MAIS
Multichannel Adaptive Information Systems

Fondo investimenti per la Ricerca di Base (FIRB)
Programma Strategico Tecnologie abilitanti per la Società della Conoscenza - ICT

Modelli e tecniche per l'adattatività del livello di rete

Giuseppe Di Battista, Paolo Giacomazzi, Federico Mariani, Maurizio Patrignani, Maurizio Pizzonia, Gabriella Saddemi, Giacomo Verticale

Data	5/11/2004
Tipo prodotto	Rapporto Tecnico
Stato	Versione definitiva
Unità responsabile	Roma Tre
Unità coinvolte	Roma Tre, Polimi
Autore da contattare	Maurizio Pizzonia

Sommario

Executive Summary

In questo documento sono presentati alcuni risultati relativi al supporto per reti adattative nel contesto del livello OSI 3 (livello *rete*). I risultati presentati contribuiscono allo sviluppo di tecniche per reti adattative sia gestite da una singola organizzazione (intra-domain) sia nello sviluppo di tecniche adattative per l'Internetworking (inter-domain).

1. Introduzione

Il ruolo della rete per i moderni software applicativi è di primaria importanza. Molti di essi infatti prevedono l'interazione tra applicazioni remote nel ruolo di client, server o peer. Con l'evolversi delle tecnologie e con il diffondersi di diversi canali di comunicazione, il trasferimento di dati multimediali e stimolati dall'interazione della macchina con l'uomo sta diventando sempre più una necessità.

La multimedialità interattiva ha presto spinto le esigenze al di là dei limiti di scalabilità delle attuali infrastrutture di rete. Se da un lato e possibile fa sì che l'applicazione si adatti ai limiti della rete (supponendo che questi si possano conoscere facilmente) più complessa è la situazione in cui la rete si possa adattare alle esigenze dell'applicazione. Lo scenario che stiamo delineando prevede che l'applicazione faccia richiesta alla rete di servizi con certi standard qualitativi (*Quality of Service, QoS*). Si devono cioè introdurre classi di servizi che vengano in qualche modo privilegiati dall'infrastruttura di rete, dando così luogo ad un mercato di connettività, parallelo all'attuale connettività best-effort, a che assicuri la consegna dei dati a destinazione entro certi parametri di qualità quali limitato ritardo, sufficiente banda, alta affidabilità del servizio, ecc.

Nell'ambito del progetto MAIS si studiano tematiche relative alle reti adattative a tutti i livelli dello standard OSI. Ciascun livello ha infatti problematiche proprie. L'obiettivo di questo rapporto è di analizzare tematiche relative al livello OSI 3 (detto livello di *rete*).

Il ruolo del livello 3 è tradizionalmente quello di rendere le tecnologie scalabili oltre i limiti delle tecnologie del sottostante livello OSI 2 (detto livello *data-link*). Infatti, le tecnologie di livello data-link attualmente disponibili permettono di connettere un certo numero di stazioni (nell'ordine di alcune migliaia al più) all'interno di un area ristretta (*Local Area Network, LAN*) oppure di connettere 2 stazioni poste a enormi distanze, ad esempio in continenti diversi (*Wide Area Network*, *WAN*). Tutto ciò non è sufficiente per una connettività globale come quella offerta dalla rete Internet. Il livello di rete permette di interconnettere tecnologie di livello data-link tra di loro in modo da raggiungere estensioni della rete molto più vaste.

Per quanto riguarda la qualità del servizio offerto, mentre a livello data-link le problematiche sono per lo più di tipo tecnologico, a livello di rete comprendono aspetti di coordinamento dei vari apparati, aspetti organizzativi e aspetti gestionali. Nel contesto di una rete gestita da una singola organizzazione è importante essere in grado di allocare risorse in base alle richieste degli utenti eventualmente rifiutando (admission control) che non possono essere soddisfatte. Per questo è importante poter modellare il traffico, eventualmente in senso statistico, al fine di prevedere la qualità del servizio che la rete dell'organizzazione è in grado di offrire.

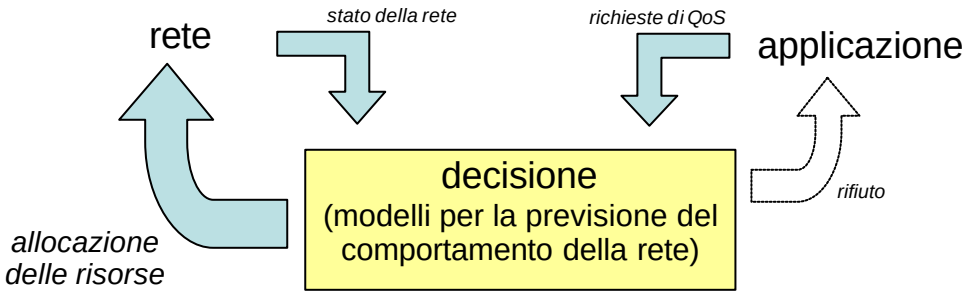
Se ora spostiamo l'attenzione alla rete Internet, la gestione dell'enorme numero di apparecchiature coinvolte non è sotto una unica organizzazione ma un gran numero di operatori competono (nel senso

economico del termine) e cooperano (in senso sia economico che tecnologico) per offrire servizi di connettività all'utente finale.

In questo scenario, a fianco a problemi tecnologici di interoperabilità e coordinamento tra le varie apparecchiature e con le varie tecnologie di livello data-link, compare anche un problema di coordinamento con altre organizzazioni. Ecco perché, a oggi, il traffico su Internet è tutto relativo ad una sola classe di servizio: *best-effort*, cioè con nessuna garanzia sulla affidabilità, sul ritardo o su altri parametri di qualità. Infatti la classe di servizio best-effort è quella più facile da gestire in un ambiente così disomogeneo dal punto di vista tecnologico ed organizzativo.

La domanda di servizi interattivi multimediali e multicanale stanno creando un nuovo mercato che può rendere vantaggioso l'overhead tecnologico ed organizzativo necessario per fornire altre classi di servizi con qualità migliore che best-effort.

Una rete adattativa deve essere in grado di modificare il proprio comportamento in base alle richieste delle applicazioni. Nell'ambito del progetto MAIS si considerano richieste relative a servizi con un certa qualità specificata a livello applicazione. L'obiettivo della rete adattativa in questo contesto è quello di "autoconfigurarsi" in modo da garantire il livello di qualità richiesto. L'attività di autoconfigurazione viene svolta in base alle richieste dell'applicazione, allo stato interno della rete e in base a modelli, del traffico e del comportamento della rete, che sono in qualche modo interni al processo di autoconfigurazione. Tale processo può essere considerato come un processo decisionale il cui output è un adattamento della rete o eventualmente una segnalazione all'applicazione della impossibilità di erogare il servizio con la qualità richiesta. Il processo è rappresentato nella seguente figura.



Varie sono le direzioni in cui la ricerca sta studiando, o ha già prodotto dei risultati, per realizzare reti adattative:

- lo sviluppato di meccanismi di segnalazione e allocazione di risorse che permettono di configurare automaticamente le apparecchiature in modo da garantire certi comportamenti per certe classi di traffico
- lo sviluppo di meccanismi per ottenere dati dalla rete sulle risorse disponibili al fine di alimentare i processi per l'adattatività

– lo studio di tecniche per l'analisi delle prestazioni al fine di poter dedurre garanzie sulla qualità offerta da certe configurazioni di rete e a certi modelli di traffico.

Il primo punto non è un problema dal punto di vista teorico e già esistono protocolli standardizzati per allocare risorse. Per gli altri due problemi molti sono gli studi in corso e i lavori illustrati da questo rapporto contribuiscono in questi ultimi due campi.

Questo rapporto è strutturato come segue. Nella sezione 2 viene presentato uno studio basato su metodi probabilistici per la stima della frazione di traffico che viene servita con un ritardo superiore ad una certa soglia. Nella sezione 3 viene mostrato un modello per le relazioni commerciali tra autonomous systems. Nella sezione 4 viene mostrato un semplice modello architetturale per una rete adattativa inter-domain basata sulla conoscenza della rete stessa. Nella sezione 5 si introduce un modello per l'inferenza di relazioni di tipo commerciale tra autonomous systems, le tecniche presentate sono tratte da (G. Di Battista, M. Patrignani, M. Pizzonia, 2003). Nella sezione 6 viene mostrato il tool BGPlay per la visualizzazione del routing a livello inter-domain, tale tool è dettagliatamente descritto in (G. Di Battista, F. Mariani, M. Patrignani, M. Pizzonia, 2003).

2. Qualità del Servizio nelle reti a pacchetto attraverso inviluppi di traffico statistico

Nelle reti a pacchetto i requisiti di Qualità di Servizio (QoS) possono essere espressi in modo deterministico o in modo statistico. Nel primo caso si richiede che tutti i pacchetti di un dato flusso non superino un preassegnato valore limite di ritardo end-to-end e nessun pacchetto sia scartato nella rete. In tal modo le prestazioni ottenute sono le migliori possibili, ma, al tempo stesso, si ha un uso inefficiente delle risorse di rete.

Nel caso statistico le prestazioni di QoS possono essere indicate mediante la probabilità di violazione della soglia dei ritardi.

Permettendo che una percentuale del traffico violi la QoS richiesta, si può ottenere un sensibile guadagno nella multiplazione statistica con un aumento del grado di utilizzazione dei collegamenti.

Nell'approccio statistico il traffico entrante nella rete deve essere modellato, ovvero occorre conoscere le caratteristiche statistiche delle sorgenti di traffico e la correlazione statistica dei flussi considerati.

Poiché non è sempre possibile avere una caratterizzazione valida delle sorgenti di traffico, la ricerca in corso sulla QoS su base statistica è stata affrontata senza fare riferimento a specifici modelli di sorgente. In particolare, questo lavoro è basato sulla seguente ipotesi: flussi statisticamente indipendenti e ogni flusso limitato da un regolatore deterministico, per esempio, di tipo leaky bucket. Un traffico che soddisfa queste ipotesi viene detto *regulated adversarial traffic*.

In letteratura sono disponibili molti lavori (C. Li e E. W. Knightly, 2002),(J. Qiu e E. Knightly, 1999) sul calcolo della probabilità di violazione della soglia di ritardo; in particolare molti di essi fanno riferimento all'inviluppo di traffico in ingresso deterministico e altri a quello statistico (J. Qiu e E. Knightly, 1999).
Questo documento segue il calcolo della probabilità di violazione della soglia di ritardo secondo la trattazione proposta da E. W. Knightly (J. Qiu e E. Knightly, 1999)(E. Knightly, 1997)(E. Knightly, 1998).
Il metodo di Knightly presenta, però, una forte limitazione in quanto presuppone che sia soddisfatta l'ipotesi di gaussianità del traffico multiplato in ingresso al nodo della rete. Nel documento vengono evidenziati i vantaggi ed i limiti del metodo in relazione al grado di approssimazione del traffico in ingresso alla rete ad un processo gaussiano.
L'obiettivo del documento è di evidenziare le potenzialità offerte dall'uso di metodi statistici per una migliore allocazione delle risorse. Tale metodo può essere usato per stimare la banda necessaria a soddisfare la QoS richiesta dai flussi di traffico, indipendentemente dal numero di sorgenti ma solo sulla base della caratteristiche dell'inviluppo statistico del traffico offerto. Esempi di applicazione sono: la funzione di *Flow Admission Control* dei flussi appartenenti a classi di traffico diverse e i protocolli di instradamento dei flussi con QoS.

2.1. Inviluppi di traffico

Data una *sorgente* che genera traffico secondo un processo stocastico stazionario ed ergodico, definiamo $X(t_1,t_1+t)$ il traffico generato dalla sorgente nell'intervallo di tempo $[t_1,t_1+t]$, essendo $X(t_1,t_1+t)$ una variabile casuale reale non-negativa definita nell'intervallo [0,8). Nel seguito si userà la semplice notazione $X(t)$ per indicare $X(t)=X(t_1,t_1+t)$.

Definizione 1. Dicesi *Inviluppo di traffico deterministico* la funzione *b(t)*, continua non-decrescente e non-negativa, che soddisfi la seguente relazione: $X(t)\le b(t)<\infty \quad \forall t<\infty$.

L'inviluppo di traffico deterministico *b(t)* costituisce un limite superiore (*upper bound*) per il traffico generato $X(t)$.

Se l'inviluppo di traffico deterministico viene usato per assegnare la capacità ai flussi di traffico, il valore di capacità, ottenuto imponendo particolari requisiti di QoS in termini di ritardo e probabilità di violazione di tale ritardo, risulta significativo solo nel caso in cui l'inviluppo di traffico deterministico non sia troppo conservativo.
Non sempre è possibile definire l'inviluppo di traffico deterministico. Inoltre l'inviluppo di traffico deterministico è generalmente troppo conservativo, il che comporta una sovrastima del traffico generato dalle sorgenti con conseguente allocazione della capacità eccessivamente conservativa e

relativo spreco di risorse. Un metodo più efficace per il calcolo della capacità da allocare ai flussi di traffico usa l'inviluppo di traffico statistico.

Definizione 2. L'inviluppo di traffico statistico è un processo stocastico, *B(t)*, tale che:

$$P[X(t) > z] \leq P[B(t) > z] \quad \forall z,t\,.$$

Le sorgenti con un inviluppo di traffico deterministico, *b(t)*, hanno anche un inviluppo di traffico statistico, poiché basta scegliere *B(t)=b(t)*. Tuttavia sorgenti senza inviluppo di traffico deterministico possono avere un inviluppo di traffico statistico.

2.1.1. Rate-Variance Envelope

L'allocazione di capacità è tanto più efficiente se si utilizza l'inviluppo di traffico statistico invece di quello deterministico. Ma il calcolo dell'inviluppo di traffico statistico di una sorgente è in genere molto complicato ed è per questo motivo che in letteratura sono state proposte varie approssimazioni. Un inviluppo di traffico statistico può essere calcolato in modo semplice con metodi approssimati, a partire da quello deterministico. Il primo passo per questo calcolo è l'introduzione della definizione di *rate-variance envelope* (J. Qiu e E. Knightly, 1999).

La velocità istantanea di una sorgente, *r(t)*, è definita come:

$$r(t) = \frac{d}{dt} X(t)\,.$$

Inoltre la velocità media nell'intervallo di tempo t, $\tilde{r}(t)$, vale:

$$\tilde{r}(t) = E\left[X(t)/t\right]$$

mentre la velocità media, $\tilde{r}$, vale:

$$\tilde{r} = \lim_{t\to\infty} \tilde{r}(t) = \lim_{t\to\infty} X(t)/t \leq \lim_{t\to\infty} b(t)/t\,.$$

Definizione 3. La Rate Variance, $Var\left(X(t)/t\right)$, rappresenta la varianza della velocità media della sorgente in un intervallo di durata t.

In generale il calcolo della *rate variance* è complesso, ma è possibile calcolarne un'approssimazione (*upper bound*) a partire dall'inviluppo di traffico deterministico.

Definizione 4. Si definisce *Rate-Variance Envelope*, $RV(t)$, il limite superiore (*upper bound*) della *rate-variance*.

Il *Rate-Variance Envelope* è molto usato in quanto cattura le caratteristiche di correlazione del momento secondo di un processo d'arrivo al pari della funzione di autocorrelazione. Tale valore può essere calcolato seguendo due approcci, ampliamente documentati in letteratura (E. Knightly, 1997),

detti l'*adversarial approach* e il *non-adversarial approach*. Per una sorgente il cui traffico può essere descritto come un processo stazionario, si può ottenere la seguente limitazione nel caso di $RV(t)$ con *adversarial approach:*

$$RV(t) \le \tilde{r}\, b(t)/t - \tilde{r}^2 ,$$

conoscendo l'inviluppo di traffico deterministico *b(t)*.

È possibile estendere tale risultato al caso di *N* sorgenti.

Se l'i-esima sorgente è caratterizzata dall'avere un inviluppo di traffico deterministico pari a $b_i(t)$, la somma della *N* sorgenti avrà un inviluppo di traffico deterministico pari a

$$b(t) = \sum_{i=0}^{N} b_i(t) ,$$

tale che:

$$X_i(t) \le b_i(t) \Rightarrow \sum_{i=1}^{N} X_i(t) \le \sum_{i=1}^{N} b_i(t) \Rightarrow X(t) \le b(t) \quad \forall t .$$

Il *rate-variance envelope* di *X(t)* è uguale a:

$$RV(t) = \sum_{i=1}^{N} RV_i(t) ,$$

essendo $RV_i(t)$ il *rate-variance envelope* della singola sorgente i, cioè il flusso aggregato può essere trattato come una singola sorgente con *rate-variance envelope* pari alla somma dei *rate-variance envelope* delle singole sorgenti.

Tale risultato è dovuto al fatto che le sorgenti sono assunte statisticamente indipendenti e pertanto vale la relazione:

$$Var\big(X(t)/t\big) = Var\left(\left(\sum_{i=1}^{N} X_i(t)\right)/t\right) = \sum_{i=1}^{N} Var\big(X_i(t)/t\big) .$$

In generale, data una sorgente con inviluppo di traffico deterministico, $b(t)$, e *rate-variance envelope*, $RV(t)$, l'inviluppo di traffico statistico può essere facilmente calcolato assumendo che $f(x) \sim N(m, s^2)$, cioè che il traffico in ingresso abbia inviluppo distribuito normalmente con media m e varianza s^2. La validità di questa affermazione è avvalorata dal teorema del limite centrale; l'approssimazione normale è tanto più corretta quanto più elevato è il numero di sorgenti. Si osservi inoltre che l'approssimazione, quando corretta, è valida solo al centro come assicura il teorema del limite centrale, mentre le code non è detto che la soddisfino. Questo rappresenta un grande limite al metodo basato sull'uso della *Maximun Variance Approximation* (MVA).

In particolare considerando ancora le *N* sorgenti che generano un traffico pari a:

$$X(t) = \sum_{i=1}^{N} X_i(t),$$

il flusso aggregato avrà una densità di probabilità $f(x)$ espressa dalla convoluzione della densità di probabilità delle singole sorgenti:

$$f(x) = f_1(x) * f_2(x) * ... * f_N(x).$$

Il teorema del limite centrale garantisce che per un grande valore di *N* l'approssimazione

$$f(x) \sim N(m, s^2)$$

sia corretta, con:

$$m = \sum_{i=1}^{N} m_i \text{ e } s^2 = \sum_{i=1}^{N} s_i^2.$$

Si consideri una sorgente, o un aggregato di *N* sorgenti, con un inviluppo di traffico deterministico pari a $b(t)$ e un *rate-variance envelope* pari a $RV(t)$. Esprimiamo media e varianza in funzione della velocità media, sapendo che il traffico totale generato nell'intervallo di tempo t è pari a $X(t)$. Per le definizioni precedenti si ha:

$$\tilde{r}(t) = E[X(t)/t] \Rightarrow E[X(t)] = \tilde{r}(t)t;$$

$$Var(X(t)/t) \Rightarrow Var(X(t)) = Var(X(t)/t)t^2 \le RV(t)t^2.$$

Applicando il teorema del limite centrale su *X(t)* si ottiene che

$$f(t) \approx N\left(\tilde{r}(t)t, Var(X(t)/t)t^2\right).$$

L'inviluppo di traffico Statistico Approssimato di *X(t)* è pari a:

$$B(t) = N(\tilde{r}(t)t, RV(t)t^2).$$

2.1.2. Probabilità di violazione della soglia di ritardo

In un *link* condiviso tra più classi di servizio, alla classe *k* sia allocata la capacità c_k. Anche se temporaneamente la classe *k* ha dei pacchetti *backlogged*, ossia dei pacchetti in coda che non possono essere inviati per assenza di risorse, la classe riceve certamente un servizio ad una velocità pari almeno a c_k. Se la classe *k* non è *backlogged*, la capacità non usata da tale classe viene distribuita equamente a tutte le sessioni con pacchetti da trasmettere. In questo modo le classi avranno garantite le rispettive richieste di QoS.

Con riferimento ad uno schedulatore con *L* classi di servizio definiamo classe di servizio *i* il flusso di traffico generato da N^i sorgenti di traffico. La *j-esima* sorgente della classe *i* sia caratterizzata

dall'inviluppo di traffico deterministico $b_j^i(t)$. Si supponga inoltre che tutte le sorgenti di traffico della classe *i* abbiano le stesse richieste di QoS[i].

Il traffico totale generato dalla classe di servizio *i* al tempo t è espresso dalla relazione

$$X^i(t) = \sum_{j=1}^{N^i} X_j^i(t).$$

Analogamente il t*raffico totale servito,* $Y_j^i(t)$, è il traffico totale servito per la *j-esima* sorgente della classe *i*, mentre il t*raffico servito per* N^i *sorgenti,*

$$Y^i(t) = \sum_{j=1}^{N^i} Y_j^i(t),$$

è il traffico totale della classe *i.*

Vengono date anche le seguenti definizioni:

Traffico backlog, $Q^i(t)$, è l'ammontare di traffico che deve essere ancora spedito, bloccato per mancanza di risorse, dell'i-esima classe nel tempo t,

$$Q^i(t) = \max_t \left\{X^i(t) - Y^i(t)\right\}.$$

Intervallo backlog, è l'intervallo di tempo per la classe *i* che soddisfa la relazione:

$$Q^i(t) > 0 \quad \forall t.$$

Minimo processo d'ingresso backlogged, $\tilde{X}^i(t)$, è la minima quantità di traffico che la classe *i* può generare nell'intervallo di tempo t per essere continuamente *backlogged*.

Servizio disponibile, $\tilde{Y}^i(t)$, è il traffico in uscita alla classe *i* nell'intervallo di tempo t.

Definizione 5. Inviluppo di servizio deterministico per la classe *i* è la funzione del tempo $s^i(t)$, non-decrescente e non-negativa che soddisfa la seguente relazione:

$$\tilde{Y}^i(t) \geq s^i(t) \quad \forall t.$$

Definizione 6. Virtual Delay del traffico della classe *i*, $D^i(t)$, è il ritardo sperimentato dal traffico della classe *i* giunto nel tempo t,

$$D^i(t) = \min\left\{\Delta t : \Delta t \geq 0 \, e \, X(0,t) \leq Y(0, t+\Delta t)\right\};$$

Δt è il più piccolo intervallo di tempo sufficiente a smaltire la coda di traffico accumulata nell'intervallo di tempo t.

La metrica chiave adottata per valutare la QoS è la probabilità di violare il limite di ritardo d^i, espressa dalla relazione $P\left\{D^i(t) > d^i\right\}$; essendo $X^i(t)$ un processo stazionario ed ergodico, tale relazione converge alla relazione $P\left\{D^i > d^i\right\}$.

Per una classe di servizio *i*, con un inviluppo di servizio deterministico $s^i(t)$ e un inviluppo di traffico statistico $B^i(t)$, la probabilità di violare la soglia di ritardo è espressa dalla seguente relazione

$$P\{D^i > d^i\} \le P\{\max_{t\ge 0}\{B^i(t) - s^i(t+d^i)\} > 0\}$$ (J. Qiu e E. Knightly, 1999).

