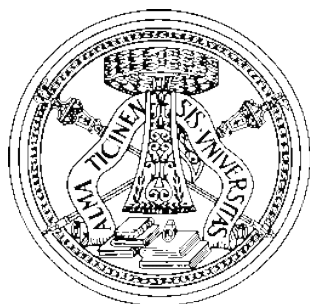




Università degli Studi di Pavia
Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

**NUOVI APPROCCI NELLA STIMA
DELLA BANDA DISPONIBILE IN
UNA RETE A PACCHETTO**

Relatore:

Prof. Giuseppe F. Rossi

Correlatore:

Prof. Paolo E. Gamba

Tesi di laurea di:

Emanuele Goldoni

Matricola: 3335618/61

Anno Accademico 2006/07

Indice

Indice	i
Elenco delle figure	iv
Elenco delle tabelle	vi
Elenco degli Acronimi	vii
1 Introduzione	1
2 Le reti a pacchetto	7
2.1 Le origini	7
2.2 Il funzionamento	8
2.3 I vantaggi	9
2.3.1 Maggior efficienza	9
2.3.2 Minori latenze	9
2.3.3 Maggior robustezza	10
2.4 Internet, la rete delle reti	11
2.5 Ulteriori approfondimenti	11
3 I dispositivi di interconnessione	12
3.1 Il modello a code	12
3.1.1 La gestione delle code	13
3.2 Gli effetti di accodamento	14
3.2.1 Coppie di pacchetti	14
3.2.2 Treni di pacchetti	15
3.2.3 Cross-traffico	15
3.3 Ulteriori approfondimenti	15
4 Le misurazioni in rete	17
4.1 Modalità di misurazione	17
4.1.1 Misurazioni passive	17
4.1.2 Misurazioni attive	18
4.1.3 Misurazioni miste	19
4.2 Le metriche	19
4.2.1 Il prodotto banda-ritardo	19
4.2.2 La capacità	20
4.2.3 La banda disponibile	21

4.2.4	Bulk Transfer Capacity	22
4.3	Ulteriori approfondimenti	22
5	La stima della banda disponibile	23
5.1	L'importanza della banda disponibile	24
5.2	Lo stato dell'arte	24
5.2.1	I modelli affini	25
5.2.2	I modelli di gap probing (PGM)	26
5.2.3	I modelli di rate probing (PRM)	28
5.3	I problemi nella stima della banda disponibile	29
5.3.1	I dispositivi invisibili	30
5.3.2	La gestione delle code	30
5.3.3	La gestione del tempo	31
5.3.4	La descrizione del traffico	31
5.3.5	Le tempistiche di misurazione	32
5.3.6	La verifica delle misurazioni	33
6	L'ambiente di prova	34
6.1	La rete	34
6.2	Pathchirp	35
6.2.1	La notazione	36
6.2.2	L'algoritmo di stima	37
6.2.3	Il confronto con gli altri metodi esistenti	38
6.2.4	Configurazione utilizzata per i test	40
6.3	Iperf	41
7	Le prove in laboratorio	42
7.1	L'invasività di pathChirp	43
7.1.1	Il modello teorico	43
7.1.2	La verifica del modello	45
7.2	Il problema della sovrastima	45
7.3	I limiti nella risoluzione	47
7.3.1	L'entità dell'errore	47
7.3.2	Accorgimenti per aumentare la risoluzione	48
7.4	Available-bandwidth Smart Smoothing On-Line	49
7.4.1	I requisiti del sistema di filtraggio	50
7.4.2	L'algoritmo di ASSOLO	50
7.4.3	Il confronto con gli altri algoritmi	56
7.4.4	Quantizzazione	64
7.5	Il tasso di servizio variabile	64
8	Conclusioni	68
A	La suite TCP/IP	70
A.1	IPv4	72
A.2	TCP/UDP per IPv4	73
A.2.1	User Datagram Protocol	74
A.2.2	Transmission Control Protocol	75

<i>INDICE</i>	v
B La rete GARR	77
C Codice sorgente	78
C.1 pseudocodice.c	78
C.2 compute inst bw excursion.c	79
Bibliografia	82

Elenco delle figure

1.1	Misurazione attiva in rete con traffico di prova	2
1.2	Confronto tra risultati ottenuti con pathChirp e con ASSOLO	3
2.1	Mappa dei nodi connessi ad ARPANET (1974)	8
2.2	Più flussi di pacchetti possono condividere uno stesso canale (da [3])	9
2.3	La divisione in pacchetti riduce il ritardo complessivo	10
3.1	Modello a code utilizzato per descrivere un router (Fonte [8])	13
3.2	Modello semplificato di rete di code (da [12])	14
3.3	Effetti di accodamento su coppie di pacchetti (da [12])	15
3.4	Effetti di spaziatura imputabili alla presenza di cross-traffico (da [12])	16
4.1	Rete con misurazioni passive lungo l'intero percorso (da [25])	18
4.2	Misurazioni attive in rete con traffico di prova (da [8])	18
4.3	Occupazione effettiva della banda nel tempo (da [32])	21
5.1	Esempio di canale a) completamente utilizzato e b) con ancora una porzione di banda disponibile (da [8])	23
5.2	Il modello di traffico in rete fluido (da [31])	24
5.3	Il principio alla base dei modelli Probe Gap Model (PGM) (da [12])	27
5.4	Il principio alla base dei modelli Probe Rate Model (PRM) (da [31])	28
5.5	pathChirp individua la banda disponibile A inviando pacchetti sempre più ravvicinati	30
5.6	Strategie del traffico medio in transito rilevate su diverse scale tem- porali (Fonte UniPV - Rete GARR)	32
6.1	Schema della rete di prova utilizzata in laboratorio	35
6.2	Tipico andamento dei ritardi in un chirp (da [58])	37
6.3	Confronto tra i più noti strumenti di stima	39
7.1	Pacchetti in un chirp spazati esponenzialmente	43
7.2	Profilo reale di un chirp	46
7.3	Velocità istantanee di pacchetti successivi (chirp con $N = 6$, $P=1$ e $B_{min}, B_{max} = \pm 40\%A$)	48
7.4	Risultato di una breve misurazione effettuata da pathChirp	49
7.5	Analogia tra l'andamento degli indici finanziari e il traffico di rete .	52
7.6	Confronto tra VHF ed il filtro VIDYA-StdDev	54
7.7	Confronto tra VHF ed il filtro VIDYA-CMO	55

7.8	Confronto tra ASSOLO ed il filtro LpEMA	58
7.9	Confronto tra ASSOLO e Stability Filter	60
7.10	Confronto tra ASSOLO e pathChirp (<i>MovingAverage(11)</i>)	61
7.11	Confronto tra ASSOLO e pathChirp (<i>MovingAverage(4)</i>)	62
7.12	Confronto tra ASSOLO e un filtro mediano	63
7.13	Quantizzazione con isteresi effettuata da ASSOLO	65
7.14	Differenza tra capacità nominale ed effettiva nella rete considerata	66
7.15	Jitter medio calcolato da Iperf in funzione del traffico presente in rete	67
A.1	Stack architetturale ISO-OSI	71
A.2	Formato del <i>datagram</i> Internet Protocol (IP)	72
A.3	Formato del <i>datagram</i> User Datagram Protocol (UDP)	74
A.4	Formato del <i>segment</i> Transmission Control Protocol (TCP)	76
B.1	Mappatura della rete Gruppo per l'Armonizzazione delle Reti della Ricerca (GARR)	77

Elenco delle tabelle

5.1	Software di stima della banda attualmente disponibili	25
6.1	Parametri di efficienza dei diversi strumenti di stima della banda (dati ricavati con $Pktsize = 500 \text{ byte}$, fonte [69])	39
6.2	Parametri di configurazione di pathChirp 2.4.1 utilizzati in laboratorio.	40
7.1	Dimensioni dei <i>chirp</i> ottenute con diverse configurazioni	45
7.2	Rate associati ai vari pacchetti nell'implementazione software	47
7.3	Parametri di configurazione utilizzati per le prove	56
7.4	Costo computazionale dei diversi algoritmi di stima	56
7.5	Risultati ottenuti con diverse configurazioni	66

Elenco degli Acronimi

ACK ACKnowledgment

ADR Asymptotic Dispersion Rate

ADSL Asymmetric Digital Subscriber Line

AQM Active Queue Management

ARP Address Resolution Protocol

ARPANET Advance Research Projects Agency NETwork

BART Bandwidth Available in Real-Time

BTC Bulk Trasfer Capacity

CAIDA Cooperative Association for Internet Data Analysis

CBR Constant Bit Rate

CMO Chande Momentum Oscillator

CPU Central Processing Unit

D-ITG Distributed Internet Traffic Generator

DLC Data Link Control

EWMA Exponentially Weighted Moving Average

FCFS First Come First Served

FQ Fair Queuing

GARR Gruppo per l'Armonizzazione delle Reti della Ricerca

GGF Global Grid Forum

GPS Global Positioning System

ICMP Internet Control Message Protocol

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IGI	Initial Gap Increasing
IP	Internet Protocol
IPTV	Internet Protocol Television
IPv4	Internet Protocol versione 4
IPv6	Internet Protocol versione 6
ISO	International Organization for Standardization
LDR	Long Range Dependence
MP3	MPEG-1 Layer 3
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
MWM	Multifractal Wavelet Model
NIC	Network Interface Card
NTP	Network Time Protocol
OSI	Open Systems Interconnection
OWD	One Way Delay
P2P	peer-to-peer
PDU	Protocol Data Unit
PGM	Probe Gap Model
PRM	Probe Rate Model
PTR	Packet Transmission Rate
QoS	Quality of Service
RAND	Research ANd Development
RED	Random Early Detection
RFC	Request For Comment
RTT	Round Trip Time
SLA	Service Level Agreement
SLoPS	Self-Loading of Periodic Streams
SNMP	Simple Network Management Protocol
Spruce	Spread PaiR Unused Capacity Estimate

TCP Transmission Control Protocol

TLC Telecomunicazioni

TOPP Train of Packet Pairs

UDP User Datagram Protocol

VHF Vertical Horizontal Filter

VIDYA Variable Inde DYnamyc Average

VoIP Voice over IP

VPS Variable Packet Size

WRR Weighted Round Robin

Capitolo 1

Introduzione

Le reti di calcolatori comunemente utilizzate sono basate sulla commutazione a pacchetto (*packet switching*), un paradigma di comunicazione introdotto negli anni '60 [1] e grazie al quale oggi Internet riesce a collegare ogni parte del globo. L'approccio adottato nelle reti a pacchetto risulta infatti più economico ed efficiente rispetto alla commutazione di circuito, tradizionalmente utilizzata nelle reti telefoniche, permettendo così la realizzazione di sistemi di comunicazione su larga scala e a costo contenuto.

Effettuare misurazioni nelle reti di telecomunicazione è da sempre una delle principali necessità per chi opera nel settore: le prime pubblicazioni riguardanti le prestazioni di ARPANET, la rete sperimentale da cui sarebbe poi nato Internet, risalgono infatti ai primi anni '70 [2, 3]. Negli anni successivi gruppi di lavoro dell'Internet Engineering Task Force (IETF), del Global Grid Forum (GGF) e ricercatori di diverse università e centri di ricerca hanno quindi proposto numerose metodologie e metriche di misurazione [4, 5, 6, 7]; a questo proposito occorre sottolineare come però non esista ancora oggi uno standard univoco che definisca la terminologia e gli strumenti da adottare per la descrizione delle performance delle reti a pacchetto.

Una delle metriche più utilizzate per caratterizzare una rete di telecomunicazioni è la larghezza di banda o, più brevemente, banda. Nel contesto delle reti digitali, il termine si riferisce alla quantità di dati che il link è in grado di trasmettere nell'unità di tempo; dal momento che l'unità base è il bit, la banda si misura solitamente in bit al secondo (*bps*). Lo stesso termine banda può, in realtà, indicare diverse proprietà: in particolare, si definisce capacità la quantità massima di dati trasferibili dalla rete nell'unità di tempo e banda disponibile la quantità di capacità inutilizzata del link in un certo periodo di tempo. Talvolta si fa riferimento anche alla Bulk Transfer Capacity, una terza misura della capacità del canale di comunicazione legata al protocollo TCP, comunemente utilizzato in Internet. Insieme a quelle sopra presentate, esistono numerose altre metriche interessanti per determinare lo stato di una rete, come la latenza in una direzione, il Round Trip Time, particolarmente utile nel caso di cammini asimmetrici e utilizzato anche per determinare le tempistiche di trasmissione in alcuni protocolli, o il tasso di perdita dei pacchetti; in questa trattazione ci si concentrerà però sulla stima della banda disponibile, problematica che ancora oggi, dopo diversi anni, continua a rappresentare una sfida per i ricercatori e che risulta di grande interesse in molte realtà [8, 9].

La conoscenza della disponibilità di banda ha infatti un impatto diretto sulle prestazioni di molte applicazioni oggi comunemente impiegate: si pensi ad esempio ai sistemi interattivi di controllo remoto, alle reti peer-to-peer o alla trasmissione di contenuti multimediali on-demand. Misurazioni analoghe sono utilizzate anche dai diversi operatori, che fatturano spesso ai clienti in base alla quantità di banda disponibile, definita in un Service Level Agreement. Il VoIP è un'ulteriore tecnologia in grado di trarre grandi benefici dalla stima della banda [10]: grazie alla possibilità di variare il livello di compressione della voce è infatti possibile adattare il traffico inviato in funzione della available bandwidth, mantenendo così un buon livello di fluidità del parlato. Nonostante i molti campi di applicazione possibili, i lavori focalizzati sulla stima della banda disponibile prodotti sono decisamente pochi, se confrontati con la mole di ricerche che hanno interessato altre metriche di caratterizzazione delle prestazioni di rete.

Per il lavoro di ricerca in laboratorio si è preferito evitare l'utilizzo di simulatori di rete ritenendo che questi strumenti, per quanto utili in moltissime situazioni, non siano in grado di rappresentare con sufficiente realismo tutte le componenti rumorose ed i comportamenti imprevedibili del traffico e degli elementi di interconnessione. Al contrario, per poter individuare e sperimentare approcci innovativi nella stima della banda disponibile è stato realizzato in Laboratorio Reti un testbed con calcolatori e dispositivi effettivamente presenti anche in reti reali. Come punto di partenza per le ricerche si è fatto riferimento al lavoro svolto in passato dagli autori del software pathChirp [11]: sviluppato congiuntamente nel 2003 dai laboratori della Rice University e Stanford, ad oggi pathChirp rimane uno dei migliori strumenti sviluppati nell'ambito di progetti di ricerca e liberamente disponibile.

Il funzionamento di questo software è comune a molti altri strumenti di misurazione esistenti e non richiede alcuna conoscenza a priori dello stato o delle caratteristiche della rete di interconnessione, considerata invece alla stregua di una black box. Il sistema attivo di misura è costituito da un processo in esecuzione sul nodo sorgente ed uno passivo, in ascolto sul destinatario; la rilevazione viene effettuata *end-to-end*, inviando del traffico apposito (*probe flow*) dal mittente verso ricevente. Supponendo nota la distribuzione iniziale dei pacchetti spediti è possibile, analizzando come questa viene modificata dall'attraversamento della rete, effettuare delle valutazioni sullo stato della rete.

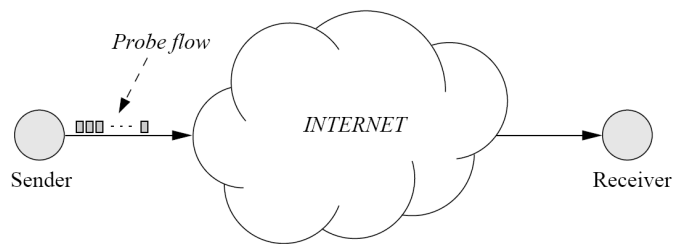


Figura 1.1: Misurazione attiva in rete con traffico di prova

L'algoritmo utilizzato da pathChirp è basato sul principio di autocongestione: per un breve periodo di tempo viene immesso in rete un traffico ad una velocità maggiore di quanto supportato dai nodi intermedi, causando una temporanea congestione

del cammino. Il profilo del traffico inviato è la vera peculiarità del programma sviluppato alla Rice University: anziché inviare stream periodici di pacchetti a rate costante o linearmente crescente, vengono utilizzati treni (chiamati *chirp*, da cui il nome) costituiti da N pacchetti con distanze esponenzialmente decrescenti. La banda disponibile A può quindi essere calcolata dal destinatario individuando il valore esatto in coincidenza del quale il traffico di misurazione inizia a sperimentare un aumento dei ritardi. Questo approccio consente di verificare diversi rate all'interno della stessa misurazione, riuscendo così a fornire un risultato inviando un solo stream; inoltre l'aumento esponenziale delle velocità di spedizione riduce notevolmente il numero di pacchetti utilizzati e limita perciò l'invasività dello stesso strumento.

Per quanto valido e consolidato, l'approccio adottato da pathChirp è tuttavia perfettibile ed alcuni aspetti non sono stati ancora oggi sufficientemente approfonditi. Nonostante diversi gruppi di ricerca abbiano effettuato studi comparativi tra pathChirp e gli altri metodi esistenti, non è stato possibile rintracciare alcuna analisi approfondita sull'invasività del software. In questo lavoro di tesi si è pertanto cercato innanzitutto di individuare una formula in grado di descrivere esattamente il profilo e l'entità del traffico inviato dalle misurazioni, verificando poi quanto trovato con prove effettive. E' stato poi possibile riprodurre in laboratorio un problema di sovrastima, già noto e più volte citato in letteratura [12, 13], che affligge le misurazioni ottenute con pathChirp. La descrizione formale del traffico ricavata inizialmente ha permesso di individuare precise correlazioni tra gli errori prodotti e le proprietà del traffico; un'approfondita analisi ha poi consentito di individuare un errore presente nel codice sorgente del programma, causa della polarizzazione delle stime. Correggendo l'implementazione software dell'algoritmo di stima è stato così possibile ottenere risultati estremamente precisi, superiori a quelli forniti da qualsiasi altro strumento di stima attualmente disponibile. Poiché il software utilizza tuttavia valori discreti per le misurazioni, la possibilità di commettere errori è intrinseca nell'algoritmo: in questa tesi sono stati suggeriti alcuni accorgimenti da adottare per poter ridurre a piacere l'imprecisione, nell'ipotesi che la quantità di banda utilizzata rimanga costante sufficientemente a lungo.

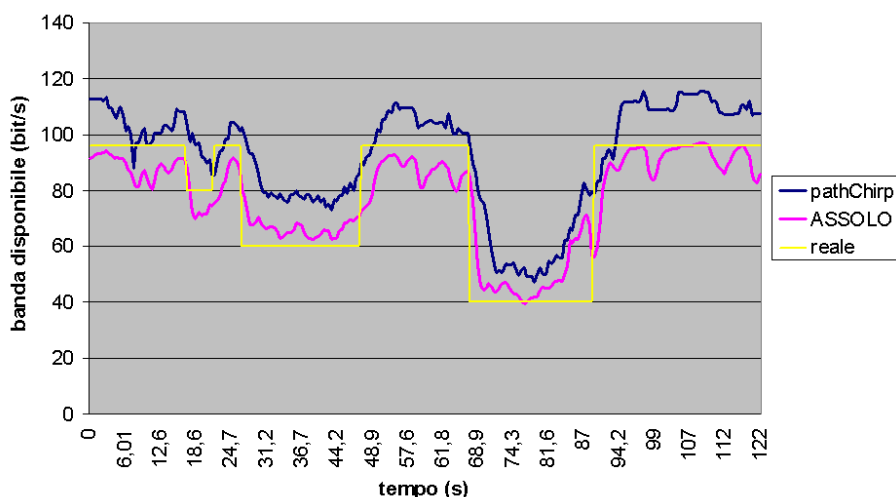


Figura 1.2: Confronto tra risultati ottenuti con pathChirp e con ASSOLO

Inoltre, sebbene gli autori di questo strumento abbiano profuso notevoli sforzi per minimizzare l'effetto di possibili errori nella determinazione di una misura, i risultati ottenuti con pathChirp risultano estremamente rumorosi. Per questo motivo il principale contributo alla soluzione del problema della valutazione della banda offerto da questa tesi sta nella realizzazione di ASSOLO (Available-bandwidth Smart Smoothing OnLine), una versione modificata di pathChirp che implementa un sistema di filtraggio on-line innovativo, computazionalmente efficiente ed in grado di sfruttare efficacemente le forti correlazioni esistenti tra misurazioni successive. Un esempio dei risultati ottenibili con il nuovo metodo è mostrato in Figura 1.2.

La tecnica identificata, mutuata da analoghe applicazioni in ambito finanziario [14, 15], permette inoltre di regolare la risposta del filtro stesso agendo su un numero di parametri limitato. I risultati ottenuti sono stati confrontati con altri filtri presentati in letteratura [16, 17]. Un algoritmo di quantizzazione con isteresi è stato infine proposto per l'impiego in tutte le applicazioni a carattere multimediale (voce e video), che richiedono stime piuttosto stabili e che sono in grado di adattare il rate di trasmissione a livelli discreti.

Altri esperimenti hanno poi permesso di verificare l'importanza delle ipotesi soggiacenti ai diversi metodi di stima della banda disponibile. Gli strumenti esistenti, pathChirp incluso, ipotizzano infatti che il tempo di servizio dei router sia costante; tuttavia si è riscontrato che, in talune condizioni, le tempistiche del software sono tali da non poter più essere trascurate e, al contrario, introducono un ulteriore ritardo non deterministico e non costante. Il problema è di attualità in tutte quelle realtà che hanno in programma di effettuare aggiornamenti nell'infrastruttura di rete e reso ancora più interessante dal fatto che è impossibile utilizzare molti degli strumenti di stima della banda oggi disponibili.

Per poter meglio spiegare quanto brevemente indicato in questa introduzione, la presente tesi è strutturata come segue. Dopo una breve introduzione alle reti a commutazione di pacchetto, trattate nel Capitolo 2, verrà analizzata nel dettaglio la modellizzazione di un dispositivo intermedio nel Capitolo 3. Nel Capitolo 4 sarà quindi introdotto il problema delle misurazioni nelle reti a pacchetto, presentando metriche ed approcci comunemente utilizzati. Il successivo Capitolo 5, dedicato nel dettaglio al problema della stima della banda disponibile, illustrerà invece lo stato dell'arte in materia presentando brevemente i metodi di stima proposti in letteratura e le varie problematiche riscontrabili durante le misurazioni. Nel Capitolo 6 verrà descritta la configurazione dell'ambiente di prova realizzato in laboratorio ed illustrato nel dettaglio il funzionamento del software pathChirp, utilizzato per le misurazioni. Il Capitolo 7 ospiterà infine la descrizione matematica del modello di traffico e il nuovo algoritmo di stima proposto ed implementato in ASSOLO, insieme ad un confronto dei risultati ottenuti in laboratorio con diversi filtri esistenti. Alcune riflessioni conclusive sul lavoro svolto e possibili sviluppi verranno presentati nell'ultimo capitolo.

REFERENCES

- [1] P. Baran. *On Distributed Communications*. The Rand Corporation, Memorandum RM-3420-PR, 1964.
- [2] G.D. Cole. *Computer network measurements: techniques and experiments*. Technical report, University of California, Los Angeles, October 1971.
- [3] L. Kleinrock and W.E. Naylor. *On measured behavior of the arpa network*. In AFIPS Press, editor, National Computer Conference, volume 43, pages 767-80, 1974.
- [4] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. *Framework for IP Performance Metrics*. RFC 2330 (Informational), May 1998.
- [5] J. Mahdavi and V. Paxson. *IPPM Metrics for Measuring Connectivity*. RFC 2678 (Proposed Standard), September 1999.
- [6] Nevil Brownlee and Chris Loosley. *Fundamentals of internet measurement: A tutorial*. CMG Journal of Computer Resource Management, Issue 201, 2001.
- [7] Bruce B. Lowekamp, Brian Tierney, Les Cottrell, Richard Hughes-Jones, Thilo Kielmann and Martin Swany. *A hierarchy of network performance characteristics for grid applications and services*. Global Grid Forum Proposed Recommendation, May 2004.
- [8] R. Prasad, M. Murray, C. Dovrolis, and K. Claffy. *Bandwidth estimation: Metrics, measurement techniques, and tools*, 2003.
- [9] Andreas Johnsson. *Modeling, Implementation and Evaluation of IP Bandwidth Measurement Methods*. PhD thesis, Department of Computer Science and Electronics, Mälardalen University, 2007.
- [10] Luca De Cicco, Saverio Mascolo, and Vittorio Palmisano. *An experimental investigation of the congestion control used by skype voip*. In Proceedings of Wired/Wireless Internet Communications, 5th International Conference, WWIC 2007, Coimbra, Portugal, May 23-25, 2007
- [11] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. *pathchirp: Efficient available bandwidth estimation for network paths*, 2003. Passive and Active Measurement Workshop (2003).
- [12] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, and K. Claffy M. Fomenkov. *Comparison of public end-to-end bandwidth estimation tools on high-speed links*. In Proceedings of 6th Passive and Active Measurements Workshop, PAM, Boston, San Diego Supercomputer Center, mar 2005.
- [13] A. A. Ali, F. Michaut, and F. Lepage. *End-to-end available bandwidth measurement tools : A comparative evaluation of performances*. ArXiv e-prints, 706:1-13, June 2007.

- [14] Tushar S. Chande. *Adapting moving averages to market volatility*. Stocks & Commodities, 10:3:108-114, 1992.
- [15] Adam White. *The vertical horizontal filter*. Futures Magazine, Volume XX, No. 10:1-10, 1991.
- [16] Lars Burgstahler and Martin Neubauer. *New modifications of the exponential moving average algorithm for bandwidth estimation*. In 15th ITC Specialist Seminar, Würzburg, Germany, jul 2002.
- [17] Minkyong Kim and Brian Noble. *Mobile network estimation*. In MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking, pages 298-309, New York, NY, USA, 2001. ACM Press.

Capitolo 2

Le reti a pacchetto

We will soon be living in an era in which we cannot guarantee survivability of any single point. However, we can still design systems in which system destruction requires the enemy to pay the price of destroying n of n stations. If n is made sufficiently large, it can be shown that highly survivable system structures can be built...

da *On Distributed Communications*
Paul Baran

Le reti di calcolatori comunemente utilizzate sono basate sulla commutazione di pacchetto (*packet switching*), un paradigma di comunicazione introdotto negli anni '60 e che oggi permette ad Internet di collegare ogni parte del globo. L'approccio adottato nelle reti a pacchetto risulta infatti più economico ed efficiente rispetto alla commutazione di circuito, tradizionalmente utilizzata nelle reti telefoniche, permettendo così la realizzazione di sistemi di comunicazione su larga scala e a costo contenuto.

2.1 Le origini

Le prime pubblicazioni nel campo della commutazione digitale, ad opera principalmente di Leonard Kleinrock [1], fornirono il punto di partenza per il lavoro di Paul Baran e Donald Davies, due ricercatori che, in maniera indipendente, introdussero e svilupparono il concetto di rete a commutazione di pacchetto. Baran definì i principi alla base delle reti a pacchetto nei laboratori della RAND Corporation, in parte finanziata dalla *U.S. Air Force*, durante il periodo di ricerca sulle reti di comunicazione impiegabili in ambito militare. Le prime idee, pubblicate già nel 1962, vennero espresse e presentate nel 1964 in una raccolta di undici volumi [2]. Baran descrisse una generica architettura per la costruzione di una rete estesa su larga scala, distribuita ed in grado di continuare ad operare anche in caso di guasti o caduta di nodi. L'idea alla base di tutto può essere riassunta in tre punti fondamentali: utilizzare una rete decentralizzata, con più di un cammino disponibile tra due qualsiasi punti, dividere i messaggi degli utenti in blocchi (successivamente chiamati pacchetti) e, per ultimo,

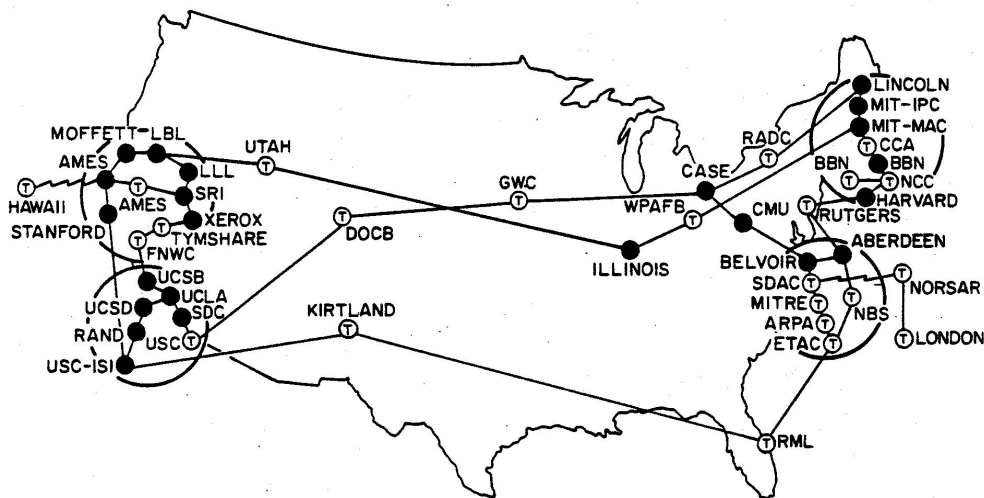


Figura 2.1: Mappa dei nodi connessi ad ARPANET (1974)

consegnare questi stessi messaggi utilizzando tecniche di inoltro *store-and-forward*¹. Negli stessi anni, come citato sopra, Donald Davies, un ricercatore inglese attivo presso il Laboratorio Nazionale di Fisica, sviluppò in maniera autonoma il principio della commutazione di pacchetto e propose di realizzare una rete globale oltremarina. Successivamente i due gruppi di ricerca si unirono ma è curioso osservare come Davies avesse determinato, in maniera del tutto indipendente, alcuni parametri di rete analoghi a quelli individuati da Baran². Gli studi in questo campo influenzarono tra gli altri Lawrence Robert al punto che quando venne incaricato di sviluppare Advance Research Projects Agency Network (ARPANET), la rete del Dipartimento della Difesa americano da cui poi nascerà Internet, decise di adottare proprio la tecnologia a commutazione di pacchetto.

