;
- indicatori che misurano il contenuto informativo locale relativo al contrasto, ovvero il degrado della qualità prodotto da una riduzione della banda (blurring), con conseguente perdita di contrasto, o da un rumore additivo;
- indicatori basati sull'informazione temporale assoluta che misurano le distorsioni relative al campo di moto e che sono sensibili quindi a degradazioni prodotte dalla perdita o dalla ripetizione di uno o più quadri, e da rumore additivo;
- indicatori basati sul prodotto tra il contrasto e l'informazione temporale che tengono conto sia della variabilità delle visibilità di un artefatto in funzione della rapidità con cui varia la scena (ovvero dell'entità del moto), sia della variazione del grado di apprezzamento delle distorsioni temporali in funzione del minore o maggiore numero di dettagli presenti nella singola scena.

Negli esperimenti effettuati, al fine di valutare in modo dinamico l'impatto degli errori sulla qualità percepita, è stata applicata una finestra mobile nel calcolo della metrica VQM-NTIA.

4.1.2 Metrica no-reference considerata

La seconda metrica di qualità considerata è una metrica *no-reference* progettata per la valutazione della qualità di trasmissione video su reti basate su IP eterogenee, [20]. Questa metrica si basa sull'analisi della correlazione interframe misurata a lato ricevitore, al fine di valutare la presenza di cluster e blocchi persi isolati. Non si necessita quindi di informazioni sul tipo di errori e ritardi che hanno afflitto il collegamento e le contromisure introdotte dal decoder per affrontare la potenziale perdita di qualità.

La metrica no-reference presentata è stata messa a punto tramite l'adattamento al Mean Opinion Score, ottenuto tramite esperimenti soggettivi, come per la metrica full-reference Video Quality Metrics. Inoltre assume valori compresi nell'intervallo $[1, 5]$, dove 5 rappresenta il punteggio associato la massima qualità. Tuttavia l'intervallo della metrica no-reference è continuo e non quantizzato come nel caso della Mean Opinion Score soggettivo.

4.2 Sistema proposto

Il sistema proposto per la sperimentazione è così fatto: il trasmettitore è composto da un server video, un convertitore USB-ASI ed un gateway ASI-IP. Il server video trasmette un programma televisivo a bit rate costante sull'interfaccia USB. Dopo la conversione da USB ad ASI il flusso posto in ingresso al gateway ASI-IP, che provvede ad incapsulare il TS nella pila di protocolli RTP/UDP/IP come definito dal DVB, genera i pacchetti di correzione relativi ai pacchetti dati e trasmette sia i pacchetti informativi che i pacchetti FEC su un'interfaccia Ethernet verso un router IP.

Il segmento di rete consiste in un emulatore di rete, ''Netem'', che può introdurre

sui flussi in ingresso perdite casuali e simulare lo scarto di pacchetti a causa di una interfaccia di un router in congestione. In questo modo è possibile simulare i comportamenti di una vera e propria rete IP best-effort.
La prima parte del ricevitore è un blocco che registra i flussi ricevuti; un PC riceve direttamente i pacchetti RTP dati dalla rete, ma non i pacchetti FEC. In questo modo il flusso di dati ricevuto senza l'ausilio dell'AL-FEC e quindi con tutti i pacchetti persi è registrato e conservato. È possibile far ciò in quando l'AL-FEC considerato, l'SMPTE 2022-1, è sistematico.
La seconda parte del blocco ricevente è un IP video DVB-ASI gateway che prende i pacchetti RTP dati e i pacchetti FEC e tenta il recupero dei pacchetti persi, utilizzando i pacchetti FEC ricevuti. L'IP video DVB-ASI gateway permette di valutare la percentuale di pacchetti persi e la percentuale di pacchetti recuperati. Il TS recuperato viene poi inviato ad un interfaccia ASI che alimenta in parallelo un decoder MPEG-2 utilizzato per riprodurre i flussi ricevuti e, attraverso l'ASI-USB, un PC che registra il TS ricevuto.
Questa catena sperimentale consente il confronto tra i flussi video decodificati con e senza supporto della tecnica AL-FEC. Alla fine del set-up sperimentale viene valutata la qualità del video nel caso in cui non si usino i pacchetti di correzione e nel caso invece in cui vengano sfruttati i pacchetti FEC. Il risultato ottenuto viene analizzato al fine di decidere se è necessario un aumento dell'overhead dell'AL-FEC e quindi di una maggiore protezione o se l'overhead può essere diminuito per evitare uno spreco del consumo di banda.

4.3 Risultati sperimentali

In questa sperimentazione sono stati analizzati quattro tipi diversi di flussi video, due in Standard Definition e due in High Definition. I flussi in SD sono un tele-

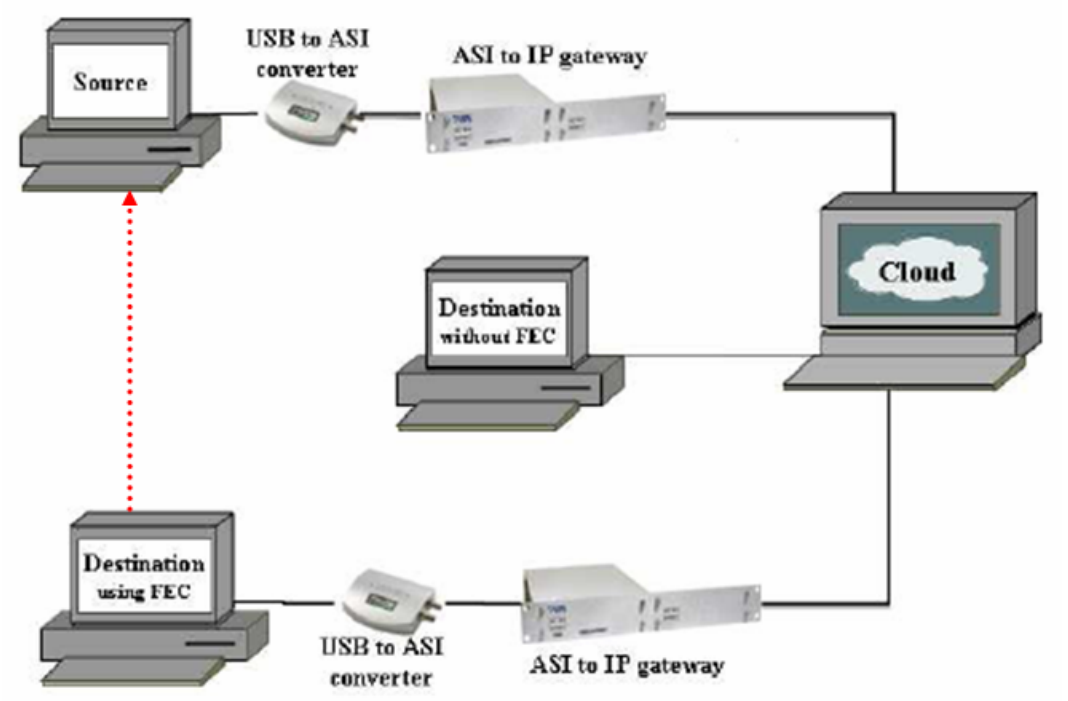


Figura 4.1: Set-up sperimentale.

giornale a 2Mb/s e un altro a 4Mb/s, e un cartone animato a 2Mb/s e un altro a 4Mb/s. I flussi in HD sono due sequenze significativamente differenti della presentazione delle Olimpiadi Invernali Vancouver 2010 ad 8Mb/s e due sequenze video significativamente differenti di una partita di calcio a 15Mb/s. Tutte le sequenze hanno la durata di un minuto e tutte sono state sottoposte a perdite random i.i.d. dovute ad un evento di congestione durante la trasmissione nella catena sperimentale.

Le prove sono state effettuate variando il Packet Loss Rate e conseguentemente l'overhead introdotto dall'AL-FEC SMPTE 2022-1. Si è preferito non utilizzare un overhead maggiore del 20% per non creare una elevata occupazione della banda, ma ottenendo allo stesso tempo delle buone prestazioni dell'AL-FEC considerato, avendo analizzato i risultati delle sperimentazioni riportate nel capitoli precedenti. Gli overhead considerati, con le rispettive configurazioni, sono:

1. 10% con configurazione ‘‘solo colonne’’,
2. 16.67% con configurazione ‘‘solo colonne’’,
3. 20% con configurazione ‘‘righe e colonne’’.

Per ogni prova effettuata il confronto è stato fatto solamente tra il video che ha usufruito dell'ausilio dei pacchetti di riparazione e tra il video che non ne ha usufruito, questo perchè non tutti i pacchetti hanno la stessa importanza da un punto di vista della qualità dell'esperienza, come è stato detto precedentemente, ma le coppie di video considerate hanno gli stessi identici pacchetti persi.
Inoltre la qualità del video è stata valutata su 700 frame e dopo ogni valutazione si è aumentato o il Packet Loss Rate o l'overhead.
Si arriva in ogni caso ad un punto in cui la capacità di correzione del Forward Error Correction non è più in grado di contrastare le perdite in rete.

Alcuni dei risultati ottenuti sono mostrati nei grafici 4.2, 4.3, 4.4, 4.5, 4.6 e 4.7. Da tali grafici è possibile notare come il caso in cui venga utilizzata la tecnica di correzione d'errore riporti sempre una qualità del video più alta. Malgrado ciò la differenza tra i livelli di qualità non è sempre costante, in quanto i pacchetti non hanno tutti la stessa importanza e nel momento in cui l'AL-FEC considerato non riesce a recuperare determinati pacchetti con importanza maggiore, allora il livello di qualità scende molto. Per questo motivo ultimamente vengono molto studiate le tecniche di ‘‘Unequal Protection”, che permettono una maggiore protezione nei confronti di quei pacchetti che contengono informazioni di importanza più elevata.

In generale i risultati sperimentali hanno mostrato come l'aumento dell'overhead (per esempio dal 16.67% di overhead a 20% di overhead) permetta di

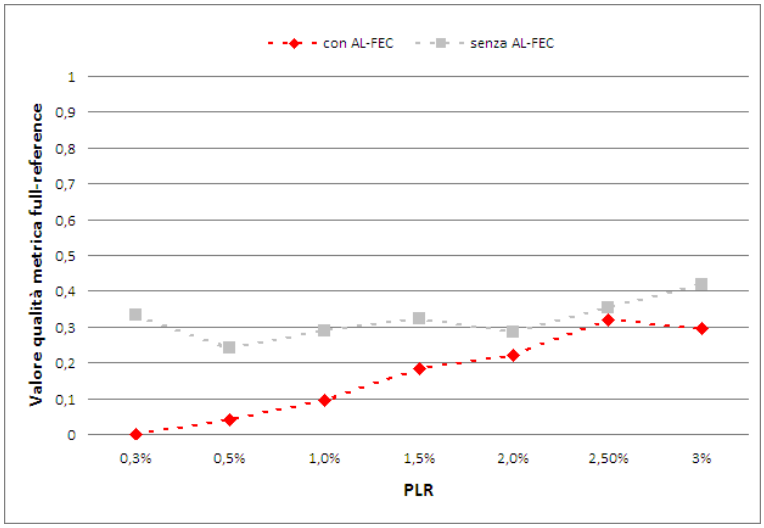


Figura 4.2: Cartone animato, video SD: risultati della metrica VQM-NTIA con e senza la protezione FEC per un overhead del 10%.

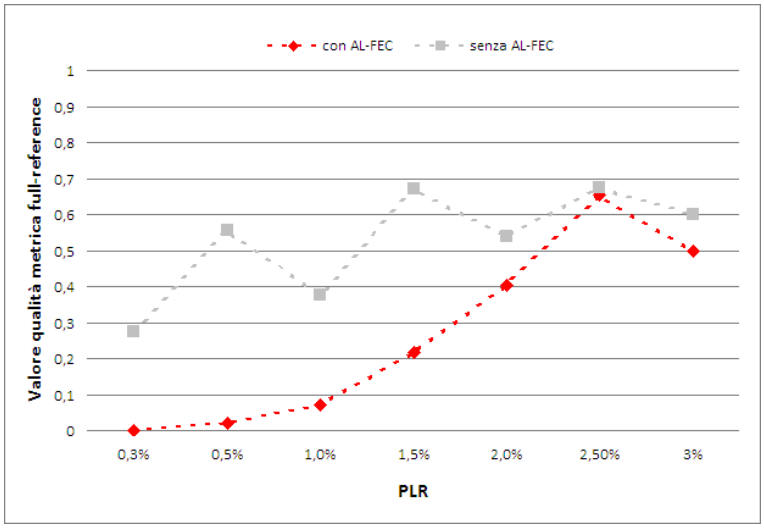


Figura 4.3: Telegiornale, video SD: risultati della metrica VQM-NTIA con e senza la protezione FEC per un overhead del 10%.

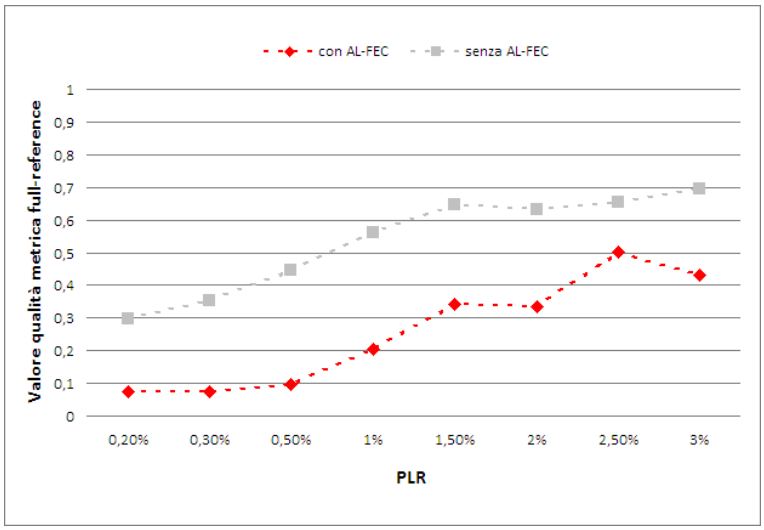


Figura 4.4: Olimpiadi, video HD: risultati della metrica VQM-NTIA con e senza la protezione FEC per un overhead del 10%.

