



Universita' Degli Studi "La Sapienza" Roma
Facolta' di Ingegneria
Dipartimento di Elettronica

Tesi di Laurea

**Accesso multiplo di sorgenti di traffico con requisiti di
Qualità del Servizio in WLAN centralizzate.**

Candidato

Marialuisa Assunta Polino
Matricola 09087425

Relatore

Prof.ssa Maria Gabriella Di Benedetto

Correlatore

Ing. Guerino Giancola

Anno Accademico 2001/2002

Introduzione	Errore. Il segnalibro non è definito.
CAPITOLO 1: WIRELESS LOCAL AREA NETWORKS: W-LANs	Errore. Il segnalibro non è definito.
1.1 WIRELESS LOCAL AREA NETWORKS	Errore. Il segnalibro non è definito.
1.1.1 Applicazioni delle reti Wireless LAN	Errore. Il segnalibro non è definito.
1.1.2 Funzionamento di una Wireless LAN	Errore. Il segnalibro non è definito.
1.2 STANDARD ESISTENTI	Errore. Il segnalibro non è definito.
1.2.1 Lo standard ETSI HiperLAN II	Errore. Il segnalibro non è definito.
1.2.2 Lo standard IEEE 802.11	Errore. Il segnalibro non è definito.
CAPITOLO 2: MODULAZIONE E TECNICHE DI ACCESSO MULTIPLA AL MEZZO	Errore. Il segnalibro non è definito.
2.1 OFDM	Errore. Il segnalibro non è definito.
2.2 STRATEGIE DI ACCESSO MULTIPLA	Errore. Il segnalibro non è definito.
2.2.1 FDMA	Errore. Il segnalibro non è definito.
2.2.2 TDMA	Errore. Il segnalibro non è definito.
2.2.3 CDMA	Errore. Il segnalibro non è definito.
2.3 COMBINAZIONE DELLA MODULAZIONE OFDM CON STRATEGIE DI ACCESSO MULTIPLA	Errore. Il segnalibro non è definito.
2.3.1 OFDM-TDMA	Errore. Il segnalibro non è definito.
2.3.2 OFDM-FDMA	Errore. Il segnalibro non è definito.
2.3.3 OFDM-CDMA	Errore. Il segnalibro non è definito.
2.3.4 OFDMA	Errore. Il segnalibro non è definito.
2.4 OFDM combinato con strategie di accesso multiplo: confronti, vantaggi e svantaggi	Errore. Il segnalibro non è definito.
CAPITOLO 3: QUALITA' DEL SERVIZIO: QoS	Errore. Il segnalibro non è definito.
3.1 QUALITA' del SERVIZIO: INTRODUZIONE	Errore. Il segnalibro non è definito.
3.2 Fasi del meccanismo di QoS	Errore. Il segnalibro non è definito.
3.2.1 Specifica	Errore. Il segnalibro non è definito.
3.2.2 Negoziazione	Errore. Il segnalibro non è definito.
3.2.3 Mappatura	Errore. Il segnalibro non è definito.
3.2.4 Controllo e Gestione	Errore. Il segnalibro non è definito.
3.3 Modello Integrated Service: IntServ	Errore. Il segnalibro non è definito.
3.4 Classi di traffico e classi di servizio nel modello IntServ	Errore. Il segnalibro non è definito.
3.4.1 La classe di servizio Best-effort	Errore. Il segnalibro non è definito.
3.4.2 La classe di servizio Guaranteed Bandwidth	Errore. Il segnalibro non è definito.
3.4.3 Supporto della QoS	Errore. Il segnalibro non è definito.
3.4.4 Allocazione delle risorse secondo il modello Dual Leaky Bucket	Errore. Il segnalibro non è definito.
3.4.5 Controlled Load Service	Errore. Il segnalibro non è definito.
3.5 Resource reservation Protocol:RSVP	Errore. Il segnalibro non è definito.
3.5.1 Messaggi del protocollo RSVP	Errore. Il segnalibro non è definito.
CAPITOLO 4: MODELLI DI TRAFFICO	Errore. Il segnalibro non è definito.
4.1 INTRODUZIONE	Errore. Il segnalibro non è definito.
4.2 CARATTERIZZAZIONE E MODELLIZZAZIONE DELLE SORGENTI DI TRAFFICO	Errore. Il segnalibro non è definito.
4.2.1 Caratterizzazione delle sorgenti	Errore. Il segnalibro non è definito.
4.2.2 Modelli di Traffico	Errore. Il segnalibro non è definito.
4.3 Traffico vocale	Errore. Il segnalibro non è definito.
4.4 Traffico Video	Errore. Il segnalibro non è definito.
4.5 Traffico dati	Errore. Il segnalibro non è definito.

4.5.1	Traffico WEB.....	Errore. Il segnalibro non è definito.
4.5.2	Traffico FTP.....	Errore. Il segnalibro non è definito.
4.5.3	Traffico Smtp.....	Errore. Il segnalibro non è definito.
CAPITOLO 5: MODELLI SIMULATIVI DELLE SORGENTI DI TRAFFICO		
5.1	Introduzione.....	Errore. Il segnalibro non è definito.
5.2	Sorgenti On-Off.....	Errore. Il segnalibro non è definito.
5.3	Sorgenti HTTP.....	Errore. Il segnalibro non è definito.
5.4	Sorgenti FTP.....	Errore. Il segnalibro non è definito.
5.5	Sorgenti SMTP.....	Errore. Il segnalibro non è definito.
5.6	Sorgenti video MPEG.....	Errore. Il segnalibro non è definito.
CAPITOLO 6: DIMENSIONAMENTO DEI BUFFER		
6.1	Introduzione _____	Errore. Il segnalibro non è definito.
6.1.1	Teoria delle Code_____	Errore. Il segnalibro non è definito.
6.1.2	Caratteristiche prestazionali e Traffico in un Sistema di Servizio _	Errore. Il segnalibro non è definito.
6.1.3	Code con ingresso e servizio a distribuzione esponenziale	Errore. Il segnalibro non è definito.
6.1.4	Code con tempi servizio a distribuzione generale _	Errore. Il segnalibro non è definito.
6.2	Simulazioni: descrizione e obiettivi _____	Errore. Il segnalibro non è definito.
6.3	Considerazioni sulla validità delle simulazioni	Errore. Il segnalibro non è definito.
6.4	Sorgenti FTP _____	Errore. Il segnalibro non è definito.
6.5	Sorgenti On-Off_____	Errore. Il segnalibro non è definito.
6.6	Sorgenti HTTP _____	Errore. Il segnalibro non è definito.
CAPITOLO 7: CONCLUSIONI		
APPENDICE A		
A.1:	Sorgenti On-Off_____	Errore. Il segnalibro non è definito.
A.2:	Sorgenti HTTP _____	Errore. Il segnalibro non è definito.
A.3 :	Sorgenti FTP _____	Errore. Il segnalibro non è definito.
A.4:	Sorgenti SMTP_____	Errore. Il segnalibro non è definito.
A.5:	Sorgenti MPEG _____	Errore. Il segnalibro non è definito.
A.6 :	Sorgente Poissoniana_____	Errore. Il segnalibro non è definito.
A.7 :	Occupazione Buffer _____	Errore. Il segnalibro non è definito.
A.8 :	Ritardo di coda _____	Errore. Il segnalibro non è definito.
BIBLIOGRAFIA		

Introduzione

Il presente lavoro di tesi si pone nell'ambito del progetto di una rete Wireless LAN, che in particolare ha come obiettivo la definizione dello strato MAC. Nell'architettura protocollare ipotizzata si è assunto direttamente al di sopra dello strato MAC la presenza dello strato IP.

Il sistema considerato deve essere in grado di supportare il trasferimento di flussi di dati con diverse caratteristiche in termini di traffico e requisiti di qualità del servizio. Lo scopo sarà quello di garantire un trattamento differenziato alle applicazioni con diversi requisiti, in termini di banda e/o ritardo di trasmissione, e quindi sarà necessario ottenere un sistema capace di fornire differenti livelli di qualità del servizio.

Questo lavoro di tesi in particolare si occupa dell'interfacciamento dello strato MAC con lo strato di rete. Data la struttura protocollare assunta e le caratteristiche desiderate per il sistema, si è reso necessario fornire una descrizione delle differenti sorgenti di traffico in termini di volume di traffico offerto alla rete e dei requisiti di Qualità del Servizio, QoS, da queste richieste.

Il problema di progetto che si presenta in generale può essere così espresso: date le richieste di servizio e i requisiti di QoS delle diverse applicazioni, occorre determinare le risorse di rete necessarie al conseguimento degli obiettivi prefissati.

Per tale scopo occorrono dei modelli matematici per :

- la caratterizzazione delle richieste di servizio, che corrisponde alla definizione del traffico nella rete ;
- la descrizione dell'interazione tra attività e risorse, basata sulla teoria delle code;
- individuazione di metodi per garantire la qualità del servizio.

L'obiettivo principale del presente lavoro è stato quello di considerare dei modelli di traffico delle sorgenti, presenti in letteratura, ottenuti da misure di traffico reali piuttosto che utilizzare i classici modelli di sorgenti proposti in letteratura.

In tal modo si ottiene uno scenario più verosimile con maggiore validità come strumento di progetto nel dimensionamento delle risorse di rete e nella valutazione delle prestazioni raggiungibili.

Il lavoro dopo un'introduzione generale riguardante le caratteristiche principali di una Wireless LAN, insieme alle applicazioni, alle tecnologie alla base delle stesse e agli standard esistenti, HIPERLAN 2 e IEEE 802.11, riporta una breve descrizione della tecnica di modulazione, OFDM, e delle tecniche di accesso al mezzo, OFDM-CDMA e OFDM-TDMA, utilizzate nel progetto.

Sono poi descritte le caratteristiche principali dei meccanismi in grado di supportare la Qualità del Servizio, con riferimento alla letteratura presente per lo strato IP.

Segue quindi la parte centrale che come già detto ha come obiettivo la caratterizzazione e modellizzazione delle sorgenti di traffico, a partire da misure di traffico reali.

Infine a partire dai dati ottenuti con tali modelli sono state effettuate delle simulazioni per verificare l'andamento del traffico offerto alla rete dalle diverse applicazioni e l'occupazione dello spazio di memoria richiesto.

Lo scopo è quello di fornire, a partire dai modelli di traffico realistici considerati, uno strumento che consenta di valutare l'impatto del carico di traffico offerto e delle richieste di QoS delle applicazioni sul dimensionamento delle risorse di rete necessarie.

CAPITOLO 1:

WIRELESS LOCAL AREA NETWORKS: W-LANs

1.1 WIRELESS LOCAL AREA NETWORKS

Le reti LAN, Local Area Networks, ossia le reti in area locale, sono state sviluppate allo scopo di permettere l'interconnessione, in ambienti quali, ad esempio, uffici o università, di apparecchiature terminali, in generale PC, (Personal Computer), con archivi di dati (database) e periferiche costose (stampanti di qualità, dispositivi grafici,..) che per ragioni di efficienza di gestione e/o di costo rimangono centralizzate. In tal modo il singolo utente può reperire le informazioni necessarie al proprio lavoro e utilizzare i dispositivi centralizzati. Le apparecchiature terminali nella terminologia utilizzata per le LAN sono indicate come stazioni [8].

Le reti Wireless LAN, WLAN, sono un tipo di reti locali che utilizza, per la trasmissione, dei dati onde radio ad alta frequenza piuttosto che fili (wire).

Le reti WLAN possono essere utilizzate come estensione, o a volte, come alternativa ad una LAN di tipo wired all'interno di un edificio o in un ambiente campus, consentendo la trasmissione dei dati senza dover sostenere i costi del cablaggio della rete.

Caratteristica distintiva, delle reti LAN in generale, è la limitata estensione geografica, che comprende un edificio o un gruppo di edifici tra loro vicini (ad esempio la rete di un'università o di un'azienda).

Quando la collocazione delle stazioni non varia in modo sensibile e la mobilità è limitata si parla di ambiente *"in building tethered"*. Tale soluzione è in generale quella utilizzata nei vecchi edifici dove è spesso difficile e, comunque costoso, installare nuove reti cablate. In un ambiente *"in building non tethered"* vengono invece sfruttate al meglio le potenzialità di una LAN e gli utenti possono

spostarsi liberamente all'interno degli edifici. Quando una rete LAN può gestire più edifici all'interno di un'area limitata si parla di ambiente “*campus*”.

La mobilità è una delle principali caratteristiche di una Wireless LAN, gli utenti di una WLAN possono ricevere e trasmettere dati in tempo reale ovunque nella zona di copertura, senza dover cercare un posto dove inserire la spina. Vengono così garantite produttività e opportunità di servizi non ottenibili con le reti di tipo wired.

Una WLAN presenta anche altri benefici cui di seguito si accenna brevemente.

La velocità e la semplicità di installazione rendono più semplice l'installazione di una WLAN ed eliminano la necessità di far passare cavi tra i muri, con un evidente guadagno per il risparmio sul cablaggio.

La flessibilità di installazione consente, inoltre, alla tecnologia wireless di far giungere la rete dove i tradizionali cavi non possono arrivare.

La scalabilità consente ai sistemi wireless la configurazione in diverse topologie e il soddisfacimento delle necessità di specifiche applicazioni. Le configurazioni sono facilmente modificabili al variare del numero di utenti.

Inoltre mentre i costi richiesti dall'investimento iniziale per la parte hardware di una WLAN, possono essere più alti di quelli di una LAN cablata, le spese di installazione e le spese nel ciclo di vita possono essere significativamente più basse.

A fronte di questi vantaggi vanno però ricordati alcuni problemi, in particolare :

Non esiste uno standard accettato a livello globale;

L'utilizzo delle frequenze, senza licenza (unlicensed), ISM (Industrial, Scientific e Medical) potrebbe determinare problemi di interferenza [1].

1.1.1 Applicazioni delle reti Wireless LAN

Le reti WLAN come detto accrescono le potenzialità delle wired LAN, esse costituiscono il mezzo più agile e snello per gestire il collegamento cosiddetto dell'ultimo miglio, permettendo l'interfacciamento tra la tradizionale infrastruttura di rete fisica, PSTN (Public Switched Telephone Network) e gli utenti mobili in building o in ambiente campus.

Di seguito sono riportati alcuni esempi di possibili applicazioni di Wireless LAN :

- In ambito ospedaliero, il personale può comunicare in modo efficiente scambiandosi informazioni sui pazienti in tempo reale attraverso calcolatori portatili o palmari collegati tra loro mediante una WLAN;
- Il personale di aziende e strutture pubbliche o private può accedere ai database da qualsiasi postazione nella rete e in ogni momento, accrescendo in tal modo il rendimento.
- Nelle aziende gli amministratori di rete minimizzano gli overhead di gestione dovuti a spostamenti, estensioni di rete o altro tipo di cambiamento, si ha quindi una gestione più facile delle reti evitando la creazione di strutture costose.
- Nelle attività commerciali con lo scambio di informazioni, attraverso dispositivi portatili, tra personale lavorativo e database centrale, si velocizza e accresce la produttività.
- Dirigenti di aziende in sede di conferenze prendono decisioni più rapide grazie alle informazioni real-time che ottengono tramite la WLAN.
- Nei vari ambienti quindi lo scambio di informazioni real-time, tra gli utenti di una WLAN, aumenta l'efficienza e la produttività degli stessi, accrescendo i vantaggi, già citati, derivanti dalla riduzione dei costi di gestione e realizzazione.

Nella figura 1.1 è riportato un esempio di una rete WLAN in ambiente aziendale.



Fig.1.1 Esempio di WLAN in ambiente aziendale

1.1.2 Funzionamento di una Wireless LAN

Una rete WLAN utilizza le onde elettromagnetiche (radio o infrarossi) per il trasferimento dell'informazione da un punto ad un altro, senza il supporto di una connessione fisica. Ogni sistema radio a banda stretta trasmette e riceve informazioni ad una certa frequenza radio: i dati sono modulati su portanti radio in modo da poter esser estratti correttamente in ricezione. La presenza di portanti diverse consente l'utilizzo della rete da parte di più utenti contemporaneamente, e l'accurata sintonizzazione al lato ricezione elimina il problema dell'interferenza. Il ricevitore, per estrarre i dati, si sintonizza sulla frequenza del relativo segnale, rigettando i segnali radio a frequenze diverse.

Tipicamente nella configurazione di una WLAN è presente un dispositivo, ricetrasmittitore, *transreceiver*, detto *Access Point (AP)*, che si connette alla rete cablata con dei cavi standard Ethernet. Esso deve ricevere, bufferizzare e trasmettere dati tra la WLAN e l'infrastruttura della rete fissa.

Gli Access Point possono essere di due tipi:

Dedicated Hardware Access Points (HAP) : sono dei dispositivi che si connettono direttamente alla rete cablata e fungono da interfaccia tra questa e i dispositivi wireless della propria area di copertura.

La figura 1.2 mostra la configurazione di una WLAN in cui è presente un HAP.

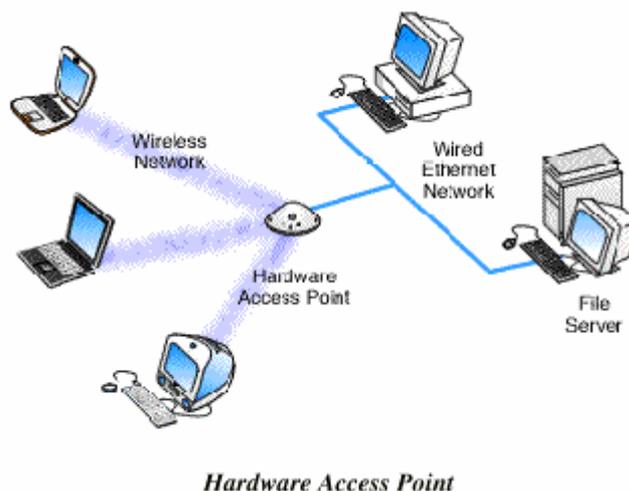


Fig.1.2 Hardware Access Point

Software Access Point (SAP) : mentre un HAP prevede solo la possibilità di collegare la rete Ethernet cablata con la WLAN, con un SAP è possibile implementare estensioni wireless per reti con differente topologia (Token Ring, Token Bus..).

Alcuni produttori hanno implementato dei software da far girare su terminali dotati di tutte le interfacce di rete necessarie, in grado di estrapolare i dati dai pacchetti dei protocolli di rete non supportati e trasformarli in pacchetti per il protocollo wireless. Un esempio di configurazione di una WLAN che prevede un SAP è illustrato nella figura 1.3.

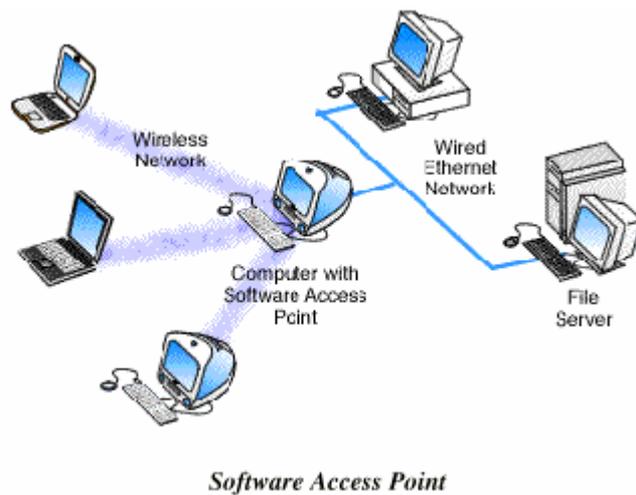


Fig. 1.3 Software Access Point

Un singolo AP può supportare un piccolo gruppo di utenti, con una zona di copertura che varia tra i 30 e i 150 metri. L' Access Point, o l'antenna connessa a questo, è posto in alto o comunque in posizione tale da raggiungere la copertura radio desiderata. Gli utenti accedono alla WLAN tramite degli appositi adattatori, che possono essere implementati ad esempio come schede PCMCIA per notebooks.

Tali adattatori forniscono l'interfaccia tra il sistema operativo di rete e le onde elettromagnetiche per mezzo dell'antenna.

La più semplice configurazione di una rete WLAN, è quella detta “*peer to peer*” o “*ad hoc*”, e consiste in una WLAN indipendente che connette un insieme di PC con adattatori wireless. In ogni momento due o più adattatori wireless si trovano nella zona di copertura di ogni altro e possono organizzare una rete indipendente. Queste reti tipicamente non richiedono alcuna amministrazione o configurazione. Ogni utente può avere accesso alle risorse di ogni altro utente, ma non ad un sistema centralizzato. Le stazioni comunicano direttamente tra loro senza la necessità di installare alcuna infrastruttura. Unico svantaggio di tale configurazione è l'area di copertura limitata. Si può ovviare a ciò con l'utilizzo di access points che fungono da ripetitori estendendo la zona di copertura della WLAN. Nella figura 1.4 è riportato un esempio di tale tipo di

rete WLAN.

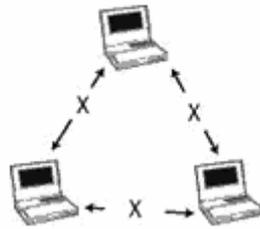


Fig.1.4 WLAN indipendente o “ad hoc” (peer-to-peer)

Nelle WLAN “*ad infrastruttura*” più access points collegano la rete wireless all’infrastruttura fissa e permettono agli utenti di condividere efficientemente le risorse di rete. Più access points possono provvedere alla copertura di aree che si estendono per un intero edificio o in un campus.

Le comunicazioni wireless, in generale, sono limitate, a dei limiti massimi di distanza raggiungibili dai segnali trasmessi, a causa delle basse potenze in gioco. In una WLAN, in particolare, il raggio della singola *microcella*, cioè della zona di copertura di un access point, è notevolmente limitato, per cui si sovrappongono le microcelle singole per garantire una certa continuità della comunicazione all’interno della rete ed evitare i problemi dovuti ad un hand-off poco fluido. La configurazione di una WLAN ad infrastruttura è illustrata in figura 1.5.

1.2 STANDARD ESISTENTI

Sebbene le reti LAN siano state originariamente definite e realizzate in ambienti privati, e quindi basate su architetture e protocolli di tipo proprietario, varie organizzazioni internazionali hanno provveduto ad emettere norme per una definizione univoca dei protocolli relativi ai vari strati funzionali di una rete LAN. Come già detto però, riguardo alle wireless LAN in particolare, al momento non esiste uno standard accettato a livello globale.

Esistono diversi standard proposti per l'implementazione di wireless LAN ad alta velocità fra cui: ETSI HiperLAN II, la famiglia delle norme IEEE 802.11, il Bluetooth e l'HomeRF.

I più diffusi sono quello proposto dalla comunità di studi europea, EC, HiperLAN II, e quello della comunità americana, IEEE, 802.11.

Fra gli standard minori si ricorda il Bluetooth, uno standard industriale, proposto dalla Ericsson, che è caratterizzato da un raggio di copertura limitato, dell'ordine della decina di metri e lavora intorno alla frequenza di 2.4GHz, con velocità di 1Mb/s.

Lo standard HomeRF, sul cui progetto stanno lavorando diverse società, quali Motorola, Siemens ed altre, è stato concepito per ambienti domestici. Dovrebbe essere utilizzato per connettere i computers portatili tra loro, i cordless, gli elettrodomestici, in modo da poterne gestire il funzionamento a distanza. Anch'esso lavora intorno alla frequenza di 2.4 GHz ,con velocità di 1Mb/s.

1.2.1 Lo standard ETSI HiperLAN II

Lo standard HiperLAN, (High Performance Radio Local Area Network), è stato sviluppato dall'ETSI (European Telecommunications Standards Institute) nell'ambito del progetto BRAN (Broad Radio Access Network) [3]. In realtà si tratta di una famiglia di standards, e cioè HiperLAN I/ II, HiperLink progettato per i backbone all'interno di edifici, HiperAccess per l'uso fisso all'esterno per fornire l'accesso ad una struttura cablata.

L'architettura HiperLAN II consente la connessione con vari tipi di infrastrutture di rete, tra cui Ethernet, ATM, IP. La banda di frequenza utilizzata è di 5 GHz, con una velocità che può raggiungere fino ai 54 Mb/s, in realtà la velocità in aria è di 54 Mb/s, mentre il throughput delle applicazioni è intorno ai 20 Mb/s. Il range di copertura è di 30 metri in ambienti indoor, di

150 metri in quelli outdoor, con una mobilità media. Un'altra caratteristica importante è che tale standard garantisce il supporto della Qualità del Servizio, QoS, (Quality of Service) [4].

Lo standard HiperLAN II definisce i tre seguenti strati del modello OSI:

- Convergence Layer che permette di adattare la rete allo strato DLC;
- Data Link Control (DLC);
- Physical Layer (PL) per il trasporto dei dati.

Sono considerate le due tipologie di rete: “ad infrastruttura” e “ad hoc”.

Lo strato DLC è di tipo *Connection Oriented*, la trasmissione dei dati avviene dopo che sia stata stabilita una connessione temporanea tra il Radio Terminal, RT, interessato nella comunicazione e l'access point. E' proprio questo meccanismo che consente la negoziazione dei parametri di QoS, come ritardo e larghezza di banda. La tecnica di modulazione utilizzata è l'OFDM (Orthogonal Frequency Division Multiplexing), molto efficace nel combattere l'interferenza intersimbolica, causata dai ritardi variabili, che le diverse componenti del segnale radio possono sperimentare in un ambiente dispersivo nel tempo. Le componenti del segnale, infatti per giungere a destinazione possono seguire cammini diversi (multipath), accumulando ritardi, che date l'alta velocità, possono costituire una porzione significativa delle stesse componenti.

Lo standard è caratterizzato dall'allocazione automatica della frequenza; l'AP ha un sistema per la selezione automatica del canale radio più appropriato in base alle risorse disponibili e ai parametri di QoS negoziati. Riguardo la sicurezza e la protezione HiperLAN II supporta, dapprima l'autenticazione, cioè si richiede l'autorizzazione di accesso alla rete, e successivamente "l'encryption", ossia dopo essere stati identificati si ha la protezione del messaggio.

Altra caratteristica dello standard è la *mobilità* durante la comunicazione, cioè un RT potrebbe accorgersi che un altro AP può garantire un migliore rapporto

segnale-rumore, (SNR), per cui ordina un hand-over e sposta la connessione sul nuovo AP, rischiando però in questo modo di perdere dei pacchetti.

Segue una descrizione degli strati definiti dallo standard HiperLAN II.

Lo strato fisico ha le seguenti caratteristiche:

- La trasmissione è “a burst”, costituiti da un preambolo e da una parte utile;
- La tecnica di modulazione è l'OFDM;
- Ogni canale è suddiviso in 52 portanti, 48 per i dati e 4 pilota per permettere la sincronizzazione, i canali sono spazati di 20 MHz, con un intervallo di guardia di 800 ns;
- Le bande utilizzate sono due: 5150-5350 GHz(8 canali) e 5470-5725 GHz (12 canali).

Lo strato DLC è costituito da tre sottostrati:

Il sottostrato MAC (Medium Access Control), è il protocollo che regola l'accesso multiplo al collegamento radio utilizzato per la trasmissione da parte dei diversi RT.

Il controllo è centralizzato nell' AP che comunica ai RT il momento in cui possono trasmettere i loro dati. Per permettere a più utenti di accedere alla risorsa condivisa viene utilizzata la tecnica TDMA (Time Division Multiple Access).

I canali di trasporto sono:

- Broadcast Channel (BCH): Usato in downlink e contiene le informazioni di controllo spedite nel “mac-frame”.
- Frame Control Channel (FCH): Contiene una informazione esatta di come sono state usate le risorse.
- Access Feedback Channel (ACH): contiene le informazione sui precedenti tentativi di accesso.

- Random Access Channel (RACH): viene usato dai RT per richiedere risorse per il *downlink-phase* ed *uplink-phase* nel mac-frame successivo.

Il “downlink-phase” e l’”uplink-phase” sono treni di PDU di utente (U-PDU) e di controllo (C-PDU).

I canali logici sono:

- Slow Broadcast Channel (SBCH): trasporta informazioni di controllo nel downlink e solamente se necessario.
- Dedicated Control Channel (DCCH): trasporta segnali del sottostrato RLC una volta stabilita la connessione e quindi dopo che il RT ha ottenuto il proprio identificatore (MAC-ID).
- User Data Channel (UDCH): trasporta DLC-PDU per il Convergence Layer (CL).
- Link Control Channel (LCCH): Trasporta informazioni sul controllo d'errore.
- Association Control Channel (ASCH): Trasporta messaggi di richiesta di nuova associazione.

L'associazione tra canali logici e di trasporto non prescinde dal verso della comunicazione, abbiamo due configurazioni differenti a seconda se parliamo di “uplink” o “downlink”.

Nel sottostrato EC (Error Control) viene utilizzato il Selective Repeat ARQ, questa tecnica rivela i bit errati e ne richiede la ritrasmissione, assicurando inoltre la consegna, in ordine, delle User-PDU allo strato CL. Dopo un certo numero di ritrasmissioni andate male, il pacchetto viene scartato e quindi perso. Gli strati superiori provvederanno all'eventuale recupero.

Sottostrato RLC (Radio Link Control), fornisce un servizio di trasporto per le segnalazioni:

- AFC: si occupa dell'associazione e della disassociazione;
- DCC: provvede al controllo di connessione per il canale DLC;
- RRC: gestisce la risorsa radio.

Lo strato CL (Convergence Layer) svolge due funzioni:

- Adattare le richieste degli strati superiori al servizio offerto dallo strato DLC;
- Convertire le dimensioni, variabili o fisse, dei pacchetti degli strati superiori nella dimensione fissa usata nel DLC.

Sono stati definiti 2 tipi di strato CL:

- “Cell based”: usato per l'interconnessione con una rete ATM.
- “Packet based”: usato in una varietà di configurazioni dipendenti dal tipo di rete fissa.

Per quanto riguarda lo “*scheduling*”, il mac-frame viene gestito secondo due algoritmi:

- Round Robin Esaustivo: se ci sono risorse ancora disponibili nel mac-frame, viene servita una nuova connessione. Questo algoritmo è preferito per le “*best effort*”.
- Round Robin Non Esaustivo: non vengono accettate nuove connessioni, ma le risorse disponibili vengono distribuite a chi è già connesso.

1.2.2 Lo standard IEEE 802.11

L'IEEE 802.11 è il gruppo di lavoro che si occupa della definizione dello strato MAC (Medium Access Control) e dello strato fisico delle reti locali di tipo wireless. Il gruppo si è dedicato allo sviluppo di uno standard per sistemi radio e reti operanti nella banda di frequenze unlicensed, ISM, dei 2.4 GHz, con data rates dell'ordine di 1-2 Mb/s. In questa banda c'era, infatti, la disponibilità di più di 80 MHz da poter utilizzare per comunicazioni dati ad alta velocità.

Lo standard non specifica tecnologie o implementazione, ma semplicemente definisce le specifiche per gli strati fisico e MAC per connessioni wireless per stazioni fisse, portatili ed in movimento all'interno di un'area locale (in building o campus) in grado di supportare velocità trasmissive multiple, scelte a seconda dello stato del mezzo e della capacità delle stazioni, e comunque superiori ad 1 Mb/s. Uno degli scopi principali di questo standard è che un singolo MAC possa supportare più livelli fisici, anche se questi fanno uso di tecnologie diverse. Il wireless MAC supporta sia servizi connectionless, sia servizi di tipo isocrono (time-bounded) per il controllo di processi voce e video.

Lo strato fisico definisce le caratteristiche di modulazione e trasmissione delle informazioni, a tale livello sono stati definiti due metodi di trasmissione a radio frequenze, RF, ed uno ad infrarossi.

Gli standards di trasmissione a RF sono:

- Frequency Hopping Spread Spectrum (FHSS)
- Direct Sequence Spread Spectrum (DSSS)

Il metodo ad infrarossi opera nella banda base, mentre i due metodi a RF operano nella banda tra i 2.4GHz e i 2.483GHz.

I sistemi ad infrarossi utilizzano le stesse frequenze usate nelle fibre ottiche e sono caratterizzati dal fatto di “scoprire” solo l'ampiezza del segnale, in tal modo l'interferenza è notevolmente ridotta. Inoltre, essendo sistemi non limitati in banda, possono raggiungere velocità maggiori rispetto ad altri sistemi. La tecnologia IR era inizialmente molto popolare perché caratterizzata da elevati data rates e costi relativamente bassi. Lo svantaggio dei sistemi IR è quello che lo spettro di trasmissione è condiviso con sole ed altre luci fluorescenti. Se c'è molta interferenza da altre sorgenti la LAN può diventare inutile. I sistemi IR richiedono una LOS, line of sight, libera da ostacoli ed inoltre i segnali IR non sono in grado di penetrare oggetti opachi. Ciò vuol dire

che oggetti come muri divisorii, cortine o anche nebbia possono ostruire il segnale.

I sistemi a radio frequenza utilizzano la tecnologia Spread Spectrum Frequency Hopping e la Direct Sequence Spread Spectrum (DSSS). A causa di valori di overhead più elevati rispetto ai sistemi IR, i data rates supportati sono inferiori.

Con la DSSS la trasmissione del segnale è allargata su tutta la banda permessa (per esempio 25MHz). Una sequenza binaria, detta codice di spreading, è usata per modulare il segnale da trasmettere. I bit di informazione sono mappati in una struttura di “chips”, a loro volta mappati a destinazione in un bit. Il numero di chips rappresentanti un bit è il cosiddetto rapporto di spreading. Maggiore è quest'ultimo valore, più il segnale risulterà resistente all'interferenza. Più basso è tale rapporto, maggiore sarà la quantità di banda disponibile all'utente. L'F.C.C. (Federal Communication Commission) ha imposto che il rapporto di spreading deve essere maggiore di 10, lo standard 802.11 ha imposto un valore pari ad 11, trasmettitore e ricevitore devono essere sincronizzati sul medesimo codice di spreading, per consentire una ricezione corretta. L'utilizzo di codici di spreading ortogonali consente a più LAN di condividere la stessa banda.

La tecnica FHSS suddivide la banda in tanti piccoli sottocanali (1 MHz). Il segnale salta allora da un sottocanale all'altro per trasmettere piccoli burst di dati su ogni canale per un certo periodo di tempo, detto “*dwell time*”. La sequenza di hopping deve essere sincronizzata al trasmettitore e al ricevitore, altrimenti l'informazione è persa. L'FCC richiede che la banda sia suddivisa in almeno 75 sottocanali e che il “*dwell time*” non sia maggiore di 400ms. Il FHSS è meno sensibile all'interferenza poiché la frequenza viene continuamente “*shiftata*”. Ciò rende i sistemi FHSS difficili da intercettare, garantendo un elevato grado di sicurezza. Non a caso tali sistemi vengono usati in ambito militare. Se sono usate frequenze di hopping ortogonali molte LAN FHSS

possono essere situate vicino. Poiché i sottocanali sono più piccoli di quelli dei sistemi DSSS, il numero di LAN collocate nella stessa area geografica può essere più grande con i sistemi FHSS.

Nella figura 1.6 è riportato un grafico con le principali caratteristiche dello standard IEEE 802.11.

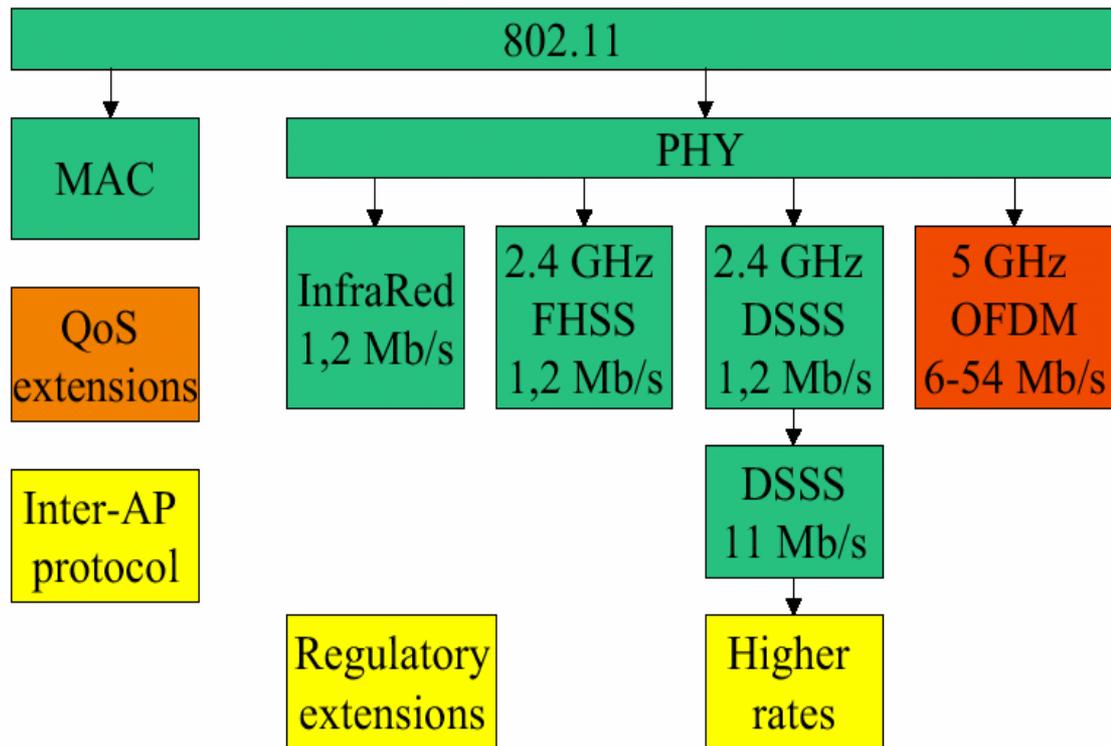


Fig.1.6 Struttura dello standard IEEE 802.11

Lo strato MAC è caratterizzato da un insieme di protocolli responsabili di un utilizzo intelligente della risorsa condivisa. Lo standard 802.11 specifica il protocollo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Esso è utilizzato per la trasmissione asincrona, e può essere affiancato da una funzione di coordinamento centralizzata: PCF (Point Coordination Function), garantendo priorità superiore per i servizi tempo limitati. In questo protocollo, quando una stazione deve trasmettere un pacchetto, per prima cosa deve verificare che nessun'altra stazione stia

trasmettendo, “ascoltando” il mezzo, (Carrier Sense). Se è riconosciuta la presenza di trasmissioni si mette in attesa. Quando il mezzo si libera attende che rimanga tale per un intervallo di tempo minimo: DIFS (Distributed InterFrame Space). Successivamente inizia una fase di contesa (Contention Window): la stazione sceglie un intervallo casuale (backoff) al termine del quale, se il mezzo è ancora libero, la stazione può trasmettere il pacchetto. Poiché la probabilità che due nodi scelgano lo stesso fattore di backoff è piccola, le collisioni tra pacchetti saranno minimizzate. Quando una stazione, in attesa che termini il backoff, “sente” che il mezzo non è più libero, congela il tempo di backoff rimasto. Quando rileva il mezzo libero per un tempo pari a DIFS, non sceglie un nuovo tempo di attesa ma termina il precedente. La rilevazione di una collisione, (Collision Detection), non può essere utilizzata poiché, una stazione che trasmette, non può contemporaneamente “ascoltare” un’altra eventuale stazione in trasmissione nel sistema, dal momento che, il proprio segnale copre qualsiasi altro segnale arrivi alla stazione. Se un pacchetto sta per essere trasmesso, la stazione trasmittente invia prima un pacchetto corto RTS (Ready-To-Send) contenente l’informazione della lunghezza del pacchetto. Se la stazione ricevente ascolta l’RTS, risponde con un altro pacchetto corto CTS (Clear-To-Send). Dopo questo scambio, la stazione trasmittente manda il suo pacchetto. Quando il pacchetto è ricevuto con successo, come determinato dal CRC (Cyclic Redundancy Check), la stazione ricevente trasmette un pacchetto di acknowledgment (ACK).

Nella figura 1.7 è illustrato il meccanismo di scambio di messaggi appena descritto.

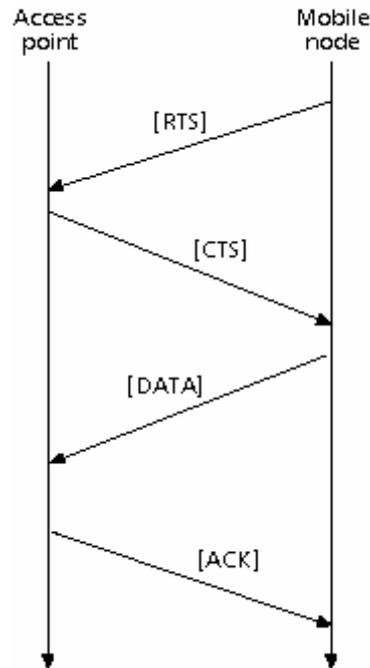


Fig. 1.7 Scambio di messaggi RTS-CTS-DATA-ACK

Questo scambio avanti ed indietro è necessario per evitare il problema del terminale nascosto (hidden node). L' "Hidden Node" è un terminale che si trova al limite tra il terminale di destinazione e il terminale di sorgente. La figura 1.8 illustra questo problema.