2.2 Il funzionamento

Una generica rete di telecomunicazione è formata da due elementi fondamentali: i canali di comunicazione, chiamati spesso anche link, e i nodi. La rete può essere quindi considerata un sistema che interconnette tra loro gli end-system, cioè i nodi sorgente e destinatario delle comunicazioni; tale sistema complesso può essere visto come l'unione di sottosistemi più semplici e via dicendo, sino a giungere al sistema più elementare, ovvero il canale di trasmissione. I nodi intermedi hanno pertanto il compito di interconnettere tra loro i vari sottosistemi che formano la rete di TLC. Più precisamente, la comunicazione a pacchetto è una tecnica di multiplexing statico ove il canale fisico viene diviso in un numero arbitrario di canali logici, associati ai diversi flussi (*stream*) di dati. Ogni trasmissione inviata da un end-system è a sua volta divisa

¹Cioè si attende la ricezione dell'intero pacchetto prima di effettuare l'inoltro dei dati

²Ad esempio la dimensione del pacchetto, pari a 1024 bit

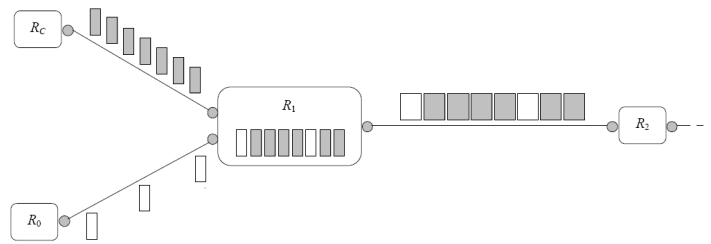


Figura 2.2: Più flussi di pacchetti possono condividere uno stesso canale (da [3])

in più “parti”, dette appunto pacchetti, aventi una lunghezza massima prestabilita e solitamente inoltrate dai diversi nodi presenti lungo il cammino in modalità asincrona. A ciascun pacchetto spedito vengono poi accodate informazioni aggiuntive, necessarie per determinare il percorso verso il destinatario; questo accorgimento permette di trasmettere lungo la rete ciascun pacchetto in maniera completamente indipendente dagli altri. I dati vengono infatti instradati verso la destinazione in base ad un algoritmo di routing e la scelta del percorso, effettuata da un apposito algoritmo, può basarsi su diversi criteri³ ed utilizzare varie figure di merito.

2.3 I vantaggi

L’impiego di questo approccio introduce numerosi benefici, permettendo di ottimizzare l’utilizzo dei canali, minimizzare le latenze o il tempo che i dati impiegano per la rete ed aumentare la robustezza della rete stessa.

2.3.1 Maggior efficienza

Nelle tradizionali reti telefoniche a commutazione di circuito, tutte le risorse necessarie al flusso comunicativo vengono riservate per l’intera durata della connessione: questo implica ad esempio che, per trasferire N comunicazioni tra due nodi adiacenti, sono necessari altrettanti canali. Al contrario, poiché le reti a pacchetto effettuano la commutazione esclusivamente sulla base del contenuto del pacchetto, l’intera banda è disponibile per più comunicazioni contemporanee⁴. La possibilità di utilizzare un unico canale per più flussi comunicativi consente perciò di sfruttare a pieno le risorse di ciascun link, nonché ridurre i costi di realizzazione della rete stessa.

2.3.2 Minori latenze

Un secondo beneficio derivante dall’impiego di questa tecnica di commutazione è noto come *pipelining*: i diversi canali (e, analogamente, i diversi nodi intermedi) possono trasmettere pacchetti in contemporanea e, come in una catena di montaggio, il pacchetto in output da un nodo diventerà poi l’input del nodo successivo presente lungo il percorso. Questo impiego simultaneo dei canali consente un elevato

³Ad esempio caratteristiche dei link statiche o dinamiche, numero di nodi attraversati etc...

⁴Per questo talvolta si dice che una rete a commutazione di pacchetto fornisce una connessione *logica* tra gli end-system

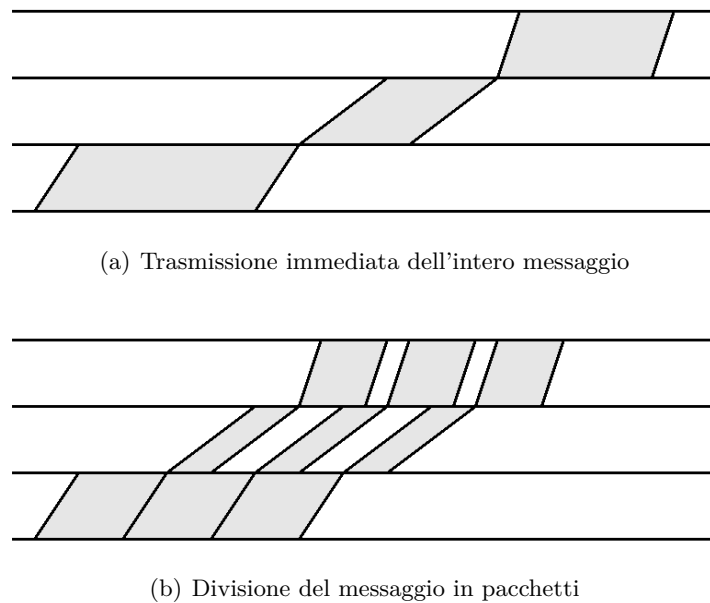


Figura 2.3: La divisione in pacchetti riduce il ritardo complessivo

incremento di efficienza, un maggior utilizzo dei link nonché una diminuzione del ritardo (*delay*) totale di trasmissione rispetto ad una rete che non divide i messaggi in pacchetti.

2.3.3 Maggior robustezza

La maggior robustezza della rete è sicuramente un ulteriore vantaggio derivante dall'adozione della commutazione di pacchetto. Poiché non esiste infatti un controllo centralizzato e vi sono solitamente più cammini alternativi per congiungere due punti della rete, la caduta di un nodo o di un canale consente comunque al resto del sistema di continuare ad operare. Inoltre, adottando opportuni algoritmi, la rete può riconfigurarsi dinamicamente per utilizzare nuovi percorsi. Questo fu uno dei motivi principali per cui la stessa ARPANET decise di adottare l'approccio a commutazione di pacchetto: all'epoca infatti, pur essendo i nodi di commutazione ed i collegamenti poco affidabili, l'obiettivo era quello di riuscire comunque a garantire, ai ricercatori delle varie università, l'accesso con continuità ai pochi supercalcolatori allora disponibili. Uno dei miti che circondano la rete ARPANET vuole che essa sia stata progettata per poter resistere anche ad un attacco nucleare; pur non essendo realmente questo uno degli scopi dei progettisti, taluni episodi hanno dimostrato, a distanza di molti anni, la robustezza dell'approccio anche in situazioni analoghe. Nel 2001 il *World Trade Center* ospitava infatti numerosi router di dorsale ed importanti link di interconnessione in fibra; ciò nonostante, la caduta delle Torri Gemelle non ha compromesso irrimediabilmente Internet. Tutti gli altri router, nelle ore successive, hanno infatti rilevato la caduta dei nodi di New York e modificato le regole di instradamento di conseguenza, permettendo alla quasi totalità del pianeta di continuare a comunicare senza particolari problemi [4].

2.4 Internet, la rete delle reti

Come già ricordato poco sopra, cioè che oggi conosciamo come Internet è in realtà l'evoluzione della rete ARPANET, attivata sperimentalmente nel 1969 [5]; da allora quest'ultima si è infatti evoluta, divenendo una complessa infrastruttura che interconnette tra loro centinaia di migliaia di reti universitarie, commerciali e governative. Le moltissime network che costituiscono oggi Internet sono solitamente gestite da operatori indipendenti che, per ragioni di business, si limitano ad utilizzare degli standard comuni e rispettare accordi di transito reciproco (*peering*). Un esempio di come diversi gestori di rete interoperino tra loro è dato dalla rete italiana GARR, che interconnette tutte le università e i centri di ricerca nazionali e che, a sua volta, si appoggia ad altre reti internazionali commerciali e non. Una panoramica più dettagliata della rete GARR è presentata in Appendice B. Per come è concepito Internet, non c'è quindi un'unità principale a cui tutti fanno riferimento o che controlla il traffico: le reti, seppur interconnesse tra loro, sono autonome e possono, al proprio interno, attuare politiche indipendentemente dagli altri operatori. Questa situazione ovviamente limita il controllo che gli operatori e, a maggior ragione, gli utenti possono avere; a questi ultimi, in particolare, Internet appare come una “*black box*” in grado di trasferire informazioni da un capo all'altro. Ciò discende anche da quello che è uno dei criteri ispiratori dell'architettura della Rete: il principio del controllo *end-to-end*. L'idea di fondo è che i diversi nodi interni della rete, i router, abbiano funzionalità estremamente ridotte, limitandosi sostanzialmente al solo inoltro del traffico da un canale ad un altro. Al contrario, il controllo di come e se inviare traffico sulla rete è prerogativa esclusiva degli *end point*, situati sul “bordo” della rete. Questo è peraltro in contrasto con le tradizionali reti telefoniche, dove l'intelligenza è inserita nei nodi interni della rete.

2.5 Ulteriori approfondimenti

Quella delle *packet switching network* è indubbiamente una tematica complessa e richiederebbe molto più spazio per essere adeguatamente affrontata; ai fini di questa trattazione è tuttavia sufficiente sapere che in queste reti di comunicazione i dati vengono scambiati in forma di pacchetti e che gli stessi pacchetti, quando ricevuti da un nodo intermedio, vengono ritrasmessi su un altro link per giungere così verso il router successivo e, infine, al destinatario. Non è necessario essere a conoscenza di ulteriori dettagli circa il funzionamento e le caratteristiche dei protocolli utilizzati comunemente in Internet e nelle più diffuse reti locali; in Appendice A è possibile comunque trovare una breve descrizione della suite TCP/IP, la cui lettura potrebbe risultare utile per una miglior comprensione dei capitoli seguenti.

Capitolo 3

I dispositivi di interconnessione

Osservazione di Zenone: “*L'altra coda va più veloce.*”

Variante di O'Brien sull'osservazione di Zenone: “*Se si cambia coda, quella che si è appena lasciata diventerà immediatamente la più veloce.*”

Corollario di Kenton: “*Continuare a cambiar coda le rallenta entrambe e fa infuriare tutti.*”

da *La legge di Murphy*

Un pacchetto, inviato attraverso una rete di comunicazione, deve solitamente attraversare più dispositivi intermedi prima di giungere a destinazione. Ciascuno di essi può essere pensato come un elemento avente più interfacce per la ricezione e la spedizione dei pacchetti, ciascuna con una o più code associate, ed una CPU, necessaria per analizzare i pacchetti in esse presenti e determinare la porta di uscita, una volta eseguiti gli opportuni algoritmi di ricerca. Questi dispositivi, chiamati più comunemente nodi, possono operare a livello 2 o 3 del modello Open Systems Interconnection (OSI) [6] [7] e funzionano solitamente in modalità *store-and-forward*. I dispositivi di maggior interesse sono i router, operanti a livello Network; la trattazione successiva riguarderà quindi questi nodi, anche se i concetti fondamentali introdotti rimangono validi, seppur con alcune semplificazioni, anche per gli apparati di livello 2 quali, ad esempio, i *bridge* Ethernet¹.

3.1 Il modello a code

Si consideri un semplice modello a code che descrive il funzionamento di un nodo di commutazione intermedio: un pacchetto che proviene da un'interfaccia di ingresso viene accodato in attesa della commutazione e, successivamente, la CPU del nodo determina il percorso e quindi lo invia ad una coda associata ad un'interfaccia d'uscita, in attesa della trasmissione.

L'esistenza delle code è facilmente comprensibile se si considera che in queste reti non vi è solitamente alcun sistema di controllo centralizzato del traffico. Talvolta

¹Meglio noti come *switch*

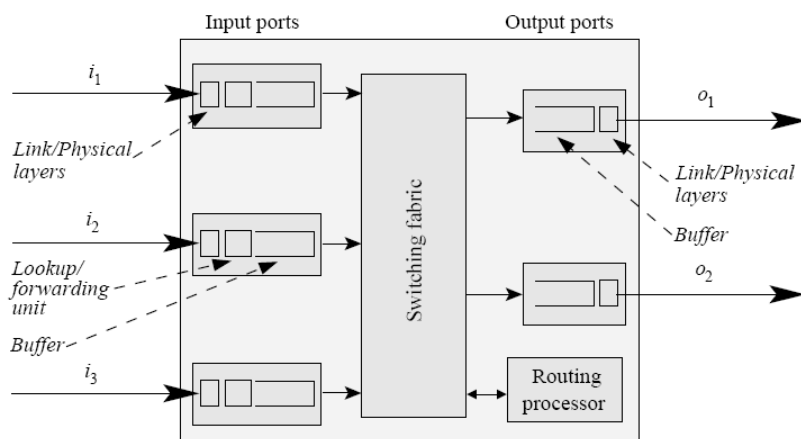


Figura 3.1: Modello a code utilizzato per descrivere un router (Fonte [8])

può infatti accadere che i pacchetti ricevuti da un *host* giungano con un tasso che eccede le capacità del server o dei link in uscita, congestionando così il nodo in questione. Per poter gestire questi sovraccarichi temporanei vengono impiegati dei *buffer* di memorizzazione sulle interfacce in ingresso e uscita: il surplus di dati ricevuti non viene così perso ma rimane in attesa che il server esaurisca tutte le richieste giunte in precedenza. In questa situazione i pacchetti che vengono accodati subiscono un ritardo extra, che si aggiunge al tempo di ritardo accumulato dal segnale per la propagazione². Se poi la situazione di sovraccarico permanesse per un tempo sufficiente, i buffer esaurirebbero la propria capacità e tutti i pacchetti successivi inviati al nodo verrebbero scartati.

3.1.1 La gestione delle code

Il fatto che i pacchetti vengano talvolta accodati comporta anche la necessità di interrogarsi sui criteri adottati per stabilire l'ordine di prelievo dalla coda e la trasmissione. Ciò viene determinato dalla politica di *scheduling* del router: la più semplice, e anche la più utilizzata in assoluto in Internet, è First Come First Served (FCFS)³ ma esistono tuttavia discipline più sofisticate quali Fair Queuing (FQ) e Weighted Round Robin (WRR) [9]. La ragione principale per adottare una particolare disciplina sta nella necessità di differenziare il servizio offerto a diversi traffici: è infatti possibile prevedere che ad un certo flusso sia garantita una determinata porzione della capacità di link, piuttosto che un limite massimo sui tempi di permanenza in coda. Per esempio, per un flusso che trasporta dati voce, la varianza nel ritardo (il cosiddetto *jitter*) dei pacchetti risulta problematica se troppo ampia; in questi casi è necessario imporre alcuni vincoli sulle tempistiche, impossibili da garantire con una schedulazione FCFS. Collegate a queste problematiche vi sono anche le tecniche di gestione attiva delle code (AQM) [10], quali DropTail ed il più evoluto Random Early

²Ai fini della trattazione, i tempi di servizio dei nodi verranno solitamente considerati trascurabili e il tasso di servizio supposto quindi uguale alla capacità del link

³Ovvero i pacchetti vengono ritrasmessi nello stesso ordine in cui sono giunti nella coda

Detection (RED) [11]; tali algoritmi intervengono sui *buffer* decidendo quando e quali pacchetti scartare, al fine di prevenire o uscire da una situazione di congestione.

3.2 Gli effetti di accodamento

In generale quindi, per quanto basati su semplici principi di funzionamento ed in grado di effettuare un numero limitato di operazioni sul contenuto dei pacchetti in transito, questi dispositivi possono perciò intervenire attivamente sulla distribuzione temporale del traffico in transito. In particolar modo l'esistenza degli accodamenti lungo il percorso risulta fondamentale per eseguire misurazioni sullo stato della rete. Si consideri a questo proposito il modello semplificato di rete illustrato in Figura 3.2, utilizzato per descrivere il cammino (*path*) che collega la sorgente S al destinatario D . Ogni elemento di commutazione intermedio dispone di una sola coda in entrata, connessa ad un link, ed una coda in uscita, a sua volta connessa in cascata ad un altro dispositivo; il tempo di servizio μ è supposto costante. Sulla rete, insieme al flusso di pacchetti P inviato dal mittente, può essere presente anche un ulteriore traffico X .

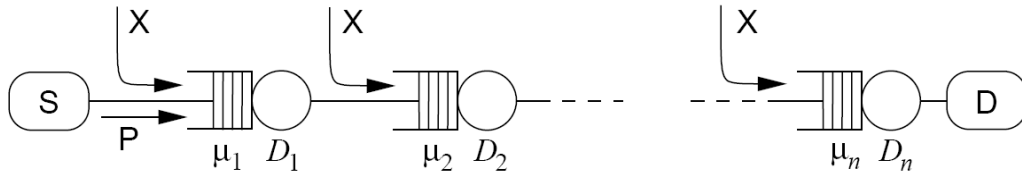


Figura 3.2: Modello semplificato di rete di code (da [12])

3.2.1 Coppie di pacchetti

L'effetto del ritardo introdotto dagli accodamenti può essere facilmente verificato inviando due pacchetti di dimensione identica S_p *back-to-back*, ovvero senza che vi sia una pausa tra l'istante di fine trasmissione del primo e l'inizio della trasmissione del secondo. Se non vi è altro traffico in rete, la differenza dei tempi di arrivo dei due pacchetti sul destinatario (anche detto tempo di interarrivo) corrisponde infatti esattamente alla spaziatura introdotta dal link strozzante, quello avente cioè una minor velocità della CPU del server o un canale con un rate⁴ di trasmissione inferiore rispetto agli altri.

Effettuando una misurazione di questo Δt di interarrivo sul destinatario, è possibile ricavare facilmente il tasso di servizio μ del collo di bottiglia:

$$\mu = \mu_2 \frac{S_p}{\Delta t}$$

Il discorso potrebbe essere esteso anche a coppie di pacchetti di dimensione differente e con spaziatura variabile, per quanto una trattazione analitica dei metodi basati su dimensione variabile dei pacchetti (*Variable Packet Size*) esulerebbe dagli scopi di questa trattazione; a tal proposito si veda [13], [14] e [15]

⁴Lett. tasso ma in questo testo è spesso tradotto, seppur impropriamente, anche con velocità

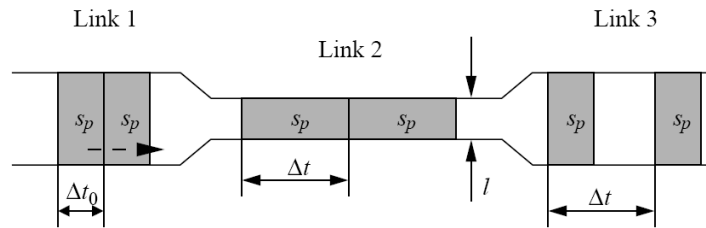


Figura 3.3: Effetti di accodamento su coppie di pacchetti (da [12])

3.2.2 Treni di pacchetti

Si consideri invece il caso di una serie di pacchetti distanziati tra loro e inviati verso lo stesso router; se quest'ultimo ha una capacità di servizio superiore a quella del mittente, riuscirà ad instradare ogni pacchetto prima che sia giunto il successivo. Viceversa, un *throughput* inferiore in spedizione costringerebbe i pacchetti a sostare in una coda per il tempo necessario al router per soddisfare le richieste giunte in precedenza. Come già ricordato, i *buffer* svolgono quindi una funzione fondamentale, poiché consentono di gestire temporanee condizioni di congestione; un trend crescente nel ritardo di un flusso unidirezionale può perciò essere considerato un indicatore di un'imminente congestione del cammino di rete. Alcuni degli strumenti di misura che verranno in seguito presentati utilizzano proprio un approccio di questo tipo per determinare le caratteristiche di una rete.

3.2.3 Cross-traffico

Un ulteriore aspetto da non trascurare è la presenza di traffico in rete diverso da quello considerato, anche chiamato cross-traffico. In particolare i router dispongono di più interfacce: è quindi possibile che differenti flussi di traffico debbano essere inoltrati dalla stessa interfaccia di uscita. In questo caso è compito del server decidere, in base alla disciplina adottata, l'ordine di instradamento dei pacchetti ricevuti: per semplicità si suppone solitamente che la politica adottata sia FCFS, anche se nulla vieta che vengano preferite altre tecniche. Indipendentemente da quale sia la disciplina impiegata, la presenza di cross-traffico implica comunque la necessità di mescolare i flussi di pacchetti e pertanto, con buona probabilità, modificare le tempistiche rispetto a quelle stabilite dal mittente.

3.3 Ulteriori approfondimenti

Esiste peraltro una vera e propria teoria di modellizzazione delle code che permette di descrivere realtà anche molto complesse, nonché di individuare la probabilità che il sistema si trovi in un determinato stato ad un certo istante. Sebbene risulti fondamentale per descrivere formalmente i fenomeni di accodamento sopra citati e sia largamente impiegata al calcolatore nelle simulazioni di rete, la teoria dei modelli a reti di code non verrà però approfondita ulteriormente. Un approccio di questo tipo è infatti difficilmente proponibile nel caso di grandi reti a pacchetto e, in particolare, per Internet. In queste realtà è impossibile conoscere con precisione di

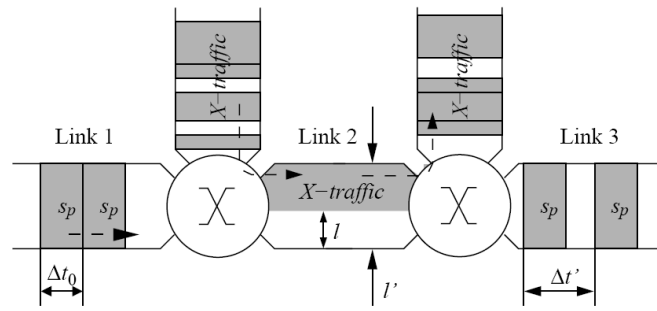


Figura 3.4: Effetti di spaziatura imputabili alla presenza di cross-traffico (da [12])

quante interfacce dispongano i vari nodi attraversati, quando siano lunghe le code, che discipline vengano adottate per la gestione e, soprattutto, come possa essere descritto in termini probabilistici il traffico. Per ulteriori informazioni sui modelli a code è possibile fare riferimento a [16] e [17], per citare solo alcuni dei numerosi testi disponibili sull'argomento.

Capitolo 4

Le misurazioni in rete

Effettuare misurazioni nelle reti di telecomunicazione è da sempre una delle principali necessità per chi opera nel settore: le prime pubblicazioni riguardanti le prestazioni di ARPANET [18] [19] risalgono infatti ai primi anni '70. Prima però di iniziare ad analizzare più in dettaglio i diversi strumenti di misurazione, è opportuno introdurre alcuni concetti fondamentali per il seguito della trattazione. Sino ad ora, infatti, si è parlato in generale di misurazione ma non è stato chiarito quali siano effettivamente le caratteristiche di una rete fondamentali per poterne valutare le prestazioni e come queste possano essere rilevate. A questo proposito occorre sottolineare come non esista effettivamente uno standard né regole universalmente accettate che definiscano la terminologia da adottare per la caratterizzazione delle *performance* di rete. Negli anni infatti gruppi di lavoro dell'IETF, del Global Grid Forum e ricercatori di diverse università e centri di ricerca, primo tra tutti il CAIDA¹, hanno proposto convenzioni per definire metodologie e metriche di misurazione [20] [21] [22] [23]. Di seguito verranno presentati ed adottati quelli che risultano essere i termini oggi più diffusi in letteratura.

4.1 Modalità di misurazione

Le modalità con cui possono essere effettuate le rilevazioni sono varie e ancora oggi oggetto di indagine. E' possibile individuare tre macro-famiglie in cui classificare le tecniche di misurazione: passive, attive e miste.

4.1.1 Misurazioni passive

Le prime hanno l'indubbio vantaggio di non perturbare il traffico esistente e non sovraccaricare la rete. Per effettuare una misurazione passiva di un flusso occorre però che, lungo il percorso, siano presenti dei rilevatori di traffico. Questi devono poi essere in grado di determinare le prestazioni della rete ed il comportamento dei vari flussi di pacchetti semplicemente osservando il traffico di passaggio, senza alterarlo. Una misurazione passiva richiede quindi hardware, software e protocolli adeguati a questo scopo, nonché la possibilità di controllare gli stessi nodi di misurazione lungo tutto il percorso. Questo approccio è quindi proponibile in reti di dimensione

¹Cooperative Association for Internet Data Analysis, San Diego CA

limitata e sulle quali si ha il pieno controllo; misurazioni passive sono infatti adottate quasi esclusivamente in reti locali [24], sistemi di prova controllati o simulati al calcolatore².

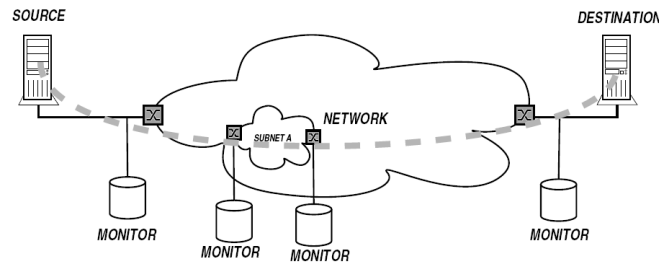


Figura 4.1: Rete con misurazioni passive lungo l'intero percorso (da [25])

4.1.2 Misurazioni attive

Più comunemente la rete di interconnessione viene considerata alla stregua di una *black box* e, per le due entità comunicanti, l'unico metodo possibile è effettuare misure attive *end-to-end*. In questo caso il sistema è costituito da un processo in esecuzione sul nodo sorgente ed uno passivo, in ascolto sul destinatario; la misurazione viene effettuata dal sorgente inviando del traffico di prova (*probe flow*), spedito esclusivamente per questo fine verso il ricevente. Supponendo nota la distribuzione iniziale dei pacchetti immessi sul percorso è possibile, analizzando come questa viene modificata dall'attraversamento della rete, effettuare delle valutazioni sullo stato della rete.

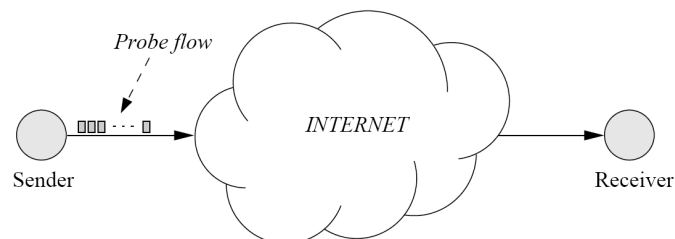


Figura 4.2: Misurazioni attive in rete con traffico di prova (da [8])

In realtà, anche così facendo, le informazioni ottenibili sono limitate e incomplete. In primo luogo poiché è possibile studiare esclusivamente lo specifico cammino attraversato dal flusso di prova e non l'intera rete. Inoltre questo genere di misurazioni è ideale per determinare le caratteristiche sperimentate “ai morsetti esterni” e non invece le peculiarità di uno specifico link; in altre parole, da un singolo flusso di pacchetti è possibile solo dedurre che uno dei canali lungo il percorso ha determinate proprietà ma, solitamente, non quale sia quello incriminato. Infine, dal momento che la conoscenza della rete avviene in maniera indiretta, è possibile determinare in

²Il più utilizzato in ambito accademico è probabilmente *ns-2* (<http://www.isi.edu/nsnam/ns/>)

che stato essa si trovi ma non quale sia la causa. Lo svantaggio principale di queste misurazioni rimane tuttavia il loro stesso carattere attivo e, quindi, invasivo: una delle grandi sfide per chi progetta e realizza questo tipo di strumenti è perciò riuscire ad avere valori il più possibile corrispondenti alla realtà, minimizzando nel contempo l'intrusività. I vantaggi di tali rilevazioni sono tuttavia numerosi: variando infatti la distribuzione, la dimensione o le tempistiche di invio dei pacchetti, è possibile focalizzare la misurazione su differenti aspetti e, rispetto ad una misurazione passiva, la quantità di dati da analizzare è molto più contenuta.

4.1.3 Misurazioni miste

Le misurazioni miste non prevedono invece l'invio di pacchetti specifici, bensì sfruttano il traffico generato da altri protocolli per effettuare i calcoli. Un esempio è il protocollo TCP Westwood/Westwood+ [26] [27] [28] [29] [30], un'estensione del tradizionale Transmission Control Protocol in grado di utilizzare i pacchetti dati e gli ACKnowledgment (ACK) di conferma, ricevuti durante la connessione, per stimare la banda disponibile e regolare, di conseguenza, l'apertura della finestra di congestione. Seppur legati ad uno specifico protocollo, questi meccanismi possono aumentare notevolmente l'efficienza in reti Wireless.

4.2 Le metriche

Il cammino di rete che congiunge due *host* consiste fisicamente in una sequenza di canali e nodi di commutazione che possono presentare caratteristiche anche molto differenti tra loro: i mezzi trasmissivi utilizzati oggi infatti sono i più disparati, dal filo elettrico alla fibra ottica o persino lo spettro radio. Definire delle caratteristiche facilmente misurabili e confrontabili è perciò fondamentale per poter valutare le prestazioni di una rete di comunicazione anche piuttosto complessa.

4.2.1 Il prodotto banda-ritardo

Una delle metriche più utilizzate per descrivere un singolo canale di comunicazione è il cosiddetto prodotto banda-ritardo. Il ritardo di un link, più propriamente detto ritardo di propagazione, è il tempo che richiede un bit per essere trasferito da un capo all'altro del canale ed è caratteristico dello specifico mezzo trasmissivo utilizzato. Nelle comunicazioni a livello fisico il termine *bandwidth* (larghezza di banda o, più brevemente, banda) si riferisce all'ampiezza spettrale del segnale elettromagnetico o alle caratteristiche del sistema di comunicazione. Nel contesto delle reti digitali, la banda è un valore che indica invece la quantità di dati che il link è in grado di trasmettere nell'unità di tempo; dal momento che l'unità base è il *bit*, la banda si misura solitamente in *bit* per secondo (*bps*). Sulle dorsali principali oggi non è infrequente trovare connessioni con capacità nell'ordine della decina di *Gigabit*³ per secondo; a queste si connettono poi le reti, dette *Access network*, che connettono gli host ai collegamenti principali. In questo caso la velocità scende notevolmente,

³Un Gigabit, abbreviato in *Gbit* o *Gb*, equivale a 10^9 *bit* = 1.000.000.000 *bit*

nell'ordine dei 10-100 *Mbps*⁴. Le reti residenziali di accesso, che connettono i computer domestici al Web, hanno poi velocità ancora più modeste: la banda larga oggi più diffusa in Italia è forse la Asymmetric Digital Subscriber Line (ADSL), con velocità di alcuni *Megabit* per secondo. Fino a non molto tempo fa, infine, l'unico mezzo per accedere ad Internet (e, per molte località, ancora oggi l'unico disponibile) era la tradizionale linea telefonica: in questo caso le capacità sono decine di volte inferiori a quelle della stessa ADSL. In sostanza, quindi, Internet è estremamente eterogeneo sia in termini di nodi che sono connessi ad esso, sia per quel che riguarda le caratteristiche banda-ritardo dei canali di comunicazione da cui è costituito. Lo stesso termine banda può, in realtà, indicare diverse proprietà, siano esse riferite ad un singolo collegamento diretto tra due nodi (anche detto *hop*) o ad un cammino *end-to-end*. In particolare si definisce capacità la quantità massima di dati trasferibili nell'unità di tempo e banda disponibile la percentuale inutilizzata in un canale o in *path*; talvolta si fa riferimento anche alla Bulk Transfer Capacity, una terza misura di seguito illustrata più in dettaglio.