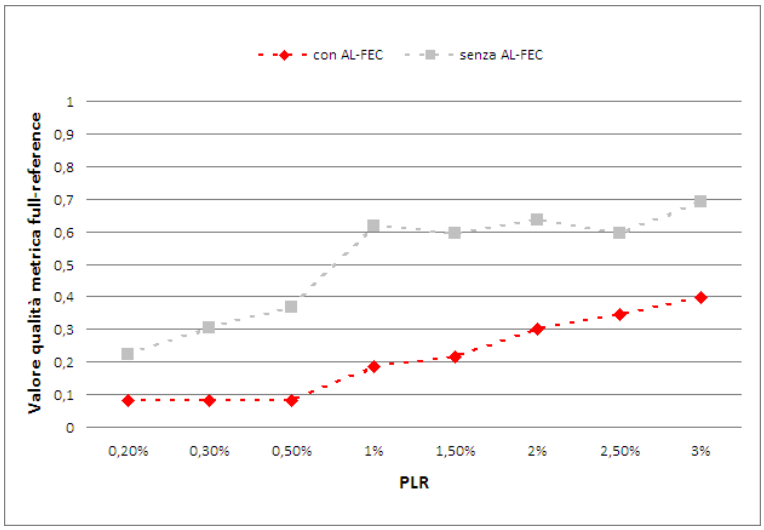


Figura 4.5: Olimpiadi, video HD: risultati della metrica VQM-NTIA con e senza la protezione FEC per un overhead del 16.67%.

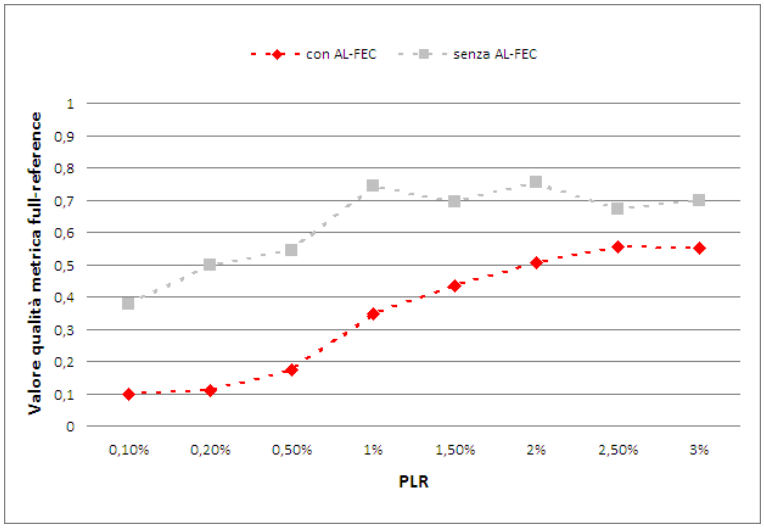


Figura 4.6: Partita di calcio, video HD: risultati della metrica VQM-NTIA con e senza la protezione FEC per un overhead del 10%.

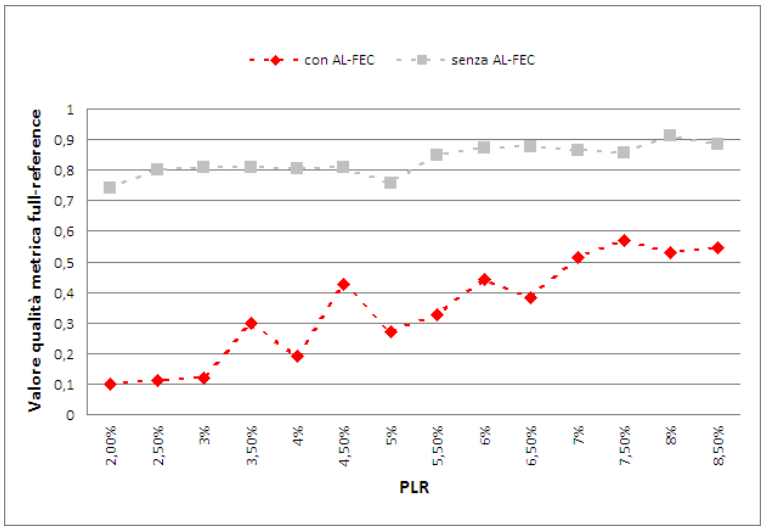


Figura 4.7: Partita di calcio, video HD: risultati della metrica VQM-NTIA con e senza la protezione FEC per un overhead del 16.67%.

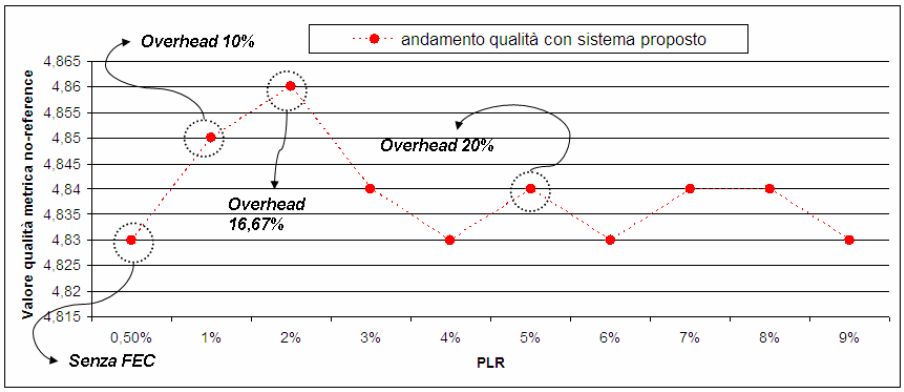


Figura 4.8: Cartone animato a 4Mb/s, video SD: andamento della qualitàpercepita nel sistema proposto.

guadagnare un punto sul MOS (Mean Opinion Score). Tuttavia all'aumentare del PLR di molto oltre le capacità di correzione dell'AL-FEC, la differenza tra la qualità percepita valutata tramite le metriche oggettive sul video sul quale non viene utilizzata la strategia di recupero dei pacchetti persi e la qualità percepita valutata tramite le metriche oggettive sul video sul quale viene utilizzata la strategia di recupero dei pacchetti persi tende a diminuire.

Alcuni esempi sull'andamento della qualità percepita all'aumentare del Packet Loss Rate, attuando il sistema proposto sono posti in 4.8, 4.9 e in 4.10. I grafici proposti non hanno ovviamente un andamento lineare in quanto all'occorrenza è stato aumentato l'overhead.

In figura 4.11 è possibile vedere gli effetti visivi per differenti Packet Loss Rate sia nel caso in cui venga utilizzata la tecnica di correzione dei pacchetti, sia nel caso in cui non venga utilizzata.

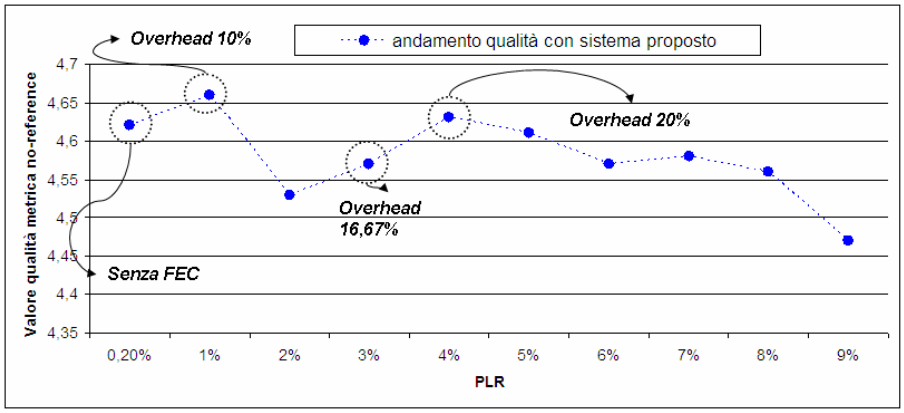


Figura 4.9: Olimpiadi a 8Mb/s, video HD: andamento della qualitàpercepita nel sistema proposto.

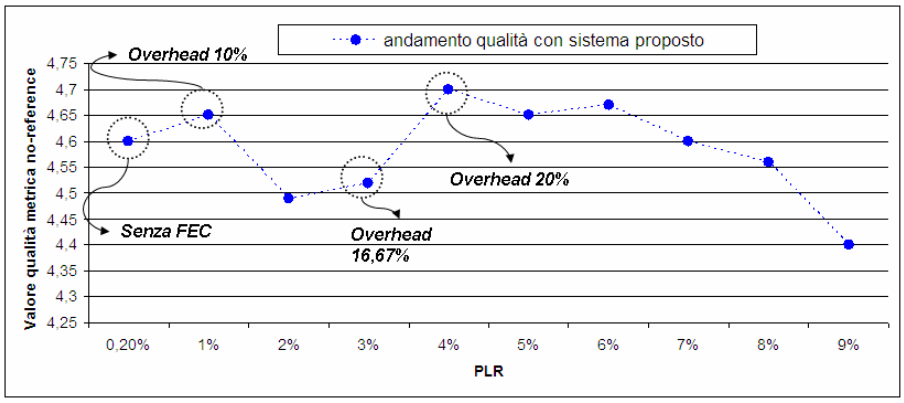


Figura 4.10: Partita di calcio a 15Mb/s, video HD: andamento della qualitàpercepita nel sistema proposto.

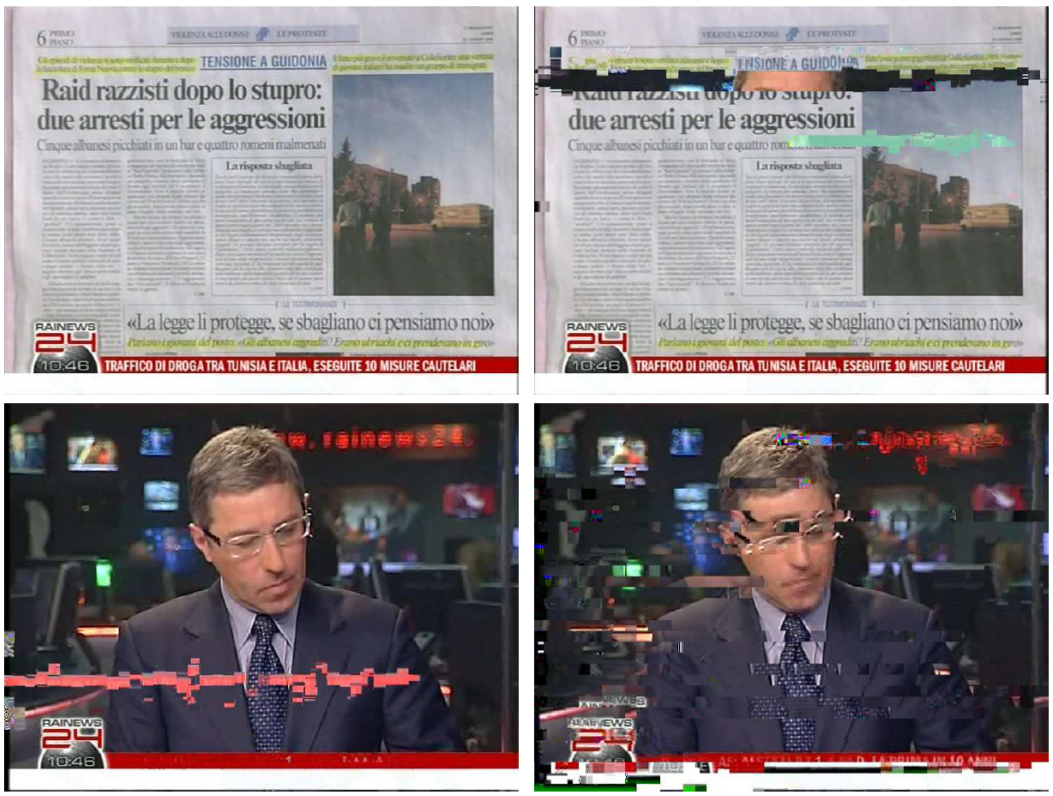


Figura 4.11: Effettivi visivi a differenti PRL nel caso di utilizzo del FEC (sinistra) e il non utilizzo (destra).

5

Multicast efficace tramite codici a fontana in reti MANET

L'affidabilità per servizi real-time trasmessi su reti con perdite è un obiettivo di non facile raggiungimento, ma allo stesso tempo molto cercato. Quel che viene proposto in questa ultima sperimentazione è un sistema che permetta una trasmissione multicast efficace su reti di tipo MANET sfruttando i codici a fontana. Viene quindi combinata la prima realizzazione pratica dei codici a fontana, i codici LT, con la trasmissione di tipo multicast su reti wireless sfruttando il protocollo PUMA. Le proprietà dei codici a fontana di essere rateless e di generare pacchetti linearmente indipendenti, rendono tali codici una soluzione interessante per applicazioni multicast su reti con perdite e con condizioni di canale varianti, come sono le reti MANET. L'obiettivo finale è quello di ottenere il livello più alto possibile di qualità del servizio trasmesso.

5.1 Codici a fontana

I codici a fontana sono una classe di codici creata per canali con perdite a pacchetto, che ha rivoluzionato il paradigma di trasmissione di tipo standard, [27]. Infatti, mentre nelle trasmissioni standard i pacchetti inviati vengono creati dalla semplice suddivisone del file sorgente, tali codici producono dei pacchetti ognuno dei quali è funzione dell'intero file sorgente.
L'idea base dei codici a fontana è la seguente: dato un numero finito K di pacchetti del file sorgente, il trasmettitore produce un flusso di pacchetti codificati, da inviare sul canale, potenzialmente infinito. A lato destinatario il ricevitore colleziona pacchetti fino a quando non raggiunge una quantità di essi pari a $K(1+\varepsilon)$, dove $\varepsilon * K$ è una percentuale del numero dei pacchetti del file sorgente, che gli permette di ricostruire l'intero messaggio sorgente, senza controllare quali pacchetti gli sono arrivati e quali no. Quindi il trasmettitore invia pacchetti senza sapere quali pacchetti siano stati realmente ricevuti e il destinatario colleziona pacchetti senza considerare quali pacchetti stia ricevendo, ma solamente la quantità. Un codice a fontana sarà tanto più efficiente, tanto più ε è tende a zero.
Da questa idea base nasce la metafora della fontana, [23], dove il codificatore è una fontana che produce una quantità infinita di gocce, che sono i pacchetti codificati, e chiunque voglia ricostruire il flusso sorgente deve solamente raccogliere $K(1+\varepsilon)$ gocce.