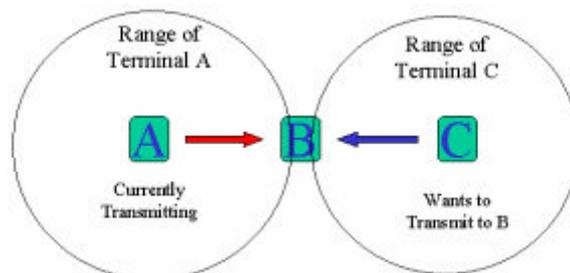


Fig. 1.8 Hidden Node

Il nodo A trasmette al nodo B. Il nodo C non può sentire la trasmissione di A. Durante questa trasmissione, quando C sente il canale, pensa erroneamente che il canale sia "idle". Se, a questo punto, il nodo C inizia una trasmissione verso B, lui stesso interferirà con la ricezione dei dati del nodo B. In questo caso il nodo C è un "hidden node" (terminale nascosto) per A e, come tale, può

causare collisioni sulla trasmissione dei dati. Il meccanismo di Handshake serve a far fronte al problema.

Da chiarire che per evitare che durante i messaggi di protocollo si entri in una nuova fase di contention window, il tempo di attesa per i messaggi di risposta e per l'invio dei dati dopo il CTS è più corto del DIFS, tale tempo è detto SIFS (Short InterFrame Space). Per la bassa affidabilità della trasmissione, una stazione potrebbe ricevere i messaggi e iniziare una trasmissione generando una collisione. Per evitare questo, il protocollo realizza una sorta di “carrier sense virtuale”. I messaggi RTS e CTS contengono informazioni sulla durata della trasmissione successiva, che le stazioni non interessate alla ricezione caricano in un registro detto NAV (Net Allocation Vector). Questo registro viene via via decrementato ed ogni stazione ne attende l'azzeramento prima di cominciare la procedura di trasmissione.

Per quanto riguarda la funzione di coordinamento centralizzata precedentemente accennata, questa può essere gestita solo da alcune stazioni, come gli Access Point. La PCF usa una struttura a superframe in cui si alternano periodi di contesa in cui è attivo il DCF a periodi senza contesa dove è attiva la PCF. Il point coordination diventa padrone del mezzo trasmissivo mediante un accesso prioritario.

CAPITOLO 1: WIRELESS LOCAL AREA NETWORKS: W-LANs	3
1.1 WIRELESS LOCAL AREA NETWORKS	3
1.1.1 Applicazioni delle reti Wireless LAN	5
1.1.2 Funzionamento di una Wireless LAN	6
1.2 STANDARD ESISTENTI	9
1.2.1 Lo standard ETSI HiperLAN II	10
1.2.2 Lo standard IEEE 802.11	14

CAPITOLO 2: MODULAZIONE E TECNICHE DI ACCESSO MULTIPLO AL MEZZO

2.1 OFDM

Il sempre crescente sviluppo delle applicazioni multimediali, in ambiente wireless, si accompagna ad un aumento di richieste di sistemi di comunicazione a banda larga e in grado di supportare richieste di Qualità del servizio (Quality of Service - QoS) [6].

Il problema principale da risolvere in ambiente wireless, diventa dunque quello di riuscire a fornire le velocità di trasmissione necessarie per il soddisfacimento di tali specifiche, soprattutto in relazione al fatto che il canale radio indoor di interesse è affetto dal fenomeno dei cammini multipli (multipath).

Infatti il principale problema nella trasmissione dei segnali radio in generale, e in ambiente indoor in modo particolare, è costituito dall'attenuazione causata dal fenomeno dei cammini multipli, che unitamente alla interferenza intersimbolica e alla selettività in frequenza del canale, rende molto variabile la qualità del collegamento. La propagazione multipath, è dovuta al fatto che il segnale, nel percorso da sorgente a destinazione, subisce molte riflessioni, per cui in ricezione arrivano molte repliche dello stesso in istanti di tempo differenti, si parla di dispersione temporale. Le varie repliche del segnale interagiscono tra di loro causando interferenza intersimbolica (ISI). Il problema della dispersione temporale diventa ancora più rilevante in presenza di trasmissioni ad elevato data -rate, in cui la distanza tra simboli è minore, e in ambiente indoor in cui le riflessioni sono presenti in numero maggiore.

Va inoltre ricordato che un segnale a banda larga sperimenterà anche la selettività in frequenza del canale trasmissivo, per cui alcune sue componenti

spettrali potrebbero essere attenuate o completamente cancellate, con inevitabile distorsione del contenuto informativo del segnale.

Per la soluzione di tale problema si potrebbero utilizzare tecniche di equalizzazione in ricezione, ma che in pratica presentano notevoli difficoltà realizzative nelle applicazioni real-time ad elevato bit rate.

Una delle tecniche per raggiungere trasmissioni ad elevato data-rate in un ambiente mobile è l'OFDM (Orthogonal Frequency Division Multiplexing).

L'OFDM è una tecnica di modulazione a portante multipla che consente di combattere la selettività in frequenza del canale. Il principio base dell'OFDM è quello di dividere un flusso di dati ad elevato bit rate, in un certo numero di flussi con minore bit rate e trasmettere tali dati in parallelo utilizzando differenti sottoportanti [5], [6].

A tale scopo la banda del segnale è suddivisa in più sottocanali su ognuno dei quali vengono trasmessi i suddetti flussi di dati a minore bit rate. Con l'aumentare del numero dei canali di trasmissione paralleli diminuisce il data rate che ogni portante deve trasportare e si allunga la durata di ciascun simbolo, riducendo gli effetti del delay spread, cioè della dispersione temporale del segnale dovuta al multipath e della conseguente interferenza intersimbolica.

L'OFDM inoltre consente di raggiungere un'elevata efficienza spettrale rispetto alle precedenti tecniche di modulazione in frequenza (FDM, Frequency Division Multiplexing), grazie all'utilizzo di portanti ortogonali, che consente la sovrapposizione parziale dello spettro delle sottoportanti, senza pregiudicare la ricostruzione del loro contenuto informativo in ricezione.

Inoltre l'utilizzo delle operazioni di trasformazione di Fourier IFFT/FFT, nelle operazioni di modulazione e demodulazione rispettivamente, riduce la complessità realizzativa di modulatori e demodulatori e aumenta l'efficienza computazionale.

Considerata una banda di W [Hz] ed un data rate di R [bit/s], sia t_s la durata di simbolo, τ il massimo delay spread del canale ed f_d la frequenza Doppler.

Il principio dell'OFDM si basa sul fatto di suddividere la banda totale W in N sottoportanti spaziate di W/N Hz.

Il flusso informativo del segnale viene suddiviso in sottoflussi di rate R/N [bit/s] e trasmesso in parallelo sulle N sottoportanti.

La durata del simbolo OFDM sarà N volte più lunga, rispetto al caso di modulazione con portante singola, consentendo di ridurre l'interferenza intersimbolica (ISI). Tale interferenza può inoltre essere controllata introducendo un intervallo di guardia temporale per ogni simbolo prima della trasmissione. Infatti durante la trasmissione, a causa del multipath del canale, la condizione di perfetta ortogonalità tra le sottoportanti potrebbe venir meno. Se l'intervallo di guardia è più grande del delay spread e se la durata di simbolo OFDM, che è pari a $T_s = Nt_s$, è minore del tempo di coerenza: $1/f_d$, allora il singolo sottocanale apparirà come un canale a fading piatto e quindi con risposta in frequenza piatta. Il passaggio dalla trasmissione seriale a parallelo semplifica anche il problema dell'equalizzazione, perché ogni sottocanale, nell'ipotesi di sottobanda sufficientemente stretta, risulta caratterizzato da attenuazione piatta in banda.

Scegliendo la distanza tra le sottoportanti pari all'inverso della durata di simbolo, si ottengono sottoportanti ortogonali tra loro. In questo modo si riduce il problema dell'interferenza tra sottoportanti (ICI), perché in pratica il massimo dello spettro di ogni sottoportante cade nel punto in cui quello delle altre è nullo, consentendo la sovrapposizione degli spettri e garantendo la corretta ricostruzione dei simboli.

L'utilizzo della trasformata discreta di Fourier, per le operazioni di modulazione e demodulazione consente di mantenere l'ortogonalità tra le portanti.

Infatti, se al lato trasmissione il segnale viene campionato con frequenza pari all'inverso della durata di simbolo, si ottiene per il segnale OFDM un'espressione equivalente a quella della IDFT, con i campioni in frequenza

calcolati in corrispondenza delle frequenze delle sottoportanti. L'utilizzo della IFFT, quindi l'elaborazione numerica del segnale, ha l'ulteriore vantaggio di ridurre il carico computazionale e la complessità hardware.

La IFFT restituisce i campioni del segnale da trasmettere, successivamente i valori risultanti dalla IFFT sono convertiti in dati seriali per la trasmissione (digitale). Le distorsioni del canale possono pregiudicare l'ortogonalità delle portanti, determinando interferenza tra sottoportanti (ICI), e quindi pregiudicare la ricezione corretta dei simboli. Per eliminare tale problema il simbolo OFDM è preceduto da un'estensione periodica del simbolo stesso, che consiste nell'introduzione del cosiddetto prefisso ciclico, cioè ogni simbolo OFDM è preceduto da un'estensione del simbolo stesso. Nel ricevitore viene rimosso l'intervallo di guardia, i dati vengono demodulati mediante la FFT, e infine convertiti da parallelo in serie. Quello che si ottiene è la sequenza di simboli trasmessi moltiplicata per la funzione di trasferimento del canale sulla corrispondente sottoportante. La figura 2.1 mostra lo schema di un sistema OFDM.

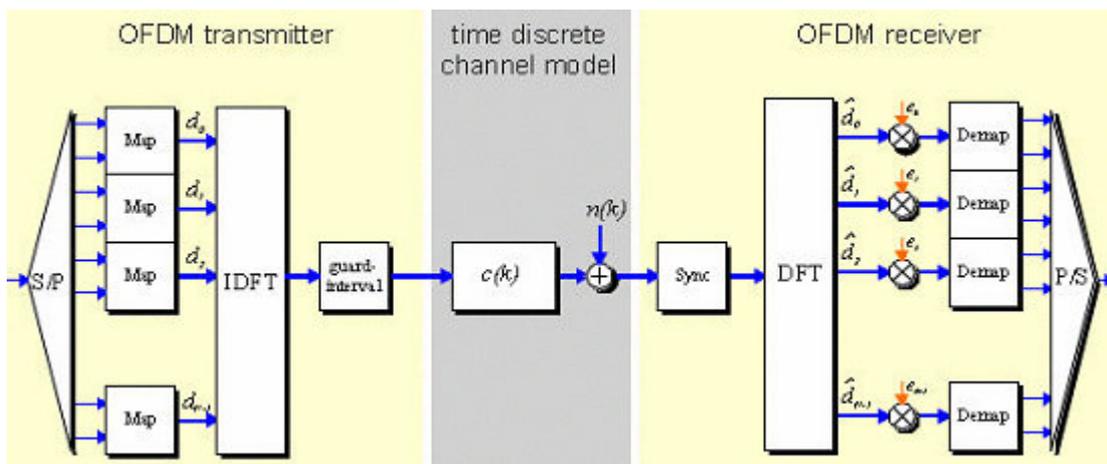


Fig. 2.1 Sistema OFDM

Come già detto un segnale OFDM è ottenuto con un'operazione IFFT e l'aggiunta del prefisso ciclico per eliminare le interferenze ICI e ISI, per un ulteriore miglioramento nella riduzione delle interferenze tra le bande di frequenze adiacenti, si utilizzano delle finestre temporali, come quella a coseno rialzato, che determinano un andamento dello spettro fuori banda rapidamente decrescente.[5]

L'utilizzo di codici per la correzione degli errori, che implica l'aggiunta di bit extra al trasmettitore, rende possibile la correzione di gran parte dei bit ricevuti in modo errato a causa del fading del canale.

Uno svantaggio dell'OFDM è legato alla necessità di sincronizzazione di portante tra trasmettitore e ricevitore, poiché errori di sincronizzazione possono distruggere l'ortogonalità e causare interferenze. L'OFDM è pertanto molto sensibile agli off-set di frequenza dovuti a differenze negli oscillatori locali (usati per la sincronizzazione) e all'effetto Doppler per movimenti dei terminali mobili.

2.2 STRATEGIE DI ACCESSO MULTIPLO

Per consentire la coesistenza di diverse trasmissioni radio nel tempo e nello spazio è nata l'esigenza di gestire la condivisione del mezzo comune e regolarne l'accesso da parte di più utenti.

Le tecniche di modulazione a radiofrequenza consentono a più sorgenti di utilizzare contemporaneamente lo stesso mezzo senza interferire tra loro grazie, all'utilizzo di frequenze portanti diverse.

2.2.1 FDMA

Frequency Division Multiple Access (FDMA) è una tecnica di accesso multiplo utilizzato soprattutto nei sistemi analogici di comunicazione cellulare.

Nel FDMA la banda allocata viene divisa in un certo numero di sottobande e ognuna di queste viene assegnata ad un singolo utente. Si tratta di

un'allocazione statica, infatti ad ogni utente è pre-assegnata individualmente una specifica sottobanda. Poiché ogni utente utilizza una sottobanda diversa, tutti gli utenti possono trasmettere contemporaneamente [8]. La figura 2.2 mostra lo schema di principio della strategia di accesso multiplo FDMA.

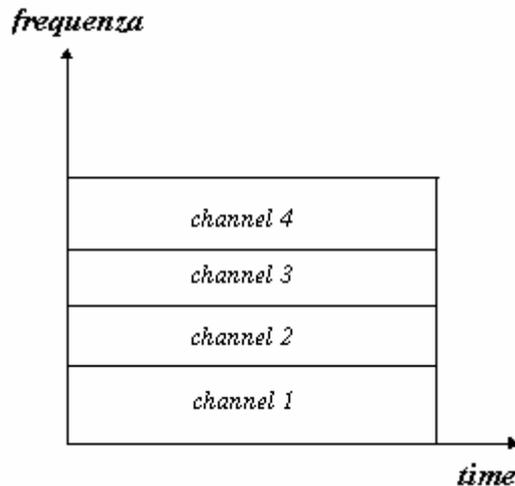


Fig.2.2 FDMA

A lato ricezione un filtraggio di tipo passabanda o eterodina abilita la distinzione di ogni sottocanale mediante demodulazione, unico inconveniente è dato dalla generazione di prodotti di intermodulazione, originati da non linearità, tra le portanti contemporaneamente presenti.

2.2.2 TDMA

La seconda modalità di accesso multiplo con allocazione statica, TDMA (Time Division Multiple Access), si basa sulla suddivisione dell'asse temporale in time-slots, intervalli di tempo di uguale durata, raggruppati in trame, costituite da uno stesso numero di time-slots. L'assegnazione delle risorse avviene su base temporale, ogni utente può accedere al sistema in un fissato time-slot per trama, il canale consiste quindi in uno slot temporale [8]. Un inconveniente del

TDMA è legato dalla necessità di sincronizzazione per le trame in trasmissione e ricezione. Questo non accade invece con la tecnica FDMA, in cui le trasmissioni non sono coordinate nel tempo e pertanto non è necessaria alcuna sincronizzazione tra gli utenti. Nella figura 2.3 è illustrato lo schema del TDMA.

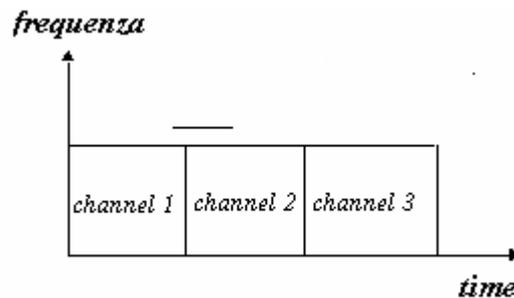


Fig.2.3 TDMA

2.2.3 CDMA

La tecnica di accesso multiplo CDMA (Code Division Multiple Access) consente la trasmissione contemporanea di tutti gli utenti sull'intera banda mediante l'assegnazione ad ogni utente di un codice diverso da quello degli altri. [7]

Al lato ricezione viene operata la correlazione tra il segnale ricevuto e una replica del codice assegnatogli, per cui noto il codice di ogni utente, è possibile discriminare i vari flussi informativi, grazie anche alla proprietà di ortogonalità dei codici, per cui la cross -correlazione tra i codici dei vari utenti è bassa

La banda del segnale di codifica è molto più grande della banda del segnale di informazione, infatti si parla di tecniche di modulazione Spread-Spectrum (cioè a spettro espanso o a dispersione di spettro). I criteri da soddisfare per l'applicazione di tali tecniche di modulazione sono:

- la banda di trasmissione deve essere molto più larga della banda del segnale di informazione;
- la larghezza di banda risultante a RF deve essere statisticamente indipendente dall'informazione trasmessa.

Il rapporto tra la banda del segnale trasmesso: B_t e quella del segnale di informazione: B_i è detto *guadagno di processo* del sistema spread-spectrum: $G_p = B_t/B_i$.

La figura 2.4 mostra lo schema di principio del CDMA:

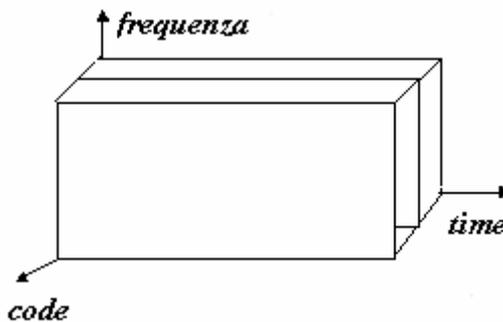


Fig. 2.4 CDMA

Esistono tre tecniche di modulazione che generano segnali spread-spectrum:

- *Direct Sequence (DS) CDMA* : il segnale di informazione è moltiplicato direttamente per uno spreading code ad alto chip rate;
- *Frequency-Hopping (FH) CDMA* : la frequenza portante a cui l'informazione è trasmessa cambia nel tempo in accordo ad uno spreading code;
- *Time-Hopping (TH) CDMA* : l'informazione non è trasmessa in modo continuo, ma, in brevi sequenze, in intervalli di tempo stabiliti per ogni utente dal rispettivo spreading code.

Esistono inoltre tecniche ibride, ottenute dalla combinazione tra le suddette, oppure della CDMA con l'OFDM, la cosiddetta *MC-CDMA (Multi Carrier CDMA)*.

Viene ora riportata una breve descrizione delle tecniche spread-spectrum elencate in precedenza.

Con la tecnica *DS-CDMA* il segnale di informazione è moltiplicato direttamente per un segnale di codice, costituito da un certo numero di bit, chiamati chips, che possono assumere i valori +1 e -1. Naturalmente per ottenere lo spreading del segnale, il chip rate del segnale di codice deve essere molto più grande del simbol rate del segnale di informazione. Viene utilizzata un'unica portante per tutti gli utenti e durante la trasmissione ogni utente utilizza l'intera banda. Oltre a garantire l'accesso multiplo al canale contemporaneamente, l'uso di codici ortogonali consente di limitare l'interferenza da multipath. Una replica del segnale, ritardata per più della durata di un chip, viene interpretata come interferenza, infatti la cross-correlazione, operata in fase di decodifica, dà risultato non nullo solo per il segnale diretto.

Di seguito sono elencati alcuni vantaggi dell'utilizzo del DS-CDMA:

- facile implementazione del segnale codificato, è necessaria una semplice moltiplicazione;
- non è necessario sincronizzare i vari utenti, ma solo trasmettitore e ricevitore per le operazioni di decodifica.

Fra gli svantaggi si ricordano:

- la difficoltà nella sincronizzazione che deve essere mantenuta in una frazione di codice, altrimenti la cross-correlazione può essere nulla
- l'effetto near-far : poiché tutti gli utenti trasmettono in modo continuo sull'intera banda, l'utente più vicino al ricevitore può mascherare quelli più lontani. Tale effetto può essere contenuto con l'uso di algoritmi per il controllo di potenza, in modo che tutti gli utenti siano ricevuti con la stessa potenza media.

Con la tecnica *FH-CDMA* il segnale di informazione è trasmesso, in ogni intervallo di tempo, su portanti diverse. L'insieme dei salti di frequenze è stabilito per ogni utente dallo spreading code relativo. Al ricevitore viene

generata una sequenza di codice, sincronizzata con quella dell'insieme dei salti, in modo da realizzare il corretto despreading del segnale ricevuto.

Se l' hopping rate è maggiore del simbol rate : Fast Frequency Hopping(F-FH), uno stesso simbolo è trasmesso su più portanti. Per il caso opposto : Slow Frequency Hopping (S-FH) sulla stessa portante sono trasmessi più simboli.

Uno dei vantaggi principali è la riduzione dell'effetto near-far rispetto al DS-CDMA, poiché la probabilità che più utenti trasmettano contemporaneamente nella stessa banda di frequenza è piccola. Il problema più complesso è la realizzazione del generatore delle portanti.

Con la tecnica *TH-CDMA* l'asse dei tempi è diviso in trame contenenti un uguale numero di time-slots. Durante ogni trama l'utente trasmette nel time-slot stabilito dal segnale di codice assegnatogli (non necessariamente nello stesso come accade per il TDMA). In questo caso quindi ogni utente utilizza l'intera banda per piccoli intervalli di tempo. La capacità di accesso multiplo è garantita rendendo piccola la probabilità che più trasmissioni avvengano nello stesso intervallo di tempo. Si ottiene ciò assegnando agli utenti codici diversi. Se si verificano trasmissioni multiple nello stesso time-slot, codici di correzione di errore assicurano che il segnale desiderato sia ricostruito correttamente.

Tra i vantaggi del TH-CDMA si ricorda la più facile implementazione rispetto alla FH-CDMA, poiché non richiede il generatore di portante. Come controparte però il ricevitore deve realizzare la sincronizzazione in tempi molto brevi.

Per tutte le forme di CDMA si deve ricordare la capacità di ridurre le interferenze a banda stretta. Infatti, grazie all'allargamento dello spettro ottenuto con la modulazione, la potenza di tali segnali interferenti sarà distribuita su una piccola frazione dello spettro del segnale utile.

2.3 COMBINAZIONE DELLA MODULAZIONE OFDM CON STRATEGIE DI ACCESSO MULTIPLO

2.3.1 OFDM-TDMA

L'OFDM può essere combinato con la tecnica di accesso multiplo TDMA, *OFDM-TDMA*, in questa modalità il simbolo OFDM fissa la durata dello slot, ad ogni utente è assegnato un time-slot a cadenza di trama in cui può trasmettere utilizzando l'intera banda, ossia tutte le sottoportanti. Gli utenti possono essere multiplati in tempo tramite un accesso di tipo random oppure tramite un controllo centralizzato che assegna dinamicamente gli slots. In tal caso, un certo numero di slots sono destinati alla trasmissione broadcast di informazione di controllo, da parte della BS (Base Station). L'assegnazione dinamica degli slots consente di moltiplicare statisticamente il canale, e come in un tipico sistema TDMA, l'interferenza multiutente è limitata all'interferenza dovuta a celle adiacenti che riutilizzano la stessa banda trasmissiva.

Può verificarsi interferenza ICI a causa del non perfetto sincronismo di portante del ricevitore, del rumore di fase degli oscillatori in trasmissione e ricezione e delle variazioni della risposta del canale in generale.

La figura 2.5 illustra la suddivisione del canale radio nell'OFDM-TDMA.

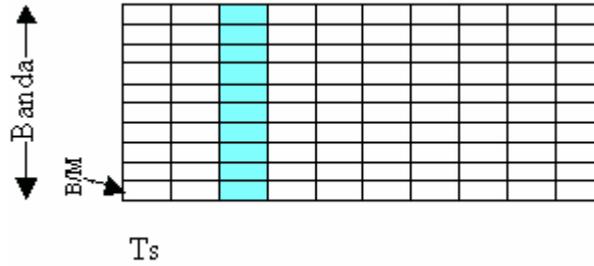


Fig.2.5 OFDM-TDMA

2.3.2 OFDM-FDMA

Nello schema OFDM –FDMA ad ogni utente viene assegnata una sottobanda, che può utilizzare per una durata che comprende più simboli OFDM. La durata del time-slot si allunga e più utenti possono accedere contemporaneamente al mezzo. A causa del non perfetto sincronismo tra le sottoportanti (perdita di ortogonalità) può esserci interferenza multiutente (MUI). L’OFDM-FDMA richiede pertanto l’utilizzo di guardie in frequenza e quindi non tutte le portanti possono essere utilizzate. L’interferenza MUI può inoltre provocare l’effetto near–far, utenti il cui segnale sia ricevuto a livelli di potenza relativamente elevati possono mascherare utenti i cui segnali hanno livelli di potenza minori. La figura 2.6 mostra la suddivisione del canale radio nello schema OFDM-FDMA.

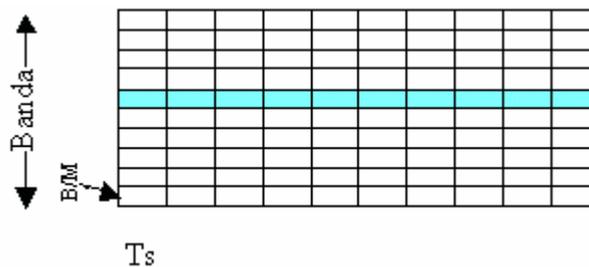


Fig. 2.6 OFDM-FDMA

2.3.3 OFDM-CDMA

La tecnica OFDM –CDMA ovvero Multi Carrier CDMA (MC – CDMA) è ottenuta dalla combinazione della tecnica di modulazione OFDM con la strategia di accesso multiplo CDMA.[7] Tale schema sfrutta le caratteristiche positive dell'OFDM quali la ridotta sensibilità al fading selettivo in frequenza e l'uso efficiente dello spettro, e la ridotta interferenza multiutente tipica del CDMA.

Un trasmettitore MC - CDMA utilizza uno spreading code nel dominio della frequenza, cioè le diverse parti di un simbolo, corrispondenti ad un chip del codice, sono trasmessi su differenti sottoportanti. Quindi lo schema MC - CDMA trasmette lo stesso simbolo, espanso dallo spreading code, in parallelo sulle sottoportanti ortogonali, mentre l'OFDM trasmette simboli diversi sulle diverse sottoportanti. Il procedimento al lato trasmissione è il seguente: la sequenza di dati di informazione in ingresso viene prima eventualmente convertita in una sequenza di dati paralleli lunga P (se il simbol rate è troppo elevato) e ogni uscita del convertitore serie/parallelo viene moltiplicata per lo spreading code di lunghezza K . La sequenza da trasmettere ha quindi un chip-rate che è K volte maggiore quella del simbolo di partenza. Tutti i dati, sono modulati come per l'OFDM mediante la IFFT. In tale schema utenti diversi possono trasmettere contemporaneamente sulle stesse sottoportanti utilizzando codici ortogonali a quelli degli altri. In ricezione è possibile ricostruire il segnale trasmesso mediante le operazioni duali, quindi la demodulazione mediante FFT, la moltiplicazione di ogni portante per il rispettivo elemento di codice e la somma dei termini risultanti.

Tale schema presenta una doppia ortogonalità:

Ortogonalità delle portanti mutuata dall'OFDM

Ortogonalità dei codici propria del CDMA.

di salto. Inoltre con l'uso di sequenze di salti ortogonali per i vari utenti si riducono le interferenze all'interno di una stessa cella.

Mentre nel MC –CDMA tutti gli utenti utilizzano contemporaneamente tutte le portanti, nell'OFDMA tutti utilizzano portanti diverse minimizzando la possibilità di interferenze ICI e ISI all'interno di una stessa cella. Le risorse tempo e frequenza sono allocate in base al tipo di servizio richiesto. Il numero di time-slots e band-slots per utente è variabile per poter trasmettere flussi di dati a rates variabili. L'asse dei tempi è diviso in trame costituite da slots: ogni simbolo OFDM è mappato su uno slot.

2.4 OFDM combinato con strategie di accesso multiplo: confronti, vantaggi e svantaggi

La relativa semplicità ed il controllo dell'interferenza (MUI) garantiti dall'OFDM-TDMA rappresentano il maggiore vantaggio di questa soluzione. Un'accurata progettazione di trasmettitore e ricevitore, infatti, può mitigare gli effetti dell'ICI. Il maggiore problema di questa soluzione è la rigidità della suddivisione delle risorse: il MAC di fatto non utilizza l'OFDM per aumentare la granularità delle risorse. Pertanto, non considerando i vantaggi dell'OFDM come tecnica di modulazione in se, l'OFDM potrebbe essere sostituito da un sistema single carrier con rate di simbolo pari ad M/T_s , con M pari al numero delle sottoportanti dell'OFDM e T_s durata del simbolo, rimanendo il MAC inalterato. Dunque l'OFDM-TDMA è sostanzialmente un sistema TDMA con una specifica tecnica di modulazione e ciò chiaramente fa comprendere che ci sono elementi di OFDM di cui l'OFDMA-TDMA non beneficia e che il MAC può sfruttare meglio per aumentare il throughput e la qualità di servizio.

Simili conclusioni possono essere tratte a proposito dell'OFDM-FDMA. In aggiunta, l'OFDM-FDMA con un'allocazione statica e non ottimizzata delle sottobande dedicate agli utenti è affetto da fading come un sistema di

comunicazione a banda stretta, a meno che non vengano risolti adattativamente complessi problemi di ottimizzazione nell'allocare le sottobande. Una contromisura per il fading può essere l'uso dell'OFDM-FDMA in combinazione con tecniche di Frequency Hopping.

L'OFDM-FDMA è affetto da ICI ma anche da interferenza multiutente (MUI), dovuta all'asincronismo di portante tra i vari utenti.

L'OFDMA è certamente la soluzione più efficiente e flessibile, perché sfrutta ogni tassello della griglia, come una risorsa assegnabile per trasmettere informazione, lasciando la massima flessibilità al MAC. Il MAC può combinare o dividere le risorse in modo da soddisfare al meglio le necessità dell'utenza. Due sono i problemi di OFDMA:

Nello scenario ideale di perfetto sincronismo tra utenti, il fatto che ogni WTIU (Wireless Terminal Interface Units)¹ abbia una diversa funzione di trasferimento del canale fa in modo che ciascun sottocanale ha un diverso rapporto segnale a rumore a seconda dell'utente. Pertanto, il problema di suddividere le risorse in modo da soddisfare le richieste delle WTIU è un complicato problema di ottimizzazione non lineare. Tuttavia l'uso di tecniche di Frequency Hopping può parzialmente compensare questo inconveniente.

L'OFDMA è afflitto da MUI ed ICI, per combattere le quali occorre fare affidamento su una complicazione della struttura del ricevitore notevole e che è in contrasto con la semplificazione del ricevitore a cui dovrebbe portare la scelta dell'OFDM come tecnica di modulazione e multiplexazione.[7]

Rimanendo nello spirito OFDMA ed assumendo che lo strato fisico non sia in grado di separare altrimenti i segnali provenienti da più utenti, la soluzione auspicabile è di non utilizzare alcuni sottocanali, ponendo delle guardie tra utenti.

Il sistema OFDM-CDMA ha due svantaggi rispetto ai precedenti :

¹ WTIU sono i terminali fissi e mobili con interfaccia radio presenti nel sistema considerato

- richiede una complicazione dello strato fisico (lo spreading dei dati);
- se il sistema viene usato senza sovrapporre gli utenti nelle sottobande il sistema è inefficiente; viceversa, se si sovrappongono gli utenti sulle varie sottobande usando codici diversi e non si usano tecniche di joint detection o decorrelation receivers, l'interferenza degrada le prestazioni e richiede tecniche di power control, difficili da far convergere in caso di comunicazioni a pacchetto, [7].

L'OFDM-CDMA è anche affetto da MUI ed ICI, anche se, a parità di condizioni, ne risente in misura minore di entrambi i metodi citati sopra.

Altro vantaggio dell'OFDM-CDMA è che può chiaramente degenerare nei due casi precedenti, pertanto può essere usato in modalità che sono una soluzione di compromesso tra OFDM-TDMA e OFDMA. I nuovi gradi di libertà che OFDM-CDMA aggiunge, possono essere sfruttati per raggiungere buoni compromessi tra l'efficienza e la robustezza della tecnica trasmissiva che, accoppiate ad una efficace strategia MAC, determinano la capacità del sistema di fornire differenti parametri di qualità di servizio.

Un rapido confronto tra questi schemi deve essere fatto in termini di flessibilità e overhead di segnalazione, aspetto di notevole importanza nel dimensionamento della velocità di trasmissione e del ritardo.

Un efficiente schema di accesso dovrebbe garantire una elevata flessibilità nella distribuzione della risorsa disponibile tempo-banda tra tutti gli utenti. Da un lato deve essere considerato il comportamento selettivo in frequenza del canale e dall'altro le richieste degli utenti per data rate differenti (dipendenti dal tipo di servizio richiesto).

I tre schemi mostrano all'incirca lo stesso grado di flessibilità. Il numero delle sottoportanti per utente (OFDM-FDMA), o il numero di time slots (OFDM-TDMA) per utente, o il numero di codici per utente (OFDM-CDMA) possono

essere adottati in accordo alle richieste correnti. L'ultima tecnica presuppone che tutti i codici siano conosciuti ad ogni utente.

Adottando una tecnica di tipo OFDM-TDMA l'accesso multiplo è garantito suddividendo l'asse temporale in trame e time-slots (TS). Ad ogni utente viene assegnato un TS a cadenza di trama e quando è il suo turno, l'utente può trasmettere i propri pacchetti sull'intera banda disponibile (su tutte le sottoportanti), all'interno di una singola trama TDMA, che copre alcuni simboli OFDM. Il numero di simboli OFDM per trama può essere variato, in accordo alle richieste di ogni utente. Le risorse vengono così assegnate in maniera dinamica sulla base delle esigenze di ogni singolo utente ottenendo uno sfruttamento omogeneo della banda da parte di tutti gli utenti, anche se l'impiego della risorsa nel tempo potrebbe non essere ottimale, se il canale è un canale rapidamente variante nel tempo.

Nell'OFDM-FDMA tutti gli utenti accedono contemporaneamente al sistema, ma ogni utente usa un sottoinsieme di sottoportanti che gli viene assegnato in modo esclusivo. Un'allocazione efficiente deve garantire che ogni utente venga servito al meglio. La tecnica OFDM-FDMA può sfruttare il fatto che ogni utente prova un differente canale radio allocando soltanto sottoportanti "buone" con elevato rapporto segnale/rumore e aumentando con ciò la capacità di canale. Questo tipo di tecnica è però anche quella che richiede più overhead di segnalazione poiché deve essere trasmesso il numero di utenti per ogni sottoportante.

Nell'OFDM-CDMA, [7], tutti gli utenti condividono sempre tutta la banda a disposizione (tutte le sottoportanti) usando differenti codici ortogonali. In tal modo vengono sfruttate al massimo le caratteristiche di diversità in frequenza, le proprietà Spread Spectrum e le eventuali variazioni nel tempo del canale. Con tale metodo di accesso viene assegnato un numero di parole di codice tale da poter garantire la risorsa richiesta. L'overhead di segnalazione dell'OFDM-

CDMA è paragonabile a quello del TDMA. La tecnica CDMA è una buona candidata per supportare servizi multimediali, soprattutto in comunicazioni radiomobili, grazie alla sua capacità di adattarsi alla natura asincrona del traffico dati multimediale, per fornire una capacità di trasferimento informativo maggiore di quella offerta da TDMA e FDMA. Se l'allocazione delle risorse è gestita dalla base-station, l'informazione circa l'allocazione sottoportante/codice deve essere trasmessa sia per il downlink che per l'uplink. Dall'analisi o meglio dal confronto di tali sistemi si nota come la tecnica MC-CDMA sia efficiente solo quando siamo in grado di garantire in ricezione l'ortogonalità (o almeno, la bassa correlazione) tra i codici, cosa che nella realizzazione di una tratta Downlink, data la particolarità che ogni utente riceve su un canale separato, è senz'altro assicurata. Per la tratta Uplink come già detto possono esserci problemi determinati dalla possibilità della perdita di ortogonalità dei codici dei vari utenti dovuta alla selettività in frequenza del canale.

Nella figura 2.8 è riportato un confronto tra le tre strategie di accesso multiplo introdotte.

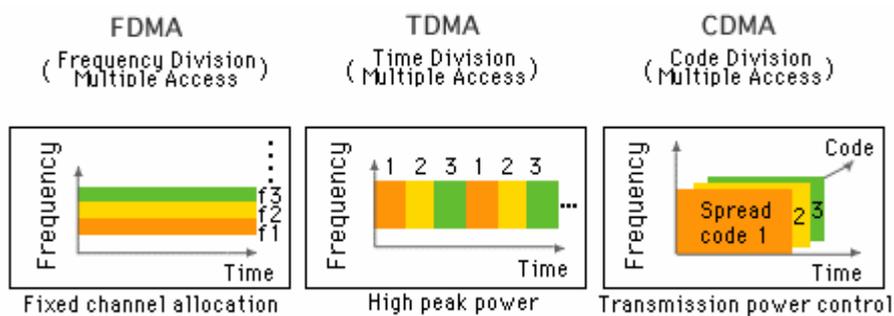


Fig. 2.8 Confronto tra le tecniche di accesso multiplo

CAPITOLO 2: MODULAZIONE E TECNICHE DI ACCESSO MULTIPLA AL MEZZO.....	21
2.1 OFDM	21
2.2 STRATEGIE DI ACCESSO MULTIPLA.....	25
2.2.1 FDMA	25
2.2.2 TDMA.....	26
2.2.3 CDMA	27
2.3 COMBINAZIONE DELLA MODULAZIONE OFDM CON STRATEGIE DI ACCESSO	
MULTIPLA	31
2.3.1 OFDM-TDMA	31
2.3.2 OFDM-FDMA.....	32
2.3.3 OFDM-CDMA.....	33
2.3.4 OFDMA.....	34
2.4 OFDM combinato con strategie di accesso multiplo: confronti, vantaggi e svantaggi.....	35

CAPITOLO 3: QUALITA' DEL SERVIZIO: QoS

3.1 QUALITA' del SERVIZIO: INTRODUZIONE

Nella definizione del sistema in esame, una Wireless LAN di tipo indoor, si è ipotizzata un'architettura protocollare in cui al di sopra dello strato MAC, oggetto principale dello studio, si assume direttamente la presenza dello strato IP. Il sistema considerato deve essere in grado di supportare il trasferimento di flussi di dati con diverse caratteristiche in termini di traffico e requisiti di qualità del servizio. Lo scopo sarà quello di garantire un trattamento differenziato alle applicazioni con diversi requisiti, in termini di banda e/o ritardo di trasmissione, e quindi sarà necessario ottenere un sistema capace di fornire differenti livelli di qualità del servizio.

Nello studio ci si è riferiti al modello di servizi integrati, presente : IntServ (Integrated Service), in grado di fornire servizi a qualità garantita, oltre al servizio best-effort.

Pertanto nel seguito viene riportata la descrizione delle caratteristiche principali dei meccanismi in grado di supportare la qualità del servizio, per la quale si farà riferimento alla letteratura presente sulla trattazione della qualità del servizio di strato IP.

Il servizio offerto dallo strato IP è quello originariamente previsto da Internet, basato sul paradigma Best-Effort, di tipo senza connessione e inaffidabile.

La comunità degli studi riguardanti lo sviluppo di Internet, la IETF, ha condotto esperimenti di trasferimento di voce e video digitalizzati sull'infrastruttura di tipo packet switched, rilevandone l'inadeguatezza alla trasmissione di tali tipi di dati, dovuta agli eccessivi ritardi di coda e perdite per congestione della rete. La struttura di Internet nella sua versione originaria non è pertanto adatta a

supportare le applicazioni real-time caratterizzate da vincoli stringenti in termini di ritardi di trasmissione e throughput.

La crescente evoluzione tecnologica ha portato, tuttavia, un gran numero di nuove applicazioni multimediali con diversi requisiti in termini di throughput, ritardo e perdita, che richiedono una rete in grado di supportare differenti livelli di servizio, in contrapposizione al tradizionale livello Best-Effort.

Le applicazioni multimediali devono quindi presentare una certa qualità per soddisfare gli utenti, nasce perciò l'esigenza di introdurre il concetto di Qualità del Servizio, QoS(Quality of Service).

Fornire ad un'applicazione una certa Qualità del Servizio, dal punto di vista del sistema di comunicazione, significa fornire a questa i mezzi per soddisfare le proprie necessità.

Il concetto di QoS dipende da diversi fattori :tipo di applicazione, tipo di codifica dei dati, mezzo di comunicazione utilizzato, oltre che dallo strato dell'architettura del sistema cui ci si riferisce [10], [11]. Tuttavia i parametri di QoS generalmente considerati, cioè throughput, ritardo e probabilità di perdita sono presenti in ogni strato e spesso usati con altri parametri per la valutazione delle prestazioni di un sistema.

Si presenta comunque la necessità di una parametrizzazione del servizio offerto dai diversi sistemi alle varie applicazioni. Pertanto ognuna di queste deve poter specificare i propri requisiti in modo preciso, ma allo stesso tempo generale, perché possano essere interpretati e mappati correttamente nei vari strati dell'architettura di comunicazione, al fine di un utilizzo opportuno ed efficiente delle risorse trasmissive.