Mentre l'inviluppo di servizio deterministico $s^i(t)$ fornisce isolamento fra le classi di servizio e semplifica il controllo delle ammissioni, si preclude la condivisione statistica delle risorse tra le classi e del relativo guadagno che la multiplazione apporterebbe. Al fine di migliorare la condivisione delle risorse, non solo all'interno di una singola classe di servizio, ma anche tra le diverse classi è opportuno introdurre il concetto di inviluppo di servizio statistico $S^i(t)$.

L'inviluppo di servizio statistico nella classe *i* è la sequenza di variabili casuali $S^i(t)$ che soddisfano la seguente relazione:

$$P[\tilde{Y}^i_{t_1,t} > z] \le P[S^i(t) > z] \quad \forall z,t\,.$$

Utilizzando la definizione dell'inviluppo di servizio statistico la probabilità di violazione della soglia di ritardo diventa:

$$P\{D^i > d^i\} \le P\{\max_{t\ge 0}\{B^i(t) - S^i(t+d^i)\} > 0\}.$$

2.2. Criterio della Maximum Variance Approximation per il calcolo della probabilità di violazione della soglia di ritardo

Nel presente paragrafo viene ricavata la formula della probabilità di violazione della soglia di ritardo nel particolare caso in cui lo schedulatore adottato sia di tipo *Static Priority* (SP) con *L* classi di servizio (Figura 1).

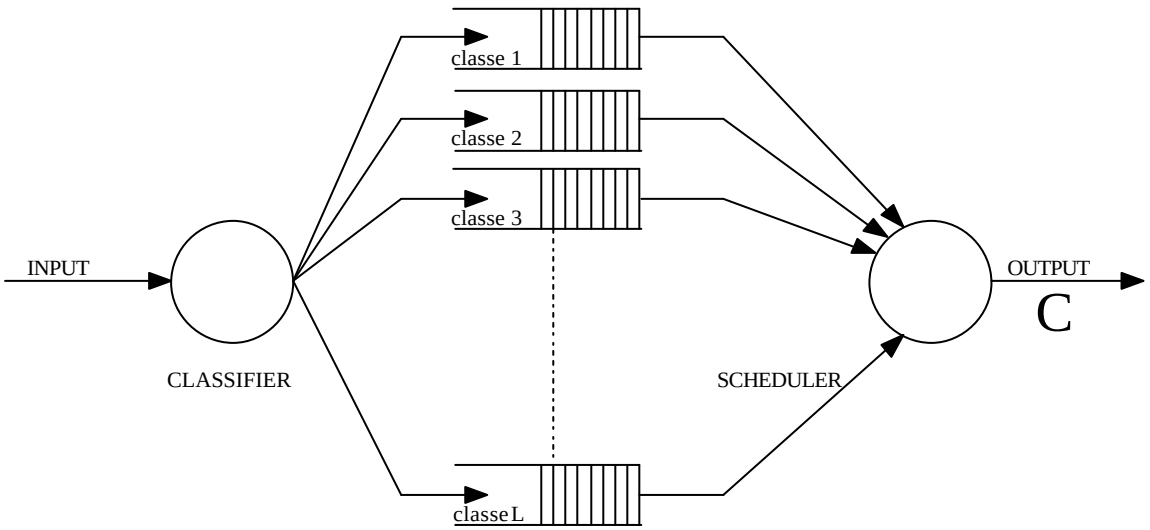


Fig.1- Schedulatore SP con *L* classi di servizio

Con uno schedulatore SP avente un *link* di capacità *C* bit/s, il traffico aggregato della classe *i* è limitato dall'inviluppo di traffico statistico $B^i(t)$ e da quello deterministico $b^i(t)$ con *i=1,…,L*.

Per la classe *i* l'inviluppo di servizio statistico è:

$$S^i(t) = \max\left\{0, (Ct - \sum_{j=1}^{i-1} B^j(t))\right\},$$

e l'inviluppo di servizio deterministico è:

$$s^i(t) = \max\left\{0, (Ct - \sum_{j=1}^{i-1} b^j(t))\right\},$$

essendo $b^i(t) = \sum_{j \in C_i} b_j(t)$ con $b_j(t)$ inviluppo deterministico del *j-esimo* flusso appartenente alla classe *i* e $B^i(t) = \sum_{j \in C_i} B_j(t)$ con $B_j(t)$ inviluppo statistico del *j-esimo* flusso appartenente alla classe *i*.

Per ogni classe di traffico i parametri di QoS sono espressi in termini di limite di ritardo, d^i, e probabilità di violare tale limite di ritardo, P^i. Si dimostra che per tutte le classi di servizio dello schedulatore le richieste di QoS sono soddisfatte se:

- per tutte le classi di servizio deterministico con $P^i = 0$ è soddisfatta la relazione

$$\max_t \left\{ b^i(t) + \sum_{k=1}^{i-1} b^k(t + d^i) - C(t + d^i) \right\} \le 0;$$

- per tutte le classi di servizio statistiche con $P^i > 0$ è soddisfatta la relazione

$$P\left\{ \max_t \left\{ B^i(t) + \sum_{k=1}^{i-1} B^k(t + d^i) - C(t + d^i) \right\} > 0 \right\} \le P^i.$$

Per ottenere queste relazioni si tenga presente che per uno schedulatore SP solo le classi di servizio a priorità più alta influenzano quelle a priorità più bassa e non viceversa.

Il calcolo dell'inviluppo di traffico statistico è in generale molto complesso, a meno che non si usi la *Maximum-Variance Approximation* (MVA). Infatti, ricordando che:

$$RV(t) = Var(X(t)/t) \text{ e } E[X] = m,$$

il valor medio e la varianza dell'inviluppo di traffico statistico del flusso *j* diventano rispettivamente:

$$E[B_j(t)] = m_j t, \; Var[B_j(t)] = t^2 RV_j(t).$$

Quando un numero sufficiente di flussi sono multiplati, l'inviluppo del traffico aggregato per la classe *i* converge ad un inviluppo gaussiano per il Teorema del Limite Centrale (E. Knightly, 1998).

Pertanto l'inviluppo di traffico statistico, $B^i(t)$, avrà media e varianza rispettivamente

$$E[B^i(t)] = \sum_{j \in C_i} m_j t \,,\; Var[B^i(t)] = \sum_{j \in C_i} RV_j(t) t^2 \,.$$

La *Maximum-Variance Approximation* si basa sull'ipotesi che il processo

$$\left\{B(t) - S(t + d^i)\right\}$$

sia gaussiano e che la sua funzione di autocovarianza soddisfi le relazioni C1 e C2 riportate in (E. Knightly, 1998).

Facendo uso dell'approssimazione *maximum variance* e nel caso in cui le condizioni C1 e C2 siano soddisfatte probabilità di violare la soglia di ritardo:

$$P\left\{D^i > d^i\right\} \le P\left\{\max_{t \ge 0}\left\{B^i(t) - S^i(t + d^i)\right\} > 0\right\}$$

si può calcolare con la relazione (J. Qiu e E. Knightly, 1999):

$$P\left\{D^i > d^i\right\} \le P[\max_t \{B(t) - C(t + d)\} > 0] \le e^{-\frac{a^2}{2}}$$

dove:

$$s_t^2 = Var\left\{B(t) - S(t + d^i)\right\}$$

$$a_t = \frac{0 - E\left\{B(t) - S(t + d^i)\right\}}{s_t}$$

$$a = \inf_t a_t$$

La relazione trovata fornisce un *Upper bound* della probabilità di violazione della soglia di ritardo.

Se l'ipotesi d'inviluppo di traffico gaussiano non fosse valida queste formule fornirebbero solo delle approssimazioni, ma non dei limiti, per la probabilità di violazione dei ritardi.

Nei prossimi paragrafi verranno presentati due esempi; il primo nel caso in cui il processo d'ingresso sia gaussiano, ed il secondo nel caso in cui l'approssimazione gaussiana non sia verificata.

2.3. Processi d'ingresso gaussiani e non gaussiani

Al fine di evidenziare vantaggi e limiti del metodo descritto vengono di seguito proposti due casi di studio: processi statistici d'ingresso approssimabili a processi gaussiani e non approssimabili a processi gaussiani.

2.3.1. Processi d'ingresso gaussiani

Si consideri una sorgente poissoniana con pacchetti di lunghezza gaussiana con le seguenti caratteristiche (Figura 2):

- Arrivi di Poisson con media ?=1000 pacchetti/s;
- La lunghezza dei pacchetti segue la distribuzione gaussiana: L~Gauss(l,s^2) con $l = 1000 bit$ e s^2=10000 bit.

La velocità media della sorgente è:

$$R_{sorgente} = ll = 1 Mbit / s$$

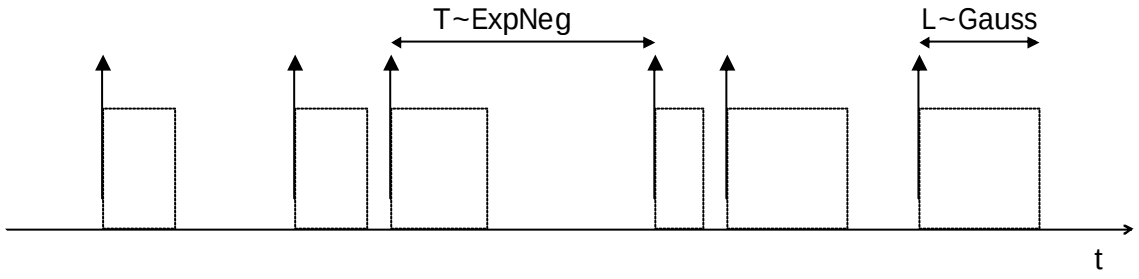


Fig. 2- Sorgente poissoniana con pacchetti di lunghezza gaussiana

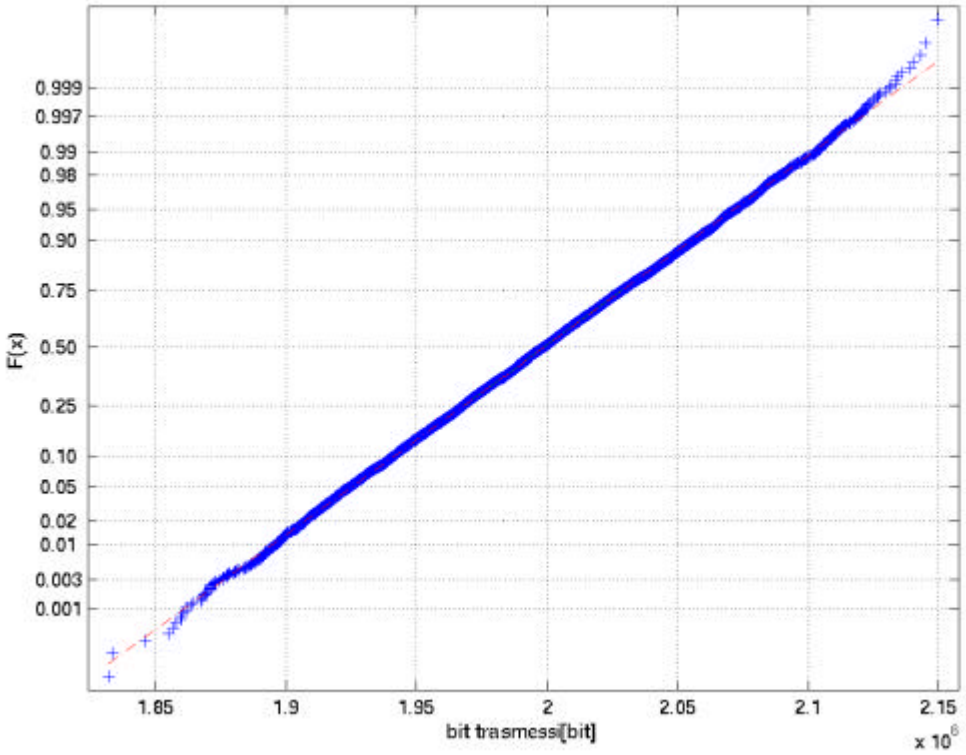


Fig. 3- Confronto della funzione di ripartizione tra una sorgente poissoniana con lunghezza dei pacchetti gaussiana e una sorgente gaussiana di pari media e varianza.

In Figura 3 vengono mostrate la funzione di ripartizione dei bit trasmessi dalla sorgente poissoniana, e la funzione di ripartizione gaussiana di pari media e varianza, rappresentata dalla linea tratteggiata.

La sorgente di traffico poissoniana con lunghezza dei pacchetti distribuita secondo una v.a. gaussiana L~Gauss(l,s^2) è approssimata in modo ottimale da una sorgente gaussiana.

In Figura 4 viene mostrata la probabilità di violare la soglia del ritardo pari a 100 ms, in funzione della capacità del canale d'uscita. Le due curve sovrapposte rappresentano rispettivamente: la simulata, la probabilità dei pacchetti di superare la soglia di ritardo imposta, mentre l'approx MVA è l'approssimazione ottenuta con il metodo *Maximun Variance Approximation,* descritto in precedenza, a partire dalla media e dalla varianza del processo dei bit emessi dalla sorgente.

Il fatto che le curve siano sovrapposte testimonia la correttezza dell'approssimazione nel caso in cui sia soddisfatta l'ipotesi di gaussianità degli ingressi.

I risalatati in Figura 4 confermano che per sorgenti a inviluppo gaussiano la MVA garantisce che la probabilità di violazione della soglia dei ritardi

$$P\{D^i > d^i\} \le P[\max_t \{B(t) - C(t+d)\} > 0] \le e^{-\frac{a^2}{2}}$$

fornisce un *upper bound*.

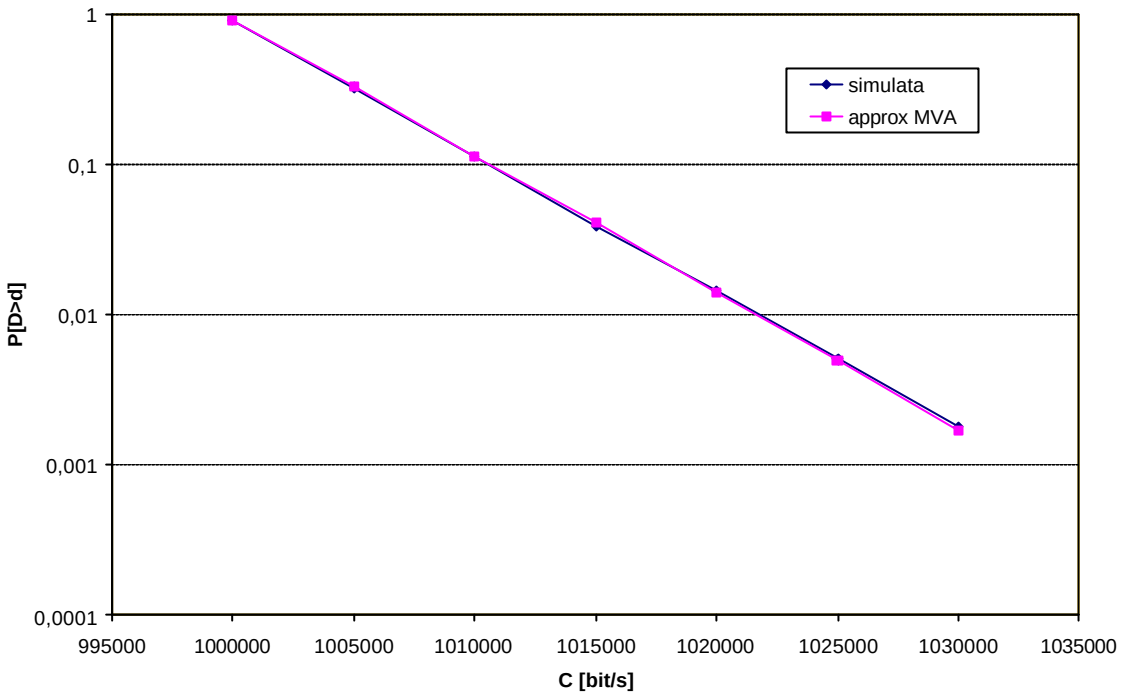


Fig. 4-Probabilità di violazione della soglia di ritardo di una sorgente poissoniana con lunghezza dei pacchetti $L \sim Gauss(\lambda, \sigma^2)$ in funzione della capacità del canale

2.3.2. Processi d'ingresso non gaussiani

Il metodo della *Maximum Variance Approximation* in assenza dell'ipotesi di ingressi con distribuzione gaussiana, non fornisce un *bound* alla probabilità di violazione del limite di ritardo, ma solo un'approssimazione. Inoltre il teorema del limite centrale, che dovrebbe garantire la gaussianità del traffico d'ingresso, perde validità se il numero di sorgenti non risulta sufficientemente elevato.

Si consideri una sorgente 3GPP [6], a due stati: uno di trasmissione di durata media pari a T_1 e uno di silenzio di durata media pari a T_0, come mostrato in Figura 5.

Durante lo stato di trasmissione la sorgente emette mediamente 100 pacchetti con interarrivo deterministico pari a t, e lunghezza fissa dei pacchetti pari a λ.

Fig. 5-Sorgente 3GPP

I parametri della sorgente sono:

- numero medio di pacchetti trasmessi nello stato T_1, n=100 pacchetti;
- interarrivo fisso tra i pacchetti nello stato di trasmissione: t =2.6 ms;
- durata dello stato di trasmissione distribuita esponenzialmente con media $T_1 = n \cdot t = 100 \cdot 260 \cdot 10^{-5} = 260 \cdot ms$;
- lunghezza costante del pacchetto: $l = 960 bit$;
- durata dello stato di silenzio distribuita esponenzialmente con media $T_0 = 8\ s$;
- la velocità di trasmissione nello stato 1 $R_1 = \frac{l}{t} = \frac{960\,bit}{2.6 ms} = 369.231 kbit/s$;
- la velocità media di una singola sorgente è

$$R_{sorgente} = \frac{T_1}{T_1 + T_0} \cdot R_1 = \frac{0.26}{0.26 + 8} \cdot 369231 = 11.622\,kbit/s;$$

- la velocità media dell'aggregato è:

$$R_{100sorg} = 1.1622\,Mbit/s;$$

- la velocità media dell'aggregato è:

$$R_{1000sorg} = 11.622\,Mbit/s.$$

Multiplando 100 sorgenti di tipo 3GPP il grafico del test di gaussianità ottenuto con riferimento agli intervalli temporali (0,1) e (0,5) secondi è riportato rispettivamente nelle Figure 6 e 7, che mostrano le funzioni di ripartizione, F(x), della sorgente 3GPP e della sorgente gaussiana con pari media e varianza. L'ipotesi di gaussianità migliora considerando intervalli di tempo sempre più grandi.

In Figura 8 e 9 vengono mostrati i test di gaussianità ottenuti con riferimento agli intervalli temporali (0,1) e (0,5) secondi per 1000 sorgenti 3GPP. Aumentando il numero di sorgenti fino a 1000 migliora l'ipotesi di gaussianità anche per brevi intervalli di tempo (0,1) secondi.

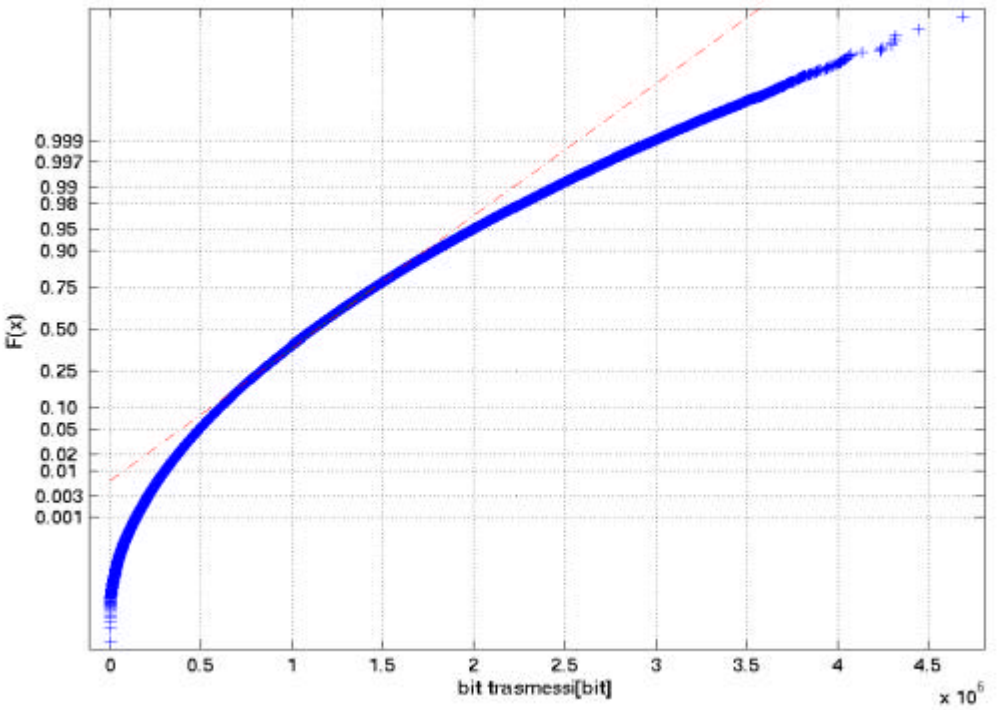


Fig. 6- Funzione di ripartizione dei bit trasmessi nell'intervallo di tempo (0,1) secondi da 100 sorgenti GPP

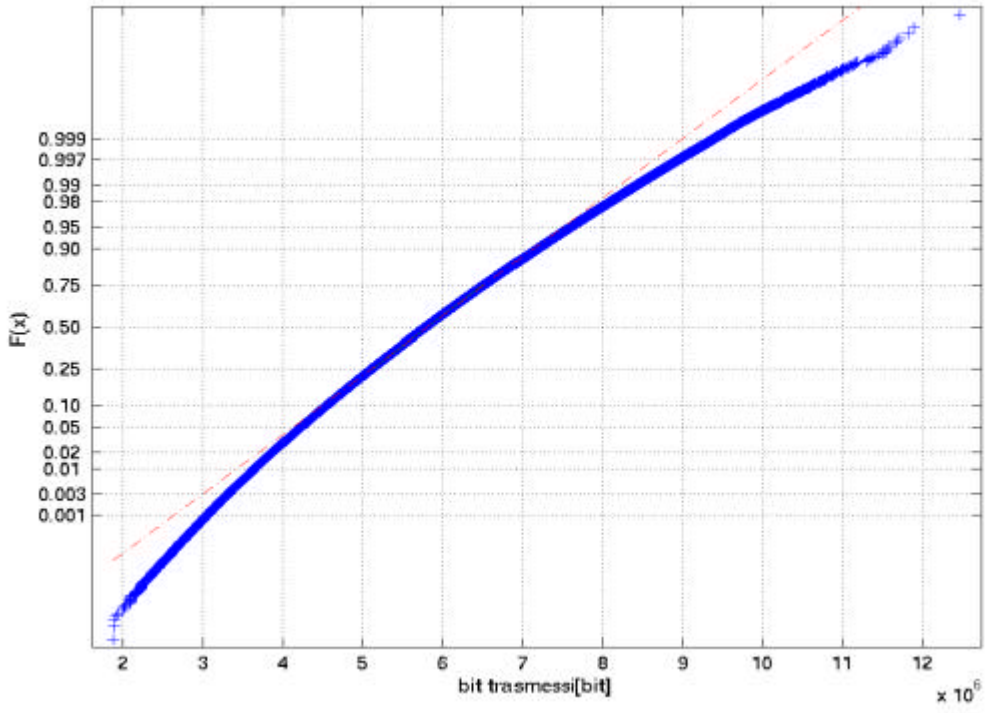


Fig. 7- Funzione di ripartizione dei bit trasmessi nell'intervallo di tempo (0,5) secondi da 100 sorgenti 3GPP