A differenza dei codici di correzione classici, i codici a fontana sono rateless, ovvero senza un ritmo di fissato a priori, e universali, ovvero la lunghezza del pacchetto non influisce sulla codifica. Inoltre i pacchetti codificati sono ottenuti tutti in maniera casuale ed indipendente. Ogni pacchetto codificato contiene al suo interno l'informazione dei pacchetti sorgente che hanno contribuito alla sua generazione. In generale i pacchetti codificati sono dati ciascuno dalla somma di

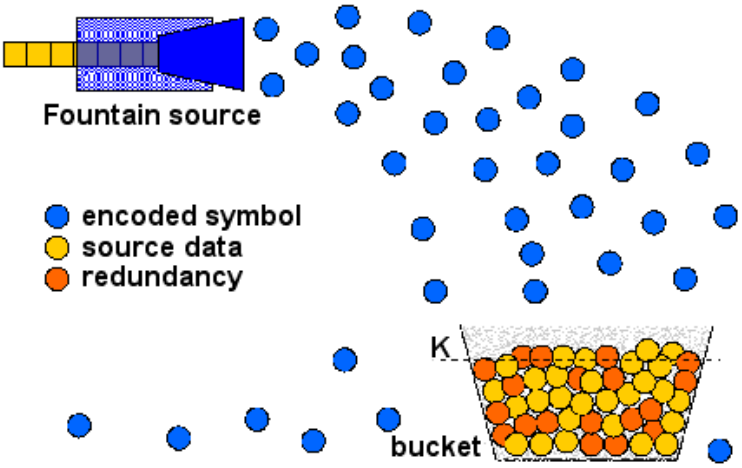


Figura 5.1: Metafora della fontana.

un insieme di pacchetti sorgente scelti in base ad una determinata distribuzione.

Per illustrare il funzionamento del processo di codifica e di decodifica dei codici a fontana, si prende ora in considerazione il codice a fontana lineare con grafo generato in maniera casuale, ovvero il codice più semplice della sua categoria.
Il processo di codifica è scandito dai clock di un orologio. Inizialmente si suddivide il file sorgente in K pacchetti sorgente, ovvero $\mathbf{s} = [s_1, s_2, s_3, ..., s_K]$ e il pacchetto viene considerata l'unità elementare. Quindi ad ogni ciclo di clock n il codificatore genera K bit in maniera random, creando, di volta in volta, la matrice di codifica G_{kn}. Se si vogliono generare N pacchetti codificati, allora ci saranno N cicli di clock. In questa maniera la matrice di codifica G_{kn} sarà di dimensioni $K \times N$. Ogni *k-esimo* pacchetto sorgente viene quindi moltiplicato interamente per il *k-esimo* bit corrispondente della matrice G_{kn}. Quindi tutti i pacchetti risultanti vengono sommati bit a bit. Questa coppia di operazioni viene fatta per ogni ciclo di clock n. In questa maniera da ogni coppia di operazioni viene prodotto un pacchetto di lunghezza pari alla lunghezza dei pacchetti sorgente.
Il pacchetto codificato da trasmette t_n sarà quindi dato dalla somma modulo due

dei pacchetti sorgente per i quali l'elemento corrispondente della matrice G_{kn} è pari ad 1.

$$t_n = \sum_{k=1}^{K} s_k G_{kn} \tag{5.1}$$

Dato che la coppia di operazioni deve essere fatta per N volte, verranno prodotti in tutto N pacchetti di lunghezza pari alla lunghezza del pacchetto sorgente.

Per la decodifica, a lato ricevente, il destinatario deve raccogliere $K(1+\varepsilon)$ pacchetti t_n per poter recuperare i K pacchetti sorgente.
Affinchè la decodifica abbia successo la matrice utile per la decodifica deve essere invertibile. Affinchè una matrice sia invertibile, questa deve essere quadrata e le righe devono essere linearmente indipendenti. La matrice deve quindi avere rango pieno ed il determinante diverso da zero. Maggiore è il numero dei pacchetti che si hanno in ricezione e maggiore è la probabilità trovare K pacchetti le cui corrispondenti righe della matrice siano linearmente indipendenti, potendo così attuare l'operazione di inversione della matrice.

Presi i K pacchetti, per i quali la cui matrice corrispondente è invertibile, i restanti pacchetti si scartano. La matrice di decodifica $\mathbf{G}$ deve essere quadrata, $K \times K$, ed è un frammento della matrice di codifica. Per poter invertire la matrice si utilizza il processo di *eliminazione di Gauss*. I K pacchetti vengono quindi moltiplicati per i corrispondenti elementi della matrice invertita e sommati tra di loro bit a bit. Per ogni operazione, si ricava un pacchetto sorgente.

$$s_k = \sum_{n=1}^{N} t_n G_{nk}^{-1} \tag{5.2}$$

Per quando riguarda il processo di decodifica, si presuppone che il decodifi-

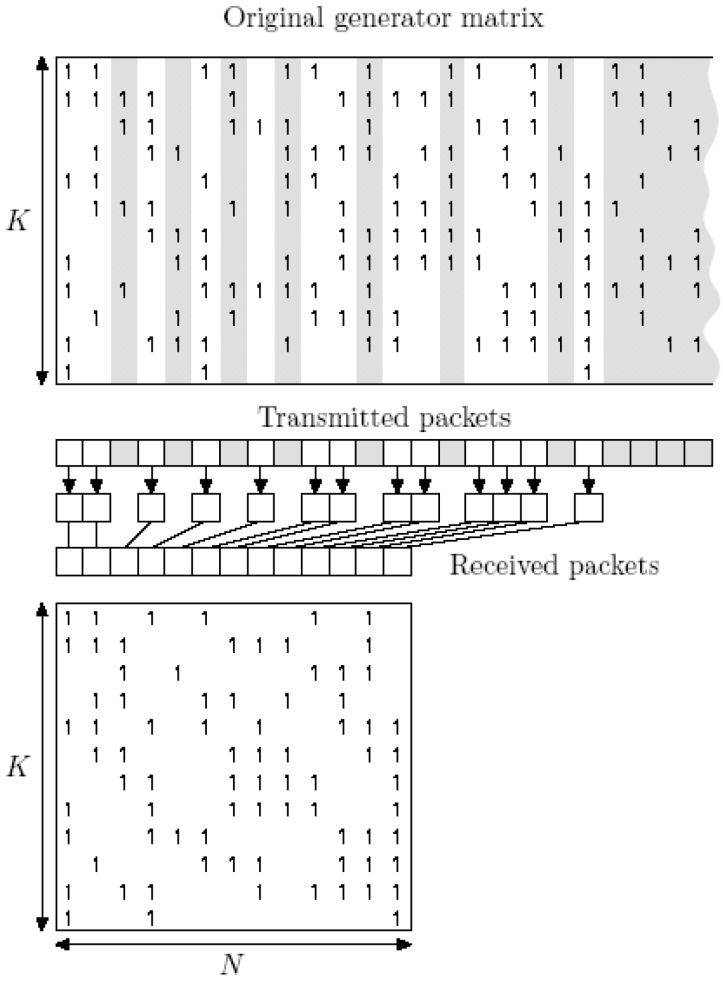


Figura 5.2: Matrice generatrice di un codice lineare random e matrice di decodifica.

catore conosca la matrice G_{kn} e questo è possibile se la matrice viene generata in maniera pseudo-casuale e quindi il codificatore e il decodificatore possiedono lo stesso generatore random e sono sincronizzati tra di loro. In alternativa i pacchetti t_n trasmessi possono contenere al loro interno, nella parte di overhead, una chiave che indica quali pacchetti sorgente hanno contribuito alla loro generazione; in questa maniera il decodificatore è in grado di risalire alle connessioni del grafo e quindi al frammento di matrice utile per la decodifica.

La probabilità di successo della decodifica è legata alla probabilità di successo dell'operazione di inversione della matrice di decodifica. Logicamente se il numero dei pacchetti ricevuti N è minore del numero dei pacchetti sorgente K, allora il ricevitore non ha abbastanza informazioni per poter ricostruire il file sorgente e quindi la decodifica fallisce.
Se invece $N = K$ allora è teoricamente possibile invertire la matrice, ma bisogna verificare che la matrice $K \times K$ abbia rango pieno. Minore è il numero dei pacchetti ricevuti e minore è la probabilità di trovare una matrice con rango pieno.
La probabilità che una matrice $\mathbf{G}$ con dimensioni $K \times K$ abbia rango pieno è data dal prodotto di K probabilità, ciascuna delle quali è la probabilità che una nuova colonna di $\mathbf{G}$ sia linearmente indipendente dalle colonne precedenti e che tutte le righe siano linearmente indipendenti dalle altre righe. Il primo elemento di questo prodotto è la probabilità che la prima colonna di $\mathbf{G}$ non sia nulla e quindi pari a $\left(1 - 2^{-K}\right)$, il secondo elemento è la probabilità che la seconda colonna non sia nulla e sia diversa dalla prima colonna, quindi pari a $\left(1 - 2^{-(K-1)}\right)$; iterando questo processo, si ottiene che la probabilità di invertire la matrice $\mathbf{G}$ è pari a:

$$\left(1-2^{-K}\right)\left(1-2^{-(K-1)}\right)\times\ldots\times\left(1-\frac{1}{8}\right)\left(1-\frac{1}{4}\right)\left(1-\frac{1}{2}\right) \tag{5.3}$$

Se $N = K$ allora questa probabilità complessiva sarà molto bassa, circa pari a 0.289 con K maggiore di 10.

Per ottenere una maggiore probabilità di successo della decodifica è necessario che N sia maggiore di K e quindi la probabilità di successo è strettamente legata alla quantità ε. Infatti, definendo δ la probabilità di fallimento della decodifica, e quindi $1-\delta$ la probabilità di successo, posto $K = 100$, è possibile graficare le prestazioni del codice a fontana random lineare, valutando la probabilità di fallimento in funzione di ε, 5.3.

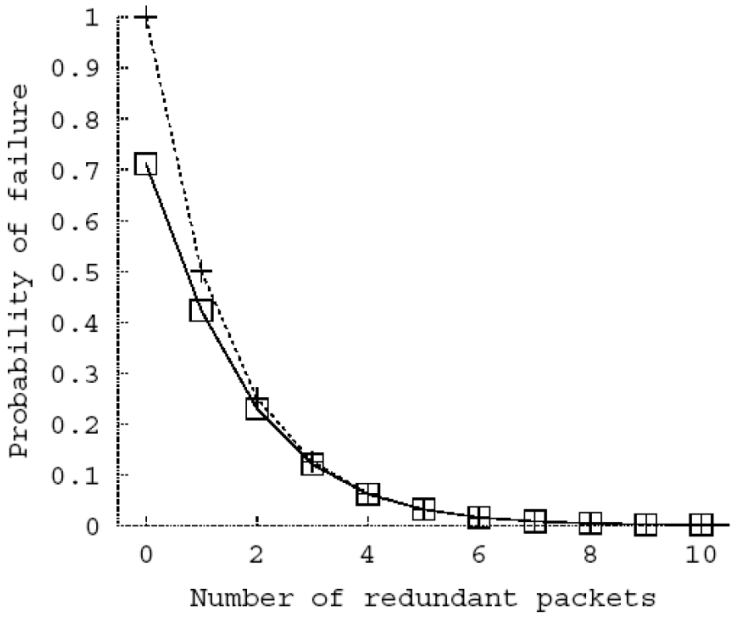


Figura 5.3: Andamento della probabilità di fallimento in funzione di ε.

Da queste prestazioni si deduce che il limite superiore della probabilità di fallimento in funzione di ε è pari a $2^{-\varepsilon}$, quindi all'aumentare di ε diminuisce la probabilità di fallimento; mentre la probabilità di successo sarà pari a $K+\log_2 1/\delta$. La probabilità di successo quindi aumenta all'aumentare di K. Tuttavia all'au-

mentare di K aumenta anche il costo computazionale della codifica, in maniera quadratica, e della decodifica, in maniera cubica.

5.1.1 Codici LT

I codici LT sono la prima realizzazione pratica dei codici a fontana. Infatti i codici LT riprendono i codici a fontana random lineari, ma con un algoritmo di decodifica con costo computazionale meno oneroso, in quanto alla creazione di ogni singolo pacchetto codificato t_n non contribuiscono tutti i pacchetti sorgente, ma solo un numero limitato di essi. Per stabilire quanti pacchetti sorgente devono contribuire alla creazione dei pacchetti codificati, è allora necessaria una ''distribuzione dei gradi''. Il cuore dei codici LT è proprio questa distribuzione. Il grado d_n indica, appunto, il numero dei pacchetti sorgente a cui il pacchetto codificato è collegato. La codifica è composta da due passi:

1. per ogni pacchetto da codificare t_n viene stabilito in maniera random il grado d_n, in base alla distribuzione dei gradi scelta;

2. vengono scelti in maniera casuale ed uniforme quali d_n pacchetti sorgente sommare bit a bit per creare il pacchetto codificato t_n.

Il processo di decodifica può esser fatto sfruttando il *message passing*, al fine di evitare l'utilizzo dell'eliminazione di Gauss, ben più onerosa dal punto di vista del costo computazionale. Per effettuare il *message passing* devono esser i seguenti tre step:

1. si cerca un pacchetto codificato t_n collegato ad un solo pacchetto sorgente s_k e si pone $s_k = t_n$;

2. si somma il precedente s_k a tutti i pacchetti codificati ad esso connessi e che quindi ha contribuito a generare;

3. si cancellano tutti i collegamenti relativi ad s_k.

Dopo l'ultimo passo, si prendono in considerazione tutti i pacchetti codificati modificati al passo due e si ritorna al primo passo, cercando il pacchetto codificato modificato che è collegato ad un solo pacchetto sorgente.