In questa prospettiva appare evidente che il concetto di QoS ha un significato diverso rispetto al passato; prima la QoS era introdotta per descrivere le caratteristiche tecniche della trasmissione dei dati, osservabili ma non

manipolabili dall'utente, ora sono proprio le applicazioni a sollevare la necessità di gestire differenti livelli di qualità nel trasferimento dell'informazione.

3.2 Fasi del meccanismo di QoS

Il meccanismo di garanzia di QoS ad una data applicazione si può vedere articolato in quattro fasi [10]:

- Specifica della QoS;
- Negoziazione;
- Mappatura;
- Controllo e Gestione.

3.2.1 Specifica

L'applicazione, nella **fase di specifica** della QoS, formula le proprie richieste che il sistema poi valuta se accettare o meno. La specifica riguarda un insieme di parametri, che definiscono alcune caratteristiche di sistema che influenzano la percezione della qualità dei risultati. Tali caratteristiche possono essere raggruppate in diverse categorie di parametri:

- **Parametri temporali**, come il tempo di risposta, le latenze, i ritardi assoluti e le relative variazioni (jitter): essi sono espressi in genere in millisecondi o nell'unità di tempo applicabile in relazione all'ordine di grandezza dei requisiti da soddisfare.
- **Parametri di volume**, utilizzati per rappresentare la quantità di dati trasferiti nell'unità di tempo (throughput). La banda viene misurata nei classici termini informatici (Mb/s,...) oppure in termini legati al tipo di dati da trasferire (frames al secondo, operazioni al secondo,...).
- **Parametri di affidabilità** descrittivi di aspetti, come loss-rate, (e similmente Bit error rate o packet error rate), percentuale temporale di

disponibilità del servizio, MTBF (Medium Time Between Failure) o MTTR (Medium Time To Repair).

La specifica di tali parametri può essere espressa in vari modi: specificando un unico valore, una coppia di valori, il minimo accettabile e quello atteso in media, o un intervallo di valori, naturalmente caratterizzati da livelli di accuratezza della specifica decrescenti, ma dal vantaggio di maggiore libertà nella realizzazione delle procedure di negoziazione e mappatura.

Altra caratteristica della fase di specifica è il modo in cui il sistema andrà a garantire il rispetto dei vincoli sui parametri specificati. Si introduce quindi il concetto di classe di servizio, che definisce il grado di affidabilità con cui il fornitore del servizio rispetta i vincoli sui parametri specificati dall'utente. Fondamentalmente si distinguono

-classi di servizio a garanzia deterministica o a qualità garantita:

caratterizzate dal livello di affidabilità massimo: le prestazioni verranno garantite per tutta la durata del servizio;

-classi di servizio a garanzia statistica per cui è prevista la possibilità che i requisiti negoziati non vengano rispettati per definite percentuali di tempo o di unità informative;

oltre a queste è naturalmente presente la tradizionale: **“classe di servizio” best-effort.**

3.2.2 Negoziazione

La **negoziazione** avviene tra le entità preposte alla fornitura e quelle che richiedono una determinata QoS. Può essere di tipo unilaterale, nel caso in cui l'entità richiesta può solo accettare o rifiutare la richiesta di QoS, oppure di tipo cosiddetto bilaterale, in tal caso il servizio richiesto può alterare la specifica di QoS presentata da una data applicazione. Ad esempio, da una specifica su un

singolo valore può passare a proporre una garanzia di QoS definita su un intervallo di valori, e quindi rispedire la specifica modificata all'entità richiedente. Una possibilità potrebbe essere quella di iterare questo meccanismo fino al raggiungimento di un compromesso tra le parti accumulando però, molto probabilmente, ritardi proibitivi.

Un altro approccio è quello di centralizzare le operazioni di negoziazione in un QoS Broker, isolando i partners della negoziazione e soprattutto svincolandoli dalla conoscenza dei particolari della comunicazione tra di essi.

3.2.3 Mappatura

Con l'operazione di **mappatura** si realizza la traduzione dei requisiti di QoS, caratteristici di uno strato, in una forma compatibile con quella degli strati sottostanti, per dare inizio alla successiva negoziazione su tali strati. La mappatura è un'operazione in genere difficile da realizzare, infatti, nel percorso da sorgente a destinazione, può essere molto eterogeneo l'ambiente di comunicazione interessato e molto diversi tra loro i parametri di QoS considerati, pertanto tale operazione si rivela critica per quanto riguarda un uso efficiente delle risorse.

3.2.4 Controllo e Gestione

Il controllo e la gestione sono le due componenti dell'architettura di QoS che, a differenza delle precedenti già considerate, operano durante la fase di trasferimento dei dati.

Il compito del **meccanismo di controllo** è quello di garantire i requisiti di QoS in tutti i componenti del sistema che prendono parte al trasferimento dei dati. Questo implica ad esempio la gestione dei buffer nei sistemi terminali, e il packet scheduling e il controllo del flusso nei nodi di reti, per garantire il rispetto dei vincoli su ritardo e/o throughput. Inoltre provvede alla gestione dei meccanismi di rivelazione e/o correzione degli errori, sia nei sistemi terminali che negli host

intermedi, per il soddisfacimento dei vincoli sui parametri di affidabilità pattuiti con l'applicazione.

Infine il **meccanismo di gestione** della QoS opera un monitoraggio delle risorse al fine di verificare che la QoS pattuita sia realmente assicurata dal fornitore del servizio.

Da quanto detto segue che la necessità di offrire garanzie di qualità di servizio richiede dei meccanismi di gestione delle risorse. Infatti operando in ambienti con risorse limitate, il sistema non solo deve avere risorse a sufficienza, ma queste devono essere organizzate in modo da essere disponibili quando è necessario.

Le risorse devono quindi essere esplicitamente gestite per soddisfare i requisiti delle applicazioni e questo implica che i blocchi chiave del sistema devono essere i meccanismi di resource reservation e admission control.

3.3 Modello Integrated Service: IntServ

La crescente presenza in Internet di applicazioni con diversi requisiti, in termini di ritardo, throughput e perdita, ha reso necessario introdurre il concetto di QoS anche in tale architettura di comunicazione. [12]

Come già accennato la rete tradizionale offre un servizio non affidabile, nel senso che non è garantita la consegna delle unità informative, IP-PDU, queste possono essere perse, duplicate o consegnate fuori sequenza; inoltre possono subire un ritardo variabile e non predefinito. Il servizio è inoltre senza connessione e ogni IP-PDU è trattata in modo indipendente dalle altre. In tale contesto quindi sembra non trovare spazio la possibilità di garantire qualità di servizio e si è perciò posta l'esigenza di arricchire la struttura esistente per offrire garanzie di QoS.

L'approccio seguito, a tal fine dalla comunità IETF, è stato quello di estendere l'architettura Internet esistente, piuttosto che proporre una nuova infrastruttura parallela destinata alle applicazioni emergenti, quelle real-time in particolare. [14]

Il modello proposto è stato chiamato Integrated Service, IntServ, e il nome stesso rivela la strategia seguita, avente l'obiettivo di considerare un modello valido per differenti classi di traffico, e non solo per il supporto della QoS delle applicazioni real-time.

Il modello IntServ infatti, mira ad integrare i vantaggi delle reti a pacchetto, in grado di massimizzare l'efficienza di utilizzazione delle risorse di rete, con la moltiplicazione di più flussi di dati e capace di fornire il tradizionale servizio BE, e quelli delle reti a circuito che garantiscono qualità dei servizi, ma implicano un utilizzo inefficiente delle risorse di rete in presenza di traffico dati di tipo bursty e si adattano con difficoltà a peggioramenti delle condizioni del canale.

Mentre Internet tradizionale riserva lo stesso 'trattamento' a tutti i pacchetti, il modello IntServ introduce il concetto di flusso, cioè un insieme di pacchetti aventi gli stessi requisiti di QoS. [9] Un flusso è identificato da cinque attributi: source e destination IP address, source e destination port number, e tipo di protocollo di trasporto (in relazione al tipo di applicazioni, si ha ad esempio TCP per il trasferimento di dati e UDP per applicazioni real-time).

In tal modo ogni flusso è "protetto" dagli altri ed è possibile riservare a questo il trattamento adeguato al soddisfacimento dei requisiti di QoS richiesti. Va però sottolineato che un tale approccio si rivela non utilizzabile nella parte più centrale della rete (backbone), dove il numero di flussi è molto elevato e dove è stata proposta una struttura che considera un meccanismo di raggruppamento dei pacchetti, in classi ognuna delle quali riceve un diverso trattamento. Questo approccio assume che i pacchetti di una stessa classe abbiano requisiti di QoS simili, indipendentemente dal flusso di appartenenza. In questo modo sono risolti i problemi di scalabilità dell'altro meccanismo. Si tratta dell'altro approccio

proposto dalla IETF per supportare la QoS in Internet, il cosiddetto Differentiated Service, DiffServ, della cui trattazione non viene riportato oltre in questa sede, in quanto nel progetto che si sta considerando è stato preso come riferimento il modello IntServ.

L'approccio dell'IntServ è caratterizzato dal mantenimento del cosiddetto "flow-related state" per ottenere la riservazione delle risorse, per il soddisfacimento dei requisiti di QoS, determinando così un allontanamento dal tradizionale approccio Internet di tipo end-to-end. Anche in questo caso però non si passa definitivamente dal servizio senza connessione ad uno orientato alla connessione, IntServ assume, grazie al protocollo di riservazione delle risorse, RSVP, un soft-state, in cui le richieste di riservazione sono periodicamente aggiornate, mantenendo la caratteristica di robustezza di Internet e in particolare del tradizionale paradigma senza connessione.

Il meccanismo che permette ai routers di gestire le risorse allo scopo di garantire la QoS richiesta ai diversi flussi è il Traffic Control, composto dai seguenti blocchi: [10]

il **packet classifier** avente lo scopo di mappare i pacchetti in classi, i pacchetti della stessa classe ricevono lo stesso trattamento dal packet scheduler ;

il **packet scheduler** il cui compito è di regolare l'ordine di trasmissione dei pacchetti appartenenti ai vari flussi memorizzati nei buffer;

l'**admission control** che implementa l'algoritmo di decisione, utilizzato da un router o da un host per stabilire se un nuovo flusso di dati può essere accettato o meno nel sistema.

Come già accennato l'altra componente del modello IntServ che concorre al meccanismo di garanzia della qualità di servizio desiderata è il **protocollo di segnalazione RSVP** (Resource reSerVation Protocol) con cui vengono riservate le risorse necessarie per soddisfare le richieste di una data applicazione.

3.4 Classi di traffico e classi di servizio nel modello IntServ

Per il supporto di servizi a qualità garantita il modello IntServ ha definito le seguenti classi di traffico :[14]

Inelastic Traffic o Real time traffic: è costituito dal traffico con requisiti di tempo reale, in cui la qualità è fortemente dipendente dal ritardo di transito nella rete, per il quale è richiesto un valore limite garantito. Il trattamento del traffico real-time è preferenziale rispetto a quello di tipo elastic e necessita di procedure di riservazione delle risorse e di admission control;

Elastic Traffic :comprende il tradizionale traffico Internet, che non presenta vincoli sul ritardo di transito dei pacchetti, le applicazioni di questo tipo sono sensibili alla perdita di pacchetti, attendono quelli in ritardo piuttosto che procedere senza di questi. Al contrario delle applicazioni real-time non sono soggette ad admission control.

Nell'ambito di queste classi di traffico sono previste, dal modello IntServ, tre classi di servizio:

- **Guaranteed Service (GS);**
- **Controlled Load service;**
- **Best-Effort service (BE);**

Le prime due classi sono comprese nella classe di traffico real-time, differendo tra loro per la priorità con cui il servizio viene portato a termine: è concessa maggiore priorità alla classe GS, mentre l'ultimo tipo di servizio rientra nella classe di traffico di tipo elastic.

Di seguito viene riportata una breve descrizione delle classi di servizio introdotte.

3.4.1 La classe di servizio Best-effort

In tale classe di servizio rientrano le tradizionali applicazioni Internet, come Web browsing, trasferimento di file e posta elettronica, che non hanno requisiti stringenti di tempo reale e richiedono solo il soddisfacimento di vincoli

sull'integrità informativa. L'esigenza fondamentale di tali servizi è relativa soprattutto al trasporto dei dati che deve essere il più possibile esente da errori.

All'interno di questa classe è, tuttavia, opportuno distinguere tra applicazioni che sono legate al tempo con cui il servizio è espletato e quelle per cui non ha alcuna importanza il tempo necessario al trasporto dell'informazione. Tra le prime rientrano, ad esempio, le operazioni di trasferimento dei file, che pur non essendo sensibili al ritardo quanto le applicazioni real-time, sono comunque vincolate dal tempo che l'utente è disposto ad aspettare prima che il servizio sia completato. Nella seconda sottoclasse rientrano gli usuali servizi di posta elettronica. Per esse deve essere assicurato un elevato livello di integrità informativa, le perdite di informazione degradano notevolmente il servizio offerto, perciò l'obiettivo primario è garantire un servizio che sia esente da errori. A tale scopo sono preposti i meccanismi di FEC e ARQ o un insieme dei due, introdotti in modo da raggiungere un compromesso tra utilizzazione della risorsa radio e vincolo sull'integrità informativa stessa.

La classe BE oltre a consentire il corretto trattamento delle informazioni che non impiegano il protocollo RSVP, consente di ottenere un'efficiente utilizzazione della risorsa trasmissiva. Infatti, il fatto di dover riservare risorse per la classe GB non consente di tener conto, per queste, del guadagno statistico che si otterrebbe dalla moltiplicazione dei flussi. Una gestione che considera accanto ai flussi a qualità garantita anche quelli di tipo BE, consente di riempire le carenze di risorsa lasciate dai flussi GB. Nel progetto in esame sono stati proposti, a tale scopo, degli algoritmi di scheduling di tipo non work-conserving che consentono di ridurre, per quanto, possibile gli sprechi di risorsa.

3.4.2 La classe di servizio Guaranteed Bandwidth

In tale classe di servizio sono comprese quelle applicazioni con vincoli più stringenti in termini di ritardo end-to-end e throughput. A tali applicazioni va

inoltre garantito un certo grado di integrità informativa, anche se non come requisito più stringente.[15]

Tra le applicazioni tipiche che rientrano in questa classe ricordiamo le applicazioni real-time audio e video, la telefonia via Internet, le videoconferenze, i giochi interattivi,.

Questa classe va quindi trattata al fine di garantire la QoS pattuita in termini di banda, ritardo e perdita, pertanto per ogni flusso sarà necessario allocare le risorse necessarie. Tali risorse sono :banda e buffer.

Per raggiungere questo scopo è necessaria una caratterizzazione del traffico offerto dalle applicazioni che richiedono una certa QoS. A partire dalla conoscenza del traffico emesso dalla sorgente, si devono dimensionare la banda da riservare a ciascun flusso, per garantire il vincolo sul ritardo di trasferimento di un pacchetto da estremo ad estremo, che deve essere minore di un valore massimo dato, per avere una probabilità di perdita per overflow nulla.

Si deve quindi definire una maschera che rappresenti l'andamento nel tempo dei bit emessi; in realtà non si ha una conoscenza esatta di tale maschera di emissione, si ricorre perciò ad una descrizione approssimata del flusso.

3.4.3 Supporto della QoS

Nell'ambito della classe di servizio Guaranteed Bandwidth è necessaria, quindi, la caratterizzazione dei flussi, cui destinare la richiesta QoS, che viene comunemente realizzata attraverso i parametri cosiddetti *Traffic Descriptors*. [17] Il modello *token bucket* è la specifica più comunemente usata nella forma detta della Tspec (Traffic specification). La Tspec combina il modello token bucket con i seguenti parametri:

- p il peak rate (rate di picco) della sorgente considerata;
- M maximum datagram size,caratterizzato da un valore tipico della rete in esame;

-m minimum police unit, che indica la dimensione minima consentita per ogni pacchetto, ogni pacchetto di dimensioni inferiore è assunto di lunghezza pari ad m.

Il token bucket ha una profondità b e un rate r, dove b indica la massima dimensione di burst accettabile ed r il massimo rate di servizio. Quando un pacchetto di x bytes è servito, x bytes sono rimossi dal token bucket, se esso è vuoto il pacchetto deve aspettare in coda affinché il token bucket si riempia con *tokens* a sufficienza.

Il modello token bucket, in unione ai parametri p ed M detti, attua il test di conformità, che identifica i pacchetti che possono ricevere le garanzie di servizio richieste.

Il test definisce cioè la massima quantità di traffico che l'utente può immettere nella rete e per la quale può aspettarsi di ricevere la QoS pattuita in fase di specifica. La massima quantità di traffico è espressa in termini di un involuppo di traffico $A(t)$, che specifica un limite superiore al traffico generato in un qualsiasi intervallo di tempo t, secondo la relazione :

$$A(t) \leq \min (M+pt, b+rt) .$$

Tale equazione stabilisce che l'utente può inviare fino a b bit di dati al suo rate di picco, ma da tale punto in poi deve abbassare il suo rate a quello di servizio del token bucket: r.

Tale modello è essenzialmente quello tradizionale del *leaky bucket* , detto Dual Leaky Bucket, che permette all'utente di specificare un rate di trasmissione a lungo termine: r, conservando la sua capacità di inviare burst a velocità più elevate: p per limitati intervalli di tempo, pari a $(b-M)/(p-r)$.

3.4.4 Allocazione delle risorse secondo il modello Dual Leaky Bucket

Il passo successivo sarà quello di tradurre i requisiti di QoS richiesti da una data applicazione in termini di risorse da allocare, [15].

Se il canale di trasmissione ha un rate di servizio pari ad R , segue che il numero massimo di bit trasmessi in un intervallo di tempo τ è rappresentato dalla retta di consumo $R\tau$. Da quanto detto in precedenza si deduce che, se si vuole trasmettere su tale canale un flusso, con maschera di traffico $A(t)$, una quantità di bit pari a:

$$A(t) - R\tau$$

andrà ad occupare spazio di buffer e volendo evitare trabocco e, conseguente perdita di dati, occorre dimensionare il buffer in modo che risulti:

$$B \geq \text{Max}_{\tau}(A(\tau) - R\tau).$$

La figura 3.1 mostra graficamente la relazione tra l'involuppo di traffico e la retta di consumo.

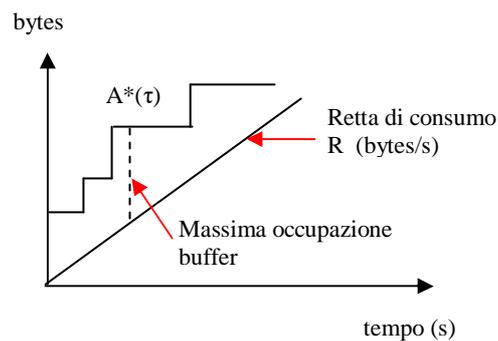


Fig. 3.1 Involuppo e retta di consumo

Inoltre se B indica il massimo riempimento dei buffers nelle condizioni dette, essendo il buffer regolato con la disciplina FIFO, risulta che affinché l'ultimo bit in coda, sia trasmesso sul canale di capacità R , il tempo di coda dovrà essere caratterizzato dalla seguente espressione:

$$t_{\text{coda}} = B/R \text{ , cioè}$$

$$t_{\text{coda}} = (\text{Max}_{\tau}(A(\tau) - R\tau))/R.$$

A questo punto interviene il vincolo sul ritardo richiesto dalle applicazioni di tipo GB, che si stanno considerando.

Indicando con t_{syst} il valore massimo di altri eventuali ritardi introdotti dal sistema, affinché il ritardo massimo totale sia minore del massimo ritardo end-to-end, deve risultare:

$$t_{\text{coda}} + t_{\text{syst}} \leq D_{\text{max}}$$

Come già detto in realtà non si conosce l'andamento esatto nel tempo del traffico generato dalla sorgente in esame, per cui si passa a considerare una seconda maschera: $M(\tau) \geq A(\tau)$

che consente, attraverso i parametri del modello DLB, il calcolo del valore di banda da garantire a quel flusso, $R' > R$, che sia il più possibile prossimo al valore di R che si sarebbe ottenuto considerando la vera curva di inviluppo. Questo al fine di evitare sovrastima di banda e spazio di buffer da allocare.

La condizione che dovrà essere verificata per rispettare il vincolo sul ritardo diventa:

$$(\text{Max}_{\tau}(A(\tau) - R'\tau))/R' + t_{\text{syst}} \leq D_{\text{max}}$$

I parametri DLB forniscono un'approssimazione lineare a tratti della maschera del flusso: $M(\tau)$.

Come già detto secondo tale modello la sorgente trasmette al suo rate di picco per un intervallo τ , pari a $(b-M)/(p-r)$, deducibile come punto di intersezione delle rette $p*\tau+M$ e $r*\tau+b$, visibili nella figura 3.2.

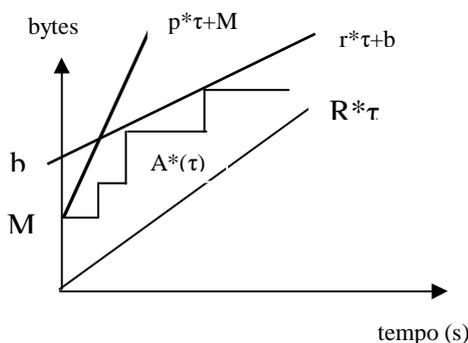


Fig. 3.2 Determinazione dei parametri DLB dall'involuppo

Il vincolo sul ritardo in termini dei parametri DLB viene così espresso:

$$\left(\frac{r^*(b-M)}{(p-r)} + b - R^* \frac{(b-M)}{(p-r)} * \frac{1}{R'} + D_{\text{sys}} \right) \leq D_{\text{max}}$$

ed esplicitando rispetto ad R' si ottiene :

$$R' \geq \frac{p^*b - r^*M}{(D_{\text{max}} - D_{\text{sys}}) * (p-r) + b - M}$$

E posto a sistema con la condizione $R' > r$ porta alla relazione seguente:

$$R' \geq \text{MAX} \left(\frac{p^*b - r^*M}{(D_{\text{max}} - D_{\text{sys}}) * (p-r) + b - M}, r \right).$$

Tale relazione fornisce la quantità di banda da riservare, al flusso in esame, al fine di garantire il rispetto del vincolo richiesto sul ritardo.

Come detto sopra la massima distanza tra la maschera DLB e la retta di consumo rappresenta la massima occupazione del buffer, per cui dalle equazioni precedenti segue che, affinché non ci siano perdite per overflow nei buffers, la quantità di spazio di memoria da riservare deve soddisfare la relazione :

$B \geq \text{Max}_{\tau} (A(\tau) - R' \tau)$; che in termini dei parametri DLB diventa:

$$B \geq \frac{p^*b - r^*M}{p-r} - R' * \frac{b-M}{p-r}.$$

Per completezza dell'esposizione, occorre aggiungere che il modello sinora considerato, per determinare il ritardo sperimentato da un pacchetto appartenente ad un flusso GB, è un'approssimazione del cosiddetto "modello fluidico" (Fluidic Model).

Il modello "fluidico" con rate R' è essenzialmente il servizio che dovrebbe essere garantito da un collegamento dedicato di banda pari ad R' . Il modello Guaranteed Service con rate di servizio pari ad R' , approssima il comportamento del modello "fluidico", dove R' è una porzione di banda piuttosto che la banda di una linea dedicata. Nell'espressione sopra riportata per il ritardo massimo tollerabile vanno perciò introdotti dei termini correttivi: C e D , necessari per considerare le deviazioni dal modello suddetto.

C è un termine dipendente dal rate del flusso, è, ad esempio, indicativo del ritardo di pacchettizzazione.

D, in una rete di tipo slotted, tiene conto del ritardo legato alla posizione degli slots destinati al flusso in esame.

Le espressioni precedenti assumono la forma seguente:

$$D_{\max} = \frac{(b-M)(p-R')}{R'(p-r)} + \frac{M+C}{R'} + D \quad \text{con la condizione } p \geq R' \geq r$$

$$D_{\max} = \frac{M+C}{R'} + D \quad \text{con la condizione } R' \geq p \geq r.$$

Da cui si ricava l'espressione conseguente per il rate di servizio da garantire:

$$R' = \frac{p*(b-M) + (M+C)*(p-r)}{(D_{\max} - D_{\text{sysf}})*(p-r) + b-M} \quad \text{se } p \geq R' \geq r$$

$$R' = \frac{M+C}{D_{\max} + D_{\text{sysf}}} \quad \text{se risulta } R' \geq p \geq r.$$

Analogamente l'espressione per la dimensione del buffer si modifica nella seguente:

$$B = \frac{(b-M)*(p-R')}{p-r} + M + C + R'*D.$$

3.4.5 Controlled Load Service

Di seguito sono riportati alcuni cenni sulla descrizione della terza classe di servizio considerata dal modello IntServ.

La definizione della classe di servizio Controlled Load: CL è di natura qualitativa, essa approssima il servizio di tipo Best-Effort in condizioni di carico della rete lontani dalla congestione, ottenendo assenza quasi totale di ritardo e perdite. [16]

Più precisamente assumendo che la rete stia funzionando correttamente, le applicazioni di tipo CL possono assumere che:

-un 'alta percentuale dei pacchetti trasmessi giungerà correttamente a destinazione. La percentuale di pacchetti ricevuti non correttamente dovrà avere un valore molto vicino al packet error rate del mezzo di trasmissione.

-il ritardo sperimentato da un'alta percentuale di pacchetti trasmessi non supererà di molto il ritardo minimo sperimentato da ogni pacchetto giunto correttamente a destinazione.

Il servizio CL è stato concepito per supportare un 'ampia classe di applicazioni oggi diffuse in Internet e molto sensibili a condizioni di sovraccarico della rete, ad esempio la classe delle applicazioni real-time di tipo adattativi.

Nonostante la definizione qualitativa, la classe CL richiede una specifica quantitativa del traffico di utente nella forma delle TSpec, già illustrate per la classe GB. Questo è importante ai fini dell'admission control, perché permette di controllare ed, eventualmente limitare, il numero di flussi che la rete può accettare per garantire il desiderato comportamento di rete "unloaded".

Va ricordato che il termine "unloaded" non è da intendersi come assenza totale di qualsiasi altro flusso di traffico, ma semplicemente come indicativo di una condizione di carico offerto alla rete lontana da situazioni di congestione.

Quando un elemento della rete accetta la richiesta di un servizio CL, deve assicurare che siano disponibili la banda necessaria e le risorse di elaborazione al fine di soddisfare le richieste specificate nelle TSpec, attuando le operazioni di admission control necessarie.

Il servizio fornisce essenzialmente la garanzia di un rate di trasmissione pari al token bucket rate: r , e la possibilità di trasmettere bursts, con dimensioni limitate dalla token bucket depth: b , al rate di picco p .

La definizione del servizio Controlled Load è intrinsecamente imprecisa, è più opportuno parlare di una qualità del servizio relativa ad eventi con scarsa frequenza di presentazione.

Un flusso che sta ricevendo un servizio di tipo CL deve attendersi di sperimentare:

-piccoli ritardi di coda dei pacchetti su scale temporali maggiori del “burst time”, ossia dell’intervallo di tempo in cui la sorgente può trasmettere al suo rate di picco, come specificato dalle TSpec;

-perdite per congestione della rete piccole o nulle, su intervalli temporali eccedenti il burst time. Tali perdite sono dovute a perdite di pacchetti determinate da mancanza delle risorse richieste, come banda e buffer. Sebbene sono ammesse delle perdite occasionali, perdite sostanziali sono da considerarsi un fallimento dell’algoritmo di admission control.

In pratica le definizioni si basano sul valore di aspettazione della durata degli eventi di perdita e ritardo.

Eventi di breve durata sono visti come effetti statistici che possono presentarsi occasionalmente, mentre eventi di durata maggiore sono indicativi di fallimento dei meccanismi di allocazione delle risorse necessarie al servizio CL.

Si può utilizzare un approccio di tipo statistico per decidere se sono disponibili risorse sufficienti per accettare le richieste di servizio, e consentire ad esempio una “sovallocazione” di risorse, senza tuttavia deteriorare il livello di servizio dei flussi.

Cioè gli elementi di una rete che forniscono un servizio Controlled Load possono permettere che i requisiti di banda e buffer, ottenuti sommando le richieste di tutti i flussi di tipo CL, superino i valori massimi realmente disponibili dello stesso elemento. Naturalmente è necessario che ciò avvenga solo nei casi in cui si è sicuri che l’utilizzazione effettiva della rete sia minore del carico corrispondente alla somma delle TSpec dei flussi CL presenti, in modo da non compromettere le prestazioni desiderate. Le informazioni a riguardo si possono ottenere da misure di traffico aggregato o da conoscenze di statistiche di traffico.

Il servizio Controlled Load assicura quindi il controllo della QoS fornita a quei flussi di traffico che risultano “conformi” alle TSpec specificate nella fase di negoziazione. Cioè come già detto per la classe GB, il carico di traffico offerto dall'applicazione in questione non deve essere superiore alla quantità $r \cdot T + b$, dove r e b sono i parametri del modello token bucket e T la lunghezza dell'intervallo temporale. I pacchetti che non rispettano tale vincolo sono considerati non conformi. In presenza di pacchetti non conformi la rete deve comunque continuare a garantire la qualità del servizio negoziata agli altri flussi ed evitare che gli “eccessi” di traffico CL non vadano a pregiudicare la possibilità di trasmissione dei flussi best-effort.

A tale proposito va quindi considerato il problema dell' interazione del traffico Controlled Load con quello di tipo Best-Effort. Infatti le condizioni di non conformità alle TSpec possono presentarsi come condizioni operative normali e non solo essere conseguenza di errori. Se l'elemento della rete non prevede una distinzione negli algoritmi di “queueing” tra pacchetti best-effort e pacchetti CL, l'eccesso di traffico CL potrebbe dominare l'utilizzo della banda destinata al traffico Best-Effort.

L'approccio per risolvere il problema potrebbe essere quello di mantenere separata l' allocazione delle risorse per le due differenti classi di traffico Best-Effort, in modo che nessuna classe domini l' utilizzo della banda.

Alternativamente si potrebbe assegnare una priorità minore all'elaborazione degli eccessi di traffico per evitare che questi abbiano il monopolio della banda a scapito del traffico Best-Effort.

In ultimo va considerato il livello di QoS riservato ad un flusso in presenza di pacchetti non conformi. Anche qui si hanno due possibilità, ossia scegliere di degradare in modo eguale il servizio offerto per tutti i pacchetti del flusso, oppure dividere i pacchetti in due insiemi, conformi e non, e riservare differenti livelli di servizio ai due insiemi.

Nel primo caso saranno complessivamente peggiorate le prestazioni in termini di ritardo e perdite di tutto il flusso, nel secondo caso alcuni pacchetti sperimenteranno ritardi maggiori, ma altri avranno prestazioni in termini di ritardo molto più vantaggiose del caso di trattamento egualitario, e questa seconda circostanza si rivela preferibile per quelle applicazioni con stringenti requisiti real-time, come ad esempio le video conferenze.

3.5 ReSource reserVation Protocol:RSVP

Come già detto fornire la richiesta QoS ad un'applicazione implica la necessità di riservare le risorse di rete necessarie. Per tale motivo è necessario una componente per il trasporto delle informazioni necessarie per il trattamento adeguato dei pacchetti tra sorgente e destinazione.

L'RSVP è un protocollo di prenotazione usato nei routers delle reti IS per soddisfare le richieste di QoS [18]. Tale protocollo consente a mittenti e destinatari di comunicare tra loro tramite messaggi di controllo, per stabilire opportuni stati e modalità di trattamento dei datagrammi all'interno dei routers, in modo da poter supportare le classi di servizio prima definite. Con l'uso del protocollo RSVP gli host specificano le loro richieste di QoS alla rete. RSVP trasferisce le richieste ai routers presenti sul percorso e mantiene le informazioni di stato relative al servizio richiesto nei routers e negli host interessati, il cosiddetto flow-related state.

E' bene ricordare che l'RSVP nel modello di rete a strati è posto al di sopra dello strato IP, prescinde quindi da meccanismi di routing, in realtà esso non trasporta i dati dell'applicazione, ma opera solo come un protocollo di controllo, la sua funzione è quella di prenotare le risorse lungo un cammino già identificato dai protocolli di routing.

Naturalmente affinché i messaggi di controllo si propaghino correttamente tra sorgente e destinatario, i routers partecipanti alla sessione devono conoscere i

percorsi correnti nella rete. A tale scopo le tabelle di routine, che definiscono tutti i percorsi di rete, vengono aggiornate ad intervalli regolari.

Il protocollo RSVP è un protocollo del tipo “iniziato dal destinatario”, *receiver oriented*, in cui il destinatario deve conoscere e mantenere le informazioni relative agli indirizzi della sola sorgente dei dati ad esso destinati ed ai suoi requisiti di QoS. Si evitano così i problemi di scalabilità eventualmente connessi ad una soluzione di tipo sender oriented, in cui la quantità di dati da gestire al lato trasmissione potrebbe diventare eccessiva, nel caso di comunicazioni che interessano un gran numero di destinatari, [10].

Nel protocollo RSVP un flusso di dati è modellato come un albero, detto Distribution Tree, in cui, la sorgente è la radice, e che si estende verso tutte le destinazioni, le foglie, e in cui i routers sono i nodi intermedi.

3.5.1 Messaggi del protocollo RSVP

Il protocollo RSVP definisce sei tipi di messaggi, riportati nella tabella 3.1:

Tipo di Messaggio	Mittente del messaggio	Significato del Messaggio
PATH REQUEST	trasmettitore	Attivazione di uno stato di “waiting for session setup”. Fornitura di informazioni al destinatario circa le caratteristiche di traffico di sorgente e del percorso da estremo a estremo in modo che il destinatario possa effettuare richiesta opportuna ai router lungo tutto il percorso da sorgente a destinazione
PATH ERROR	router lungo il cammino	Generazione d'errore scaturita da una richiesta di PATH
PATH TEARDOWN	trasmettitore	Accelera la liberazione delle risorse
RESV (reservation) request	ricevitore	Originato dall'utente destinatario del flusso. Il suo scopo è quello di portare la richiesta di riservazione della risorsa ai router lungo il percorso da sorgente a destinazione
RESV ERROR	router lungo il percorso	Rifiuto di riservazione della risorsa (o perché non disponibile o perché è fallita l'autorizzazione)
RESV TEARDOWN	ricevitore	Accelera la liberazione delle risorse

Tabella 3.1 - Messaggi RSVP

Il *path message* ha due scopi fondamentali :

- portare le specifiche del flusso a tutti i ricevitori;
- stabilire lo stato nei routers intermedi per la propagazione della richiesta di riservazione lungo tutto il percorso sorgente destinazione.

Il messaggio di path contiene le seguenti informazioni:

- Indirizzo di destinazione;
- Phop* , indirizzo dell'ultimo nodo che lo ha generato;
- un campo detto *Sender Template* che identifica la sorgente;
- Adspec* (opzionale) che identifica le caratteristiche del cammino da estremo ad estremo,usato dai destinatari per conoscere in modo più preciso il livello del servizio che può essere ottenuto.

Ogni router che riceve il path message ne controlla la validità, in caso negativo risponde con un *Path Error* a colui dal quale lo ha ricevuto.In caso positivo crea una variabile di stato in cui memorizza le specifiche di traffico, il prossimo Hop ed eventualmente l'Adspec.

Ogni ricevitore può usare l'informazione contenuta nel path message ed eventuali altre informazioni locali per determinare i suoi requisiti di QoS. A questo punto è lui stesso che dà inizio alla riservazione delle risorse. Genera pertanto un messaggio di *Resv* che viaggia verso la sorgente nel verso opposto al path message. Tutto ciò è possibile perché i routers intermedi possono usare le informazioni del path state, fornite dal precedente path message, per propagare le richieste di riservazione verso la corrispondente sorgente. Il messaggio di Resv contiene un descrittore del traffico di sorgente(*Tspec*) che in genere è posto uguale al *Tspec* della sorgente, e le caratteristiche della riservazione (*Rspec*), che comprendono la banda calcolata da riservare ad ogni router ed un termine di correzione, detto Slack Term, che serve a tenere in conto eventuali margini del sistema per il ritardo da estremo ad estremo rispetto a quello tollerabile dall'applicazione. In particolare il campo *flow spec* specifica la quantità di risorse

richieste e il campo *filter spec* identifica la sessione, cioè l'insieme di pacchetti che utilizzano tali risorse.

Le richieste di riservazione nel loro percorso verso la sorgente sono sottoposte a test di admission control locali nei routers attraversati, allo scopo di stabilire se sono disponibili banda e buffers a sufficienza per soddisfare la richiesta. Se la richiesta non può essere accettata viene emesso un messaggio di errore: Resv Error.

Va comunque ricordato il meccanismo di negoziazione della QoS previsto dal protocollo RSVP, detto *One-Pass With Advertisement* (OPWA). In questo modello le sorgenti di dati periodicamente inviano i cosiddetti advertisement messages, l'Adspec del path message, che forniscono ai ricevitori le informazioni sul livello di servizio per il quale possono aspettarsi che l'admission control abbia successo. In questo modo è semplificato il compito dei ricevitori nella formulazione delle richieste di servizio che possano essere verosimilmente accettate.

I messaggi di Path e Resv sono inviati periodicamente dal protocollo RSVP, per cui esso mantiene il cosiddetto *soft-state* nei nodi intermedi. Attraverso i messaggi di Path Teardown e Resv Teardown, allo scadere di un timer avviene la cancellazione dello stato, non utilizzato, e quindi la "liberazione" delle risorse. La cancellazione è evitata se e solo se prima dello scadere del timer arriva un nuovo messaggio, di Path o di Resv, rispettivamente. In questo modo, come già detto, si cerca di mantenere il paradigma senza connessione dell'architettura Internet tradizionale, con tutti i suoi vantaggi ricordati.

La figura 3.3 mostra il meccanismo appena descritto dell'instaurazione di una sessione RSVP.

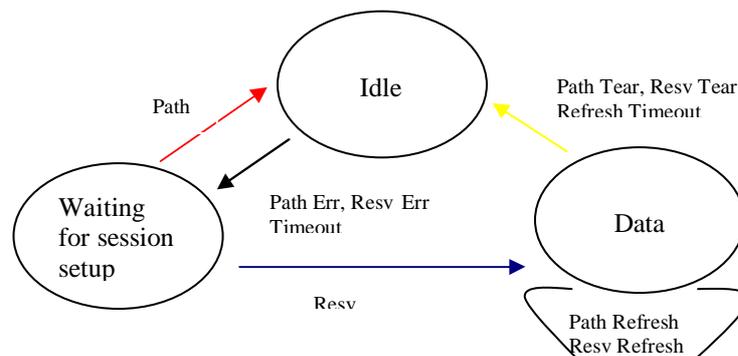


Fig.3.3 Diagramma di stato dell'instaurazione di una sessione RSVP:

Nel caso del sistema in esame e facendo riferimento alla tratta di Downlink, quando un messaggio di PATH viene inviato da una sorgente remota, arriva, tramite il router di accesso, al RN. Questo consulta la tabella di routing nella quale l'indirizzo di ogni RT è associato a tutti gli indirizzi di tutti gli utenti attestati allo stesso RT.

Se l'indirizzo IP non viene trovato, il RN invia un messaggio di PATH ERROR, altrimenti il messaggio di PATH viene segmentato e mandato in Downlink come un set di MACPDU destinate al RT a cui l'utente è connesso. A questo punto se il RT ha ricevuto correttamente il messaggio lo riassume e lo invia all'utente destinatario. Ricevuto il messaggio di PATH, se l'utente finale decide di accettare l'instaurazione della sessione manda un messaggio di RESV al RT, il quale riserva risorse tra lui e l'utente come specificato dal protocollo RSVP, richiedendo al RN l'assegnazione di banda per inviare le MACPDU in cui è stato segmentato il messaggio. Se la trasmissione del messaggio di RESV avviene prima dello scadere del timeout dello stato di Path attivato dal RN, il RN riserva banda e buffer all'interfaccia radio e inoltra il messaggio di RESV verso la sorgente remota.

Le scelte di progetto attualmente prevedono un approccio di tipo deterministico alla soluzione del problema della garanzia della QoS richiesta dalle applicazioni. Seguendo il modello IntServ si sono individuate le sorgenti con più stringenti

requisiti in termini di ritardo e banda, e per queste vengono allocate le risorse allo scopo di ottenere il soddisfacimento dei vincoli richiesti.

Tuttavia è emerso dalle simulazioni sulle condizioni di carico della rete che con tale approccio si incorre in sprechi di risorse, cioè la valutazione della quantità di risorse da riservare alle sorgenti di tipo GB si è rivelata in alcuni casi di tipo conservativo. A tali sorgenti è stata riservata più banda di quanto in realtà necessitano per soddisfare i propri requisiti di QoS, determinando in tal modo il rischio di un sottoutilizzo complessivo della rete. Il motivo di ciò risiede nell'approccio deterministico scelto per allocare le risorse richieste, è stato previsto il soddisfacimento assoluto dei vincoli sul ritardo.