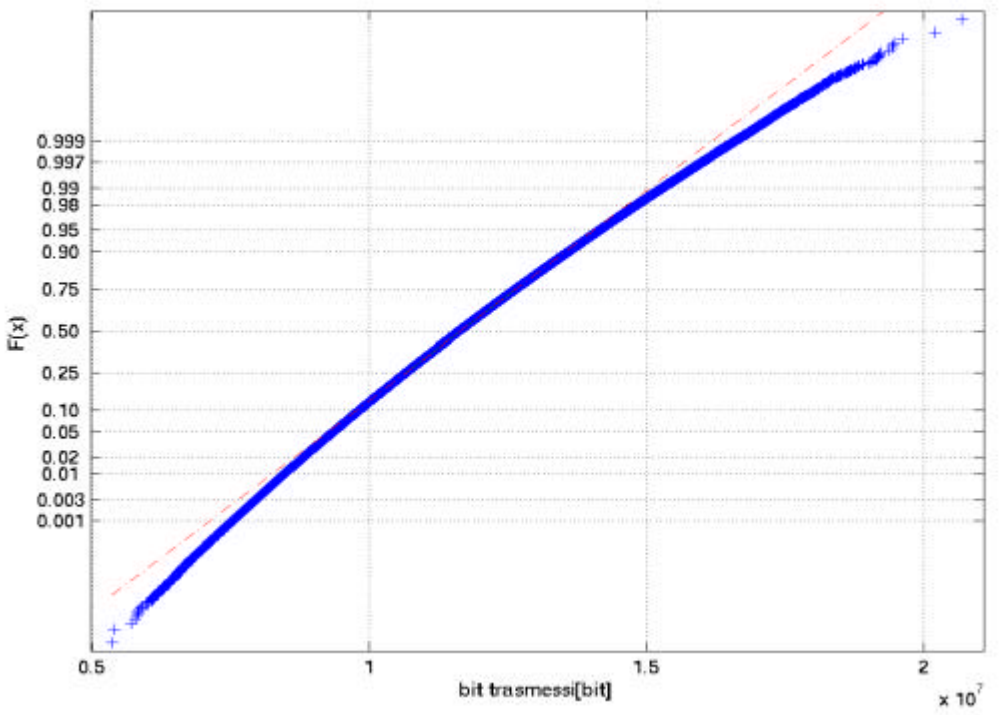


Fig. 8- Funzione di ripartizione dei bit trasmessi nell'intervallo di tempo (0,1) secondi da 1000 sorgenti 3GPP

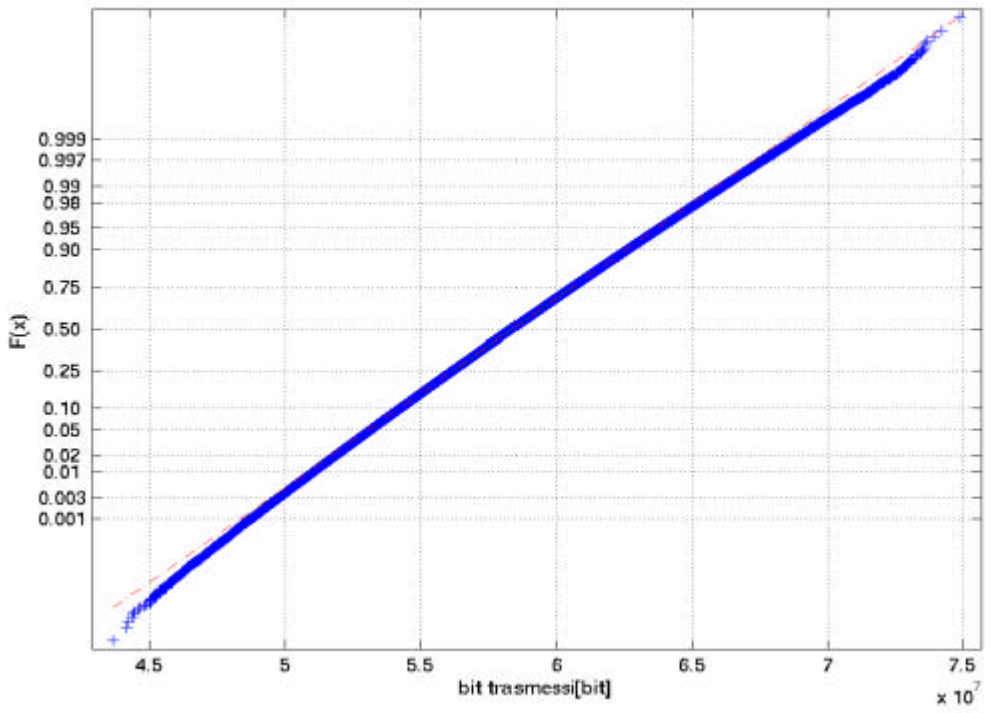


Fig. 9- Funzione di ripartizione dei bit trasmessi nell'intervallo di tempo (0,5) secondi da 1000 sorgenti 3GPP

Per completare il confronto in Figura 10 viene mostrata la probabilità di violare la soglia di ritardo, posta pari a 100 ms, da parte di un flusso aggregato costituito da 100 sorgenti 3GPP multiplate, in funzione della capacità del canale d'uscita dello schedulatore.

L'approssimazione MVA che è stata ricavata come *upper bound* in realtà rimane sopra la curva simulata solo per valori di capacità bassi; questo è in parte giustificato dal fatto che l'ipotesi di gaussianità non è pienamente verificata e che si ha grande divergenza in corrispondenza delle code della gaussiana, cioè per valori piccoli della probabilità.

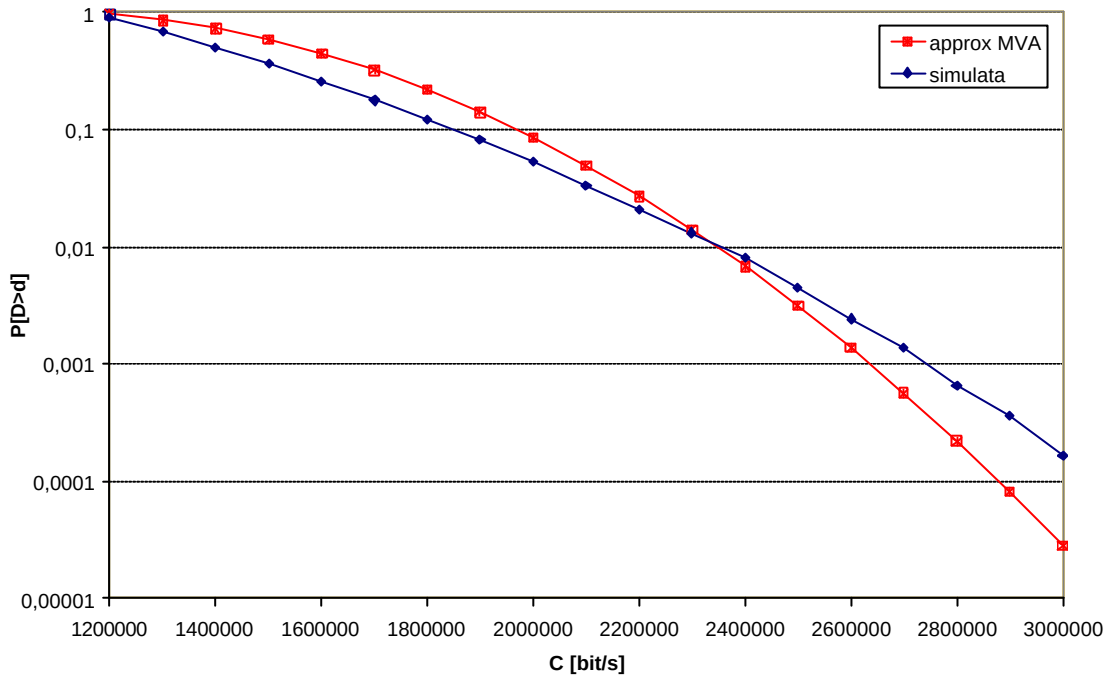


Fig. 10- Probabilità di violare la soglia di ritardo per 100 sorgenti 3GPP in funzione della capacità del canale

La Figura 11 mostra la probabilità di violare la soglia di ritardo, posta pari a 100 ms, da parte di un flusso aggregato costituito da 1000 sorgenti multiplate, in funzione della capacità del canale d'uscita dello schedulatore.

L'approssimazione MVA che è stata ricavata come *upper bound* si comporta abbastanza bene, sicuramente meglio che nel caso di 100 sorgenti; questo perchè l'ipotesi di gaussianità in questo caso risulta verificata.

Per piccoli valori di probabilità di violazione della soglia del ritardo l'approssimazione comincia a vacillare a causa delle code della gaussiana e l'*upper bound* diventa una approssimazione.

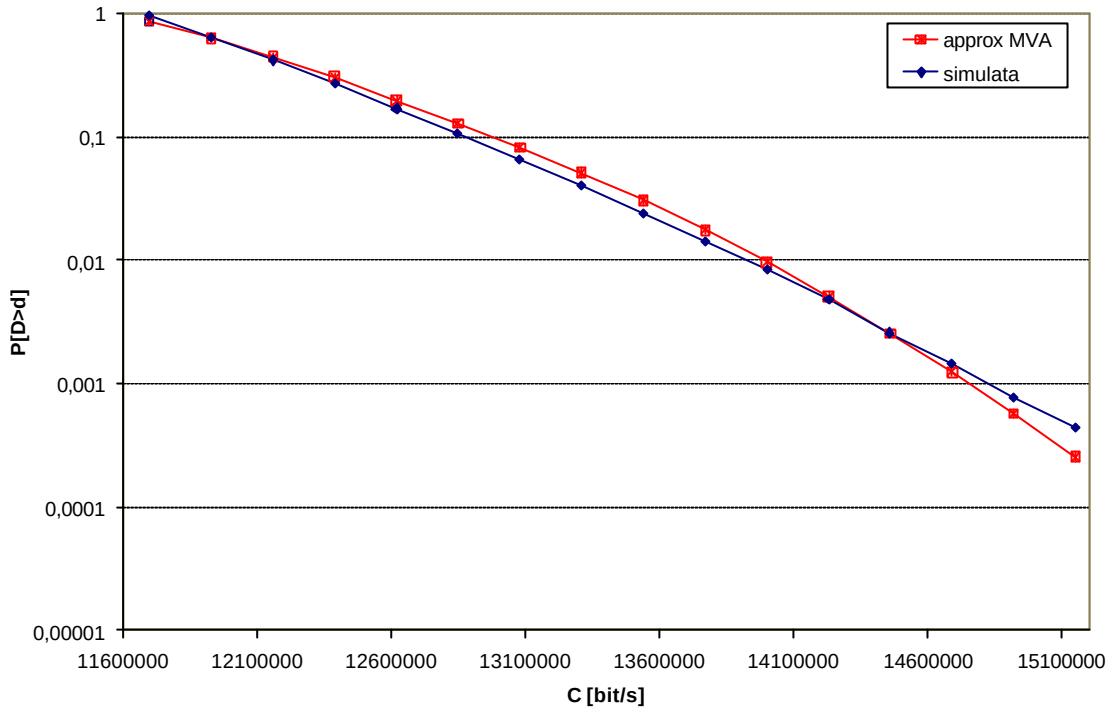


Fig. 11- Probabilità di violare la soglia di ritardo per 1000 sorgenti 3GPP in funzione della capacità del canale

2.4. Conclusioni e sviluppi futuri

Dalla definizione degli inviluppi statistici di servizio è possibile ricavare, attraverso approssimazioni, un semplice metodo per garantire QoS nelle reti IP DiffServ attraverso il calcolo della probabilità di violazione della soglia di ritardo.

Il metodo che conduce al calcolo della probabilità di violazione dei ritardi presuppone la validità dell'approssimazione del Teorema del Limite Centrale con un numero sufficiente di sorgenti multiplato in un nodo della rete.

Nella prima fase del lavoro abbiamo approfondito attraverso simulazione i limiti di tale metodo, quando non risulta soddisfatta l'ipotesi di gaussianità del traffico in ingresso.

Le fasi successive del lavoro prevedono lo studio analitico e simulativo di uno scenario in cui ogni sorgente di traffico viene modellata attraverso un token bucket di parametri r,b, dove r è la velocità media di riempimento del bucket e b la sua profondità.

In tal caso il vantaggio è quello di poter lavorare direttamente con i flussi di traffico offerti alla rete e non con i pacchetti di ogni singolo flusso, con una notevole guadagno computazionale e di semplicità.
La trattazione analitica può essere diversificata in relazione al numero di classi di servizio previste nella rete ed in base al tipo di schedulatore adottato (SP Static Priority, GPS Generalized Processor Sharing, ecc.).
La conoscenza del metodo degli inviluppi di traffico potrà essere usata per definire nuovi meccanismi di *Flow Admission Control* e protocolli alternativi di instradamento, per consentire QoS nei sistemi di telecomunicazione.

3. Modello per il routing inter-domain

Lo scopo di questa sezione è di introdurre un modello di routing inter-domain in grado di descrivere reti in cui un gran numero *Autonomous Systems* (*AS*) gestisce in autonomia parte dell'infrastruttura che forma la rete complessiva. Tale modello è usato nelle sezioni 4 e 5 in cui si mostra come tali informazioni possono essere reperite e come con esse è possibile introdurre un semplice modello di rete adattativa a livello inter-domain.

3.1. Relazioni commerciali tra Autonomous Systems

Le relazioni commerciali, tra organizzazioni che gestiscono le infrastrutture Internet, non sono il risultato di una azione pianificata ma di una interazione tra forze di tipo tecnologico e forze di tipo economico (G. Huston, 1999). La situazione in Internet è molto diversa da quella del mondo della telefonia dove ciascun operatore deve possedere una particolare licenza, rilasciata da apposite autorità, per poter far parte del gruppo di organizzazioni che offrono certi tipi di servizi di telecomunicazione: una volta acquisita tale licenza gli operatori si rapportano l'uno con l'altro come tra pari, mentre tutte le organizzazioni non dotate di tale licenza sono automaticamente considerate clienti delle prime.
Un Internet Service Provider, invece, non ha bisogno di alcuna licenza e quindi ciascun rapporto tra organizzazioni viene regolato in base a specifiche contrattazioni.
Un AS fa sempre capo ad una singola organizzazione (normalmente un *Internet Service Provider, ISP*) che gestisce le apparecchiature in autonomia. Tale gestione ha dei costi e induce presumibilmente dei vantaggi di tipo economico. Tutte le attività di dell'organizzazione che gestisce l'AS sono quindi guidate da motivazioni di tipo economico. Tra gli AS si possono quindi individuare delle relazioni contrattuali che sono, nella pratica, relazioni contrattuali tra organizzazioni (normalmente distinte). L'oggetto del contratto è un servizio di connettività da e verso certe parti della rete o da e verso tutta la rete. Il prezzo da pagare per l'erogazione di tale servizio è o monetario o di connettività rispetto ad una certa altra parte della rete.

Sebbene ci possano essere innumerevoli varianti negli aspetti contrattuali tra AS e non sono rari casi di rapporti "ibridi" rispetto alle categorie che saranno introdotte, possiamo schematizzare le relazioni tra AS nelle seguenti tre: customer-provider, peer-peer, sibling-sibling (Gao 2001).

Customer-provider. In una relazione customer-provider il *provider* vende un servizio di connettività da e verso il resto di Internet ad un suo *customer*. Il servizio di connettività non può essere altro che best-effort poiché non vi è alcuna informazione del cammino che il traffico percorre in Internet e quindi del deterioramento che questo può subire nel suo cammino da e verso la destinazione. Il costo del link tra customer e provider è a carico del customer e la tariffazione applicata al customer può essere di vari tipi, ad esempio

- flat: la tariffazione è indipendente dal traffico che il customer invia/riceve sul link
- a traffico: la tariffazione è proporzionale al traffico che il customer invia/riceve sul link (o ai picchi di traffico osservati ad esempio in ciascuna giornata)
- a traffico con minimo fissato: la tariffazione è proporzionale al traffico che il customer invia/riceve sul link ma il customer non può pagare meno di una quota minima di traffico, ciò permette al provider di coprire le spese di gestione del link in ogni caso.

Nulla vieta al customer di rivendere la connettività acquistata dal provider ad un suo customer diventando esso stesso provider di quest'ultimo. Un customer può acquistare connettività da vari provider per vari motivi, ad esempio backup.

Peer-peer. In una relazione peer-peer due AS, approssimativamente della stessa importanza, stipulano un accordo per lo scambio di traffico relativo ai propri customer. Normalmente questo accordo mira ad apportare un vantaggio simile ai due contraenti. Supponendo che il bacino di clienti dei due peer sia paragonabile, anche il traffico lo sarà. I due peer risparmiano il costo di dover passare attraverso un provider per connettere i rispettivi clienti. Il costo del link tra i peer è normalmente condiviso. Molto spesso gli ISP afferiscono ai cosiddetti Internet eXchange point (IX), previo pagamento di un canone, dove possono liberamente accordarsi per effettuare peering con un sottoinsieme qualsiasi degli ISP che afferiscono all'IX. Il costo di una operazione di questo tipo è molto piccolo in quanto il canone per l'IX viene pagato una sola volta, per fare peering si usa l'infrastruttura di switching dell'IX. Rimangono solo i costi di gestione del peering.

Fare peering agli IX è molto conveniente per i piccoli e medi ISP. I grandi operatori mirano ad avere relazioni di peering con grandi operatori su link privati.

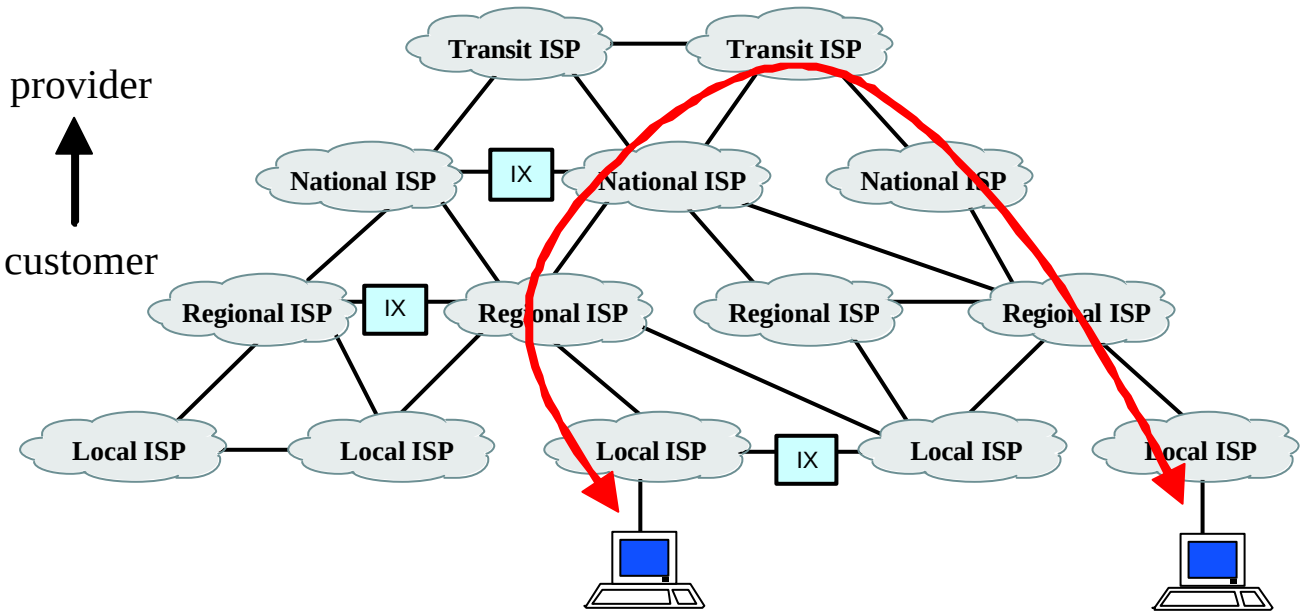
Il rapporto di tipo peer-peer ha un senso quando il traffico in entrambi i versi è bilanciato. Qualora il bacino di customer tra i due peer diventi molto sbilanciato il peer più grande chiede una ricontrattazione in una forma di tipo customer-provider. Proprio per evitare che una relazione di peering

diventi anomala possono essere introdotte nell'accordo tra le parti delle penali che renda il peering sconveniente nel caso in cui il traffico raggiunga un certo livello di sbilanciamento.

Sibling-sibling. In una relazione sibling-sibling due AS si scambiano traffico senza alcun limite. Tale accordo è sensato qualora i due AS fanno parte della stessa organizzazione o di organizzazioni con una cooperazione molto stretta (ad esempio tra organizzazioni che operano sotto uno stesso marchio in stati differenti). In tal caso le motivazioni economiche sono, in qualche modo, comuni e l'aspetto competitivo tende ad essere ininfluente. In linea di principio non ci sarebbe motivo di avere due AS distinti, tuttavia per motivi organizzativi può essere conveniente mantenere AS distinti per zone geograficamente distinte. Inoltre, in caso di riorganizzazioni societarie (ad esempio acquisizioni o fusioni) può essere conveniente mantenere invariata la configurazione delle apparecchiature.

3.2. La struttura di Inernet

È possibile individuare una struttura gerarchica che nasce dalle relazioni commerciali tra ISP (o tra AS). Come mostrata nella seguente figura.

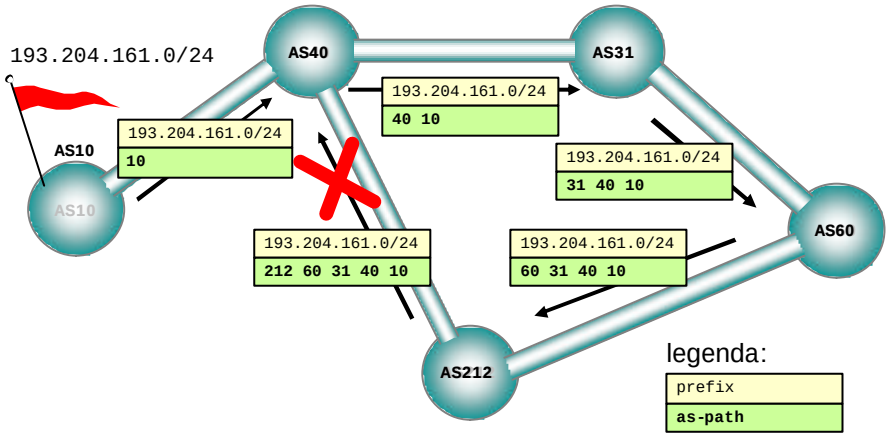


ISP di livello più basso acquistano connettività da ISP di livello superiore. Solo gli ISP a livello più alto, i cosiddetti tier-1, non acquistano connettività ma hanno solo relazioni peer-peer tra tutti gli altri ISP tier-1. In questa situazione il traffico segue un percorso che tende a salire fino a ISP tier-1 e quindi a scendere verso la destinazione. Gli ISP di livello basso o intermedio possono accordarsi per creare relazione peer-peer al fine di risparmiare il costo del transito attraverso i loro provider. In tal caso il

traffico tra customer di due ISP che hanno un accordo di tipo peer-peer non viene instradato tramite i provider ma tramite un link diretto.