4.2.2 La capacità

Un segmento di rete è normalmente in grado di trasmettere dati ad un *bit rate* costante⁵, limitato sia dalla larghezza di banda del mezzo di propagazione fisico che dalle tempistiche dei dispositivi di ricezione/trasmissione ottici o elettronici. Inoltre, a livello IP, un collegamento consegna i pacchetti ad una velocità inferiore a quella nominale a causa dell'*overhead* di incapsulamento H_{L2} introdotto a livello 2. E' quindi possibile definire la capacità C_i di un singolo collegamento come la velocità massima di trasmissione sperimentabile a livello IP. Con semplici considerazioni si può quindi descrivere analiticamente C_i in funzione della capacità nominale C_{nom} , della quantità di overhead H_{L2} introdotta e della massima dimensione del pacchetto supportata dal link (MTU):

$$C_i = C_{L2} \frac{1}{1 + \frac{H_{L2}}{MTU}}$$

La definizione precedente può essere estesa ad un cammino per indicare il massimo rate di trasmissione supportato dal cammino di rete in un trasferimento *end-to-end*. In questo caso è l'*hop* del percorso avente la capacità inferiore, detto *narrow link*⁶, a determinare la capacità massima sperimentabile *end-to-end*:

$$C = \min_{i=1, \dots, H} C_i$$

⁴Il Megabit (*Mb*) è l'unità di misura più utilizzata per descrivere la capacità delle reti di comunicazione ed equivale a 10^6 *bit* = 1.000.000 *bit*

⁵Questo non è vero invece per i link Wireless IEEE 802.11, che possono trasmettere i dati a velocità variabile, in funzione del *bit error rate* sperimentato nella comunicazione via etere

⁶Un termine molto utilizzato in questi casi è *bottleneck*, cioè collo di bottiglia; purtroppo quest'ultimo in letteratura è stato alternativamente utilizzato per indicare sia il link avente la capacità minima, sia quello con la banda disponibile minima. Per evitare ambiguità, si è preferito adottare i termini in lingua originale proposti in [31]

4.2.3 La banda disponibile

Altro importante parametro di caratterizzazione delle prestazioni di rete è la banda disponibile (*available bandwidth*) in un collegamento, ovvero la quantità di capacità inutilizzata del link in un certo periodo di tempo. Perciò, mentre la capacità di un collegamento dipende dallo specifico mezzo trasmissivo e dalle tecnologie impiegate, la banda disponibile è un parametro tempo-variante anche legato alla quantità di traffico presente sul link nel momento di osservazione. In un dato istante di tempo, infatti, un link⁷ può generalmente essere impegnato nel trasferimento di un pacchetto al massimo rate oppure essere inattivo (*idle*); l'utilizzo istantaneo U del canale perciò può essere solo 0 o 1.

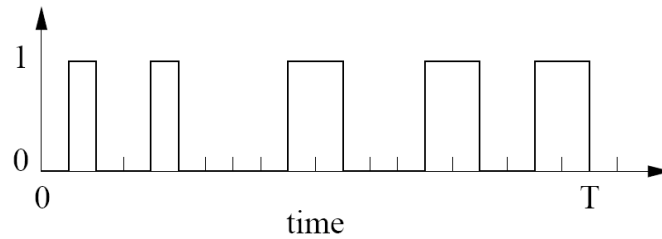


Figura 4.3: Occupazione effettiva della banda nel tempo (da [32])

Da ciò discende la necessità di definire la banda disponibile introducendo un intervallo di tempo, sul quale verrà poi effettuata una media degli utilizzi istantanei. Detto $[t - \tau, t]$ l'intervallo di interesse, l'utilizzo medio nell'intervallo considerato $U(t - \tau, t)$ è così definito:

$$U(t - \tau, t) = \frac{1}{\tau} \int_{t-\tau}^t u(x) dx$$

Noto quindi l'utilizzo medio nell'intervallo di tempo, è immediato ricavare la banda media disponibile (o, più brevemente, banda disponibile) per l' i -esimo *hop* in questione e riferita allo stesso lasso di tempo; questa infatti altro non è che la percentuale inutilizzata di capacità:

$$A_i = (1 - u_i)C_i$$

Il concetto può poi essere esteso per un cammino composto da un numero qualsiasi H di hop; in questo caso la banda disponibile sperimentabile dai due *end point* è pari al valore minimo misurabile sui link attraversati dalla trasmissione.

$$A = \min_{i=1, \dots, H} A_i$$

Il cammino avente maggior disponibilità di banda, cioè percentuale di utilizzo maggiore, è detto *tight link*⁸.

⁷Per link in questo caso si intende il collegamento di comunicazione 'logico' tra le due entità e non lo specifico canale trasmissivo, che in realtà potrebbe consentire più trasmissioni in contemporanea e non essere quindi mai completamente inattivo

⁸ Cfr. nota 6

4.2.4 Bulk Transfer Capacity

La Bulk Trasfer Capacity (BTC) [33] è una misura della velocità di una rete nel trasferire dati attraverso una singola connessione TCP ideale. Poiché una gran parte del traffico in Internet utilizza proprio questo protocollo, tali misurazioni risultano essere di notevole interesse; tuttavia vi sono numerosi aspetti che rendono complesso effettuare misurazioni di questo tipo e, ancora più paradossale, definire in maniera univoca il concetto stesso di BTC. Esistono infatti diverse varianti implementative del protocollo TCP, ad esempio Reno [34], NewReno [35] [36], Taohe [37], SACK [38], Eifel [39] e Vegas [40] [41] per citare solo le più diffuse. Molti altri fattori possono influenzare il comportamento e la stabilità [42] di una connessione: la quantità di dati da trasferire, la natura e la stabilità del traffico già presente in rete, la dimensione dei buffer TCP sul mittente e sul ricevente etc. . .

4.3 Ulteriori approfondimenti

Insieme a quelle sopra presentate, esistono numerose altre metriche interessanti per determinare lo stato di una rete, come la latenza in una direzione⁹, il Round Trip Time (RTT) [43] [44], particolarmente utile nel caso di cammini asimmetrici e utilizzato anche per determinare le tempistiche di trasmissione in alcuni protocolli, o il tasso di perdita dei pacchetti [45]; in questa trattazione ci si concentrerà però sulla stima banda disponibile, problematica che ancora oggi, dopo diversi anni, continua a rappresentare una sfida per i ricercatori e che risulta di grande interesse in molte realtà.

⁹In letteratura indicata più brevemente con One Way Delay (OWD)

Capitolo 5

La stima della banda disponibile

Just because something doesn't do what you planned it to do doesn't mean it's useless.

Thomas A. Edison

Una definizione intuitiva della banda (media) disponibile può essere data considerando un link con una determinata capacità, ovvero il massimo *rate* di trasmissione raggiungibile dal canale, ed un singolo *host* in grado di spedire informazioni. In questa situazione infatti, anche se il nodo fosse in grado di generare dati più velocemente, il link sarebbe in grado di trasferirli solo al massimo rate supportato ed i pacchetti verrebbero trasmessi *back-to-back*, sfruttando al massimo la capacità del canale. In caso contrario, una certa porzione di banda del link rimarrebbe inutilizzata; questo *surplus*, corrispondente in Figura 5.1 alla distanza tra i vari pacchetti, rappresenta la porzione di banda teoricamente disponibile per altre trasmissioni.

Normalmente un link è condiviso tra più *host* ma il discorso appena fatto non perde di validità e, anche considerando un cammino di rete composto da più link, la banda disponibile corrispondente equivale al surplus inferiore tra tutti quelli dei collegamenti appartenenti al path. Spesso viene utilizzata una rappresentazione grafica del traffico come in Figura 5.2, ove il traffico viene modellizzato come un liquido ed i link come tubature. Il traffico (ideale) è descritto infatti come un fluido: ad un incremento infinitesimo della quantità di dati trasmessi corrisponde un innalzamento analogo nell'altezza del livello. La larghezza A_i di ciascun tubo indica la rispettiva

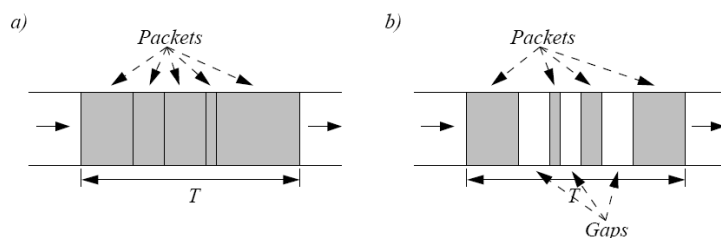


Figura 5.1: Esempio di canale a) completamente utilizzato e b) con ancora una porzione di banda disponibile (da [8])

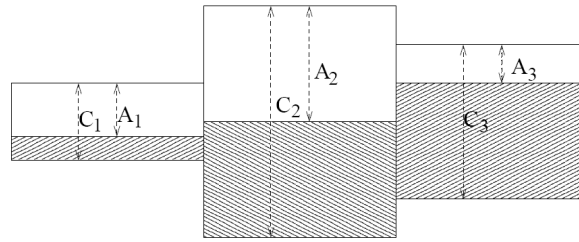


Figura 5.2: Il modello di traffico in rete fluido (da [31])

capacità del link, mentre lo spazio libero C_i rappresenta la portata inutilizzata, ovvero la banda disponibile.

Un grafico di questo tipo è sicuramente molto efficace nel mostrare come il link del cammino con minor capacità non coincida necessariamente con quello avente banda disponibile inferiore, nonostante alcuni strumenti di misura si basino su questa ipotesi. Per contro, questo modello introduce ovviamente una forte semplificazione, trascurando la natura discreta dei pacchetti e la variabilità del traffico su diverse scale temporali.

5.1 L'importanza della banda disponibile

L'individuazione di metodi affidabili ed efficaci di stima della banda disponibile è una necessità per molte applicazioni *data-intensive*, quali ad esempio il trasferimento di file o la distribuzione di contenuti multimediali, ove la disponibilità della rete ha un impatto diretto sulle performance degli stessi servizi. Molte altre applicazioni e componenti tecnologiche possono poi utilizzare proficuamente la conoscenza della banda; le applicazioni P2P, ad esempio, formano dinamicamente la rete tra gli utenti in funzione della banda disponibile tra i vari peers. Discorso analogo alle reti peer-to-peer potrebbe valere per tutte le *overlay network*, che sono in grado di configurare le proprie regole di instradamento in base alla banda dei diversi link. Misurazioni di questo genere sono utilizzate anche dai diversi operatori, che pianificano incrementi nella capacità della rete basandosi sul tasso di crescita dell'utilizzo della rete da parte degli utenti. Gli stessi operatori di rete fatturano spesso ai clienti in base alla quantità di banda disponibile acquistata, definita in un Service Level Agreement. Un'ulteriore tecnologia, destinata a conquistare sempre più utenti (e fette di mercato) nonché in grado di trarre grandi benefici dalla stima della banda è il Voice over IP (VoIP); grazie alla possibilità di variare il livello di compressione della voce è infatti possibile adattare la qualità del suono in funzione del traffico presente in rete, mantenendo così un buon livello di fluidità del parlato.

5.2 Lo stato dell'arte

Nonostante i molti campi di applicazione possibili, i lavori focalizzati sulla stima della banda disponibile prodotti sono decisamente pochi, se confrontati con la mole di ricerche che hanno interessato le tecniche per la misura della capacità. Per poter meglio illustrare lo stato dell'arte si è cercato di individuare le caratteristiche di fondo

Categoria	Metodo	Anno
Affini	Packet-Pair	1995
	Cprobe	1996
	Coeff. β	1998
PGM	Delphi	2000
	IGI	2003
	Spruce	2003
PRM	TOPP	2000
	PTR	2003
	pathChirp	2003
	Pathload	2003
	BART	2006

Tabella 5.1: Software di stima della banda attualmente disponibili

dei diversi metodi proposti negli anni. Gli strumenti (*tool*) di misura attualmente disponibili, riassunti Tabella 5.1, sono quindi stati raggruppati in tre macro-categorie in base al principio di funzionamento.

5.2.1 I modelli affini

A questa categoria appartengono diversi strumenti che, seppur differenti tra loro, hanno in comune la caratteristica di non individuare direttamente la banda disponibile ma altri parametri ad essa correlati. I primi tentativi di misurare la banda disponibile sono misure indirette: tra questi vanno elencati il metodo Packet Pair, proposto da Keshav, i pionieristici tool Cprobe e pipechar, così come il coefficiente relativo di banda disponibile β , definito da Paxson. Non tutte queste tecniche sono nate come affini ma, ad una analisi più approfondita, hanno mostrato come in realtà quanto misurato non fosse effettivamente la banda disponibile.

Packet Pair

Il primo tentativo di stima della banda disponibile effettuando misurazione end-to-end si deve a Keshav [46] [47] [48]. Il metodo Packet Pair da lui proposto si basa sull'effetto di spaziatura introdotto dai router intermedi strozzanti: due pacchetti, aventi uguale dimensione, vengono inviati in sequenza lungo la rete ed il destinatario, una volta ricevutigli, risponde inviando a sua volta due pacchetti di conferma al mittente. La banda disponibile viene ricavata calcolando la dispersione, cioè la differenza nei tempi di arrivo, dei pacchetti; questo è possibile solo se ipotizza che la dispersione dei pacchetti di ACKnowledgment sia la medesima di quelli inizialmente spediti. In altre parole, occorre supporre che tutti i router intermedi adottino una politica di Fair Queuing *per-flow*. L'idea di ottenere informazioni dalle tempistiche dei pacchetti di conferma anziché direttamente da quelli di prova nasconde però alcune insidie: per fare solo alcuni esempi, potrebbero esistere instradamenti asimmetrici oppure fenomeni di interferenza su canali condivisi tra i pacchetti di misura e gli ACK. L'ipotesi di gestione delle code con schedulazione FQ su tutti i router è poi,

nella realtà, eccessivamente restrittiva; come già detto, la disciplina più adottata in Rete è infatti FCFS. Poiché i flussi che attraversano i link non vengono quindi trattati in maniera equa, il metodo proposto è sostanzialmente inutilizzabile in Internet e nelle reti locali.

Cprobe

Cprobe [49], primo metodo effettivamente implementato in un software di misurazione, utilizza un approccio simile a Packet Pair: in entrambi viene infatti sfruttata la dispersione dei pacchetti di probe, ricavando le tempistiche dalle conferme. La differenza principale di Cprobe consiste nel numero di pacchetti inviati: anziché una coppia, questo metodo prevede l'invio di un treno (*train*) di otto pacchetti consecutivi, tutti *back-to-back*. Un approccio analogo è stato utilizzato nel software pipechar, ipotizzando anche in questo caso che la dispersione dei treni di pacchetti sia inversamente proporzionale alla banda disponibile. A differenza di Packet Pair, gli autori di Cprobe non hanno ipotizzato l'impiego della disciplina FQ sui nodi intermedi; in tempi più recenti tuttavia è stato dimostrato [50] come, senza questa ipotesi, la misura ottenuta si riferisca ad un valore più propriamente chiamato Asymptotic Dispersion Rate (ADR), sempre legato alla banda disponibile ma non coincidente con essa.

Coefficiente β

Paxson [48] ha definito il coefficiente β , una misura relativa dell'utilizzo di un cammino di rete. La definizione di questa metrica è basata sulla variazione del ritardo sperimentato in una direzione da un flusso di pacchetti: β riflette la percentuale del ritardo dei pacchetti imputabile al carico introdotto in rete dal flusso stesso. Perciò se ogni pacchetto viene accodato direttamente dietro il predecessore, il percorso di rete può essere considerato scarico e $\beta \approx 1$. Al contrario, se le variazioni nel ritardo sono sostanzialmente tutte dovute alla presenza di cross-traffico, il percorso di rete considerato è prossimo alla saturazione e $\beta \approx 0$. Rispetto alla classica concezione di utilizzo, dove valori prossimi allo zero indicano scarso carico e viceversa, β ha perciò un significato opposto. Il coefficiente è ricavato ipotizzando che il *tight link* ed il link con capacità inferiore coincidano, evento non sempre verificato. A questa ipotesi limitativa si aggiunge il problema che, anche supponendo che ciò sia vero, non è chiaramente definibile la relazione tra β e la quantità effettiva di banda disponibile.

5.2.2 I modelli di gap probing (PGM)

Il PGM sfrutta le informazioni sulla distanza dei tempi che separa l'arrivo sul destinatario di due pacchetti di probe. Una coppia di questi viene infatti inviata con una distanza interna Δ_{in} e, una volta giunta a destinazione, presenta un gap temporale tra i due pacchetti pari a Δ_{out} . Ipotizzando che vi sia una singola strozzatura e che la coda associata non venga svuotata nel tempo che intercorre tra l'arrivo del primo e del secondo pacchetto, allora Δ_{out} corrisponde al tempo impiegato dal link con banda disponibile minore per trasmettere il secondo probe packet ed il traffico ricevuto nell'intervallo Δ_{in} .

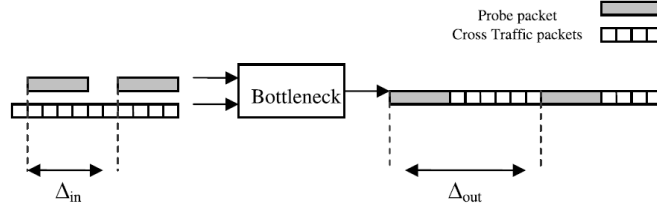


Figura 5.3: Il principio alla base dei modelli PGM (da [12])

Pertanto il tempo impiegato per trasmettere il cross-traffico è $\Delta_{out} - \Delta_{in}$ e $(\Delta_{out} - \Delta_{in}) \cdot C$ il relativo rate, dove C indica la capacità del link strozzante, supposta nota. La banda disponibile è pari a:

$$A = C \left(1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}} \right)$$

Delphi, IGI, Spruce e BART sono strumenti che si basano su questo principio.

Delphi

Delphi [51] è un metodo che adotta un approccio interessante nel problema della stima, utilizzando un modello statistico di descrizione del cross-traffico per calcolare la banda disponibile. Utilizzando un modello multifrattale wavelet (MWM) e tecniche di stima Bayesiane, il tool è in grado di descrivere le caratteristiche del traffico su diverse scale temporali, nonché il comportamento delle code, effettuando misurazioni poco invasive. In particolare Delphi introduce il concetto di *chirp*, un treno di pacchetti distanziati tra loro da un tempo esponenzialmente crescente; questa idea, seppur utilizzata in tutt'altro modo, è stata ripresa successivamente da un altro software (Cfr. 5.2.3). Delphi ipotizza però che gli accodamenti avvengano solamente sul canale limitante, che deve essere contemporaneamente *tight* e *narrow link*. Inoltre, per valutare la banda disponibile, la capacità di quest'ultimo deve essere nota a priori¹.

IGI

Initial Gap Increasing (IGI) è un metodo di stima ibrido proposto da Hu e Steenkiste [52] [53], in grado di ricavare la stima in maniera diretta inviando treni di pacchetti ma determinando il rate di trasmissione in maniera iterativa, come accade con TOPP o PTR. Nonostante gli incoraggianti risultati ottenuti con IGI nei test in laboratorio, prove sul campo effettuate in un secondo momento hanno però fornito risultati poco affidabili [12].

Spruce

Spread Pair Unused Capacity Estimate (Spruce) [12] è un altro tool basato sul Probe Gap Model e quindi, analogamente a IGI e Delphi, presuppone l'esistenza di un unico link strozzante nel cammino di rete. Anziché utilizzare treni, per ricavare i valori della

¹O stimata in anticipo, il che introduce però un'ulteriore complicazione

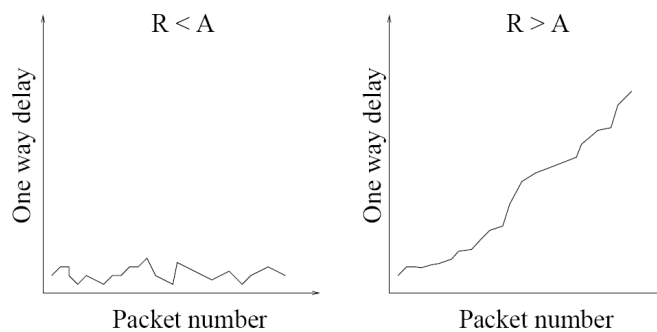


Figura 5.4: Il principio alla base dei modelli PRM (da [31])

banda disponibile Spruce invia 100 coppie di pacchetti ad un rate pari alla capacità del *tight* link (che si suppone nota). Inoltre, per simulare campionamenti poissoniani, le coppie di pacchetti sono distanziate da tempi di interarrivo esponenziali. Alcuni test, condotti dagli stessi autori di Spruce, hanno mostrato come questo approccio fornisca misurazioni più affidabili e sia meno invasivo dei due strumenti precedentemente citati.

5.2.3 I modelli di rate probing (PRM)

I modelli PRM sono basati sul concetto di congestione autoindotta². L'idea di fondo si basa su alcune evidenze empiriche, piuttosto che sulla costruzione di modelli statistici di cross-traffico. Si suppone infatti che l'immissione in rete di traffico di prova con un rate R , superiore alla banda disponibile A del percorso, provochi accodamenti lungo i router intermedi, fornendo così tempi di trasferimento crescenti. Viceversa, come visibile in Figura 5.4, inviando un certo numero di pacchetti ad una velocità $R < A$, non verranno sperimentati ritardi crescenti.

Software quali TOPP, Pathload/SLoPS, PTR e Pathchirp utilizzano il modello di rate probing.

Trains of Packet Pairs

Train of Packet Pairs (TOPP) [54] è l'esempio classico di misurazione iterativa; uno stream, costituito da più coppie di pacchetti e caratterizzato da un certo rate, viene spedito verso il destinatario. Basato sul principio della congestione autoindotta, questa tecnica incrementa linearmente il rate per le trasmissioni successive, individuando così il rate massimo di invio oltre il quale si manifestano fenomeni di congestione. In aggiunta a ciò, TOPP è anche in grado di stimare la capacità del *tight link*, che può essere superiore a quella del cammino di rete considerato. DietTOPP [55] utilizza una versione ridotta dell'algoritmo alla base di TOPP, implementando un algoritmo di ricerca modificato per l'impiego nelle reti wireless.

²Autoindotta poiché è il software di misura che volutamente immette, per un breve periodo di tempo, un traffico ad una velocità maggiore di quanto non possa essere gestito dalla rete stessa senza accodamenti

Pathload

Pathload [56] [57] un software piuttosto noto che implementa la metodologia di misura Self-Loading of Periodic Streams (SLoPS). SLoPS prevede l'invio di flussi periodici, composti da ≈ 100 pacchetti aventi dimensioni uguali, ad un determinato rate R . Così come prevede TOPP, il valore R di spedizione viene adattato in base alla variazione dell'One Way Delay dei pacchetti di *probe*. Le differenze tra i due metodi sono infatti principalmente a livello di analisi statistica delle misurazioni; inoltre, mentre TOPP incrementa il rate linearmente, SLoPS utilizza una ricerca binaria per convergere ad un valore di R il più possibile vicino alla banda disponibile. Poiché la disponibilità di banda stimata può variare durante la misurazione, il risultato potrebbe non mostrare chiari trend crescenti o stabili; in questi casi Pathload fornisce un'area di incertezza (*grey region*) corrispondente all'intervallo di variazione entro cui è compresa la banda disponibile.

PTR

Packet Transmission Rate (PTR) [53] è una tecnica proposta dagli stessi autori di IGI ed è molto simile a TOPP, poiché basata su misurazioni iterative con variazione lineare del rate di invio e convergenza della stima ad unico valore. A differenza di TOPP, vengono però utilizzati treni di 60 pacchetti anziché coppie.

pathChirp

pathChirp [58] è un ulteriore software basato sul principio di autocongestione. Anziché inviare trasmissioni periodiche di pacchetti a rate costante, pathChirp utilizza però dei treni costituiti da pacchetti con distanze esponenzialmente decrescenti detti *chirp*³. Questo approccio consente di sperimentare diversi rate inviando un numero di pacchetti estremamente contenuto, riuscendo inoltre a fornire un risultato con un'unica trasmissione. Come illustrato in Figura 5.5, la banda disponibile A può essere individuata trovando il pacchetto in coincidenza del quale il traffico di misurazione inizia a sperimentare ritardi crescenti.

BART

Bandwidth Available in Real-Time (BART) è uno degli strumenti più recenti proposti in questo settore [59]; sviluppato congiuntamente da Ericson e gruppi di ricerca universitari svedesi, è utilizzato per esperimenti nell'infrastruttura di rete europea per le misurazioni Etomic⁴. BART si basa sul principio di TOPP ma invia treni di packet pairs a rate casuali e, ad ogni spedizione, effettua una stima della banda disponibile in *real-time* utilizzando un filtro di Kalman [60].

5.3 I problemi nella stima della banda disponibile

I limiti delle misurazioni attive effettuate tramite traffico di *probing*, già citati in 4.1.2 non sono tuttavia gli unici problemi che devono essere affrontati. Esistono

³L'idea è stata originariamente proposta in Delphi (Cfr. 5.2.2)

⁴Etomic (<http://www.etomic.org>) è parte del progetto di ricerca europeo Evergrow

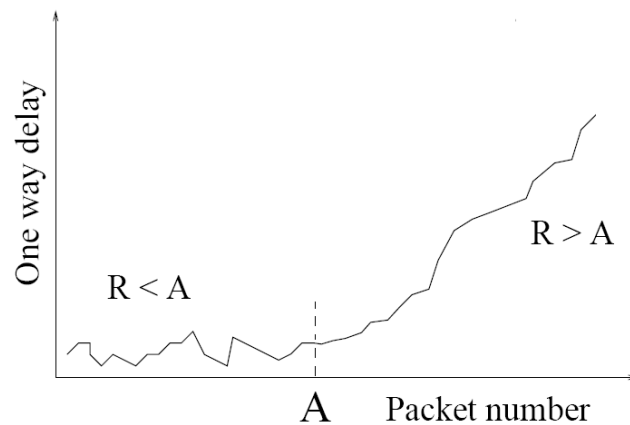


Figura 5.5: pathChirp individua la banda disponibile A inviando pacchetti sempre più ravvicinati

infatti numerosi fattori che possono influenzare in maniera significativa l'accuratezza delle rilevazioni e che devono quindi essere considerati in fase di progettazione di algoritmi e di analisi dei risultati ottenuti [61] [3]. Le ipotesi alla base dei modelli di stima non permettono infatti di descrivere esattamente le caratteristiche del traffico, i meccanismi interni dei dispositivi di commutazione o la topologia di rete. Inoltre le tempistiche software, spesso considerate trascurabili, possono in realtà essere fonte di errori difficilmente evitabili. A tutto questo si aggiungono le difficoltà nel validare le stime ottenute: le simulazioni di laboratorio, per quanto precise, non consentono di rappresentare adeguatamente tutti gli aspetti della realtà mentre le prove “sul campo” non permettono di verificare con certezza i risultati.

5.3.1 I dispositivi invisibili

Determinare la topologia di rete attraversata dal traffico di prova è estremamente complesso. In particolar modo non è possibile rilevare con certezza l'esistenza di tutti i dispositivi di commutazione, specialmente se questi operano al di sotto del livello IP. Ciò rappresenta un problema dal momento che tali apparecchiature, pur operando in maniera trasparente, effettuano operazioni di commutazione in modalità *store-and-forward* e dispongono solitamente di code di attesa. Funzionando quindi in maniera analoga ai router, dispositivi quali gli *switch* Ethernet possono alterare la spaziatrice e la distribuzione dei pacchetti di prova in maniera imprevedibile, inficiando la validità delle misurazioni.

5.3.2 La gestione delle code

Il diverso comportamento dei router intermedi può violare l'ipotesi di gestione di tutte le code con disciplina FCFS, assunta da quasi tutti i modelli di stima considerati. Per esempio diversi router hanno recentemente iniziato ad implementare meccanismi di Active Queue Management (AQM) quali l'algoritmo Random Early Detection [11]. A differenza di FCFS, queste nuove discipline hanno lo scopo di prevenire la congestione, scartando trasmissioni di pacchetti che potrebbero condurre il nodo ad

un sovraccarico. Per questo motivo i meccanismi basati sulla congestione autoindotta potrebbero non riuscire mai a saturare il link con minor disponibilità di banda, fornendo così misurazioni imprecise.

5.3.3 La gestione del tempo

Occorre poi valutare attentamente i problemi legati alla risoluzione temporale. Tutti i metodi di stima presentati si basano infatti sul calcolo delle variazioni nella distanza temporale tra i pacchetti ricevuti; indipendentemente dal metodo adottato (tempo assoluto o relativo di interarrivo), è quindi necessario interagire con un *clock*. Sebbene esistano oggi circuiti in grado di generare temporizzazioni stabili e con un'elevata risoluzione, generalmente la lettura avviene attraverso *routine* software. I tool di misurazione, eventualmente integrati con i servizi applicativi, devono quindi attendere che lo *stack* di rete, gestito dal sistema operativo, catturi la segnalazione dell'hardware e quindi elabori i dati nel *buffer*. Sfortunatamente, dall'istante in cui il pacchetto viene ricevuto dall'interfaccia al momento in cui viene passato al software di misurazione, intercorre un certo periodo di tempo (solitamente non deterministico) durante il quale il sistema può dover gestire operazioni a più alta priorità come rischedulazioni, svuotamento di *buffer* o interruzioni *hard*. Inoltre diverse schede di rete cercano di ottimizzare l'interazione con il sistema operativo, attendendo un certo intervallo di tempo prima di generare una richiesta di interrupt per il pacchetto ricevuto. Questo permette di servire messaggi ravvicinati con una sola richiesta, riducendo il numero di *context switch* effettuati. D'altro canto i pacchetti vengono consegnati al sistema operativo in contemporanea, annullando così la differenza esistente tra gli effettivi istanti di arrivo⁵ [32]. La sensibilità a questi errori e ai ritardi nelle tempistiche aumenta al crescere della effettiva quantità di banda disponibile. Per rendere l'idea, l'invio di un pacchetto in una rete locale FastEthernet richiede $\approx 100\mu s$: arrotondamenti nell'ordine della decina dei μs possono quindi condurre ad imprecisioni del 10% nella stima.