Nella trattazione introduttiva si è infatti accennato all'esistenza di altri tipi di garanzie di servizio, quella di tipo statistico in particolare. In tali tipi di servizio è prevista la possibilità di violazione dei vincoli di QoS richiesti per alcune percentuali di tempo e/o per alcune percentuali di unità informative richiedenti un determinato servizio, senza deterioramento sensibile delle prestazioni.

Oggetto del presente lavoro è quello di individuare e verificare i vantaggi di un simile approccio di garanzia di QoS di tipo statistico.

Dai lavori presenti in letteratura emerge che con tale tipo di garanzie si ha un incremento del numero di sorgenti che possono essere ammesse e ricevere il servizio richiesto, pur preservando le prestazioni delle applicazioni con più stringenti vincoli di QoS. Questo è quello che viene indicato come guadagno di moltiplicazione statistica.

CAPITOLO 3: QUALITA' DEL SERVIZIO: QoS.....	40
3.1 QUALITA' del SERVIZIO: INTRODUZIONE.....	40
3.2 Fasi del meccanismo di QoS	42
3.2.1 Specifica.....	42
3.2.2 Negoziazione	43
3.2.3 Mappatura.....	44
3.2.4 Controllo e Gestione.....	44
3.3 Modello Integrated Service: IntServ.....	45
3.4 Classi di traffico e classi di servizio nel modello IntServ.....	48
3.4.1 La classe di servizio Best-effort	48
3.4.2 La classe di servizio Guaranteed Bandwidt.....	49
3.4.3 Supporto della QoS	50
3.4.4 Allocazione delle risorse secondo il modello Dual Leaky Bucket.....	51
3.4.5 Controlled Load Service	55
3.5 ReSource reserVation Protocol:RSVP	59
3.5.1 Messaggi del protocollo RSVP	60

CAPITOLO 4: MODELLI DI TRAFFICO

4.1 INTRODUZIONE

Come già detto, nel progetto in esame si vuole che il sistema sia in grado di supportare il trasferimento di flussi di dati con caratteristiche diverse in termini di traffico e requisiti di qualità del servizio.

Il problema di progetto che si presenta può quindi essere così espresso: date le richieste di servizio e i requisiti di QoS, occorre determinare le risorse di rete necessarie al conseguimento degli obiettivi prefissati.

Per tale scopo occorrono dei modelli matematici per :

- la caratterizzazione delle richieste di servizio, che corrisponde alla definizione del traffico nella rete ;
- la descrizione dell'interazione tra attività e risorse, basata sulla teoria delle code;
- individuazione di metodi per garantire la qualità del servizio.

Nell'ambito del lavoro di progetto è stata, quindi, seguita questa linea, procedendo in una prima fase alla caratterizzazione del traffico e, successivamente, con l'ausilio dei modelli matematici forniti dalla teoria delle code, si procederà al dimensionamento delle risorse necessarie al soddisfacimento delle richieste di QoS delle varie applicazioni.

4.2 CARATTERIZZAZIONE E MODELLIZZAZIONE DELLE SORGENTI DI TRAFFICO

Una rete di telecomunicazioni può essere schematizzata come nella figura 4.1, dove sono presenti delle stazioni sorgenti che generano dati che vengono immessi nella rete, e delle stazioni destinazioni che li ricevono.

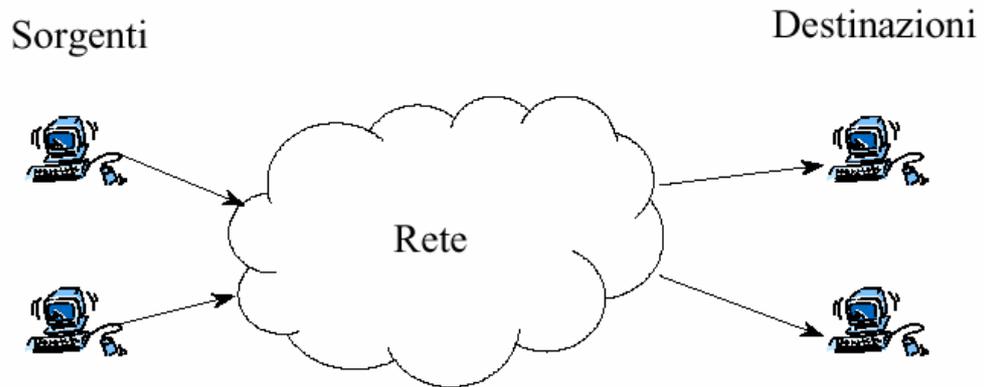


Fig. 4.1 Schema di una Rete di Telecomunicazioni

La caratterizzazione e la modellizzazione del traffico offerto dalle varie applicazioni di una rete di telecomunicazioni costituiscono, in fase di progettazione, un passo importante per il dimensionamento e la valutazione delle prestazioni. Allo scopo di ottenere valutazioni verosimili ci si avvale di strumenti di simulazione che utilizzano dati di misure di traffico su reti reali.

4.2.1 Caratterizzazione delle sorgenti

Le sorgente di traffico rappresentano il modo in cui l'utente esprime le proprie richieste di servizio, [19].

Caratterizzare una sorgente significa definire statisticamente il comportamento della stessa in termini del traffico offerto alla rete. Una sorgente di traffico (DTE) può essere modellata tramite una sorgente ed un codificatore, come illustrato nella figura 4.2.

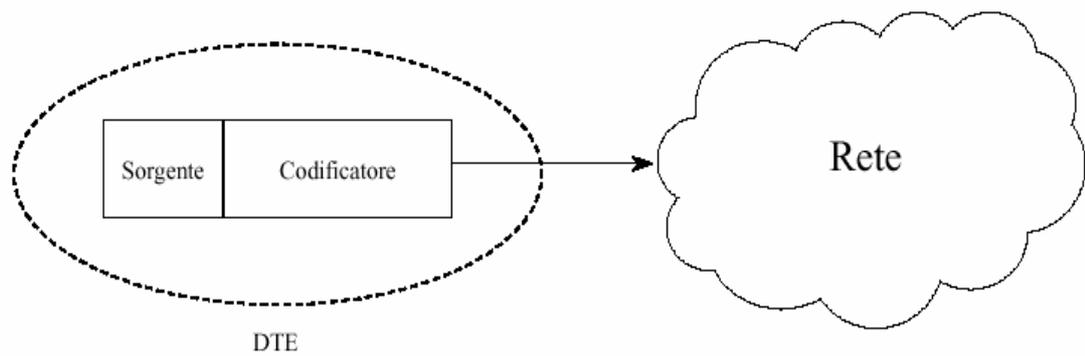


Fig.4.2 Modello di una sorgente di traffico DTE

Le sorgenti di traffico si possono raggruppare in quattro famiglie:

Sorgenti Audio;

Sorgenti Video;

Sorgenti Dati;

Sorgenti Multimediali, queste rappresentano un aggregato delle sorgenti precedentemente elencate.

Una prima classificazione delle sorgenti di traffico può essere effettuata in base alla modalità con cui i dati vengono emessi dal codificatore.

Si distinguono in particolare :

-sorgenti *CBR* (*Constant Bit Rate*), caratterizzate da emissione a bit rate costante, come visibile nella figura 4.3.

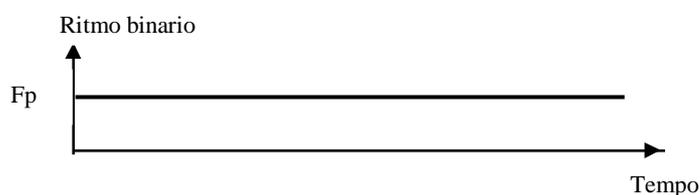


Fig.4.3 Comportamento di una sorgente CBR

Le sorgenti CBR sono caratterizzate da :

- bit rate (bit/s);
- durata (s);
- processo di generazione delle chiamate.

Un esempio di sorgente CBR è costituito dalla codifica PCM (Pulse Code Modulation) che codifica la voce con bit rate costante (64kbit/s). Essa è però poco efficiente in quanto non tiene conto dell'alternanza dei periodi di interlocuzione e di ascolto. In tali periodi di pausa il codificatore emette sequenze di cifre binarie che non sono portatrici di informazione telefonica.

-Sorgenti *VBR* (*Variable Bit Rate*) con ritmo di emissione variabile nel tempo, un esempio è costituito dalle immagini video in movimento con codifica MPEG (Moving Picture Export Group).

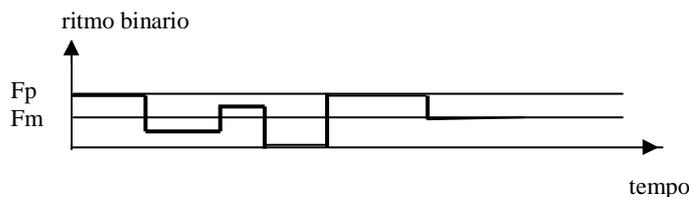


Fig.4.4 Comportamento di una sorgente VBR

Per caratterizzare una sorgente VBR occorre fornire:

- il rate di picco (bit/s);
- il rate medio (bit/s);
- la durata (s);
- il processo di generazione delle chiamate.

Vanno poi introdotte le sorgenti VBR di un tipo particolare, detto a due stati, le sorgenti *ON-OFF*. Esse sono caratterizzate dal fatto che l'emissione o avviene al picco o non avviene proprio. Un esempio di sorgente ON-OFF è costituito dalla voce codificata ADPCM (Activity Detector PCM), cioè con rivelazione di tratti vocali.

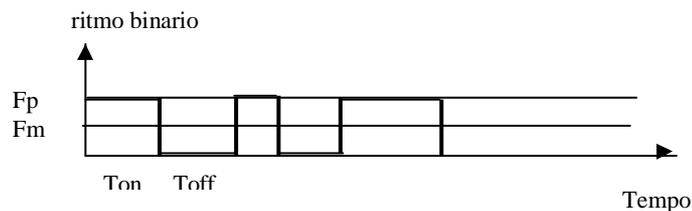


Fig.4.5 Comportamento di una sorgente On-Off

4.2.2 Modelli di Traffico

Per ottenere dei modelli di traffico che siano validi strumenti nel dimensionamento e nella valutazione delle prestazioni di una rete di telecomunicazioni si fa riferimento ai risultati delle misure di traffico offerto dalle diverse applicazioni ad una rete reale.

In letteratura sono disponibili diversi risultati di tale tipo di misure ed i relativi modelli ricavati, per la descrizione statistica del comportamento delle sorgenti tipiche in una rete di telecomunicazione.

Uno dei metodi più diffusi per effettuare le misure del traffico IP è stato il **server logs**, basato sulla proprietà dei server di conservare “traccia” dei files serviti. Con questa informazione è possibile creare un modello del carico di traffico offerto al server. In questo modo però non è possibile investigare il comportamento del singolo utente e quindi non si può ricavare un modello a questo relativo.

La strategia di misura del traffico Internet più comunemente usata di recente è la cosiddetta “ **packet traces** “. In questo caso le misure sono realizzate ‘seguendo’ la traccia dei pacchetti, in una sottorete interessata da traffico IP, attraverso opportuni strumenti di monitoraggio , *sniffer*. Questi sono situati in posizioni tali da collezionare tracce di tutto il traffico della rete, e tali naturalmente da non falsare, con la loro presenza, le misure stesse, in particolare non devono generare traffico aggiuntivo significativo. In una Wireless LAN in genere il punto di

monitoraggio del traffico coincide con il router, in corrispondenza di un Access Point verso le reti Internet o ATM.

Dalla analisi della intestazione dei pacchetti è possibile ricavare diverse informazioni fra cui source e destination address, source e destination port number, protocol index, ampiezza dei pacchetti. In questo modo è possibile identificare il protocollo di trasporto delle varie applicazioni e distinguere, grazie al port number, a quale applicazione i pacchetti appartengono. Con questo approccio è quindi possibile caratterizzare il comportamento del singolo utente.

Il principale svantaggio di questo tipo di misure è che si perdono le informazioni, sui pacchetti, relative agli strati superiori, come il tipo di file o dei documenti cui appartengono i pacchetti, [20].

A partire dall'osservazione dei dati osservati, sono poi ricavate le leggi delle distribuzioni statistiche che meglio approssimano gli andamenti delle grandezze misurate.

Le caratteristiche delle statistiche di traffico ottenute sono in genere funzioni di molti parametri, come il momento della giornata in cui è effettuata la misura, la direzione del traffico, uscente o entrante dalla rete in esame, oltre che naturalmente dell'applicazione, del protocollo di trasporto e della dimensione dei pacchetti.

Un modello di una sorgente di traffico consiste di due parti:

Il processo di descrizione della fase di attività .

Il processo di arrivo dell'attività di utente, [21].

Il modello della fase di attività dipende naturalmente dal tipo di applicazione in esame, mentre il processo di arrivo determina gli istanti in cui un'applicazione inizia la sua attività, tali istanti sono chiamati *tempi di arrivo*.

Allo scopo di ottenere uno scenario di riferimento realistico nel progetto della Wireless LAN in esame, si sono individuate dapprima le applicazioni tipiche in

tale contesto e quindi sono stati considerati dei modelli di traffico relativi, da utilizzare nella simulazione per la verifica delle prestazioni.

Nel seguito quindi passiamo ad analizzare le sorgenti individuate e i modelli di traffico proposti.

4.3 Traffico vocale

Il traffico vocale è ottenuto campionando ad intervalli regolari, e poi comprimendo, il segnale proveniente da una sorgente vocale. I metodi di codifica sono tali che il messaggio originale venga ricostruito al ricevitore senza degradarne la qualità. Inoltre, a seconda del tipo di codifica utilizzato, la perdita di unità informative può essere compensata o meno. La perdita di un certo numero di unità informative può causare a destinazione periodi di silenzio o troncamento del segnale ricostruito.

Come già detto, a seconda del tipo di codifica utilizzato per le sorgenti vocali, il traffico generato può essere modellato come CBR o come VBR a due stati.

Se viene utilizzato un codificatore PCM (Pulse Code Modulation) a 64kbit/s, il traffico generato è di tipo CBR. Se invece la codifica sull'uso di tecniche *Speech Activity Detector* (SAD) e *Digital Speech Interpolation* (DSI), il traffico risultante è di tipo VBR a due stati.

Queste tecniche sfruttano la ridondanza intrinseca in un segnale vocale ed eliminano la non necessaria trasmissione degli intervalli di silenzio durante una chiamata.

Il modello, proposto in letteratura, [8], assume una sorgente vocale come un processo ON-OFF, cioè un processo di rinnovamento (renewal process) che può assumere due stati: uno stato di attività (talkspurt o ON) ed uno stato di inattività (silenzio o OFF). Ogni volta che il processo cambia di stato si ha un rinnovamento, cioè viene dimenticata la storia passata.

Le sorgenti vocali che emettono questo tipo di traffico sono dette *bursty* e il periodo in cui la sorgente emette è detto *burst*. In genere il burst ha una durata temporale limitata, ma è caratterizzato da un elevato bit rate. A tale proposito si definisce una quantità detta *burstiness*, che misura l'indice di attività della sorgente, ed è data dal rapporto tra il ritmo binario di picco ed il ritmo binario medio della sorgente. Essa indica quanto il comportamento della sorgente è vicino al tipo CBR o al tipo VBR. Le sorgenti CBR hanno un valore di burstiness unitario.

Il tipico comportamento di una sorgente bursty è illustrato nella figura 4.6 :

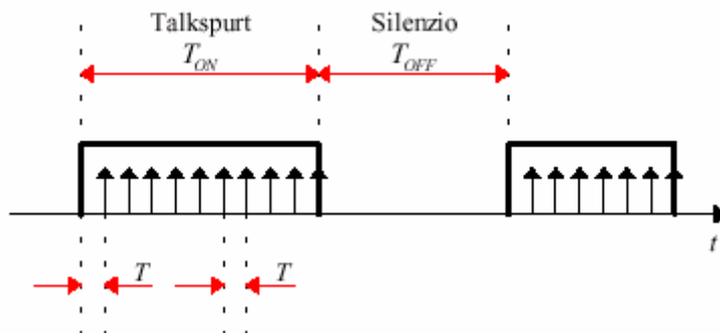


Fig.4.6 Comportamento di una sorgente vocale

Una sorgente vocale è attiva quando il talker parla, mentre è inattiva, e non genera pacchetti, durante i periodi in cui il talker è in silenzio. Pertanto nel caso di modellizzazione di una sorgente vocale come sorgente ON-OFF, il numero di celle in trasmissione nella rete si riduce del 35-40% rispetto al caso in cui la sorgente venisse codificata non considerando i periodi di silenzio.

E' stato verificato che, in una normale conversazione, la distribuzione esponenziale è adatta per la descrizione della durata dei periodi di attività, mentre la durata dei periodi di silenzio è approssimata meno bene da questa distribuzione. Tuttavia in genere per facilitare l'analisi si assume che entrambi i periodi siano distribuiti esponenzialmente con valor medi pari a \bar{T}_{ON} e \bar{T}_{OFF} , i cui valori sono disponibili in letteratura.

4.4 Traffico Video

Le sorgenti video considerate sono di due tipi: la videoconferenza, caratterizzata da pochi o limitati cambi di scena, e le immagini video in movimento con più frequenti cambi di scena.

A causa dell'eccessiva richiesta di banda dei flussi video non compressi, sono stati sviluppati degli algoritmi di codifica per la compressione di questo tipo di dati. L'algoritmo di compressione ha lo scopo di ridurre il data rate del flusso video prima che questo sia trasmesso nella rete di telecomunicazioni. Questo viene realizzato riducendo la ridondanza spaziale e temporale dei flussi video, [26].

Per una videoconferenza può essere ottenuto un alto rate di compressione utilizzando un algoritmo di codifica relativamente semplice, spesso è utilizzata una codifica CBR. In tal modo il traffico generato presenta delle caratteristiche statistiche relativamente semplici ed inoltre l'allocazione delle risorse è più facile da realizzare.

Per le sorgenti video in movimento la qualità video varia molto rapidamente, i cambi di scena comportano variazioni del bit rate, pertanto mantenere il rate di trasmissione costante porta ad una inevitabile distorsione delle immagini.

Tale tipo di sorgente richiede quindi tecniche di codifica più complicate per raggiungere una sufficiente compressione e garantire allo stesso tempo una qualità delle immagini accettabile. Si utilizzano codifiche di tipo VBR. Il traffico generato presenta, di conseguenza, delle caratteristiche più complesse e sono pertanto necessari dei modelli matematici accurati per la descrizione dello stesso.

La codifica di tipo VBR rende più difficile l'allocazione della banda, ma allo stesso tempo consente di ottenere un migliore guadagno di multiplazione statistica. L'aumento di tale guadagno, rispetto al caso di codifica CBR, è dovuto al fatto che con la codifica VBR la probabilità che più sorgenti di tipo bursty trasmettano al loro rate di picco è bassa, è consentito quindi che la somma dei

rate di picco in ingresso superiori al rate di uscita del multiplexer. Uno svantaggio legato alla codifica VBR è la possibilità di perdita di pacchetti per overflow dei buffers.

Lo schema di codifica utilizzato per la maggior parte delle applicazioni è quello MPEG, Moving Picture Export Group. Le versioni successive, MPEG 2 ad esempio, prevedono rispetto alla prima, MPEG 1, una codifica a strati in cui oltre ad uno strato base che contiene la parte più importante dei dati video, esistono altri strati, di “rafforzamento”, che possono essere usati per migliorare la qualità della sequenza video.

Secondo lo schema MPEG una sequenza video viene suddivisa in GOP, Group Of Picture, contenenti tre tipi di trame: [8]

-*Intraframe* (I): codificate in modo indipendente dalle altre trame;

-*Predictive* (P): codificate (con algoritmi di motion compensation) rispetto alla precedente trama (I o P) o alla successiva (P o I);

-*Bidirectional o Interpolative* (B): codificate come le trame P, ma con predizione in avanti o indietro e interpolazione dei risultati.

Un GOP è definito dalla sua lunghezza N (o la distanza tra le trame I) e la distanza tra le trame P, M. La tipica struttura di un GOP è:

IBBPBBPBBPBB, con N=12 ed M=3.

Il modello di traffico proposto per una sequenza video con codifica MPEG è stato ricavato da misure di tracce video reali, [26], [27]. Osservando tracce video di differenti durata, qualità delle immagini e contenuto delle stesse è stata ricavata una descrizione statistica del traffico video a livello di trama, in termini di frequenza di arrivo e dimensioni in byte delle trame.

4.5 Traffico dati

Esistono due strategie di modellizzazione del traffico dati, una prima detta “black-box”, in cui appunto le caratteristiche statistiche del traffico sono modellate senza tener conto dei meccanismi della generazione del traffico stesso. La seconda categoria di modelli è costituita dalla modellizzazione strutturale o gerarchica, basata su una caratterizzazione più significativa del processo di generazione del traffico. La descrizione statistica che ne consegue è più complessa, ma in compenso prevede l'uso di parametri, per la descrizione del processo, aventi un significato fisico (numero di pagine in una sessione http, dimensione dei pacchetti IP,..), e che possono essere perciò facilmente adattati alle evoluzioni future del tipo di traffico offerto ad una rete, basta aggiornare i relativi valori.

La modellizzazione del traffico dati scelta nella presente trattazione è pertanto di quest'ultimo tipo.

I modelli riportati si riferiscono al traffico entrante nella rete.

4.5.1 Traffico WEB

Dai dati ottenuti dalle misure emerge che la maggiore quantità di traffico Internet generato in una rete appartiene alle applicazioni basate sul protocollo di trasporto TCP, e tra queste prevalgono le applicazioni HTTP, i cui pacchetti sono identificati dal source port number 80.

Con il termine ‘traffico WEB’ si intende tutto il traffico HTTP generato durante una sessione con un tipico web browser.

Naturalmente non tutto il traffico è trasportato allo stesso modo.[21] Ci sono differenze sia tra le differenti versioni HTTP , sia tra differenti web browser. Tutto ciò ha effetto sul modo in cui una pagina viene scaricata. Il processo di downloading di una pagina inizia quando un utente clicca su un link a questa pagina. Viene, quindi inviata una *GetRequest* all'URL (Universal Request Locator)

indicato dal link e viene stabilita una connessione TCP tra l'utente e il web server, per trasmettere tale richiesta. In genere una pagina web è costituita dal codice HTML, l'oggetto principale (main object), e da altri oggetti (della classe Java), detti inline objects. Ricevuta la richiesta, il server invia l'oggetto principale al browser, a questo punto intervengono le diverse versioni HTTP, che presentano differenti modalità nello stabilire le connessioni TCP.

Le prime versioni prevedevano l'instaurazione di una connessione TCP per ogni oggetto inline e la trasmissione di tutti gli oggetti in successione. In versioni successive sulla stessa connessione TCP vengono trasmessi diversi inline objects ed è possibile l'instaurazione di più connessioni in parallelo (HTTP 1.0).

La versione più recente (HTTP 1.1) prevede la possibilità di lasciare attiva una connessione TCP anche dopo che la pagina è stata completamente trasmessa, consentendo una trasmissione di tipo pipeline. È ragionevole ritenere che la prossima pagina desiderata si trovi sullo stesso server, utilizzando così per la nuova GetRequest la connessione TCP esistente, piuttosto che instaurarne un'altra, con notevole miglioramento delle prestazioni.

La complessità del meccanismo di generazione del traffico WEB richiede una descrizione dettagliata che tenga conto delle differenti componenti o livelli esistenti dietro una sessione di WEB browsing. La modellizzazione strutturale del traffico porta, in questo caso, ad una struttura detta di tipo multilivello.

Il modello consiste dei seguenti livelli [22], [23]:

Livello di sessione: rappresenta la sessione di utilizzo del WEB browser da parte di un utente, dall'istante di inizio del web browsing sino alla fine della navigazione. Questo livello consiste di pagine considerate non sovrappoventesi.

Livello di pagina: visita di una pagina WEB, considerando una pagina come l'insieme di files (HTML, immagini, suoni, video,..) componenti la pagina WEB;

Livello di connessione: ogni pagina visitata è costituita da un insieme di oggetti che sono trasportati attraverso una o più connessioni TCP;

Livello di pacchetti: caratterizza la trasmissione di un pacchetto IP.

Ognuno di questi livelli è caratterizzato dai seguenti parametri.

Livello di sessione :

Tempo di interarrivo delle sessioni (è il tempo tra l'inizio di due sessioni consecutive);

Pagine per sessione (il numero di pagine che un utente visita in una sessione);

Livello di pagina :

Tempo tra due pagine (è il tempo tra la fine di una pagina e l'inizio della successiva nella stessa sessione);

Dimensione della pagina (quantità di informazione trasferita in una pagina. In genere i valori per la direzione uplink e downlink sono diversi);

Livello di connessione :

Numero di connessioni per pagina ;

Tempo tra due connessioni consecutive;

Dimensione delle connessioni;

Livello di pacchetto:

Tempo di interarrivo dei pacchetti (tempo tra due pacchetti consecutivi nella stessa pagina. Tali valori sono in genere diversi per l' uplink e il downlink.);

Dimensione dei pacchetti (numero di bytes contenuti in ciascun pacchetto, anche questi cambiano con la direzione del traffico.).

Poiché da misure è emerso che non esistono rilevanti differenze tra le proprietà del traffico generato dalle versioni HTTP più recenti, la HTTP1.0 e la HTTP1.1, il modello utilizzato per la descrizione del traffico WEB non prevede la caratterizzazione del livello di connessione, [23].

4.5.2 Traffico FTP

L'altra applicazione Internet tipica è il trasferimento di file, realizzata mediante il protocollo FTP (File Transfer Protocol), i cui pacchetti IP sono in genere individuati dal source port number 20, [25].

La principale funzione di FTP è il trasferimento efficiente e affidabile di archivi tra host remoti. Esso consente anche di visualizzare il contenuto di sistemi di archiviazione remoti ed eventualmente di modificare o cancellare files ivi residenti. Per garantire l'affidabilità utilizza TCP come protocollo di trasporto.

Le misure di traffico effettuate mostrano che la presenza del traffico generato da sessioni FTP è in diminuzione rispetto al passato, per un maggiore utilizzo di HTTP.

Il modello di traffico proposto per la descrizione di una sessione FTP è ottenuto da quello valido per le sessioni HTTP. Una sessione FTP viene infatti modellata come una sessione HTTP con un'unica pagina WEB.

Tale modello presenta pertanto la stessa struttura a strati del modello di traffico WEB, naturalmente con inevitabili differenze in termini della quantità di traffico offerto alla rete, e, come già detto, con una frequenza di presentazione, di tale tipo di sorgente, inferiore.

4.5.3 Traffico Smtp

Il protocollo SMTP è il protocollo di comunicazione utilizzato per lo scambio di messaggi di posta elettronica, [25]. Esso offre i meccanismi per il trasferimento di messaggi in forma elettronica secondo modalità affidabili ed efficienti attraverso Internet. A tal fine utilizza TCP come protocollo di trasporto e adotta quindi una modalità di trasferimento con connessione, che consente di soddisfare l'obiettivo di affidabilità.

Il modello di traffico utilizzato prevede la fase di arrivo dell'applicazione e generazione dei relativi pacchetti, e sarà descritto nella sezione successiva. I dati utilizzati per ottenere il modello sono anche qui quelli relativi a misure di traffico reale effettuate in una WLAN, [21], seguendo la "traccia" dei pacchetti SMTP individuati dal source port number 25.

Nel presente lavoro di tesi si è proceduto alla traduzione dei modelli di traffico suddetti in linguaggio C e alla simulazione del comportamento delle applicazioni tipiche individuate, al fine di ottenere dati verosimili sul carico di traffico offerto alla rete, da poter utilizzare per procedere alla valutazione delle prestazioni della rete in esame.

I modelli simulativi ottenuti sono descritti in dettaglio nel capitolo successivo. In questo sono anche riportati i grafici ottenuti dalle simulazioni che illustrano quanto detto finora riguardo al comportamento delle varie sorgenti.

CAPITOLO 4: MODELLI DI TRAFFICO	65
4.1 INTRODUZIONE.....	65
4.2 CARATTERIZZAZIONE E MODELLIZZAZIONE DELLE SORGENTI DI TRAFFICO	65
4.2.1 Caratterizzazione delle sorgenti.....	66
4.2.2 Modelli di Traffico	69
4.3 Traffico vocale	71
4.4 Traffico Video	73
4.5 Traffico dati	75
4.5.1 Traffico WEB.....	75
4.5.2 Traffico FTP.....	77
4.5.3 Traffico Smtip	78

CAPITOLO 5: MODELLI SIMULATIVI DELLE SORGENTI DI TRAFFICO

5.1 Introduzione

Di seguito, come già accennato, viene riportata una descrizione più dettagliata dei modelli di traffico utilizzati per descrivere il comportamento delle applicazioni tipiche considerate nel contesto in analisi, insieme al commento al codice C utilizzato per descrivere il traffico generato dalle suddette applicazioni. I valori considerati si riferiscono al singolo utente, si considera ad esempio la frequenza con cui un generico utente effettuerà una chiamata telefonica, o il numero di pagine che esso visiterà in una sessione HTTP e così via.

Inoltre va sottolineato che i valori dei parametri riportati si riferiscono al traffico entrante nella rete considerata. La caratterizzazione del traffico è stata considerata per la sola direzione downlink perché, come rilevato dalle misure su reti reali, è quella interessata dai maggiori volumi di traffico.

5.2 Sorgenti On-Off

Tra questo tipo di sorgente si è considerato il traffico vocale con codifica ADPCM. Il modello assunto [8], prevede un processo di arrivo delle chiamate di tipo esponenziale negativo, la cui funzione di densità di probabilità ha l'espressione seguente:

$$f(x) = \lambda * \exp(-x * \lambda) \quad (5.2.1)$$

dove λ rappresenta il valor medio della frequenza di interarrivo. In questo caso si tratta della frequenza media di interarrivo ed è pari a $5 * 10^{-3} \text{ s}^{-1}$.

La durata di ogni chiamata è stata assunta caratterizzata dalla distribuzione Gamma, l'espressione della relativa funzione di densità di probabilità è:

$$f(x) = \frac{(x^{(\alpha-1)}) * \exp(-\frac{x}{\beta})}{(\beta^\alpha) * \Gamma(\alpha)} \quad (5.2.2)$$

in cui α e β , sono i parametri tipici della distribuzione, legati al valor medio: m , e alla deviazione standard: σ .

In particolare risulta:

$$m = \alpha * \beta \quad (5.2.3)$$

$$\sigma^2 = \alpha * \beta^2 \quad (5.2.4)$$

Nel caso in esame con valor medio pari a 98s e deviazione standard pari a 100s, si ottiene dalle 5.2.3 e 5.2.4:

$$\alpha = \frac{m^2}{\sigma^2} = 0.960 \quad (5.2.5)$$

$$\beta = \frac{\sigma^2}{m} = 102.04s \quad (5.2.6)$$

Nella figura 5.1 è illustrata la funzione densità di probabilità Gamma che descrive la durata di una chiamata dati i parametri α e β definiti dalle 5.2.5 e 5.2.6.

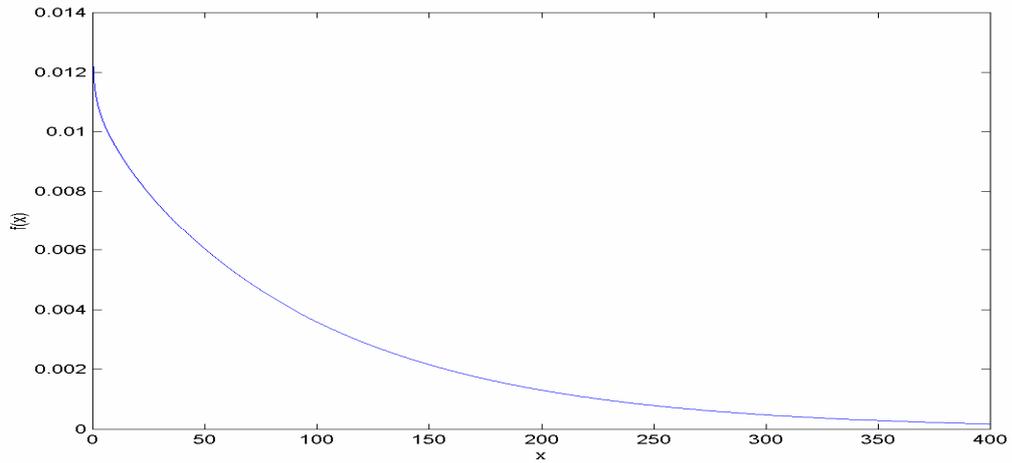


Fig. 5.1 Funzione densità di probabilità della durata di una sessione On-Off: distribuzione Gamma

Per la descrizione del processo di generazione del traffico offerto alla rete, si sono considerati i valori medi della durata dei periodi di attività e di silenzio, caratterizzati da distribuzione esponenziale negativa, la cui espressione è fornita dalla 5.2.1 e disponibili in letteratura, [8].

In particolare: $\bar{T}_{ON}=352*10^{-3}$ s e $\bar{T}_{OFF}=625*10^{-3}$ s.

L'altro parametro necessario per la caratterizzazione della sorgente è la lunghezza dei pacchetti IP, per la quale si è considerato il tipico valore di 72 bytes del pacchetto voce.

Il valore del rate di picco è pari a 32kbit/s.

Nella tabella 5.1 sono riportati i parametri utilizzati nel modello di una sorgente On-Off considerato.

T_{on} (s)	$352*10^{-3}$
T_{off} (s)	$625*10^{-3}$
Rate di picco (bit/s)	$32*10^3$
Rate medio (bit/s)	$11.2*10^2$
Durata media (s)	100
L_p (Byte)	72
λ (s^{-1})	$5*10^{-3}$

Tabella 5.1

A partire dalla conoscenza dei dati suddetti si è simulato il processo di generazione dei pacchetti IP, nei periodi di attività presenti nell'arco di una chiamata con una data durata temporale. In particolare estratta la variabile aleatoria rappresentativa della durata, sono stati calcolati la dimensione e gli istanti di generazione dei pacchetti, tenendo conto per questi ultimi del tempo di "produzione" del singolo pacchetto (dato dal rapporto tra lunghezza del pacchetto e rate di picco) e dell'alternarsi dei periodi di attività e di silenzio, i pacchetti sono trasmessi nei soli periodi di attività. Il processo di generazione dei pacchetti si arresta con la fine del tempo di chiamata. Tale modello è stato tradotto in linguaggio C e il codice relativo è riportato nell'appendice A, paragrafo A.1.

I dati ottenuti dalla funzione C che descrive il modello della sorgente On-Off, relativi alla dimensione dei vari pacchetti generati e all'istante di trasmissione degli stessi, sono stati graficati con l'ausilio di MATLAB. Dai grafici riportati di seguito appare evidente il carattere bursty della sorgente con l'alternarsi dei periodi TON e TOFF.

Si è considerato inoltre un intervallo di osservazione di 3000 secondi al fine di osservare l'andamento nel tempo del carico di traffico offerto da tale tipo di sorgente.

Nel considerare il processo di generazione del traffico di una singola sorgente voce ADPCM, si è scelto di considerare chiamate non 'sovrappontesi'. Se una chiamata inizia prima del termine della precedente, si assume il suo istante di inizio coincidente con l'istante finale della prima.

Di seguito sono allegati i grafici che illustrano alcune caratteristiche del traffico generato dalla sorgente ON-OFF considerata, nell'intervallo suddetto.

L'andamento complessivo del traffico generato da una sorgente ON-OFF nell'intervallo di osservazione è illustrato nella figura 5.2.

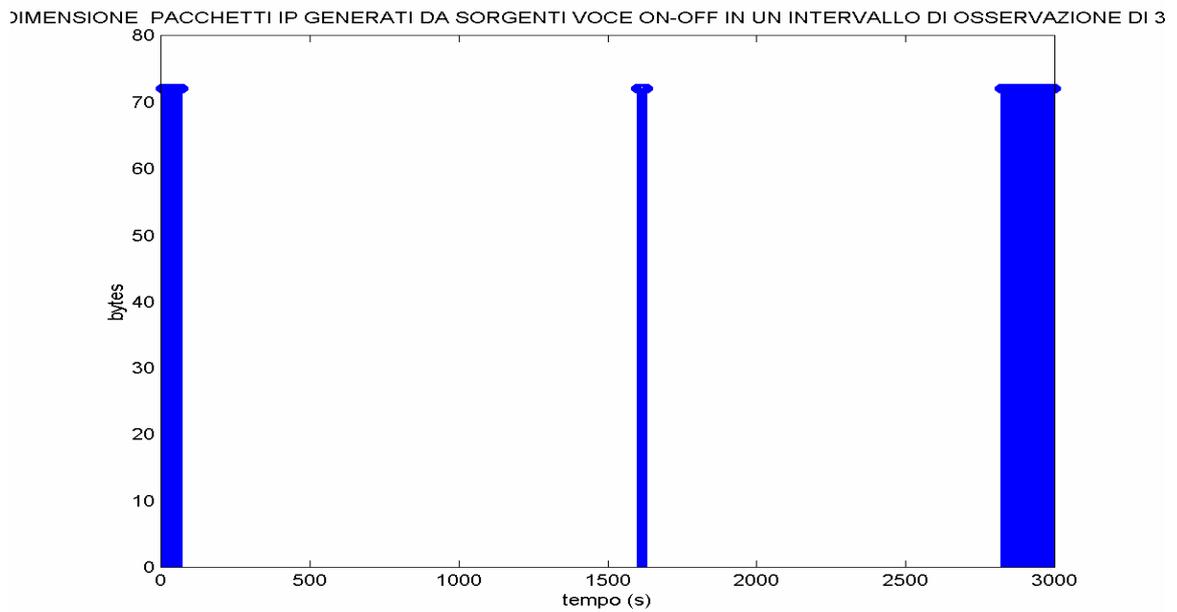


Fig. 5.2 Nascita di sorgenti On-Off su un intervallo di osservazione di 3000s

Appare evidente come durante l'intervallo di osservazione ci siano più chiamate con differente durata e differenti tempi di interarrivo.

La figura 5.3 illustra l'andamento delle dimensioni dei pacchetti IP generati da una sorgente ON-OFF durante una sola sessione e l'istante di trasmissione degli stessi.



Fig.5.3 Traffico generato da una sorgente On-Off

Appare evidente il carattere On-Off della sorgente con l'alternarsi dei periodi di attività e di silenzio, e come tali periodi abbiano durate differenti.

Nella figura 5.4 viene illustrato un tratto dell'intervallo di osservazione per meglio evidenziare le caratteristiche di emissione ON-OFF della sorgente, ossia l'alternarsi dei periodi di attività e silenzio:

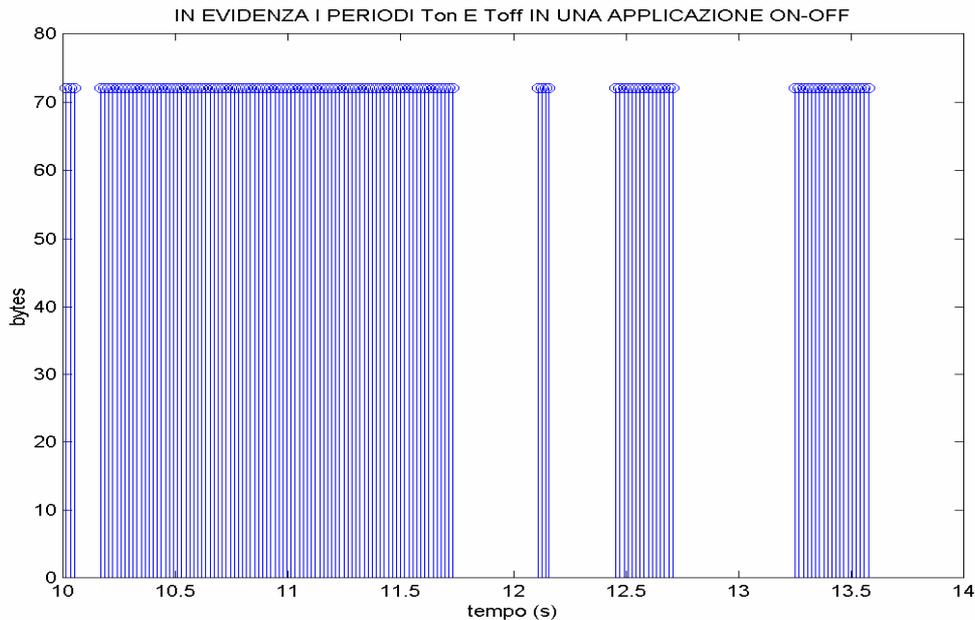


Fig.5.4 Particolare dell'andamento del traffico di una sorgente On-Off

5.3 Sorgenti HTTP

Il modello considerato come già detto è di tipo multilivello, [21].

Il primo livello è quello di sessione che individua la frequenza di interarrivo delle applicazioni HTTP, si assume una distribuzione esponenziale negativa con valore medio pari a $\lambda s = 5 \cdot 10^{-4} \text{ s}^{-1}$.

Individuato l'istante di nascita della sessione viene estratto il valore della variabile aleatoria che descrive il numero di pagine visitate nella sessione

caratterizzato da una distribuzione LogNormale. L'espressione della funzione densità di probabilità della distribuzione LogNormale è data dalla 5.3.1:

$$f(x) = \frac{1}{x \cdot \sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left[-\frac{(\ln(x) - \mu)^2}{2 \cdot \sigma^2}\right] \quad (5.3.1)$$

in cui μ e σ sono i parametri della distribuzione.