3.3. Configurazioni BGP e flussi del traffico

Le relazioni customer provider hanno un impatto immediato sulle politiche di routing realizzate dagli ISP per i loro AS. Tramite il protocollo di routing inter-domain BGP (Border Gateway Protocol) due AS si scambiano informazioni di raggiungibilità detti *annunci*. Un *prefisso IP* identifica un insieme di indirizzi IP "vicini tra loro". Un annuncio è una informazione del tipo "tramite me puoi raggiungere un *prefisso IP* e passerai per la seguente sequenza di AS: ASx ASy....". Il cammino di AS associato al prefisso viene detto AS-path. La coppia formata da prefisso e AS-path è detta *rotta*. Un ISP che fornisce un servizio di transito permette all'annuncio di fluire attraverso di esso verso altri AS e quindi tale ISP si può proporre ad altri ISP come via per raggiungere un certo prefisso IP anche se tale prefisso non è direttamente gestito da lui. Nel propagare l'annuncio inserisce in testa all'AS-path il suo numero di AS. Tale tecnica in BGP viene utilizzata per evitare che il routing produca dei cicli come mostrato nella seguente figura.



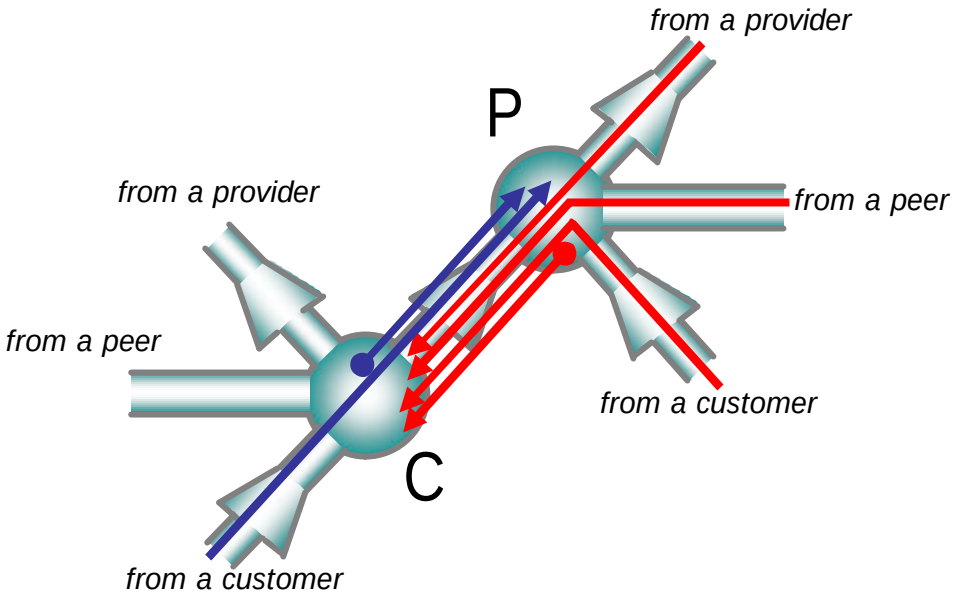
Un provider P annuncia ad un suo customer C

- tutti i suoi prefissi
- i prefissi annunciati a P dai suoi customer
- i prefissi annunciati a P dai suoi peer
- i prefissi annunciati a P dei suoi provider.

mentre C annuncia a P

- tutti i suoi prefissi
- i prefissi annunciati a C dai suoi customer

C non annuncia a P i prefissi dei suoi altri provider o peers altrimenti offrirebbe ad essi un servizio di transito gratuito. La situazione è riassunta nella seguente figura



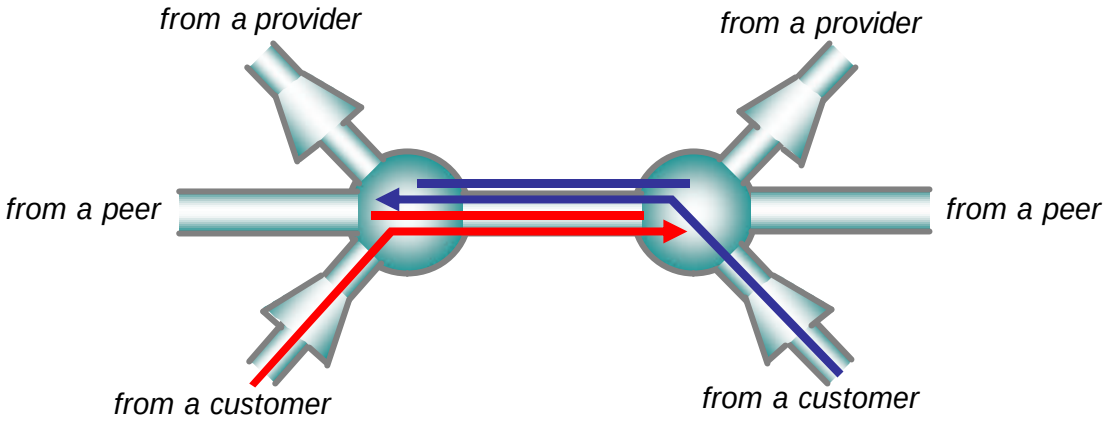
customer-provider relationship

L'AS P vende a C connettività globale verso Internet e quindi annuncerà rotte per raggiungere tutti i prefissi di Internet (*full routing table*). Qualora lo stesso prefisso venga fornito da più AS P preferisce annunciare a C, nell'ordine, le rotte dei propri customer, quelle dei propri peer e, quindi, quelle dei propri provider. Il motivo è molto semplice: il link di connettività verso i customer vengono pagati dai customer stessi quindi si possono sfruttare senza alcun aggravio economico per P. I link di connettività verso i peer sono pagati in condivisione con gli stessi peer. I link di connessioni verso i provider di P vengono pagati da P e, inoltre, il traffico su tali link può essere tariffato in maniera proporzionale al volume prodotto.

Nella seguente figura viene mostrato come due peers si annunciano

- tutti i loro prefissi
- i prefissi annunciati dai loro customer

Essi non si annunciano i prefissi annunciati dai loro provider o da altri peers altrimenti offrirebbero ad essi un servizio di transito gratuito.



peer-peer

4. Modello architetturale per reti adattative inter-domain

La conoscenza della topologia di rete a livello inter-domain può giocare un ruolo importante in una architettura di rete adattativa. Infatti, l'applicazione che richiede un servizio di connettività con una certa QoS demanda alla rete le operazioni necessarie perché un tale servizio possa essere erogato. Prendiamo in considerazione una classe di servizio standard, che gli ISP siano disposti a supportare (esempio il servizio VoIP). Sotto la pressione della richiesta di tale servizio da parte del mercato una parte degli AS si possono dotare di sistemi di marcatura di pacchetti (ad esempio del tipo DiffServ) per il trattamento dei flussi relativi alla classe di servizio in esame e/o di meccanismi di allocazione di risorse (ad esempio ReSerVation Protocol, RSVP o Multi Protocol Label Switching, MPLS) per essere in grado di garantire la QoS prevista dalla classe di servizio in esame. La rimanente parte degli AS non è in grado di supportare tale classe di servizio.

La sovrapposizione della nuova classe di servizio alla classe best-effort comporta dei problemi a livello inter-domain. Il routing BGP basato principalmente su politiche che provengono da relazioni contrattuali e sulla scelta del cammino minimo può portare i flussi di traffico ad attraversare AS che non supportano la nuova classe di servizio. Abbiamo sostanzialmente tre casi.

1. Il cammino che supporta la QoS richiesta tra sorgente e destinazione non esiste.
2. Il cammino che supporta la QoS richiesta tra sorgente e destinazione esiste e coincide con quello calcolato dal protocollo di routing inter-domain
3. Il cammino che supporta la QoS richiesta tra sorgente e destinazione esiste e ma non coincide con quello calcolato dal protocollo di routing inter-domain

I casi 1 e 2 sono di poco interesse. Infatti, nel primo caso è tecnicamente impossibile assicurare la QoS richiesta dalla connessione e per adeguarsi alle richieste del mercato si dovranno prendere accordi con

altri provider per coprire le zone o i servizi richiesti. Nel secondo caso la QoS viene assicurata dalle tecniche standard supponendo che gli ISP siano in grado di propagare la segnalazione che mira a garantire una certa QoS lungo tutto il percorso. Il caso 3 è invece interessante, in quanto in questo caso le potenzialità tecniche per offrire il servizio sono presenti nella rete ma il protocollo di routing non fa le scelte adeguate.

La verifica di quale delle tre situazioni sopra esposte si applica al routing tra due destinazioni date è un problema difficile da risolvere con i protocolli attualmente disponibili. Proposte per algoritmi di routing inter-domain che siano in grado di supportare il transito delle informazioni di QoS sono state proposte (Xiao et al. 2002; Bonaventure, 2001; Cristallo e Jacquenet, 2002) ma ancora non è stata trovata una proposta soddisfacente che coniughi flessibilità e scalabilità.

Il modello che si propone prevede una adattamento del routing inter-domain basato sulla conoscenza della rete e guidato dall'autonomous system di residenza della stazione destinataria del traffico.

Consideriamo l'esempio in figura 12 in cui gli AS sono numerati e i collegamenti sono etichettati con "peers" se la relazione e di tipo peer-peer o con i ruoli degli estremi ("customer" o "provider") se la relazione è di tipo customer-provider.

Il destinatario AS1 desidera ricevere traffico dalla sorgente AS2 con una certa QoS. Perché questo possa avvenire tutti gli AS intermedi devono supportare tale classe di servizio. Supponiamo che l'AS4 sia l'unico a non supportare la classe di servizio in questione. Il protocollo di routing BGP, usato secondo le politiche convenzionali, in questa rete sceglierebbe il cammino AS2 AS4 AS3 AS1 che non assicura la QoS richiesta. Il cammino AS2 AS5 AS6 AS3 AS1 invece, benché più lungo e contrario alle politiche di preferenza tipiche, è in grado di supportare la QoS richiesta. AS1 può forzare il routing BGP annunciando un cammino in cui è fittiziamente inserito AS4. Poiché BGP evita di propagare gli annunci ad AS presenti nell'AS-path tale annuncio non viene propagato da AS3 ad AS4 ma solo ad AS6.

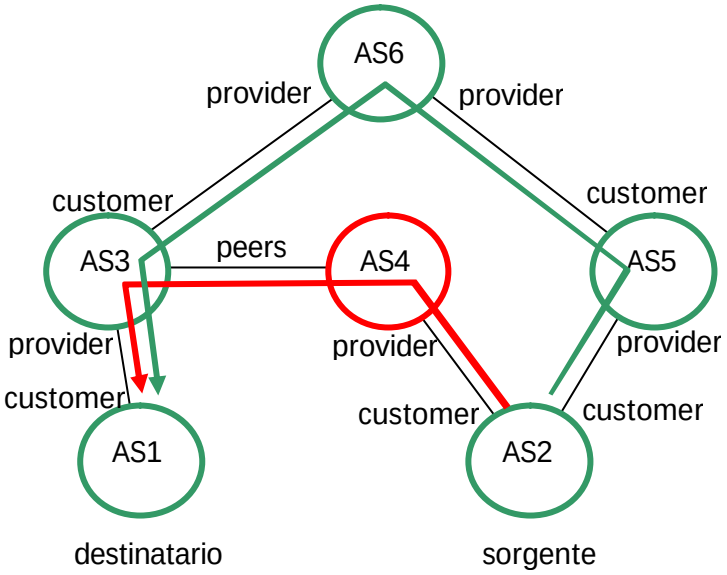


Fig. 12 Un esempio in cui la rete ha la capacità di fornire un servizio con una certa QoS (mediante il path AS2 AS5 AS6 AS3 AS1) ma il protocollo BGP fornisce un percorso (AS2 AS4 AS3 AS1) in cui è presente un AS che non supporta la QoS richiesta. Le frecce indicano il flusso del traffico, gli annunci BGP fluiscono in verso opposto.

In generale, supponendo di avere sufficienti informazioni sulla rete a livello inter-domain, chi origina un certo prefisso per cui desidera una certa QoS in ingresso può intraprendere delle azioni che gli permettano di ottenere dal routing inter-domain un cammino che sia compatibile con la QoS richiesta.
In particolare chi annuncia il prefisso, per poter prendere decisioni, dovrebbero conoscere

- se la classe di servizio richiesta è supportata o meno per ciascun AS della rete,
- la topologia della rete a livello inter-domain,
- le relazioni commerciali tra gli AS.

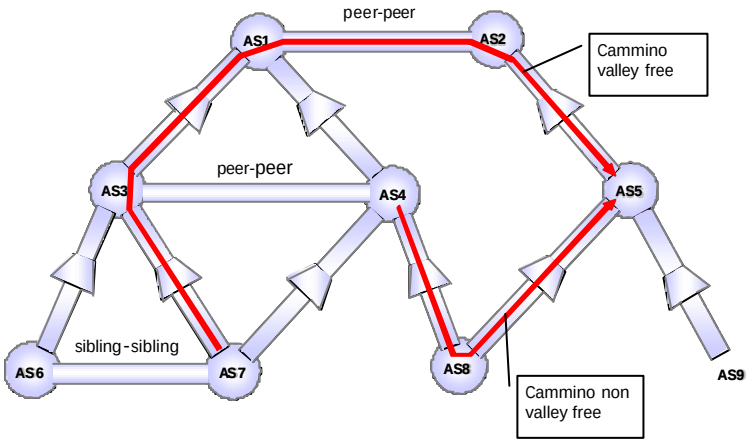
Per quanto riguarda il primo punto si può immaginare o che tale informazione sia disponibile in registri di tipo amministrativo o che venga veicolata con una opportuna estensione del protocollo BGP. Una estensione di questo tipo è stata già proposta in (Bonaventure, 2001). L'interesse per tale problema è tuttora vivo, infatti, in (Nichols e Carpenter 2001, RFC 3086) si propone il concetto di "per domain behavior", una forma aggregata del "per hop behavior" tuttora in fase di standardizzazione per la caratterizzazione in termini di QoS del comportamento delle singole apparecchiature.
Il problema della topologia della rete è risolvibile con tecniche di discovery basate sui dati ricavati dal protocollo BGP. Più complesso è ricavare i dati delle relazioni commerciali tra AS. Notiamo come tale informazione sia indispensabile nell'esempio mostrato prima. Infatti, se AS5 fosse stato customer di AS2 non avrebbe lasciato passare il traffico verso AS6. La seguente sezione introduce modelli e tecniche per l'inferenza delle relazioni customer-provider dalle tabelle di routing BGP.

5. Modello per l'inferenza delle relazioni tra Autonomous Systems

Le relazioni commerciali tra AS hanno un impatto sulla configurazione dei router degli AS stessi. Tali configurazioni hanno un immediato effetto sulle tabelle di instradamento dei router BGP. In (G. Di Battista, et al. 2003) è descritto un algoritmo per l'inferenza delle relazioni customer-provider tra AS da tabelle di routing BGP.

Brevemente i principi su cui tale algoritmo si basa sono i seguenti. Consideriamo il grafo di connettività degli AS in cui ciascun nodo è un AS e esiste un arco tra due AS e c'è un relazione di qualsiasi tipo tra di loro. Una tabella BGP può essere pensata come un insieme di cammini, scelti da BGP per i flussi di traffico, nel grafo di connettività degli AS. Se supponiamo che le uniche relazioni presenti siano quella customer-provider e quella peer-peer e che tutti gli AS configurano correttamente i loro router.

Diciamo che un cammino è valley free se, nella sequenza degli AS, ciascun passo attraversa solo collegamenti nel verso customer→provider fino a raggiungere una "vetta" in cui viene attraversata al più una relazione peer-peer e quindi ciascun passo attraversa solo collegamenti da provider→customer. Si può dimostrare che sotto le precedenti ipotesi in una tabella BGP tutti i cammini sono valley free. Intuitivamente in un cammino non valley free esiste almeno un AS che è customer rispetto ad altri due AS, e quindi compra il servizio di connettività da essi, e contemporaneamente effettua per essi un servizio di transito tra di loro. Tale situazione è contraddittoria dal punto di vista economico e non si verifica in reti correttamente configurate. Questa situazione può essere schematizzata come nella figura seguente.



Nel caso in cui si considerino solo relazioni customer-provider la soluzione dell'inferenza consiste in una orientazione (nel verso customer-provider) degli archi grafo di connettività degli AS. Data una tabella di routing BGP non è immaginabile che esista una orientazione che permetta di rendere i cammini della tabella di routing tutti valley free. Questo accade per vari motivi:

- in Internet altre relazioni oltre quella customer-provider sono adottate che quindi ricadono fuori dal modello qui esposto,

– non tutti i router sono configurati correttamente.

Ci si pone l'obiettivo di massimizzare il numero di cammini soddisfatti. Formalmente si definisce il seguente problema di ottimizzazione.

TYPE OF RELATIONSHIPS (ToR): Dato un insieme di AS-path P e chiamato G il grafo che è unione di tutti i cammini di P dare una orientazione degli archi di G tali che il numero di AS-path validi (cioè valley free) in P sia massimo.

Si può dimostrare che il problema ToR è NP-hard. La prova di NP-hardness è basata su una riduzione del problema MAX2SAT (soddisfacibilità del numero massimo di clausole di un insieme di clausole con 2 letterali).

Consideriamo ora il seguente problema di decisione.

Dato un insieme di AS-path P e chiamato G il grafo che è unione di tutti i cammini di P decidere se è possibile dare una orientazione degli archi di G tali che tutti gli AS-path siano valley free.

Tale problema, restrizione del problema ToR, è invece risolvibile in tempo lineare mediante una riduzione ad una istanza di 2SAT (soddisfacibilità di un formula in forma normale congiuntiva in cui tutte le clausole contengono due letterali). Il problema 2SAT è risolvibile in tempo lineare con l'uso di tecniche standard (Mehlhorn 1984).

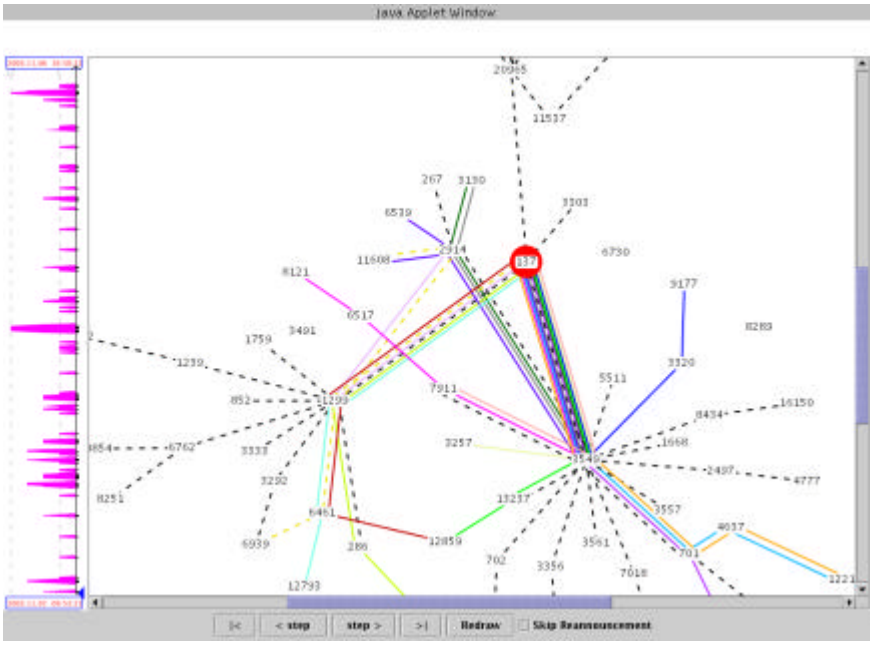
Tale teoria ha permesso di sviluppare una soluzione euristica al problema ToR. La soluzione prevede come risultato un insieme massimale di cammini valley-free. I primi risultati sperimentali mostrano che Il numero di cammini che resta fuori dall'insieme massimale (ciascuno, se inserito nell'insieme, dei quali da luogo a una soluzione non valley-free) è meno dell'1%. Ulteriori dettagli si possono trovare in (Di Battista, Patrignani, Pizzonia 2003, allegato).

6. Visualizzazione del routing inter-domain

Al fine di raggiungere una comprensione più accurata della struttura del routing Internet e della sua evoluzione si sono sviluppati sistemi per la visualizzazione delle informazioni di routing disponibili on-line. Il sistema BGPlay, descritto in (Di Battista, Mariani, Patrignani, Pizzonia, 2003, allegati) permette di visualizzare il routing per un certo prefisso e, mediante un'animazione grafica, la sua variazione in un intervallo di tempo fornito dall'utente.

Molti dati sugli aggiornamenti del routing BGP sono collezionati dal progetto RouteViews (University of Oregon) e dal Routing Information Service (RIPE NCC) e pubblicati per usi scientifici o di debugging di rete.

BGPlay integra tali dati e fornisce una visualizzazione del routing all'istante iniziale dell'intervallo di tempo specificato. Permette quindi di animare le variazioni del routing o di spostarsi in un istante di tempo qualsiasi all'interno dell'intervallo. La seguente è una snapshot del sistema BGPlay che mostra il routing del prefisso 193.204.0.0/15. Sulla sinistra la *time line* mostra la densità di eventi relativi al prefisso nell'intervallo di tempo considerato (2/11/2003-6/11/2003). Nella parte centrale viene mostrato la connettività tra gli AS (al centro l'AS del prefisso considerato). Le linee tratteggiate mostrano cammini che non cambiano nell'intervallo di tempo considerato. Le linee piene mostrano invece cammini che hanno subito cambiamenti. Nella parte bassa una serie di comandi permette di spostare l'istante temporale e quindi animare i cammini.



Il sistema è stato adottato dalle due organizzazioni che collezionano dati. BGPlay è quindi entrato a far parte degli strumenti per interrogare i loro database. La seguente tabella riassume le organizzazioni, i progetti e l'URL per l'accesso a BGPlay.

Organizzazione	**Progetto**	**Accesso a BGPlay**
RIPE NCC	Routing Information Service (RIS) http://www.ris.ripe.net	http://www.ris.ripe.net/bgplay
University of Oregon	RouteViews http://www.routeviews.org	http://bgplay.routeviews.org

La risposta dell'utenza è stata positiva. La seguente tabella riassume alcune statistiche di uso di BGPlay nelle due installazioni.