Proprio per risolvere questi problemi alcuni progetti [59] hanno utilizzato hardware apposito, in grado di effettuare una marcatura temporale dei pacchetti a livello di Network Interface Card (NIC), eventualmente sfruttando Network Time Protocol (NTP) o un segnale GPS per la sincronizzazione dei diversi dispositivi presenti in rete. In altri casi [25] è stato invece utilizzato un sistema operativo *real time* quale RT-Linux. Entrambe le soluzioni, pur fornendo risultati migliori, richiedono hardware o software dedicato: questo ha quindi il grosso svantaggio di limitare l'effettivo impiego su larga scala degli strumenti di misura proposti.

5.3.4 La descrizione del traffico

I modelli di stima della banda disponibile ipotizzano la presenza in rete di cross-traffico ideale; questa è una forte semplificazione che trascura anche la semplice interazione tra i pacchetti, nella realtà entità discrete e non infinitesime. In molte simulazioni di laboratorio viene sfruttata una sorgente di traffico a bit rate costante (CBR): per quanto semplice, essa è in grado di rappresentare con fedeltà alcune applicazioni di rete esistenti ed inoltre è la miglior approssimazione del modello di traffico

⁵Il problema è indicato in letteratura con il termine *Interrupt Coalescence*

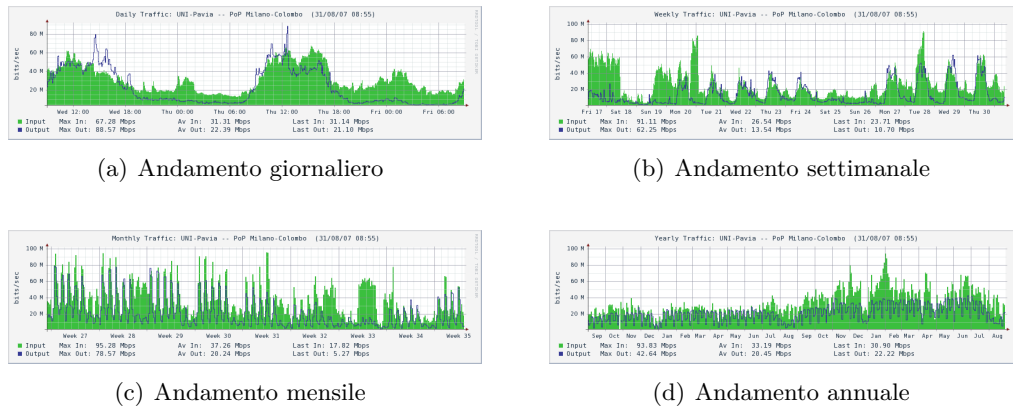


Figura 5.6: Stratifiche del traffico medio in transito rilevate su diverse scale temporali (Fonte UniPV - Rete GARR)

fluido. Tuttavia, considerando più flussi in contemporanea, i dati che transitano in una rete mostrano una distribuzione temporale che può essere modellizzata con una sorgente poissoniana o persino utilizzando modelli caotici [62]. Se questo è vero per osservazioni aventi durata nell'ordine dei centesimi di secondo, la situazione è però diversa per scale temporali più ampie. Diversi studi [63] [64] [65] hanno mostrato infatti come il traffico in Internet presenti forti correlazioni, dette di dipendenza sul lungo periodo (Long Range Dependence): misurazioni successive, distanti tra loro da pochi minuti ad un'ora, hanno infatti spesso fornito valori coincidenti in media. Come evidente in Figura 5.6, il traffico sui router presenta anche forti caratteristiche periodiche e di auto-somiglianza su differenti scale temporali [66]; a questo scopo sono quindi stati proposti anche modelli di traffico frattali basati su Wavelet [51].

La scelta del tipo di sorgente di traffico da utilizzare nelle simulazioni in laboratorio è quindi fondamentale per poter validare lo strumento. A seconda del modello adottato, è possibile infatti verificare il funzionamento degli strumenti di valutazione della banda in presenza di un sottoinsieme delle proprietà caratteristiche di Internet. Non è invece facilmente simulabile l'andamento reale del traffico in rete sotto ogni aspetto dal momento che, come ricordato poca sopra, non ne esiste nemmeno una descrizione univoca. Per contro, nemmeno l'utilizzo di tracciati reali è esente da problemi: non vi è infatti alcuna garanzia che la singola "istanza" di traffico considerata sia adeguatamente rappresentativa della realtà.

5.3.5 Le tempistiche di misurazione

Legato agli aspetti di caratterizzazione del traffico vi è anche il problema delle tempistiche di invio del traffico. La dimensione ed il numero di pacchetti spediti per volta, nonché il numero di misurazioni che occorre effettuare per poter convergere al valore della stima, sono informazioni essenziali. Variando la durata della trasmissione di misurazione è possibile, ad esempio, ridurre l'impatto di *burst* di traffico. Analogamente un controllo sulla banda occupata dal traffico di prova è esercitabile distanziando adeguatamente nel tempo le diverse misurazioni. Se da un lato è importante contenere la quantità di traffico immesso in rete, pena rendere eccessivamente

invasivo il programma, d'altro canto occorre stabilire un tempo massimo tra due misurazioni successive. Infatti, data la complessità e la variabilità del traffico in rete, misurazioni eccessivamente distanziate non riuscirebbero a “seguire” adeguatamente la dinamica della banda disponibile, fornendo così valori di scarso interesse.

5.3.6 La verifica delle misurazioni

Sebbene l'impiego degli strumenti in applicazioni reali sia fondamentale, risulta estremamente difficile valutare con precisione i risultati ottenuti. In Internet non è infatti possibile disporre di un controllo completo sui router intermedi, utilizzando ad esempio il protocollo Simple Network Management Protocol (SNMP): come metro di confronto vengono spesso utilizzati i risultati ottenuti con stime della Bulk Transfer Capacity. Tuttavia le due metriche sono differenti e non è lecito supporre che coincidano: come già ricordato, i valori relativi ad un trasferimento TCP dipendono da numerosi fattori quali la dimensione dei *buffer* sui nodi intermedi, il tasso di perdita dei canali e persino la tipologia di cross-traffico presente in rete. Uno studio dettagliato sulla relazione tra la banda disponibile ed il *throughput* di una connessione TCP è disponibile in [67] ma non è stata ancora identificata una formula univoca.

Capitolo 6

L'ambiente di prova

Parte della disumanità del computer sta nel fatto che, una volta programmato e messo in funzione, si comporta in maniera perfettamente onesta.

Isaac Asimov

Per il lavoro di ricerca in laboratorio si è preferito evitare l'utilizzo di simulatori di rete ritenendo che questi strumenti, per quanto utili in moltissime situazioni, non siano in grado di rappresentare con sufficiente realismo tutte le componenti rumorose ed i comportamenti imprevedibili del traffico e dei dispositivi di rete. Al contrario, per poter individuare e sperimentare nuovi approcci nella stima della banda disponibile è stato implementato in Laboratorio Reti dell'Università degli Studi di Pavia un *testbed* con calcolatori e dispositivi effettivamente presenti in reti reali. Inoltre si è sfruttato il lavoro svolto in passato dagli autori del software pathChirp come punto di partenza per le ricerche. Sviluppato nel 2003 da ricercatori della Rice University e di Stanford [58], ad oggi pathChirp¹ rimane uno dei migliori strumenti sviluppati nell'ambito di progetti di ricerca e liberamente disponibile: i test comparativi effettuati in passato da numerosi ricercatori hanno confermato la bontà del metodo proposto, ottenendo risultati discreti anche ove altri meccanismi hanno fallito. La sorgente di cross-traffico presente in rete è stata invece simulata utilizzando Iperf [68], uno strumento che consente di inviare un semplice flusso di dati controllato e riproducibile.

6.1 La rete

La rete utilizzata per i test è stata realizzata utilizzando sei computer Compaq PentiumIV aventi la medesima configurazione hardware e forniti di interfacce di rete FastEthernet Realtek 8139. Su tutte le macchine è stato poi installato Debian GNU/Linux 4.0 (kernel 2.6), scelta peraltro obbligata sugli host per poter utilizzare i software di misura. Onde evitare poi eventuali conflitti tra gli *stack* di rete di sistemi operativi differenti, è stato utilizzato lo stesso ambiente anche sui router. Al primo router sono stati connessi la sorgente di traffico controllato ed uno degli *end-point*

¹La versione più recente, utilizzata anche per le prove in laboratorio, è la 2.4.1

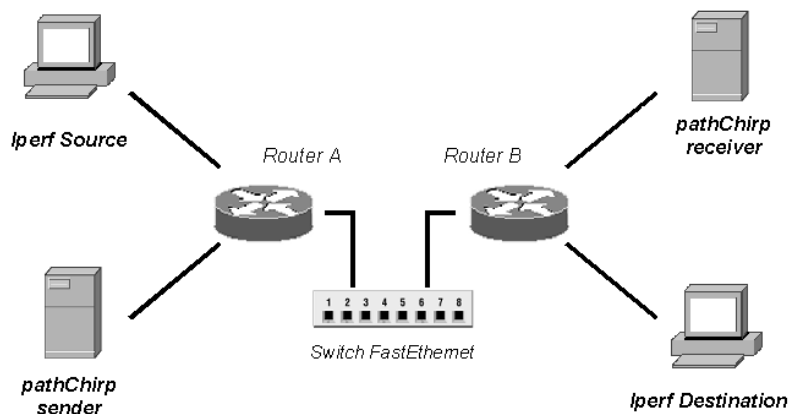


Figura 6.1: Schema della rete di prova utilizzata in laboratorio

coinvolti nella misurazione. Discorso analogo per il secondo dispositivo, connesso ai due segmenti di rete ospitanti l'altro nodo incaricato di stimare la banda disponibile ed il destinatario del cross-traffico. I due router sono stati poi interconnessi attraverso uno Switch Ethernet HP a 100 *Mbps*: questo dispositivo è stato utilizzato per inserire un'ulteriore fonte di potenziale rumore durante le misurazioni.

La relativa semplicità della rete è motivata dalla necessità di poter disporre di un elevato controllo sul sistema di prova. Proprio per questa ragione non sono stati effettuati test attraverso Internet: non potendo conoscere l'esatta banda disponibile in ogni istante, sarebbe stato impossibile valutare con precisione l'accuratezza delle misure.

6.2 Pathchirp

Basato sul concetto di congestione autoindotta già presentato in precedenza (Cfr. 5.2.3), pathChirp modella il cammino di una rete di comunicazione come una sequenza di nodi *store-and-forward*, ciascuno dei quali dotato di code FCFS e caratterizzato da un tasso di servizio μ costante. Nonostante la semplicità del modello, esso risulta comunque in grado di descrivere con sufficiente approssimazione il comportamento di Internet. Lo schema utilizzato da pathChirp per determinare la banda disponibile è innovativo e si basa sull'invio di sequenze, dette *chirp*, di pacchetti distribuiti esponenzialmente. Questo *pattern* risulta estremamente efficiente, dal momento che consente di inviare un minor numero di dati se confrontato con altri algoritmi. Infatti i metodi derivati da Packet Pairs devono inviare $2N$ pacchetti (cioè N coppie) per poter studiare gli effetti su N diverse spazature; con un *chirp* è possibile invece ottenere risultati analoghi spedendo solo $N + 1$ messaggi. Inoltre, decrementando esponenzialmente il tempo che separa due pacchetti successivi, la quantità di traffico inviato cresce meno rapidamente del range della banda oggetto di indagine². Un ulteriore vantaggio di tutti gli strumenti che inviano treni di pacchetti, anziché coppie, è la possibilità di sfruttare la correlazione esistente tra i ritardi

²Come ricavato successivamente in 7.1.1, per un spettro di possibili velocità $[G_1, G_2]$ il numero di pacchetti cresce proporzionalmente alla differenza $\log G_2 - \log G_1$

sperimentati dai vari pacchetti. PathChirp consente perciò di ottenere una stima della banda disponibile in maniera rapida, relativamente affidabile e senza risultare eccessivamente invasivo.

6.2.1 La notazione

Prima di procedere nella descrizione dell'algoritmo è opportuno introdurre la notazione che verrà utilizzata nel resto della trattazione. Detto m un generico *chirp*, costituito da N pacchetti di P byte e distanziati esponenzialmente, si definisce γ il fattore di spaziatura temporale tra essi. Il k -esimo pacchetto del chirp m , inviato al tempo t_k , sperimenta lungo il percorso un tempo di accodamento q_m ; inoltre la distanza tra il pacchetto k ed il successivo è detta Δ_k e, pertanto, il rate di trasmissione istantaneo per il k -esimo pacchetto è

$$R_k = \frac{P}{\Delta_k}$$

Ipotizzando uno scenario con cross-traffico fluido ideale e a bit rate costante (CBR), si ha che

$$\begin{aligned} q_k &= 0, & \text{se } B[t_1, t] \geq R_k \\ q_k &> q_{k-1}, & \text{altrimenti} \end{aligned}$$

Da questo si può ricavare facilmente la stima della banda media disponibile nell'intervallo di tempo occupato dalla misurazione come

$$A_{avg}[t_1, t_N] = R_k$$

corrispondente cioè alla velocità di trasmissione del pacchetto k^* , in corrispondenza del quale i ritardi di accodamento iniziano a mostrare un trend crescente. L'ipotesi di cross-traffico fluido CBR per l'intera durata della misurazione è ovviamente una forte semplificazione della realtà che, per quanto efficace, non contempla la possibilità di sporadici picchi istantanei di traffico (*burst*) e l'esistenza di interazioni tra i pacchetti. Per questo motivo gli autori di pathChirp hanno introdotto alcuni accorgimenti, riuscendo a gestire in maniera efficace le interferenze di uno o più *burst* con il treno di pacchetti di prova.

Il grafico in Figura 6.2 rappresenta il profilo di un chirp (*signature*), cioè l'andamento caratteristico dei tempi di accodamento rilevabili per una misurazione. Tipicamente è possibile individuare diversi spostamenti (più propriamente detti escursioni) rispetto all'asse zero, in corrispondenza di ritardi imputabili a cross-traffico temporaneo. In questi casi, poiché il rate istantaneo R_k non è comunque superiore alla capacità del link strozzante, all'esaurirsi del picco di traffico le code riescono ad accorciarsi e la *signature* ritorna a zero. L'ultima escursione invece, dal momento che ha un rate R_k maggiore di quello massimo supportato dal percorso di rete attraversato, è la causa stessa dell'aumento della spaziatura tra i pacchetti. PathChirp utilizza proprio il profilo per calcolare una stima E_k della banda disponibile per ogni coppia di pacchetti consecutivi. Successivamente viene trovato il surplus di banda *per-chirp* $D(m)$ attraverso una media pesata dei vari valori E_k ed infine fornita una stima, ottenuta calcolando una media delle varie misure D ottenute in una finestra temporale ampia τ .

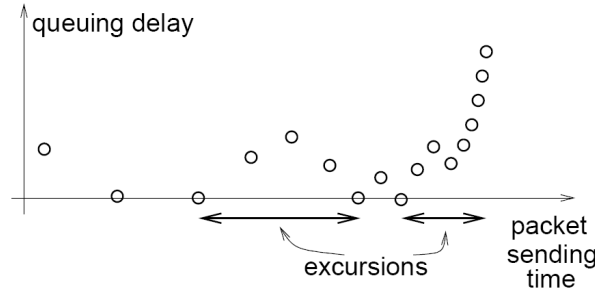


Figura 6.2: Tipico andamento dei ritardi in un chirp (da [58])

6.2.2 L'algoritmo di stima

Per poter calcolare con precisione E_k , pathChirp ha introdotto un algoritmo di segmentazione binaria del profilo in grado di distinguere le sequenze corrispondenti ad una escursione da quelle con un profilo costante. Dal momento che il principio su cui esso si basa è quello della congestione autoindotta, vale il discorso precedentemente affrontato: ritardi crescenti dovuti ad accodamenti indicano che la banda disponibile è inferiore al rate istantaneo del chirp, così come vale anche il contrario.

$$E_k \geq R_k, \quad \text{se } q_k \geq q_{k+1} \quad (6.1)$$

$$E_k \leq R_k, \quad \text{altrimenti} \quad (6.2)$$

La 6.2 vale solo nel caso in cui i pacchetti k e $k + 1$ possano effettivamente condurre ad uno stato di temporanea congestione³ un nodo, ovvero che siano sufficientemente ravvicinati. Pertanto, affinché la 6.2 sia verificata, occorre segmentare ogni profilo nelle regioni di escursione ed applicare la formula solamente ad esse. Intuitivamente, se q_k aumenta e rimane al di sopra dello zero per diversi pacchetti consecutivi, è allora probabile che questi appartengano al medesimo *busy period*. Nello specifico, l'algoritmo di segmentazione implementato da pathChirp opera nel seguente modo per riuscire a determinare l' i -esimo ed il j -esimo pacchetto di m , rispettivamente corrispondenti al primo e all'ultimo di un'escursione:

1. Il primo pacchetto i della sequenza m per cui vale $q_i < q_{i+1}$ è un potenziale inizio per l'escursione
2. Il punto j di fine dell'escursione è definito come il primo pacchetto per cui vale

$$q(j) - q(i) < \frac{\max_{i \leq k \leq j} [q(k) - q(i)]}{F}$$

dove F è un parametro detto fattore di discesa. In altre parole, in j la spaziatura dovuta agli accodamenti relativa a $q(i)$ è diminuita di un fattore F rispetto al massimo ritardo sperimentato nell'intervallo $[i, j]$.

3. Se la regione $j - i$ è sufficientemente lunga, ad esempio maggiore di un certo valore L , allora la regione individuata corrisponde ad un'escursione e non ad un errore isolato.

³Questa situazione viene detta anche *busy period* poiché la coda, per il periodo di tempo considerato, non è mai vuota e il servente è sempre occupato

Poiché l'ultima escursione è dovuta ad una congestione non-temporanea, esisterà solitamente un pacchetto l oltre il quale il profilo non tornerà mai a 0; questa escursione verrà considerata diversamente nel calcolo di E_k , come spiegato di seguito. L'algoritmo calcola inizialmente E_k , cioè la stima della banda relativa a ciascun pacchetto k ; questi valori sono quindi utilizzati, per la valutazione della disponibilità di banda D relativa all'intero chirp m , secondo tre modalità:

Caso a) Se k appartiene alla parte crescente, cioè $q_k \leq q_{k+1}$, di un'escursione che torna a zero, allora vale

$$E_k = R_k$$

Caso b) Per tutti i pacchetti k appartenenti invece all'escursione senza termine, si pone:

$$E_k = R_l, \quad \forall k > l$$

dove l corrisponde all'inizio dell'escursione. Questo caso dovrebbe quindi includere tutti i rate superiori alla banda disponibile del sistema: in realtà, nel caso in cui il trend dei ritardi q_k considerati sia decrescente seppur non tornando a 0, si potrebbe ottenere una stima E_k conservativa.

Caso c) Per tutti i k non compresi nei casi precedenti, ovvero quando k non appartiene ad una escursione o comunque nella parte con trend decrescenti della spaziatura, E_k viene posto:

$$E_k = R_l$$

Nel caso in cui l'ultima escursione della *signature* termini, l viene scelto pari a N . Il verificarsi di questa condizione significa che il range $[R_{min}, R_{max}]$ di probe non è sufficientemente ampio da includere il valore corrispondente alla banda disponibile: in questo caso il software provvede ad aumentare e traslare lo spettro delle velocità testate. In Appendice C.1 è stato riportato lo pseudo-codice relativo all'algoritmo sopra descritto ed implementato originariamente in pathChirp.

6.2.3 Il confronto con gli altri metodi esistenti

Test condotti dagli autori di questosoftware e da numerosi altri ricercatori hanno confrontato pathChirp con diversi strumenti ed in molteplici contesti. Dai risultati ottenuti è possibile dedurre che l'approccio adottato da questo strumento è di gran lunga superiore a TOPP e a Pathload/SLoPS sotto ogni punto di vista; l'unico strumento che ha ottenuto risultati comparabili in termini di accuratezza è IGI, in particolar modo se utilizzato in link caratterizzati da scarsa presenza di traffico.

I dati presentati in Tabella 6.1 richiedono particolare attenzione per poter essere correttamente interpretati: il valore medio di banda occupata non può infatti venir considerato indipendentemente dal tempo richiesto dallo strumento per convergere alla misura. Si è perciò pensato di rappresentare i dati in un grafico 2D, utilizzando come dimensioni il tempo di misurazione e la quantità di traffico necessario per fornire un risultato.

Dal grafico in Figura 6.3 appare evidente quindi come pathChirp e IGI siano preferibili agli altri strumenti citati e, in particolare, come il primo sia in grado di fornire misurazioni in tempi estremamente ridotti introducendo comunque quantità di traffico accettabili.

Parametro	TOPP	SLoPS	PathChirp	IGI
Load (<i>KB</i>)	977	1079	127	84
Probing time (<i>s</i>)	19,96	3,96	0,76	2.03
Average rate (<i>Mbps</i>)	0,40	2,23	1,37	0.34

Tabella 6.1: Parametri di efficienza dei diversi strumenti di stima della banda (dati ricavati con $Pktsize = 500$ *byte*, fonte [69])

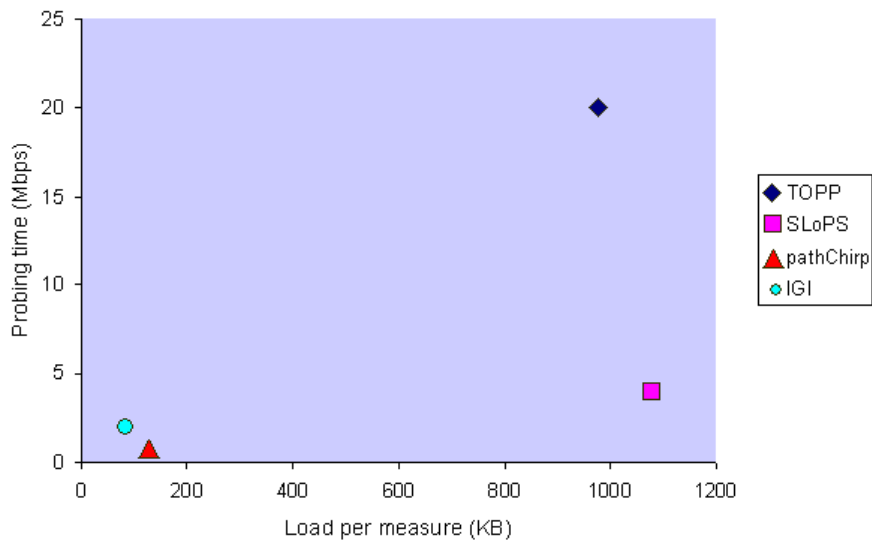


Figura 6.3: Confronto tra i più noti strumenti di stima

Variabile	Opzione	Parametro rappresentato	Valore
P	-p	dimensione dei pacchetti	1000 <i>byte</i>
γ	-s	fattore di spaziatura	1,2
F	-d	fattore di discesa	1,5
B	-b	durata minima dell'escursione	5
T_{probe}	-t	durata dell'esperimento	120 <i>s</i>
n	-n	dimensione finestra <i>MA</i>	1
B_{avg}	-a	banda media utilizzata	0,3 <i>Mbps</i>
B_{min}	-l	rate minimo del chirp	10 <i>Mbps</i>
B_{max}	-u	rate massimo del chirp	200 <i>Mbps</i>

Tabella 6.2: Parametri di configurazione di pathChirp 2.4.1 utilizzati in laboratorio.

6.2.4 Configurazione utilizzata per i test

Per effettuare la stima pathChirp invia un treno di pacchetti UDP aventi dimensione P e contenenti un *timestamp* inserito dal mittente in fase di spedizione. Poiché non è sempre possibile ipotizzare che gli orologi presenti sul mittente e sul destinatario siano sincronizzati, il ritardo sperimentato da ciascun pacchetto viene calcolato utilizzando come riferimento l'istante di ricezione del primo. Inoltre, poiché l'effettiva durata ed entità degli accodamenti può variare notevolmente da link a link, non viene utilizzato un livello prefissato per discriminare le varie sezioni del profilo. I parametri di pathChirp impostabili sono perciò numerosi: il fattore di dispersione temporale γ , il fattore di discesa F ed la soglia L per l'individuazione di un'escursione provocata da un busy period. Analogamente, è possibile impostare il range di velocità istantanee associate ad un chirp o l'occupazione media di banda associata al traffico di prova.

PathChirp affronta inoltre gli effetti indesiderati di comportamenti anomali del sistema operativo o delle interfacce di rete. Ritardi nella gestione del traffico di rete da parte del sistema operativo potrebbero infatti alterare la spaziatura effettiva tra i diversi pacchetti, causando pesanti errori nella stima della disponibilità di banda. Per questo motivo il programma scarta tutti i treni contenenti pacchetti con tempi di interarrivo inferiori ad una certa soglia d . Il valore d individuato dagli autori del software è pari a $30\mu s$; attualmente questo dato è modificabile solo intervenendo sui sorgenti di pathChirp e ricompilando l'applicativo. Per la determinazione dei parametri di configurazione utilizzati nelle prove si è fatto riferimento al lavoro degli autori, che hanno effettuato misurazioni in diverse condizioni per determinare le migliori combinazioni di assegnazione dei valori. Le impostazioni utilizzate, riportate in Tabella 6.2, sono in buona parte le stesse suggerite di *default* dal software.

Occorre tuttavia precisare, sebbene i risultati ottenuti in laboratorio siano soddisfacenti, quella operata è indubbiamente una scelta sub-ottima: per l'individuazione dei parametri ottimali si dovrebbero infatti tenere in considerazione le caratteristiche statistiche del cross-traffico e delle code intermedie nonché ipotizzare che queste non varino in alcun modo durante le misurazioni.

6.3 Iperf

Come ricordato in precedenza, la possibilità di poter controllare la quantità ed il tipo di cross-traffico immesso in rete permette di valutare con precisione i risultati ottenuti e garantisce la riproducibilità degli stessi, componenti aleatorie permettendo. Esistono a tale scopo numerosi strumenti in grado di generare diversi tipi di traffico: tra questi vanno sicuramente citati Distributed Internet Traffic Generator (D-ITG), sviluppato in Italia, il Poisson-traffic generator della Rice University e Iperf. Quest'ultimo è un applicativo sviluppato dall'Università dell'Illinois in grado di inviare un flusso UDP ad un bit rate costante X ed impostabile dall'utente, fornendo poi dettagliate informazioni sulla capacità, il *jitter* ed il tasso di perdita dei pacchetti spediti. Per le prove effettuate in laboratorio la scelta è caduta proprio su questo strumento, sebbene il flusso Constant Bit Rate (CBR) generato sia il più semplice tipo di traffico non ideale esistente. Occorre infatti considerare che qualsiasi software per la stima della banda deve essere in grado di fornire misurazioni accurate in presenza di cross-traffico costante prima di poter essere utilizzato in reti più complesse. Inoltre, per quanto semplice, questo tipo di flusso di pacchetti è assimilabile ad una comunicazione VoIP, alla trasmissione di una radio on-line o ad un trasferimento a regime di un file di dimensioni considerevoli. Inoltre il comportamento di pathChirp in presenza di varie tipologie di cross-traffico è già stato oggetto di numerose ricerche [70] [71] [69], rendendo perciò meno interessante uno studio di questo tipo. Al contrario, utilizzando Iperf è stato possibile simulare lunghi periodi di traffico a rate costante separati però da brusche variazioni, studiando così il comportamento di pathChirp e degli approcci innovativi proposti in queste situazioni.

Capitolo 7

Le prove in laboratorio

If the facts don't fit the theory, change the facts.

Albert Einstein

Per quanto valido e consolidato, l'approccio adottato da pathChirp è tuttavia perfettibile ed alcuni aspetti non sono stati ancora oggi sufficientemente approfonditi. Non è stato infatti possibile rintracciare alcuno studio approfondito sulla invasività del programma. All'inizio del lavoro di tesi si è pertanto cercato di individuare una formula in grado di descrivere esattamente il volume di traffico introdotto durante le misurazioni, verificando poi quanto trovato con prove effettive. Grazie a questa analisi è stato possibile anche affrontare un noto problema di sovrastima che affligge da sempre pathChirp, individuandone la causa. Inoltre l'algoritmo implementato in pathChirp risulta infatti estremamente efficace nel rilevare, attraverso l'analisi del profilo temporale del chirp, sequenze isolate di pacchetti in grado di disturbare la singola misurazione. Tuttavia non viene altrettanto efficacemente affrontato il problema su scale temporali più ampie; il software si limita ad effettuare una media pesata delle stime ottenute con i *chirp* più recenti per valutare la banda disponibile. Nessun accenno viene però fatto al numero di misurazioni da includere nella finestra né a come considerare eventuali misurazioni rumorose. Si è perciò cercato di individuare un nuovo approccio, in grado di effettuare un'analisi dei risultati utilizzando tecniche di filtraggio e di stima mutuati anche da settori apparentemente lontani, quali la finanza; i risultati ottenuti sono stati poi confrontati con altri algoritmi proposti in letteratura. Un ulteriore accorgimento, utilizzabile in generale con qualsiasi strumento di stima della banda, è statopoi suggerito per tutte le applicazioni a carattere multimediale (voce e video) che richiedono stime piuttosto stabili e che sono in grado di adattare il rate di trasmissione a livelli discreti. Un ultimo esperimento ha poi fornito risultati inattesi ma che, ad una più attenta analisi, hanno confermato quanto precedentemente detto circa la difficoltà di descrivere con precisione il comportamento dei nodi intermedi. I diversi metodi infatti, pathChirp incluso, ipotizzano che il tempo di servizio dei router sia costante; tuttavia si è riscontrato che, in talune condizioni, le tempistiche software sono tali da non poter più essere trascurate e, al contrario, introducono un ulteriore ritardo non deterministico. Il problema è di attualità in tutte quelle realtà che hanno in programma di effettuare

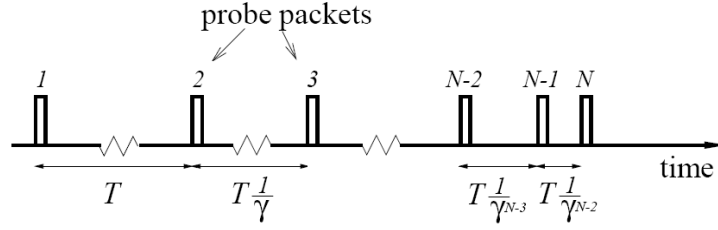


Figura 7.1: Pacchetti in un chirp spaziatamente esponenzialmente

aggiornamenti nell'infrastruttura di rete e reso ancora più interessante dal fatto che è impossibile utilizzare molti degli strumenti di stima della banda oggi disponibili.