La funzione di distribuzione LogNormale è direttamente ricavabile da quella Normale cui è legata dalla relazione:

$$F(x) = G\left(\frac{\ln(x) - \mu}{\sigma}\right) \quad (5.3.2)$$

dove $G()$ è proprio la funzione di distribuzione Normale. I parametri μ e σ si ricavano a partire dal valor medio: m e dalla deviazione standard: s della variabile a distribuzione LogNormale, attraverso le relazioni seguenti, σ dalla 5.3.3:

$$\sigma^2 = \ln\left(\frac{s^2}{m^2} + 1\right) \quad (5.3.3)$$

e μ dalla 5.3.4:

$$\mu = \ln(m) - \frac{\sigma^2}{2} \quad (5.3.4)$$

In genere la distribuzione LogNormale è indicata come $\text{Ln}(\mu, \sigma)$. Nel caso presente dati il valor medio di circa 25 pagine per sessione e la deviazione standard pari a 78.75 pagine per sessione, si ottiene:

$\sigma = 1.527$ e

$\mu = 2.085$

Nella figura 5.5 si riporta l'andamento della funzione di densità di probabilità della distribuzione LogNormale che descrive il numero di pagine generate in una sessione HTTP .

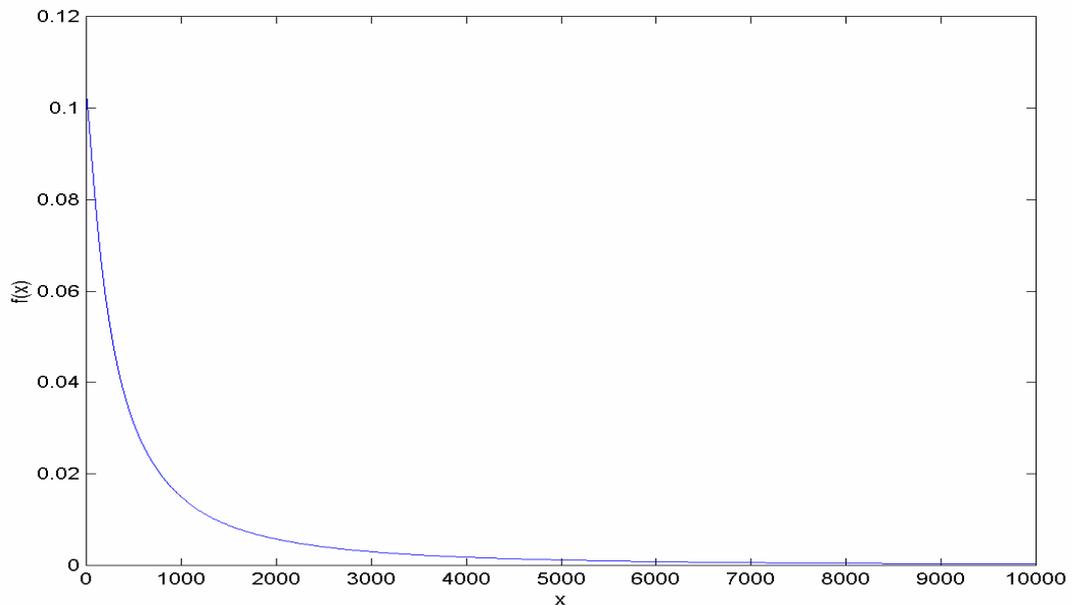


Fig.5.5 Funzione densità di probabilità del numero di pagine in una sessione HTTP : LogNormale

Per ogni pagina si passa quindi a calcolare la dimensione della stessa in bytes, mediante l'estrazione di una variabile aleatoria descritta dalla distribuzione di Pareto, con funzione di densità di probabilità data dalla 5.3.5:

$$f(x) = \alpha * \beta^\alpha * (\beta + x)^{-\alpha-1} \quad (5.3.5)$$

I parametri α e β sono ottenuti dal valor medio e dalla deviazione standard della variabile aleatoria, con la seguente relazione:

$$E(x^n) = \frac{\beta^n * \alpha}{\alpha - n} \quad (5.3.6)$$

con la condizione per $n < \alpha$;

dove $E(x^n)$ è il momento di ordine n . In media si hanno valori di circa 50kB per pagina, cui corrispondono i valori:

$$\alpha=1.55$$

$$\beta=31341.$$

Nella figura 5.6 è riportato l'andamento della funzione densità di probabilità di Pareto che descrive la dimensione delle pagine generate in una sessione HTTP.

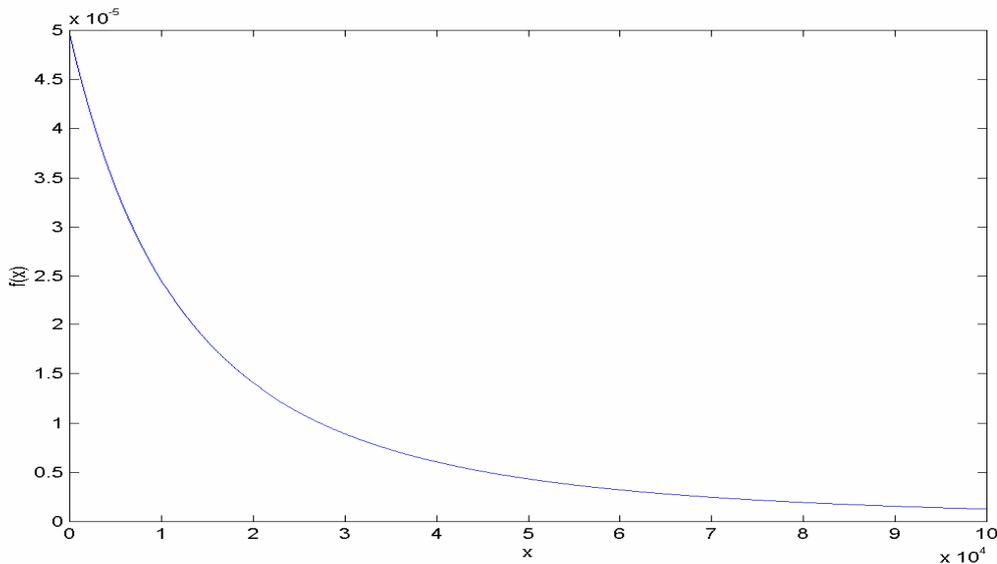


Fig. 5.6 Funzione densità di probabilità della dimensione delle pagine: distribuzione di Pareto

Nota tale valore si passa a considerare il livello più basso del modello di traffico, quello dei pacchetti IP generati all'interno della pagina. La dimensione di tali pacchetti segue una distribuzione Multimodale, assumendo con differenti probabilità quattro valori tipici, riportati nella tabella 5.2 ognuno con il relativo valore di percentuale di occorrenze,

Lunghezza Pacchetti (byte)	Percentuale di occorrenze
40	30.17
552	12.27
576	13.08
1500	44.48

Tabella 5.2

Il modello di traffico HTTP , [21], è stato tradotto in una funzione in linguaggio C che è riportata in appendice, nel paragrafo A.2, di seguito si riporta una descrizione dei passi di generazione del traffico secondo il modello considerato.

Il processo di generazione dei pacchetti prosegue tenendo conto della frequenza di interarrivo degli stessi, a distribuzione esponenziale negativa. Si procede estraendo la variabile aleatoria a distribuzione Multimodale, rappresentativa della dimensione del pacchetto, fino all' esaurimento dei bytes della pagina corrente. In particolare si è scelto di assumere l'ultimo pacchetto della pagina di dimensione pari ai bytes restanti, nel caso in cui questi fossero insufficienti per generare un pacchetto di lunghezza pari alla variabile estratta. Procedendo così si determina per ogni pacchetto la dimensione e l'istante di trasmissione dello stesso.

Esauriti i bytes della pagina corrente si procede al calcolo della dimensione della successiva e all'istante di nascita di questa, considerando, oltre al tempo necessario alla produzione dei pacchetti della stessa, anche il *viewing time*, ossia il tempo di lettura di una pagina. Anche per quest'ultimo è nota la descrizione statistica, caratterizzata da distribuzione Gamma, la cui espressione ha la forma già introdotta dalla 5.2.2, con valor medio di circa 30 secondi e deviazione standard di 40secondi, cui corrispondono i seguenti valori di α e β ,

$$\alpha=0.5625 \text{ e}$$

$$\beta=53.33.$$

La figura 5.7 riporta l'andamento della funzione densità di probabilità Gamma che descrive la distribuzione del tempo tra due pagine web consecutive, il valore di x è in secondi.

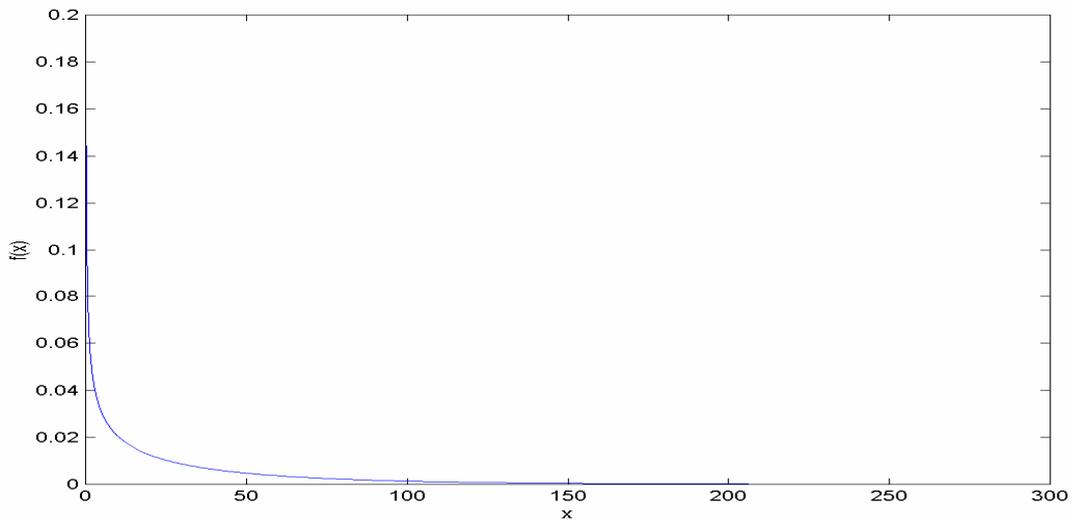


Fig. 5.7 Funzione densità di probabilità del tempo tra le pagine Web: distribuzione Gamma

Il processo di generazione del traffico offerto da una sessione HTTP termina quando sono state generate tutte le pagine e i relativi pacchetti.

Anche in questo caso si è considerato un intervallo temporale esteso allo scopo di osservare l'evoluzione del traffico HTTP generato in diverse sessioni, e si è ipotizzato di considerare sessioni non sovrappoventesi.

I parametri del modello utilizzati sono riportati nella tabella 5.3.

$\lambda_s(s^{-1})$	0.0005
Tempo di pagina (s)	35
$L_p(\text{byte})$	822.34
$\lambda_p(s^{-1})$	0.93

Tabella 5.3

I dati ottenuti sono stati graficati e di seguito sono riportati alcuni degli andamenti osservati. La figura 5.8 riporta la dimensione delle diverse pagine generate in una sessione HTTP.

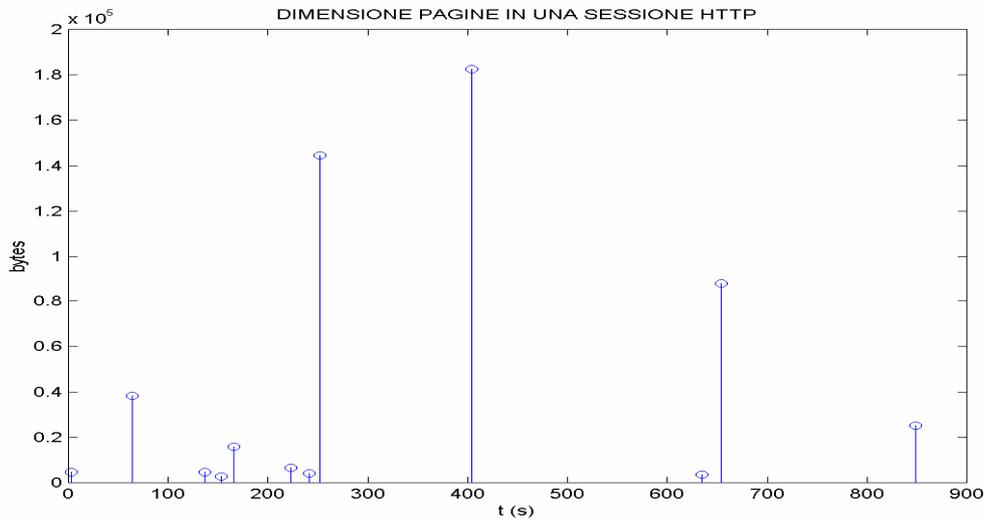


Fig. 5.8 Dimensione delle pagine di una sessione HTTP

La figura 5.9 illustra l'andamento delle dimensioni dei pacchetti IP generati in una sessione HTTP insieme all'istante di nascita degli stessi. Risulta l'alternarsi dei quattro valori delle dimensioni indicati nella tabella 5.2.

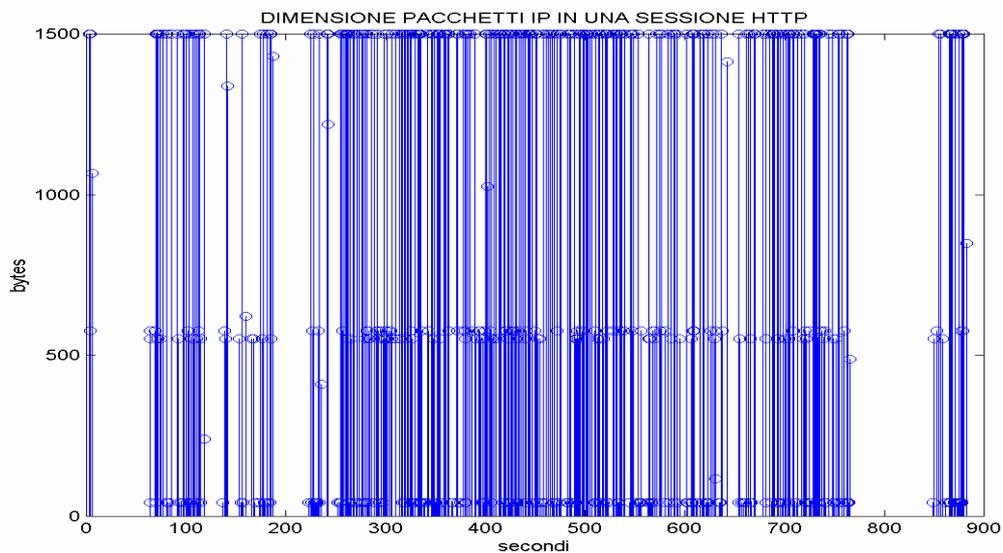


Fig.5.9 Dimensioni dei pacchetti IP generati in una sessione HTTP

Nella figura 5.10 è mostrato un particolare delle dimensioni e degli istanti di nascita dei pacchetti in una sessione HTTP, per evidenziare l'alternarsi dei diversi valori di lunghezza dei pacchetti IP, ed in particolare la predominanza del valore di 1500 bytes, che è appunto il più frequente nel traffico WEB in

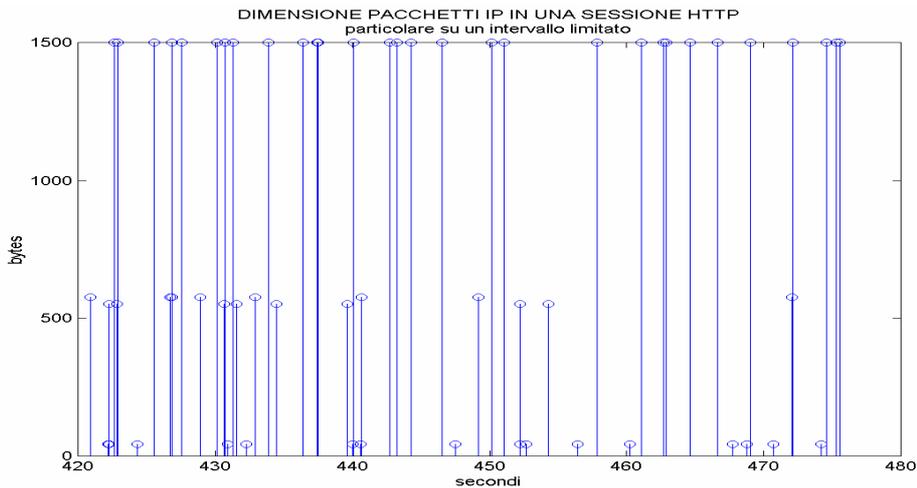


Fig. 5.10 Dimensione dei pacchetti IP in una sessione HTTP :particolare entrata nella rete come rilevato dalle misure citate in precedenza, [21].

Come detto prima si è osservato l'andamento del traffico WEB generato su un intervallo esteso e di seguito se ne riportano alcuni grafici. Nella figura 5.11 sono rappresentati la dimensione e gli istanti di nascita delle pagine WEB nell'intero intervallo di osservazione, in blu sono evidenziati gli istanti di nascita delle sessioni HTTP .

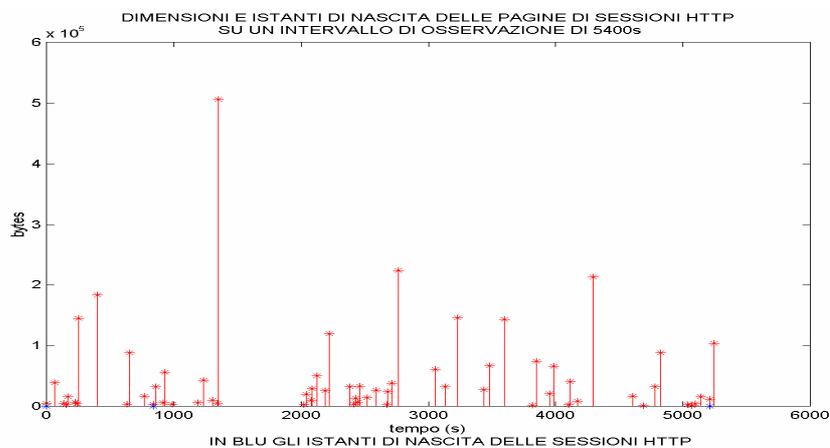


Fig. 5.11 Dimensione delle pagine generate in diverse sessioni HTTP

Nella figura 5.12 viene riportata la dimensione di tutti i pacchetti IP generati durante l'intervallo di osservazione nelle diverse sessioni HTTP, in essa sono evidenziati in rosso gli istanti di nascita delle diverse sessioni.

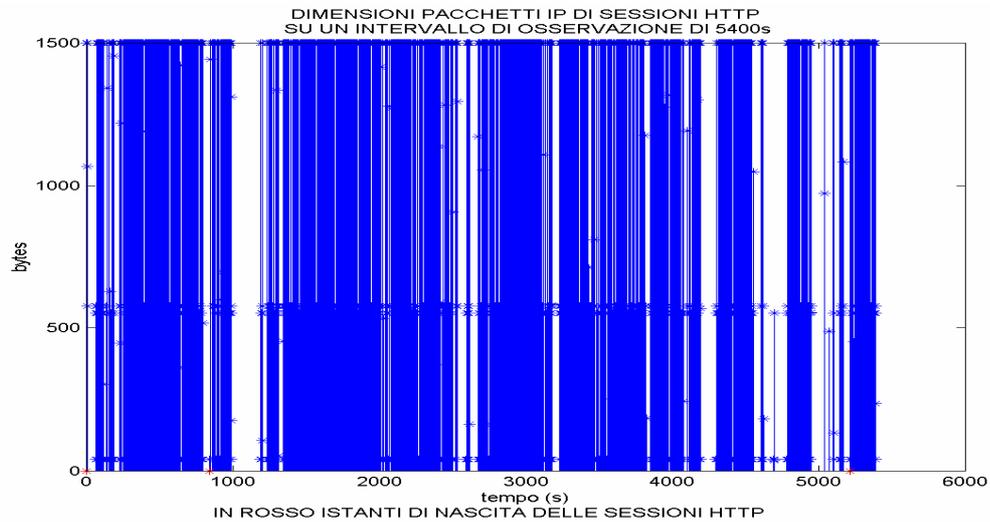


Fig. 5.12 Pacchetti IP generati in diverse sessioni HTTP

Per una visione complessiva si riporta la figura 5.13 in cui sono indicati gli istanti di nascita delle diverse sessioni HTTP, le dimensioni e gli istanti di nascita delle relative pagine ed infine dei pacchetti IP generati all'interno di queste.

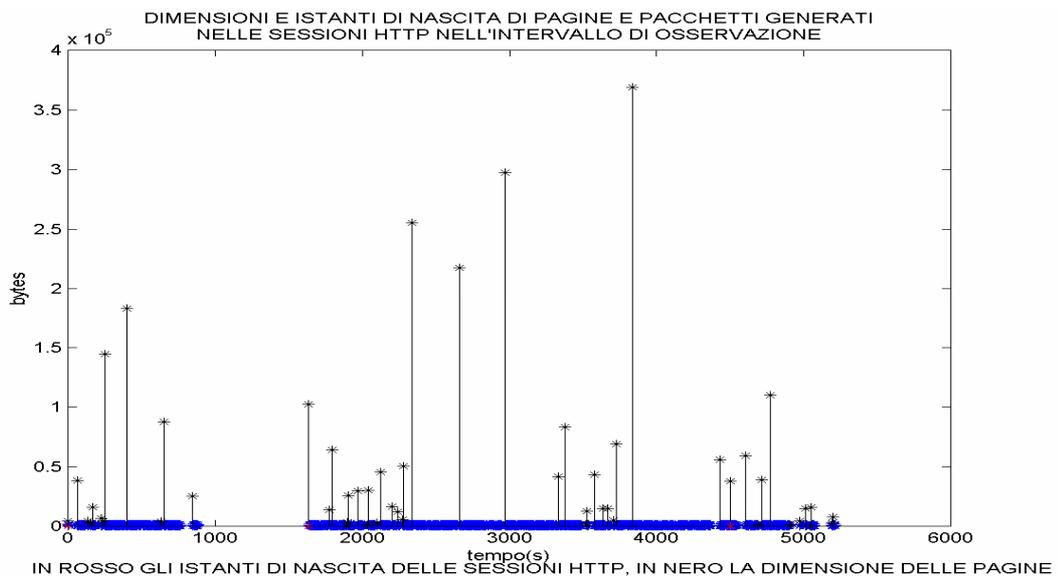


Fig. 5.13 Pagine e pacchetti IP generati nelle diverse sessioni HTTP

Infine nella figura 5.14 si riporta un particolare per evidenziare come sono generati i pacchetti all'interno di una pagina di una sessione WEB, in rosso sul grafico è segnato l'istante di inizio della sessione, mentre la linea nera è indicativa della dimensione della pagina corrente.

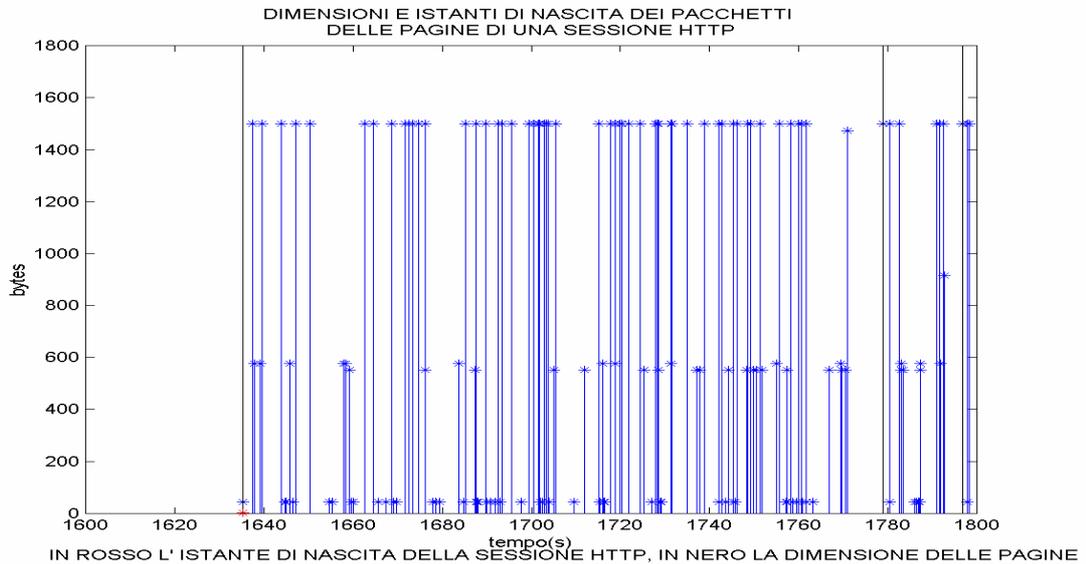


Fig. 5.14 Pacchetti in una pagina WEB

5.4 Sorgenti FTP

Come già detto in precedenza il modello adottato per il traffico FTP è analogo a quello HTTP, pertanto anche per la descrizione del codice ci si può riferire al precedente. Si ricorda comunque che il processo consiste di un'unica pagina WEB, con caratteristiche analoghe a quelle delle sessioni http, ma di dimensioni maggiori. La funzione di distribuzione della dimensione della pagina, qui l'intera sessione, è quella di Pareto con parametri:

$$\alpha = 1.754 \text{ e}$$

$$\beta = 183563,$$

a partire da un valor medio $m = 243373.9$ bytes per sessione.

La funzione densità di probabilità di Pareto che descrive la distribuzione di una sessione FTP è riportata nella figura 5.15, in questa x è espresso in byte.

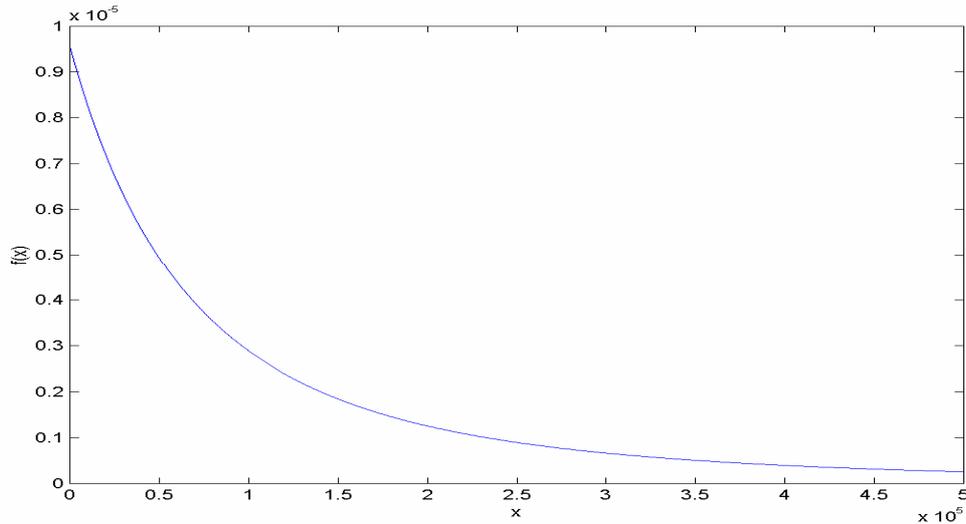


Fig. 5.15 Funzione densità di probabilità della durata di una sessione FTP: distribuzione di Pareto

Per le lunghezze dei pacchetti e i tempi di interarrivo dei pacchetti si assumono gli stessi valori considerati per una sessione HTTP , per questi si rimanda pertanto alla tabella 5.2.

Il tempo di interarrivo delle sessioni FTP, è sempre a distribuzione esponenziale negativa, ma con valor medio pari a $\lambda=6*10^{-5}$ sessioni per secondo.

Il processo di generazione del traffico avviene con le stesse modalità del caso di sessioni HTTP .

Anche in questo caso si è osservato l'andamento del traffico FTP generato, su un intervallo di tempo.

Nella tabella 5.4 sono riportati i dati caratteristici utilizzati per il modello delle sorgenti FTP.

$\lambda_s(s^{-1})$	0.00006
$\lambda_p(s^{-1})$	0.93
$L_p(\text{byte})$	822.34

Tabella 5.4

Nella figura 5.16, sono riportati gli istanti di nascita e le dimensioni delle sessioni nate nell'intervallo di osservazione.

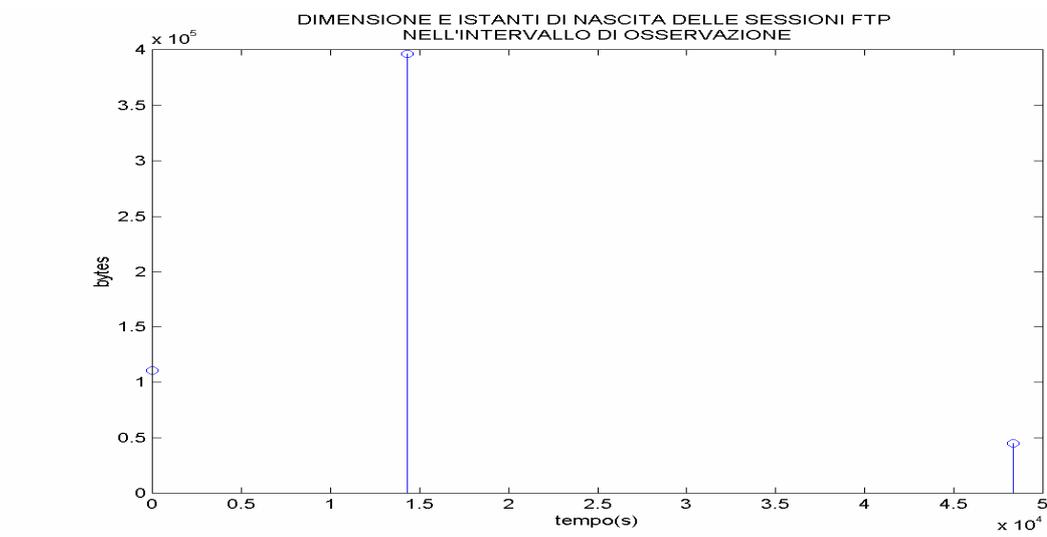


Fig. 5.16 Istanti di nascita e dimensioni delle sessioni FTP nell'intervallo di osservazione

In ogni sessione saranno generati i diversi pacchetti IP nella figura 5.17 è illustrato l'andamento delle dimensioni e gli istanti di nascita degli stessi.

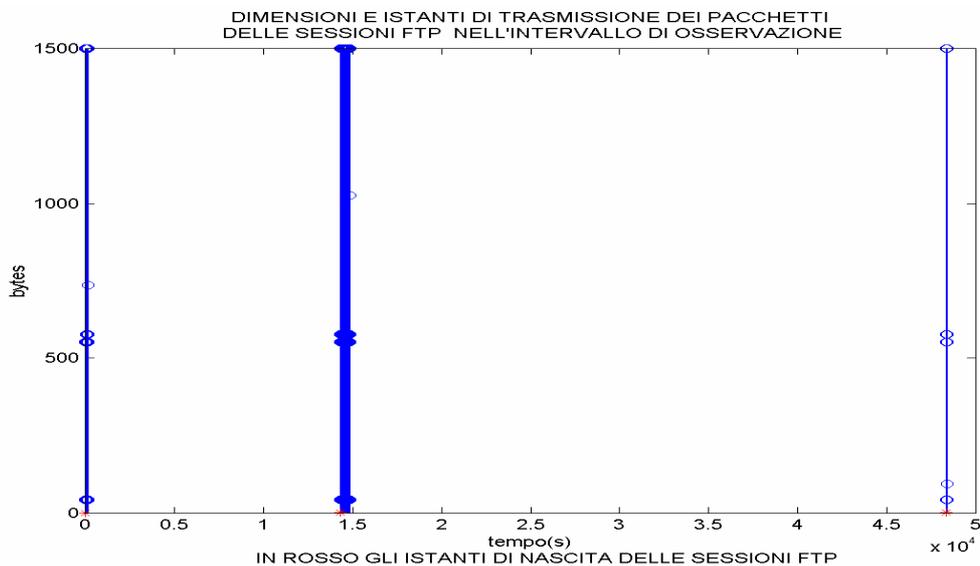


Fig. 5.17 Pacchetti IP generati nelle sessioni FTP

Infine nella figura 5.18 si riporta un tratto dell'intervallo di osservazione che evidenzia meglio le caratteristiche del traffico FTP a livello di pacchetto, in

particolare l'alternarsi i valori, riportati nella tabella 5.2, delle dimensioni dei pacchetti generati in una sessione FTP, il cui istante di inizio è evidenziato in rosso.

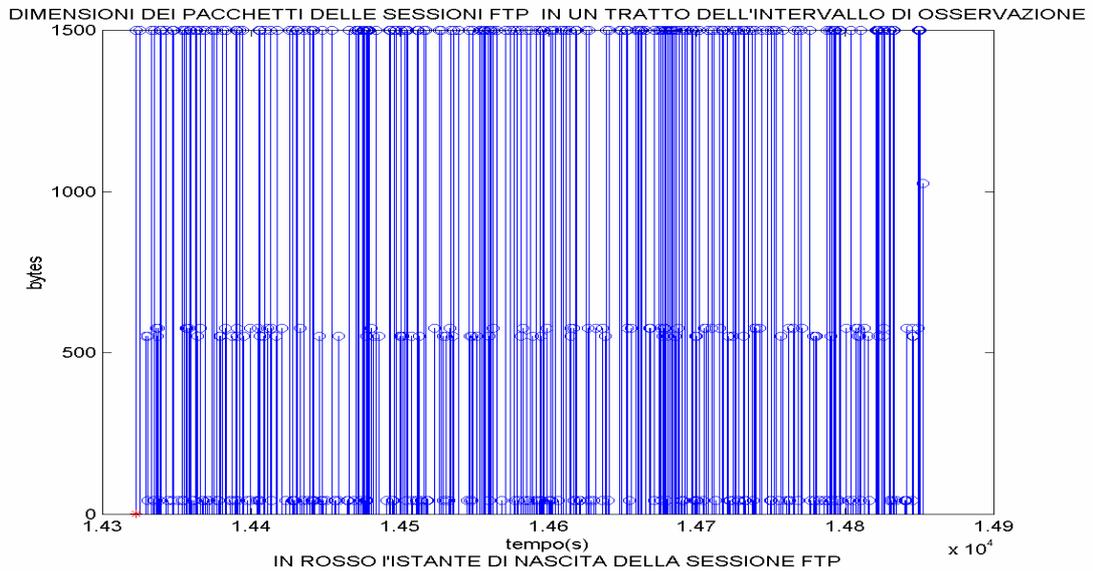


Fig. 5.18 Pacchetti IP generati in una sessione FTP

5.5 Sorgenti SMTP

Il modello del traffico di tipo SMTP considera il processo di nascita dell'applicazione di tipo esponenziale negativo, con una frequenza di interarrivo di 10⁻³ E-MAIL al secondo, [21]. La quantità di traffico generato da una sessione SMTP è descritto da una variabile aleatoria con distribuzione di Pareto, la cui funzione di densità di probabilità ha l'espressione seguente:

$$f(x) = \alpha * \beta^\alpha * (\beta + x)^{-(\alpha+1)} \quad (5.5.1)$$

con parametri $\alpha=2.009$ e $\beta=12710$ corrispondenti ad una dimensione di valor medio $m=25306$ bytes.

La figura 5.19 mostra l'andamento della funzione densità di probabilità di Pareto che caratterizza la dimensione di una sessione SMTP, la dimensione indicata con x è espressa in byte.

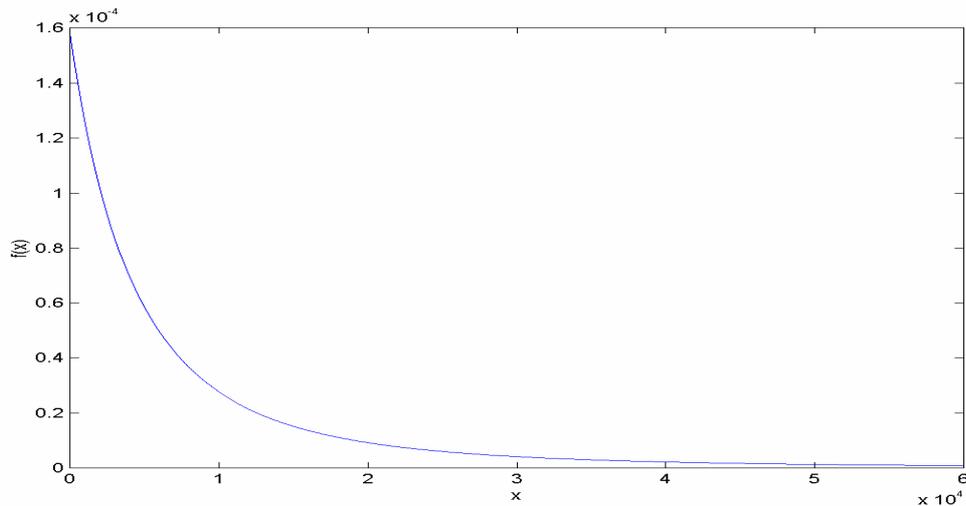


Fig.5.19 Funzione densità di probabilità della durata di una sessione SMTP: distribuzione di Pareto.

A partire dal valore della dimensione della sessione sono generati i pacchetti IP. Per questi è stata considerata una lunghezza media di 72 bytes e un tempo di interarrivo a distribuzione esponenziale negativa, con valor medio calcolato come rapporto tra la lunghezza media del pacchetto e il rate di picco di emissione, pari ad 5 kb/s.

esaurisce i bytes della sessione. L'ultimo pacchetto avrà lunghezza pari ai bytes restanti della sessione se questi non sono sufficienti per generare un pacchetto di 72 bytes.

Anche qui si assume che le sessioni del singolo utente siano non sovrappoventesi.

Nella tabella 5.5 sono riassunti i parametri principali utilizzati nel modello delle sorgenti SMTP.

$\lambda_s(s^{-1})$	0.001
$L_p(\text{byte})$	72
$\lambda_p(s^{-1})$	0.1152

Tabella 5.5

Di seguito sono riportati alcuni grafici che mostrano l'andamento del traffico SMTP.

Nella figura 5.20 sono mostrati gli istanti di inizio delle sessioni SMTP e le relative dimensioni nell'intervallo di osservazione considerato.

Appare evidente come le varie sessioni siano differenti per durata e dimensione

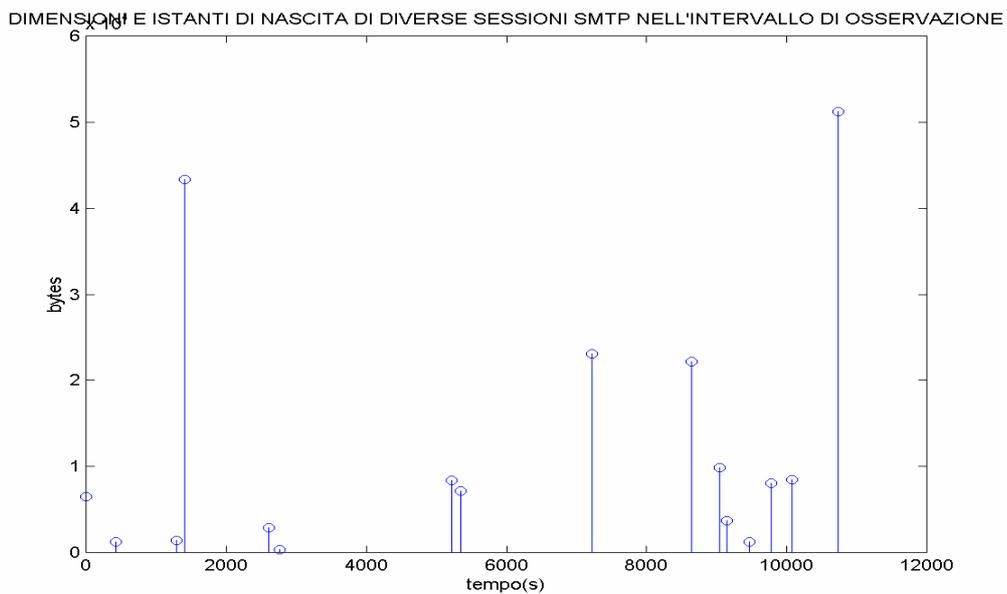


Fig. 5.20 Dimensioni sessioni SMTP nell'intervallo di osservazione

.Il processo di generazione del traffico SMTP a livello di pacchetto è illustrato nella figura 5.21, dove sono indicati lunghezza ed istante di trasmissione dei pacchetti e sono evidenziati in rosso gli istanti di inizio delle relative sessioni nate nell'intervallo di osservazione.