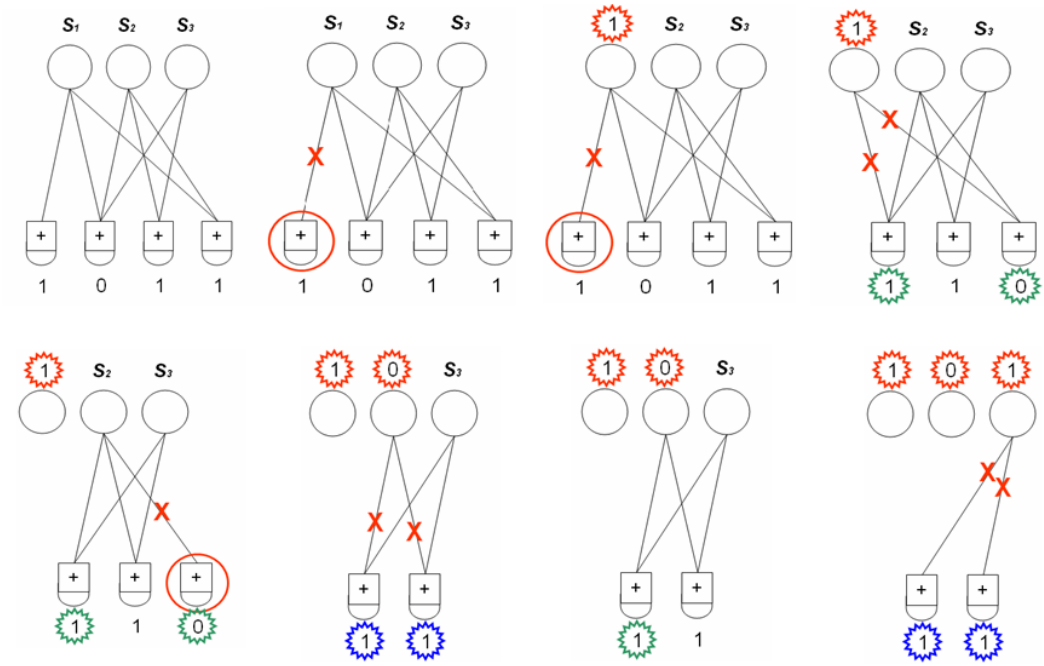


Figura 5.4: Message passing

Si consideri che in questo algoritmo di codifica e decodifica l'unità elementare è il pacchetto e che il canale considerato è un canale PEC, per cui un pacchetto o arriva completamente corretto o viene scartato. Non esiste in questo caso la correzione a livello di bit; basta quindi un solo bit errato per far sì che il pacchetto venga completamente scartato.

Il cuore dei codici LT sta quindi nella distribuzione dei gradi, infatti è necessario che tutti i pacchetti sorgente siano collegati ad almeno un pacchetto codificato, altrimenti si perde l'informazione contenuta in quel pacchetto sorgente. Inoltre è necessario che ci siano sufficienti pacchetti codificati con un grado basso,

in maniera tale da consentire al processo di decodifica di partire e di non interrompersi, e per avere allo stesso tempo un costo computazionale basso. Infatti maggiore sono i collegamenti, più complessa è la decodifica; a questo punto la quantità cruciale è il grado medio.

Dato che il codificatore crea collegamenti tra pacchetti codificati e pacchetti sorgenti in maniera casuale, allora il numero di collegamenti deve essere almeno dell'ordine di $K * ln(K)$ e, supponendo di collezionare in ricezione un numero di pacchetti codificati vicino all'ottimo, ovvero K, allora il grado medio di ciascun pacchetto sarà $ln(K)$, con costo computazionale di codifica e decodifica pari a $K * ln(K)$. Tali valori ottimi possono essere raggiunti con una corretta scelta della distribuzione dei gradi.

Si avrebbe un comportamento ideale in fase di decodifica se ad ogni iterazione fosse presente un solo pacchetto codificato con grado uno. Tale comportamento è raggiungibile dalla *distribuzione dei gradi solitona ideale*:

$$\rho(d) = \begin{cases} \frac{1}{K} & d = 1 \\ \frac{1}{d(d-1)} & d = 2, \ldots, K \end{cases} \tag{5.4}$$

Il problema di tale distribuzione sta nelle fluttuazioni intorno al comportamento ideale. Infatti tali fluttuazioni fanno sì che il processo di decodifica ad un certo punto possa bloccarsi, avendo esaurita la quantità di pacchetti codificati con grado uno.

La distribuzione solitona robusta è una versione modificata della distribuzione solitona ideale, che supera il problema dell'interruzione del processo di decodifica, tramite l'introduzione di due parametri: c e δ. c è un parametro libero, mentre δ è un parametro che limita la probabilità che la decodifica fallisca ed indica quando la matrice di codifica sia sparsa una volta fissato N. Diminuendo δ aumentano il

numero dei collegamenti medio e di conseguenza aumenta il costo computazionale. In tale distribuzione il numero atteso di pacchetti codificati con grado uno è pari a $S = c \ln\left(\frac{K}{\delta}\right)\sqrt{K}$. La distribuzione robusta solitona è:

$$\mu(d) = \frac{\rho(d) + \tau(d)}{Z} \tag{5.5}$$

dove $\rho(d)$ è la distribuzione solitona ideale:

$$\rho\left(d\right) = \begin{cases} \frac{1}{K} & d = 1 \\ \frac{1}{d(d-1)} & d = 2, \ldots, K \end{cases} \tag{5.6}$$

e $\tau(d)$ è pari a:

$$\tau(d) = \begin{cases} \frac{s}{K} \cdot \frac{1}{d} d = 1, 2, ..., \left(\frac{K}{S}\right) - 1 \\ \frac{s}{K} \log\left(\frac{S}{\delta}\right) d = \frac{K}{S} \\ 0 d > \frac{K}{S} \end{cases} \tag{5.7}$$

e

$$Z = \left(\sum\nolimits_d \left(\rho\left(d\right) + \tau\left(d\right)\right)\right) \tag{5.8}$$

L'andamento $\tau(d)$ per valori piccoli di d garantisce che il processo di decodifica abbia inizio, mentre il picco che presenta per $d = K/S$ assicura che ogni pacchetto sorgente abbia un collegamento. Con un'appropriata scelta di c e δ, sarà quindi possibile recuperare tutti i pacchetti sorgenti, avendo ricevuto una quantità di pacchetti codificati pari a $K + 2\ln\left(\frac{S}{\delta}\right) S$.

5.1.2 Ottimizzazione dei codici LT

Le prestazioni della distribuzione solitona robusta sono molto dipendenti dai valori assegnati ai parametri c e a δ. Uno studio riportato nell'articolo [28] riporta

dei nuovi criteri per la selezione dei valori dei parametri critici della distribuzione solitona robusta. Tale studio è stato ripreso per la sperimentazione fatta in merito al multicast efficace tramite codici a fontana in reti MANET. è infatti noto che i codici LT sono vicini all'ottimo per qualsiasi canale con perdita, ma questo è vero solo con una buona progettazione della distribuzione dei gradi.
In tale studio è stato evidenziato come il parametro c influenzi notevolmente le prestazioni del codice LT; sperimentalmente diminuendo c diminuisce il numero medio di gradi e la probabilità di grado uno; allo stesso tempo il primo picco aumenta e il secondo piccolo diminuisce allontanandosi dal primo picco. I limiti del parametro c sono i seguenti:

$$\frac{1}{k-1}\frac{\sqrt{k}}{\ln\left(\frac{k}{\delta}\right)} \leq c \leq \frac{1}{2}\frac{\sqrt{k}}{\ln\left(\frac{k}{\delta}\right)} \tag{5.9}$$

L'ottimizzazione della distribuzione solitona robusta è stata effettuata per valori di K compresi tra 10^4 e 10^6; l'obiettivo principale è l'individuazione dei valori di c e di δ che minimizzino l'overhead totale, ovvero la quantità $\varepsilon * K$ necessaria per la decodifica. Nel range di valori di K considerati le migliori prestazioni si ottengono per $c = 0.02$, mentre si è trovato che l'overhead totale non dipende da δ, che può essere quindi selezionato come parametro che gioca il ruolo del *trade-off* tra la probabilità di fallimento richiesta e quando la matrice di codifica, fissato N, sia sparsa.

5.1.3 Codici LT per file sorgente di piccole dimensioni

I codici LT con distribuzione solita robusta hanno solitamente buone prestazioni per file sorgente di grandi dimensioni, ovvero sull'ordine almeno di 10^4; in generale si può dire che al crescere di K si migliorano le prestazioni. Tuttavia è spesso

richiesto che il file sorgente sia di piccole dimensioni per diminuire la latenza dal lato utente. Cambiando però la distribuzione dei gradi è possibile ottenere delle prestazioni comunque buone per file sorgente sull'ordine di 10^4; ciò è possibile applicando un algoritmo iterativo per l'ottimizzazione della distribuzione dei gradi ideale solitona [38].

Una caratteristica delle distribuzioni solitone è che le probabilità di avere pacchetti codificati di grado uno è minore rispetto a quella di avere pacchetti codificati di grado due; ciò implica il rischio di non ricevere abbastanza pacchetti codificati con grado uno. Poichè la distribuzione ideale solitona non opera in maniera adeguata, sono stati cambiati i primi due elementi della distribuzione dei gradi solitona ideale ed il centesimo elemento, ottenendo una distribuzione dei gradi più idonea per messaggi sorgente di piccole dimensioni. Considerando quindi, un valore di $K = 1000$, la distribuzione applicata è la seguente:

$$p_i = \begin{cases} \eta_1, & \text{for} \quad i = 1, \\ \eta_2, & \text{for} \quad i = 2, \\ \eta_3, & \text{for} \quad i = 100, \\ \frac{1}{i(i-1)}, & \text{for} \quad i = 3, \ldots, 99 \text{ and } i = 101, \ldots, n. \end{cases} \tag{5.10}$$

Questa distribuzione viene di seguito normalizzata per ottenere la corretta probabilità per ogni componente. I valori ottimali per $K = 1000$ sono stati sperimentalmente trovati e sono $\eta_{opt} = (0.083, 0.487, 0.032)$; una corretta scelta dei valori dei parametri è fondamentale in quanto varia di molto le prestazioni del codice LT. Tale distribuzione riporta prestazioni migliori della distribuzione solitona robusta nel caso di messaggi di piccole dimensioni, ovvero con $K = 100$ o con $K = 1000$.

Tale distribuzione è stata ripresa per la sperimentazione fatta in merito al multicast efficace tramite codici a fontana in reti MANET.

5.1.4 Codici Raptor

I codici Raptor sono codici a fontana che discendono dai codici LT, [37]. L'obiettivo dei codici Raptor è la diminuzione del costo computazionale dei codici LT, mantenendo allo stesso tempo il livello prestazionale di questi. I codice LT rimane in ogni caso il cuore della struttura dei codici Raptor. L'idea è infatti quella di richiedere che solo una frazione costante dei pacchetti sorgente sia collegata ai pacchetti da trasmette. Per poter far ciò, è necessario concatenare un altro codice di correzione d'errore, affinchè i pacchetti sorgenti non collegati ai pacchetti codificati, non vadano persi grazie alla protezione data dal codice d'errore concatenato. La configurazione classica dei codici Raptor è data dalla figura 5.5.

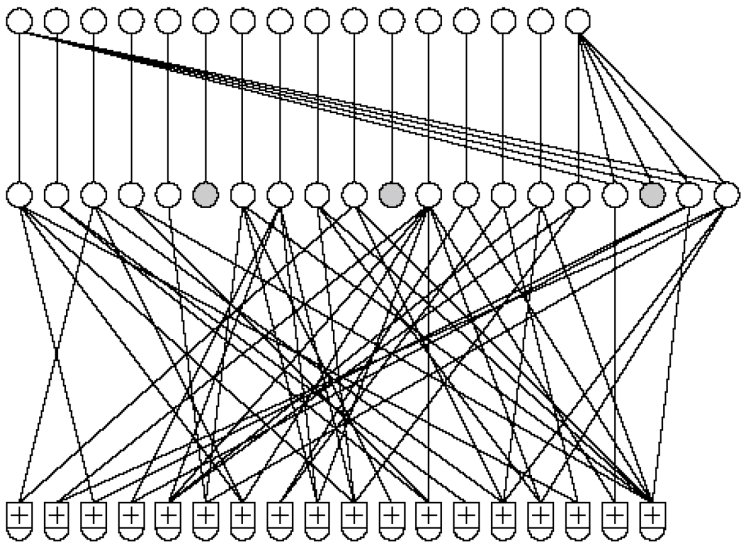


Figura 5.5: Modello di codifica Raptor

In questo approccio, si è dimostrato che il costo computazionale di codifica e decodifica varia linearmente con la dimensione del file sorgente. I codici Raptor si differenziano per il codice esterno di correzione d'errore concatenato utilizzato e per la distribuzione dei gradi della codifica LT scelta.

In generale i codici Raptor, a differenza dei codici LT, possono essere sia sistem-

atici, ovvero permetto al destinatario l'accesso diretto ai dati sorgente, sia non-sistematici. La differenza tra un codice Raptor sistematico e uno non-sistematico, non sta nella struttura del codice, ma nel pre-processamento dei pacchetti in ingresso. Se il codice Raptor è non-sistematico, allora i pacchetti sorgente vengono tutti pre-codificati da un codice esterno i cui pacchetti in uscita vengono posti in ingresso al codificatore LT "indebolito", ovvero con un grado medio di collegamenti basso, che genera i pacchetti codificati da trasmettere.

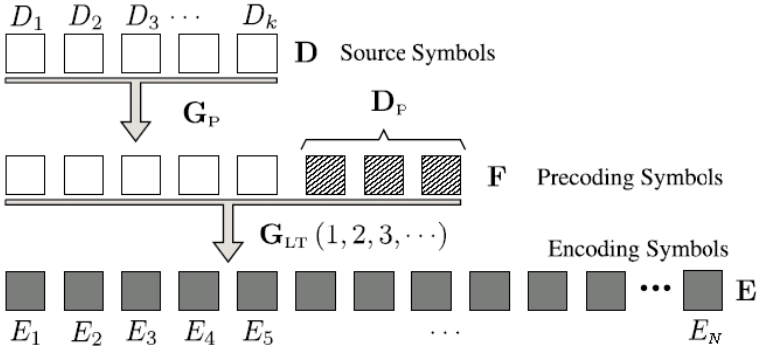


Figura 5.6: Codifica Raptor non sistematica

Se il codice Raptor deve essere sistematico, allora è necessario trovare una funzione appropriata di mappaggio tra le parole di informazione e le parole di codice, in maniera tale che i primi K simboli corrispondano ai K simboli sorgente. Questo è possibile farlo se i pacchetti in ingresso al codice esterno vengono pre-processati tramite l'inversa della matrice di codifica e l'output del pre-processamento viene dato in ingresso al codice esterno.