7.1 L'invasività di pathChirp

Sebbene diversi studi [69] abbiano misurato sperimentalmente la quantità di traffico immessa in rete da pathChirp, non è stato possibile individuare alcuno studio completo sulla effettiva intrusività e, ancora più importante, sul numero di pacchetti contenuti in un chirp. Ritenendo essenziale il poter disporre di una base teorica adeguata, si è quindi cercato di ricavare formule in grado di descrivere con precisione il traffico di prova utilizzato. Come verrà mostrato in seguito, questa analisi ha peraltro consentito di quantificare con maggior precisione l'errore commesso nella stima e persino risolvere un noto problema presente nel codice di pathChirp.

7.1.1 Il modello teorico

Si consideri la distribuzione caratteristica di un chirp rappresentata in Figura 7.1¹.

Supponendo che la banda disponibile cercata sia compresa tra i valori B_{min} e B_{max} , in base a quanto detto sul funzionamento dell'algoritmo di pathChirp, si può facilmente ricavare che

$$T = \frac{P}{B_{min}} \quad (7.1)$$

$$\frac{T}{\gamma^{N-2}} = \frac{P}{B_{max}} \quad (7.2)$$

Sostituendo la 7.1 nella 7.2, si può ricavare

$$\frac{P}{B_{min} \cdot \gamma^{N-2}} = \frac{P}{B_{max}} \quad \rightarrow \quad \frac{\gamma^2}{B_{min}} B_{max} = \gamma^N$$

Calcolando il logaritmo di entrambi i termini dell'uguaglianza, ne segue

$$\begin{aligned} N \log \gamma &= \log \frac{\gamma^2}{B_{min}} B_{max} \\ &= \log \gamma^2 + \log B_{max} - \log B_{min} \\ &= 2 \log \gamma + \log B_{max} - \log B_{min} \end{aligned}$$

¹ Alcuni parametri sono stati indicati diversamente nell'articolo originale; ai fini della trattazione questo risulta comunque irrilevante.

Il numero N di pacchetti per ogni chirp è quindi pari a

$$N = 2 + \frac{1}{\log \gamma} (\log B_{max} - \log B_{min}) \quad (7.3)$$

o anche, riscrivendo la differenza di logaritmi e approssimando all'intero inferiore

$$N = \left\lfloor 2 + \frac{1}{\log \gamma} \log \left(\frac{B_{max}}{B_{min}} \right) \right\rfloor \quad (7.4)$$

Come facilmente comprensibile, il numero di pacchetti per chirp è direttamente collegato alla quantità di traffico immessa in rete. Al contrario di quel che si potrebbe però ipotizzare, nell'implementazione reale di pathChirp non è il valore N ad imporre le tempistiche di invio e la quantità media di banda occupata. Viceversa, occorre impostare a priori il valore della banda occupata dal traffico di prova; sarà poi l'algoritmo, noto anche il range di valori $[B_{min}, B_{max}]$ tra cui si presume compreso il valore da stimare, a determinare la frequenza di invio dei chirp secondo la formula

$$\frac{N \cdot P}{T_{mis}} = B_{avg} \quad \text{valida per } T_{mis} \gg T_{chirp} \quad (7.5)$$

Può essere utile calcolare anche il tempo T_{chirp} impiegato per la trasmissione dell'intero chirp: tale valore fornisce infatti indicazioni sulla minima distanza temporale tra due misurazioni consecutive. Inoltre, supponendo di conoscere in dettaglio le caratteristiche del cross-traffico, sarebbe possibile individuare la probabilità che un chirp incontri un *burst* durante la misurazione. Detto quindi $T_{chirp}(N)$ il tempo di trasmissione richiesto per un generico chirp di N pacchetti, è facile ricavare che:

$$T_{chirp} = \sum_{i=0}^{N-2} T \left(\frac{1}{\gamma} \right)^i \quad \text{con } \gamma > 1 \quad (7.6)$$

La somma dei primi $N - 2$ termini può essere scritta come

$$S_{N-2} = T + T \frac{1}{\gamma} + \dots + T \frac{1}{\gamma^{N-3}} + T \frac{1}{\gamma^{N-2}} \quad (7.7)$$

e, moltiplicando il tutto per $\frac{1}{\gamma}$, si trova anche che

$$S_{N-2} \cdot \frac{1}{\gamma} = T \frac{1}{\gamma} + T \frac{1}{\gamma^2} + \dots + T \frac{1}{\gamma^{N-2}} + T \frac{1}{\gamma^{N-1}} \quad (7.8)$$

sottraendo la 7.8 alla 7.7 quasi tutti i termini a destra si semplificano e, con pochi passaggi, si ottiene

$$\left(1 + \frac{1}{\gamma} \right) S_{N-2} = T + T \frac{1}{\gamma} - T \frac{1}{\gamma} + \dots + T \frac{1}{\gamma^{N-2}} - T \frac{1}{\gamma^{N-2}} - T \frac{1}{\gamma^{N-1}} \quad (7.9)$$

$$= T - T \frac{1}{\gamma^{N-1}} \quad (7.10)$$

Quindi, riscrivendo la 7.10, è immediato trovare il T_{chirp} ²:

$$T_{chirp} = S_{N-2} \quad (7.11)$$

²Peraltro massimizzabile con $N \cdot T$, un valore assai più semplice da calcolare.

B_{min}	B_{min}	γ	N teorico	N effettivo
10	200	1,2	18,43	18
50	120	1,2	6,8	6
50	120	1,1	9,19	9

Tabella 7.1: Dimensioni dei *chirp* ottenute con diverse configurazioni

$$= \frac{T \left(1 - \frac{1}{\gamma^{N-1}}\right)}{1 - \frac{1}{\gamma}} \quad (7.12)$$

$$= T \cdot \frac{\gamma^{N-1} - 1}{\gamma - 1} \cdot \frac{1}{\gamma^{N-2}} \quad (7.13)$$

7.1.2 La verifica del modello

Sono stati eseguite alcune prove con link a varia velocità, rilevando il numero di pacchetti effettivamente spediti per ogni chirp: come si può verificare in Tabella 7.1, i valori sperimentali misurati soddisfano esattamente la formula 7.3 trovata.

I test in laboratorio hanno consentito anche di verificare sperimentalmente la validità della funzione ricavata per la determinazione della frequenza di produzione della misura. Impostando opportunamente i parametri di configurazione, è stato possibile ottenere l'invio di chirp contenenti 14 pacchetti da 1500 *byte*. Poiché il valore predefinito per la banda utilizzata da pathChirp è pari a 0,3 *Mbps*, si ricava facilmente che

$$T_{mis} = \frac{1500 * 8 * 14bit}{300000bit/s} = 0,56s$$

Come previsto dalla teoria, il software ha effettivamente fornito stime distanziate tra loro di circa 0,6 *s*.

7.2 Il problema della sovrastima

Diversi studi comparativi tra i software per la stima della banda disponibili [72] [69] [73], pur confermando l'efficienza di pathChirp, hanno evidenziato come le stime ottenute con questo software forniscano valori superiori del 10-20% rispetto all'effettiva disponibilità di banda. Questa sovrastima peraltro persiste anche in totale assenza di cross-traffico: su un cammino completamente scarico ad 1 *Gb/s*, il tool ha fornito stime anche pari a 1100 *Mbps*. L'unica spiegazione ad oggi fornita per questo problema [73] è purtroppo insoddisfacente: l'ipotesi di errori nel calcolo delle tempistiche giustificerebbe infatti, come indicato dagli stessi ricercatori, misurazioni errate in presenza di cammini con scarsa banda disponibile. Al contrario, il problema è presente anche in reti completamente scariche e con quantità molto elevate di banda disponibile; inoltre non è chiaro per quale motivo questi errori dovrebbero presentarsi sistematicamente e comportare solo sottostime nel calcolo dei tempi. In accordo con i risultati ottenuti dagli altri gruppi di ricerca, le prove effettuate in laboratorio hanno fornito risultati sovrastimati. In particolar modo, connettendo il

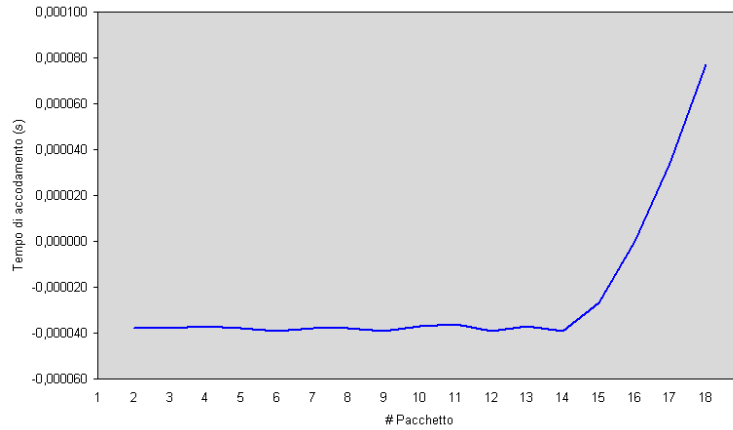


Figura 7.2: Profilo reale di un chirp

ricevente ed il destinatario con un link diretto a 100 *Mbps* e variando il tasso d'invio dei chirp tra 10 e 200 *Mbps* ($\gamma = 1, 2$), sono state ottenute misure sovrastimate di circa il 15%. Grazie allo studio effettuato in laboratorio sull'invasività di pathChirp, è stato possibile calcolare con estrema facilità il numero di pacchetti inviati per chirp e, più importante, il rate corrispondente a ciascuno di essi; i risultati ottenuti hanno mostrato dati molto interessanti. Esclusi infatti i valori rumorosi³, le misurazioni hanno mostrato un trend costante ed assestato in corrispondenza dei 115 *Mbps*. E' facile verificare che la misura trovata è esattamente corrispondente alla velocità di spedizione associata al quindicesimo pacchetto, il primo che dovrebbe sperimentare un tempo di accodamento crescente, anziché al sedicesimo. E' stato perciò modificato il software, riuscendo ad ottenere in output anche i tempi di accodamento calcolati per ogni chirp. Come mostra tuttavia il grafico 7.2, il profilo dei ritardi presenta l'inizio dell'ultima escursione in corrispondenza del pacchetto 15.

Verificata quindi la correttezza del modello teorico e la corrispondenza con i risultati ottenuti, si è compreso come la causa dell'errore fosse da ricercare nell'implementazione software. Un'attenta analisi del codice ha mostrato come, curiosamente, tra i diversi tempi di accodamento venisse erroneamente considerato anche quello sperimentato dal primo pacchetto; poiché tutti i ritardi dei successivi vengono calcolati utilizzando il primo come riferimento, è facile intuire come questo non possa che valere sempre zero. L'introduzione di questo valore superfluo nell'analisi del profilo dei chirp non ha conseguenze rilevanti sul calcolo della media; occorre tuttavia considerare che, poiché ad esso non è associato alcun rate di spedizione, il vettore contenente le velocità di invio dovrebbe essere conseguentemente traslato di una unità. In altre parole, come visibile in Tabella 7.2, l'introduzione del valore "inutile" tra i tempi di ritardo ha comportato un'errata corrispondenza tra il profilo del chirp ed i valori di banda utilizzati.

Il software è stato pertanto corretto, ottenendo risultati molto più precisi.

³Principalmente dovuti a chirp parzialmente scartati per errate tempistiche del sistema operativo o delle schede di rete

N pacchetto	Rate associato	Rate effettivo
1	10,817	-
2	12,981	10,817
3	15,577	12,981
4	18,693	15,577
5	22,431	18,693
6	26,918	22,431
7	32,301	26,918
8	38,761	32,301
9	46,514	38,761
10	55,816	46,514
11	66,980	55,816
12	80,376	66,980
13	96,451	80,376
14	115,740	96,451
15	138,889	115,740
16	166,667	138,889
17	200,000	166,667
18	<i>out of bounds</i>	200,000

Tabella 7.2: Rate associati ai vari pacchetti nell'implementazione software

7.3 I limiti nella risoluzione

Accanto al problema della sovrastima, vi è un altro aspetto estremamente insidioso e trascurato dalla ricerca. Come è noto dalla formula precedentemente ottenuta, il numero N di pacchetti per chirp è proporzionale al rapporto tra i valori B_{min} e B_{max} istantanei utilizzati e non al loro valore assoluto. Si supponga di conoscere il valore della banda disponibile A con un'incertezza pari a $\pm 40\%$: dopo aver impostato i valori limite della banda a $B_{min} = 0,6 \cdot A$ e $B_{max} = 1,4 \cdot A$, sarebbe quindi possibile eseguire misurazioni con pathChirp. E' facile verificare che, in accordo con la 7.1.1, N rimane costante e pari a 6 qualsiasi sia la quantità della banda disponibile A . Questo rappresenta un problema all'aumentare del valore di A poiché, se il numero di pacchetti è il medesimo, al crescere del rate essi si devono distribuire su uno spettro di bande più ampio e la distanza in valore assoluto aumenta di conseguenza. In Figura 7.3 sono indicate le velocità istantanee R_i associate al 3° ed al 4° del *chirp* inviato nelle condizioni sopra descritte. E' interessante osservare come la differenza tra i rate di invio R_i di pacchetti successivi cresca linearmente con l'aumentare dell'ampiezza dell'intervallo $[B_{min}, B_{max}]$.

7.3.1 L'entità dell'errore

Poiché il numero di tassi di invio sperimentabili per ogni chirp è necessariamente limitato, è possibile commettere nella stima un errore ϵ di approssimazione. Infatti pathChirp individua il primo rate, tra quelli utilizzati in spedizione, che inizia a saturare il cammino di rete e fornisce come risultato la precedente velocità istantanea

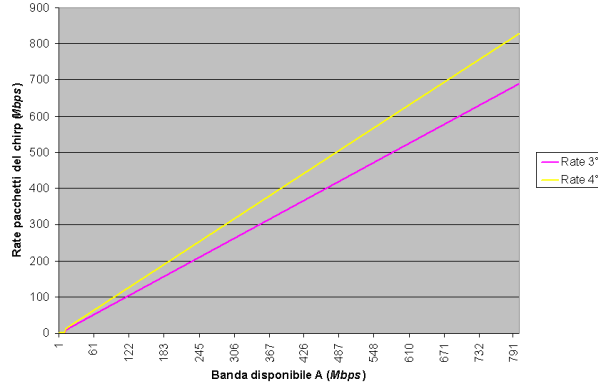


Figura 7.3: Velocità istantanee di pacchetti successivi (chirp con $N = 6$, $P=1$ e $B_{min}, B_{max} = \pm 40\%A$)

adottata per l'invio del pacchetto. Sebbene quindi venga fornito dal software un singolo risultato, l'effettiva banda disponibile è compresa tra questi due valori. Ipotizzando che la banda stimata corrisponda alla velocità di spedizione dell' n -esimo pacchetto del chirp, l'errore (negativo perché si ha una sottostima) è pari a:

$$0 \geq \epsilon > -B_{min}\gamma^{n-2} \left(1 - \frac{1}{\gamma}\right) \quad (7.14)$$

Come visto anche in Figura 7.3, l'imprecisione può quindi arrivare a parecchi *Mb*; considerando invece l'errore di sovrastima introdotto originariamente dal software, il valore assoluto dell'incertezza potrebbe persino raddoppiare. Alla luce di questi risultati, non deve perciò stupire l'entità della sovrastima ottenuta dai ricercatori del CAIDA nelle High-Speed Network [69]: infatti, se a bassi rate questa approssimazione è meno visibile⁴, in un *G-link* può benissimo introdurre 100 *Mbit* di surplus.

7.3.2 Accorgimenti per aumentare la risoluzione

Come indicato nella formula 7.14, l'errore commesso nella stima aumenta in media al crescere di N . Questo è facilmente comprensibile poiché la distanza tra i rate di trasmissione di pacchetti successivi cresce esponenzialmente; perciò, quanto più vicino alla fine del chirp sarà il punto di congestione, tanto più ampio sarà l'intervallo di incertezza. Per ridurre l'errore è quindi importante spostare l'istante di saturazione del cammino di rete nella parte iniziale del treno di pacchetti inviato: per esempio, individuato con la prima misurazione un range indicativo della banda disponibile, si potrebbe semplicemente traslare lo spettro delle velocità di spedizione avvicinando B_{min} a N^5 . Sono stati individuati altri possibili accorgimenti per migliorare la risoluzione della stima effettuata da pathChirp, specialmente in presenza di cammini di rete ad elevata capacità, dove l'entità delle approssimazioni è rilevante. Una prima soluzione potrebbe consistere nell'invio di più chirp ravvicinati e con intervalli di

⁴Ma comunque presente, come hanno mostrato alcuni rapidi test effettuati in laboratorio reti su link nominali a 10 *Mbps*

⁵Mantenendo comunque una "distanza di sicurezza", utile nel caso in cui la banda disponibile diminuisca leggermente negli istanti successivi

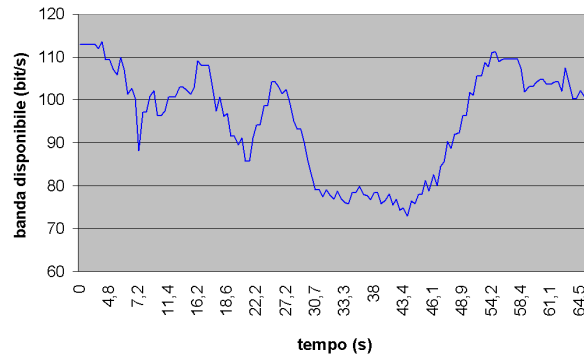


Figura 7.4: Risultato di una breve misurazione effettuata da pathChirp

$[B_{max}, B_{min}]$ aventi stessa ampiezza ma sfasati tra loro; adottando poi un algoritmo di ricerca lineare o binaria, si potrebbe riuscire a determinare il valore esatto della banda disponibile. Perché l'algoritmo possa però convergere ad un valore, è necessario ipotizzare che la banda disponibile rimanga costante per l'intera durata delle misurazioni. In alternativa, come visibile anche nella formula 7.14, si potrebbe diminuire il coefficiente di dispersione γ per ridurre l'errore [74]. In questo modo il numero di pacchetti per chirp aumenterebbe, diminuendo di conseguenza la distanza tra essi in termini di banda misurata. Anche questa soluzione non è ovviamente esente da problemi, poiché l'utilizzo di treni più lunghi comporta una maggior probabilità di incontrare burst di traffico durante il transito in rete. In entrambi i casi queste tecniche, a parità di frequenza di produzione della misura, comportano l'invio di una maggiore quantità di traffico di prova: se impiegate però in reti ad elevata velocità (≥ 100 Mbps), il surplus può essere considerato comunque trascurabile.

7.4 Available-bandwidth Smart Smoothing On-Line

Se si supera l'idea del modello ideale di rete, caratterizzato da traffico fluido e da tempi di *processing* e problematiche di accesso al canale trascurabili, è facile intuire come le stime possano venir perturbate da numerose componenti rumorose. In Figura 7.4 è mostrato il risultato di una breve rilevazione della banda disponibile fornita da pathChirp senza alcuna post-elaborazione.

Si potrebbe cercare di estendere il modello di rete utilizzato, considerando la presenza di errori intrinseci nel sistema di calcolo delle tempistiche effettuato a livello software, di canali condivisi e di dispositivi di interconnessione di livello 2. Per alcuni di questi problemi non esiste tuttavia ad oggi una soluzione definitiva e, in ogni caso, il modello ottenuto risulterebbe eccessivamente complesso. L'approccio adottato in questo lavoro è invece differente: anziché cercare di includere nel modello di rete la descrizione delle componenti non deterministiche, si è cercato di introdurre nuove tecniche al fine di minimizzarne l'impatto sul risultato finale fornito all'utente.

7.4.1 I requisiti del sistema di filtraggio

Per questo motivo il principale contributo alla soluzione del problema della valutazione della banda offerto da questa tesi sta nella realizzazione di ASSOLO (Available-bandwidth Smart Smoothing OnLine), una versione modificata di pathChirp. Sono stati inizialmente identificati alcuni requisiti fondamentali per lo sviluppo di questo nuovo strumento di stima:

- capacità di fornire stime online⁶
- limitato costo computazionale
- tempo massimo di esecuzione (*wcet*) noto, per consentire l'utilizzo anche in sistemi con vincoli real time
- possibilità di regolare la risposta del filtro stesso agendo su un numero di parametri limitato
- capacità di sfruttare le correlazioni esistenti tra misurazioni successive per filtrare i valori rumorosi

Quest'ultimo aspetto è estremamente importante per poter ridurre l'effetto di componenti rumorose nel calcolo della stima. Diversi studi hanno mostrato infatti come il traffico in Internet presenti forti correlazioni, dette di dipendenza sul lungo periodo (LDR): misurazioni successive, distanti tra loro dalle decine di secondi ad un'ora, hanno infatti spesso fornito risultati mediamente comparabili. Effettivamente l'osservazione di alcune statistiche sul traffico presente sui nodi di rete permette di identificare, nonostante la rumorosità delle misurazioni istantanee, chiari trend di crescita o diminuzione⁷. Una singola stima non dovrebbe essere perciò valutata indipendentemente dalle altre, bensì confrontata con quelle ottenute nei secondi precedenti.

Di seguito verrà quindi presentata la soluzione a questo problema identificata durante il lavoro di tesi; i risultati ottenuti saranno confrontati con altri algoritmi proposti in letteratura e con il semplice calcolo utilizzato dagli autori di pathChirp. Quest'ultimo, come si vedrà, soddisfa solo parzialmente i criteri sopra enunciati.

7.4.2 L'algoritmo di ASSOLO

I dati in ingresso all'algoritmo di stima implementato in ASSOLO sono ricavati inviando sequenze di N pacchetti spazati esponenzialmente, esattamente come accade con pathChirp. Allo stesso modo è stato mantenuto il metodo originale per la determinazione della banda disponibile istantanea D_k descritto in 6.2.2. A differenza del primo strumento, che utilizza una semplice media mobile, ASSOLO determina una stima della banda disponibile utilizzando VIDYA (Variable Inde DYnamyc Average), un sistema innovativo in grado di superare alcuni problemi che affliggono i tradizionali filtri Exponentially Weighted Moving Average (EWMA).

⁶Un algoritmo online è in grado di analizzare i dati in ingresso direttamente all'arrivo, senza dover attendere l'intera sequenza di valori per fornire risultati

⁷Si veda ad esempio l'andamento del grafico in Figura 5.6(a)

I limiti dei filtri EWMA classici

L'idea di utilizzare questa famiglia di algoritmi passa-basso nelle misurazioni di rete non è affatto recente. In passato famosi ricercatori quali Joe Postel [75] e Van Jacobson [42] hanno proposto algoritmi per la valutazione dell'RTT in una rete attraverso l'impiego di un tradizionale filtro a media mobile esponenziale. Gli stessi filtri EWMA sono stati proposti anche nell'ambito di ricerche sulla stima della disponibilità di banda [76] [77].

Il filtri a media mobile esponenziale [78] assegnano alle varie misurazioni un peso che decresce esponenzialmente, prestando sempre meno attenzione alle misurazioni distanti nel tempo. Detta ad esempio M_t la misura della banda istantanea disponibile e A_t la stima all'istante t , la formula generalmente impiegata è del tipo

$$A_t = \alpha M_t + (1 - \alpha)A_{t-1}$$

dove α è una costante (detta *smoothing factor*) con valore compreso tra 0 e 1. Purtroppo una scelta di questo tipo non è affatto esente da problemi: l'utilizzo di un determinato *smoothing factor* si scontra infatti con le tempistiche ed il comportamento del traffico. La scelta iniziale del peso α potrebbe premiare la stabilità a scapito della velocità di risposta, non riuscendo però a seguire adeguatamente le variazioni più rapide. Al contrario, dando maggior peso alle nuove letture e meno al passato, l'effetto di misurazioni rumorose sarebbe visibile anche nella stima fornita. Inoltre la determinazione del parametro α , che avviene spesso in modo empirico, può fornire buoni risultati in determinate condizioni di traffico e di risposta della rete ma, allo stesso modo, dimostrarsi scarsamente efficace in condizioni differenti da quelle considerate in fase di taratura.

Il filtro Variable Inde DYnamyc Average

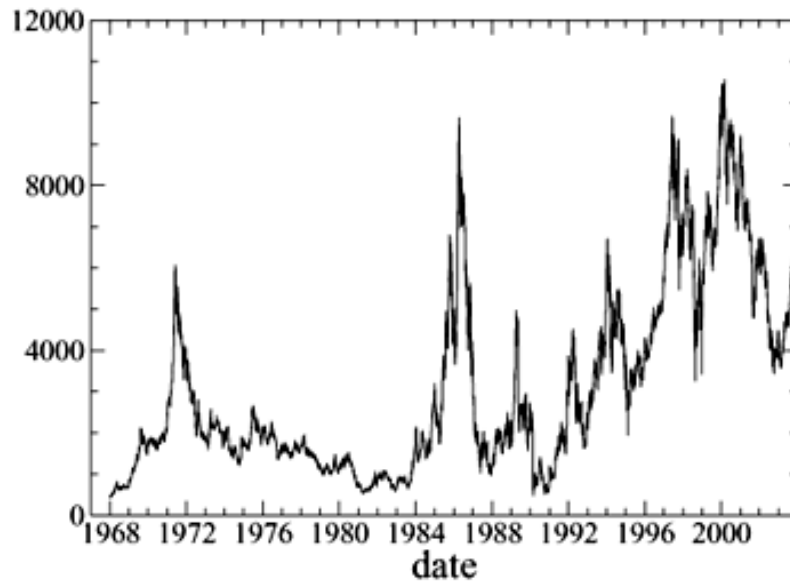
Aniché utilizzare uno *smoothing factor* statico per un filtro EWMA si è cercato di individuare una funzione di merito $f(-)$ in grado di determinare dinamicamente il coefficiente α ad ogni misurazione. Questa variante è stata introdotta allo scopo di riuscire ad assegnare pesi differenti nel calcolo della media ai valori rumorosi e alle misurazioni stabili.

$$\alpha = f(M_t, M_{t-1}, \dots, M_{t-n})$$

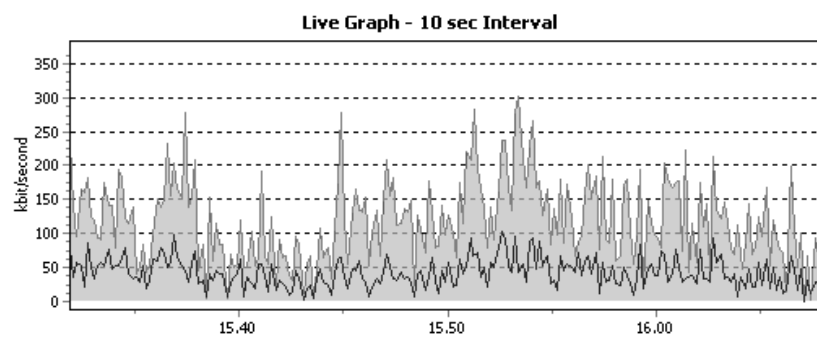
$$A_t = \alpha M_t + (1 - \alpha)A_{t-1}$$

L'algoritmo utilizzato nello specifico da ASSOLO è stato mutuato dalla finanza, un settore apparentemente lontano dalle rete di comunicazione ma che, in realtà, deve affrontare quotidianamente problematiche analoghe. In Figura 7.5 è riportato l'andamento di un indice di borsa nel tempo: esattamente come accade per il traffico in rete, l'andamento dei titoli presenta infatti dei transitori, caratterizzati da misurazioni instabili e senza un pattern ben definibile, alternati da momenti di crescita o decrescita secondo una ben determinata tendenza. Alcuni economisti hanno lavorato alla definizione di algoritmi di identificazione dei trend per poter effettuare previsioni. Tra questi un lavoro molto interessante è Variable Inde DYnamyc Average (VIDYA), una media mobile esponenziale con guadagno variabile proposta da Tushar Chande [79] [80].

Esistono in letteratura tre principali varianti di VIDYA:



(a) Indice Ibovespa (Borsa Valori di San Paolo)



(b) Statistiche di traffico fornite da un router Cisco 2006

Figura 7.5: Analogia tra l'andamento degli indici finanziari e il traffico di rete

1. VIDYA-StdDev [79] è l'algoritmo iniziale proposto da Chande ed utilizza la deviazione standard per la determinazione del parametro α . La stima della banda A_t può quindi essere scritta

$$A_t = k\sigma_n M_t + (1 - k\sigma_n)A_{t-1}$$

dove σ_n è la deviazione standard calcolata sulle ultime n misurazioni M_t e k un parametro da determinare empiricamente in fase di taratura del filtro.

2. VIDYA-CMO [81] è una variazione successivamente introdotta, basata sull'utilizzo di un nuovo coefficiente in sostituzione della deviazione standard. Il valore proposto è Chande Momentum Oscillator (CMO), un indicatore proporzionale all'affidabilità del trend individuabile nelle misurazioni più recenti. Per la determinazione del coefficiente, riferito agli ultimi n valori M_i trovati, viene calcolata la variazione ΔM_i di ogni misurazione rispetto al dato precedente:

$$\Delta M_i = M_i - M_{i-1}$$

Il valore di CMO è quindi trovato con la formula

$$\text{CMO} = \frac{\sum_{up} - \sum_{down}}{\sum_{up} + \sum_{down}}$$

Dove $\sum_{up}$ è la somma dei ΔM_i positivi e $\sum_{down}$ il totale riferito ai $\Delta M_i < 0$.

La stima della banda A_t può quindi essere scritta

$$A_t = k \cdot \text{CMO} \cdot M_t + (1 - k \cdot \text{CMO})A_{t-1}$$

dove k , parametro da determinare empiricamente, è stato introdotto in ASSO-LO per consentire un maggior controllo della velocità di risposta.