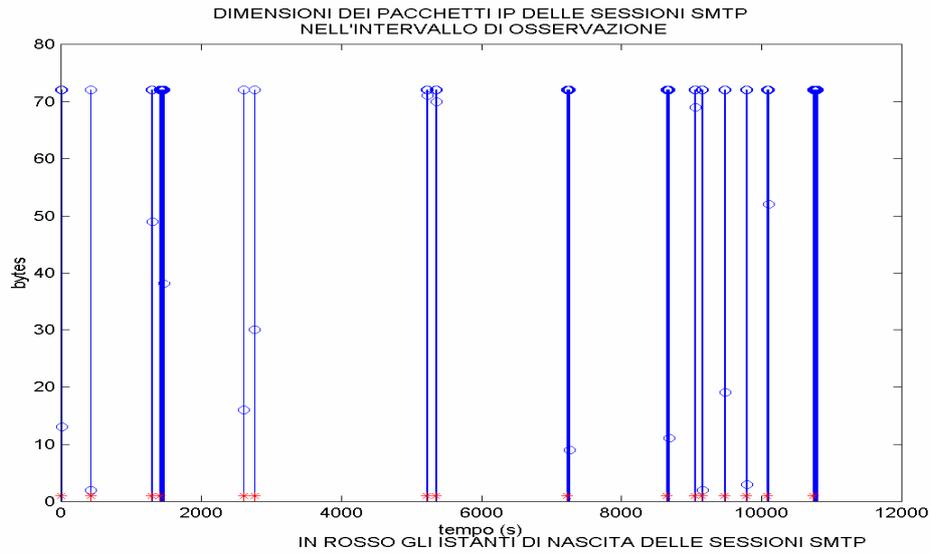


Fig. 5.21 Istanti di arrivo e dimensioni dei pacchetti IP delle sessioni SMTP nell'intervallo di osservazione

Infine si riporta un particolare del precedente grafico, per illustrare meglio le caratteristiche del traffico a livello di pacchetto. Nella figura 5.22 è mostrato l'andamento delle lunghezze e istanti di nascita dei pacchetti in una singola sessione.

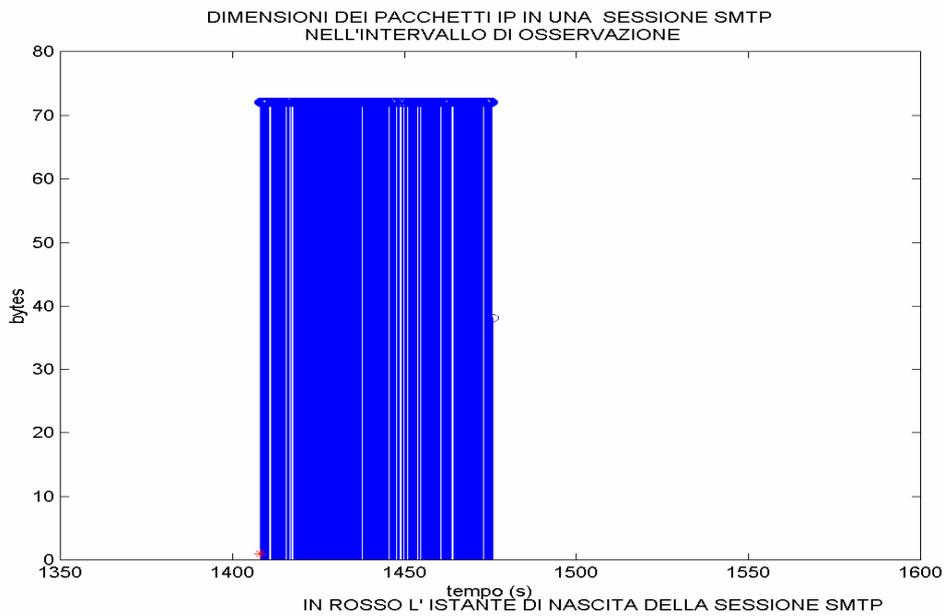


Fig.5.22 Dimensioni e istanti di arrivo dei pacchetti IP in una sessione SMTP

5.6 Sorgenti video MPEG

Per il modello di traffico assunto per le sorgenti video con codifica MPEG è stato un processo di nascita delle sessioni di tipo esponenziale negativo con una frequenza di interarrivo pari $\lambda=2*10^{-4}$ sequenze video al secondo.

La caratterizzazione del traffico video come già detto nel capitolo 5 avviene a livello di trama, ossia a partire dai dati delle misure di tracce video reali [26], [27], si è ottenuto che la dimensione delle trame è descritta da una variabile aleatoria con distribuzione LogNormale . Il valor medio della dimensione delle trame vale:

$m=5.5 *10^{-3}$ byte, mentre la varianza vale:

$s=5.5 *10^{-6}$.

L' espressione della funzione densità di probabilità della distribuzione LogNormale è data da:

$$f(x) = \frac{1}{x * \sigma * \sqrt{2 * \pi}} * \exp\left(-\frac{(\ln(x) - \mu)^2}{2 * \sigma^2}\right) \quad (5.6.1)$$

In cui i parametri della distribuzione assumono i seguenti valori:

$\mu=8.533$ e

$\sigma=0.398$.

Tali parametri sono legati al valor medio e alla varianza della variabile aleatoria che descrive la dimensione delle trame dalle espressioni 5.3.3 e 5.3.4 già riportate per le sorgenti HTTP .

Nella figura 5.23 è illustrato l'andamento della funzione densità di probabilità LogNormale utilizzata per la caratterizzazione della dimensione delle trame di una sequenza video.

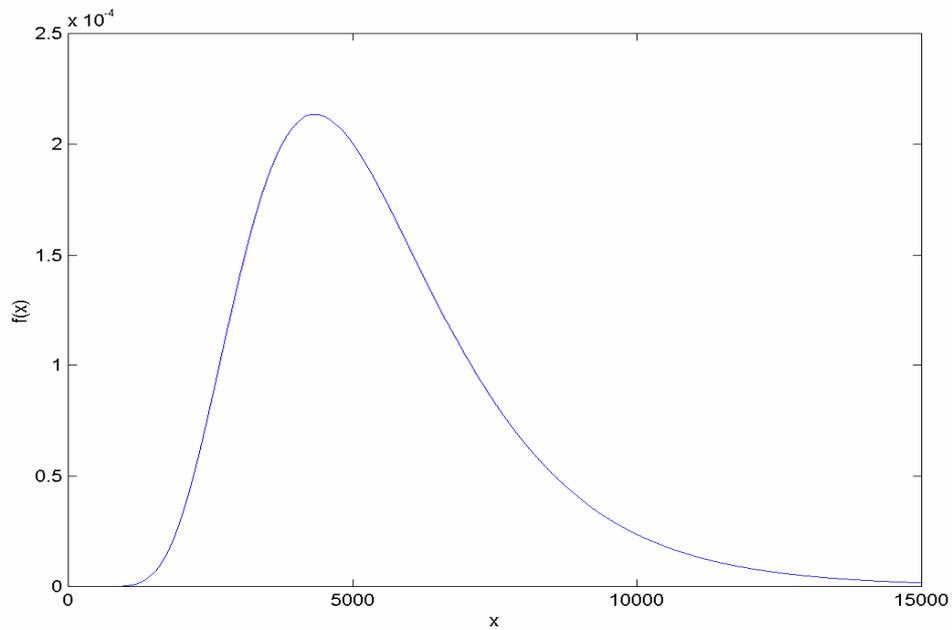


Fig. 5.23 Funzione densità della dimensione delle trame di una sequenza video : distribuzione LogNormale

Nella figura 5.24 è riportato l'andamento della funzione di distribuzione della dimensione di una sequenza video che si è rivelata in ottimo accordo con i dati ricavati dalle misure di traffico video ricordate in precedenza, [27].

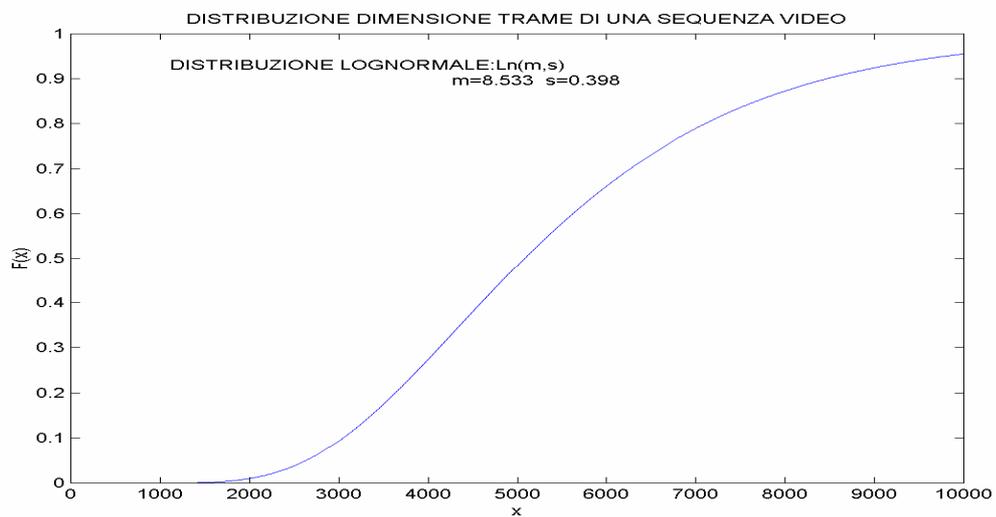


Fig. 5.24 Funzione distribuzione della dimensione delle trame di una sequenza video: Distribuzione LogNormale

La durata di una sequenza video è stata assunta descritta dalla distribuzione Gamma la cui espressione è data da:

$$f(x) = \frac{x^{(\alpha-1)} * \exp(-\frac{x}{\beta})}{\beta^\alpha * \Gamma(\alpha)} \quad (5.6.2)$$

Il valor medio della durata è pari a:

$m=1680$ s e la deviazione standard è :

$\sigma=2000$ s.

I parametri della distribuzione Gamma sono legati ad m e σ dalle 5.2.3 e 5.2.4 riportate nella trattazione delle sorgenti On-Off., ed hanno i seguenti valori:

$\alpha=0.71$ e $\beta=2380.9$.

Nella figura 5.24 si riporta l'andamento della funzione densità di probabilità che caratterizza la durata di una sequenza video.

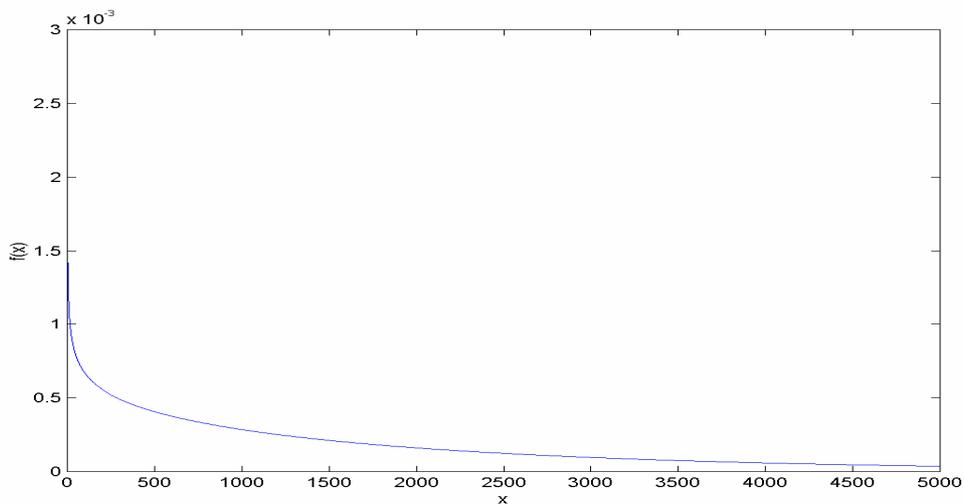


Fig. 5.24 Funzione densità di probabilità della durata di una sequenza video:distribuzione Gamma

Il modello di traffico delle sequenze video a codifica MPEG è stato tradotto come nel caso delle altre sorgenti in una funzione scritta in linguaggio C e riportata in appendice A paragrafo A.5. Grazie a tale funzione sono stati ottenuti gli andamenti del carico di traffico offerto da una sorgente video alla rete per un intervallo di osservazione di 300s.

I parametri del modello utilizzati sono riportati nella tabella 5.6.

$\lambda_s(s^{-1})$	0.0002
Tempo di sessione (s)	1680
Rate medio (bit/s)	$1.1 * 10^6$
Rate di picco (bit/s)	$3.6 * 10^6$
$N_t (s^{-1})$	25

Tabella 5.6

Il processo di interarrivo delle sessioni è stato assunto a distribuzione esponenziale negativa e, nella tabella 5.6, λ_s indica il valor medio della frequenza di interarrivo. Tempo di sessione indica la durata media di una sequenza video, mentre N_t è il numero di trame al secondo trasmesse in una sequenza video.

L'andamento del traffico generato a livello di trama è riportato nella figura 5.25.

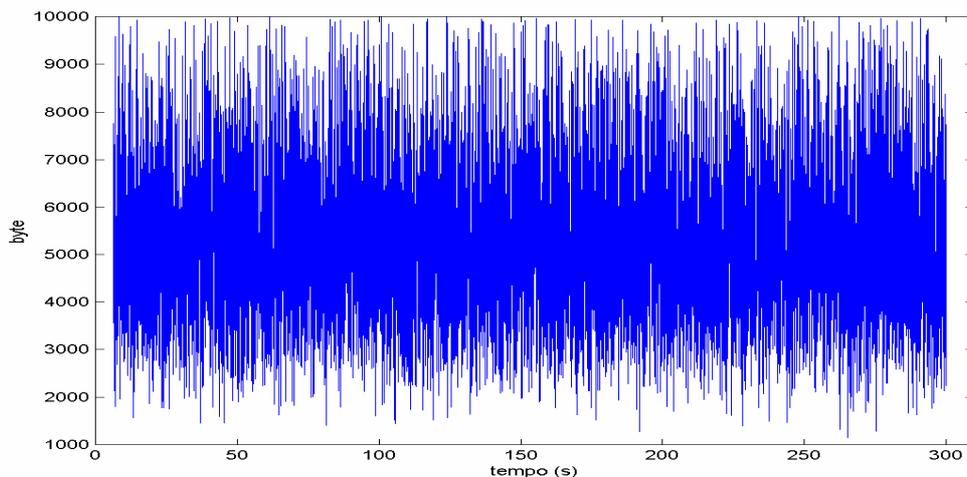


Fig.5.25 Dimensione delle trame generate da una sequenza video nell'intervallo di osservazione

E' evidente come sia notevole il volume di traffico offerto da una sorgente video considerato anche che l'intervallo di osservazione è piuttosto limitato . Ciò conferma quanto detto nel capitolo 4 a commento del sostanziale carico di traffico offerto alla rete da una sorgente video, inoltre l'andamento bursty conferma che non sarebbe efficiente, in termini di allocazione delle risorse, considerare la sorgente video come sorgente CBR..

CAPITOLO 5: MODELLI SIMULATIVI DELLE SORGENTI DI TRAFFICO.....	80
5.1 Introduzione.....	80
5.2 Sorgenti On-Off	80
5.3 Sorgenti HTTP.....	85
5.4 Sorgenti FTP.....	94
5.5 Sorgenti SMTP.....	97
5.6 Sorgenti video MPEG	101

CAPITOLO 6: DIMENSIONAMENTO DEI BUFFER

6.1 Introduzione

Come si è già sottolineato nei capitoli precedenti, a partire dai dati relativi al volume di traffico offerto ad una rete, da applicazioni con requisiti diversi di Qualità del Servizio, (QoS), si vuole procedere ad un dimensionamento delle risorse di rete, i buffer in particolare, che consentano il soddisfacimento di tali requisiti.

Il problema che si presenta è quello di dover gestire l'accesso ad una risorsa condivisa da diverse applicazioni modellabili come sorgenti di traffico. Tale scenario può essere descritto da particolari modelli matematici, chiamati sistemi di servizio o sistemi a coda, la cui trattazione matematica è nota come Teoria dei Sistemi di Servizio o Teoria delle Code. Tali modelli schematizzano l'interazione tra domande di servizio e soddisfacimento di tali domande, [28], [29].

I sistemi di servizio sono, infatti, utilizzati per la soluzione di problemi di valutazione delle prestazioni, di dimensionamento delle risorse e di gestione delle stesse in una infrastruttura per telecomunicazioni o di sue parti componenti.

La situazione che si presenta in generale è la seguente: le diverse sorgenti di traffico presentano alla rete le loro domande di servizio, caratterizzate da un certo volume di traffico e requisiti di QoS, la rete sarà caratterizzata dalle sue potenzialità di lavoro, ossia dalla propria capacità di soddisfare le richieste di servizio, mentre le attività di gestione provvedono alle modalità di trattamento delle domande di servizio nei limiti di quanto consentito dalle capacità della rete.

I parametri prestazionali che qualificano l'interazione tra attività di utilizzazione e risorse condivise, sotto il controllo delle attività di gestione, sono il rendimento di utilizzazione delle risorse condivise e la qualità del servizio che un'attività può sperimentare durante la sua evoluzione. Quest'ultima è quantificata attraverso dei parametri che descrivono il cosiddetto grado di accessibilità, cioè delle quantità che mettono in evidenza i limiti di operatività connessi all'utilizzo di risorse condivise, come ad esempio la probabilità di blocco o perdita di cui si dirà nel seguito.

Il problema del dimensionamento delle risorse consiste quindi nel determinare le capacità della rete, con il vincolo di minimizzazione dei relativi costi, quando siano noti le richieste di servizio, le funzioni di gestione e i parametri prestazionali.

Nel paragrafo successivo sono illustrati i concetti fondamentali della teoria delle code, allo scopo di comprendere le implicazioni di questa con il traffico offerto alla rete dalle varie applicazioni, nel dimensionamento dei buffer.

6.1.1 Teoria delle Code

Una coda o un sistema a coda, è un sistema stocastico il cui modello è rappresentato in figura 6.1. Viene illustrato in particolare un sistema a coda con un solo servente.

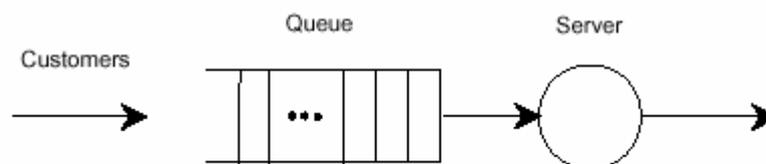


Fig.6.1 Schema di un sistema a coda ad unico servente

Il processo assunto alla base della maggior parte dei sistemi a coda è il seguente. Dei clienti (customers) richiedenti un dato servizio, sono generati nel tempo da una sorgente. Questi clienti entrando nel queueing system formano una coda. A certi istanti, un membro della coda viene scelto come prossimo cliente da servire, secondo una certa politica nota come disciplina della coda (per esempio la disciplina potrebbe essere FIFO, LIFO, etc.). Il servizio richiesto dal cliente viene quindi svolto dal server e il customer può uscire dal sistema a coda.

Il numero degli utenti e dei serventi è variabile, così come la dimensione della coda.

In generale quindi per caratterizzare un sistema a coda occorre specificare i processi di ingresso e di servizio, e la disciplina usata per gestire la coda.

Il processo di ingresso descrive le modalità statistiche con cui gli utenti presentano le loro richieste di servizio. Le variabili aleatorie usate in genere per tale descrizione sono i tempi di interarrivo, ossia l'intervallo di tempo che intercorre tra due richieste di servizio consecutive.

Le variabili aleatorie che descrivono il processo di servizio sono i tempi di servizio, esse costituiscono l'intervallo di tempo che un servente deve dedicare per soddisfare la richiesta di servizio di un utente.

La teoria di base dei sistemi di servizio utilizza una ipotesi semplificativa secondo la quale i processi di ingresso e di servizio siano entrambi processi di rinnovo, ossia le variabili aleatorie che compongono ciascuno di essi siano equidistribuite e statisticamente indipendenti. Tale ipotesi è ben rispondente alla realtà nel caso di applicazioni telefoniche, meno o del tutto inadeguata per altri tipi di applicazioni. Il vantaggio è tuttavia enorme, per una descrizione completa dei due processi aleatori è sufficiente fornire la funzione di distribuzione del primo ordine per una variabile aleatoria. In questo caso le variabili considerate sono il generico tempo di interarrivo e il generico tempo di servizio.

Un'ulteriore ipotesi è che i due processi detti siano statisticamente indipendenti. Il processo di coda è il processo aleatorio che risulta dall'interazione dei processi di ingresso e di servizio e costituisce la risposta del sistema alle richieste di servizio.

Se questo processo è ergodico nell'evoluzione temporale del sistema si distinguono un regime transitorio e un regime permanente (condizioni di equilibrio statistico). In genere l'analisi dei sistemi a coda è limitata allo studio del regime permanente, dopo aver verificato che siano soddisfatte le condizioni di ergodicità.

La caratterizzazione di un processo di coda in regime permanente avviene attraverso la definizione delle seguenti variabili aleatorie:

- k è il numero di utenti presenti all'interno del sistema a coda in un generico istante di tempo;
- δ è il tempo speso da un generico utente all'interno della coda, detto tempo di coda;
- l è il numero di utenti in coda nel generico istante di osservazione;
- η è il tempo che un utente trascorre in fila d'attesa: tempo di attesa;
- h è il numero di utenti che stanno ricevendo servizio, ovvero il numero di serventi contemporaneamente occupati nel generico istante di osservazione.

L'ipotesi di regime permanente consente di considerare le variabili aleatorie k , l e h indipendenti dall'istante di osservazione, e le variabili δ e η indipendenti dall'ordine di ingresso dell'utente cui si riferiscono.

I processi di coda sono indicati con una notazione sintetica che fa uso di cinque simboli. Il primo ed il secondo si riferiscono rispettivamente al processo di ingresso e di servizio, e denotano il tipo di distribuzione dei tempi di interarrivo e di servizio, secondo le corrispondenze riportate nella tabella 6.1.

Simbolo	Distribuzione
M	Esponenziale negativa
E_r	Erlangiana
H_r	Iper-Esponenziale di ordine r
D	Deterministica
G	Generale

Tabella 6.1 Simboli usati nella notazione sintetica per i processi di coda

Il terzo, il quarto ed il quinto simbolo individuano, rispettivamente, il numero dei serventi, la lunghezza della fila d'attesa, ovvero la dimensione della coda, e la cardinalità della popolazione di utenti potenziali.

Ad esempio la notazione

G/M/3/8/100

indica un processo di coda risultante dall'interazione di un processo di ingresso a distribuzione generale, un processo di servizio a distribuzione esponenziale negativa, in un sistema di servizio, la cui struttura è caratterizzata da tre serventi, otto posti in fila d'attesa e 100 clienti potenziali.

6.1.2 Caratteristiche prestazionali e Traffico in un Sistema di Servizio

Se in un sistema a coda n è il numero di utenti potenziali, L la lunghezza della fila di attesa ed m il numero dei serventi, la coda potrebbe contenere al massimo $L + m$ utenti. Se il numero di utenti all'interno del sistema è uguale ad $L + m$ si dice che il sistema è nello stato di blocco. Eventuali richieste presentate quando il sistema è in tale stato non potranno essere accolte. Le code in cui si verifica una situazione di questo tipo sono dette a perdita. Il comportamento di un sistema a perdita può essere caratterizzato attraverso la probabilità di rifiuto. Essa descrive la probabilità condizionale che, in regime

permanente, il sistema si trovi nello stato di blocco quando viene presentata una richiesta di servizio.

La probabilità di rifiuto è espressa in funzione della frequenza media delle richieste di servizio offerte, Λ_0 , accolte Λ_s , e di quelle rifiutate in seguito allo stato di blocco del sistema, Λ_p . Tra queste sussiste in regime permanente la relazione ovvia:

$$\Lambda_o = \Lambda_s + \Lambda_p \quad (6.1.1)$$

L'espressione della probabilità di rifiuto è data dalla relazione 6.1.2

$$\Pi_p = \frac{\Lambda_p}{\Lambda_o} = \frac{\Lambda_o - \Lambda_s}{\Lambda_o} \quad (6.1.2)$$

Quando la lunghezza L della fila di attesa è diversa da zero, gli utenti che giungono nel sistema quando tutti i serventi sono occupati, ossia quando si verifica uno stato di blocco del servizio, devono attendere in coda prima di vedere soddisfatta la loro richiesta di servizio. In questo caso gli utenti sperimenteranno un ritardo prima di poter accedere ad un servente. Le code in cui si verificano situazioni di questo tipo sono dette ad attesa. La probabilità di ritardo caratterizza il comportamento del sistema quando si verifica l'evento di ritardo, ovvero quando una richiesta accolta nel sistema, trova il servizio nello stato di blocco. In questo caso gli arrivi precedenti hanno saturato la capienza del servizio, pertanto il nuovo arrivo viene posto in fila d'attesa e quindi ritardato.

In questo caso la frequenza media delle richieste di servizio accolte si divide in due componenti, una Λ_{si} si riferisce alle richieste accolte senza ritardo, l'altra Λ_{sr} rappresenta la frequenza media delle richieste di servizio accolte con ritardo. Con tali notazioni la probabilità di ritardo è espressa dalla 6.1.3:

$$\Pi_r = \frac{\Lambda_{sr}}{\Lambda_s} = \frac{\Lambda_s - \Lambda_{si}}{\Lambda_s} \quad (6.1.3)$$

Le probabilità di rifiuto e di ritardo fanno parte dei parametri prestazionali cui si è accennato nell'introduzione. In particolare esse sono indicatori del grado di accessibilità ad una risorsa condivisa quando le contese di utilizzazione sono risolte a perdita e/o ad attesa.

Per una descrizione qualitativa e quantitativa dei fenomeni di impiego delle risorse condivise si utilizza il concetto di traffico. Una misura dell'intensità di traffico che in un fissato intervallo di osservazione interessa un insieme di risorse condivise è rappresentata dal numero di risorse che sono contemporaneamente impegnate, in quell'istante, da parte delle attività di utilizzazione.

Si considerano in particolare l'intensità media del traffico smaltito, A_s , ossia il numero medio di serventi contemporaneamente occupati, data da:

$$A_s = \bar{h} = \Lambda_s * \bar{\theta} \quad (6.1.4)$$

in cui Λ_s è la frequenza media delle richieste di servizio accolte e $\bar{\theta}$ è il tempo medio di servizio.

Analogamente è l'intensità media del traffico offerto: A_o , data dalla 6.1.5:

$$A_o = \Lambda_o * \bar{\theta} \quad (6.1.5)$$

dove Λ_o è la frequenza media delle richieste di servizio offerte, dall'espressione 6.1.5 si deduce che l'intensità media del traffico offerto è definita come il numero medio di richieste di servizio offerte in un intervallo di tempo pari al tempo medio di servizio.

Infine l'intensità media del traffico perduto, A_p ; è pari a :

$$A_p = \Lambda_p * \bar{\theta} \quad (6.1.6)$$

Nella 6.1.6 Λ_p rappresenta la frequenza media delle richieste di servizio rifiutate.

A_p è definita quindi come il numero medio di richieste di servizio rifiutate durante il tempo di servizio. In regime permanente le intensità di traffico appena definite sono legate dalla relazione 6.1.7 seguente:

$$A_o = A_s + A_p \quad (6.1.7)$$

Le intensità di traffico sono numeri puri espressi convenzionalmente nell'unità di misura detta erlang (Erl)

L'altro parametro prestazionale che qualifica un sistema di servizio è il rendimento di utilizzazione dei serventi : ρ , che in condizioni di equilibrio statistico è così espresso:

$$\rho = \frac{A_s}{m} = \frac{\Lambda_s * \bar{\theta}}{m} = \frac{\bar{h}}{m} \quad (6.1.8)$$

Le intensità medie di traffico possono essere espresse in funzione della probabilità di rifiuto Π_p attraverso le relazioni seguenti, molto utili in fase di valutazione delle prestazioni e dimensionamento; per l'intensità del traffico perduto si ha:

$$A_p = \Pi_p * A_o = \Pi_p * \frac{\bar{\theta}}{\bar{\tau}} \quad (6.1.9)$$

e per l'intensità media del traffico smaltito l'espressione è la seguente:

$$A_s = (1 - \Pi_p) * A_0 = (1 - \Pi_p) * \frac{\bar{\theta}}{\bar{\tau}} \quad (6.1.10)$$

Nelle 6.1.9 e 6.1.10 il parametro $\bar{\tau}$ indica il valor medio del tempo di interarrivo delle richieste di servizio.

Le altre relazioni molto utili per l'analisi ed il progetto di un sistema di servizio sono costituiti dal cosiddetto risultato o teorema di Little. Esse legano le variabili aleatorie che descrivono le dimensioni delle componenti di una coda ai relativi tempi caratteristici, e hanno le seguenti espressioni:

$$\bar{h} = \Lambda_s * \bar{\theta} \quad (6.1.11)$$

$$\bar{l} = \Lambda_s * \bar{\eta} \quad (6.1.12)$$

$$\bar{k} = \Lambda_s * \bar{\delta} \quad (6.1.13)$$

6.1.3 Code con ingresso e servizio a distribuzione esponenziale

Se i tempi di interarrivo e di servizio sono descritti da distribuzione esponenziale, per cui risulta, per i tempi di interarrivo:

$$f_{\tau}(t) = \frac{1}{\bar{\tau}} * \exp\left(-\left(\frac{t}{\bar{\tau}}\right)\right) \quad (6.1.14)$$

e per i tempi di servizio:

$$f_{\theta}(t) = \frac{1}{\bar{\theta}} * \exp\left(-\left(\frac{t}{\bar{\theta}}\right)\right) \quad (6.1.15)$$

dalle ipotesi assunte, come già detto, di equidistribuzione e indipendenza statistica delle variabili aleatorie relative, segue che i relativi processi sono di tipo Markoviano, e quindi descrivibili come processi di nascita e di morte. Lo stato del processo di coda in un generico istante di osservazione è descritto in modo completo dal solo numero di utenti contemporaneamente presenti nel sistema in quell'istante. Infatti per l'assenza di memoria tipica delle distribuzioni esponenziali negative, sono superflui i dati relativi all'età dell'ultimo arrivo e dei servizi in corso.

Il sistema a coda assunto generalmente è del tipo M/M/1, caratterizzato da un solo servente, in cui gli utenti arrivano secondo un processo di Poisson con frequenza λ e la distribuzione dei tempi di servizio è esponenziale con valor medio $1/\mu$. Questo tipo di sistemi sono i più semplici da studiare e anche i più conservativi, in quanto tendono a sovradimensionare il sistema.

In questo caso per il calcolo dei parametri prestazionali e delle variabili aleatorie descrittive del processo di coda saranno utilizzate le espressioni relative al processo Markoviano a questo associato e derivate dalla teoria delle catene di Markov [28].

In particolare sono disponibili le espressioni della probabilità di blocco e delle funzioni di distribuzioni di probabilità di quelle variabili come il tempo di attesa in coda e il numero di utenti nel sistema che consentono di valutare le prestazioni del sistema e/o dimensionare opportunamente la dimensione della coda una volta noti i requisiti richiesti dalle applicazioni in termini di probabilità di perdita e ritardo accettabili. Infatti ricordiamo che il nostro scopo è quello di utilizzare i sistemi a coda per modellare dei comportamenti presenti in una rete dati, pertanto dobbiamo specificare a cosa corrispondono queste entità nel nostro ambiente. Considerando la figura 6.1.1 i clienti o utenti sono i pacchetti delle varie applicazioni che arrivano e devono essere assegnati al link per la

trasmissione, mentre la coda rappresenta il buffer associato al canale di trasmissione, tramite cui i pacchetti devono essere spediti.

Supponendo che i pacchetti abbiano dimensione variabile con valor medio di L bit e che il canale trasmissivo abbia capacità di trasmissione pari a C bit/sec, il tempo medio di trasmissione di un pacchetto è dato da:

$$\frac{1}{\mu} = \frac{L}{C} \quad (6.1.16)$$

Per le code M/M/1/L l'intensità media del traffico offerto è data da:

$$A_o = \frac{\lambda}{\mu} \quad (6.1.17)$$

A partire da questa quantità si può ricavare la probabilità di rifiuto, Π_p che è il parametro prestazionale di interesse:

$$\Pi_p = \frac{1 - A_o}{1 - A_o^{L+2}} * A_o^{L+1} \quad (6.1.18)$$

L'altro parametro di interesse è il ritardo sperimentato dai pacchetti che giunti nel sistema trovano il server già occupato, e quindi devono attendere in coda, l'espressione per questa quantità è la seguente:

$$\eta = \frac{1}{\mu} * \frac{A_o - (L+1) * A_o^{(L+1)} + L * A_o^{(L+2)}}{(1 - A_o^{(L+1)}) * (1 - A_o)} \quad (6.1.19)$$

6.1.4 Code con tempi servizio a distribuzione generale

Come già detto il modello di coda M/M/1 è poco adatto a descrivere l'interazione tra attività e risorse nel caso in cui si considerino applicazioni

diverse da quelle telefoniche, ed inoltre si rivela conservativo nel dimensionamento del buffer.

In generale si avranno sistemi del tipo M/G/1, ossia con arrivi sempre poissoniani, ma con tempi di servizio che seguono una distribuzione di tipo generale.

Nel caso in cui i clienti siano serviti secondo la disciplina FIFO (First In First Out), è possibile calcolare i parametri di interesse a partire dai momenti del primo e del secondo ordine della variabile aleatoria descrittiva del tempo di servizio, con le formule di Pollaczek-Khinchin (P-K), [28], una volta verificata la condizione sul coefficiente di utilizzazione ρ , cioè :

$$\rho = \lambda * \theta < 1 \quad (6.1.20)$$

Le formule P-K consentono di calcolare il tempo medio di attesa in coda e da questo con il teorema di Little, il numero medio di utenti in coda e quindi la probabilità di perdita, utile per valutazioni delle prestazioni.

Comunque bisogna evidenziare anche che, se per un dato sistema non è possibile conoscere il momento del primo e del secondo ordine del tempo di servizio, come in genere accade è giustificato l'utilizzo di un sistema M/M/1 come modello analitico del sistema in esame, in quanto esso porta eventualmente al sovra-dimensionamento del sistema.

6.2 Simulazioni: descrizione e obiettivi

Come già ricordato lo scopo del presente lavoro di tesi è quello di analizzare per il sistema in esame delle condizioni verosimili di carico di traffico offerto dalle varie applicazioni. Pertanto una volta individuati i modelli di traffico delle sorgenti descritti nei precedenti capitoli 4 e 5, utilizzando i modelli suddetti, si è voluto verificare con delle simulazioni quale fossero le prestazioni ottenibili in

termini di occupazione dei buffer e ritardi sperimentati dai pacchetti IP delle diverse applicazioni.

Le simulazioni sono state effettuate con l'ausilio di MATLAB, in particolare l'utilizzo delle funzioni riportate in appendice A nei paragrafi A.7 e A.8 , ha consentito una valutazione dell'occupazione dei buffer e dei ritardi di coda sperimentati dai pacchetti IP di diverse applicazioni.

I modelli di traffico suddetti, si ricorda sono stati tradotti in funzioni scritte in linguaggio C e riportate in appendice A. Con tali funzioni sono stati ottenuti, come evidenziato nel capitolo precedente, dati relativi all'andamento temporale del volume di traffico generato dalle diverse sorgenti, quali dimensione in bytes delle sessioni, dimensione e numero dei pacchetti IP di una applicazione , durata temporale della stessa. A partire da questi dati si è voluto analizzare l'occupazione di spazio di memoria richiesto dalle diverse sorgenti di traffico. Il modello assunto è quello di una coda a server singolo, caratterizzata da un dato rate di trasmissione, in cui arrivano i pacchetti IP trasmessi da una sorgente e la coda è gestita secondo la disciplina FIFO. Le simulazioni sono state effettuate considerando per ogni sorgente un intervallo di osservazione dello stato del buffer, indicato come T_{oss} , con valore pari circa alla durata della sessione e comunque tale da consentire lo svuotamento completo del relativo buffer. Si è assunto inoltre che le osservazioni dello stato del buffer avvengano ad intervalli regolari ed il cui valore è indicato con δ .Cioè ogni δ secondi il contenuto del buffer sarà aggiornato, con svuotamenti e/o eventuali nuovi arrivi di pacchetti.

Allo scopo di ottenere dei risultati più attendibili per ogni tipo di applicazioni sono state effettuate simulazioni relative a diverse sorgenti. Sono stati cioè osservati i comportamenti di più sorgenti dello stesso tipo, ma con durate temporali e volumi di traffico diversi per avere una panoramica di situazioni più ampia, in modo da fare scelte più significative.

Per ogni sorgente sono state fatte più simulazioni con rate di trasmissione del canale diversi. I valori ottenuti sono relativi ai valori massimi e medi dell'occupazione del buffer e dei ritardi di coda dei pacchetti IP. Al variare del rate di trasmissione sarà diverso il volume di traffico che si accumula nel buffer e quindi saranno variabili i tempi di attesa in coda dei pacchetti che sopraggiungono nel buffer.

Ricordando che l'obiettivo è quello di valutare l'impatto dei requisiti di qualità del servizio richiesti dalle diverse applicazioni, in termini di ritardo e perdita massimi tollerabili, sulle dimensioni di memoria necessarie, si procederà ad un confronto tra i dati ottenuti dai modelli reali con quelli previsti dai modelli teorici mutuati dalla teoria delle code.

In generale il modello teorico di coda di tipo $M/M/1/L$ assunto, per la caratterizzazione del traffico nelle reti di telecomunicazioni tende a sovradimensionare le risorse del sistema, l'obiettivo delle simulazioni con i dati di traffico reale sarà quello di sottolineare questo aspetto e condurre ad un dimensionamento con minore "spreco" di risorse.

6.3 Considerazioni sulla validità delle simulazioni

Nel paragrafo precedente è stato definito lo scenario di riferimento con la descrizione delle modalità con cui sono effettuate le simulazioni e del modello teorico utilizzato per realizzare un confronto con i dati ottenuti dalle simulazioni stesse. Ora si procederà con alcuni commenti sulla validità del simulatore utilizzato, che come già detto è stato realizzato in MATLAB e riportato in appendice A, paragrafi A:7 e A.8.

Poiché il modello teorico considera una coda $M/M/1$, quindi in pratica il caso di sorgente Poissoniana, si potrebbe procedere alla validazione del simulatore a disposizione considerando proprio una sorgente Poissoniana in ingresso allo stesso. Da tale simulazione ci si attenderebbe una rispondenza perfetta con il caso teorico: stiamo considerando infatti lo stesso tipo di sorgente. In realtà si è

già detto che la teoria si limita a considerare il solo regime permanente, per ottenere risultati identici dovrei aspettare di raggiungere le condizioni di regime, osservare il sistema per tempi molto lunghi. Nei casi reali, però ogni sorgente ha una fase di attività di durata limitata nel tempo, pertanto già si intuisce che i dati ottenuti dalle simulazioni non potranno rispecchiare completamente la teoria.

Le simulazioni daranno dei risultati differenti, rispetto a quanto previsto dalla teoria delle code in termini dell'occupazione media del buffer, anche con una sorgente Poissoniana che dovrebbe seguire quanto previsto dalla teoria che si riferisce appunto a sorgenti di questo tipo. Cosa fra l'altro intuibile perché le sorgenti reali non prolungano la loro durata per un tempo infinito, come dovrebbe essere per raggiungere lo stato di regime cui si riferiscono le formule della teoria delle code.

Inoltre va anche sottolineato che le sorgenti considerate non sono sorgenti Poissoniane e quindi a maggior ragione i risultati delle simulazioni non potranno concordare con la teoria delle code.

L'utilizzo della teoria delle code per dimensionare lo spazio di buffer necessario alle varie applicazioni non è utilizzabile in tutti i casi. D'altra parte nel nostro caso il calcolo dello spazio di buffer viene fatto a livello di sessione perché è a livello di sessione che vengono allocate le risorse dallo strato MAC del progetto in esame.

Quindi siamo ben lontani dalle ipotesi della teoria delle code, come già detto le varie sessioni hanno una durata limitata nel tempo, il calcolo dello spazio di memoria necessario ad ogni sorgente va fatto quando questa si presenta al sistema e limitatamente alla sua durata temporale effettiva. Pertanto non sorprenderà se con l'utilizzo della teoria delle code arriveremo a risultati non concordi con le esigenze delle applicazioni reali.

Nei paragrafi successivi sono riportati i risultati delle simulazioni relative alle sorgenti reali i cui modelli di traffico sono stati introdotti nei precedenti capitoli 4 e 5.

6.4 Sorgenti FTP

Di seguito sono riportati i dati relativi alle sorgenti FTP utilizzate nelle simulazioni effettuate al fine di dimensionare la capacità dei buffer in relazione alla quantità di traffico offerto dalle varie applicazioni e ai requisiti da queste richiesti.

Per ogni sorgente è riportata la descrizione delle caratteristiche di traffico in termini di durata temporale, dimensione in bytes della singola sessione, numero di pacchetti IP generati dalla stessa.

Inoltre è presente, per ogni sorgente, una tabella con i dati relativi alle condizioni di simulazione e ai valori ottenuti dalle stesse. In ogni tabella si riporta il rate di trasmissione del canale, assunto nella simulazione, il valore dell'intervallo tra le osservazioni, i dati ottenuti relativi all'occupazione media e massima del buffer, e i valori medi e massimi del ritardo di coda sperimentato dai vari pacchetti della sessione nelle condizioni di simulazione considerate.