Si può far ciò in quanto, secondo la teoria dei codici, le proprietà di un codice sono determinate dalle parole di codice e non dalle regole attraverso le quali esse sono ottenute (ovvero le regole di mappaggio tra parole d'informazione e parole di codice).

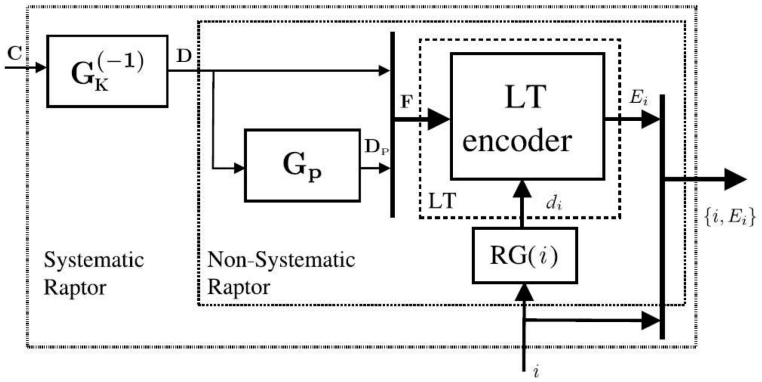


Figura 5.7: Codifica Raptor sistematica

5.2 Reti ad-hoc

In generale le reti di comunicazione si dividono in fisse, ovvero cablate, e temporanee, ovvero wireless. Una rete cablata, presentando una connettività via cavo, può raggiungere alte velocità di connessione e allo stesso tempo non è affetta da interferenze; una rete wireless, non necessitando invece di una connettività a livello fisico tra nodi, non presenta gli alti costi di cablaggio, ma è affetta da interferenze dovute alla presenza di altri dispositivi o di ostacoli presenti nella comunicazione tra due terminali.
Inoltre in una rete wireless si può cambiare configurazione del sistema di comunicazione in modo veloce e si ha una maggiore flessibilità e adattamento ai cambi di configurazione della rete. D'altra parte i problemi maggiori per una rete wireless sono la propagazione multipath, il path-loss, le interferenze e lo spettro di frequenza limitato.

La tecnologia di comunicazione wireless si basa su due principali modelli di rete sviluppati per i sistemi di comunicazione. Nel primo si ha un'infrastruttura portante e un'insieme di nodi mobili che si connettono ad essa utilizzando un mezzo di comunicazione wireless. Il secondo modello riguarda le reti ad-hoc mo-

bile, ovvero le MANET (Mobile Ad-hoc NETwork) [26].
Le MANET sono un insieme di nodi mobili, in grado di *auto-organizzarsi* formando dinamicamente una rete su un canale wireless condiviso senza l'utilizzo di una infrastruttura portante o di una amministrazione centrale. Ogni nodo è provvisto di antenna che consente di avere la capacità di ricevere e trasmettere informazioni. Quindi ogni nodo può prendere parte alla costruzione della rete e ad una eventuale individuazione di ulteriori nodi che si inseriscono; inoltre si possono mantenere anche i percorsi tra un nodo e l'altro. In base al range trasmissivo di un terminale è possibile avere una comunicazione diretta tra sorgente e destinatario, oppure, come avviene nella maggior parte dei casi, la comunicazione tra due nodi si ha utilizzando i nodi intermedi della rete mediante la tecnica del *multi-hop*, utilizzando determinati protocolli di instradamento per far comunicare le due parti. Per tale motivo i nodi delle MANET possono operare sia come ''host'' sia come ''router'', instradando i pacchetti per gli altri nodi di rete nel caso in cui non ci sia una comunicazione diretta tra le due parti interessate. Le MANET possono anche essere chiamate reti radio a pacchetto multi-hop, permettendo ai nodi di partecipare attivamente alla ricerca di percorsi di rete per instradare i pacchetti e possono esser connesse all'Open Internet.

Per la realizzazione di queste reti bisogna tenere in considerazione molti aspetti, quali l'alta dinamicità della rete causata dal continuo movimento dei nodi mobili che portano a cambiamenti frequenti della topologia di rete, ma anche la diversa capacità del canale wireless che causa la perdita di pacchetti. Inoltre la natura dei mezzi wireless introduce i problemi di nodi nascosti e di nodi esposti. Bisogna tenere in considerazione anche la risorsa di banda disponibile. è evidente che questa tecnologia può avere delle capacità inferiori a quella di rete wired, ma

si adatta efficacemente a diversi scenari applicativi.
L'utilizzo di appropriati protocolli di instradamento porta dei benefici in termini di cambiamenti di rete, consumo di energia e altri aspetti fondamentali. Tuttavia è impossibile dire quale protocollo di routing sia migliore, perchè cambia in base alle diverse condizioni di rete.

5.2.1 Protocol Unified Manet Announcement

PUMA è un protocollo di tipo multicast, [24], utilizzato per reti MANET, in grado di stabilire e mantenere un rete condivisa di tipo mesh per ogni gruppo multicast, [25], senza l'utilizzo di un protocollo di instradamento unicast, oppure l'uso di nodi core per i gruppi della rete.
Una rete mesh è realizzata con una combinazione di nodi fissi e mobili interconnessi tra di loro con collegamenti wireless per formare una rete ad-hoc multihop. PUMA realizza un alto rate di trasmissione dei dati con un limitato overhead di controllo, caratteristica che lo rende un protocollo costante per un ampio range di condizioni di reti. La scelta del protocollo PUMA tra tanti protocolli di routing è stata fatta tramite il confronto tra le caratteristiche di PUMA e le caratteristiche di altri protocolli come MAODV e ODMRP. Il primo protocollo si basa sulla creazione dei percorsi di rete utilizzando il criterio di instradamento ad albero, mentre il secondo è basato anch'esso, come PUMA, sulla creazione di percorsi di rete mesh. Valutando i tre protocolli in scenari di diverso tipo, come ad esempio la variabilità della mobilità, il numero di nodi membri di un gruppo, il numero dei mittenti, il carico di traffico e il numero di gruppi multicast, PUMA mostra dei valori più alti di PDR (Packet Delivery Ratio) rispetto a MAODV e ODMRP ricorrendo ad un minore overhead di controllo. La novità di PUMA deriva dall'utilizzo di messaggi di segnalazione (multicast announcements) molto semplici,

in grado di realizzare tutte le funzioni necessarie per la costruzione e il mantenimento di una struttura di instradamento multicast nella MANET. I multicast announcements vengono utilizzati per i seguenti scopi:

- selezionare dinamicamente i nodi core;
- determinare i percorsi per le sorgenti al di fuori di un gruppo multicast per inoltrare pacchetti multicast di dati verso il gruppo;
- entrare o lasciare la mesh di un determinato gruppo;
- mantenere la mesh di un gruppo.

In una rete dinamica ogni nodo può inondare di pacchetti la rete per raggiungere i ricevitori di un determinato gruppo multicast, ma anche tutti gli altri nodi che fanno parte della rete; in alternativa i ricevitori selezionano un nodo intermedio, in questo caso il core, che viene utilizzato come punto di contatto tra il gruppo multicast e i nodi che non ne fanno parte, detti non-membri, e i nodi intermedi che devono inondare la rete con l'informazione necessaria per informare gli altri nodi della loro esistenza. In PUMA viene scelto il secondo approccio, infatti dalle diverse analisi effettuate si nota come l'overhead sia indipendente dai fattori che caratterizzano le diverse condizioni degli scenari di rete.

PUMA supporta il modello di servizio multicast IP, consentendo ad ogni sorgente di inviare pacchetti multicast ad un determinato gruppo multicast, senza conoscere da quali nodi è composto il relativo gruppo. Inoltre non c'è bisogno che una sorgente si unisca ad un gruppo multicast per inviare pacchetti dati al gruppo stesso. Inoltre utilizza un approccio di tipo *receiver-initiated*, quindi basato sul nodo ricevitore, utilizzando l'indirizzo di un nodo speciale, evitando l'inondazione di pacchetti di controllo e dati da parte di ogni sorgente di gruppi diversi.

Con PUMA viene eliminata la necessità di utilizzare un protocollo di unicast e della pre-assegnazione dei core per i gruppi multicast. Questo protocollo implementa un algoritmo distribuito per selezionare un nodo core tra i ricevitori di un gruppo multicast, e per informare i nodi router di almeno un salto dal rispettivo core di ciascun gruppo. Con questo algoritmo vengono creati più percorsi tra un nodo router e un nodo core, relativamente alla distanza che c'è tra i due nodi. Ogni ricevitore si collega ad un nodo core tramite tutti i percorsi più brevi che si vengono a creare tra questi due nodi. L'insieme di tutti di tutti i percorsi che si vengono a creare tra core e nodi ricevitori collettivamente costituiscono la mesh. Ogni mittente invierà pacchetti dati su tutti i percorsi più brevi che ci sono tra sorgente e core. Quando un pacchetto dati raggiunge un nodo membro della mesh, questo verrà inoltrato all'interno della mesh, dove ogni nodo manterrà una cache con i relativi packet ID per poter scartare i duplicati.

PUMA utilizza un singolo messaggio di controllo per realizzare tutte queste funzioni, il multicast announcements. Ogni messaggio di annuncio è composto da:

- un numero di sequenza,
- un indirizzo del gruppo (group ID),
- un indirizzo del core (core ID),
- la distanza dal nodo core,
- un flag relativo ad un membro mesh che viene settato quando un nodo mittente appartiene alla mesh,
- e un *nodo parent* che seleziona un nodo vicino preferito per raggiungere il core.

Il successivo multicast announcement avrà un sequence number più alto rispetto al precedente inviato dal core stesso. Con questi messaggi i nodi selezionano i core, i percorsi relativi tra i mittenti esterni ai gruppi per inoltrare pacchetti dati multicast all'interno dei gruppi stessi, per notificare gli altri nodi quando un nodo vuole entrare a far parte di una rete mesh o vuole lasciarla.

5.3 Metodo proposto

L'obiettivo del metodo proposto è il miglioramento della robustezza e la riduzione della latenza end-to-end di un servizio multicast. Sfruttando la codifica a fontana, è possibile effettuare la decodifica di un file senza sapere quali pacchetti sono stati ricevuti, ma solamente la quantità; infatti è possibile ricostruire il file originale dopo aver collezionato $K(1 + \varepsilon)$ pacchetti diversi. Sfruttando questa proprietà si è pensato che se un nodo ricevesse pacchetti non da una sola sorgente, ma da più sorgenti contemporaneamente, una volta raggiunta la quantità di $K(1 + \varepsilon)$ pacchetti, allora potrebbe iniziare a decodificare prima che le sorgenti finiscano di inviare tutto il flusso, diminuendo così latenza end-to-end complessiva. Allo stesso tempo, sia sfruttando le capacità di recupero delle perdite dei codici LT, sia sfruttando il fatto di non avere un solo nodo trasmettente, il metodo proposto aumenta la robustezza del sistema, permettendo di raggiungere degli alti livelli di qualità dello stream trasmesso. La diminuzione della latenza è possibile se i nodi che trasmettono i pacchetti ne cambiano l'ordine, permettendo così al nodo ricevente di iniziare effettivamente la decodifica prima della trasmissione di tutto il flusso.

Vengono considerati quindi degli scenari MANET in cui c'è una fonte di stream multimediale, nodi in movimento che costituiscono la rete di trasmissione, dei nodi

di destinazione intermedi che collezionano i pacchetti e li rilanciano e un numero di ricevitori finali; i nodi intermedi per la trasmissione sfruttano il protocollo PUMA, quindi ogni nodo che deve trasmettere o ritrasmettere invia i pacchetti di dati attraverso il percorsi più brevi esistenti. L'intera rete può essere vista come una serie di diverse sotto-reti con nodi di destinazione intermedi che hanno il compito di collezionare dai nodi trasmettenti ad un-hop $K(1+\varepsilon)$ pacchetti, ovvero la quantità di pacchetti utili per la decodifica, di decodificare il flusso informativo e ricodificare il flusso originale, cambiando l'ordine dei pacchetti codificati, ritrasmettendo infine i pacchetti nella sotto-rete successiva.
Dato che i nodi di destinazione intermedi prendono pacchetti da più di un nodo, in ordine differente, sono in grado di raccogliere la quantità sufficiente di pacchetti prima della fine della durata della trasmissione.

5.4 Algoritmo di *scrambling* proposto

Al fine di diminuire i tempi di ricezione dei pacchetti da parte dei nodi destinati intermedi e dei nodi destinatari, si è pensato di effettuare lo *scrambling* a livello pacchetto dello stream informativo. Così facendo il nodo che colleziona i pacchetti per effettuare la codifica può prendere, per esempio, i primi pacchetti da un nodo e contemporaneamente gli ultimi da un altro.

Inoltre lo scrambling dei pacchetti viene spesso utilizzato per proteggere lo stream da punto di vista della sicurezza. Infatti scambiando l'ordine dei pacchetti, con bassissimo costo computazionale, e non aumentando la latenza nel caso dei codici LT, perchè bisogna in ogni caso avere tutti i $K(1+\varepsilon)$ pacchetti prima di decodificare, si ottiene contemporaneamente anche un livello di sicurezza. Infatti un eventuale attaccante non può leggere lo stream in chiaro, ma deve in

ogni caso riordinare i pacchetti intuendo l'algoritmo utilizzato per lo scrambling ed effettuare la decodifica. Inoltre anche l'attaccante necessita di $K(1+\varepsilon)$ pacchetti per poter leggero lo stream. Così facendo l'attaccante, anche se riuscisse a riordinare e ricostruire il flusso, impiegherebbe tempo per farlo ed il tempo è un fattore chiave nei servizi real-time, infatti il troppo tempo impiegato per l'attacco potrebbe portare ad una inutilità dell'attacco stesso.