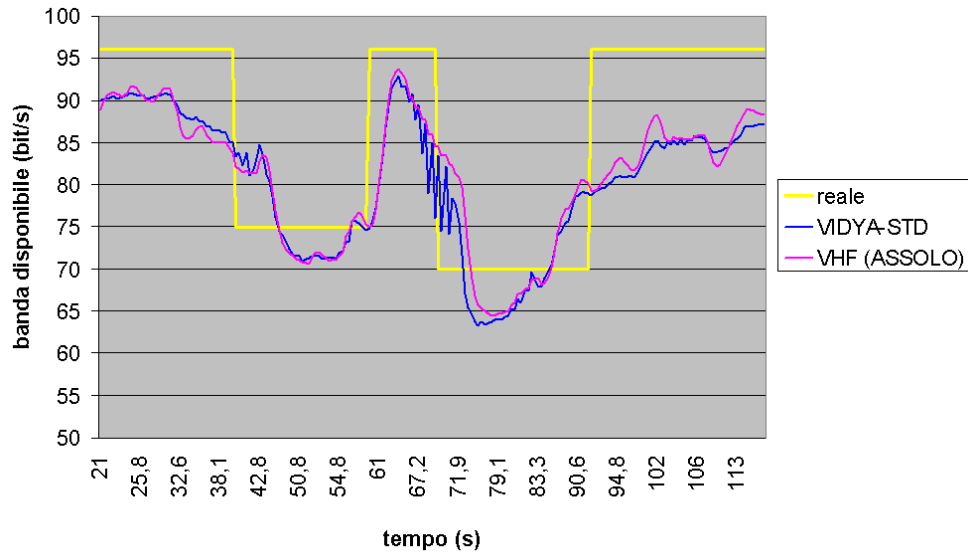
3. VHF (Vertical Horizontal Filter) [82], proposto dall'economista Adam White, è un ulteriore filtro utilizzato per individuare i trend e l'affidabilità nelle misurazioni. Considerando sempre una finestra storica di n misurazioni, il coefficiente proposto da White si calcola nel seguente modo

$$\text{VHF} = \frac{|\max_n M_t - \min_n M_t|}{|\sum_n M_t|}$$

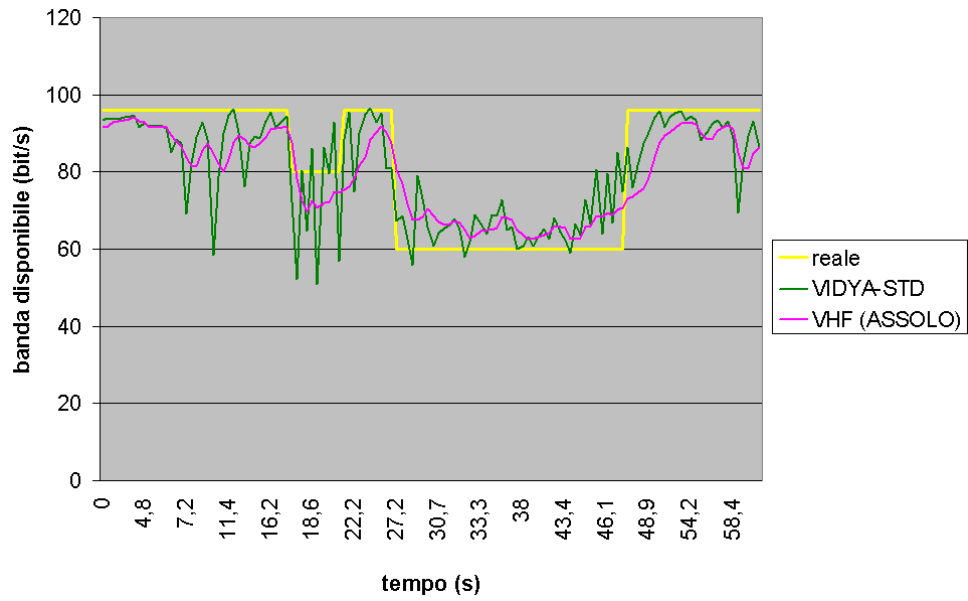
ed il valore di VHF è tanto maggior quanto più marcato è il trend che stanno iniziando ad assumere i dati in ingresso. La stima della banda A all'istante t è valutata con la formula

$$A_t = k \cdot \text{VHF} \cdot M_t + (1 - k \cdot \text{VHF})A_{t-1}$$

Anche in questo caso è stato introdotto in ASSOLO il parametro k per variare la velocità di risposta.

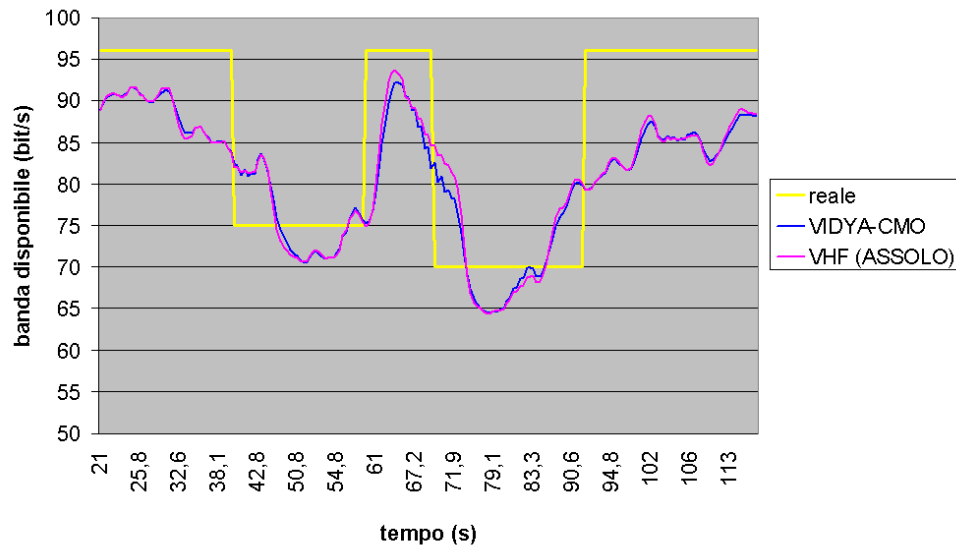


(a) Presenza di cross-traffico CBR a basso rate

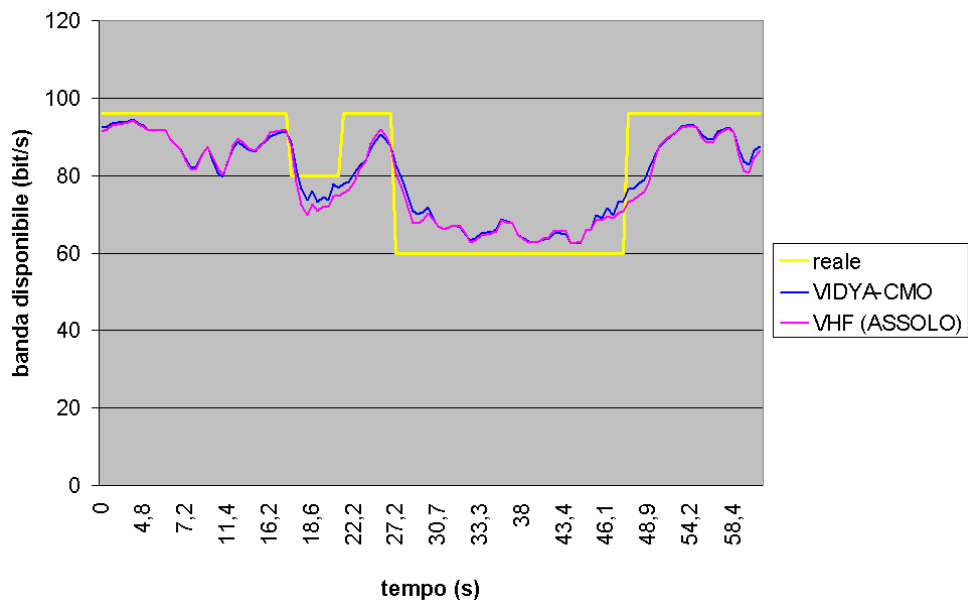


(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.6: Confronto tra VHF ed il filtro VIDYA-StdDev



(a) Presenza di cross-traffico CBR a basso rate



(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.7: Confronto tra VHF ed il filtro VIDYA-CMO

	Parametri	Valori
StdDev	$n; k$	11 ($\approx 6,5$ s); 0,3
CMO	$n; k$	7 ($\approx 4,5$ s); 0,25
VHF	$n; k$	4 ($\approx 2,5$ s); 0,35
LpEMA	$\alpha_{max}; \rho$	0,3; 0,5
SF	$n; \beta$	10 (≈ 6 s); 0,3
Median	n	10 (≈ 6 s)
Moving Average	n	4/11 ($\approx 3,5/7$ s)

Tabella 7.3: Parametri di configurazione utilizzati per le prove

	Somme	Prodotti	Divisioni	Confronti
StdDev	$O(3n)$	$O(n)$	$2 + \sqrt{-}$	
CMO	$O(n)$		1	$O(2n)$
VHF	$O(n)$		1	$O(2n)$
LpEMA	4	3	3	
SF	3	2	1	$O(n)$
Median				$O(n^2)$
Moving Average	$O(n)$		1	

Tabella 7.4: Costo computazionale dei diversi algoritmi di stima

La soluzione adottata in ASSOLO

Per valutare le prestazioni delle diverse varianti del filtro VIDYA è stato utilizzata una sequenza di misurazioni ottenute nella rete di prova in laboratorio. In particolar modo, alla stessa serie di dati sono stati applicati i diversi meccanismi di filtraggio e si è quindi cercato di individuare i parametri di configurazione, riassunti in Tabella 7.3, in grado di fornire risposte adeguate.

Come visibile in Figura 7.6 e Figura 7.7, il filtro VIDYA-StdDev ha fornito risultati discreti ma piuttosto instabili. Le varianti CMO e VHF hanno invece prodotto risultati sostanzialmente comparabili, decisamente robusti nei confronti di disturbi ma, nel contempo, rapidi nel seguire le variazioni.

Detto n il numero di misurazioni storiche considerate, le risorse computazionali⁸ richieste dai diversi algoritmi sono sostanzialmente identiche per le due varianti VHF e CMO. Come visibile in Tabella 7.4, l'algoritmo originale basato sulla deviazione standard è invece più esigente in termini di tempo di calcolo.

Nell'implementazione di ASSOLO e nel confronto con gli altri algoritmi proposti in questo settore è stato utilizzato VHF, sebbene i test condotti non abbiano evidenziato particolari vantaggi rispetto al metodo VIDYA-CMO.

7.4.3 Il confronto con gli altri algoritmi

Nonostante la semplicità di questi filtri, in letteratura esistono pochissimi lavori di questo tipo. In [83] e [84] vengono proposti ulteriori algoritmi per la determinazione

⁸Somme e differenze vengono considerate computazionalmente equivalenti

del guadagno α del filtro: Stability Filter, Error-Based filter, Low Pass e Gradient Adaptive. In accordo con quanto trovato dagli stessi ideatori, ci si è concentrati sui due metodi che hanno fornito risultati migliori, specialmente in termini di stabilità: Stability Filter e LowPass. Il comportamento di ASSOLO è stato poi confrontato anche con le risposte ottenute grazie ad un filtro mediano e, infine, con il semplice meccanismo utilizzato da pathChirp.

Per i test comparativi sono stati utilizzati alcuni tracciati ottenuti nel *testbed* in laboratorio con la versione base di pathChirp, senza alcun algoritmo intermedio di filtraggio. In una prima simulazione sono stati inviati in rete due distinti traffici a bit rate costante e pari rispettivamente a 20 *Mbps* e 30 *Mbps*, preceduti e seguiti da brevi scambi di messaggi della durata di pochi secondi. Nel secondo esperimento sono state modificate le entità dei due cross-traffici, aumentando la prima a 40 *Mbps* e portando il secondo trasferimento a 60 *Mbps*. Si è preferito applicare gli algoritmi *off-line* per poter verificare il comportamento in risposta ai medesimi dati in ingresso; effettuando solo prove nella rete reale, non sarebbe stato possibile riprodurre esattamente l'effetto delle componenti rumorose e degli errori sulla stima.

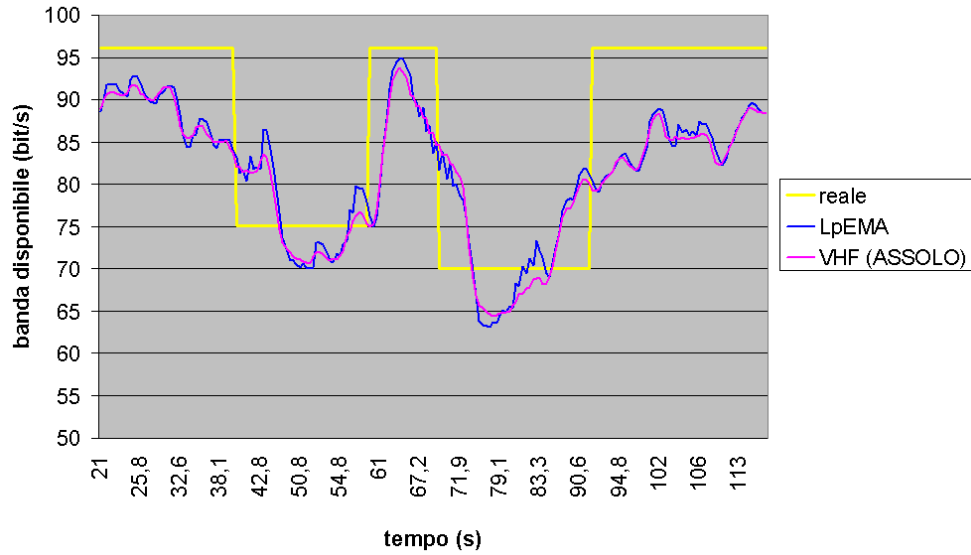
Low Pass Exponential Moving Average

LowPass EMA (LpEMA) [83] implementa un filtro passa-basso del primo ordine per determinare l' i -esimo valore del peso α . Derivato dalla classica formula per la descrizione del filtro *low pass*, il valore di α è pari a:

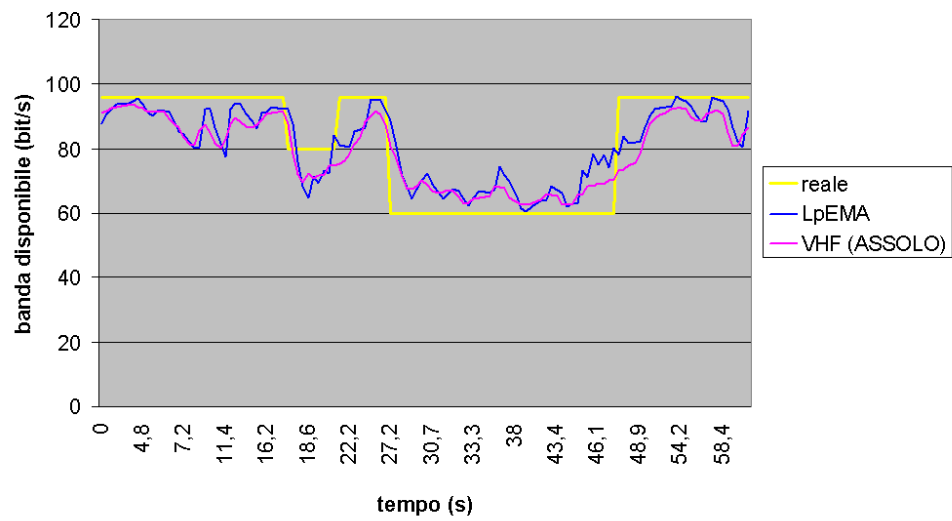
$$\alpha_i = \alpha_{max} \frac{1}{1 + \frac{|m_i|}{m_{norm}}}$$

Il gradiente m_i relativo all'ultima misurazione (e normalizzato rispetto a m_{norm}) è utilizzato come frequenza di taglio; α_{max} introduce invece un limite superiore al peso utilizzato nel filtro. I valori empirici da determinare in questo tipo di filtro sono quindi il guadagno massimo α_{max} ed il gradiente normalizzato m_{norm} . Come suggerito dagli autori, il valore di m_{norm} è stato a sua volta calcolato utilizzando un semplice filtro EWMA con peso (ρ) 0.5⁹; analogamente, è stato utilizzato per α_{max} il valore 0,3 indicato in letteratura. Queste scelte rappresentano sicuramente soluzioni sub-ottime: non è tuttavia chiaro se possa esistere e, nell'eventualità, quale possa essere un modo per migliorarne la determinazione. I risultati ottenuti utilizzando i valori proposti sono stati piuttosto deludenti. Si è quindi applicato un semplice pre-filtraggio passa-basso ai dati con un filtro a media mobile: come visibile in Figura 7.8, il sistema migliora in stabilità nei periodi a bassa frequenza ma risponde ancora con oscillazioni più ampie a transitori poco stabili o con molti disturbi (si noti ad esempio, a burst di traffico (si noti ad esempio il comportamento in b) intorno ai 10-15 *s*)

⁹In alternativa potrebbe essere calcolato come media degli ultimi n valori di m_i o impostato al massimo gradiente sperimentabile sulla rete, pari a $\frac{1}{2} \frac{C}{t_i - t_{i-1}}$



(a) Presenza di cross-traffico CBR a basso rate



(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.8: Confronto tra ASSOLO ed il filtro LpEMA

Stability Filter

L'altro metodo sopra citato, Stability Filter [84], prevede l'utilizzo di un primo filtro EWMA per determinare un coefficiente U di instabilità.

$$U_t = \beta U_{t-1} + (1 - \beta) |x_t - x_{t-1}|$$

Questi, una volta normalizzato rispetto all'instabilità massima osservata in una finestra storica di un certo numero n di valori, viene preso come coefficiente del filtro EWMA utilizzato per la valutazione della banda disponibile.

$$\alpha_t = \frac{U_t}{U_{max}}$$

Tra tutti quelli citati, il filtro in questione si è dimostrato maggiormente in grado di competere con l'algoritmo utilizzato in ASSOLO. In situazioni con misure *spot* e andamenti tendenzialmente costanti presenta una stabilità maggiore ed una velocità nella risposta leggermente superiore. Per contro, quando un disturbo o un burst coinvolgono almeno due letture successive oppure, in alternativa, quando il cross-traffico è presente ma senza pattern stabili, il filtro risponde in maniera più aggressiva “seguendo” le oscillazioni.

pathChirp

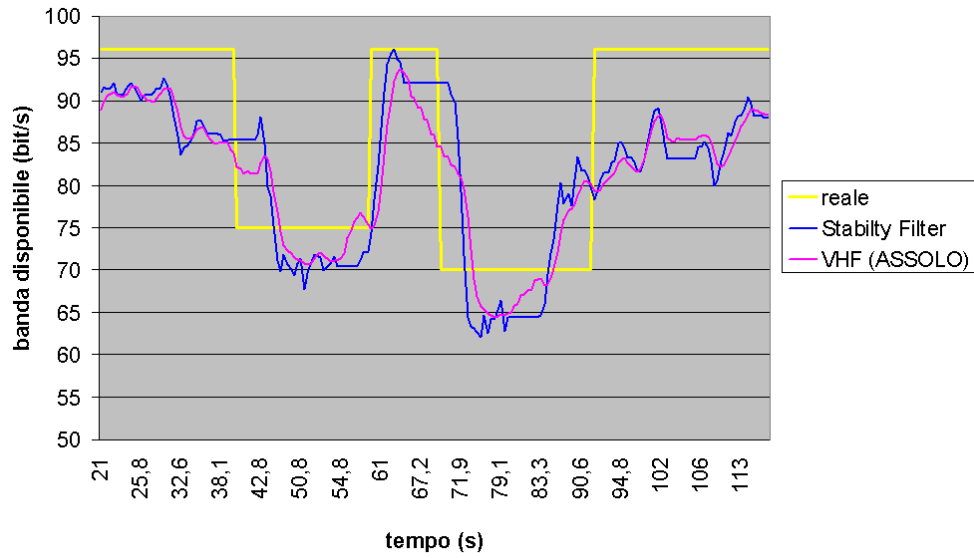
pathChirp utilizza un semplice filtro a media mobile (MA) per il calcolo della stima della banda disponibile. Questo tipo di sistema è estremamente semplice ed equivale ad effettuare una media degli ultimi n valori¹⁰.

$$A_t = \frac{M_t + M_{t-1} + \dots + M_{t-n+1}}{n}$$

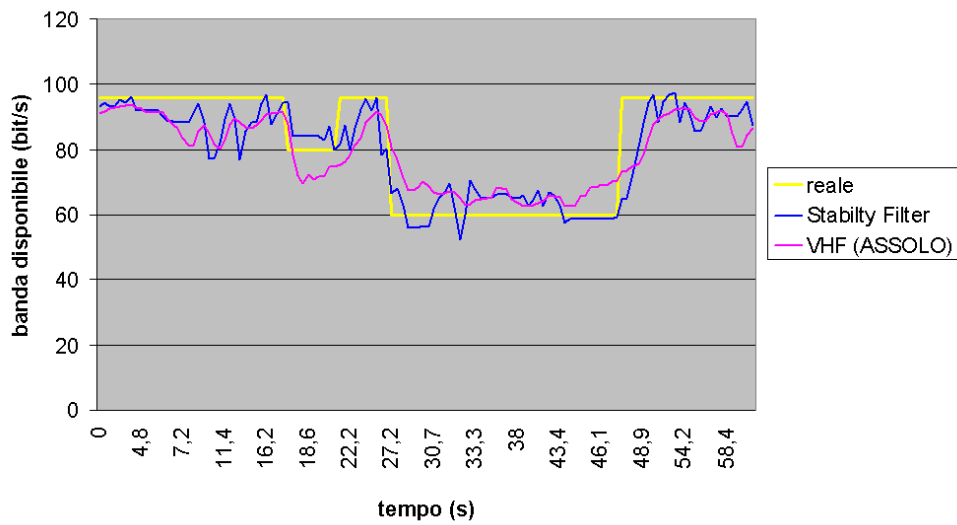
La dimensione della finestra è quindi un parametro fondamentale per la caratterizzazione del comportamento del filtro. Se infatti n assume un valore elevato, l'effetto di misurazioni rumorose viene attenuato maggiormente ma il sistema risulta piuttosto lento a reagire a bruschi cambiamenti. Per contro, un numero minore di campioni considerati nella media consente di seguire le variazioni molto rapidamente ma, allo stesso tempo, presenta maggior instabilità e fluttuazioni, inseguendo anche i valori outlier. Il valore di default utilizzato da pathChirp è pari a undici misurazioni: come visibile in Figura 7.10 i risultati presentano un andamento sostanzialmente stabile e poco sensibile ai valori rumorosi, se confrontato con i dati inizialmente presentati in Figura 7.4. Questa soluzione premia tuttavia la stabilità rispetto alla velocità di risposta: si noti infatti come il valore della stima fornito reagisca lentamente a bruschi cali nella banda disponibile. Il discorso duale vale invece per la seconda misurazione, ottenuta con un valore di n relativamente basso, pari a 4. In questo caso la risposta del filtro, illustrata in Figura 7.11 è più rapida ma caratterizzata da un'elevata instabilità.

Il filtro a media mobile proposto dagli autori di pathChirp risulta estremamente inefficiente poiché, nel calcolo della media, viene assegnato lo stesso peso a misurazioni rumorose e rilevazioni esatte. Inoltre, variando il numero di dati storici considerati

¹⁰ n è anche detta apertura della finestra

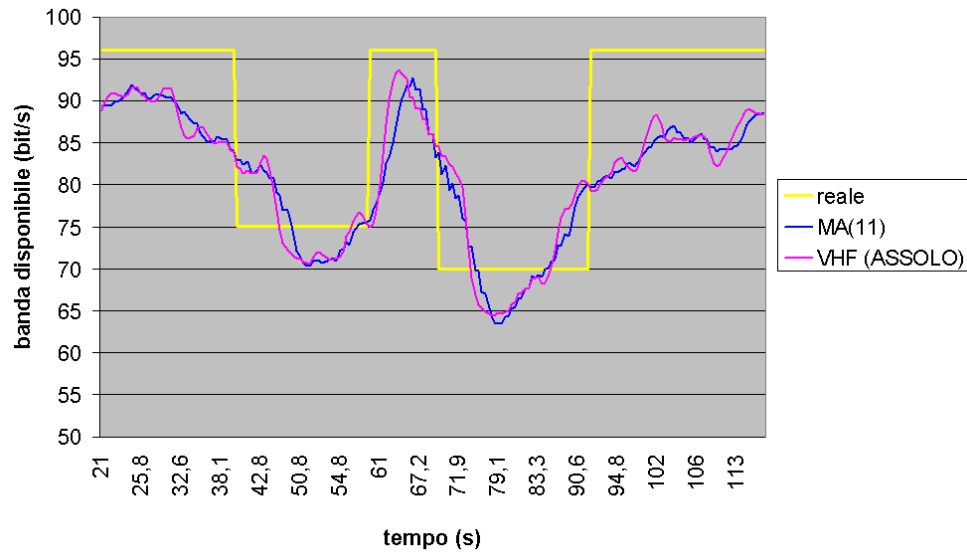


(a) Presenza di cross-traffico CBR a basso rate

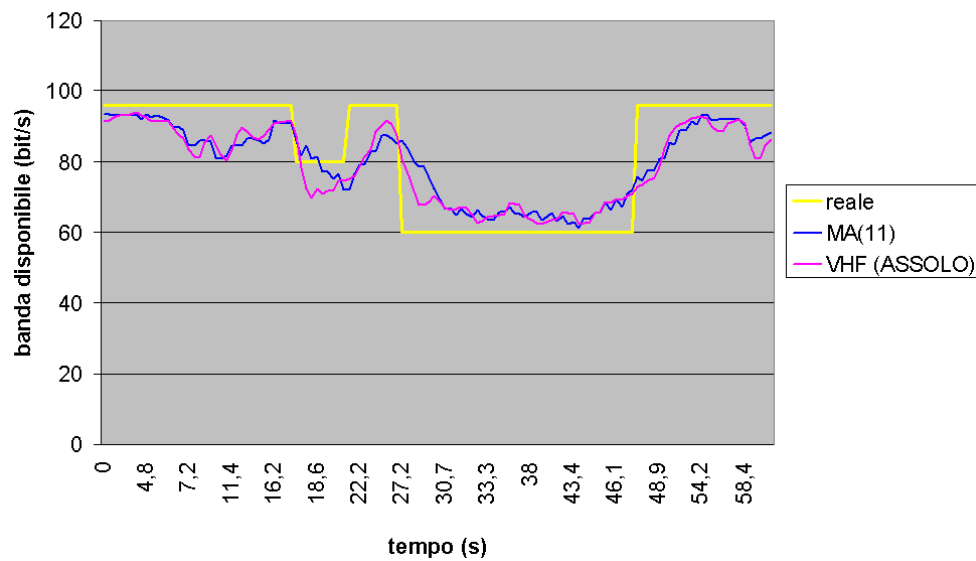


(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.9: Confronto tra ASSOLO e Stability Filter

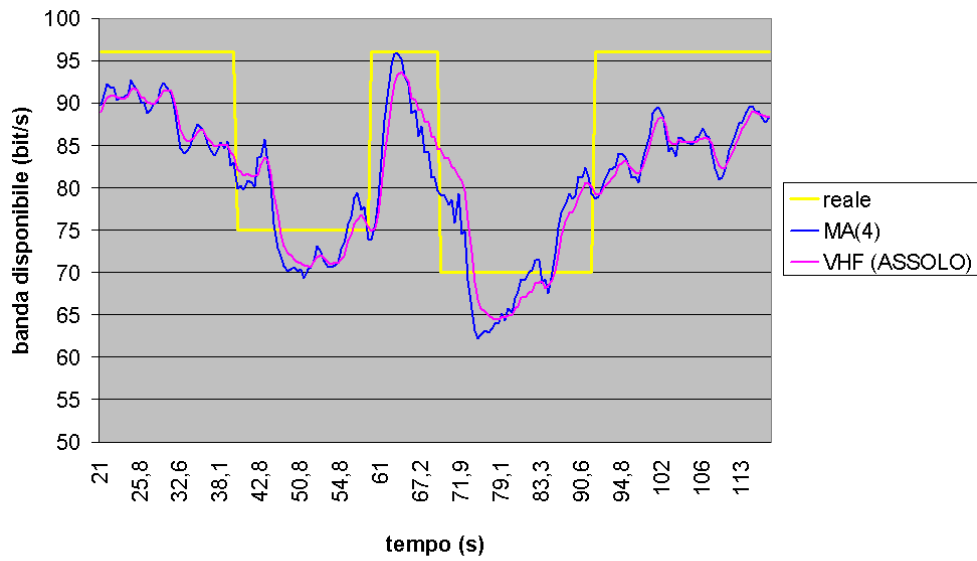


(a) Presenza di cross-traffico CBR a basso rate

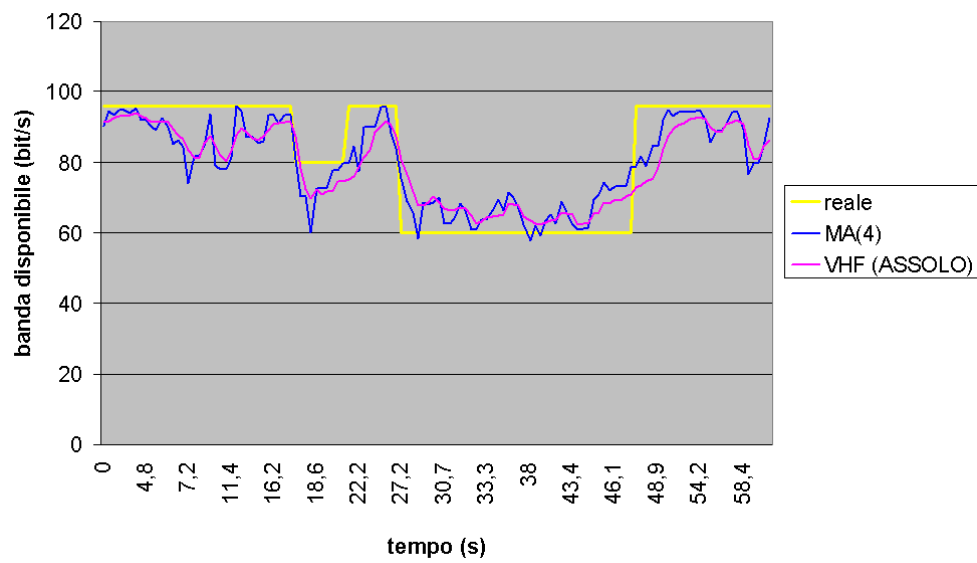


(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.10: Confronto tra ASSOLO e pathChirp (*MovingAverage(11)*)



(a) Presenza di cross-traffico CBR a basso rate



(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.11: Confronto tra ASSOLO e pathChirp ($MovingAverage(4)$)

nel calcolo della media, è possibile aumentare la stabilità dei risultati o la velocità di risposta. Questi aspetti sono però duali e non è possibile favorirne uno senza sacrificare l'altro.

Median Filter

La mediana per una serie finita di valori può essere individuata disponendo gli ultimi n campioni in ordine crescente e selezionando il valore centrale¹¹. Un filtro mediano fornisce una misura della tendenza centrale di una distribuzione estremamente robusta nei confronti di dati spuri isolati e outlier [85]: la dimensione della finestra determina quindi la velocità di risposta del filtro. Nel caso ideale di un ingresso a scalino senza rumore, il filtro mediano rileverà il salto della funzione in ingresso con un ritardo pari a $\frac{n}{2}$. Inoltre, sebbene i singoli valori outlier vengano facilmente eliminati, sequenze di dati piuttosto instabili non vengono gestite ottimamente. Come visibile in Figura 7.12, se in questi casi i filtri a media mobile non eliminano completamente il rumore ma effettuano uno smoothing del profilo originale, la mediana può invece inserire valori isolati fuori scala all'interno di profili regolari. Infine occorre considerare come in questo lavoro di tesi siano state utilizzate sorgenti di cross-traffico CBR; non esistono invece studi riguardanti il filtro mediano in situazioni reali o, ad esempio, in presenza di trasmissioni con distribuzione poissoniana.