Inoltre al fine di effettuare il confronto con quanto previsto dalla teoria è riportato il valore di Δ_B ossia della differenza tra il valore B_{mean} di occupazione media del buffer ottenuto con la simulazione e il valore calcolato a partire dalle formule della teoria delle code, ossia :

$$\bar{l} = \frac{\rho^2}{1-\rho} \quad (6.4.1)$$

Tale formula è quella utilizzata in fase di progettazione in cui si utilizzano le formule semplificate valide per le code di tipo M/M/1/ ∞ per determinare la lunghezza della coda a partire dai dati della sorgente e del rate di trasmissione del canale. Nella 6.4.1 ρ è il rendimento di utilizzazione che è dato dall'espressione:

$$\rho = \lambda_s * \bar{\theta} \quad (6.4.2)$$

in cui $\bar{\theta}$ è il tempo medio di servizio.

L'altro parametro presente nelle tabelle è la differenza: Δ_D tra il ritardo medio di coda dei pacchetti ottenuto dalle simulazioni e il valore calcolato dalla teoria con la seguente espressione:

$$\eta = \frac{\rho}{\mu} * \frac{1}{1-\rho} = \frac{\bar{l}}{\lambda_s} \quad (6.4.3)$$

valida per code di tipo M/M/1.

I valori relativi alla durata temporale, alla dimensione della sessione in termini di bytes e numero di pacchetti IP sono stati ottenuti dai modelli di traffico [21] trattati nei capitoli 4 e 5, che come già detto sono stati descritti in linguaggio C e i relativi codici sono riportati in appendice A, paragrafo A.3.

Per le sorgenti FTP, come per tutte le sorgenti dati in generale, il requisito di qualità del servizio più importante è quello relativo alla integrità informativa. Pertanto la valutazione dello spazio di buffer, da riservare a queste sorgenti, va fatta considerando il soddisfacimento del vincolo sulla probabilità di perdita che deve risultare inferiore ad un valore massimo dato.

Per consentire il confronto con il modello teorico si riportano le formule utilizzate e i valori ottenuti a partire da queste, [28], [29].

La teoria prevede la schematizzazione del traffico generato dalla sorgente di traffico FTP in esame secondo il modello di coda del tipo M/G/1/L, ossia coda con processo di ingresso Poissoniano, processo di servizio a distribuzione generale, un solo servente e una coda di dimensione pari ad L. Non disponendo dei valori dei momenti del secondo ordine della distribuzione dei tempi di servizio, in genere per progettare la dimensione della coda in base alla probabilità

di perdita desiderata, si procede, come già detto, con l'uso della formula relativa al caso di coda M/M/1, [28].

L'espressione della probabilità di perdita è data da :

$$\Pi_p = \frac{(1-\rho)}{1-\rho^{L+2}} * \rho^{L+1} \quad (6.4.4)$$

dove ρ rappresenta il rendimento di utilizzazione del server, è espresso in funzione dei parametri di traffico, ed è dato dalla 6.4.2.

in cui λ_p rappresenta la frequenza di interarrivo dei pacchetti e $\bar{\theta}$ il valor medio del tempo di servizio del generico utente, qui il pacchetto ed è ottenuto dalla seguente espressione:

$$\bar{\theta} = \frac{L_p}{R} \quad (6.4.5)$$

dove L_p è la dimensione media del pacchetto, riportata nel modello di traffico ed R è il rate di trasmissione del canale.

Per le sorgenti FTP risultano dal modello utilizzato [21], i seguenti valori :

$$L_p = 822.34 \text{ Byte ;}$$

$$\lambda_p = 0.93;$$

per questi valori si veda anche la tabella 5.3 del capitolo 5.

Nella tabella 6.2 sono riportati, per i diversi valori del rate di trasmissione considerati, i valori della dimensione del buffer previsti dal modello teorico, e calcolati secondo l'espressione 6.4.4 in corrispondenza a diversi valori della probabilità di perdita desiderata, e il valore del tempo di servizio calcolato con la 6.4.5.

In particolare il valore di L è ottenuto invertendo la 6.4.4, da cui si ottiene :

$$L = \frac{\ln\left(\frac{\Pi_p}{1-\rho}\right)}{\ln(\rho)} - 1 \quad (6.4.6)$$

Il valore di L ottenuto dalla 6.4.6 rappresenta il numero di pacchetti nella coda, mentre il valore riportato nella tabella 6.2 è espresso in bytes, ed è ottenuto moltiplicando il risultato della 6.4.6 per la dimensione media di pacchetto L_p .

In particolare L_1 si riferisce ad una probabilità di perdita pari a 10^{-4} , mentre L_2 ad una probabilità di perdita di 10^{-3} .

Rate(B/s)	θ (s)	$L_1(\Pi_p=10^{-4})$	$L_2(\Pi_p=10^{-3})$
5000	0.1645	3140	2131
8000	0.1034	2377	1568
48000	0.0170	1006	548.2
100000	0.0083	732.7	343.6

Tabella 6.2

Inoltre per ogni sorgente a partire dai dati relativi al traffico generato dalla stessa sono stati ricavati quanto vale in media la lunghezza di un pacchetto, che sarà indicata come L_{ps} , come rapporto tra la dimensione totale della sessione e il numero di pacchetti della stessa, e la frequenza sperimentale di interarrivo dei pacchetti come rapporto tra la durata della sessione e il numero di pacchetti trasmessi, e che sarà indicata come λ_{ps} .

Nella tabella 6.3 sono riportati i dati di traffico relativi alle sorgenti considerate e ricavati dalle funzioni in linguaggio C, già ricordate e presenti in appendice. In questa tabella sono presenti i simboli di cui si riporta il significato:

T_{oss} è la durata in secondi della sessione

D rappresenta la dimensione in bytes della sessione;

N è il numero di pacchetti trasmessi durante la sessione.

Da questi risulta:

$L_{ps} = D/N$ rappresenta la lunghezza media dei pacchetti trasmessi.

$\lambda_{ps} = N/T_{oss}$ è la frequenza di interarrivo dei pacchetti.

Sorgente	$T_{oss}(s)$	D(B)	N	$L_{ps}(B)$	λ_{ps}
S1	145	110590	128	863.98	0.88
S2	536	396418	477	831.06	0.89
S3	320	245067	319	768.23	0.99
S4	37	36148	38	951.26	1.03
S5	252	194965	217	898.45	0.86
S6	940	672397	861	780.95	0.92

Tabella 6.3

Seguono ora le tabelle con i dati relativi alle sorgenti considerate nelle simulazioni.

In esse B_{mean} e B_{max} indicano, rispettivamente il valore medio della occupazione del buffer e quello massimo, mentre D_{mean} e D_{max} il valore medio e massimo del ritardo di coda dei pacchetti della sorgente considerata e le relative differenze Δ_B e Δ_D .

I dati ottenuti dalle simulazioni e relativi alla sorgente S1 sono riportati nella tabella 6.4.

Rate (B/s)	$B_{max}(B)$	$B_{mean}(B)$	$\Delta_B(B)$	$D_{max}(s)$	$D_{mean}(s)$	$\Delta(s)$
5000	3052	113.35	90.320	0.610	0.1970	0.1673
8000	2752	111.06	102.76	0.344	0.1172	0.1063
48000	2076	76.269	76.055	0.043	0.0186	0.0183
100000	2076	76.269	76.220	0.280	0.0089	0.0088

Tabella 6.4

Nella tabella 6.5 sono riportati i dati delle simulazioni relativi alla sorgente S2

Rate (B/s)	$B_{\max}(B)$	$B_{\text{mean}}(B)$	$\Delta_B(B)$	$D_{\max}(s)$	$D_{\text{mean}}(s)$	$\Delta(s)$
5000	4000	154.84	132.11	0.80	0.190	0.1603
8000	3700	109.84	101.54	0.46	0.113	0.1021
48000	3000	73.96	73.746	0.062	0.018	0.0177
100000	3000	73.96	73.911	0.03	0.0087	0.0086

Tabella 6.5

La tabella 6.6 riporta i dati delle simulazioni relativi alla sorgente S3:

Rate (B/s)	$B_{\max}(B)$	$B_{\text{mean}}(B)$	$\Delta_B(B)$	$D_{\max}(s)$	$D_{\text{mean}}(s)$	$\Delta(s)$
5000	3000	153.62	130.89	0.600	0.172	0.1422
8000	3000	111.09	102.79	0.375	0.103	0.0921
48000	3000	76.58	76.366	0.062	0.016	0.0157
100000	3000	76.58	76.531	0.030	0.008	0.0079

Tabella 6.6

Infine la tabella 6.7 riporta i dati della sorgente S6:

Rate (B/s)	$B_{\max}(B)$	$B_{\text{mean}}(B)$	$\Delta_B(B)$	$D_{\max}(s)$	$D_{\text{mean}}(s)$	$\Delta(s)$
5000	4000	145.05	122.32	0.80	0.176	0.1463
8000	3576	103.73	95.43	0.44	0.105	0.0941
48000	3576	71.53	71.316	0.075	0.017	0.0167
100000	3576	71.53	71.481	0.036	0.008	0.0079

Tabella 6.7

I risultati delle simulazioni sono stati graficati e riportati nelle figure seguenti. In particolare nella figura 6.2 è illustrato come varia il valore massimo dell'occupazione del buffer al variare del rate di trasmissione per le sorgenti

considerate. Le diverse sorgenti sono identificate da colori diversi, il verde si riferisce alla sorgente S1, il rosso a S2, il nero ad S3 e il blu ad S6.

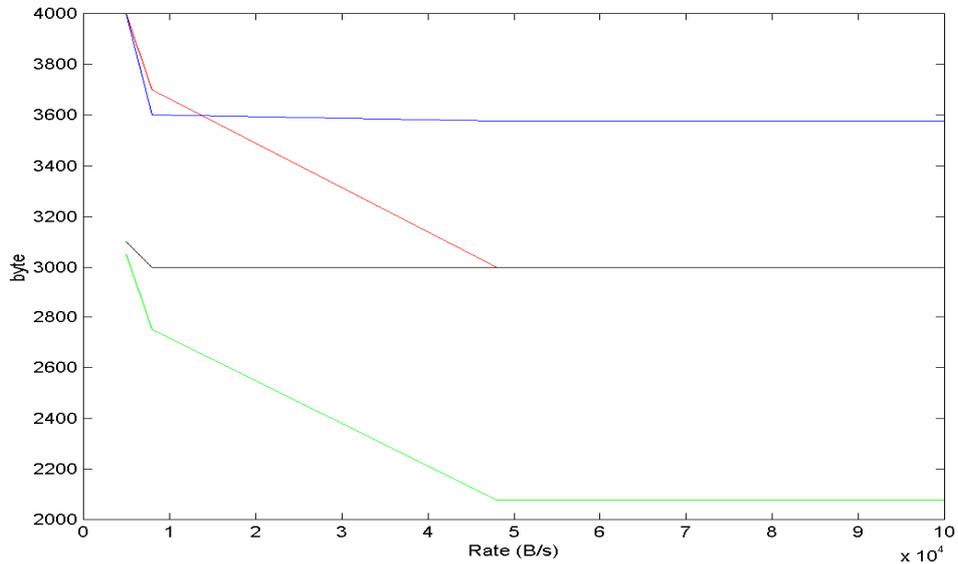


Fig.6.2 Occupazione massima del buffer: in verde S1, in rosso S2, in nero S3 e in blu S6

La figura 6.3 riporta invece l'andamento relativo all'occupazione media del buffer per le sorgenti dette al variare del rate di trasmissione, le varie sorgenti sono distinte dai colori diversi con le stesse corrispondenze della figura 6.2.

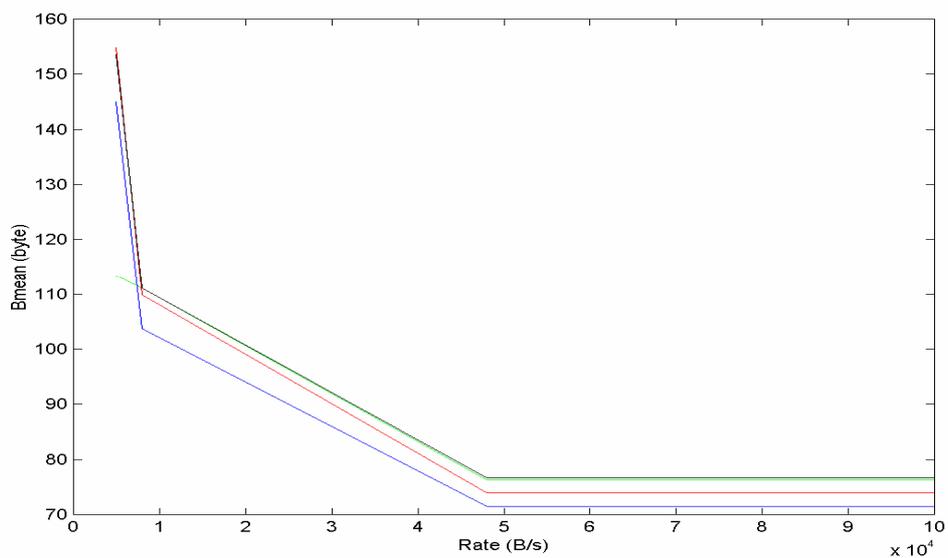


Fig.6.3 Occupazione media del buffer: in verde S1, in rosso S2, in nero S3 e in blu S6

Gli andamenti dei ritardi massimi e medi sperimentati dai pacchetti IP delle varie sorgenti sono riportati nelle figure 6.4 e 6.5 rispettivamente. Come per le figure 6.2 e 6.3 il verde si riferisce alla sorgente S1, il rosso ad S2, il nero ad S3 ed il blu ad S6.

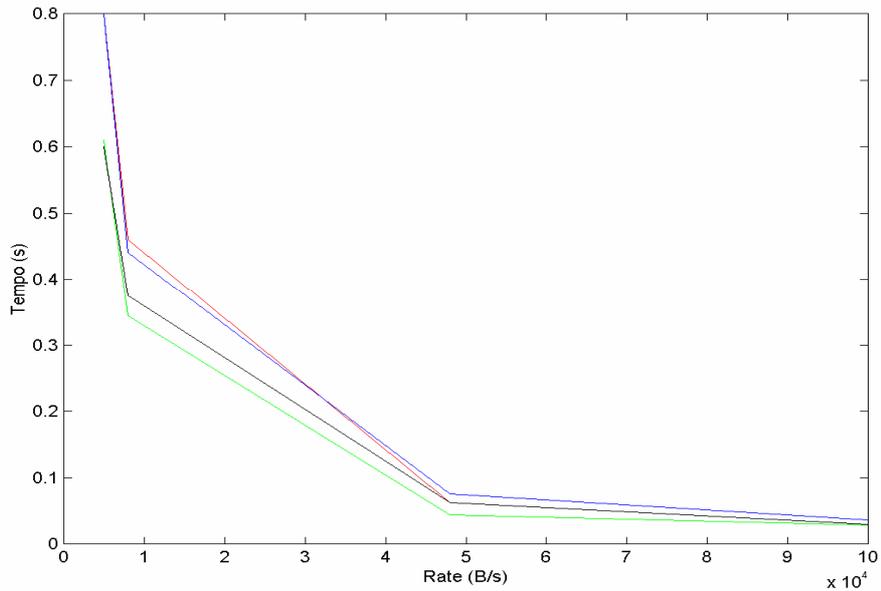


Fig.6.4 Ritardi massimi: in verde S1, in rosso S2, in nero S3 e in blu S6.

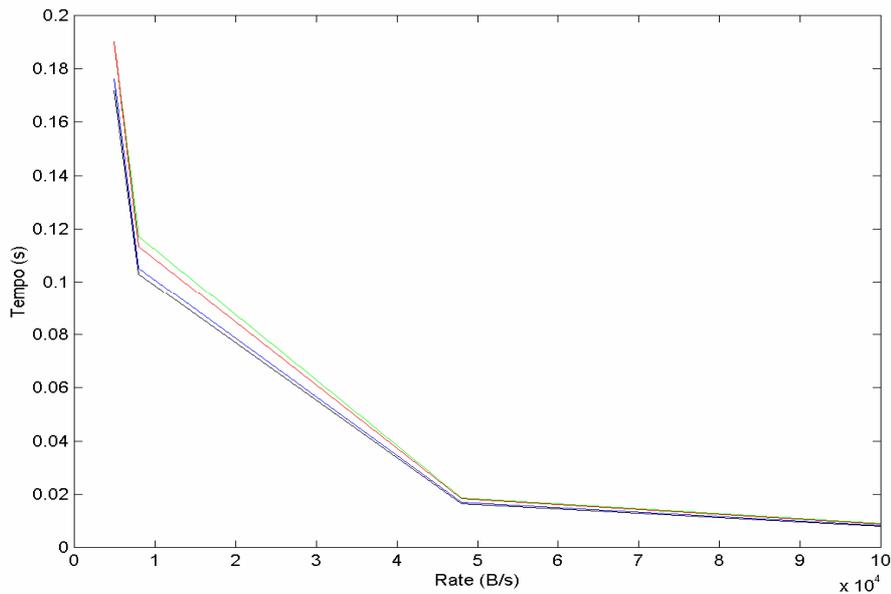


Fig.6.5 Ritardi medi: in verde S1, in rosso S2, in nero S3 e in blu S6

Utilizzando i dati sperimentali L_{ps} e λ_p calcolati per la sorgente S1 si sono ottenuti i valori relativi del tempo di servizio, indicati con θ_r e a partire da questi, con la 6.4.6 i valori delle dimensioni del buffer relativi ai valori di probabilità di perdita di 10^{-4} e 10^{-3} , indicati rispettivamente con L_{ps1} e L_{ps2} , al fine di effettuare un confronto con la teoria. I valori relativi sono riportati nella tabella 6.8.

Rate(B/s)	θ_r (s)	L_{s1} (B)	L_{s2} (B)
5000	0.1728	3284	2228
8000	0.1079	2480	1635
48000	0.0179	1057	576
100000	0.0086	765	358

Tabella 6.8

Dal confronto dei dati della tabella 6.8 con quelli delle tabelle 6.2, si rileva che il modello teorico porta ad un sovradimensionamento della risorsa nel caso si desiderino probabilità di perdita molto basse, qui 10^{-4} , mentre all'aumentare della probabilità di perdita è il modello di traffico utilizzato che si comporta peggio, questo conferma quanto già detto a commento delle simulazioni e del rapporto con la teoria nel paragrafo 6.3, il modello teorico si riferisce allo stato di regime mentre in realtà si hanno sorgenti di durata temporale limitata. In particolare si ottengono perdite maggiori di quelle previste dalla teoria. Inoltre i risultati sono peggiori nel caso in cui si utilizzano i valori di L_p e λ_p ricavati dai dati del traffico reale.

Nella tabella 6.9 sono riportati i valori ottenuti dalle simulazioni relativi alle perdite, Π_r , di pacchetti sperimentate nel caso di dimensionamento del buffer con i valori di L della tabella 6.2 previsti dal modello teorico.

Rate(B/s)	L ₂ (B)	Π _r
5000	2131	0.013
8000	1568	0.013
48000	548.2	0.013
100000	343.6	0.013

Tabella 6.9

6.5 Sorgenti On-Off

Come per il caso delle sorgenti FTP anche per le sorgenti On-off sono state effettuate delle simulazioni, in MATLAB, per valutare l'effetto del carico di traffico generato da tali sorgenti sullo spazio di buffer richiesto. Anche in questo caso si sono considerate più sorgenti di tipo On-Off, con valori diversi dei parametri caratteristici delle stesse. In particolare a partire dai dati forniti dalle funzioni in linguaggio C, contenute nell'appendice A.1, che descrivono il modello della sorgente, [8], sono state considerate sorgenti con differente durata temporale, valori diversi per i periodi di talk-spurt, \bar{T}_{on} , e di silenzio \bar{T}_{off} e di conseguenza con differente burstiness, a questi legata dalla relazione 6.5.1:

$$b = \frac{\bar{T}_{on} + \bar{T}_{off}}{\bar{T}_{on}} \quad (6.5.1)$$

I valori dei parametri relativi alle sorgenti considerate sono riportati nella tabella 6.10, in questa i simboli hanno lo stesso significato di quelli riportati nella tabella 6.3 per le sorgenti FTP.

Sorgente	T _{oss} (s)	D(B)	N _p	L _{ps} (B)	b
S1	71	101160	1405	72	2.803
S2	36	48024	667	72	3.108
S3	250	326575	4536	71.9	3.105
S4	120	186408	2589	72	2.613

Tabella 6.10

Le simulazioni per le sorgenti considerate sono state effettuate considerando valori diversi per il rate di trasmissione del canale, il dato costante emerso è che, per tutte le sorgenti al variare del rate, il valore dell'occupazione massima del buffer rimane costante. La tabella 6.11 contiene i dati ottenuti dalle simulazioni relativi al ritardo di coda massimo e medio dei pacchetti IP, anche questi uguali per tutte le sorgenti considerate, al variare del rate di trasmissione del canale.

Rate (B/s)	$D_{\max}(s)$	$D_{\text{mean}}(s)$
5000	0.0860	0.044
8000	0.0540	0.0279
20000	0.0216	0.0110
48000	0.0090	0.0046
60000	0.0072	0.0037
100000	0.0043	0.0022

Tabella 6.11

Tali valori sono stati ottenuti per tutte le sorgenti considerate. La figura 6.6 mostra l'andamento del ritardo massimo al variare del rate di trasmissione.

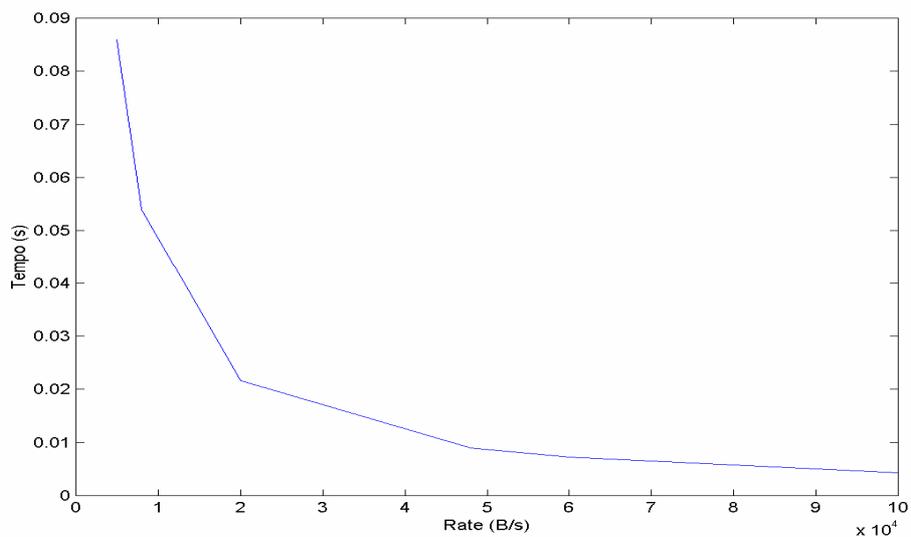


Fig.6.6 Ritardo massimo

Nella figura 6.7 è mostrato l'andamento del ritardo medio dei pacchetti IP delle sorgenti On-Off, al variare del rate di trasmissione, anche questi valori come illustrato nella tabella 6.11 sono uguali per tutte le sorgenti considerate.

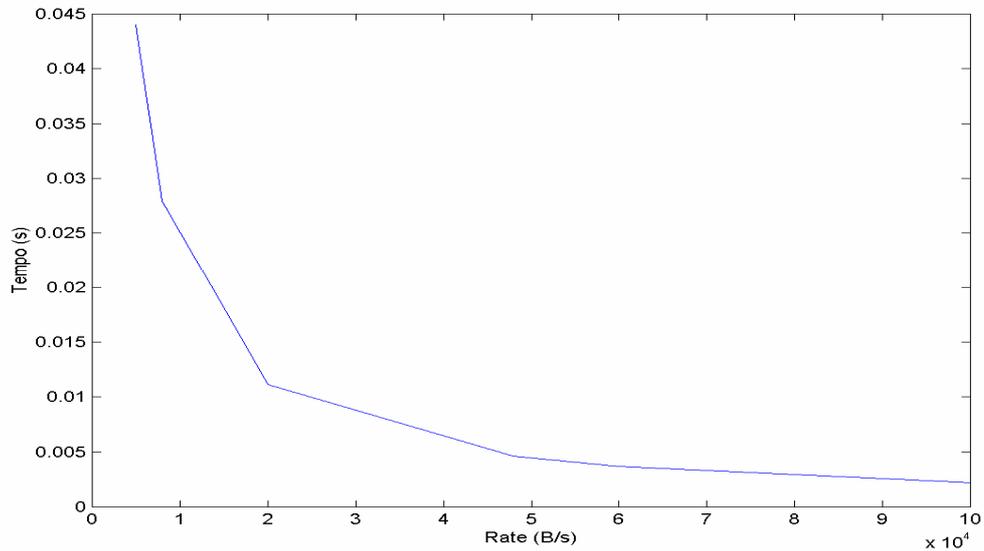


Fig.6.7 Ritardo medio

Il valore massimo ottenuto per lo stato di occupazione del buffer è identico per tutte le sorgenti e pari a:

$B_{max}=432$ byte.

I valori medi per le diverse sorgenti sono riportati nella tabella 6.12, essi non variano con il rate di trasmissione.

Sorgente	B_{mean} (byte)
S1	142.479
S2	133.40
S3	130.63
S4	155.34

Tabella 6.12

Per le sorgenti S1 ed S4 sono stati calcolati i valori dell'occupazione media del buffer forniti dalla teoria. Nella tabella 6.13 si riportano per i valori del rate di 5000 B/s e 60000 B/s le differenze tra il valore medio di occupazione del buffer ottenuto dalle simulazioni e i valori teorici: Δ_B e l'analogo differenza per i ritardi di coda: Δ_D , per la sorgente S1. Il valore della frequenza di interarrivo considerato per il calcolo teorico è quello ottenuto come rapporto tra il numero totale di pacchetti trasmessi e la durata della sessione, è indicato con λ_s .

Per la sorgente S1 è $\lambda_{s1}=19.8$, per la sorgente S4 $\lambda_{s4}=21.67$.

Rate(B/s)	Δ_B	Δ_D
5000	134.56	0.038
60000	142.43	0.043

Tabella 6.13

La tabella 6.14 riporta I dati analoghi alla tabella 6.13 per la sorgente S4.

Rate(B/s)	Δ_B (byte)	Δ_D (s)
5000	145.14	0.037
60000	155.29	0.043

Tabella 6.14

Come è evidente in entrambi I casi e per diversi valori del rate di trasmissione i risultati ottenuti sono molto distanti dal caso teorico sia per il ritardo che per l'occupazione del buffer.

6.6 Sorgenti HTTP

Come per gli altri tipi di sorgenti anche per le sorgenti HTTP sono state considerate nelle simulazioni diverse sessioni con differenti durate temporali e

differenti dimensioni in byte ed è stato effettuato un confronto con i valori forniti dalla teoria. I dati considerati nelle simulazioni sono quelli ottenuti dalle funzioni C, riportate nell'appendice A.2 che descrivono il modello di traffico utilizzato [21].

Nella tabella 6.15 sono riportati i dati relativi alle sorgenti considerate, in questa il valore della frequenza di interarrivo dei pacchetti λ_s è quella calcolata a partire dai dati reali ottenuti dal modello di traffico, ossia come rapporto tra il numero di pacchetti trasmessi e la durata temporale della sessione.

Inoltre L_{ps} è la lunghezza media dei pacchetti ottenuta come il rapporto tra la dimensione in byte della sessione e il numero totale di pacchetti trasmessi.

Sorgente	$T_{oss}(s)$	D(byte)	N_p	$L_{ps}(\text{byte})$	λ_s
S1	800	510964	614	832.19	0.77
S2	180	114262	142	832.97	0.79
S3	305	220668	264	835.86	0.86
S4	1605	1129484	1365	827.46	0.85

Tabella 6.15

Nella tabella 6.16 sono riportati i dati ottenuti dalle simulazioni per la sorgente S1, il significato dei simboli è analogo a quello delle tabelle riportate per le sorgenti FTP. In particolare in tutte le tabelle seguenti il valore teorico di occupazione dei buffer è stato calcolato a partire dalla frequenza di interarrivo dei pacchetti fornita dal modello [21].

Rate(B/s)	$B_{max}(\text{byte})$	$B_{mean}(\text{byte})$	$\Delta_B(\text{byte})$	$D_{max}(s)$	$D_{mean}(s)$	$\Delta_D(s)$
8000	3700	95.39	86.92	0.4625	0.1143	0.1034
10000	3000	95.39	90.08	0.1500	0.0430	0.036
48000	3000	63.87	63.66	0.06250	0.0184	0.0181
100000	3000	63.87	63.82	0.0300	0.0088	0.0087

Tabella 6.16

I dati delle simulazioni per la sorgente S2 sono riportati nella tabella 6.17.

Rate(B/s)	B_{\max} (byte)	B_{mean} (byte)	Δ_B (byte)	D_{\max} (s)	D_{mean} (s)	Δ_D (s)
8000	4500	96.33	87.86	0.5625	0.1094	0.098
10000	4500	66.28	60.97	0.2270	0.0434	0.036
48000	4500	63.48	63.26	0.0938	0.0180	0.017
100000	4500	63.48	63.43	0.045	0.0087	0.0086

Tabella 6.17

Nella tabella 6.18 sono contenuti i dati ottenuti dalle simulazioni per la sorgente S3.

Rate(B/s)	B_{\max} (byte)	B_{mean} (byte)	Δ_B (byte)	D_{\max} (s)	D_{mean} (s)	Δ_D (s)
8000	3000	106.85	98.38	0.3750	0.1147	0.1038
10000	3000	73.39	68.61	0.1500	0.0440	0.0370
48000	3000	72.35	72.13	0.0625	0.0185	0.0182
100000	3000	72.35	72.30	0.0300	0.0089	0.0088

Tabella 6.18

Infine nella tabella 6.19 sono riportati i dati delle simulazioni relativi alla sorgente S4.

Rate(B/s)	B_{\max} (byte)	B_{mean} (byte)	Δ_B (byte)	D_{\max} (s)	D_{mean} (s)	Δ_D (s)
8000	4976	104.63	96.16	0.622	0.1137	0.1028
10000	3076	71.66	66.35	0.1538	0.0436	0.0366
48000	3000	70.37	70.15	0.0625	0.0181	0.0178
100000	3000	70.37	70.32	0.0300	0.0087	0.0086

Tabella 6.19

Anche per le sorgenti HTTP ritorna quanto verificato per le altre sorgenti, la teoria delle code fornisce dei valori inferiori per l'occupazione media dei buffer e

i ritardi medi di coda rispetto a quelli ottenuti dalle simulazioni. Ancora una volta si stanno infatti considerando sessioni di durata temporale limitata e modelli di traffico ben più complessi di quello teorico di sorgente Poissoniana.

Considerando i valori della lunghezza media di pacchetto: L_{ps} e della frequenza di interarrivo dei pacchetti: λ_s sperimentali nelle formule della teoria delle code si ottengono ancora dei risultati lontani da quelli ottenuti nelle simulazioni. Prova ancora questa della scarsa efficacia dei modelli teorici della teoria delle code nel dimensionamento delle risorse di rete necessarie alle sorgenti reali.

Nella figura 6.8 è illustrato l'andamento dell'occupazione media del buffer al variare del rate di trasmissione del canale per le diverse sessioni HTTP considerate.

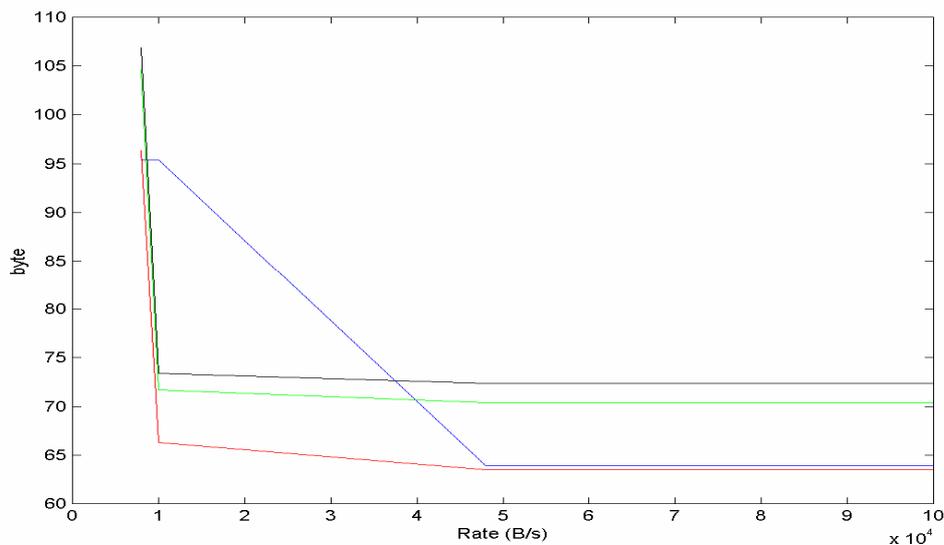


Fig. 6.8 Occupazione media del buffer: in blu S1, in rosso S2, in nero S3, in verde S4.

Nella figura 6.9 è riportato l'andamento del ritardo medio di coda sperimentato dai pacchetti delle diverse sessioni HTTP considerate.

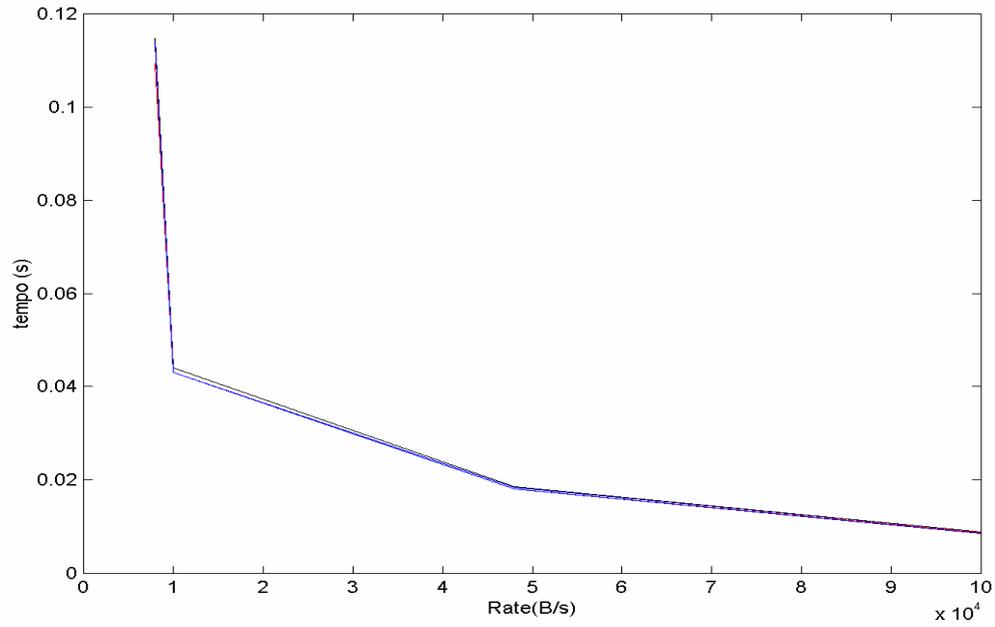


Fig. 6.9 Ritardi medi di coda dei pacchetti IP delle sessioni HTTP : in blu S1, in rosso S2, in nero S3, in verde S4

CAPITOLO 6: DIMENSIONAMENTO DEI BUFFER	105
6.1 <i>Introduzione</i>	105
6.1.1 <i>Teoria delle Code</i>	106
6.1.2 <i>Caratteristiche prestazionali e Traffico in un Sistema di Servizio</i>	109
6.1.3 <i>Code con ingresso e servizio a distribuzione esponenziale</i>	113
6.1.4 <i>Code con tempi servizio a distribuzione generale</i>	115
6.2 <i>Simulazioni: descrizione e obiettivi</i>	116
6.3 <i>Considerazioni sulla validità delle simulazioni</i>	118
6.4 <i>Sorgenti FTP</i>	120
6.5 <i>Sorgenti On-Off</i>	129
6.6 <i>Sorgenti HTTP</i>	132

CAPITOLO 7: CONCLUSIONI

Dalle analisi eseguite nel presente lavoro di tesi emerge che l'utilizzo della teoria delle code nel dimensionamento delle risorse di rete necessarie a soddisfare i requisiti di richieste dalle diverse applicazioni non si rivelano adatte al tipo di scenario considerato nell'ambito del progetto in esame.

Lo strato MAC del progetto in esame deve essere in grado di supportare applicazioni con differenti requisiti di traffico e QoS.

Nell'analisi eseguita allo scopo di ottenere dei risultati verosimili si sono considerati dei modelli di traffico ottenuti da misure reali e a partire da questi si è operata una valutazione della quantità di risorse necessarie a soddisfare le richieste di QoS delle varie applicazioni.

Quindi si è operato un confronto con quanto previsto dalla teoria e come già era intuibile i risultati ottenuti non sono stati in accordo.

I risultati ottenuti non concordano con la teoria per due ragioni fondamentali:

I modelli di traffico delle sorgenti utilizzati sono stati ottenuti da misure reali e non sono di tipo Poissoniano;

La durata delle sessioni è limitata temporalmente, per raggiungere lo stato di regime considerato dalla teoria delle code si dovrebbe attendere un tempo molto lungo.

I motivi di queste scelte come già detto sono motivate la prima dalla volontà di individuare uno scenario di riferimento verosimile, con l'utilizzo di modelli di sorgenti realistici, la seconda, ossia il fatto di considerare le singole sessioni, è dovuta al fatto che a livello dello strato MAC l'allocazione delle risorse avviene a livello di sessione.

L'allocazione delle risorse a partire dai modelli di traffico suddetti non potrà essere realizzata a partire dalla teoria delle code, ma da un' analisi delle caratteristiche reali delle diverse applicazioni.

Disponendo di dati verosimili sarà possibile in futuro procedere ad una valutazione delle prestazioni della rete in esame che possa aderire maggiormente alla realtà rispetto al caso in cui si utilizzino i modelli classici di sorgenti di traffico.

CAPITOLO 7: CONCLUSIONI..... 137

APPENDICE A

Nella presente appendice sono riportati i codici in linguaggio C utilizzati per descrivere i modelli di traffico utilizzati nel lavoro di tesi.

Sono inoltre incluse le funzioni in MATLAB utilizzate per le simulazioni effettuate per valutare l'occupazione dello spazio di buffer da parte delle sorgenti suddette ed effettuare il confronto dei risultati ottenuti con quelli forniti dal modello teorico della teoria delle code.

L'organizzazione per i modelli di traffico è la seguente:

- A.1 : funzione per la generazione del traffico delle sorgenti On-Off:
“Sorgente On-Off.ccp”
- A.2 : funzione per la generazione del traffico delle sorgenti HTTP :
“HTTP .ccp” ;
- A.3 : funzione per la generazione del traffico delle sorgenti FTP:
“Genera_Sessione_FTP.ccp”;
- A.4 : funzione per la generazione del traffico delle sorgenti SMTP:
“SMTP.ccp”;
- A.5 : funzione per la generazione del traffico delle sorgenti video MPEG:
“MPEG.ccp”;
- A.6 : funzione la simulazione del comportamento di una sorgente di traffico Poissoniana : “Poisson.ccp”

Per quanto riguarda le funzioni in MATLAB utilizzate nel capitolo 6 per il dimensionamento del buffer sono riportate le seguenti:

- A.7 : funzione per il calcolo dello spazio di buffer necessario alle diverse sorgenti di traffico individuate : “coda.m”;

- A.8 : funzione per il calcolo del ritardo di coda sperimentato dai pacchetti IP delle diverse applicazioni: “Ritardo.m”.