L'algoritmo di scrambling creato nel dominio spaziale nasce dall'algoritmo realizzato nell'articolo [34]. Per realizzare tale algoritmo vengono scelti due numeri contigui della sequenza generale di Fibonacci, G_n e G_{n+1} e viene applicata la seguente trasformazione:

$$S_k = (kG_n + \xi) \bmod G_{n+1}, \xi \in 0, 1, 2, ..., B-1 \tag{5.11}$$

permutando così la sequenza originale, ovvero l'ordine di pacchetti.
La modifica apportata al metodo nativo è stata la sostituzione della sequenza generale di Fibonacci con le *p-r-sequenze* di Fibonacci che sono state create. Tramite l'utilizzo di queste sequenze ci sono due parametri *p,r* che fungono da chiavi, senza le quali non è possibile stabile quale *p-r-sequenza* di Fibonacci è stata utilizzata.

5.4.1 Sequenze di Fibonacci

Le *p-r-sequenza* di Fibonacci, [21], [22], possono rappresentare un numero non negativo intero, ma, a differenza della classica sequenza binaria, ci sono più rappresentazioni di uno stesso numero. Per avere una rappresentazione univoca di un numero bisogna rispettare il *teorema di Zeckendorf*. Questo teorema pone un vincolo: la rappresentazione di un numero non può avere meno di $p-1$ '0' tra

due '1', inoltre non possono esserci più di r gruppi consecutivi dati da un '1' seguito da un numero di '0' pari a p. Se questa regola viene rispettata, allora la rappresentazione del numero è univoca. Inoltre è necessario che gli addendi coinvolti nella creazione di un nuovo numero della sequenza non siano più due, ma un numero maggiore. è il parametro r che indica quanti elementi vengono coinvolti nella somma.

Le *p-r-sequenze* di Fibonacci sono definite come segue:

$$F_{p,r}(n) = \begin{cases} 0, & n < 0 \\ 1, & n = 0 \\ \sum_{j=0}^{r} F_{p,r}(n-1-j\cdot p), & n > 0 \end{cases} \tag{5.12}$$

Tali sequenze possono rappresentare in maniera differente uno stesso numero. Per far sì che la rappresentazione sia univoca, in essa non possono esserci più di r gruppi consecutivi dati da un '1' seguito da un numero di '0' pari a 'p'.

5.5 Risultati sperimentali

La rete considerata, basata su standard IEEE 802.11, è composta da tre sotto-livelli. L'intera rete è la somma di quattro sotto-reti, ciascuna di sedici o di otto nodi che si muovono in modo casuale ad una velocità di 5m/s, secondo il ''Modello Waypoint Random''. In riferimento alla figura 5.8, il nodo 'S' è la sorgente iniziale che codifica il flusso sorgente e lo trasmette i pacchetti codificati nella prima rete. I nodi 'N1', 'N2' e 'N3' sono i primi nodi destinatari intermedi che tentano di eseguire la decodifica e rilanciano i pacchetti ri-codificati nella seconda sotto-rete. Stessa cosa fanno i nodi 'N4', 'N5', 'N6' e 'N7' per due sotto-reti differenti. I nodi 'D1', 'D2', 'D3' e 'D4' sono invece i nodi destinatari finali, a cui è affidato il compito di decodificare l'intero stream informativo, senza rilanciarlo.

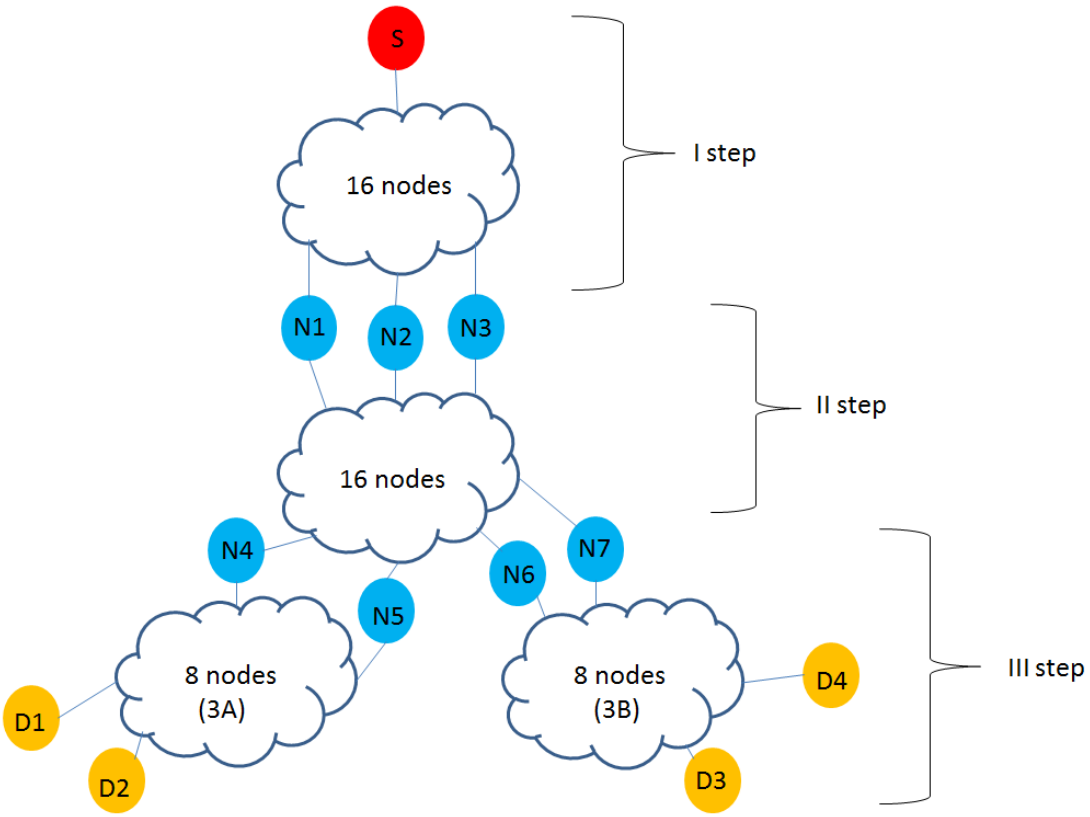


Figura 5.8: Schema di rete utilizzato nella sperimentazioni del metodo proposto

Per valutare il metodo proposto sono stati considerati due scenari di rete:

- il primo è il caso ideale in cui si suppone che non ci sia perdita di pacchetti,
- il secondo è affetto da perdita di pacchetti e ogni sotto-rete ha un diverso rate di perdita di pacchetti (PLR); le perdite considerate sono perdite random, perdite quelle dovute alle collisioni MAC, e perdite dovute ai link failure;

Di seguito è stato considerato un numero di pacchetti di 10000 pacchetti trasmessi sempre su una rete soggetta a diversi tassi di perdita di pacchetti. Infine sono state valutate le prestazioni nel caso in cui si sfrutti la tecnica di scrambling proposta rispetto al caso in cui i pacchetti vengano mandati in ordine.

Nella prima parte della rete, la sorgente codifica il flusso dati originale utilizzando i codici LT ottimizzati per un file di piccole dimensioni, ovvero un numero ridotto di pacchetti pari a 1000. La velocità di trasmissione è uguale a 10pkt/s. Dopo avere effettuato la codifica il nodo sorgente trasmette 1000 pacchetti più un overhead di 200 pacchetti che rappresenta il 20% del pacchetti utili. I pacchetti di dati vengono instradati tra nodi mobili fino a raggiungere nodi intermedi N1, N2 ed N3. A questo punto, nodi intermedi riceventi decodificano il flusso originale non appena ricevono un numero di pacchetti pari a $K(1+\varepsilon)$. Il nodo sorgente trasmette i pacchetti nella rete senza rimescolare l'ordine; in questo modo la trasmissione è comunque robusta alle perdite, ma la latenza totale non diminuisce.
Nella seconda fase, i nodi intermedi destinatari diventano le fonti della seconda sotto-rete, quindi codificano il flusso dati precedentemente decodificato, con l'overhead più adatto alle condizione della rete in cui vanno a trasmettere. In questo caso i pacchetti vengono inviati attraverso la rete in ordine differente

tramite l'uso delle *p-r-sequenze* di Fibonacci. Ogni flusso è una permutazione differente, dato che viene generato da tre chiavi differenti. Questa tecnica è stata applicata anche nel passo successivo per i nodi N4, N5, N6 e N7 quando diventano le fonti delle altre due sotto-reti, ma con diversi valori dei parametri della *p-r-sequenza* di Fibonacci.
I pacchetti inviati dalle nuove sorgenti è caratterizzato dal 25% di overhead; i nodi N4, N5, N6 e N7 ricevono il flusso in un tempo minore, decodificano, ricodificano e ritrasmettono delle ultime sottoreti.

La terza fase è caratterizzata da due differenti sotto-reti. N4 e nodi N5 condividono la stessa matrice di codifica e una diversa matrice di codifica è condivisa da N6 e N7. N4, N5 inviano pacchetti di dati con un overhead maggiore del 19% all'interno di una sotto-rete più piccola, composta da otto nodi. I destinatari finali sono D1 e D2 con il compito solo di decodificare lo stream dati. Allo stesso modo, N6 e N7 invieranno, sfruttando una diversa *p-r-sequenza* di Fibonacci, il flusso di dati nella sotto-rete con destinatati finali a D3 e D4.

Per le simulazioni è stato utilizzato il software NS2 [1] su un computer portatile con processore Intel Core2, CPU T5200 con una frequenza di clock di $1,66$ Hz e 1GB di RAM. Per tutte le simulazioni la decodifica parte una volta che sono stati raccolti 1100 pacchetti, ovvero 10% di overhead.
Nella prima fase, tempo di codifica della sorgente per 1200 pacchetti è di 17.03 secondi, la trasmissione dura 120 secondi e non sono state considerate perdite per collisione a livello di MAC o altre tipologie di perdite.

	N1	N2	N3
Tempi di decodifica (sec.)	5.78	5.79	5.80

Tabella 5.1: Tempi di decodifica nel primo step di rete

	N1	N2	N3
Tempi di codifica (sec.)	18.83	18.83	18.83

Tabella 5.2: Tempi di codifica nel secondo step di rete

5.5.1 Scenario senza l'introduzione di perdite in rete

Per ogni step di rete i tempi di arrivo di $K(1+\varepsilon) = 1100$ pacchetti, quindi $\varepsilon = 0.1$ sono state valutate confrontando il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato; tutti i risultati sono illustrati nelle figure 5.9, 5.10 e 5.11.

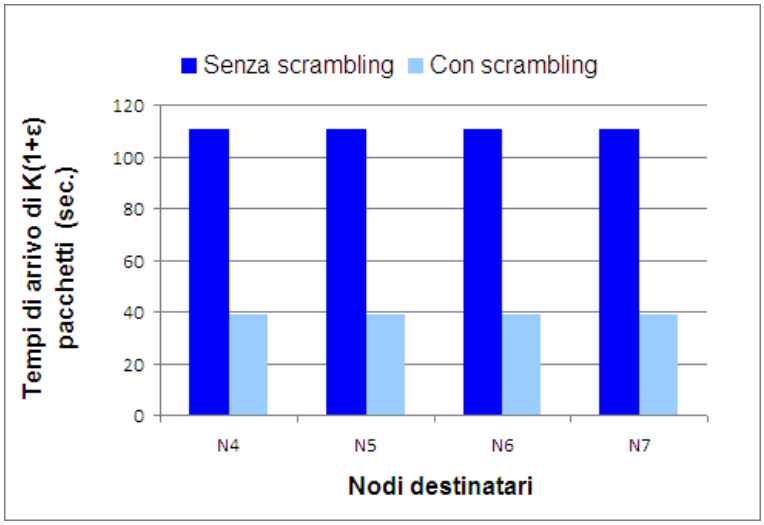


Figura 5.9: Confronto tra i tempi di arrivo di $K(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 2".

è stato valutato il tempo di decodifica dei primi 1100 pacchetti per i tre nodi di destinazione (N1, N2 e N3), del primo step e i valori sono stati riportati in 5.1.

Nel secondo step di rete la codifica inserire un overhead pari a 25% e i tempi di codifica sono stati valutato per i nodi destinatari N1, N2, N3 e riportati in 5.2.

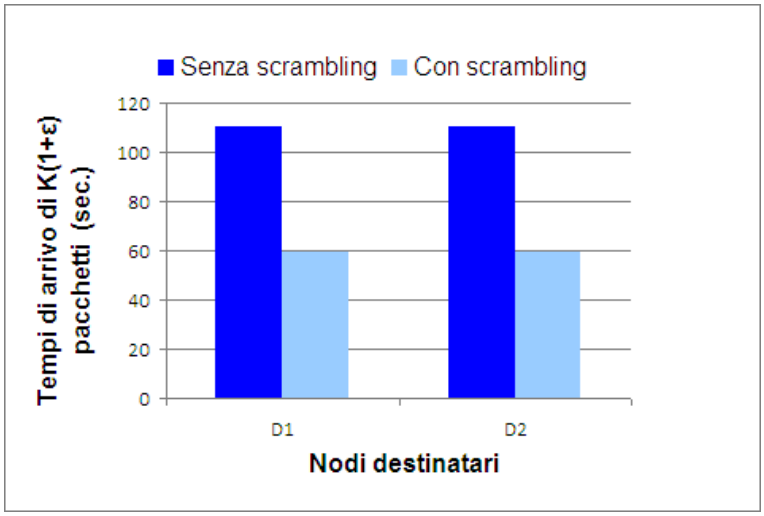


Figura 5.10: Confronto tra i tempi di arrivo di $K(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 3A".

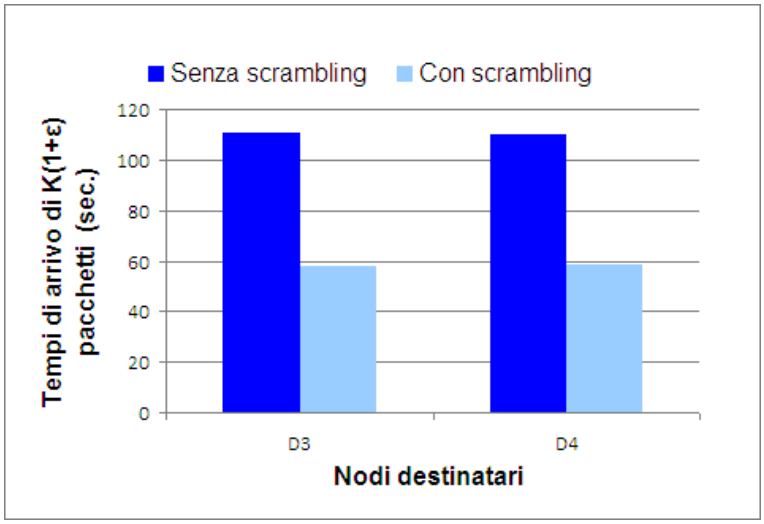


Figura 5.11: Confronto tra i tempi di arrivo di $K(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 3B".