Installazione	Date di inizio periodo	Data fine periodo	Interrogazioni	media giornaliera	picco giornaliero
RIS	5 aprile 2004	7 ottobre 2004 (185 giorni)	12752	69	1330 (12 maggio 2004)
RouteViews	22 maggio 2004	10 ottobre 2004 (141 giorni)	11356	80	459 (1 settembre 2004)

7. Bibliografia

O. Bonaventure. "Using BGP to Distribute Flexible QoS Information". Internet Draft draft-bonaventure-bgp-qos-00.txt. February 2001.

J. Choe, N. Shroff. "A central limit theorem based approach to analyze queue behavior in ATM networks," IEEE/ACM *Transactions on Networking*, vol. 6, no. 5, pp. 659-671, Oct. 1998.

G. Di Battista, F. Mariani, M. Patrignani, M. Pizzonia, "BGPlay: a System for Visualizing the Interdomain Routing Evolution", to appear in Giuseppe Liotta, editor, Graph Drawing (Proc. GD '03), Lecture Notes Comput. Sci., Springer-Verlag. (allegato)

G. Di Battista, F. Mariani, M. Patrignani, M. Pizzonia. Archives of BGP Updates: Integration and Visualization. International Workshop on Inter-domain Performance and Simulation. IPS 2003. On-line http://www.ist-intermon.org/workshop/. (allegato)

G. Di Battista, M. Patrignani, M. Pizzonia. Computing the Type of Relationships Between Autonomous Systems. IEEE INFOCOM 2003, The Conference on Computer Communications, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies. (allegato)

ETSI: Universal Mobile Telecommunications System, Procedures for the choice of Radio Trasmission Tecnologies of the UMTS – UMTS 30.03 edition 3,1,0- April 1999.

L. Gao. On inferring autonomous system relationships in the internet. IEEE/ACM Transactions on Networking, vol. 9, no. 6, pp. 733–745, Dec 2001.

G. Huston. "Interconnection, Peering and Settlements", Proc INET, 1999.

C. Li and E. W. Knightly. "Coordinated Multihop Scheduling: A Framework for End-to-End Services," IEEE/ACM *Transactions on Networking*, vol. 10, no. 6, December 2002.

K. Mehlhorn. Data Structures and Algorithms. Springer Publishing Company, 1984, vol. 1-3.

K. Nichols, B. Carpenter. Definition of Differentiated Services Per Domain Behaviors and Roules for their Speicifications. RFC 3086. April 2001

E. Knightly. "Second moment resource allocation in multiservice networks," *in Proceedings of* ACM SIGMETRICS '97, pages 181-191, Seattle, WA, June 1997.

E. Knightly. "Enforceable quality of service guarantees for bursty traffic streams," *in Proceedings of* IEEE INFOCOM '98, San Francisco, CA, Mar. 1998.

J. Qiu, E. Knightly. "Inter-Class Resource Sharing using Statistical Service Envelopes," *in Proceedings of* IEEE INFOCOM '99, New York, NY, March 1999.

Computing the Types of the Relationships between Autonomous Systems

Giuseppe Di Battista, Maurizio Patrignani, and Maurizio Pizzonia
Dipartimento di Informatica e Automazione, Università di Roma Tre, Rome, Italy
Email: {gdb,patrigna,pizzonia}@dia.uniroma3.it

***Abstract*— We investigate the problem of computing the types of the relationships between Internet Autonomous Systems. We refer to the model introduced in [1], [2] that bases the discovery of such relationships on the analysis of the AS paths extracted from the BGP routing tables. We characterize the time complexity of the above problem, showing both NP-completeness results and efficient algorithms for solving specific cases. Motivated by the hardness of the general problem, we propose heuristics based on a novel paradigm and show their effectiveness against publicly available data sets. The experiments put in evidence that our heuristics performs significantly better than state of the art heuristics.**

I. Introduction

An *Autonomous System* (AS) is a portion of Internet under a single administrative authority. Currently, there are more than $10,000$ ASes and their number is rapidly growing. They interact to coordinate the IP traffic delivery, exchanging routing information with a protocol called Border Gateway Protocol (BGP) [3].

Several authors (see, e.g. [4], [5]) have pointed out that the relationships between ASes can be roughly classified into categories that have both a commercial and a technical flavor. A pair of ASes such that one sells/offers Internet connectivity to the other is said to have a *provider-customer* relationship. If two ASes simply provide connectivity between their respective customers are said to have a *peer-to-peer* relationship. Finally, if two ASes offer each other Internet connectivity are said to be *siblings*. Of course, this classification does not capture all the shades of the possible commercial agreements and technical details that govern the traffic exchanges between ASes but should be considered as an important attempt toward understanding the Internet structure.

Since many applications would benefit from the knowledge about the Internet structure, the research on the subject has recently produced many contributions. More specifically, there is a wide research area focusing on the discovery of the topology underlying the Internet structure, either at the AS and at the router level (see, for example, [6], [7], [8]).

Other researchers concentrate more directly on the above mentioned relationships and on the hierarchy that they induce on the set of ASes. Govindan and Reddy [6] study the interplay between the *degree* of the ASes and their rank in the hierarchy, where the degree of an AS is the number of ASes that have some kind of relationship with it. Gao [1] studies, for the first time, the following problem. ASes are the vertices of a graph (*AS graph*) where two ASes are adjacent if they exchange routing information; the edges of such a graph should be labeled in order to reflect the type of relationship they have. In order to infer the relationships between ASes, Gao uses the information on the degree of ASes together with the *AS paths* extracted from the BGP routing tables. An AS path is the sequence of the ASes traversed by a connectivity offer (*BGP announcement*). In [1] a heuristic is presented together with experimental results. An analysis on the properties of the labeled graphs obtained with such heuristics is provided in [9].

Subramanian et al. [2] formally define, as a minimization problem, a slightly simplified version of the problem addressed in [1] and conjecture its NP-completeness. They also propose a heuristic based on the observation of the Internet from multiple vantage points, which does not rely on the degree of the ASes. Further, they validate the results obtained by the heuristic against a rich collection of data sets.

This paper contributes to the line of research opened in [1], [2]. Namely, its main results are the following.

- We solve a problem explicitly stated in [2]. Namely, we characterize the complexity of determining the relationships between ASes while minimizing the number of "anomalies". In particular:
  - We show that such a problem is NP-complete in the general case;
  - We produce a linear time algorithm for determining the AS relationships in the case in which the problem admits a solution without anomalies; and
  - We use such a linear time algorithm to show that for large portions of the Internet (e.g., data obtained from single points of view) it is often possible to determine the relationships between ASes with no anomalies.
- We introduce heuristics, based on a novel approach, for determining the relationships between ASes with a small number of anomalies.
- We experimentally show that the proposed approach leads to heuristics that performs significantly better than the cutting edge heuristics of [2].

Work partially supported by European Commission - Fet Open project COSIN - COevolution and Self-organisation In dynamical Networks - IST-2001-33555, by "Progetto ALINWEB: Algoritmica per Internet e per il Web", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR Fondo per gli Investimenti della Ricerca di Base.

The paper is structured as follows. Section II describes the addressed problem. Sections III and IV show an algorithm for testing if the problem admits a solution with no anomalies, and show how to find a solution if it exists. In Section V we prove the NP-completeness of the problem in the general case. Section VI shows new heuristics and compare the results with the state of the art. Finally, Section VII contains conclusions and open problems.

II. Problem Description

A *prefix* is a block of destination IP addresses. An Internet Autonomous System (AS) applies local policies to select the best *route* for each prefix and to decide whether to *export* this route to neighboring ASes.

Several authors have pointed out that ASes typically have *provider-customer* or *peer-to-peer* relationships (see, e.g. [4], [5], [10], [2]). A *customer* exports to a provider its routes and the routes learned from its own customers, but does not export routes learned from other providers or peers. A *provider* exports to a customer its routes, the routes learned from the other customers, its providers, and its peers. *Peers* export to each other their own routes and the routes learned from their customers but do not export the routes learned from their providers and other peers.

Consider the *AS paths* that are associated with the BGP announcements of the routes. If all the ASes adopted export policies according to the above model, then the AS paths would have a peculiar structure [1], [2]. Namely, (1) no AS path can contain more than one pair of ASes having a peer-to-peer relationship; and (2) once a provider-customer or a peer-to-peer pair of ASes is met in the AS path, no customer-provider can be found in the remaining part of it.

Further, the above mentioned peculiarities of the AS paths have been formally stated in a theorem of [1], that has been also re-casted in [2]. A graph-theoretic formulation of the same theorem will be given in what follows.

A. Type-of-Relationship problem

The relationships between ASes in the Internet may be represented as a graph G whose edges are either directed or undirected. Each vertex is an AS, a directed edge from vertex u to vertex v indicates that u is a customer of v (provider-customer relationship), and an undirected edge between vertex w and vertex z indicates that w and z are peers (peer-to-peer relationship). A BGP AS path corresponds to a path on G. Suppose path p is composed by the sequence of vertices $v_1, \ldots, v_n$, then p is *valid* if it is of one of the following two types.

Type 1: p is composed by a (possibly empty) sequence of forward edges followed by a (possibly empty) sequence of backward edges; more formally, there exists a vertex v_i of p such that for $j \in 1, \ldots, i-1$ edge (v_j, v_{j+1}) is directed from v_j to v_{j+1} and for $j \in i, \ldots, n-1$ edge (v_j, v_{j+1}) is directed from v_{j+1} to v_j. (See Figure 1.a).

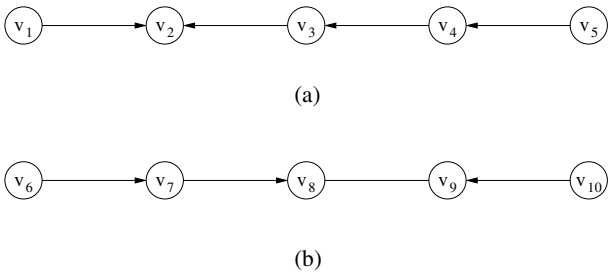


Fig. 1. An example of Type 1 (a) and of Type 2 (b) path.

Type 2: p is composed by a (possibly empty) sequence of forward edges, followed by an undirected edge, followed by a (possibly empty) sequence of backward edges; more formally, there exists a vertex v_i of p such that for $j \in 1, \ldots, i-1$ edge (v_j, v_{j+1}) is directed from v_j to v_{j+1}, edge (v_i, v_{i+1}) is undirected, and for $j \in i+1, \ldots, n-1$ edge (v_j, v_{j+1}) is directed from v_{j+1} to v_j. (See Figure 1.b).

See Figure 1 for examples of Type 1 and of Type 2 paths. An *invalid path* is a path that is not valid.

At this point the above mentioned theorem [2] can be restated as follows: if every AS obeys the customer, peer, and provider export policies, then every advertised path is either of Type 1 or of Type 2.

However, the Internet is more complex. To give a few examples: ASes operated by the same company can have a *sibling* relationship, where each AS exports all its routes to the other; two ASes may agree a *backup* relationship between them, to overcome possible failures; or ASes may have peering relationships through intermediate ASes. However, finding out which is the portion of Internet that obeys the customer, peer, and provider export policies can be considered as the first step toward a complete comprehension of the relationships between ASes. Such motivations have pushed the authors of [2] toward identifying the following problem.

> *Type-of-Relationship (ToR) Problem* [2]: Given an undirected graph G and a set of paths P, give an orientation to some of the edges of G to minimize the number of invalid paths in P.

Figure 2 shows an instance of the ToR problem for which an orientation without invalid paths cannot be found. In particular, each orientation of edge (AS701, AS5056) yields at least one invalid path. Suppose, in fact, that edge (AS701, AS5056) was directed from AS701 to AS5056. Path AS5056, AS701, AS4926, AS6461, AS2914, AS174, AS14318 (drawn solid in the figure) would be valid only if edge (AS4926, AS6461) was directed from AS6461 to AS4926. Similarly, path AS5056, AS701, AS6461, AS4926, AS4270, AS4387 (drawn dotted in the figure) would be valid only if edge (AS4926, AS6461) was directed from AS4926 to AS6461. Hence, we have a contradiction, since edge (AS4926, AS6461) should have opposite orientations. Now, suppose that edge (AS701, AS5056) was undirected. The same arguments apply, leading to the same contradiction. Finally, suppose that edge

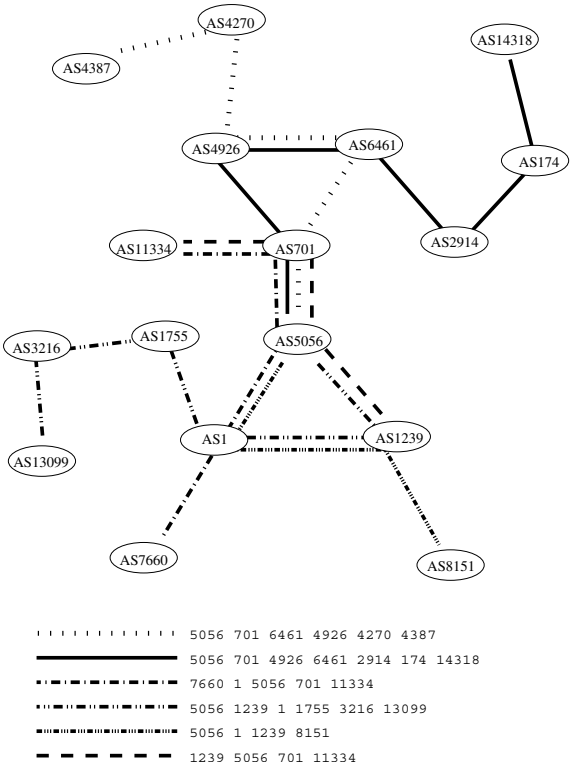


Fig. 2. An instance of the ToR problem that does not admit an orientation without invalid paths. The six paths of the instance are represented with different line styles.

(AS701, AS5056) was directed from AS5056 to AS701. It is easy to see that in this case we have a contradiction on the orientation of edge (AS1, AS1239).

Figure 3 shows an instance of the ToR problem that admits an orientation without invalid paths. Figures 4 and 5 show a possible orientation.

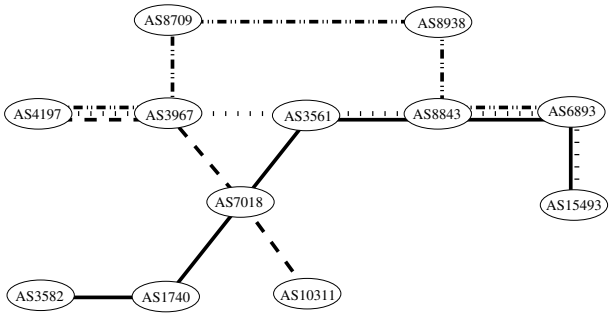


Fig. 3. An instance of the ToR problem that admits an orientation without invalid paths. The four paths of the instance are represented with different line styles.

B. Simplifying the problem

The Type-of-Relationship Problem is a minimization problem. In order to studying it, following a standard technique [11], we consider its corresponding decision version as follows.

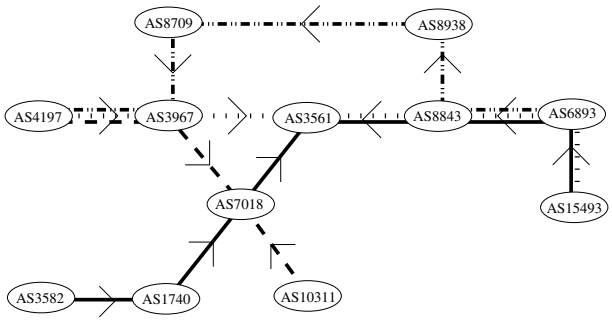


Fig. 4. An orientation for the graph of Figure 3. Note that all the paths are valid.

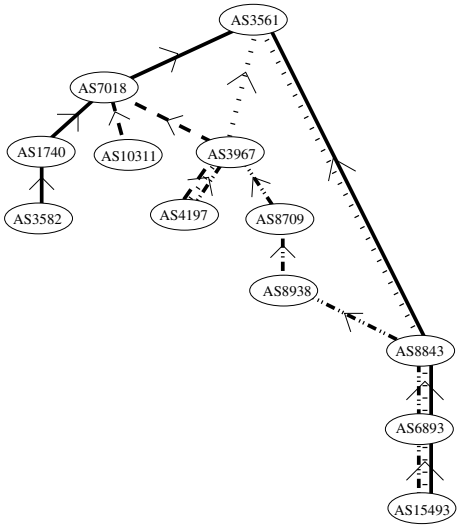


Fig. 5. The directed graph of Figure 4 is drawn in such a way to emphasize the hierarchical relationships induced by the orientation.

> *ToR-D Problem*: Given an undirected graph G, a set of paths P, and an integer k, test if it is possible to give an orientation to some of the edges of G so that the number of invalid paths in P is at most k.

One of the ingredients that make the ToR-D problem difficult is the presence of both directed and undirected edges. Fortunately, the problem can be simplified by "ignoring" the undirected edges, without loosing its generality. Namely, the ToR-D problem admits a solution if and only if the following simpler problem admits one.

> *ToR-D-simple Problem*: Given an undirected graph G, a set of paths P, and an integer k, test if it is possible to give an orientation to *all* the edges of G so that the number of invalid paths in P is at most k.

Notice that the ToR-D-simple problem considers Type 1 paths only.

In fact, consider an orientation of the edges of G that is a solution for the ToR-D-simple problem. It is clear that the same orientation is also a solution for the ToR-D problem. Conversely, consider an orientation of some of the edges of G that is a solution for the ToR-D problem and let (u, v) be an edge of G that is undirected. Consider any path p of P

through (u, v). Two cases are possible: either p is valid or p is invalid.

If p is valid (see Figure 6), then it is a Type 2 path and all the edges of p preceding u are forward edges, while all the edges of p following v are backward edges. If (u, v) is arbitrarily oriented, then the only effect on p is of transforming it from Type 2 to Type 1. Hence, the number of invalid paths does not increase. If p is invalid and (u, v) is arbitrarily oriented either it becomes valid or it remains invalid. In this case the number of invalid paths does not increase. The same process can be repeated on all the undirected edges, until an orientation of G that is a solution for ToR-D-simple is found.

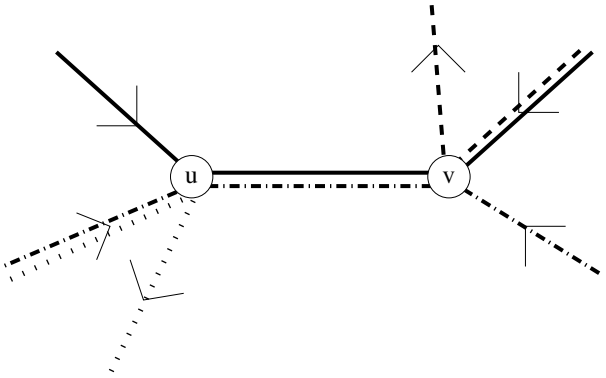


Fig. 6. An undirected edge (u, v) of a solution of the ToR-D problem. The two paths traversing (u, v) are represented one with a solid line and the other with a dot-dashed line. Both paths are valid.

To better understand the relation between the two problems, observe that the above consideration suggests that for each partial orientation of G that is a solution of ToR-D with n undirected edges there exist 2^n orientations that are a solution for ToR-D-simple.

Further, we can pick an orientation that is a solution for ToR-D-simple and consider it as a solution for ToR-D. Then, we can refine such a solution looking for edges whose orientation can be removed without increasing the number of anomaly paths. A necessary and sufficient condition, that is also easy to test, for removing the orientation of a single directed edge (u, v) is the following. Consider all the paths through (u, v) and all the edges following (u, v) in such paths. Edge (u, v) can be made undirected if such edges are all directed toward v.

The above discussion justifies a two steps approach where in the first step a solution is found for ToR-D-simple and in the second step peering edges are discovered.

III. Testing whether an AS Graph Admits a Hierarchical Structure without Path Anomalies

In Section II we have seen that the problem of detecting the types of relationships between ASes can be tackled by studying the ToR problem, its decision version ToR-D, and a simpler problem called ToR-D-simple. The relations among such problems have also been discussed. In this section we show that problem ToR-D-simple (and, consequently, ToR-D) can be solved efficiently when $k = 0$, that is when we want to check if G admits an orientation where all the paths are valid (i.e., there are 0 invalid paths).

A. Path anomalies and boolean formulas

Observe that a path p on G composed by the sequence of vertices $v_1, \ldots, v_n$ is of Type 1 if and only if it does not exist a vertex v_i $(i = 2, \ldots, n-1)$ of p such that the two edges of p incident on v_i are directed away from v_i. Hence, to impose that p is valid it suffices to rule out such a configuration. Based on this observation ToR-D-simple can be mapped to a 2SAT problem [11].

In the 2SAT problem you are given a set X of boolean variables and a formula in conjunctive normal form. Such a formula is composed by clauses of two literals, where a literal is a variable or a negated variable. You are asked to find a truth assignment for the boolean variables in X so that the formula is satisfied.

The mapping of ToR-D-simple to 2SAT is a two step process. First, all the edges of G are arbitrarily (for example randomly) oriented. Second, a boolean formula is constructed so to represent the constraints that each path imposes on the orientation of G in order to be a path of Type 1. The construction is performed as follows.

- For each directed edge (v_i, v_j) of G a variable $x_{i,j}$ is introduced. A true value for $x_{i,j}$ means that, in the final orientation, (v_i, v_j) will be directed from v_i to v_j (that is, the direction of the initial arbitrary orientation will be preserved), while a false value means that (v_i, v_j) will be directed from v_j to v_i (that is, the direction of the initial arbitrary orientation will be reversed).
- Consider a path $p \in P$ and three consecutive vertices v_{i-1}, v_i, v_{i+1} of p. Four cases are possible, according to the arbitrary orientations that we have given to the edges between v_{i-1}, v_i, and v_{i+1}.
  - Both edges are directed toward v_i, i.e. such directed edges are (v_{i-1}, v_i) and (v_{i+1}, v_i). We introduce clause $x_{i-1,i} \vee x_{i+1,i}$.
  - Both edges are directed away from v_i, i.e. such directed edges are (v_i, v_{i-1}) and (v_i, v_{i+1}). We introduce clause $\overline{x}_{i,i-1} \vee \overline{x}_{i,i+1}$.
  - One edge is directed toward v_i and the other toward v_{i+1}, i.e. such directed edges are (v_{i-1}, v_i) and (v_i, v_{i+1}). We introduce clause $x_{i-1,i} \vee \overline{x}_{i,i+1}$.
  - One edge is directed toward v_{i-1} and the other toward v_i, i.e. such directed edges are (v_i, v_{i-1}) and (v_{i+1}, v_i). We introduce clause $\overline{x}_{i,i-1} \vee x_{i+1,i}$.

In this way we introduce $n - 2$ clauses for each path of P with n vertices. We impose that all the constraints are simultaneously satisfied by considering the boolean "and" of all the clauses. Since each clause has two literals, we have mapped the ToR-D problem to a 2SAT formula.