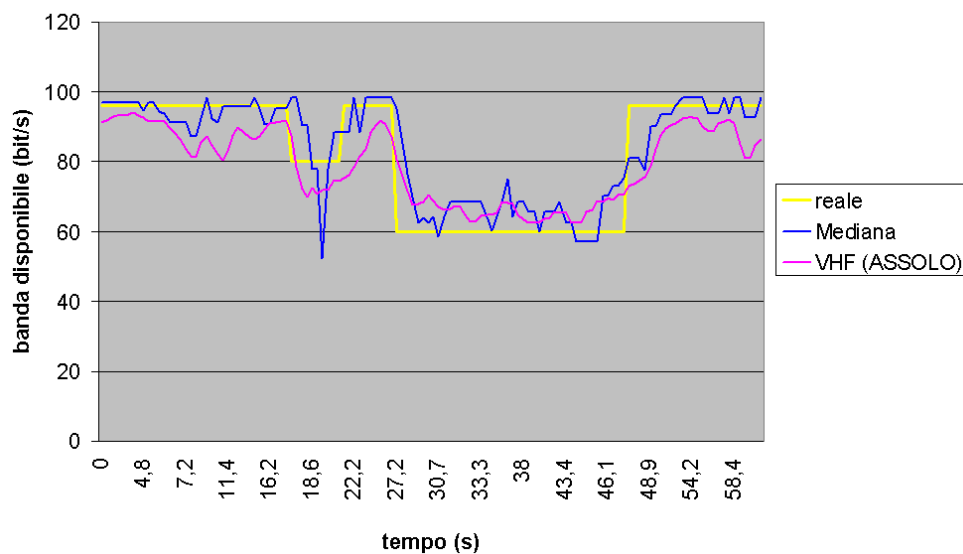


Figura 7.12: Confronto tra ASSOLO e un filtro mediano

Ulteriori filtri utilizzabili

In [84] viene anche presentato un filtro “Flip-Flop”, derivante dalla teoria delle carte di controllo e basato sull'utilizzo di due EWMA a coefficienti fissi: il primo implementa un passa-basso molto aggressivo ed il secondo uno molto conservativo. Sfruttando la

¹¹Nel caso in cui il numero di campioni sia pari, la mediana corrisponde alla media dei due valori centrali

deviazione standard delle misurazioni rispetto ai valori attesi, è possibile determinare quale filtro utilizzare di volta in volta. Anche in questo caso i parametri alfa dei due EWMA devono essere determinati empiricamente, ma i risultati iniziali presentati sono decisamente promettenti. In alternativa alla tradizionale media mobile si potrebbe utilizzare un filtro di Kalman [84] [59], algoritmo peraltro già implementato in alcuni software di misurazione della banda attualmente non disponibili (Cfr. 5.2.3). Sebbene l'ipotesi possa apparire forzata, dal momento che poco o nulla può essere ipotizzato sullo spazio degli stati della rete e sulla reale distribuzione del rumore, i risultati ottenuti sono interessanti. Un altro meccanismo già proposto, con discreti risultati, è stato uno stimatore basato su logica fuzzy, robusto rispetto a spike transitori [47].

7.4.4 Quantizzazione

Come anticipato nel Capitolo 5, la determinazione della quantità di banda disponibile è fondamentale per adattare di conseguenza la qualità di codifica di una trasmissione audio o video (VoIP, IPTV, Webcast etc..) in rete. Indipendentemente dal tipo di codifica adottata, è lecito supporre che i gradi di compressione possibili siano un numero finito e variabile a livelli discreti¹². Da ciò la necessità di quantizzare la disponibilità di banda misurata, limitandola così ad un possibile sottoinsieme di valori discreti.

Il processo di quantizzazione introdotto in ASSOLO è stato però reso ulteriormente stabile con l'impiego dell'isteresi, un fenomeno presente in natura e riprodotto anche in alcuni circuiti elettronici di comparazione (il più noto è forse il Trigger di Schmitt [86]) e nelle reti telefoniche mobili [87]. Dati alcuni livelli di quantizzazione d_i equidistanti tra loro, la misura quantizzata d dell'ingresso x_t è definita dalla funzione di quantizzazione $q(t)$ che approssima al livello discreto inferiore più vicino. L'effetto memoria introdotto dalla funzione di isteresi modifica il comportamento nel modo seguente:

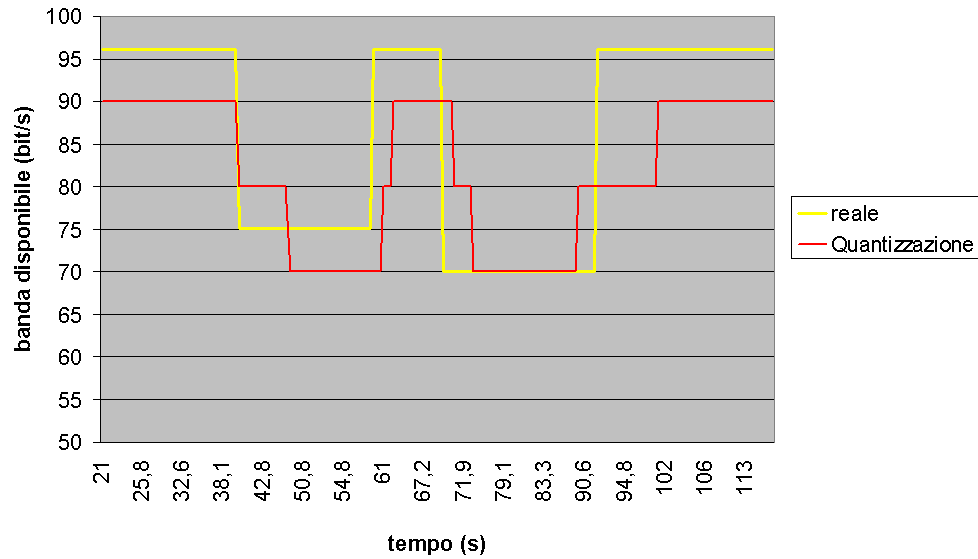
$$d(x_t) = q(|x_t - d(x_{t-1}) - \sigma|)$$

In sostanza la misurazione effettuata “entra” in un livello di quantizzazione, dal quale può uscire solo se supera di un quantità di soglia σ i valori limite dell'intervallo. Questo accorgimento permette di evitare che lievi oscillazioni a cavallo di due livelli di quantizzazione facciano assumere al segnale un andamento ad “onda quadra”; come osservabile in Figura 7.13, il meccanismo è estremamente efficace.

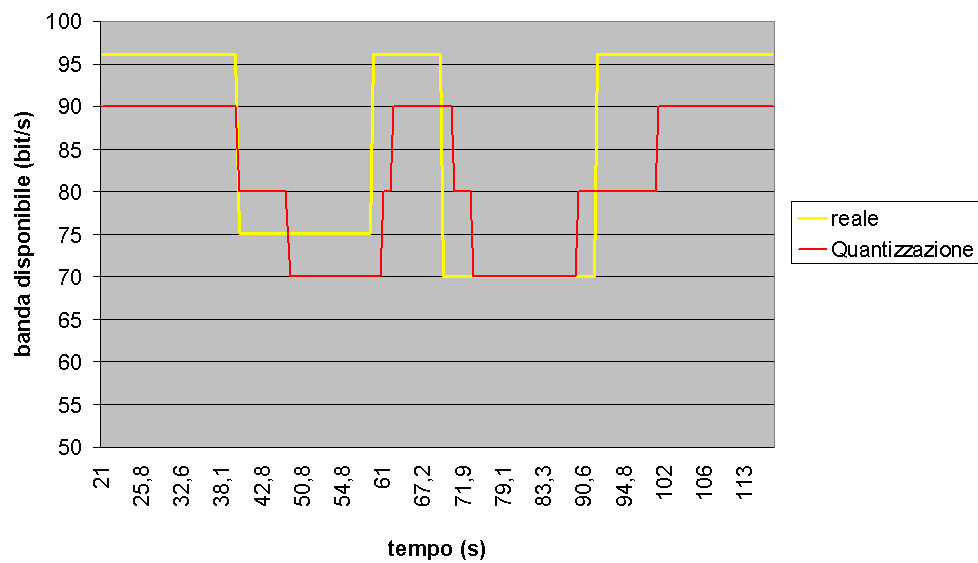
7.5 Il tasso di servizio variabile

I primi test condotti in laboratorio hanno fornito risultati completamente fuori scala: in particolare, pur utilizzando link Ethernet a 100 *Mbps* completamente scarichi, sia PTR che pathChirp hanno fornito risultati nell'ordine dei 25 *Mbps*. Fiduciosi nella bontà dei software disponibili in rete, si è perciò supposto che la causa potesse essere da ricercare invece nel *testbed* realizzato per le prove. L'ipotesi ha trovato conferma

¹²Lo standard MPEG-1 Layer 3 (meglio noto come MP3) prevede diversi livelli di codifica a bit rate costante: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 e 320 kbit/s. Le frequenze di campionamento possibili sono 32, 44.1 e 48 kHz ed è anche specificabile la modalità stereo o mono.



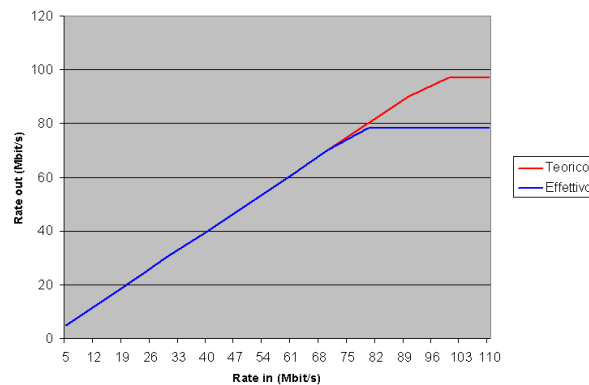
(a) Presenza di cross-traffico CBR a basso rate



(b) Presenza di cross-traffico CBR a rate più sostenuti

Figura 7.13: Quantizzazione con isteresi effettuata da ASSOLO

CPU	Sistema Operativo	10 Mbit	100 Mbit
PII@233MHz	Linux	ok	<i>sovrastima</i>
	FreeBSD	ok	<i>sovrastima</i>
PIV@1200Mhz	Linux	ok	ok
	FreeBSD	ok	ok

Tabella 7.5: Risultati ottenuti con diverse configurazioni**Figura 7.14:** Differenza tra capacità nominale ed effettiva nella rete considerata

con la sostituzione dei router di interconnessione, inizialmente implementati utilizzando computer obsoleti (PentiumII@233MHz) ed il sistema operativo FreeBSD; l'impiego di apparecchiature più recenti ha infatti consentito di effettuare misure e simulazioni senza ulteriori problemi. Si è voluto comunque approfondire ulteriormente la problematica inizialmente incontrata, cercando di individuare quale potesse essere la causa della pesante sottostima della banda disponibile. Come illustrato in Tabella 7.5, l'impiego di varie configurazioni ha consentito riprodurre il fenomeno indesiderato sui vecchi calcolatori quando dotati di schede di rete recenti, utilizzate in sostituzione di quelle a 10 *Mbps* originariamente fornite. Curiosamente però un trasferimento file in rete attraverso gli stessi dispositivi ha raggiunto velocità ben tre volte superiori a quelle massime stimate da pathChirp.

Per individuare quindi l'esatta capacità della configurazione incriminata sono state effettuate alcune prove, inviando con Iperf traffici aventi durata di 10 secondi e velocità crescenti, da 5 fino a 100 *Mbps*; come visibile in Figura 7.14, i valori ottenuti in presenza di elevate quantità di traffico sono ben distanti da quelli teorici. Il grafico mostra come effettivamente, a fronte di una capacità nominale di 100 *Mbps*, la limitata potenza di calcolo della CPU riduca ad 80 *Mbps* l'effettivo tasso di servizio massimo del router.

Ancora più interessante è tuttavia l'analisi del *jitter* medio associato ai diversi tassi d'invio di flussi CBR. La scelta della natura del traffico utilizzato per le prove non è casuale: supponendo infatti che il tasso di servizio del router sia costante e deterministico, pacchetti appartenenti allo stesso flusso dovrebbero sperimentare tempi di servizio uguali e, in condizioni di equilibrio del sistema, *jitter* sostanzialmente nullo. Anche in questo caso i risultati forniti da Iperf, visibili in Figura 7.15, hanno

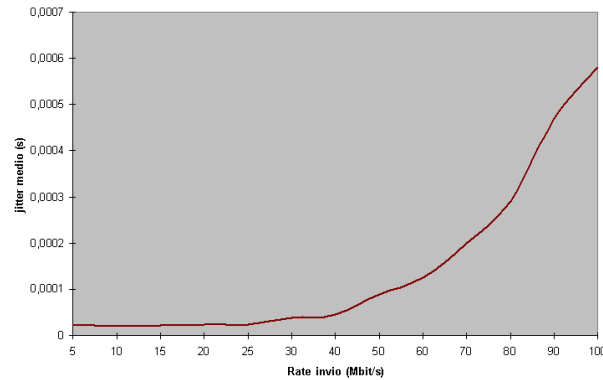


Figura 7.15: Jitter medio calcolato da Iperf in funzione del traffico presente in rete

mostrato la distanza tra il modello teorico e il comportamento reale dei dispositivi.

Alla luce di questi dati, i risultati ottenuti con pathChirp sono comprensibili se si considera che i vari strumenti basati sul modello PGM ipotizzano proprio un *jitter* costante per tassi di invio inferiori alla banda disponibile. Un valore crescente all'aumentare della velocità di invio dei pacchetti comporterebbe infatti un incremento delle distanze temporali tra i vari pacchetti, interpretato dallo strumento come un tempo di accodamento crescente sui nodi intermedi. Da ciò il valore di 25 *Mbps* fornito dai software di stima, corrispondente alla massima banda con *jitter* associato costante. L'anomalo comportamento individuato è probabilmente imputabile ai ritardi introdotti dal sistema operativo¹³, trascurabili a bassi livelli di traffico ma rilevanti al crescere del volume di dati da gestire.

Il comportamento considerato ha mostrato come l'ipotesi di tassi di servizio costanti dei router, adottata da moltissimi software di stima della banda, sia effettivamente fondamentale per il corretto funzionamento degli algoritmi. Inoltre, sebbene la maggior parte dei router di Internet non sia oggi interessata dal problema, un futuribile aggiornamento parziale dell'infrastruttura di rete potrebbe mettere in crisi le apparecchiature più datate e dar luogo a comportamenti non prevedibili del sistema.

¹³Ad esempio nella gestione dei cambi di contesto, delle *cache* o dei *buffer* e nel servire gli *interrupt* lanciati dalle interfacce di rete.

Capitolo 8

Conclusioni

Se ho potuto vedere più lontano degli altri, è perché stavo sulle spalle dei giganti.

Isaac Newton

Nonostante sia stato proposto nel 2003, pathChirp ha dimostrato ancora una volta di essere uno strumento estremamente valido e versatile. La soluzione trovata in laboratorio al noto problema di sovrastima ha consentito inoltre di ridurre notevolmente l'errore commesso durante le misurazioni. Questo mette per altro in discussione tutti i risultati ottenuti sino ad oggi nei test comparativi tra i diversi software, falsati dalla polarizzazione che ha da sempre afflitto i valori forniti dal software della Rice University.

Lo sviluppo di ASSOLO ha permesso poi di mostrare come, grazie all'introduzione di un sistema di filtraggio adeguato, sia possibile sfruttare le correlazioni esistenti tra misurazioni successive e ridurre notevolmente l'effetto di errori sporadici sui risultati finali. Studiando il comportamento dei vari algoritmi in presenza di differenti sorgenti di cross-traffico, ulteriori ricerche potrebbero individuare il filtro in grado di fornire i risultati migliori a livello globale. Sempre a questo riguardo, sarebbe poi interessante approfondire il rapporto esistente tra i parametri di configurazione del filtro, le caratteristiche del traffico e le tempistiche di misurazione, per poter così rendere automatica la taratura del sistema.

Il problema della stima della banda disponibile rimane quindi, nel complesso, una tematica ancora aperta e che offre interessanti spunti per ulteriori ricerche. Nonostante questo i risultati ottenuti sono sufficientemente precisi ed impiegabili in diverse applicazioni: già oggi alcuni piattaforme proprietarie di streaming¹ e telefonia su IP [88] impiegano meccanismi più o meno complessi per stimare la banda disponibile ed adattare, di conseguenza, il traffico inviato. Sarebbe interessante integrare gli algoritmi proposti con analoghi sistemi di comunicazione opensource, introducendo apposite API all'interno del codice di ASSOLO, e valutarne quindi i benefici.

L'individuazione di un meccanismo di filtraggio efficace ha permesso ad ASSOLO di ottenere risultati superiori a quelli forniti dai tradizionali strumenti di stima della banda disponibile. E' lecito ipotizzare che, implementando un opportuno sistema

¹Ad esempio Microsoft Windows Media (Intelligent Streaming)

di taratura automatica del guadagno, l'algoritmo di filtraggio proposto potrebbe essere utilizzato con successo anche per affrontare la più complessa tematica della Quality of Service (QoS). Con la recente esplosione di applicazioni multimediali, le problematiche legate alla gestione della qualità del servizio nelle reti stanno infatti assumendo sempre maggior rilevanza; tuttavia gli strumenti di misurazione sviluppati in questo ambito sono pochi ed in grado di fornire valori affidabili solo in limitate circostanze [89]. Utilizzando opportunamente i protocolli di routing e di allocazione delle risorse [90], ASSOLO potrebbe invece essere impiegato con successo per garantire alle applicazioni un servizio stabile e predicibile, aumentare l'efficienza della rete e semplificare la gestione del traffico.

Appendice A

La suite TCP/IP

Data l'enorme complessità delle reti di Telecomunicazioni (TLC), è evidente come un approccio di tipo "monolitico", con la costruzione di un unico componente in grado di fornire tutte le funzionalità necessarie, non possa essere affatto efficiente in termine di semplicità di manutenzione, efficienza e versatilità. La strategia adottata per l'implementazione delle funzionalità necessarie in una rete a commutazione di pacchetto è stata quindi quella della suddivisione dell'architettura di comunicazione in più strati, o livelli. Quello che si è voluto ottenere è una struttura stratificata che costruisce un servizio comunicativo sofisticato operando "per gradi", dove ciascuno degli N livelli che la compongono è incaricato di svolgere un ben preciso insieme di controlli, per implementare funzionalità via via sempre più complesse risalendo lungo la pila. L'invio di un messaggio originato dall'ultimo livello comporta la 'discesa' lungo lo *stack* architetturale e, ad ogni attraversamento di un livello, l'aggiunta (incapsulamento) da parte del livello stesso di informazioni di controllo in testa o in coda al messaggio. La nuova struttura dati che si viene così a creare costituisce il pacchetto che verrà poi trasmesso sui canali. In maniera del tutto analoga, la ricezione di un pacchetto comporta la risalita dello *stack* con l'eliminazione dei relativi incapsulamenti e, infine, la consegna del messaggio contenuto al destinatario. Questa brillante soluzione si basa su due semplici principi alla base dell'ingegneria e della progettazione in generale: scomporre un problema complesso in più sotto-problemi semplici e separare l'implementazione dall'interfaccia, ovvero distinguere il cosa deve essere fatto dal come.

Il modello a strati ha spinto negli anni '70 tutti i principali costruttori (IBM, Digital, Apple, Novell etc) a formalizzare delle architetture di comunicazione proprietarie; parallelamente al loro sviluppo si sono però avuti importanti progetti per la definizione di standard non proprietari, definite invece da organismi internazionali non legati a specifici costruttori. Tra questi ultimi vanno sicuramente citati OSI e TCP/IP.

Il modello Open Systems Interconnection (OSI), schematizzato in Figura A.1, è un'architettura comunicativa formalizzata dall'International Organization for Standardization che prevede 7 distinti livelli: Physical, Data Link Control, Network, Transport, Session, Presentation e Application. Nonostante abbia avuto uno scarso successo, ancora oggi viene utilizzata come modello di riferimento nello studio delle altre architetture.

L'architettura di comunicazione che però si è progressivamente affermata e infine

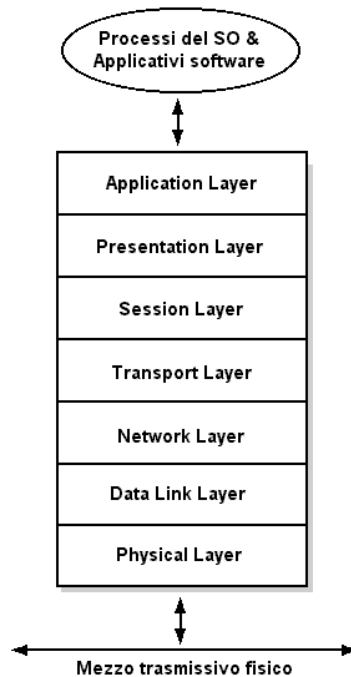


Figura A.1: Stack architetturale ISO-OSI

imposta come standard de facto è la suite TCP/IP, un modello più semplice, basato su un approccio pragmatico e molto più vicino alle reali esigenze di *networking* di quando non fosse il modello OSI. Il suo successo è sicuramente legato alla natura non proprietaria, che svincola l'infrastruttura di rete da specifici fornitori, e all'essenzialità, a volte persino esasperata ma tale da rendere questo protocollo estremamente flessibile ed è in grado di interconnettere reti basate su tecnologie eterogenee. Inoltre occorre considerare come TCP/IP sia riuscito, a differenza di molti altre suite, a guadagnare la fiducia dell'ambiente accademico, venendo adottato nel 1983 dal progetto ARPANET e implementato nello Unix di Berkley (BSD4.3), distribuito in tutto il mondo.

Le specifiche dei protocolli della suite TCP/IP, le cui idee alla base furono per la prima volta pubblicate in un articolo nel 1974 [91], partono dal 3° livello architetturale rispetto al modello di riferimento OSI. In particolare il protocollo IP, che si inserisce a livello Network, consente la costruzione di una rete a commutazione di pacchetto che in linea di principio può utilizzare qualunque Data Link Control (DLC) mentre a livello Transport operano sia TCP che UDP, fornendo tipologie di servizi “duali”; il primo infatti è affidabile ma sofisticato, il secondo è invece più snello ma inaffidabile. Esistono quindi moltissimi altri protocolli di livello applicativo che si appoggiano ai servizi forniti dai sottostanti TCP e UDP e che permettono di interconnettere diversi processi applicativi (Ftp, Http, Telnet, ssh...). Una rete TCP/IP è costituita da un dorsale intermedia (*backbone*) con commutazione dei pacchetti a livello IP: i nodi intermedi di commutazione (*router* IP) utilizzano i livelli fino al 3° mentre i

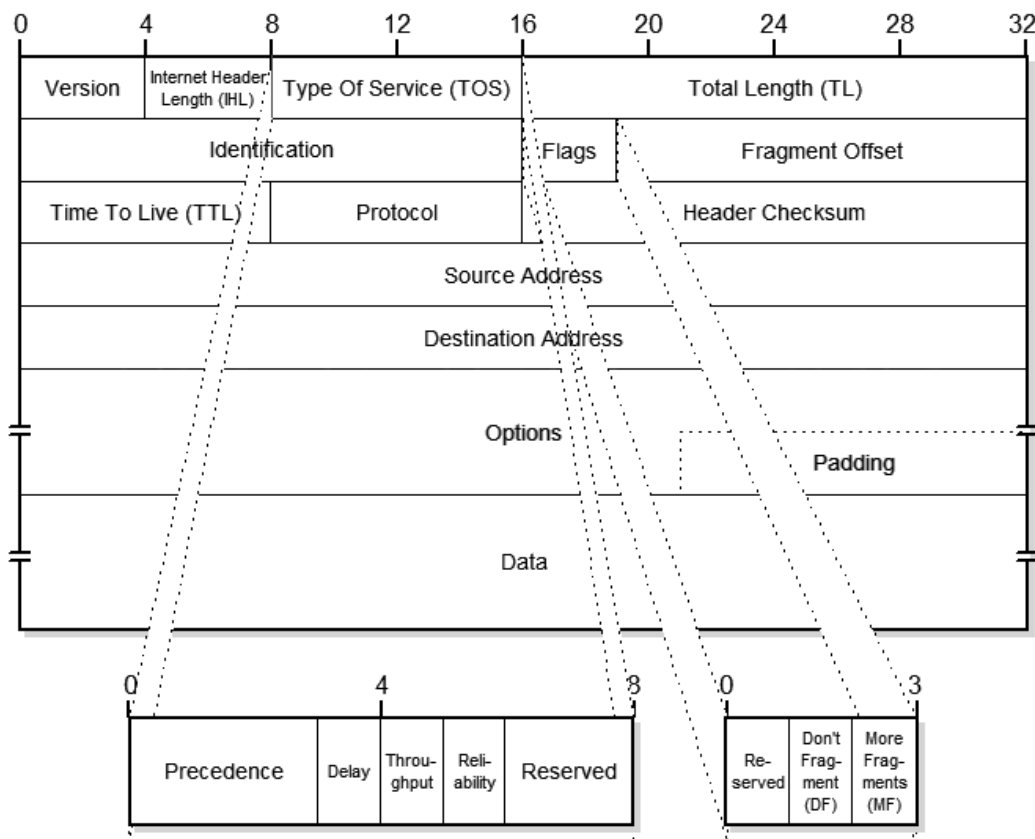


Figura A.2: Formato del *datagram* IP

nodì finali (*host* IP¹) utilizzano tutti i livelli.

A.1 IPv4

Internet Protocol versione 4 [92] appartiene alla famiglia dei protocolli di tipo *connectionless* non confermati e svolge solamente le funzioni di frammentazione e *protocol multiplexing* delle Protocol Data Unit (PDU) provenienti dai livelli superiori, non introducendo invece meccanismi per il controllo di flusso, la consegna in sequenza o il controllo degli errori. Potrebbe sembrare paradossale il fatto che l'intera rete mondiale sia basata su un protocollo così semplice e inaffidabile; in realtà è proprio questa assenza di complessità che permette ad IP di gestire un traffico di una rete così vasta, delegando eventuali controlli o funzionalità aggiuntive ai livelli superiori. La struttura della PDU-DATI (*datagram*) del protocollo Internet Protocol versione 4 (IPv4) è composta da 14 diversi campi, di cui 12 aventi lunghezza fissa, secondo lo schema riportato in Figura A.2.

IPv4 introduce un proprio sistema di indirizzamento a priori, indipendente dall'indirizzamento di livello DLC, prevedendo indirizzi di lunghezza fissa pari a

¹Gli host, in particolare, possono essere di varia natura, dal tradizionale PC desktop sino ai mainframe delle grosse compagnie, dai server di posta ai piccoli dispositivi portatili, palmari o telefoni cellulari, ed avere attivo un qualsiasi processo applicativo che utilizzi la rete

32 bit. Questi indirizzi, espressi in notazione *dotted decimal*², possono essere di tre diversi tipi: *unicast*, *multicast* e *broadcast*. Gli indirizzi *unicast* IPv4 non sono di tipo *flat* bensì sono strutturati su almeno due campi: *network address* e *host address*. Questi indirizzi vengono associati alle singole interfacce dei nodi di rete (e non al nodo) e devono rispettare due regole principali: interfacce adiacenti, ovvero sullo stesso link, devono avere stesso lo stesso *network address* e diversi valori del campo *host address*, mentre interfacce non adiacenti devono avere diversi indirizzi per quanto riguarda la prima parte. Un indirizzo di *multicast* invece ha una struttura piatta, cioè a campo unico, e identifica un gruppo di nodi: un pacchetto inviato verso un indirizzo IPv4 *multicast* viene ricevuto da tutti i nodi che in quel momento appartengono al gruppo in ascolto sull'indirizzo utilizzato. Un messaggio inviato verso un indirizzo di tipo *broadcast* deve essere recapitato a tutti i nodi IPv4 adiacenti (ovviamente non tutta Internet!). L'Internet Protocol versione 4 si appoggia poi a due protocolli di servizio rispettivamente per effettuare la risoluzione degli indirizzi e per la segnalazione di errori e messaggi di controllo tra macchine TCP/IP. La trasmissione di pacchetti tra nodi IPv4 appartenenti allo stesso link presenta un problema ricorrente nel caso in cui il DLC sottostante utilizzi un sistema di indirizzamento. Infatti è necessario determinare a quale indirizzo di livello DLC inviare i dati, "risolvere" cioè l'indirizzo di livello 2 conoscendo l'indirizzo di livello 3 corrispondente. Le tecniche di risoluzione dell'indirizzo sono varie e differiscono in base al tipo di link utilizzato; in particolare si distinguono le tre fondamentali tecniche di risoluzione statica (esiste una tabella definita a priori di corrispondenze tra indirizzi IPv4 e DLC), dinamica (automatismi ricavano all'occorrenza la corrispondenza indirizzo IPv4 - DLC) e algoritmica (l'indirizzo di livello 2 è ricavato a partire dall'indirizzo di livello 3). IPv4 dispone di Address Resolution Protocol (ARP), un protocollo assistente che svolge la funzione di costruzione dinamica delle associazione tra indirizzi IPv4 e DLC. ARP è incluso nell'implementazione di IPv4 e avrebbe poco senso se non utilizzato in combinazione con l'Internet Protocol; tuttavia i due sono concettualmente due protocolli completamente separati e le richieste ARP non sono incapsulate in *datagram* IPv4, bensì sono messaggi separati incapsulati direttamente nelle frame DLC. Internet Control Message Protocol (ICMP) è invece un protocollo di servizio, pensato ad hoc per IPv4, utilizzato per la reportistica e la segnalazione di errori (*error reporting*) alla sorgente dati. A differenza di quanto accade per ARP, i messaggi ICMP vengono incapsulati in IP e a ciascun messaggio è associato un primo codice che ne indica la tipologia e un sotto-codice che specifica il particolare errore riscontrato. Alcuni semplici comandi (es. **ping**, **traceroute**) utilizzati per verificare il funzionamento della rete utilizzano proprio ICMP, inviando particolari pacchetti e verificando quali messaggi ICMP vengono rispediti in risposta.