	N4	N5	N6	N7
Tempi di decodifica (sec.)	6.25	6.14	6.00	6.00

Tabella 5.3: Tempi di decodifica nel secondo step di rete

	N4	N5
Tempi di codifica (sec.)	17.12	17.12

Tabella 5.4: Tempi di codifica per lo step di rete 3A

In questa seconda fase, la trasmissione dura 125 secondi, l'1% dei pacchetti viene perso a causa di collisioni MAC e ogni ricevitore riceve pacchetti da tutte le fonti. I tempi di decodifica per i nodi N4, N5, N6 e N7 sono riportati in 5.3.

Si consideri la sotto-rete 3A con D1 e D2 come ricevitori finali. I tempi di codifica di 1190 i pacchetti per N4, N5 nodi sono stati valutati e riportati in 5.4.

La trasmissione dura 119 secondi, l'1% di pkts vengono persi a causa alle collisioni MAC e ogni nodo destinatario prende pacchetti da tutte le fonti. I tempi di decodifica per i nodi D1 e D2 sono riportati in 5.5.

Infine, si considera la sotto-rete con destinatari finali D3 e D4. I tempi di codifica di 1144 pacchetti per i nodi N6 e N7 sono riportati in 5.6.

La trasmissione dura 114 secondi e l'1% di pacchetti si perdono per le collisioni MAC. Ogni nodo destinatario prende pacchetti da tutte le fonti. I tempi di decodifica di 1100 pacchetti per i nodi D1 e D2 sono riportati in 5.7.

5.5.2 Scenario con l'introduzione di perdite in rete

Per il secondo scenario, sono state inserite perdite a rate diversi per ogni sotto-rete. Nella sotto-rete al secondo step, il PLR è pari al 11%; nella sotto-rete 3A il

	D1	D2
Tempi di decodifica (sec.)	5.87	5.79

Tabella 5.5: Tempi di decodifica per lo step di rete 3A

	N6	N7
Tempi di codifica (sec.)	15.92	15.92

Tabella 5.6: Tempi di codifica per lo step di rete 3B

	D3	D4
Tempi di decodifica (sec.)	5.49	5.62

Tabella 5.7: Tempi di decodifica per lo step di rete 3B

PLR è pari a 6% e, infine, nella sotto-rete 3B $PLR = 3\%$. Viene fatto il confronto prestazionale nel caso di utilizzo dello scrambling e il caso di non utilizzo. Tutti i risultati sono illustrati nelle figure 5.12, 5.13 e 5.14.

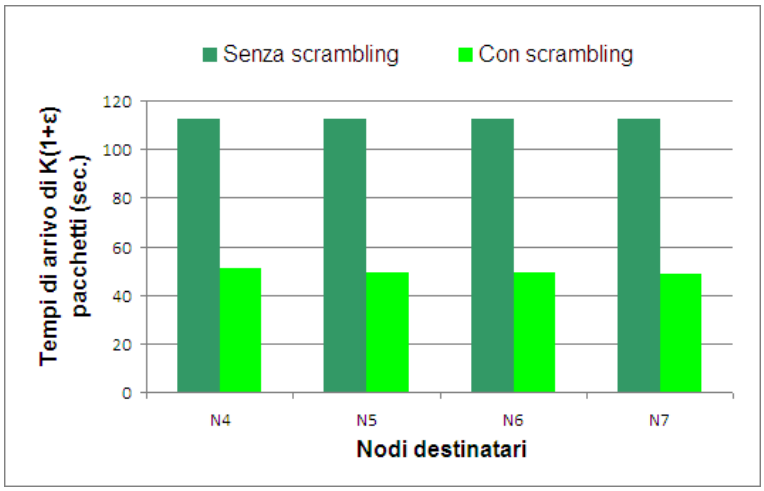


Figura 5.12: Confronto tra i tempi di arrivo di $k(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 2" con perdite inserite.

Nelle figure 5.15, 5.16 e 5.17 sono stati confrontati i tempi di arrivo per diversi tassi di perdita di pacchetti I risultati mostrano che l'applicazione della tecnica di scrambling comporta dei tempi di arrivo dei pacchetti a destinazione ridotti a differenza del multicast convenzionale.

Di seguito è stato considerato il caso di un messaggio da inviare di grande dimensioni; quindi sono stati considerati dei codici LT ottimizzati per un alto numero di pacchetti, ovvero 10000 pacchetti. In questo caso è stata considerata

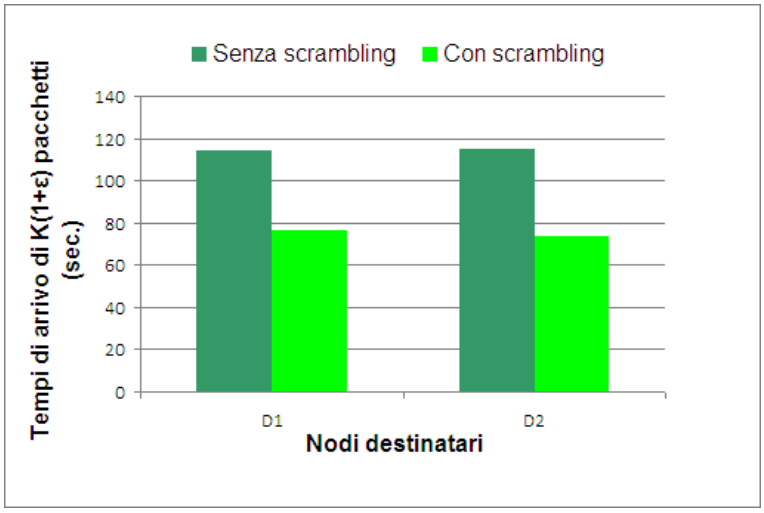


Figura 5.13: Confronto tra i tempi di arrivo di $k(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 3A" con perdite inserite.

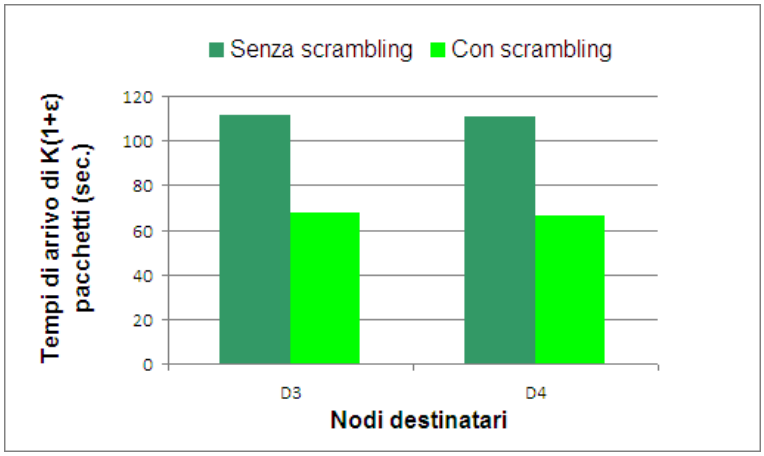


Figura 5.14: Confronto tra i tempi di arrivo di $k(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 3B" con perdite inserite.

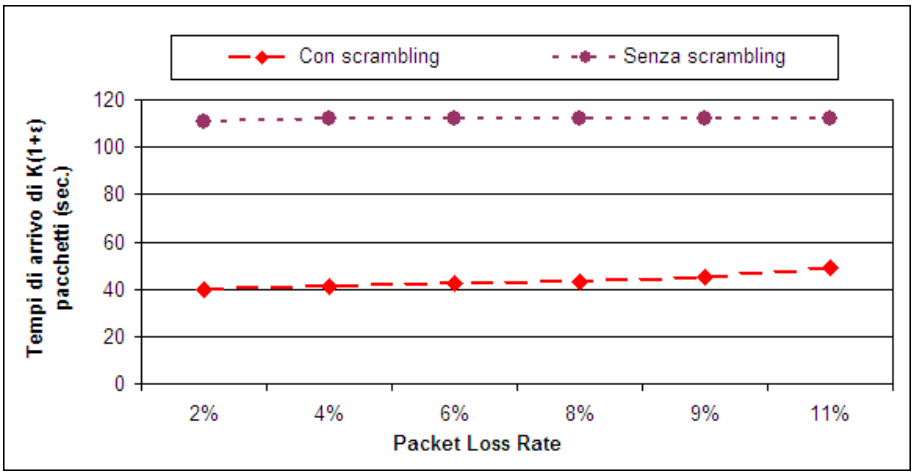


Figura 5.15: Confronto tra i tempi di arrivo a differenti PLR nella sotto-rete 2.

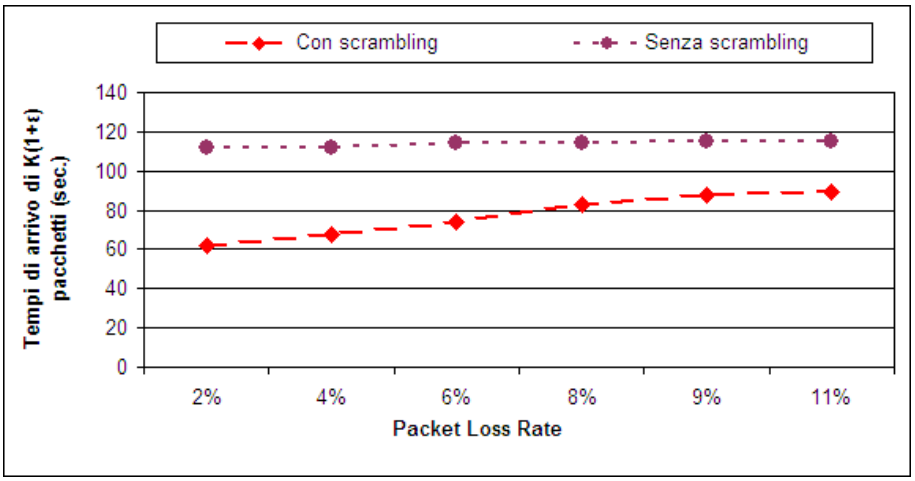


Figura 5.16: Confronto tra i tempi di arrivo a differenti PLR nella sotto-rete 3A.

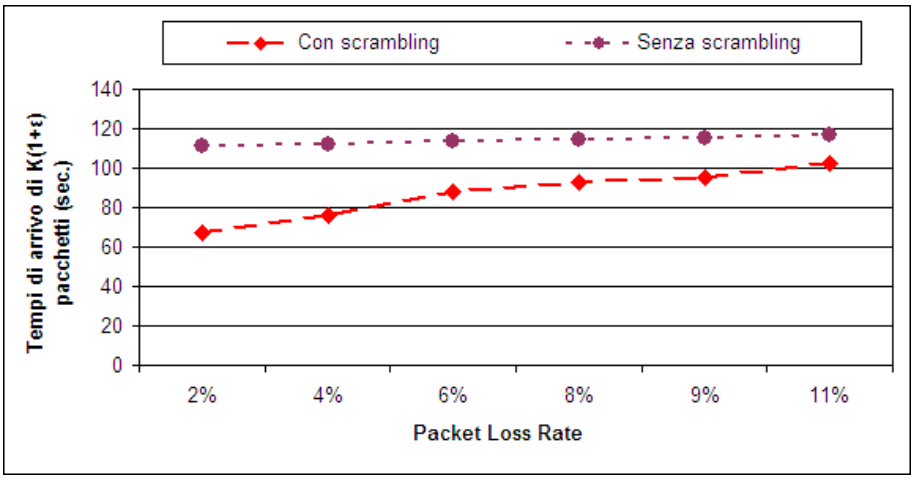


Figura 5.17: Confronto tra i tempi di arrivo a differenti PLR nella sotto-rete 3B.

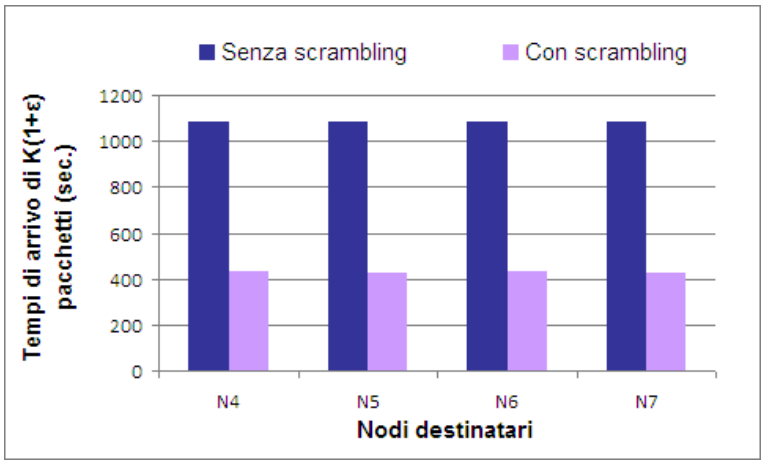


Figura 5.18: Confronto tra i tempi di arrivo di $k(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 2" per un file da trasmette di grandi dimensioni.

la distribuzione di gradi solitona robusta con i parametri c e δ ottimizzati. Nella seconda fase della rete, il PLR è pari a 11%; nello step 3A il PLR è pari a 4%, mentre nello step 3B, $PLR = 3\%$. Per questo caso, la decodifica LT inizia in media una volta che vengono raccolti 10600 pacchetti, quindi $\varepsilon = 0.6$. Sulla base di queste considerazioni, ci si aspettano valori di performance coerenti con i casi precedenti e le simulazioni effettuate hanno confermato le aspettative.
In 5.18, 5.19 and 5.20 viene riportato il confronto prestazionale nel caso di utilizzo dello scrambling e il caso di non utilizzo.