As an example consider a path composed by five vertices $v_1, \ldots, v_5$ and suppose that the initial orientation step has

TABLE I

TELNET LOOKING GLASS SERVERS AND CORRESPONDING AS GRAPHS.

AS #	AS Name	Apr 18, 2001			Apr 6, 2002		
		# Vertices	# Edges	# Paths	# Vertices	# Edges	# Paths
1	Genuity	10,203	13,001	58,156	12,700	15,946	63,744
1740	CERFnet	10,007	13,416	70,830	not available		
3549	Globalcrossing	10,288	13,039	60,409	12,533	16,025	76,572
3582	U. of Oregon	10,826	22,440	2,584,230	13,055	27,277	4,600,981
3967	Exodus Comm.	10,387	18,401	254,123	12,616	21,527	339,023
4197	Global Online Japan	10,288	13,004	55,060	12,518	15,628	59,745
5388	Energis Squared	10,411	13,259	58,832	12,659	16,822	117,003
7018	AT&T	9,252	12,117	120,283	11,706	15,429	170,325
8220	COLT Internet	8,376	10,932	46,606	12,660	18,421	154,855
8709	Exodus, Europe	10,333	15,006	114,931	12,555	18,175	126,370

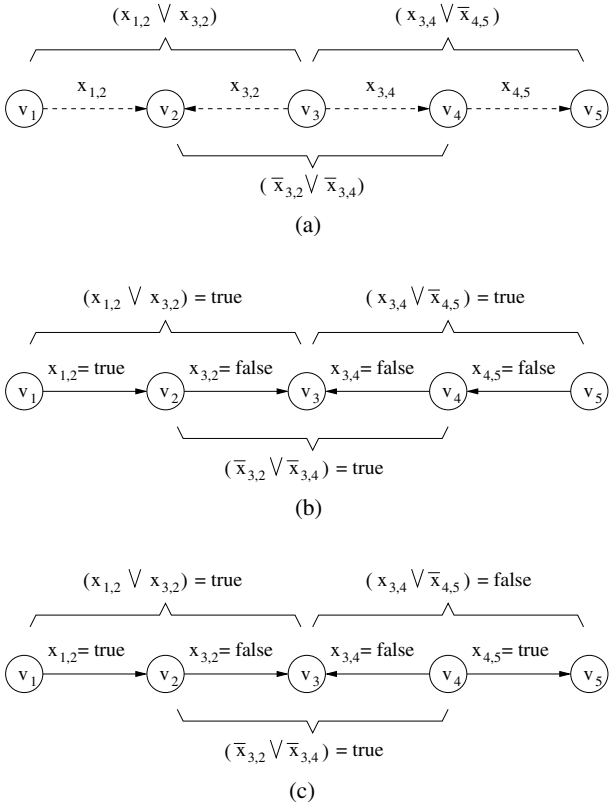


Fig. 7. (a) An initial orientation for a five vertices path and the boolean variables associated with its edges. The orientation shown in (b), which makes the path valid, corresponds to the truth assignment $x_{1,2} = true$, $x_{3,2} = false$, $x_{3,4} = false$, and $x_{4,5} = false$, which satisfies formula $(x_{1,2} \vee x_{3,2}) \wedge (\overline{x}_{3,2} \vee \overline{x}_{3,4}) \wedge (x_{3,4} \vee \overline{x}_{4,5})$ associated with the path. Conversely, the orientation shown in (c), which makes the path invalid, corresponds to the truth assignment $x_{1,2} = true$, $x_{3,2} = false$, $x_{3,4} = false$, and $x_{4,5} = true$, which does not satisfy the formula.

given to the edges of the path a direction as follows: (v_1, v_2), (v_3, v_2), (v_3, v_4), and (v_4, v_5). We have variables $x_{1,2}$, $x_{3,2}$, $x_{3,4}$, and $x_{4,5}$ (see Figure 7.a). Applying the above procedure we obtain the following 2SAT formula: $(x_{1,2} \vee x_{3,2}) \wedge (\overline{x}_{3,2} \vee \overline{x}_{3,4}) \wedge (x_{3,4} \vee \overline{x}_{4,5})$. Consider the truth assignment $x_{1,2} = true$, $x_{3,2} = false$, $x_{3,4} = false$, and $x_{4,5} = false$. It is easy to see that it satisfies the formula and that it corresponds to an orientation of the edges of the path toward vertex v_3 (see Figure 7.b). On the other hand, consider the truth assignment $x_{1,2} = true$, $x_{3,2} = false$, $x_{3,4} = false$, and $x_{4,5} = true$. It is easy to see that it does not satisfy the formula and that it corresponds to an orientation of the edges of the path that is not consistent with Type 1 (see Figure 7.c).

B. Computational aspects

Problem 2SAT may be efficiently solved by using the well known result in [12] that maps 2SAT into a problem on a suitable directed graph G_{2SAT}. Observe that G and G_{2SAT} are different graphs.

Although this result is clearly illustrated in the literature, we give here a brief description of it, that will help the reader to better understand the algorithms described in Sections IV, and VI.

Graph G_{2SAT} has two nodes for each boolean variable x of 2SAT, corresponding to its two literals x and $\overline{x}$. Further, for each clause of the form $l_1 \vee l_2$, where l_1 and l_2 are literals, the two directed edges $(\overline{l}_1, l_2)$ and $(\overline{l}_2, l_1)$ are introduced. Intuitively, edge $(\overline{l}_1, l_2)$ represents the logical implication $\overline{l}_1 \rightarrow l_2$, while edge $(\overline{l}_2, l_1)$ represents $\overline{l}_2 \rightarrow l_1$. Problem 2SAT admits a solution if and only if for no variable x there is a directed cycle in G_{2SAT} containing both x and $\overline{x}$ (i.e. a logical contradiction).

Testing, for each variable, if there exists a cycle containing its two literals can be quite time consuming. However, fortunately, the problem of testing for all the variables in 2SAT whether such a cycle exists in G_{2SAT} can be efficiently solved by computing [13] the *strongly connected components* of G_{2SAT} and by testing for each variable if x and $\overline{x}$ are in the same strongly connected component. We recall that a strongly connected component of a directed graph is a maximal set of vertices such that for each pair u, v of vertices of the set there exists a directed path from u to v and vice versa. Computing the strongly connected components of a directed graph can be done in time linear in the size of the graph [13].

From a theoretical point of view, it comes out that ToR-D-simple (and, as a consequence, ToR-D) with $k = 0$, i.e. the problem of deciding if a graph G of n vertices and m edges admits an orientation so that all the paths of a set P are valid, can be solved in $O(n + m + q)$ time, where q is the sum of the lengths of the paths of P.

More practically, we have implemented the above algorithm by exploiting a facility from the Leda [14] software library that efficiently computes the strongly connected component of a directed graph and that labels each vertex of the graph with an integer that identifies the component it belongs to. Hence, testing for a variable x if x and $\overline{x}$ are in the same strongly connected component is performed by testing whether they have the same label.

C. Experiments

This section illustrates the first group of experiments of this paper. Such experiments have the purpose of understanding if at least for partial views of the Internet graph the ToR problem admits a solution without invalid paths. This is important, in our opinion, at least for the following reason. Even if it is unlikely that the entire Internet AS graph could be classified in terms of customer-provider and peer-to-peer relationships without exceptions (and we will see evidence of this in the remainder of this paper), it is unclear if this is possible for what is visible from a specific observation point ("vantage point" in [2]) of the network.

The test bed consists of BGP data sets obtained as follows. Each data set is extracted from the BGP routing table of a Looking Glass server. First, the output of the "show ip bgp" command is collected. Second, a file of AS paths is computed by discarding the prefix column and all the BGP attributes different from the AS path. Duplicate ASes arising from *prepending* [3] are removed in each path. Note that duplicated paths may be present in the set.

There are many Looking Glass servers on the Internet and it is very difficult to say which are the most representative. In order to compare our work with previous results, we have chosen to use the collection of ten BGP data sets obtained from Telnet Looking Glass servers already adopted as a test bed in [2]. Such test beds are periodically collected and publicly distributed by the authors [15].

For each data set we have constructed a different AS graph (a partial view of the global AS graph) by using only the adjacencies contained in the AS paths of the specific data set. Table I shows the main features of the graphs constructed from the ten data sets. Note that values of Tables I and IV of [2] and values computed from data available in [15] (and that are presented in the aforementioned Table I of this paper) appear to be slightly different.

Table II shows the results of the experiments. Observe that for all the partial views, but the one of the University of Oregon server [16], the ToR problem admits a solution without invalid paths. In fact, the server of the University of Oregon is not just a Looking Glass that gives a view of Internet from a specific point of observation, but it offers an integrated view obtained from 52 peering sessions with routers spread on 39 different ASes. This clearly indicates that integrating information from different points of view makes the problem much more difficult.

Figure 8 shows six rows extracted from the routing table of the U. of Oregon dated Apr 18, 2001. Observe that the six paths are exactly those used in Figure 2 to give an example of an instance of the ToR problem that does not admit an orientation without invalid paths.

TABLE II

TESTING IF THE ToR PROBLEM HAS A SOLUTION WITHOUT INVALID PATHS FOR SEVERAL BGP ROUTING TABLES.

AS #	AS Name	Orientable w/o anomalies	
		Apr 18, 2001	Apr 6, 2002
1	Genuity	yes	yes
1740	CERFnet	yes	not available
3549	Globalcrossing	yes	yes
3582	U. of Oregon	no	no
3967	Exodus Comm.	yes	yes
4197	Global Online J.	yes	yes
5388	Energis Squared	yes	yes
7018	AT&T	yes	yes
8220	COLT Internet	yes	yes
8709	Exodus, Europe	yes	yes

It is worth noting that we have conducted all the experiments on a PC Pentium III with 1 GB of RAM. Each of the above experiments required a few seconds of computation time.

IV. COMPUTING THE AS RELATIONSHIPS

In Section III we have seen how problem ToR-D can be solved efficiently when $k = 0$, that is when we want to check if G admits an orientation where all the paths are valid (i.e., there are 0 invalid paths). We can do that solving a simpler problem, called ToR-D-simple.

In this section we deal with the problem of determining the relationships between ASes in the assumption that ToR-D admits a solution without anomalies. Essentially, this is a two steps process. In the first step an orientation that solves ToR-D-simple is computed. In the second step peering relationships are discovered by examining the solution computed for ToR-D-simple.

A. Finding an orientation for ToR-D-simple

If a solution for ToR-D-simple exists, computing it is an easy task. Since we mapped ToR-D-simple to 2SAT, we can find a solution to ToR-D-simple by computing a truth assignment for the boolean variables of the corresponding 2SAT instance. A standard method [12] for computing such assignment is the following. A function $f(v)$ can be computed for all the vertices of the graph G_{2SAT} associated with 2SAT (see Section III-B) such that, for any two vertices u and v, if there exists a directed path from u to v, then $f(u) \leq f(v)$. A true value is assigned to variable x if $f(x) > f(\overline{x})$, a false value otherwise. The satisfiability of 2SAT guarantees that $f(x) \neq f(\overline{x})$.

Function f can be efficiently computed by exploiting the decomposition of the graph into strongly connected components and by computing a special ordering, called *topological sorting* [14], on the directed acyclic graph of the components.

Of course, an instance of the problem ToR-D-simple may admit several different solutions. The structure of the problem constraints some variables to have the same truth values in all

Network	Next Hop	Path
200.1.225.0	167.142.3.6	5056 701 6461 4926 4270 4387 i
200.10.112.0/23	167.142.3.6	5056 701 4926 4926 4926 6461 2914 174 174 174 174 14318 i
204.71.2.0	203.181.248.233	7660 1 5056 701 11334 i
213.172.64.0/19	167.142.3.6	5056 1239 1 1755 1755 1755 1755 3216 13099 i
200.33.121.0	167.142.3.6	5056 1 1239 8151 i
204.71.2.0	144.228.241.81	1239 5056 701 11334 i

Fig. 8. Six rows extracted from the BGP routing table of the U. of Oregon dated Apr 18, 2001. Each orientation of the edges of the corresponding graph yields at least one invalid path.

the solutions, while other variables may assume any true/false assignment. Coming back to problem ToR-D-simple, this means that some edges have a constrained customer-provider orientation, while others may assume different orientations.

Interestingly, the proposed approach permits to "explore" the solutions space. Namely, if some knowledge is available on the customer-provider relationships between ASes, it is easy to force the solution to respect such constraints. For example, suppose to know in advance that AS v_i is a customer of AS v_j and suppose that in the initial arbitrary orientation edge (v_i, v_j) is directed from v_i to v_j. We can impose that the solution respects the constraint by adding to the 2SAT formula associated with Problem ToR-D-simple the clause $(x_{i,j} \vee x_{i,j})$. Of course, adding constraints to the problem decreases the size of the solution space and may lead to unsatisfiable instances.

B. Discovering the peering relationships

A solution for the ToR-D-simple problem provides an orientation for all the edges of the AS graph (customer-provider relationships). However, as described in Section II-B, it is possible to refine the obtained solution reintroducing peering relationships. In such a section a sufficient condition has been given for modifying a directed edge into an undirected edge still having a solution for ToR-D.

Several different criteria can be adopted to measure the quality of a solution once peerings are reintroduced. For example, one could say that a solution is especially interesting if many peering have been discovered. Unfortunately, it can be shown that, given a solution for a ToR-D-simple instance, i.e., with no peerings, the problem of producing a solution for the corresponding ToR-D instance that maximizes the number of the peering edges is a hard one.

We prove it using a reduction from the INDEPENDENT-SET problem, in which you are given a graph with nodes in N and arcs in A and you are asked to find a subset of the nodes of size k such that no two nodes of the subset are adjacent. To build the instance of the ToR-D-simple problem corresponding to the instance of the INDEPENDENT-SET problem we introduce an edge (v_i, v_{top}) for each node $n_i \in N$ and we introduce a path v_i, v_{top}, v_j for each arc $(n_i, n_j) \in A$. The edges of the ToR-D-simple instance can be directed toward vertex v_{top} in order to have a solution with no invalid path. It can be easily shown that the problem of reintroducing k peering edges without increasing the number of invalid paths is equivalent to the problem of finding an independent set of size k. We omit the details of the proof for the sake of brevity.

V. The Difficulty of Minimizing Path Anomalies

The ToR problem was conjectured to be NP-complete in [2]. In Section III we have shown that finding a solution with zero invalid path (provided that it exists) is a tractable problem. In this section we show that the ToR problem is NP-complete in the general case, that is, when it does not admit an orientation without invalid paths. In order to prove that the ToR problem is NP-hard we reduce the NP-complete problem MAX2SAT to it.

In the remaining part of this section, following a standard technique when dealing with optimization problems [11], we refer to their decision versions. For ToR we already defined in Section II the ToR-D and ToR-D-simple problems (we will use the latter one). As for MAX2SAT, its decision version of MAX2SAT-D can be defined as follows. You are given a set X of boolean variables and a collection C of disjunctive clauses, each one of 2 literals, where a literal is a variable or a negated variable. You are asked to find a truth assignment for the boolean variables in X so that the number of unsatisfied clauses of C is at most k, where k is a positive integer.

Given an instance of the MAX2SAT-D problem, we will produce an instance of the ToR-D-simple problem, such that an orientation with k invalid paths exists iff an assignment with k unsatisfied clauses of MAX2SAT-D can be found. For each variable $x_i \in X$ we introduce two vertices x'_i and x''_i. For each clause $l_1 \vee l_2$ we introduce a path of four vertices as follows. If l_1 is the negated literal of variable x_i, then the first two vertices of the path will be x''_i and x'_i, otherwise they will be x'_i and x''_i. Similarly, if l_2 is the negated literal of variable x_j, then the last two vertices of the path will be x'_j and x''_j, otherwise they will be x''_j and x'_j.

Figure 9 shows an example of an instance of the MAX2SAT-D problem and the corresponding instance of the ToR-D-simple problem.

Given an orientation for the edges of the graph, if edge (x'_i, x''_i) is directed from x'_i to x''_i, then we associate a true value with the corresponding boolean variable x_i, otherwise we associate a false value. Observe that, given an orientation for the edges of the graph, if the first edge of the four-vertex path is directed toward the first vertex of the path, then the first literal of the corresponding clause is false. Analogously, if the last edge of the path is directed toward the last vertex of the path, then the second literal of the clause is false.

If a path is valid, then its first and its last edges are not simultaneously directed toward the first vertex and the last vertex, respectively. It follows that, if a path is valid, the corresponding clause is satisfied. Thus, an orientation for the

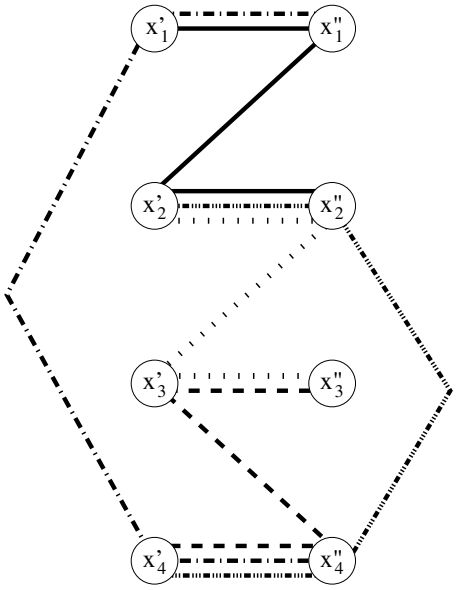


Fig. 9. The instance of the ToR-D-simple problem corresponding to the instance $(x_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3) \wedge (x_2 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_3 \vee x_4)$ of the MAX2SAT-D problem.

edges of the graph with k invalid paths corresponds to an assignment of the boolean variables with k unsatisfied clauses.

Conversely, suppose to have an assignment for the boolean variables in X that leaves k clauses of the MAX2SAT-D instance unsatisfied. If variable x_i is positive, direct edge (x'_i, x''_i) toward x''_i, otherwise direct it toward x'_i. Each satisfied clause corresponds to a four vertex path whose first and last edge are not simultaneously directed toward its first and last vertices, respectively, and an orientation for the intermediate edge of the path can be easily found so that the path is valid. Thus, an assignment for the boolean variables that leaves k clauses of the MAX2SAT-D instance unsatisfied corresponds to an orientation of the ToR-D-simple problem with k invalid paths.

Since it can be shown that the ToR-D-simple problem belongs to the class NP (it is easy to count the invalid paths yielded by a given orientation), it follows that the ToR-D-simple problem is NP-complete.

Observe that, although we used a reduction of the MAX2SAT-D problem to show the NP-hardness of the ToR-D-simple problem, an instance of the ToR-D-simple problem can not be always mapped to an instance of the MAX2SAT-D problem (for example, when two paths have one internal edge in common).

VI. Heuristics for Computing the AS Relationships

In Section V we have seen that the ToR-D problem is computationally hard and in Section III we have seen that, even if portions of Internet admit a hierarchical structure without anomalies, when the data set becomes large, such a "strong" structure does not exist (see, e.g., the AS 3582 in Table II).

This section aims at giving a method for discovering the AS relationships in a big chunk of Internet with a small number of invalid paths. Observe that, even if heuristics are known for solving the MAX2SAT problem (see, e.g., [17]), they cannot be straightforwardly applied to ToR-D. In fact, maximizing the number of satisfied clauses of the 2SAT formula does not necessarily imply maximizing the number of valid paths. Another approach would be to reduce ToR-D to a problem called Maximum Number of Satisfiable Formulas, where a collection of formulas in conjunctive normal form is given, and the target is to maximize the number of satisfied formulas. However, that problem has been shown to be not approximable in [18] and we were not able to find in the literature effective and efficient heuristics for that problem.

As a reference data set, with the purpose of comparing our results with previous contributions, we consider the same portion of Internet taken into account in [2]. Namely, we consider the union of all the paths of the Telnet Looking Glasses of Table I (version of Apr 18, 2001). The total number of paths of the data set is $3,423,460$, involving $10,916$ ASes. The graph of the adjacencies between ASes contains $23,761$ edges. We measure the effectiveness of our heuristics against such quite large data set.

The target of the proposed heuristics is the computation of a maximal set of paths (subset of the given set of paths) such that ToR-D with $k = 0$ admits a solution. A set of paths is *maximal* if no path can be added to the set without introducing anomalies.

A simple strategy for computing a maximal set of paths is the following. Starting from the empty set, add all the paths one-by-one, each time testing if the set admits an orientation without anomalies. The test can be performed in linear time by exploiting the algorithm presented in Section III. If the insertion of a path makes the set not orientable, then it is discarded, otherwise it is added to the set. At the end of the process we have a maximal set of paths. However, this simple strategy is unfeasible. In fact we would have to run the testing algorithm millions of times. Even if each run takes one second, we could wait a couple of weeks to have the maximal set.

Motivated by the above discussion, we propose a two steps approach. First, we compute a very large (albeit not maximal) set of valid paths with an ad-hoc technique. Second, we check if the discarded paths can be reinserted with the method described above.

The computation of the initial very large set of valid paths is performed as follows. Initialize P with the set of all the paths.

1) Construct the G_{2SAT} graph considering all the adjacencies of P.
2) Set-up the following data structure: for each undirected edge (v_i, v_j) of the AS adjacency graph keep the number of paths traversing (v_i, v_j); call it *covering* of (v_i, v_j).
3) Compute the strongly connected components of G_{2SAT} (e.g., with the algorithm in [13]).
4) Identify each variable x such that x and $\overline{x}$ are in the same strongly connected component of G_{2SAT}.

5) Select among those variables the variable $x_{i,j}$ whose corresponding edge (v_i, v_j) has the smallest covering and remove all the paths that cover such an edge from P.

Execute steps (1) through (5) until no strongly connected component contains both the literals of the same variable.

Observe that at each iteration, since we remove all the paths traversing a specific edge of the AS graph, the literals associated with such an edge disappear from G_{2SAT}.

In our data set the starting G_{2SAT} graph contains $47,522$ nodes and $375,100$ edges. It contains one strongly connected component with $2,156$ literals and $12,570$ edges. The other components contain just one literal. The set of valid paths computed during the first step contains $3,423,460$ paths. During the second step $222,764$ paths have been re-inserted without causing anomalies. The final maximal set of paths contains $3,399,389$ paths.

After computing a maximal set of paths we have computed an orientation for the edges of the AS graph obtained from those paths, using the technique illustrated in Section IV. A fragment of the computed orientation has been used already in this paper for the example of Figures 4 and 5. Further, following the experimental guideline of [2] we have done two types of checks of the quality of such orientation:

1) We checked how many paths of the original ten data sets are valid and the percentage of invalid paths.
2) We checked how good is the computed orientation against four additional data sets that were not input of the heuristic algorithm. Such extra group of data sets is, again, available from [15] and contains data from AS1755, AS2516, AS2548, and AS6893.

TABLE III

COMPARISON BETWEEN THE HEURISTICS PRESENTED IN THIS PAPER AND THE STATE OF THE ART.