A.1: Sorgenti On-Off

```

/*SORGENTE ON-OFF.ccp*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#define F 0.0005 /*FREQUENZA DI NASCITA DELLA SORGENTE*/
#define Ton 0.352 /*DURATA MEDIA DEL PERIODO ON IN SECONDI*/
#define Toff 0.625 /*DURATA MEDIA DEL PERIODO OFF IN SECONDI*/
#define Lp 72 /*VALORE IN BYTES DELLA LUNGHEZZA DEL
PACCHETTO IP*/
#define ratePicco 4000 /*VALORE IN BYTES,CORRISPONDE AL RATE DI 32kbit
DELLA CODIFICA ADPCM */
#define Toss 5400 /*VALORE IN SECONDI*/
#define Dmin 0 /*VALORE IN SECONDI*/
#define Dmax 0.9155 /*VALORE IN SECONDI*/
#define PAYLOAD_AL 72

double Nascita(float freq) /*FREQUENZA DI INTERARRIVO DELLE SESSIONI
DISTRIBUZIONE ESPONENZIALE NEGATIVA*/
{
double n,temp;
n=(double)rand()/RAND_MAX;
/*printf("Il valore di n e':%0.3f\n",n);*/
temp=-log(1-n)/freq;
/*printf("Il tempo di internascita e':%0.3f\n",temp);*/
return temp;
};

double genera_durata() /*GENERA LA DURATA DI UNA SESSIONE,
DISTRIBUZIONE GAMMA*/
{
FILE *fp;
float x,F1,r,X,dif,num;
if ((fp=fopen("GammaON_OFF.txt","r"))==NULL)
printf("Operazione non riuscita\n");
r=(double)rand()/RAND_MAX;
X=Dmin+r*(Dmax-Dmin);
dif=X;
while(dif>0.0)
{
fscanf(fp,"%f",&x);
}
}

```

```

        fscanf(fp,"%f",&F1);
        dif=X-F1;
    };
    num=x;
    printf("La durata della chiamata vale:%0.3f\n",num);
    return num;
    fclose(fp);
};

void main()
{
    double Tn,t,ton,toff,tpac=0,Tass=0,durata,d1,Tsim,Dses=0;
    int lpac,k;
    FILE *fp,*fp1,*fp5,*fp3,*fp6;
    for (k=1;k<=1;++k)
    {
        Tass=0;
        Dses=0;
        while (Tass<Toss)
        {
            fp1=fopen("ON_OFF1.txt","a"); /*ISTANTE DI NASCITA E DURATA DI UNA
            CHIAMATA*/
            fp5=fopen("ON_OFF2.txt","a"); /*LUNGHEZZA E ISTANCE DI TRASMISSIONE
            DEI PACCHETTI*/
            fp3=fopen("ON_OFF7.txt","a"); /*DURATE DI Ton E Toff*/
            fp6=fopen("ON_OFF4.txt","a");
            fp=fopen("ON_OFF.txt","a");
            Tn=Nascita(F); /*CALCOLA L'ISTANTE DI INTERNASCITA DELLE
            CHIAMATE*/
            fprintf(fp6,"Il tempo di internascita generato vale:%0.3f\n",Tn);
            if(Tn<Dses)
                Tn=Dses;
            Tass=Tass+(Tn-Dses); /*TEMPO ASSOLUTO*/
            printf("L'istante di nascita vale:%0.3f\n",Tass);
            durata=genera_durata();
            Tsim=Tass;
            Dses=0;
            if (Tsim<Toss)
                fprintf(fp1,"%0.3f %0.3f\n",durata,Tsim);
            while ((durata>0)&&(Tsim<Toss))
            {
                d1=durata;
                ton=Nascita(1/Ton);
                toff=Nascita(1/Toff);
                fprintf(fp3,"%0.3f %0.3f\n",ton,toff);
                t=ton;
                while((t>tpac)&&(d1>0)&&(Tsim<Toss))
                {
                    if (d1>tpac)

```

```

        lpac=Lp;
    else
    {
        lpac=(int)ceil(d1*ratePicco);
        fprintf(fp6,"Tempo insufficiente per generare un pacchetto pari a Lp,la
lung.vale:%d\n",lpac);
    };
    fprintf(fp5,"%d %0.3f\n",lpac,Tsim);
    tpac=(double)lpac/ratePicco; /*TEMPO NECESSARIO PER TRASMETTERE UN
PACCHETTO*/
    Dses+=tpac;
    Tsim+=tpac; /*RAPPRESENTA L'ISTANTE DI TRASMISSIONE DEL
PACCHETTO*/
    d1-=tpac; /*QUANTO TEMPO DI TRASMISSIONE RESTA*/
    if(Tsim<Toss)
    {
        t=t-tpac; /*RAPPRESENTA LA QUANTITA' DI Ton RIMANENTE*/
        fprintf(fp6,"Il tempo assoluto e il tempo ton restante valgono: %0.3f
%0.3f\n",Tsim,t);
        fprintf(fp6,"Il tempo di trasmissione rimanente vale :%0.3f\n",d1);
    };
};
if (t<tpac) /*SI ESCE DAL CICLO WHILE PIU' INTERNO,PERCHE' IL
TEMPO RIMASTO NON E' SUFFICIENTE PER TRASMETTERE UN ALTRO
PACCHETTO*/
{
    fprintf(fp6,"Il valore di ton,rimasto,insufficiente per generare un altro
pacchetto,vale:%0.3f\n",t);
    fprintf(fp5,"%d %0.3f\n",lpac,Tsim);
    Tsim+=t+toff;
    d1=d1-(t+toff);
    Dses=Dses+(t+toff);
};
durata=d1;
if (durata>0)
{
    fprintf(fp6,"Dses e d1 valgono:%0.3f e %0.3f\n",Dses,d1);
    fprintf(fp6,"Il tempo di trasmissione alla fine di ton corrente vale:%0.3f\n",Tsim);
    fprintf(fp6,"Il tempo rimanente alla fine di Ton vale:%0.3f\n",durata);
}
else
    fprintf(fp6,"La trasmissione e' terminata\n");
};/*TERMINE DEL CICLO WHILE RELATIVO ALLA QUANTITA' DI TEMPO DI
TRASMISSIONE*/
Tass=Tsim;
if (Tass<Toss)
{
    fprintf(fp6,"La durata della chiamata vale:%0.3f\n",Dses);
    fprintf(fp6,"Il tempo assoluto al termine della chiamata vale:%0.3f\n",Tass);
}

```

```

}
else
fprintf(fp6,"La simulazione termina perche' e' finito il tempo di osservazione.\n");
fprintf(fp1,"\n","\n");
fprintf(fp3,"\n","\n");
fclose(fp1);
fclose(fp5);
fclose(fp3);
fclose(fp6);
fclose(fp); }; /*FINE DEL CICLO WHILE PIU' ESTERNO*/
}; /*FINE DEL CICLO FOR*/
};

```

A.2: Sorgenti HTTP

```

/*HTTP.ccp*/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define NULL 0
#define ALFA 1.5549 /* ALFA e BETA sono i parametri della funzione di distribuzione
di Pareto,usata per la descrizione della dimensione di una pagina Web*/
#define BETA 31341 /*valore in bytes*/
#define d1 40
#define d2 552
#define d3 576
#define d4 1500
#define x1 0.3017
#define x2 0.4244
#define x3 0.5552
#define Dmin 0.086238
#define Dmax 0.97231
#define Nmax_Array 200 /*Indica la dimensione massima consentita per l'array che
descrive la struttura di una pagina WEB*/
#define Fsess 0.0005 /*Frequenza di interarrivo di una sessione WEB */
#define Durata_Pagina 35 /* Valore in secondi */
#define F 0.934
#define Xmin 0
#define Xmax 0.9992
#define Toss 5400 /*Valore in secondi*/

/*La funzione seguente genera il numero di pagine di una sessione WEB,descritto dalla
distribuzione Lognormale i valori utilizzati sono letti dal file "NumPag.txt */

int genera_numpag()
{
FILE *fp;

```

```

float x,F1,r,X,dif;
int num ;
if ((fp=fopen("NumPag.txt","r"))==NULL)
printf("Operazione non riuscita\n");
r=(double)rand()/RAND_MAX;
X=Dmin+r*(Dmax-Dmin);
    dif=X;
while(dif>0.0)
{
fscanf(fp,"%f",&x);
    fscanf(fp,"%f",&F1);
    dif=X-F1;
};
num=ceil((double)x);
printf("Il numero di pagine e':%d\n",num);
return num;
fclose(fp);
};

```

/*LA FUNZIONE SEGUENTE GENERA I TEMPI TRA DUE PAGINE HTTP, IN PRATICA IL VIEWING TIME,CARATTERIZZATI DA UNA DISTRIBUZIONE GAMMA E I CUI VALORI SONO LETTI DAL FILE GAMMA.txt*/

```

double Tpag()
{
FILE *fp;
float G,y,r1,d,Y;
if((fp=fopen("GAMMA.txt","r"))==NULL)
printf("operazione non riuscita");
r1=(double)rand()/RAND_MAX;
Y=Xmin+r1*(Xmax-Xmin);
printf("Il valore generato e':%0.4f\n",Y);
    d=Y;
    while(d>0.0)
    {
fscanf(fp,"%f",&y);
        fscanf(fp,"%f",&G);
        d=Y-G;
    };
    printf("Il tempo di interarrivo delle pagine vale:%0.3f secondi \n",y);
    return y;
fclose(fp);
};

```

/*FUNZIONE PER LA GENERAZIONE DELLA DIMENSIONE DI UNA PAGINA WEB,CARATTERIZZATA DALLA FUNZIONE DI DISTRIBUZIONE DI PARETO*/

```

int genera_dimensione_pagina()

```

```

{
    int dimpag;
    double num=(double)rand()/RAND_MAX;
    double esponente=1/ALFA;
    dimpag=(int)BETA*(1-pow(1-num,esponente))/(pow(1-num,esponente));    /*formula
ricavata dalla inversione della formula della distribuzione di Pareto*/;
    return dimpag;
    printf("La dimensione della pagina e' %d:\n",dimpag);
};

```

/*Funzione per l'estrazione della dimensione dei pacchetti(caratterizzata da distribuzione multimodale)*/

```

int estrai_dimensione_pacchetto()
{
    double num,N;
    int d;
    num=(double)rand();
    N=num/RAND_MAX;
    if (N > x1)
        if (N > x2)
            if (N > x3)
                d= d4;
            else d=d3;
        else d=d2;
    else d=d1;
    return d;
    printf("La dimensione del pacchetto e' %d\n",d);
};

```

/*Funzione per il calcolo del tempo di trasmissione del prossimo pacchetto,con distribuzione esponenziale negativa*/

```

double estrai_tempo_pacchetto (float f)
{
    double tp,Tpac;
    Tpac=(double)rand()/RAND_MAX;
    /*printf("Il valore generato e' %f\n",Tpac);*/
    tp= -(log(1-Tpac))/f;
    printf("Il tempo di interarrivo del pacchetto e' %0.3f:\n",tp);
    return tp;
};

```

```

void main()
{
    typedef struct pagine_sessione
    {
        double delta_temp;
        int dim;
    }

```



```

Dpag=0;          /*INDICA LA DURATA DELLA SINGOLA PAGINA*/
pag[j].dim=genera_dimensione_pagina();
fprintf(fp2,"%d %0.3f\n",pag[j].dim,Tsim);
lpac=estrai_dimensione_pacchetto();
pag[j].delta_temp=0; /*IL PRIMO PACCHETTO NASCE NELLO STESSO
ISTANTE DELLA PAGINA*/
tpac=estrai_tempo_pacchetto(F);
pag[j].dim-=lpac;
while ((pag[j].dim>0) && (Tsim<Toss))
{
fprintf(fp,"%d %0.3f\n",lpac,Tsim);
/*fprintf(fp1,"%d %d %0.3f %0.3f\n",pag[j].dim,lpac,Tsim,tpac);*/
lpac=estrai_dimensione_pacchetto();
Tsim+=tpac;
pag[j].delta_temp+=tpac; /*QUANTO TEMPO E' PASSATO DALL'INIZIO
DELLA PAGINA*/
tpac=estrai_tempo_pacchetto(F);
pag[j].dim-=lpac;
}; /*FINE DEL CICLO WHILE,LA DIMENSIONE RESTANTE DELLA
PAGINA E'INSUFFICIENTE PER GENERARE UN ALTRO PACCHETTO*/
if(pag[j].dim<0)
{
dim=pag[j].dim+lpac;
fprintf(fp,"%d %0.3f\n",dim,Tsim); /*L'ULTIMO PACCHETTO E' ASSUNTO
DI DIMENSIONE PARI AI BYTES RESTANTI DELLA PAGINA */
/*fprintf(fp1,"%d %d %0.3f %0.3f\n",0,dim,Tsim,tpac);*/
Dpag+=pag[j].delta_temp;
fprintf(fp5,"La durata della pagina vale:%0.3f\n",Dpag);
fprintf(fp5,"la dimensione della pagina e' insufficiente per generare un altro
pacchetto\n");
}
else if(Tsim>Toss)
fprintf(fp5,"La trasmissione e' interrotta perche' il tempo di simulazione e'
terminato\n");
DeltaTpg=Tpag();
Tsim=Tsim+DeltaTpg;
fprintf(fp5,"Il tempo tra le pagine vale :%0.3f\n",DeltaTpg);
};/*TERMINE DEL CICLO FOR INTERNO*/
fprintf(fp5,"Il numero di pagine realmente trasmesse vale:%d\n",j);
Tass=Tsim;
fprintf(fp5,"Il tempo assoluto alla fine della sessione vale:%0.3f\n",Tass);
Dses+=Tass-Tsess;
if(Tass<Toss)
fprintf(fp5,"La sessione e' terminata,la sua durata vale :%0.3f\n",Dses);
else fprintf(fp5,"E' finito il tempo di simulazione\n");
fprintf(fp5,"\n");
}; /*FINE DEL CICLO WHILE ESTERNO*/
fprintf(fp,"\n","\n");
fprintf(fp2,"\n","\n");

```

```

        fprintf(fp5, "\n", "\n");
        fprintf(fp4, "\n", "\n");
        fclose(fp);
    fclose(fp1);
        fclose(fp2);
        fclose(fp5);
        fclose(fp4);
};
};
};

```

A.3 : Sorgenti FTP

```

/* Genera_Sessione_FTP.ccp*/
/*NELLA FUNZIONE SEGUENTE UNA SESSIONE FTP E' MODELLATA COME
UNA SESSIONE WEB CON UN'UNICA PAGINA DESCRITTA IN TERMINI DI
DIMENSIONE DELLA STESSA E DELLA DISTRIBUZIONE DEI PACCHETTI DI
CUI E' COSTITUITA */

```

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define ALFA 1.7584 /*PARAMETRI DELLA FUNZIONE DI DISTRIBUZIONE DI
PARETO PER LA CARATTERIZZAZIONE DELLA DIMENSIONE DI UNA PAGINA
WEB*/
#define BETA 183563 /*VALORE IN BYTES*/
#define d1 40
#define d2 552
#define d3 576
#define d4 1500
#define x1 0.3017
#define x2 0.4244
#define x3 0.5552
/*#define fp 0.1*/
#define F 0.00006
#define durata_pagina 35 /*VALORE IN SECONDI*/
#define PAYLOAD_AL 72
#define Toss 36000
#define Fp 0.93 /*FREQUENZA DI INTERNASCITA DEI PACCHETTI*/

```

```

/*Funzione per generare l'istante di arrivo della sessione ftp*/

```

```

double Nascita(float freq)
{
    double n,temp;
    n=(double)rand()/RAND_MAX;
    printf("Il valore di n e':%f\n",n);
}

```

```

temp=-log(1-n)/freq;
printf("Il tempo di internascita e':%f\n",temp);
return temp;
};

```

```

/*Funzione per generare la dimensione dei pacchetti*/

```

```

int estrai_dimensione_pacchetto()
{
double num,N;
int d;
num=(double)rand();
N=num/RAND_MAX;
if (N>x1)
    if (N>x2)
        if (N>x3)
            d=d4;
        else d=d3;
    else d=d2;
else d=d1;
printf("La dimensione del pacchetto e':%d\n",d);
return d;
};

```

```

};

```

```

void main ()
{
extern int estrai_dimensione_pacchetto();
extern double Nascita(float freq);
int dimftp,lpac,dim,num_macpdu,padding,count=0,j;
double num,Tftp,Tass=0,Tsim=0,tpac,Delta_t,Dses=0;
double esponente;
FILE *fp1, *fp2, *fp0;
for (j=1;j<=4;j++) /*IN QUESTO MODO SI MODELLA IL COMPORTAMENTO DI
QUATTRO UTENTI DIFFERENTI*/
{
Tass=0;
while (Tass<Toss)
{
fp1=(fopen("FTP2.txt","a"));
fp2=fopen("FTP4.txt","a");
fp0=fopen("FTP0.txt","a");
Tftp=Nascita(F); /*TEMPI DI INTERARRIVO DELLE SESSIONI*/
if(Tftp<Dses) /*SI ASSUMONO SESSIONI NON SOVRAPPONENTESI PER IL
SINGOLO UTENTE*/
Tftp=Dses;
Tass=Tass+(Tftp-Dses);
Tsim=Tass;
}
}
}

```

```

Dses=0; /*DURATA DELLA SINGOLA SESSIONE CHE INIZIALIZZO A ZERO AD
OGNI NUOVA NASCITA*/
num=(double)rand()/RAND_MAX;

printf("Il valore generato e': %f\n",num);
esponente=(1/ALFA);
dimftp=(int)BETA*(1-pow(1-num,esponente))/(pow(1-num,esponente));/*GENERA LA
DIMENSIONE DELLA SESSIONE FTP*/
if (Tsim<Toss)
{
fprintf(fp0,"Il tempo di internascita vale:%0.3f\n",Tftp);
fprintf(fp0,"L'istante di nascita della sessione e':%0.3f\n",Tass);
fprintf(fp2,"%d %0.3f\n",dimftp,Tsim);
}
else fprintf(fp0,"La sessione non nasce perche' il tempo di osservazione non e'
sufficiente\n");
Delta_t=0; /*QUANTO TEMPO E' TRASCORSO DALL'INIZIO DELLA
SESSIONE CORRENTE*/
tpac=Nascita(Fp); /*TEMPO DI INTERARRIVO DEI PACCHETTI*/
lpac=estrai_dimensione_pacchetto();
dimftp-=lpac;
while ((dimftp>0) && (Tsim<Toss))
{
num_macpdu=ceil((double)lpac/PAYLOAD_AL);
count=count+num_macpdu;
padding=lpac%PAYLOAD_AL;
/*fprintf(fp2,"%d %d %d %d %d %d %0.3f
%0.3f\n",dimftp,lpac,num_macpdu,count,padding,Tsim,tpac);*/
fprintf(fp1,"%d %0.3f\n",lpac,Tsim);
lpac=estrai_dimensione_pacchetto();
dimftp-=lpac;
printf("La dimensione restante e':%d\n",dimftp);
Delta_t+=tpac; /*QUANTO TEMPO E' PASSATO DALL'INIZIO DELLA
SESSIONE*/
Tsim+=tpac; /*ISTANTE DI TRASMISSIONE DEL PACCHETTO
CORRENTE*/
tpac=Nascita(Fp); /*IL PROSSIMO PACCHETTO SARA' TRASMESSO TRA tpac
SECONDI*/
};
if (dimftp<0) /*FINE CICLO WHILE INTERNO,LA DIMENSIONE DI PAGINA
ESTRATTA NON E' SUFFICIENTE PER GENERARE UN PACCHETTO*/
{
dim=dimftp+lpac; /*ASSUMO L'ULTIMO PACCHETTO DI DIMENSIONE PARI AI
BYTES RIMANENTI DELLA PAGINA!!*/
num_macpdu=ceil((double)dim/PAYLOAD_AL);
padding=dim%PAYLOAD_AL;
count=count+num_macpdu;
/*fprintf(fp2,"%d %d %d %d %d %d %0.3f
%0.3f\n",0,dim,num_macpdu,count,padding,Tsim,tpac);*/

```

```

fprintf(fp1,"%d  %0.3f\n",dim,Tsim);
printf("Valore negativo dimensione insufficiente per generare un'altro pacchetto,la sessione e'
terminata.\n");
Dses+=Delta_t; /*RAPPRESENTA LA DURATA COMPLESSIVA DELLA
SESSIONE*/
fprintf(fp0,"la durata della sessione vale:%0.3f\n",Dses);
}
else if(Tsim>Toss)
fprintf(fp0,"La sessione e'interrotta perche'e' terminato il tempo di simulazione\n");
Tass=Tsim;

if (Tass<Toss)
fprintf(fp0,"Il tempo assoluto vale:%0.3f\n",Tass);
fprintf(fp1,"\n","\n");
fprintf(fp2,"\n","\n");
fprintf(fp0,"\n","\n");
fclose(fp1);
fclose(fp2);
fclose(fp0);
}; /*FINE DEL CICLO WHILE ESTERNO*/
}; /*FINE DEL CICLO FOR*/
};

```

A.4:Sorgenti SMTP

```

/*SMTP.ccp*/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define F 0.001 /*FREQUENZA DI INTERNASCITA DELLE APPLICAZIONI
SMTP*/
#define ALFA 2.009 /*ALFA E BETA SONO I PARAMETRI DELLA DISTRIBUZIONE
DI PARETO PER LA SMTP SIZE*/
#define BETA 12710 /*VALORE IN BYTES*/
#define Lp 72 /*VALORE IN BYTES*/
#define ratePicco 5000 /*BIT AL SECONDO*/
#define Toss 5400 /*VALORE IN SECONDI*/

#define Ti 0.1152 /*TEMPO DI INTERARRIVO DEI PACCHETTI*/

/*GENERA IL TEMPO DI INTERARRIVO DI UN'APPLICAZIONE SMTP E DEI
RELATIVI PACCHETTI:DISTRIBUZIONE ESPONENZIALE NEGATIVA*/

double Nascita(float freq)
{

```

```

double n,temp;
n=(double)rand()/RAND_MAX;
printf("Il valore di n e':%0.4f\n",n);
temp=-log(1-n)/freq;
printf("Il tempo di internascita e':%0.4f\n",temp);
return temp;
};

```

/*FUNZIONE PER LA GENERAZIONE DELLA DIMENSIONE DI UNA SESSIONE SMTP,DESCRITTA DALLA FUNZIONE DI DISTRIBUZIONE DI PARETO*/

```

int genera_dimensione()
{
    int dim;
    double num=(double)rand()/RAND_MAX;
    printf("Il valore di num generato vale:%0.4f\n",num);
    double esponente=1/ALFA;
    dim=(int)BETA*(1-pow(1-num,esponente))/(pow(1-num,esponente)); /*formula ricavata
dalla inversione della formula della distribuzione di Pareto*/
    printf("La dimensione della sessione e'%d:\n",dim);
    return dim;
};

```

```

void main ()
{
    extern int genera_dimensione();
    extern double Nascita(float freq);
    double Ts,tpac,Tass=0,Dses=0,Tsim,Delta_t=0;
    int j,lpac,s,dim;
    FILE *fp, *fp1, *fp2;
    for (j=1;j<=4;j++)
    {
        Tass=0;
        Dses=0;
        while (Tass<Toss)
        {
            fp=fopen("sm.txt","a"); /*DIMENSIONE E ISTANTE DI NASCITA*/
            fp1=fopen("sm1.txt","a"); /*LUNGHEZZA E ISTANTE DI NASCITA DEI
PACCHETTI*/
            fp2=fopen("sm3.txt","a");
            Ts=(double)Nascita(F); /*TEMPO DI INTERARRIVO DELLA SESSIONE*/
            fprintf(fp2,"Il tempo di interarrivo della sessione vale:%0.4f\n",Ts);
            if (Ts<Dses)
                Ts=Dses;
            Tass=Tass+(Ts-Dses);
            Tsim=Tass;
            fprintf(fp2,"L'istante di nascita della sessione vale:%0.4f\n",Tsim);
            if(Tsim<Toss)
            {

```

```

s=(int)genera_dimensione();
printf("La dimensione della sessione e':%d\n",s);
fprintf(fp2,"La dimensione della sessione e':%d\n",s);
fprintf(fp,"%d %0.4f\n",s,Tsim);
if(Lp>s)
    lpac=s;
else lpac=Lp;
fprintf(fp1,"%d %0.4f\n",lpac,Tsim);
}
else
fprintf(fp2,"La sessione non nasce perche' e' finito il tempo di simulazione\n");
Dses=0;
Delta_t=0;
while((s>0)&&(Tsim<Toss))
{
    if(Lp>s)
        lpac=s;
    else lpac=Lp;
    fprintf(fp2,"la lunghezza del pacchetto vale:%d\n",lpac);
    fprintf(fp2,"il tempo dall'inizio della sessione vale:%0.4f\n",Delta_t);
    tpac=(double)Nascita(1/Ti);
    Tsim+=tpac;          /*TEMPO DI TRASMISSIONE DEL PACCHETTO*/
    Delta_t+=tpac;      /*QUANTO TEMPO E' TRASCORSO DALL'INIZIO
DELLA SESSIONE*/
    s-=lpac;
    fprintf(fp2,"il tempo di pacc. e la dims.restante valgono:%0.4f e %d\n",tpac,s);
    if (Tsim<Toss)
        fprintf(fp1,"%d %0.4f\n",lpac,Tsim);
}; /*FINE DEL CICLO WHILE INTERNO*/
if(s<0)          /*SI ESCE DAL CICLO WHILE PERCHE' LA DIMENSIONE DELLA
SESSIONE E' INSUFFICIENTE PER GENERARE UN ALTRO PACCHETTO*/
{
    dim=s+lpac;
    fprintf(fp1,"%d %0.4f\n",dim,Tsim);
    fprintf(fp2,"La sessione e' terminata\n");
};
if(Tsim>Toss)
    fprintf(fp2,"La sessione termina perche' e' terminato il tempo di simulazione\n");
Tass=Tsim;
Dses+=Delta_t;
if(Tass<Toss)
{
    fprintf(fp2,"La durata della sessione vale: %0.4f\n",Dses);
    fprintf(fp2,"Il tempo assoluto alla fine della sessione vale:%0.4f\n",Tass);
};
fprintf(fp,"\n");
fprintf(fp1,"\n");
fprintf(fp2,"\n");
fclose(fp);

```

```

fclose(fp1);
fclose(fp2);
};/*FINE DEL CICLO WHILE ESTERNO*/
};/*FINE DEL CICLO FOR*/
};

```

A.5:Sorgenti MPEG

```

/*MPEG.CCP*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define Xmin 0
#define Xmax 0.9702
#define Dmin 0
#define Dmax 0.9556
#define Nt 25 /*NUMERO DI TRAME AL SECONDO*/
#define Toss 480
#define Lp 512
#define F 0.0002 /*FREQUENZA DI INTERARRIVO DELLE SESSIONI*/
#define rate 125000 /*IN BYTE AL SECONDO*/

/*FUNZIONE PER LA GENERAZIONE DELLA DURATA DELLA SEQUENZA
VIDEO: DISTRIBUZIONE GAMMA*/

double genera_durata()
{
FILE *fp;
float G,y,r1,d,Y;
if((fp=fopen("GAMMAv.txt","r"))==NULL)
printf("operazione non riuscita");
r1=(double)rand()/RAND_MAX;
printf("Il valore di r1 e':%f\n",r1);
Y=Xmin+r1*(Xmax-Xmin);
printf("Il valore generato e':%0.4f\n",Y);
d=Y;
while(d>0.0)
{
fscanf(fp,"%f",&y);
fscanf(fp,"%f",&G);
d=Y-G;
};
printf("La durata della sequenza vale:%0.3f secondi \n",y);
return y;
fclose(fp);
};

```

```
/*GENERA LA FREQUENZA DI INTERARRIVO DELLE SEQUENZE VIDEO:
DISTRIBUZIONE ESPONENZIALE NEGATIVA*/
```

```
double Nascita(float freq)
{
double n,temp;
n=(double)rand()/RAND_MAX;
/*printf("Il valore di n e':%0.3f\n",n);*/
temp=-log(1-n)/freq;
printf("Il tempo di internascita e':%0.3f\n",temp);
return temp;
};
```

```
void main()
{
FILE *fp,*fp1,*fp2;
float x,F1,r,X,dif,d,Tass=0,Tn,Dses=0,Tsim,tpac;
double t=0;
int num,i,n,lp,c;
while(Tass<Toss)
{
Tn=Nascita(F);
if(Tn<Dses)
Tn=Dses;
Tass=Tass+(Tn-Dses); /*TEMPO ASSOLUTO*/
d=genera_durata(); /*GENERA LA DURATA TEMPORALE DELLA
SESSIONE*/
Tsim=Tass;
n=d*Nt;
printf("Il numero di trame vale:%d\n",n);
printf("Il tempo assoluto vale:%0.3f\n",Tass);
for (j=1;j<=n;++j)
{
if(Tsim<Toss)
{
fp2=fopen("VIDEO1.txt","a");
fp1=fopen("TRAME.txt","a");
if ((fp=fopen("Frameh.txt","r"))==NULL)
printf("Operazione non riuscita\n");
r=(double)rand()/RAND_MAX;
X=Dmin+r*(Dmax-Dmin);
dif=X;
while(dif>0.0)
{
fscanf(fp,"%f",&x);
fscanf(fp,"%f",&F1);
dif=X-F1;
};
num=ceil((double)x); /*INDICA LA DIMENSIONE DELLA TRAMA*/
```

```

        if(Tsim<Toss)
            fprintf(fp1,"%d  %0.3f\n",num,Tsim);          /*DIMENSIONE E ISTANTE DI
NASCITA DELLA TRAMA*/
    };
        Tsim=Tsim+(double)1/Nt;          /*ISTANTE DI NASCITA DELLA PROSSIMA
TRAMA*/
        fclose(fp);
        fclose(fp1);
        fclose(fp2);
    };
    Tass=Tsim;
    Dses=d;
    if(Tass>Toss)
        printf("La simulazione termina perchè e' terminato il tempo di simulazione\n");
        printf("Il tempo assoluto vale:%0.3f\n",Tass);
};
};

```

A.6 : Sorgente Poissoniana

```

/*Poisson.ccp*/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define Lp 822.34
#define Fp 0.93
#define F 0.00006
#define Toss 9000
/*GENERA LA FREQUENZA DI INTERARRIVO DEI PACCHETTI IP*/
double Nascita(float freq)
{
    double n,temp;
    n=(double)rand()/RAND_MAX;
    printf("Il valore di n e':%f\n",n);
    temp=-(-log(1-n))/freq;
    printf("Il tempo di internascita e':%f\n",temp);
    return temp;
};

/*GENERA LA LUNGHEZZA DEI PACCHETTI:DISTRIBUZIONE ESPONENZIALE
NEGATIVA*/
int Lunghezza (int l)
{
    double m;
    int dim;
    m=(double) rand()/RAND_MAX;
    dim=(int)(-(-log(1-m))*l);
    printf("La lunghezza del pacchetto vale:%d\n",dim);
}

```

```

return dim;
};

void main ()
{
extern int Lunghezza (int l);
extern double Nascita(float freq);
int dimftp,lpac,dim,j;
double num,Tftp,Tass=0,Tsim=0,tpac,Delta_t,Dses=0;
double esponente;
FILE *fp1, *fp2, *fp0;
for (j=1;j<=1;j++) /*IN QUESTO MODO SI MODELLA IL COMPORTAMENTO DI
QUATTRO UTENTI DIFFERENTI*/
{
Tass=0;
while (Tass<Toss)
{
fp1=(fopen("P2.txt","a"));
fp2=fopen("P4.txt","a");
fp0=fopen("P0.txt","a");
Tftp=Nascita(F); /*TEMPI DI INTERARRIVO DELLE SESSIONI*/
if(Tftp<Dses) /*SI ASSUMONO SESSIONI NON SOVRAPPONENTESI PER IL
SINGOLO UTENTE*/
Tftp=Dses;
Tass=Tass+(Tftp-Dses);
Tsim=Tass;
Dses=0; /*DURATA DELLA SINGOLA SESSIONE CHE INIZIALIZZO A ZERO AD
OGNI NUOVA NASCITA*/
if (Tsim<Toss)
{
fprintf(fp0,"Il tempo di internascita vale:%0.3f\n",Tftp);
fprintf(fp0,"L'istante di nascita della sessione e':%0.3f\n",Tass);
}
else fprintf(fp0,"La sessione non nasce perche' il tempo di osservazione non e'
sufficiente\n");
Delta_t=0; /*QUANTO TEMPO E' TRASCORSO DALL'INIZIO DELLA
SESSIONE CORRENTE*/
/*tpac=Nascita(Fp); /*TEMPO DI INTERARRIVO DEI PACCHETTI*/
/*lpac=Lunghezza(Lp);*/
while ((Tsim<Toss))
{
lpac=Lunghezza(Lp);
fprintf(fp1,"%d %0.3f\n",lpac,Tsim);
tpac=Nascita(Fp); /*IL PROSSIMO PACCHETTO SARA' TRASMESSO TRA tpac
SECONDI*/
Delta_t+=tpac; /*QUANTO TEMPO E' PASSATO DALL'INIZIO DELLA
SESSIONE*/

```

```

Tsim+=tpac;                               /*ISTANTE DI TRASMISSIONE DEL PACCHETTO
CORRENTE*/
fprintf(fp2,"%0.3f\n",tpac);
};
fprintf(fp0,"La sessione e'interrotta perche'e' terminato il tempo di simulazione\n");
Dses+=Delta_t;
fprintf(fp0,"La durata della sessione vale:%0.3f\n",Dses);
Tass=Tsim;
if (Tass<Toss)
fprintf(fp0,"Il tempo assoluto vale:%0.3f\n",Tass);
fprintf(fp1,"\n","\n");
fprintf(fp2,"\n","\n");
fprintf(fp0,"\n","\n");
fclose(fp1);
fclose(fp2);
fclose(fp0);
}; /*FINE DEL CICLO WHILE ESTERNO*/
}; /*FINE DEL CICLO FOR*/
};

```

A.7 : Occupazione Buffer

```

function y=coda(Rate,delta,T_oss,arrivi,dimensioni)
% funzione che analizza lo stato di un buffer
% per un sistema a coda con unico servente
% con rate di uscita pari ad Rate
tempo=0;
osservazioni=floor(T_oss/delta);
buffer=zeros(1,osservazioni);
c=floor((Rate*delta));
for a=2:osservazioni
    tempo_succ=tempo+delta;
    stato_attuale=buffer(a-1);
    dopo_estrazione=stato_attuale-c;
    if dopo_estrazione<0
        dopo_estrazione=0;
    end
    buffer(a)=dopo_estrazione;
    % incremento del buffer
    for j=1:length(arrivi)
        if ((arrivi(j)<tempo_succ)&(arrivi(j)>=tempo))
            buffer(a)=buffer(a)+dimensioni(j);
        end
    end
    tempo=tempo_succ;
end
y=buffer;

```

A.8 : Ritardo di coda

```
function y=Ritardo(Rate,delta,T_oss,arrivi,dimensioni)
% funzione che analizza lo stato di un buffer
% per un sistema a coda con unico servernte
% con rate di uscita pari ad Rate
tempo=0;
osservazioni=floor(T_oss/delta);
buffer=zeros(1,osservazioni);
tempo_tx=zeros(1,length(arrivi));
ritardo=zeros(1,length(arrivi));
c=floor((Rate*delta));
for a=2:osservazioni
    tempo_succ=tempo+delta;
    stato_attuale=buffer(a-1);
    dopo_estrazione=stato_attuale-c;
    if dopo_estrazione<0
        dopo_estrazione=0;
    end
    buffer(a)=dopo_estrazione;
    % incremento del buffer
    for j=1:length(arrivi)
        if ((arrivi(j)<tempo_succ)&(arrivi(j)>=tempo))
            ritardo(j)=(buffer(a)+dimensioni(j))/Rate;
            buffer(a)=buffer(a)+dimensioni(j);
        end
    end
    tempo=tempo_succ;
end
y=ritardo;
```

APPENDICE A	139
A.1: Sorgenti On-Off	140
A.2: Sorgenti HTTP	143
A.3 : Sorgenti FTP	148
A.4:Sorgenti SMTP	151
A.5:Sorgenti MPEG	154
A.6 : Sorgente Poissoniana	156
A.7 : Occupazione Buffer	158
A.8 : Ritardo di coda	159

BIBLIOGRAFIA

- [1] Pahlavan K., Probert Thomas H., Chase Mitchell E., "Trends in Local Wireless Networks", IEEE Communication Magazine 1995.
- [2] ISO/IEC 8802-11; ANSI/IEEE Std 802.11, 1999 edn , 20 Aug. 1999 Information technology - Telecommunications and Information exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [3] Martin Johnsson: HiperLAN/2-The Broadband Radio Transmission Technology Operating in 5 GHz Frequency Band. HiperLAN/2 Global Forum, 1999.
- [4] European Telecommunications Standard Institute (ETSI) 300 652, Ottobre 1996.
- [5] Edfors O., Sandell M., Landstrom D., Sjoberg F., "An introduction to Orthogonal frequency-division multiplexing", Technical Report September 1996, Lulea University of Technology.
- [6] Matiae Dusan, "OFDM as a possible modulation technique for multimedia applications in the range of mm waves", Introduction to OFDM, II edition 10/30/98/TUD-TVS.
- [7] Van Nee R., Prasad R., "OFDM for Wireless Multimedia Communication", Artech House.
- [8] Roveri A. "Retematica : funzioni e servizi di rete", Dispense del Corso di "Reti di Telecomunicazioni", Università di Roma "La Sapienza".
- [9] Zhao W., Olshefski D., Schulzrinne H., "Internet Quality of Service: an Overview", Technical Report CUCS-003-00, Columbia University.
- 10] Schmitt J., Wolf L., "Quality of Service – An Overview", Technical Report TR-KOM-1997-01, Darmstad University of Technology.

- [11] Steinmetz R., Wolf L.C., “Quality of Service: Where are WE?”, Invited Paper and Talk at IFIP Fifth International Workshop on Quality of Service (IWQOS '97)- Building QoS into Distributed Systems-May 21-23, 1997,Columbia University, New York, USA.
- [12] Xiao X., Ni Lionel M., “Internet QoS: A Big Picture”, IEEE Network, 13(2): 8-18, March/April 1999.
- [13] Guerin R., Peris V., “Quality-of- Service in Packet Networks Basic Mechanism and Directions”, Invited Paper, Computer Networks, Vol. 31, No 3 February 1999, pp 169-179.
- [14] Braden R., Clark D., Shenker S., “ Integrated Service in the Internet Architecture”: an Overview”. Request For Comments (Proposed Standard) RFC 1633, Internet Engineering Task Force, June 1994.
- [15] Shenker S., Partridge C., Guerin R., “Specification of Guaranteed Quality of Service”. Request For Comments (Proposed Standard) RFC 2212, Internet Engineering Task Force, September 1997.
- [16] Wroclawski J., “Specification of the Controlled Load Element Service”. Request For Comments (Proposed Standard) RFC 2211, Internet Engineering Task Force, September 1997.
- [17] Shenker S., Wroclawski J, “General Characterization Parameters for Integrated Service Network Elements”. Request For Comments (Proposed Standard) RFC 2215, Internet Task Force, September 1997.
- [18] Braden R., Zhang L., Berson S., Herzog S., Jamin S., “Resource reReservation Protocol (RSVP) version 1, functional specification”. Request For Comments (Proposed Standard) RFC 2205, Internet Engineering Task Force, September 1997.
- [19] Roveri A., ”Reti di Telecomunicazioni , Principi Generali ”, SSGRR, L'Aquila, 1995.

- [20] Mah B. A., "An Empirical Model of HTTP Network Traffic", in Proceedings of INFOCOM '97, April 7-11,1997, in Kobe, Japan.
- [21] Staehle D., Leibnitz K., Tran-Gia P., "Source Traffic Modeling of Wireless Application", Technical Report No. 261, 2000, University of Wurzburg, Germany.
- [22] Reyes-Lecuona A., Gonzales-Parada E., Casilari E., Casasola J. C., Diaz-Estrella A., "A Page-Oriented WWW Traffic Model for Wireless System Simulations", Proceedings of the 16th International Teletraffic Congress(ITC'16), Edinburgh, United Kingdom, June, 1999,pp.1271-1280.
- [23] Casilari E., Reyes-Lecuona A., Gonzales F. J., Diaz-Estrella A.,Sandoval F. "Characterization of Web Traffic", Proceedings of IEEE GLOBCOM 2001, S. Antonio (Texas, USA), November 2001.
- [24] Tang D., Baker M., "Analysis of a Local-Area Wireless Network", in Proceedings of MOBICOM 2000, August 2000, pp1-10, ACM press.
- [25] Blefari-Melazzi N.,Listanti M., Roveri A.,"RETEMATICA: Infrastrutture per Applicazioni Telematiche", dispense del corso di " Reti di Telecomunicazioni", Università di Roma "La Sapienza".
- [26] Rose O., " Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems", in Proc.20th Annu. Conf. Local Computer Networks, Minneapolis, Oct.15-18, 1995.
- [27] Fitzek Frank H.P., Reisslein M., "MPEG--4 and H.263 Video Traces for Network Performance Evaluation", Technical Report TKN-00-06, October 2000, Technical University Berlin.
- [28] Decina M., Roveri A., "Code e Traffico nelle Reti di Comunicazione", EUROMA, Editrice Universitaria di Roma- La Goliardica.
- [29] Roveri A., "RETEMATICA : Modelli prestazionali e strumenti progettuali", Dispense del Corso di " Reti di Telecomunicazioni ", Università di Roma, "La Sapienza".

[30] A. De Luca “Gestione del traffico e allocazione delle risorse nella realizzazione di un protocollo MAC a QoS garantita per una W-LAN con accesso OFDM-CDMA”, Tesi di laurea, Università di Roma “La Sapienza”, 2001.

[31] A. Vitali “Trasmissioni OFDM-CDMA e OFDM-TDMA con applicazioni alle Wireless LANs”, Tesi di laurea, Università di Roma “La Sapienza”, 2001.

BIBLIOGRAFIA.....160

Ringraziamenti

Un sincero ringraziamento ai miei genitori e a mio fratello che mi hanno per primi sempre incoraggiato e sostenuto soprattutto nei momenti più difficili.

Grazie anche a tutti gli amici che hanno condiviso con me le tappe di questo percorso.

Voglio inoltre ringraziare la Professoressa Maria Gabriella Di Benedetto che mi ha dato la possibilità di svolgere questo lavoro di tesi.

Un grazie particolare all'Ing. Guerino Giancola per il suo sostegno morale e materiale durante lo svolgimento del lavoro di tesi.