Si è quindi visto che se non si effettua lo scrambling non si riesce ad ottenere un tempo di ricezione minore, tuttavia si è robusti alle perdite per due ragioni principali: in primo luogo, PUMA si basa su un approccio multi-path, quindi il nodo di destinazione riceve i pacchetti da più fonti e la seconda è l'utilizzo della codifica LT che permette in ogni caso il recupero delle perdite. Inoltre si è notato che nella seconda rete composta da 16 nodi, il numero di collisioni è maggiore che non un una rete di 8 nodi, tuttavia la perdita di pacchetti è pari a 1% in entrambi i casi. La ragione sta nel fatto che nella rete con 16 nodi tutti i desti-

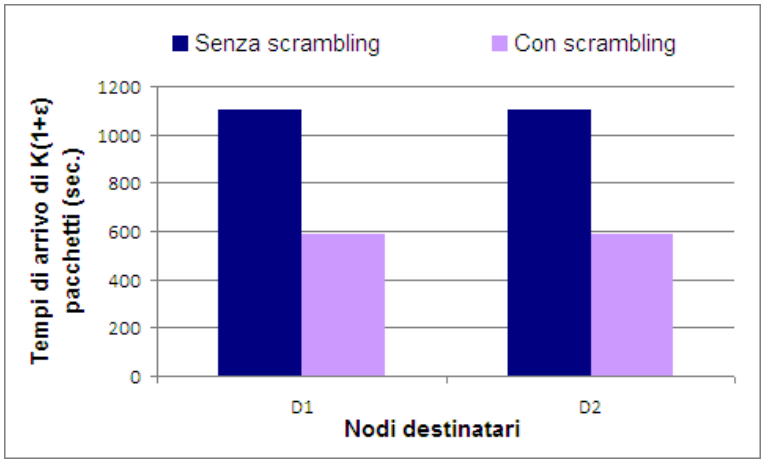


Figura 5.19: Confronto tra i tempi di arrivo di $k(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 3A" per un file da trasmette di grandi dimensioni.

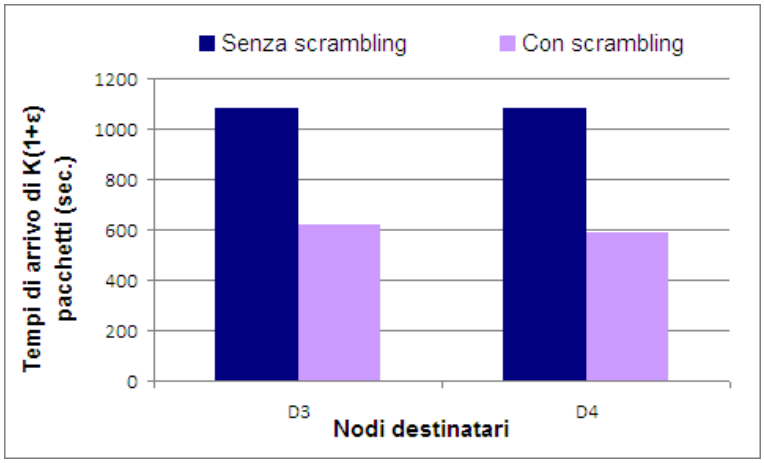


Figura 5.20: Confronto tra i tempi di arrivo di $k(1+\varepsilon)$ pacchetti tra il caso in cui viene eseguita la tecnica di scrambling e il caso in cui lo scrambling non viene considerato per la "sotto-rete 3B" per un file da trasmette di grandi dimensioni.

	N4	N5	N6	N7	D1	D2	D3	D4
Tempi di arrivo di $K(1+\epsilon)$ pacchetti	38.29	38.04	38.05	37.95	57.23	56.9	56.12	55.47

Tabella 5.8: Tempi di arrivo di $K(1+\epsilon)$ pacchetti nel caso di invio di parti differenti del flusso codificato

natari ricevono i pacchetti da tre fonti, invece da solo due sorgenti nel caso della sotto-rete a 8 nodi.

5.6 Procedimento alternativo

In alternativa al procedimento descritto in precedenza che sfrutta lo scrambling dei pacchetti codificati per ottenere tempi di arrivo minori, il flusso sorgente può essere diviso in parti pari al numero di sorgenti che codificano e rilanciano i pacchetti codificati. In questo caso ogni sorgente rilancia solo la parte designata del flusso sorgente. Così facendo la rete viene occupata per un tempo minore, ma la trasmissione è meno robusta in quanto non c'è ridondanza dei pacchetti, ma solo l'overhead creato durante la codifica.
Utilizzando la medesima rete del caso precedente sono stati valutati i tempi di arrivo dei pacchetti e si è visto che in media i tempi si accorciano, ma la riduzione è molto piccola, 5.8.

La piccola riduzione dei tempi è data dal fatto che i nodi delle varie reti non impiegano tempo a scartare i pacchetti duplicati, perchè ogni sorgente manda pacchetti codificati appartenenti ad un pezzo differente del flusso sorgente. Per le prove fatte la decodifica ha successo utilizzando lo stesso overhead dei casi precedenti, sia che si consideri un flusso sorgente di piccole dimensioni, sia che si consideri un flusso sorgente di dimensioni maggiori.

6

Conclusioni

Il presente lavoro di tesi di dottorato ha affrontato la problematica della qualità di un servizio televisivo trasmesso su una rete con perdita, quale è Internet. In una prima parte del lavoro è stata effettuata un'analisi approfondita delle prestazioni della tecnica AL-FEC SMPTE 2022 considerando gli eventi di perdita più frequenti in una rete IP reale. Sono state infatti simulate perdite casuali i.i.d. e perdite a burst, eventi di congestione, link failure e la sovrapposizione di flussi Variable Bit Rate sul flusso Costant Bit Rate che si è esaminato. I risultati ottenuti sono stati validati sul Test bed IP del Ministero dello Sviluppo Economico, al fine di avere un ambiente sperimentale più prossimo possibile ad infrastrutture di rete reali. Dallo studio effettuata si è notato che l'evento di perdita più difficile da contrastare tramite il Forward Error Correction a livello applicativo è la congestione. Per questo motivo è stata sviluppata una strategia per la quale si proteggere il flusso di interesse con una tecnica AL-FEC e si da priorità al solo flusso di protezione. Tale strategia è attuabile se il flusso di protezione è distinto dal flusso dati, come nel caso della tecnica SMPTE 2022 che è un metodo di protezione sistematica. I risultati mostrano che utilizzando la strategia proposta si riescono a recuperare completamente dei Packet Loss Rate maggiori rispetto

al caso in cui non si sfrutti la strategia proposta, utilizzando lo stesso overhead. Inoltre tale strategia consente di garantire un'alta qualità di servizio senza danneggiare gli altri flussi concorrenti, poichè la prioritizzazione di un flusso FEC su un'interfaccia non determina un incremento rilevante nella quantità di pacchetti scartati sugli altri flussi. Valutate le prestazioni dell'AL-FEC dal punto di vista della qualità del servizio, sono state analizzate le prestazioni dell'AL-FEC dal punto di vista della qualità dell'esperienza considerando una metrica oggettiva full-reference ed una no-reference. I risultati derivati dalla metrica full-reference hanno mostrato come l'uso di una tecnica di correzione permetta di avere un livello di qualità più elevato, ma che all'aumentare del Packet Loss Rate ben oltre i limiti di correzione della configurazione scelta dell'AL-FEC il guadagno tende ad annullarsi. La metrica no-reference è stata quindi utilizzata per realizzare un sistema che mantenga un certo livello di qualità percepita variando le configurazioni della tecnica di correzione d'errore. In una seconda parte del lavoro si è considerata una nuova strategia che permettesse un multicast efficace in reti MANET con perdita, sfruttando le caratteristiche di codici maggiormente innovativi, ovvero i codici a fontana. Tale strategia punta ad ottenere un'alta qualità per servizi real-time. Infatti dalle precedenti analisi si è visto quanto sia difficile recuperare le perdite ed avere allo stesso tempo una latenza molto bassa e quindi adatta ai servizi real-time. L'idea è stata quella di far ricevere ai destinatari i pacchetti del flusso d'interesse non da una sola sorgente, ma da più sorgenti contemporaneamente, in maniera tale che questi possano iniziare a decodificare il flusso, una volta raggiunta la quantità necessaria per la ricostruzione, prima che le sorgenti finiscano di inviarlo completamente, diminuendo così la latenza complessiva. I risultati sperimentali hanno dimostrato la validità di tale idea.

Gli sviluppi futuri riguardano l'unione tra i codici a fontana e il *network coding* al fine di raggiungere la più alta affidabilità possibile per uno servizio trasmesso su reti a pacchetto con perdita.

Bibliografia

[1] available at http://www.isi.edu/nsnam/ns/. 98

[2] Transmission control protocol. *IETF RFC 793*, 1981. 14

[3] Tcp selective acknowledgment options. *IETF RFC 2018*, 1996. 15

[4] Ospf version 2. *IETF RFC 2328*, 1998. 38

[5] An rtp payload format for generic forward error correction. *IETF RFC 2733*, 1999. 23

[6] The reliable multicast design space for bulk data transfer. *IETF RFC 2887*, 2000. 16

[7] Enhancing tcp's loss recovery using limited transmit. *IETF RFC 3045*, 2001. 15

[8] A conservative selective acknowledgment (sack)-based loss recovery algorithm for tcp. *IETF RFC 3517*, 2003. 15

[9] Rtp: A transport protocol for real-time applications. *IETF RFC 3550*, 2003. 23

[10] Rtp payload format for mpeg1/mpeg2 video. *IETF RFC 2250*, 2003. 23

[11] Negative-acknowledgment (nack)-oriented reliable multicast (norm) protocol. *IETF RFC 3940*, 2004. 16

[12] Triple-play services quality of experience (qoe) requirements. *Broadband Forum TR-126*, 2006. 7, 53

[13] Audiovisual media services directive. *Direttiva 2007/65/EC del Parlamento Europeo e del Consiglio*, 2007. 5

[14] Forward error correction for real-time video/audio transport over ip networks. *SMPTE 2022-1*, 2007. 23, 48

[15] Unidirectional transport of constant bit rate mpeg-2 transport streams over ip networks. *SMPTE 2022-2*, 2007. 27, 48

[16] Ospf for ipv6. *IETF RFC 5340*, 2008. 38

[17] Digital video broadcasting (dvb), transport of mpeg 2 ts based dvb services over ip based networks. *ETSI TS 102 034 v.1.4.1*, 2009. 40, 53

[18] Traffic management mechanisms for the support of iptv services. *Draft ITU-T Recommendation Y.IPTV-TM (TD 122 - WP 4/13)*, 2010. 47

[19] Pompei S.; Rea L.; Matera F.; Valenti A. Experimental investigation on optical gigabit ethernet network reliability for high-definition iptv services. *OSA J. Of Optical Networking*, 2008. 41

[20] A. Neri; M. Carli; M. Montenovo; F. Comi. No reference quality assessment of internet multimedia services. *Proc. Eusipco*, 2006. 59, 62

[21] E. Mammi; F. Battisti; M. Carli; A. Neri; K. Egiazarian. A novel spatial data hiding scheme based on generalized fibonacci sequences. *Proceedings of SPIE Electronic Imaging Science and Technology*, 2008. 94

[22] E. Mammi; F. Battisti; M. Carli; A. Neri; K. Egiazarian. Substitutive steganography in the generalized fibonacci domain. *Proc. SPIE International Conference on Electronic Imaging 2009, Image Processing: Algorithms and Systems VII*, 2009. 94

[23] J. C. Moreira; P. G. Farrell. Essentials of error-control coding. *Ed. J. Wiley and Sons, Ltd*, 2006. 73

[24] R. Vaishampayan; J.J. Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. *IEEE International Conference on Mobile Ad-hoc and Sensor System*, 2004. 89

[25] O. S. Badarneh; M. Kadoch. Multicast routing protocols in mobile ad hoc networks: A comparative survey and taxonomy. *EURASIP Journ. On Wireless Telecommunications and Networking*, 2009. 89

[26] C. Liu; J. Kaiser. A survey of mobile ad hoc network routing protocols. *Report Series, Nr 2003-08, University of Ulm. Tech*, 2005. 88

[27] D.J.C. MacKay. Fountain codes. *IEE Proceedings-Communications*, 2005. 18, 73

[28] P. Cataldi; M.P. Shatarski; M. Grangetto; E. Magli. Implementation and performance evaluation of lt and raptor codes for multimedia applications. *Intelligent Information Hiding and Multimedia Signal Processing*, 2006. 82

[29] A.Valenti; P. Bolletta; S. Pompei; F. Matera. Experimental investigations on restoration techniques in a wide area gigabit ethernet optical test-bed based on virtual private lan service. *ICTON 09*, 2009. 43

[30] E. Mammi; G. Russo; A. Neri. Evaluation of al-fec performance for ip television services qos. *Proc. SPIE International Conference Image Quality and System Performance VII*, 2010. 23

[31] S. Wolf; M. Pinson. Video quality measurement techniques. *NTIA Report 02-392*, 2002. 59, 60

[32] E. Mammi; G. Russo; P.Talone. Television over ip overview. *EUVIP 2010, 2nd European Workshop on Visual Information Processing*, 2010. 5

[33] E. Mammi; G. Russo; P.Talone. Prioritization of al-fec information for improving ip television services qos. *SPIE Electronic Imaging Science and Technology*, 2011. 48

[34] J. Zou; R. K. Ward; D. Qi. The generalized fibonacci transformations and application to image scrambling. *Proceeding of the IEEE international conference on Acoustic, speech and signal processing*, 2004. 94

[35] E. Mammi; S. Pompei; A. Valenti; G. Russo; D. Milanesio; V. Sardella. Valutazione sperimentale delle prestazioni di servizi televisivi su reti ottiche gbe di tipo unmanaged e managed. *Atti di FOTONICA 2010*, 2010. 40

[36] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656*, 1948. 10

[37] A. Shokrollahi. Raptor codes. *Technical report, Laboratoire d'algorithmique, Ecole Polytechnique Federale de Lausanne*, 2003. 85

[38] E. Hyytia; T. Tirronen; J. Virtamo. Optimizing the degree distribution of lt codes with an importance sampling approach. *RESIM, 6th International Workshop on Rare Event Simulation*, 2006. 84

[39] Y. Wang; Q. Zhu. Error control and concealment for video communication: a review. *Proceedings of the IEEE, vol. 86, pp. 974-997*, 1998. 7, 19