AS #	AS Name	Anomalies % ([2])	Anomalies % (this paper)
1	Genuity	0.65	0.45
1740	CERFnet	n.a.	0.36
3549	Globalcrossing	n.a.	0.13
3582	U. of Oregon	n.a.	0.57
3967	Exodus Comm.	n.a.	0.42
4197	Global Online J.	n.a.	0.46
5388	Energis Squared	n.a.	0.46
7018	AT&T	0.63	0.21
8220	COLT Internet	n.a.	0.22
8709	Exodus, Europe	n.a.	0.21
1755	Ebone	2.89	1.52
2516	KDDI	8.97	4.95
2548	MaeWest	1.49	0.19
6893	CW	2.92	0.64

Table III shows that the heuristics described above leaves a very small percentage of invalid paths. In particular, it performs significantly better, in terms of invalid paths, than the cutting edge heuristics of [2]. The invalid paths are about halved for ASes 1, 1755, and 2516, are about one third for AS 7018, and are one fourth or less for ASes 2548 and 6893. These results are, in our opinion, even stronger if we consider that the percentages of anomalies provided by [2] do not count as invalid Type 2 paths containing two consecutive undirected edges instead of one [19]. The basis of such relaxation of the model is that two ASes may have an "indirect peering", that is a peer-to-peer relationship through an intermediate one.

Using the condition discussed in Section II-B we have also found edges that can be made undirected still preserving the quality of the solution and have found $3,936$ edges that can be considered as candidates for being peering edges.

It is worth noting that we have conducted all the experiments with the same platform described in Section III. The above experiment, involving $3,423,460$ paths, required a computation time of about 10 hours.

VII. CONCLUSIONS AND OPEN PROBLEMS

In this paper we introduced a novel approach for computing the relationships between Autonomous Systems starting from a set of AS paths, so that the number of invalid paths is kept small. Also, we proved that the corresponding minimization problem is NP-complete in the general case (as conjectured in [2]).

Our approach consists of mapping the problem into a 2SAT formulation, which can be exploited in several ways. For example, a solution for the 2SAT formulation can be found in linear time, if it exists, determining a solution to the original problem without invalid paths. Also, we take advantage of the theoretical insight gained with the 2SAT formulation to conceive new heuristics for the general case which we experimentally prove to be more effective than previously presented approaches.

Further details on the experiments, the used source code, and the data sets are available from the Web-site `http://www.dia.uniroma3.it/~compunet` and in [20].

The classification of AS relationships in the Internet is a hot research topic. Its relevance is confirmed by the interest of other research groups on the same subject. In [21] analogous and independently discovered results concerning the time complexity of the general problem and the linearity in the case of all valid paths are shown. However, while such work puts more emphasys on the approximability of the problem, we focus more on the engineering and the experimentation of an effective heuristic approach.

Several problems remain open. We think it is interesting to understand why the portion of the Internet seen from a single observation point is very often orientable without invalid paths. Is that a matter of size or there is a more subtle property that can be formally studied? Also, the recognition of AS relationships can probably take advantage of further information provided by the BGP routing tables, for example, the size of the prefixes. Can this lead to a prefix-driven formulation of the problem instead of the as-path driven formulation adopted until now? Further, it could be interesting to improve existing tools for the visualization of the AS graph (see, e.g., [22]) in order to provide information about the relationships between ASes.

ACKNOWLEDGMENTS

We are grateful to the authors of [2] for their help. Also, we would like to thank Andrea Vitaletti and Debora Donato for interesting conversations.

REFERENCES

[1] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 733–745, Dec 2001.

[2] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, "Characterizing the internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM 2002*, 2002.

[3] J. W. Stewart, *BGP4: Inter-Domain Routing in the Internet*. Reading, MA: Addison-Wesley, 1999.

[4] C. Alaettinoglu, "Scalable router configuration for the internet," in *Proc. IEEE IC3N*, October 1996.

[5] G. Huston, "Interconnection, peering, and settlements," in *Proc. INET*, June 1999.

[6] R. Govindan and A. Reddy, "An analysis of internet inter-domain topology and route stability," in *Proc. IEEE INFOCOM 1997*, April 1997.

[7] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *Proc. IEEE INFOCOM 2000*, March 2000.

[8] W. Theilmann and K. Rothermel, "Dynamic distance maps of the internet," in *IEEE INFOCOM 2000*. Tel-Aviv, Israel: IEEE, March 2000. [Online]. Available: citeseer.nj.nec.com/theilmann00dynamic.html

[9] Z. Ge, D. R. Figueiredo, S. Jaiswal, and L. Gao, "On the hierarchical structure of the logical internet graph," in *Proc. SPIE ITCom 2001*, 2001.

[10] L. Gao, T. G. Griffin, and J. Rexford, "Inherently safe backup routing with BGP," in *IEEE INFOCOM 2001*, Apr 2001, pp. 547–556.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman, 1979.

[12] B. Aspvall, M. F. Plass, and R. E. Tarjan, "A linear-time algorithm for testing the truth of certain quantified boolean formulas," *Information Processing Letters*, vol. 8, no. 3, pp. 121–123, 1979.

[13] K. Mehlhorn, *Data Structures and Algorithms*. Springer Publishing Company, 1984, vol. 1-3.

[14] K. Mehlhorn and S. Näher, *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[15] "Characterizing the internet hierarchy from multiple vantage points," `http://www.cs.berkeley.edu/~sagarwal/research/BGP-hierarchy/`.

[16] "University of Oregon RouteViews project," `http://www.routeviews.org`.

[17] U. Feige and M. Goemans, "Approximating the value of two prover proof systems, with applications to max 2sat and max dicut," in *Proc. of 3rd Israel Symposium on the Theory of Computing and Systems*, 1995, pp. 182–189.

[18] V. Kann, "On the approximability of NP-complete optimization problems," Ph.D. dissertation, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.

[19] L. Subramanian, personal communication.

[20] G. Di Battista, M. Patrignani, and M. Pizzonia, "Computing the types of the relationships between autonomous systems," Dipartimento di Informatica e Automazione, Università di Roma Tre, Technical Report RT-DIA-73-2002, July 2002.

[21] T. Erlebach, A. Hall, and T. Schank, "Classifying customer-provider relationships in the internet," ETH Zurich, Technical Report TIK-145, July 2002.

[22] A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia, "Visualization of the high level structure of the internet with hermes," *J. of Graph Algorithms and Applications*, vol. 6, no. 3, pp. 281–311, 2002.

Archives of BGP Updates: Integration and Visualization*

Giuseppe Di Battista, Federico Mariani,
Maurizio Patrignani, and Maurizio Pizzonia
Dip. Informatica e Automazione
Università di Roma Tre
Via della Vasca Navale 79, 00146 Roma, Italy
{gdb,mariani,patrigna,pizzonia}@dia.uniroma3.it

Abstract

The possibility of analyzing updates exchanged between BGP talkers is crucial for several operational and research purposes. To give a few examples, they can be used to investigate the stability of specific routes, to monitor the effects of faults, and to analyze the behavior of the entire network in the presence of particular events. Several archives of BGP conversations, such as the Routing Information System of the RIPE and the Oregon Route Views database, give an answer to this need for information. We describe a work that aims at integrating the data from different BGP-update sources and at presenting such data with graph-based visualization techniques. We address both technological issues related to the data integration and user-interaction issues originated from the necessity of visualizing data that change over time.

1 Introduction

The exploration and visualization of the Internet attracts an increasing research interest, motivated by the growing size of the network and the significant impact that connectivity has on social and economic activities.

Contributions in this field can be roughly classified with respect to the granularity of the data they consider.

For example, Hermes [6, 1] visualizes the information provided by the Internet Registries about the interconnections between Autonomous Systems (ASes), showing their peerings and mutual policies.

*Work partially supported by European Commission - Fet Open project COSIN - COevolution and Self-organisation In dynamical Networks - IST-2001-33555; by "Progetto ALINWEB: Algoritmica per Internet e per il Web", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale; and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR Fondo per gli Investimenti della Ricerca di Base.

At a more granular level, Mercator [9], Skitter [11], and Rocketfuel [15] compute maps from data collected by means of network discovery, showing interconnections among routers. Another system using traceroutes for probing the network is described in [5].

In this paper we present BGPlay, a system that shows the routing at the interdomain level acquired from communications between BPG-speaking peers [16]. BGPlay integrates the data from different archives of BGP updates and presents such data with graph-based visualization techniques.

Currently, BGPlay integrates the BGP updates collected by the Routing Information Service [3] of the RIPE and those collected by the Oregon Route Views project [4]. BGPlay aims at giving a highly intuitive visual representation of the status of the routing at a specific time, and of its evolution in a given time interval. In order to show such an evolution the system relies on an animation which shows how the BGP-paths evolve over time while preserving the user mental map [14, 7].

BGPlay has been designed to satisfy both operational and research needs. To give a few examples, it can be used to investigate the stability of specific routes, to monitor the effects of faults, and to analyze the behavior of the entire network in the presence of particular events. Instabilities and faults of interdomain routing have been the subject of recent research (see for example [13, 10, 8, 12]).

The paper is organized as follows. In Section 2 the data sources used by BGPlay are described. The overall architecture of the system is discussed in Section 3, while Section 4 and Section 5 illustrate how data are collected and visualized. Finally, Section 6 contains our conclusions and future work.

2 Archives of BGP Data

The Oregon Route Views (ORV) project (see Fig. 1.a) provides a service for Internet operators to obtain real-time

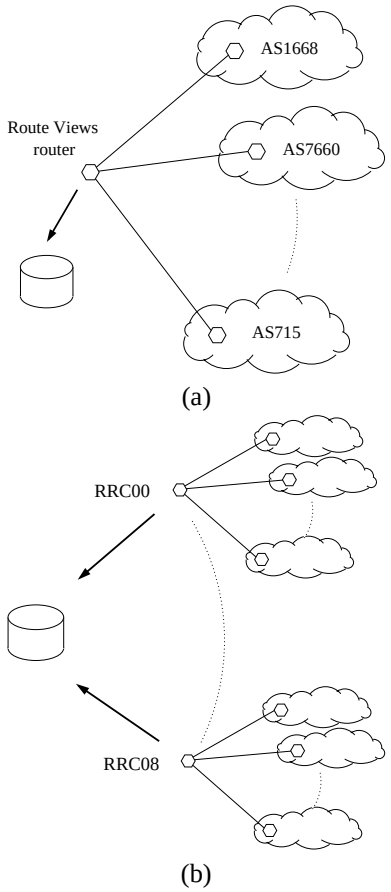


Figure 1. The architecture of ORV (a) and of RIS (b).

information about the global routing system from the perspectives of several different locations around the Internet. Currently, the BGP Route Views router has more than 60 multi-hop eBGP peering sessions with routers of Internet service providers.

ORV collects, without providing any transit service, the BGP updates coming from its peers. Such updates allow to reconstruct the evolution of the best routes (at the AS level) adopted by each peer. ORV data archives contain both snapshots of the global routing information base of ORV at different instants of time and sequences of BGP updates. The data are made available in the MRT [2] format.

The Routing Information Service (RIS) of the RIPE (see Fig. 1.b) collects historical information about Internet routing by using Remote Route Collectors (RRC) at different locations around the world. Such information is integrated into a comprehensive view. An RRC of the RIS is a BGP speaking router that only collects BGP routing information. The collected raw data is regularly transferred to a central storage area at the RIPE NCC in Amsterdam. Each RRC is denoted by a string with format RRCxx, where xx is a two digit number. Essentially, each RRC behaves analogously to the ORV router.

The RIS offers to the users several query facilities. For example, the BGP Routing Hot Spot Utility generates lists of prefixes, originating from a specified AS, for which high BGP announcement activity has been observed by some RRC. Also, the ASInuse facility determines when an AS-number last appeared in the global routing table collected by the RIS.

In this paper, we are interested in a simpler RIS query, that can be performed using the Search by Prefix utility. It allows to specify a prefix, a time interval, and a set of RRCs in order to search the RIS database. It outputs a list of BGP updates recorded by the selected RRCs in the prescribed time interval. It also outputs the status of the Local Routing Information Base (Loc-Rib) for the selected prefix in an instant of time that is "related" to the prescribed time interval.

3 System Architecture

The architecture of BGPlay is based on the following main choices.

- The user interacts with the system by means of a browser. A query identifies a time interval and a prefix to be monitored. The data sources to be used can be selected in a set of possible alternatives.
- The results of the query, i.e. the changes in the BGP routing observed during the time interval, are visualized by an "animation." Such animation relies on Graph Drawing techniques [7]. When the animation is started, a graph-like representation shows the routing at the beginning of the interval. During the animation the graph changes according to the observed BGP updates.
- The BGP data exploited to answer the user queries are partly fetched on-line at the moment they are needed and are partly locally stored. Namely, since the RIS provides a Web query facility we access those data on-line. Also, since the Route Views Project does not have a Web query facility for historical data, we periodically copy part of them locally. Currently, because of limitations on the available storage space, we maintain a copy of a limited (the most recent) period of time.
- The service is based on a client-server architecture, where the server computes the result of queries and the client is an applet running on the user's browser. We decided of using an applet instead of producing on the server standard jpeg or gif figures because of the graphical complexity of the animation.

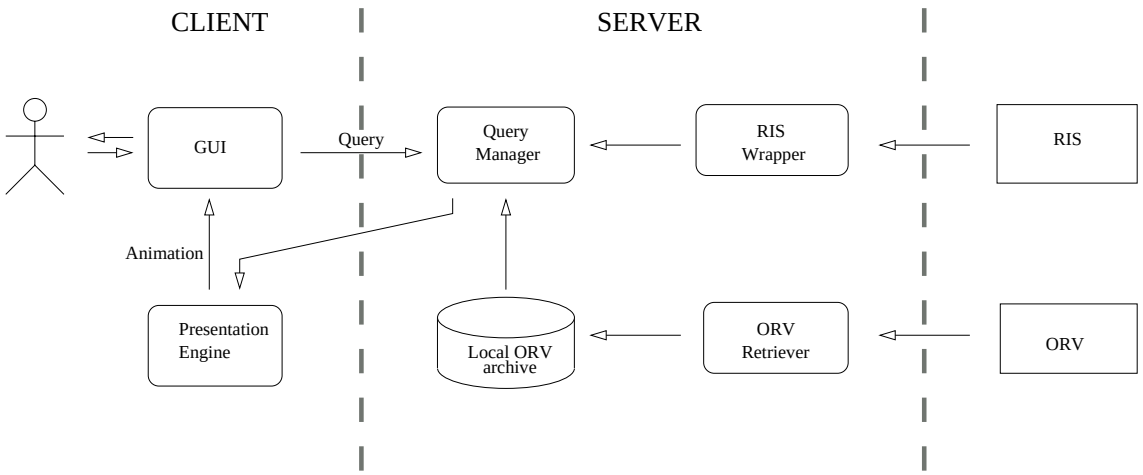


Figure 2. The system architecture.

Figure 2 illustrates the main components of the architecture:

Graphic User Interface. Provides the user with several tools for interacting with the visual representation of the routing information. For example: a *time panel* shows the time location of the most important events in the selected time interval, *animation buttons* allow to step over the events, and an *event display* gives all the available details about the currently visualized event.

Presentation Engine. Computes the layouts of the graphs representing the evolution of the routing. Further, is able to change such graph layout according to the routing events, in such a way that the user always perceives "smooth" variations. It exploits Graph Drawing methodologies and techniques.

RIS Wrapper. Queries the RIS Search by Prefix utility and retrieves the corresponding results in terms of BGP routing tables and updates.

ORV Retriever. Periodically retrieves routing tables and updates available at the ORV raw data archives in the MRT format. Updates the Local ORV Archive removing the oldest data.

Local ORV Archive. A relational database (currently, MySQL technology) storing the ORV data. Observe that, while all the RIS data refer to the UTC (Coordinated Universal Time), the ORV data refer to its local time. In BGPlay we refer to the UTC.

Query Manager. Gets a query description from the User Interface, drives the RIS Wrapper, accesses the Local ORV Archive, computes the result of the query, and delivers the result to the Presentation Engine.

An analysis of the workload and of the generated traffic lead us to a decomposition of the system in which the Presentation Engine is located on the client side, together with the Graphical User Interface.

4 Query Processing

In order to answer a query of the client, the Query Manager has to retrieve the BGP updates falling in the specified time interval. As mentioned in Section 3 two types of retrieval are involved: retrieval from the local database of ORV updates and on-line retrieval, performed by the RIS Wrapper, from the RIS Service.

The scenario is more precisely depicted in Fig. 3, where the starting and ending instants of the time interval are called t_1 and t_2, respectively. The purpose of the Query Manager is to compute:

- the status of the ORV RIB in t_1;
- the status of the RIS RIB in t_1; and
- the sequence of updates collected by ORV and RIS between t_1 and t_2.

Unfortunately, it is unlikely to have at disposal the two aforementioned RIBs at time t_1. Also, the possibilities of retrieving such information from the two data archives are quite different.

On one side, ORV offers a snapshot of its RIB every (about) two hours. Hence, we have to determine which is the last available ORV RIB before t_1. Once such RIB has been obtained (suppose it corresponds to a snapshot taken at time $t_0 \leq t_1$), we perform the following operations. First, the rows of the RIB corresponding to the given prefix p are

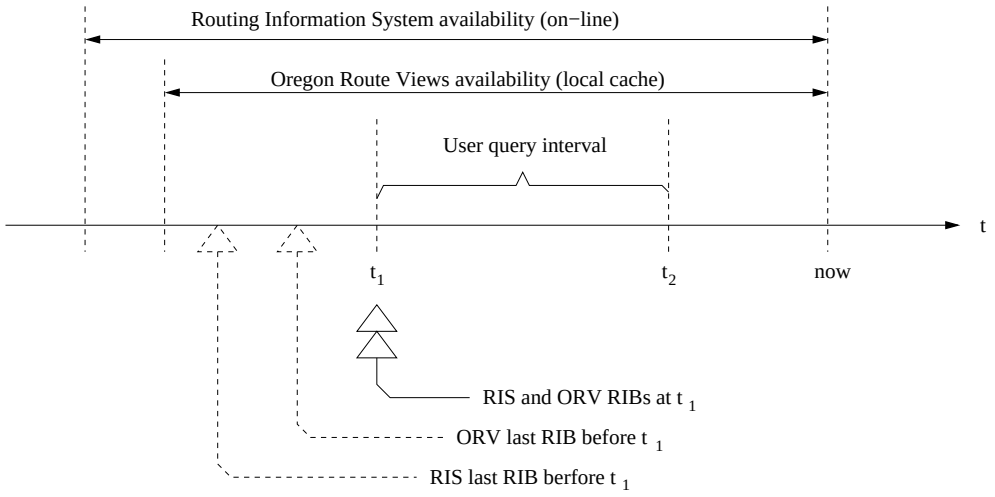


Figure 3. Available and computed RIBs.

extracted. Second, the updates occurring in the time interval t_0, t_1 are taken into account in order to compute the portion of the RIB corresponding to p at time t_1.

On the other side, the situation for RIS is more complex. Namely, as shown in Fig. 4, a query sent to the on-line RIS interface involving the time interval t'_1, t'_2 yields two types of data:

- the updates collected between t'_1 and t'_2 and
- the RIS RIB at time 23:59 of the day in which t'_1 falls.

Hence, it is not possible, for the Query Manager, to simply activate the RIS Wrapper with a query interval such that $t'_1 = t_1$ and $t'_2 = t_2$.

Because of the above discussion, the Query Manager asks the RIS Wrapper to perform a query with time interval t'_1, t'_2, where $t'_2 = t_2$ and t'_1 is the time 23:59 of the day preceding the one in which t_1 falls. Afterwords, analogously to what happens for ORV, the RIB is filtered saving only the portion concerning p. Then, the updates occurring in the time interval t'_1, t_1 are taken into account in order to compute the portion of the RIB corresponding to p at time t_1.

5 Updates Visualization

One of the purposes of BGPlay is to visualize BGP updates. However, the semantics of an update is essentially in the changes it induces in the routing. Hence, our approach relies on two types of visualization techniques. First, we exploit methods to visualize the status of the routing at a given instant of time. Second, we exploit techniques that allow the user to perceive how an update brings the routing from an initial status to a consequent one.

5.1 The Routing Graph

If we focus the attention on a prefix, the status of the routing at a given time for that prefix consists of a set of AS-paths, each representing the best route at that time to reach the prefix from a certain AS. Such a status is effectively represented with a *routing graph*. A routing graph is a decorated graph in which each vertex is an AS and edges are the pairs of ASes that appear consecutively in at least one of the paths. Each edge is labeled with the set of paths traversing it.

The AS-paths representing the status of the routing are provided to the Presentation Engine by the Query Manager. In general, part of them come from ORV and represent the routes selected by the ORV peerers to reach the prefix. Part of them come from RIS and represent the routes selected by the RRC peerers to reach the prefix. The overall information conveyed by the representation is the routing of the Internet traffic flowing toward the specified prefix at the AS level.

Visualizing a routing graph is not an easy task. In fact, even if large portions of it look like a tree, it may contain cycles and dense subgraphs. Further, some vertices may have many incident edges and a single edge may be traversed by several paths.

In designing the Presentation Engine we have identified the following requirements:

- The attention of the user should mainly be focused on the AS originating the prefix (target AS).
- An AS should appear in the drawing at a geometric distance from the target AS that is roughly proportional to the number of (AS-)hops separating them.
- Each single AS-path must be fully identified, even if traversing edges that are traversed by other paths. Ob-

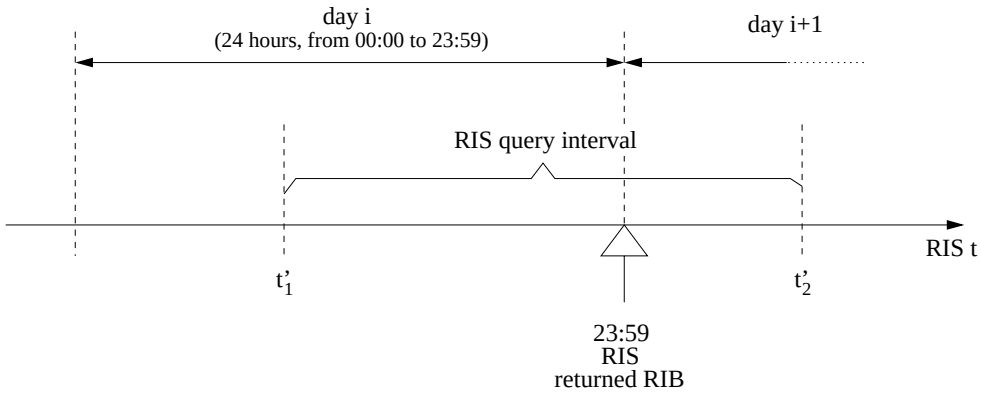


Figure 4. RIBs returned by the RIS.

serve that this requirement would be easy to meet if the graph was a tree; in fact in this case there is just one tree path from each AS to the target AS. The presence of cycles makes the problem more complex.