A.2 TCP/UDP per IPv4

La suite TCP/IP definisce due protocolli di livello Transport sotto molti punti di vista duali: Transmission Control Protocol e User Datagram Protocol. Ciò permette di fornire agli applicativi modalità comunicative su cui appoggiarsi radicalmente diverse, in modo tale che questi possano utilizzare quello migliore in funzione del

²Si indica cioè il valore decimale di ogni byte usando i punti come separatori, ad es. 193.206.71.152

tipo di servizio che devono offrire all'utente finale. La rete di router intermedi, aventi la sola funzione di inoltrare il traffico dal nodo mittente verso il nodo destinatario, è costituita da dispositivi di interconnessione di livello 3 e pertanto, per definizione, non possiede livelli Transport o superiori; questi invece saranno presenti sui nodi finali, sui quali risiedono anche le applicazioni sorgenti e destinatarie del traffico. Sarà quindi compito del nodo destinatario effettuare la multiplazione dei diversi flussi comunicativi dei livelli superiori. A questo proposito è opportuno ricordare che, poiché non esistono in TCP/IP indirizzi di livello Transport, un *layer* TCP o UDP viene identificato solamente tramite l'indirizzo del sottostante livello IP sul nodo e il tipo di protocollo stesso, specificato dal valore del campo Protocol presente nella testata (*header*) IP. Lo stesso problema si ripresenta per il livello applicativo; in questo caso ad ogni processo applicativo non viene associato un indirizzo bensì un numero di porta univoco, sul quale il processo stesso sarà in ascolto per catturare i flussi comunicativi a lui diretti.

A.2.1 User Datagram Protocol

UDP [93] è un protocollo non confermato estremamente sottile, che fornisce cioè ai livelli sovrastanti servizi simili a quanto fornito dal IP³: UDP opera infatti in modalità *connectionless* e non fornisce garanzia circa la consegna del messaggio, ovvero la comunicazione non è affidabile. Quello che viene fornito ai protocolli di livello applicativo è però un'interfaccia di programmazione applicativa, ovvero le meglio note *socket*. Non esistendo in UDP il concetto di sessione comunicativa, quando l'applicazione mittente consegna ad UDP un messaggio, questo provvede immediatamente a generare uno o più *datagram* indipendenti tra loro (*stateless*). Un protocollo di questo genere risulta molto conveniente in situazioni in cui le prestazioni hanno la precedenza sulla qualità: si pensi ad esempio ad una trasmissione audio o video, dove un risultato fluido ma con qualche imperfezione è preferibile rispetto ad un'immagine o audio perfetti ma non fruibili perché troppo lenti.

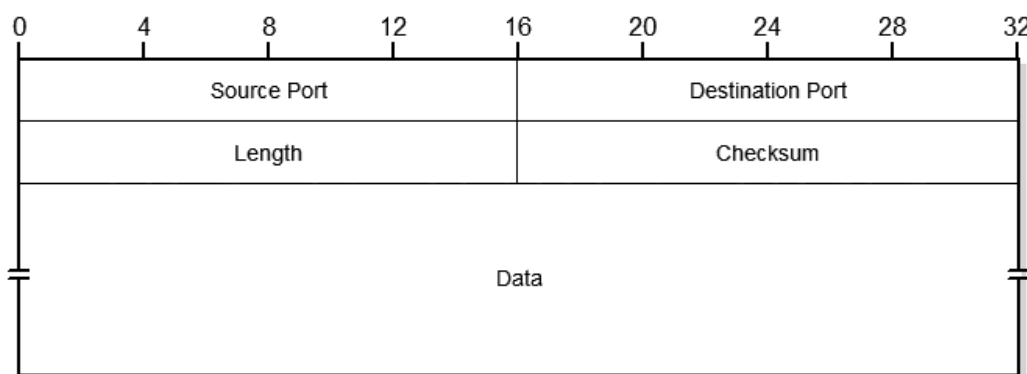


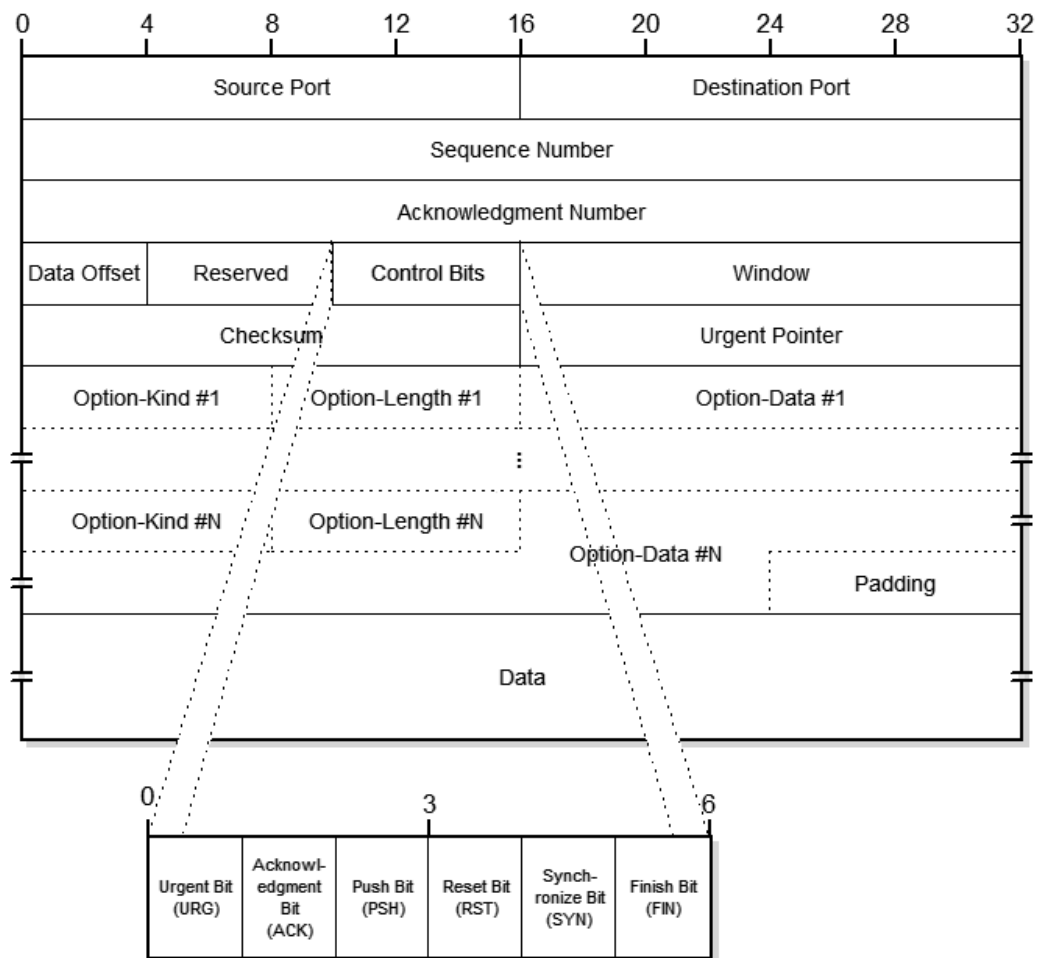
Figura A.3: Formato del *datagram* UDP

³Introducendo quindi un *overhead* computazionale decisamente contenuto

A.2.2 Transmission Control Protocol

Il protocollo TCP è stato standardizzato per la prima volta con [94] e perfezionato quindi con [75]. Esso si appoggia su servizi di livello 3 ritenuti inaffidabili come IPv4 ed ovvia alle carenze dei protocolli sottostanti implementando tutte le funzionalità necessarie per rendere la comunicazione affidabile. TCP è quindi, a differenza di UDP, un protocollo *connection-oriented* (*stateful*): esiste cioè il concetto di sessione comunicativa e il trasferimento di dati applicativi, la cui consegna in sequenza è garantita, può avvenire solo a patto che sia stata instaurata una connection TCP tra le due entità comunicanti e che questa sia attiva (non sia cioè ancora stata abbattuta). Inoltre è un protocollo confermato che adotta uno schema di controllo di flusso a finestra di messaggi ibrido (Selective-Repeat in condizioni normali, Go-Back-n in caso di errori) prevedendo PDU-ACK e *time-out* di ritrasmissione. A differenza di UDP, TCP non trasferisce singoli messaggi applicativi bensì sequenze di byte; il processo applicativo passa cioè al Transmission Control Protocol uno o più messaggi che il TCP tratta come un'unica sequenza di byte, suddividendola quindi a propria discrezione in blocchi⁴ e costruendo poi le PDU (*segment* TCP) che verranno passate all'IP. Negli anni numerosi RFC sono stati inoltre rilasciati al fine di proporre nuovi e più efficienti algoritmi di funzionamento, specialmente in situazioni di congestione della rete o di perdita di dati trasmessi. Si può quindi intuire come il TCP sia un protocollo complesso ma affidabile, adatto in tutte quelle situazioni in cui le prestazioni passano in secondo piano rispetto all'integrità dei dati ricevuti (es. trasferimento di file, e-mail...).

⁴Avanti comunque una dimensione massima, detta MSS

Figura A.4: Formato del *segment* TCP

Appendice B

La rete GARR

A differenza di quanto comunemente si pensa, Internet non è un'unica rete bensì un insieme di migliaia di reti di computer interconnesse attraverso canali in fibra ottica, cavi elettrici e collegamenti wireless. Si considerino ad esempio le reti dei centri di ricerca e delle università italiane: tutti i calcolatori presenti al loro interno sono interconnessi attraverso la dorsale di rete nazionale GARR, a sua volta unita con le altre Reti per la Ricerca europee e quelle mondiali tramite un collegamento alla rete GÉANT2. Affinché il resto dell'Internet "commerciale" sia raggiungibile, la rete nazionale dispone inoltre di connessioni fornite da *provider* mondiali (Global Crossing, Telia e Level3) e collegamenti di *peering* con le principali reti commerciali italiane (Telecom Italia, Eutelia, Fastweb, ItGate, etc) e la pubblica amministrazione centrale. La dorsale GARR però funge a sua volta da rete preferenziale di accesso ad Internet per la Santa Sede e, viceversa, consente di sfruttare l'infrastruttura italiana per accedere a materiali di grande interesse culturale (si pensi alle università o alle biblioteche vaticane) da tutto il mondo.

Data quindi l'enorme complessità ed eterogeneità delle reti che costituiscono Internet, è facile intuire quanto sia importante poter disporre di strumenti di stima *end-to-end* che non richiedano alcuna conoscenza a priori del sistema.

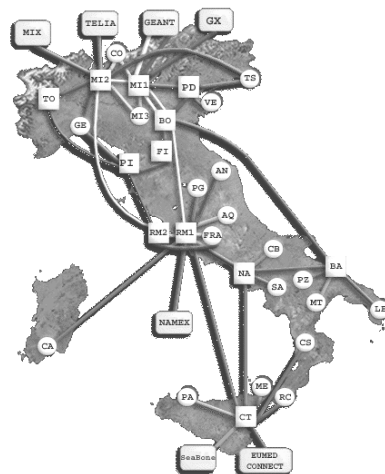


Figura B.1: Mappatura della rete GARR

Appendice C

Codice sorgente

Questa appendice raccoglie l'implementazione dell'algoritmo di pathChirp in pseudo-C e le modifiche apportate, durante il lavoro di tesi, ai sorgenti originali della versione 2.4.1.

C.1 pseudocodice.c



Le procedure implementate in *pseudocodice.c* descrivono il funzionamento dell'algoritmo di segmentazione e calcolo della stima utilizzato da pathChirp. Per maggiori dettagli si può fare riferimento all'articolo originale su pathChirp [58] o alla pagina web del software <http://www.spin.rice.edu/Software/pathChirp/>.

```
1 procedure stima_D(q) {
2   for (k = 2 to N)
3     E(k) = 0;
4   i = 1;
5   l = N;
6   while (i <= N - 1) {
7     if (q(i) < q(i+1)) {
8       j = excursion(q, i, F, L)
9       choose case(j):
10        Case(a): (j > i) and (j < N)
11          for (s = i to j - 1)
12            if (q(s) < q(s+1))
13              E(s) = R(s);
14        Case(b): j = N
15          for (s = i to N)
16            E(s) = R(i);
17        l = i;
18        if (j = i)
19          j = j + 1;
20        i = j;
21      } else {
22        i = i + 1;
23      }
```

```

24     }
25     D = 0;
26     for (i = 2 to N) {
27         if (E(i) == 0)
28             D += R(1)*delta(i);
29         else
30             D += E(i)*delta(i);
31     }
32     D = D/sum(2, N, delta);
33     return D;
34 }
35
36
37 procedure escursion(q, i, F, L) {
38     maximum_q=0;
39     while((j < N) & (q(j) - q(i) > maximum_q/F)) {
40         j = j + 1;
41         maximum_q = max(maximum_q, q(j) - q(i));
42     }
43     if (j >= N)
44         return N;
45     if (j - i >= L)
46         return j;
47     else
48         return i;
49 }
50
51
52 procedure sum(a, b, delta) {
53     sum = 0;
54     for (i = a to b) {
55         sum += delta(i);
56     }
57     return sum;
58 }

```

C.2 computeinstbwexcursion.c



La procedura fondamentale per la determinazione della stima della banda associata ad un chirp è stata corretta per risolvere un problema di sovrastima noto in letteratura. Le linee di codice in *computeinstbwexcursion.c* con accanto un commento sono state modificate rispetto all'originale prima di venir incluse in ASSOLO.

```

1  /* Calcola la banda istantanea relativa ad un chirp
2
3     Algoritmo corretto in ASSOLO per evitare
4     il problema della sovrastima

```

```

5  */
6
7  double compute_inst_bw_excursion()
8  {
9      int cur_loc=1; /* ASSOLO */
10     int cur_inc=1; /* ASSOLO */
11     int count;
12     double inst_bw=0.0,sum_iat=0.0,max_q=0.0;
13
14     memset(av_bw_per_pkt, 0, (int)(num_interarrival) * sizeof(
        double));
15
16     while(qing_delay[cur_inc]>=qing_delay[cur_inc+1] &&
        cur_inc<max_good_pkt_this_chirp)
17         cur_inc++;
18
19     cur_loc=cur_inc+1;
20
21     while(cur_loc<=max_good_pkt_this_chirp) {
22         if (qing_delay[cur_loc]>NEG_THRESH+1.0) {
23
24             if (max_q<(qing_delay[cur_loc]-qing_delay[cur_inc]))
25                 max_q=qing_delay[cur_loc]-qing_delay[cur_inc];
26
27             if (qing_delay[cur_loc]-qing_delay[cur_inc]<(max_q/
                decrease_factor)) {
28                 if (cur_loc-cur_inc>=busy_period_thresh) {
29                     for (count=cur_inc;count<=cur_loc-1;count++) {
30                         if (qing_delay[count]<qing_delay[count+1])
31                             av_bw_per_pkt[count]=rates[count];
32                     }
33                 }
34                 cur_inc=cur_loc;
35                 while(qing_delay[cur_inc]>=qing_delay[cur_inc+1] &&
                    cur_inc<max_good_pkt_this_chirp)
36                     cur_inc++;
37                 cur_loc=cur_inc;
38                 max_q=0.0;
39             }
40             cur_loc++;
41         }
42
43         /* Linee eliminate in ASSOLO
44         if (cur_inc==max_good_pkt_this_chirp)
45             cur_inc--; */
46

```

```
47   for (count=cur_inc; count<=max_good_pkt_this_chirp; count++)
48       {
49       av_bw_per_pkt [ count ] = rates [ cur_inc - 1 ];    /* ASSOLO */
50   }
51   for (count=0; count<=max_good_pkt_this_chirp; count++) {
52       if (av_bw_per_pkt [ count ] < 1.0)
53           av_bw_per_pkt [ count ] = rates [ cur_inc - 1 ];    /* ASSOLO */
54       sum_iat += iat [ count ];
55       inst_bw += av_bw_per_pkt [ count ] * iat [ count ];
56   }
57
58   inst_bw = inst_bw / sum_iat ;
59   return (inst_bw) ;
60
61 }
```

Bibliografia

- [1] L. Kleinrock. *Information Flow in Large Communication Nets - RLE Quarterly Progress Report*. PhD thesis, Massachusetts Institute of Technology, 1961.
- [2] P. Baran. *On Distributed Communications*. The Rand Corporation, Memorandum RM-3420-PR, 1964.
- [3] Hui Zhou, Yongji Wang, Xiuli Wang, and Xiaoyong Huai. Difficulties in estimating available bandwidth. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 2, pages 704–709, June 2006.
- [4] National Research Council. *Internet Under Crisis Conditions: Learning from September 11*. National Academy Press, 2003.
- [5] Janet Abbate. *From ARPANET to INTERNET: a history of ARPA-sponsored computer networks, 1966–1988*. Thesis (ph. d.), University of Pennsylvania, Philadelphia, PA, USA, 1994.
- [6] R. Popescu-Zeletin. Implementing the iso-osi reference model. In *SIGCOMM '83: Proceedings of the eighth symposium on Data communications*, pages 56–66, New York, NY, USA, 1983. ACM Press.
- [7] B. Schott, Alexander Clemm, and Ulf Hollberg. An iso/osi based approach for modeling heterogeneous networks. In Martti Tienari and Dipak Khakhar, editors, *INDC*, volume C-6 of *IFIP Transactions*, pages 377–389. North-Holland, 1992.
- [8] Andreas Johnsson. *Modeling, Implementation and Evaluation of IP Bandwidth Measurement Methods*. PhD thesis, Department of Computer Science and Electronics, Mälardalen University, 2007.
- [9] Nan Ni and Laxmi Narayan Bhuyan. Fair scheduling in internet routers. *IEEE Transactions on Computers*, 51(6):686–701, 2002.
- [10] K. B. Kim and S. H. Low. Design and analysis of AQM for stabilizing TCP. In *Proceedings of American Control Conference*, pages 260–265, Denver, CO, June 2003.
- [11] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, 1993.

- [12] Jacob Strauss, Dina Katabi, and Frans Kaashoek. A measurement study of available bandwidth estimation tools. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 39–44, New York, NY, USA, 2003. ACM Press.
- [13] S. Bellovin. A best-case network performance model, 1992.
- [14] Van Jacobson. pathchar a tool to infer characteristics of internet paths. Presented at the Mathematical Sciences Research Institute, 1997.
- [15] M. Turrubiarres, D. Torres, M. Angulo, and D. Munoz. Analysis of IP network path capacity estimation using a variable packet size method. In *Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Proceedings. 15th International Conference on*, pages 177–182, February 2005.
- [16] Robert B Cooper. *Introduction to queueing theory*. North Holland, 1981.
- [17] Ivo Adan and Jacques Resing. *Queueing Theory*. Department of Mathematics and Computing Science Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, 2001.
- [18] G.D. Cole. Computer network measurements: techniques and experiments. Technical report, University of California, Los Angeles, October 1971.
- [19] L. Kleinrock and W.E. Naylor. On measured behavior of the arpa network. In AFIPS Press, editor, *National Computer Conference*, volume 43, pages 767–780, 1974.
- [20] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP Performance Metrics. RFC 2330 (Informational), May 1998.
- [21] J. Mahdavi and V. Paxson. IPPM Metrics for Measuring Connectivity. RFC 2678 (Proposed Standard), September 1999.
- [22] Nevil Brownlee and Chris Loosley. Fundamentals of internet measurement: A tutorial. CMG Journal of Computer Resource Management, Issue 201, 2001.
- [23] Bruce B. Lowekamp, Brian Tierney, Les Cottrell, Richard Hughes-Jones, Thilo Kielmann, , and Martin Swany. A hierarchy of network performance characteristics for grid applications and services. Global Grid Forum Proposed Recommendation, May 2004.
- [24] Tobias Oetiker. Mrtg - the multi router traffic grapher. In *LISA '98: Proceedings of the 12th USENIX conference on System administration*, pages 141–148, Berkeley, CA, USA, 1998. USENIX Association.
- [25] A. Pasztor. *Accurate Active Measurement in the Internet and its Applications*. PhD thesis, Department of Electrical and Electronic Engineering, The University of Melbourne, Feb 2003.

- [26] A. Zanella, G. Procissi, M. Gerla, and M. Y. Sanadidi. TCP westwood: analytic model and performance evaluation. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 3, pages 1703–1707, San Antonio, TX, USA, 2001.
- [27] M. Gerla, M. Y. Sanadidi, Ren Wang, A. Zanella, C. Casetti, and S. Mascolo. TCP westwood: congestion window control using bandwidth estimation. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 3, pages 1698–1702, San Antonio, TX, USA, 2001.
- [28] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang. Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 287–297, New York, NY, USA, 2001. ACM Press.
- [29] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli. Performance evaluation of westwood+ tcp congestion control. *Perform. Eval.*, 55(1-2):93–111, 2004.
- [30] L. A. Grieco and S. Mascolo. Mathematical analysis of westwood+TCP congestion control. *IEE Proceedings - Control Theory and Applications*, 152(1):35–42, January 2005.
- [31] R. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth estimation: Metrics, measurement techniques, and tools, 2003.
- [32] Ravi Prasad, Manish Jain, and Constantinos Dovrolis. Effects of interrupt coalescence on network measurements. In *Passive and Active Network Measurement*, volume Volume 3015/2004, pages 247–256. Springer Berlin / Heidelberg, 2004.
- [33] M. Mathis and M. Allman. A Framework for Defining Empirical Bulk Transfer Capacity Metrics. RFC 3148 (Informational), July 2001.
- [34] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), April 1999. Updated by RFC 3390.
- [35] S. Floyd and T. Henderson. The NewReno Modification to TCP’s Fast Recovery Algorithm. RFC 2582 (Experimental), April 1999. Obsoleted by RFC 3782.
- [36] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP’s Fast Recovery Algorithm. RFC 3782 (Proposed Standard), April 2004.
- [37] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001 (Proposed Standard), January 1997. Obsoleted by RFC 2581.
- [38] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018 (Proposed Standard), October 1996.
- [39] Reiner Ludwig. The eifel algorithm for tcp rfc, draft 0, February 2001.

- [40] Lawrence S. Brakmo, Sean W. O'Malley, and Larry L. Peterson. Tcp vegas: new techniques for congestion detection and avoidance. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 24–35, New York, NY, USA, 1994. ACM Press.
- [41] L. S. Brakmo and L. L. Peterson. TCP vegas: end to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480, October 1995.
- [42] V. Jacobson. Congestion avoidance and control. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 314–329, New York, NY, USA, 1988. ACM Press.
- [43] Anurag Acharya and Joel Saltz. A study of internet round-trip delay. Technical Report CS-TR-3736, UM Computer Science Department, dec 1996.
- [44] Srinivas Shakkottai, R. Srikant, Nevil Brownlee, Andre Broido, and kc claffy. The rtt distribution of tcp flows on the internet and its impact on tcp based flow control. Technical report, Cooperative Association for Internet Data Analysis - CAIDA, feb 2002.
- [45] Jean-Chrysotome Bolot. End-to-end packet delay and loss behavior in the internet. In *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*, pages 289–298, New York, NY, USA, 1993. ACM Press.
- [46] S. Keshav. *Congestion Control in Computer Networks*. PhD thesis, UC Berkeley, sep 1991.
- [47] S. Keshav. A control-theoretic approach to flow control. *SIGCOMM Comput. Commun. Rev.*, 25(1):188–201, 1995.
- [48] Vern Paxson. End-to-end internet packet dynamics. In *SIGCOMM '97: Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 139–152, New York, NY, USA, 1997. ACM Press.
- [49] Robert Carter and Mark Crovella. Measuring bottleneck link speed in packet-switched networks. Technical report, Boston University, Boston, MA, USA, 1996.
- [50] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 905–914, Anchorage, AK, April 2001.
- [51] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk. Multifractal cross-traffic estimation. In *Proc. of ITC Specialist Seminar on IP Traffic Measurement*, September 2000.

- [52] Ningning Hu and Peter Steenkiste. Estimating available bandwidth using packet pair probing. Technical Report CMU-CS-02-166, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, Sep 2002.
- [53] Ningning Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications*, 21(6):879–894, August 2003.
- [54] B. Melander, M. Bjorkman, and P. Gunningberg. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, volume 1, pages 415–420, San Francisco, CA, USA, 2000.
- [55] Andreas Johnsson, Bob Melander, and Mats Björkman. Diettopp: A first implementation and evaluation of a simplified bandwidth measurement method. In *Second Swedish National Computer Networking Workshop*, page 5, Karlstad, November 2004.
- [56] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput. *IEEE/ACM Trans. Netw.*, 11(4):537–549, 2003.
- [57] Manish Jain and Constantinos Dovrolis. End-to-end estimation of the available bandwidth variation range. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 265–276, New York, NY, USA, 2005. ACM Press.
- [58] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. pathchirp: Efficient available bandwidth estimation for network paths, 2003. Passive and Active Measurement Workshop (2003).
- [59] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman. Real-time measurement of end-to-end available bandwidth using kalman filtering. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 73–84, 2006.
- [60] E. Hartikainen and S. Ekelin. Tuning the temporal characteristics of a kalman-filter method for end-to-end bandwidth estimation. In *End-to-End Monitoring Techniques and Services, 2006 4th IEEE/IFIP Workshop on*, pages 58–65, 03-03 April 2006.
- [61] Manish Jain and Constantinos Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 272–277, New York, NY, USA, 2004. ACM Press.
- [62] Andras Veres and Miklos Boda. The chaotic nature of TCP congestion control. In *INFOCOM (3)*, pages 1715–1723, 2000.
- [63] V. Paxson and S. Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

- [64] Matthias Grossglauser and Jean-Chrysostome Bolot. On the relevance of long-range dependence in network traffic. *IEEE/ACM Trans. Netw.*, 7(5):629–640, 1999.
- [65] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.
- [66] Daniel R. Figueiredo, Benyuan Liu, Anja Feldmann, Vishal Misra, Don Towsley, and Walter Willinger. On tcp and self-similar traffic. *Perform. Eval.*, 61(2-3):129–141, 2005.
- [67] M. Jain, R. Prasad, and C. Dovrolis. The tcp bandwidth-delay product revisited: Network buffering, cross traffic, and socket buffer auto-sizing. Technical Report GIT-CERCS-03-02, Georgia Tech Institute, Feb 2003. Available at <http://www.cerics.gatech.edu/tech-reports/>.
- [68] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. NLANR/DAST : Iperf - the TCP/UDP bandwidth measurement tool. <http://dast.nlanr.net/Projects/Iperf/>.
- [69] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, and K. Claffy M. Fomenkov. Comparison of public end-to-end bandwidth estimation tools on high-speed links. In *Proceedings of 6th Passive and Active Measurements Workshop, PAM*, Boston, San Diego Supercomputer Center, mar 2005.
- [70] Carlos Ubeda Castellanos, Dimas Lopez Villa, Oumer M. Teyeb, Jan Elling, and Jeroen Wigard. Comparison of available bandwidth estimation techniques in packet-switched mobile networks. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1–5, Helsinki,, September 2006.
- [71] Leopoldo Angrisani, Salvatore D’Antonio, Marcello Esposito, and Michele Vadursi. Techniques for available bandwidth measurement in ip networks: a performance comparison. *Comput. Networks*, 50(3):332–349, 2006.
- [72] Yu-Chen Huang, Chun-Shien Lu, and Hsiao-Kuang Wu. Queuing delay propagation model (qdpm)-based queuing region determination for available bandwidth estimation of multimedia qos. Technical Report TR-IIS-05-004, Institute of Information Science, Academia Sinica, Taipei, Taiwan, 2005.
- [73] A. A. Ali, F. Michaut, and F. Lepage. End-to-end available bandwidth measurement tools : A comparative evaluation of performances. *ArXiv e-prints*, 706:1–13, June 2007.
- [74] M. Murray, S. Smallen, O. Khalili, and M. Swany. Comparison of end-to-end bandwidth measurement tools on the 10gige teragrid backbone. In *Grid Computing, 2005. The 6th IEEE/ACM International Workshop on*, November 2005.

- [75] J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFC 3168.
- [76] Jiri Navratil and R. Les. Cottrell. Abwe: A practical approach to available bandwidth navratil, 2003. Passive and Active Measurement Workshop (2003).
- [77] Joel Sommers, Paul Barford, and Walter Willinger. A proposed framework for calibration of available bandwidth estimation tools. *iscc*, 0:709–718, 2006.
- [78] J.J. et al. Filliben. Engineering statistics handbook. 2002.
- [79] Tushar S. Chande. Adapting moving averages to market volatility. *Stocks & Commodities*, 10:3:108–114, 1992.
- [80] Tushar S. Chande and Stanley Kroll. *The New Technical Trader: Boost Your Profit by Plugging into the Latest Indicators (Wiley Finance)*. John Wiley & Sons Inc, 1994.
- [81] Tushar S. Chande. Identifying powerful breakouts early. *Stocks & Commodities*, 13:418–424, 1995.
- [82] Adam White. The vertical horizontal filter. *Futures Magazine*, Volume XX, No. 10:1–10, 1991.
- [83] Lars Burgstahler and Martin Neubauer. New modifications of the exponential moving average algorithm for bandwidth estimation. In *15th ITC Specialist Seminar, Würzburg, Germany*, jul 2002.
- [84] Minkyong Kim and Brian Noble. Mobile network estimation. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 298–309, New York, NY, USA, 2001. ACM Press.
- [85] V. Katkovnik, K. Egiazarian, and J. Astola. Median filter with varying bandwidth adaptive to unknown smoothness of the signal. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 3, pages 519–522, Geneva, Switzerland, 2000.
- [86] Otto H Schmitt. A thermionic trigger. *Journal of Scientific Instruments*, 15(1):24–26, 1938.
- [87] Nitta Kazumasa, Ueda Takashi, Yoshiyuki Yasuda, and Syunichi Kido. Mode switching method for mobile station, September 1999.
- [88] Luca De Cicco, Saverio Mascolo, and Vittorio Palmisano. An experimental investigation of the congestion control used by skype voip. In Fernando Boavida, Edmundo Monteiro, Saverio Mascolo, and Yevgeni Koucheryavy, editors, *Wired/Wireless Internet Communications, 5th International Conference, WWIC 2007, Coimbra, Portugal, May 23-25, 2007, Proceedings*, volume 4517 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2007.

- [89] Llorenç Cerdà, Michael Voorhaen, Rafael Guimarnes, José-M Barcelò, Jorge García, and Chris Blondia. A reservation scheme satisfying bandwidth QoS constraints for ad-hoc networks. In *Wireless Systems and Mobility in Next Generation Internet*, volume Volume 3427/2005, pages 176–188. Springer Berlin / Heidelberg, 2005.
- [90] Li Xiao, King-Shan Lui, Jun Wang, and Klara Nahrsted. Qos extension to bgp. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 100–109, Washington, DC, USA, 2002. IEEE Computer Society.
- [91] V. Cerf and R. Kahn. A protocol for packet network intercommunication. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 22(5):637–648, May 1974.
- [92] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.
- [93] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.
- [94] J. Postel. DoD standard Transmission Control Protocol. RFC 761, January 1980.