In order to compute drawings of the routing graphs satisfying the above requirements we used a "spring embedder." It consider the graph as a system of bodies (vertices) and forces acting between the bodies. Such forces can either attract or repel the bodies. The system is left free to oscillate until an equilibrium is reached. We made the choice to use the following types of forces:

- A repelling force is set between each pair of ASes.
- An attractive force is set between each pair of ASes connected by an edge.
- To damp the oscillations, the effect of the forces is decreased with the time. Essentially, this is obtained by progressively augmenting the "viscosity" in the system.

We also fixed the position of the target AS at the center of the area. Paths are represented with different colors and edges traversed by several paths are displayed using as many lines as the number of paths traversing that edge, where each line is colored with the color of the corresponding path.

Fig. 5 shows a drawing produced by BGPlay. It is about a prefix announced by the GARR AS (AS137). It clearly shows that the route collectors are reached by the announcements of AS137 through three main directions. Namely, part of the paths flow through AS3549 (Globalcrossing), other paths through AS1299 (TeliaNet), while others through AS20965 (GEANT). Fig. 6 shows the routing for AS7018.

5.2 Sequences of Updates

Obviously, the visualization of the routing graph before and after a BGP update occurred is sufficient to convey the information of the update. However, it is essential for the two consecutive visualizations to be "similar", while the routing change should be apparent to the user.

Consider a BGP announcement and the corresponding AS-path p'. Let AS_T be the target AS and suppose that the ending ASes of p' are AS_T and AS_S. Two cases are possible: either another AS-path p with the same pair of ending ASes was already in the routing graph or not. In the first case the user should perceive that the traffic flow from AS_S to AS_T is changing its route. In the second case it is important to make clear that a new source of traffic is becoming visible from AS_T.

A route change from p to p' involves several possible changes in the routing graph. Let us compare $p = (AS_T = AS_1, AS_2, \ldots, AS_m = AS_S)$ and $p' = (AS_T = AS'_1, AS'_2, \ldots, AS'_n = AS_S)$. We can split p and p' into sub-paths as follows. Let AS_i be the first AS of p that is equal to some AS (say AS'_j) of p'. We split p and p' into the sub-paths $(AS_1, \ldots, AS_i)$, $(AS_i, \ldots, AS_m)$ and $(AS'_1, \ldots, AS'_j)$, $(AS'_j, \ldots, AS'_n)$, respectively. We can repeat the above split process on the two sub-paths $(AS_i, \ldots, AS_m)$ and $(AS'_j, \ldots, AS'_n)$, until they are no longer decomposable. Such a process yields a decomposition of the two original paths into an equal number of sub-paths pairwise starting and ending on the same vertices.

The Presentation Engine performs a graphic "morphing" where each sub-path in which p is decomposed is mapped to the corresponding sub-path of p'. Three cases are possible:

1. the two sub-paths have equal length
2. the sub-path of p is longer than the sub-path of p', and
3. the sub-path of p is shorter than the sub-path of p'.

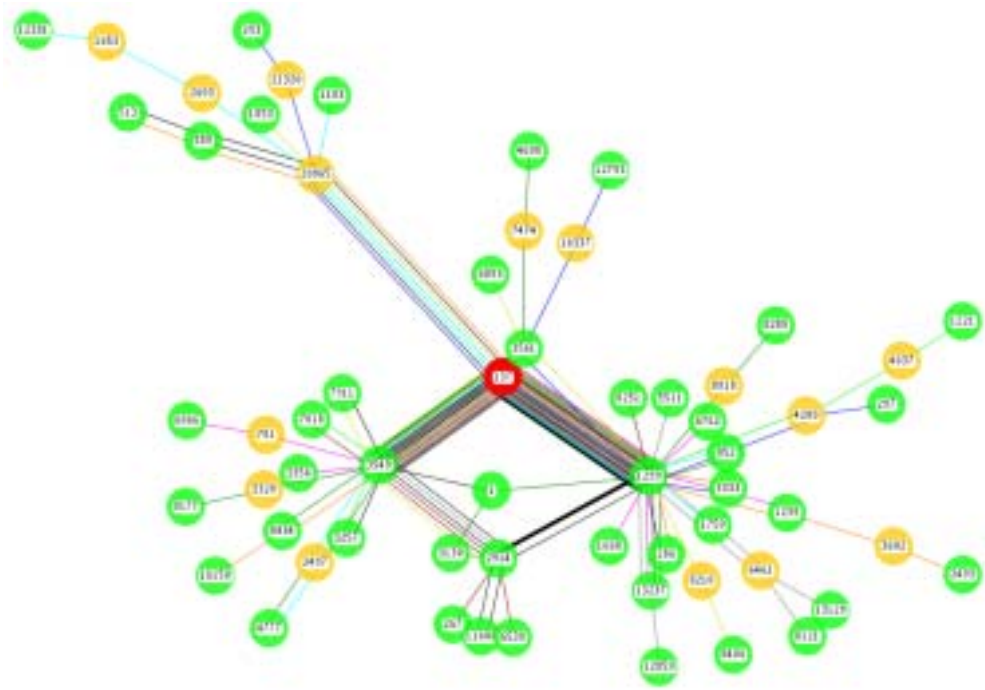


Figure 5. A drawing produced by BGPlay rapresenting the routing for prefix 193.204.0.0/15 (AS137).

In all the above cases we introduce in the routing graph a chain of additional "dummy" vertices and move them from the original sub-path to the position of the new sub-path. However, in case 2 some dummy vertices are "absorbed" into the same vertex, while in case 3 some dummy vertices are "created" from the same vertex. Of course, some ASes of the original graph may disappear since they are no longer traversed by any path and some ASes are created because they are in p' and were not in the graph.

Consider now a BGP withdrawal. It is managed by the Presentation Engine as follows. First, the involved path is highlighted, to attract the attention of the user, then the edges traversed by the path change their labels an, in case, removed.

6 Conclusions and Future Work

We have presented BGPlay, a system designed to visualize the evolution of the routing involving a given prefix at the Autonomous Systems level (BGP level). BGPlay is able to show the BGP events occurred in a given time interval through an animation illustrating the corresponding changes in the routing graph. BGPlay is available at `http://www.dia.uniroma3.it/~compunet`.

Future work will mainly focus on the following issues.

- We plan to test the effectiveness of new visualization methods, alternative to those based on the display of a routing graph.
- We would like to provide the user with more large-scale visualization facilities, that allow to track the evolution of the routing paths of several prefixes (not necessarily originated by the same AS) at the same time.
- We are interested in integrating new data sources, both publicly available and provided by private organizations.

Acknowledgements

We would like to thank Fabrizio Lombardozzi for implementing a preliminary version of the system, called Flapviewer.

References

[1] Hermes. `http://www.dia.uniroma3.it/~hermes/`.
[2] Multithreaded Routing Toolkit (MRT) Project [Online]. `http://www.mrtd.net`.
[3] Routing information service of the ripe. `http://www.ripe.net/ripencc/pub-services/np/ris/`.
[4] University of Oregon RouteViews project. `http://www.routeviews.org`.

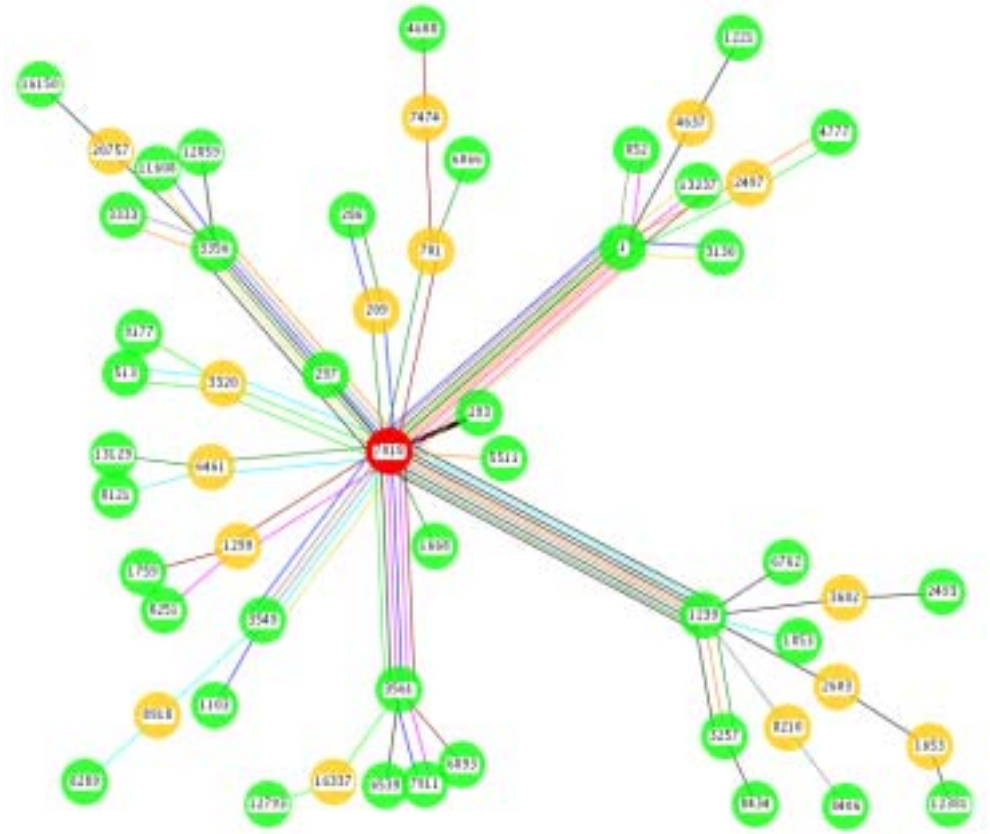


Figure 6. Flows of traffic for AS7018 represented by BGPlay.

[5] S. Branigan, H. Burch, B. Cheswick, and F. Wojcik. What can you do with traceroute? *IEEE Internet Computing*, Sept./Oct. 2001. `http://computer.org/internet/v5n5/index.htm`.

[6] A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the high level structure of the internet with hermes. *J. of Graph Algorithms and Applications*, 6(3):281–311, 2002.

[7] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[8] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Measurement and Modeling of Computer Systems*, pages 307–317, 2000.

[9] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *IEEE INFOCOM 2000*, pages 1371–1380, Tel Aviv, Israel, March 2000.

[10] T. Griffin and G. T. Wilfong. An analysis of BGP convergence properties. In *SIGCOMM*, pages 277–288, 1999.

[11] B. Huffaker, D. Plummer, D. Moore, and k claffy. Topology discovery by active probing. Technical report, Cooperative Association for Internet Data Analysis - CAIDA, San Diego Supercomputer Center, University of California, San Diego, 2002.

[12] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *SIGCOMM*, pages 175–187, 2000.

[13] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, 1998.

[14] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Lang. Comput.*, 6(2):183–210, 1995.

[15] N. Spring, R. Mahajan, and D. Wetherall. Measuring isp topologies with rocketfuel. In *Proceedings of ACM/SIGCOMM '02*, Aug. 2002.

[16] J. W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, Reading, MA, 1999.

BGPlay: a System for Visualizing the Interdomain Routing Evolution* (Long Demo)

Giuseppe Di Battista, Federico Mariani, Maurizio Patrignani, and Maurizio Pizzonia

Università degli studi "Roma Tre"
Dipartimento di Informatica e Automazione
Via della Vasca Navale 79
00146 Roma, Italy

Abstract. In this paper we describe the visual interface of BGPlay, an on-line service for the visualization of the behavior and of the instabilities of the Internet routing at the autonomous system level.
A graph showing only connections among autonomous systems is not enough to convey all the information needed to fully understand the routing and its changes. BGPlay provides specifically tailored techniques and algorithms to show the routing at specific instants of time and to animate its changes. The system obtains routing data from well known on-line archives of routing information constantly kept up-to-date.

1 Introduction

The Internet is administratively partitioned into networks, called *Autonomous Systems* (*AS*), where each AS is under a single administrative authority. Usually, each Internet Service Provider (ISP) controls one or more ASes.

Roughly speaking, at the AS level the Internet routing is described by a collection of sequences of ASes, called *AS-paths*. An AS-path $AS1, AS2 \ldots, ASn$ says that the packets directed to ASn and originated by some device in $AS1$ should traverse $AS2$, $AS3$, etc. in the specified order. The AS-paths that give the full routing status of the Internet at a certain moment can be "merged" into a graph, called a *routing graph*.

To understand our goals, consider the Network Operating Center (NOC) of some ISP and suppose that the NOC wants to know what are the paths that the packets currently follow to reach ASx which is operated by the NOC. BGPlay aims at showing the portion of the routing graph that describes how the traffic

* Work partially supported by European Commission - Fet Open project COSIN – COevolution and Self-organization In dynamical Networks – IST-2001-33555, by "Progetto ALINWEB: Algoritmica per Internet e per il Web", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR Fondo per gli Investimenti della Ricerca di Base.

flows to ASx from a selected set of ASes which are considered, in some sense, "representative" of the entire Internet.

Further, suppose that the NOC is interested in understanding how the routing evolved during a specific time interval. This can be important for several reasons: to determine faults in the pipes surrounding ASx, to check the consistency of the routers configurations, or even to monitor the behavior of the partners of the considered ISP with the purpose to verify whether they fulfill the commercial agreements. BGPlay aims at showing all the routing changes "around" ASx that occurred during the prescribed time interval.

Instabilities and faults of interdomain routing have been the subject of recent research in the networking area (see for example [17, 12, 10, 16]). Also, several works address the problem of effectively visualizing network related graphs [11, 13, 20, 5, 1]

In this paper we describe the visual interface of BGPlay [7], which runs at `http://www.dia.uniroma3.it/~compunet/bgplay`. The purpose of BGPlay is the visualization of the behavior and of the instabilities of the Internet routing at the AS level.

BGPlay has a client-server architecture. The user performs a query on a Web browser (the client side) specifying an IP prefix and a time interval. The server identifies the AS that contains the given prefix and extracts information that describes the routing evolution in the given interval from the Routing Information Service (RIS) [2] of the RIPE and from a local mirror of the Oregon Route View (ORV) project [3]. RIS and ORV are large and well known repositories of routing information.

All the visualization problems are addressed on the client side by means of a java applet. The user interface exposed by the BGPlay client can visualize the routing graph detected at any instant of the specified time interval. The BGPlay visualization puts in evidence ASes interconnections and which AS-paths are active on that interconnections in a given instant of time.

The paper is organized as follows. In Section 2 we give basic interdomain routing concepts. In Section 3 we introduce the concept of *routing graph*, discuss the requirements for its visualization, and argue about the unfeasibility of some possible visualization approaches. In Section 4 we show the techniques adopted in BGPlay for the visualization of the routing graph. In Section 5 we show how BGPlay animates the evolution of the routing. Section 6 shows a typical session with BGPlay.

2 Networking Background

In the Internet each host is identified by an IP address (32 bits, usually written in the dotted notation, e.g. 193.204.161.48). An IP prefix identifies a set of (contiguous) IP addresses having the same leftmost n bits, with $0 \leq n \leq 32$. Such IP prefix is usually indicated attaching a $/n$ at the end of the prefix (e.g. 193.204.0.0/15 indicates a prefix 15 bits long) [22]. Routing in the Internet is prefix based (like for telephone call routing). Since a prefix identifies a set of

addresses, it implicitly identifies a set of hosts having such addresses. In the following the term *prefix* is used to denote both a set of addresses and a set of hosts, the distinction will be clear by the context.

An *Autonomous System* (*AS*) is a portion of the Internet under a unique administrative authority. In the Internet each AS is identified by an integer number. ASes cooperate in order to ensure good connectivity service to their customers but are competitors from an commercial point of view [14, 15].

Traffic starting from an AS and directed to a specific prefix traverses an ordered set of ASes (*AS-path*). The configuration of such paths on the routing devices is too complicated to be manually performed. Hence, ASes exchange routing information with other ASes by means of a routing protocol called Border Gateway Protocol (*BGP*) [19, 21]. Such a protocol is based on a distributed architecture where *border routers* that belong to distinct ASes exchange the information they know about reachability of prefixes. Two border routers that directly exchange information are said to perform a *peering session*, and the ASes they belong to are said to be *adjacent*.

Each router stores information about routing into its *routing information base* (*RIB*). The RIB is a table where each line is a pair $\langle$prefix, AS-path$\rangle$ meaning that a certain prefix is reachable through the associated AS-path. Such pairs are called *routes*. The main purpose of BGP is to allow the routers to exchange the routes they know. Since RIBs may be huge, the BGP process running on a router sends to its peers the full RIB only when a peering session is set up. During regular operation only updates are sent.

A BGP *update* is either a route *announcement* or a route *withdrawal*. An announcement conveys the following information: "through me you can reach a certain prefix traversing a certain AS-path". A withdrawal nullifies a previously communicated route related to a specified prefix. In other words a withdrawal means "you can no longer reach this prefix through me".

The receiver of an update may or may not modify its routing table depending on whether the router knows routes which are considered better or not. If the router modifies the routing table, the update is propagated to its peers.

Routes related to a certain prefix "born" within an AS called the *originator* of the prefix. Then, routes are propagated to adjacent ASes, which prepend their AS identifiers to the AS-path of the route and propagate it again by means of route announcements.

3 Effective Routing Graph Visualization

The purpose of BGPlay is to provide a graphical representation of a portion of the Internet routing at a given instant of time and of the changes that affect routing over time. In this section we describe the requirements we considered and show some negative preliminary results.

We define a *collector-peer* as an AS that has a BGP peering with ORV or with RIS collectors. The collector-peers are the vantage points we use to inspect the routing in the Internet and are determined by the current configuration of

RIS and ORV. The set of collector-peers we are currently using is easy to identify by just looking at the ASes appearing as leftmost ASes in at least one AS-path among the AS-paths provided by ORV and RIS.

We focus the attention on a specific prefix that is in turn contained in one AS, the *target AS*. The *routing status* at a given time for that prefix gives for each collector-peer the AS-path representing the route chosen at that time by that collector-peer to reach the prefix. Such a status is effectively represented with a *routing graph*. A routing graph is a graph in which each vertex is an AS and edges are the pairs of ASes that appear consecutively in at least one of the AS-paths.

We have identified the following requirements for the visualization of the routing status:

- The attention of the user should mainly be focused on the target AS.
- An AS should appear in the drawing at a geometric distance from the target AS that is roughly proportional to the number of (AS-)hops separating them.
- For each visualized AS, the AS-path used to reach the target must be fully identifiable in the drawing, even if traversing edges that are traversed by other paths. Observe that this requirement would be easy to meet if the graph was a tree; in fact in this case there is just one tree path from each AS to the target AS. The presence of cycles makes the problem more complex.

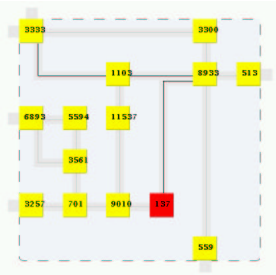


Fig. 1. In a first prototype of BGPlay the drawing standard used was orthogonal. The target AS (the dark node) was arbitrarily placed. The topological distance between AS137 and AS559 is two, while they are located very close in the drawing.

The choice of the drawing standard is biased by the first two requirements. Preliminary experiments we performed in this field have shown how the orthogonal drawing standard is not suitable for visualizing the routing graph. In fact, automatic layout tools in order to effectively compact the drawing may put the target AS in an arbitrary position and geometric distance among nodes does not convey any information about the topological distance (see Figure 1).

In the current version of BGPlay we adopt the straight-line drawing standard. For such standard there exist several layout algorithms that can fulfill the cited requirements (see Section 4).

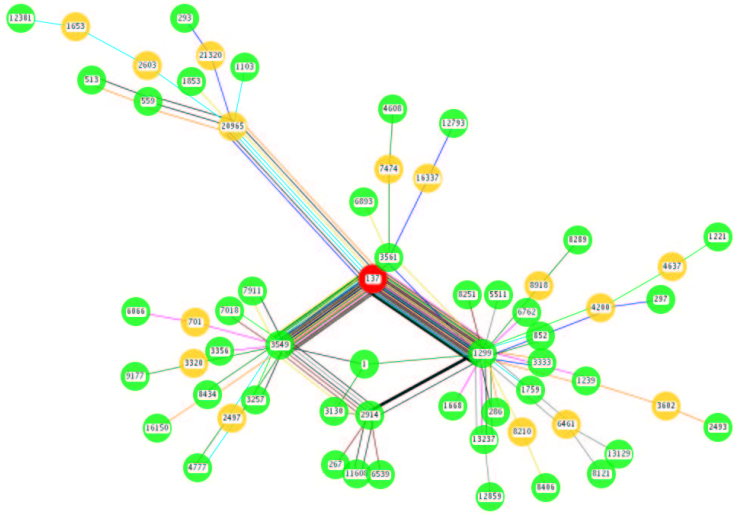


Fig. 2. In a first prototype of BGPlay all AS-paths were separately displayed. Paths were hardly distinguishable.

The third requirement is much more complex. Our first approach was to represent each AS-path separately and with a distinct color. A first prototype using this approach demonstrated that the number of paths to be shown was, usually, too high for the paths to be easily recognized due to the closeness of the paths and to the limited number of colors distinguishable on a monitor by human eyes (see Figure 2). The problem remains hard even using the techniques presented in [8] for drawing with fat edges.

Hence, we decided to adopt a different approach, described in Section 4.

4 Visualization Algorithms

In this section we describe the main algorithmic problems that we had to solve in order to design and implement BGPlay.

The third requirement stated in Section 3, that is that each AS-path must be easy to identify, is the one that was more difficult to meet. As we noted, if a set of AS-paths forms a tree, then such a set can be actually represented as a tree in an arbitrary color, satisfying the requirement. In fact, since each pair of vertices of a tree are joined by a single path, there would be no doubt about the sequence of ASes from the target AS to each collector-peer.

However, the routing graph is very often not a tree. Hence, our strategy is to partition the AS-paths into sets such that the graph obtained by merging the AS-paths of the same set is acyclic and can be unambiguously represented as a tree with a specific color. More formally, we have to solve the following problem.

Problem: Tree Partition

Instance: A set of AS-paths $\mathcal{P}$, such that each AS-path $p \in \mathcal{P}$ starts from a common AS_x.
Target: Find a partition for $\mathcal{P}$ in k sets such that (i) the graph induced by the AS-paths in the same set is acyclic (ii) every partition for $\mathcal{P}$ respecting (i) has at least k colors (that is, the number of sets is minimum).

Unfortunately, the TREE PARTITION problem is NP-hard. In order to prove this, following a standard technique, we show the NP-hardness of the corresponding decision problem, defined as follows.

Problem: K-TREE PARTITION
Instance: A positive integer K and a set of AS-paths $\mathcal{P}$, such that each AS-path $p \in \mathcal{P}$ starts from a common AS_x.
Question: Does a partition for $\mathcal{P}$ in K sets exist such that the graph induced by the AS-paths in the same set is acyclic?

We prove that K-TREE PARTITION is NP-hard by reducing the GRAPH K-COLORABILITY to it. Recall that GRAPH K-COLORABILITY is an NP-complete problem defined as follows.

Problem: GRAPH K-COLORABILITY
Instance: A graph $G = (V, E)$ and a positive integer $K < |V| + 1$.
Question: Is G K-colorable, i.e., does a coloring of the vertices of G in K colors exist such that adjacent vertices have different colors?

Theorem 1 *The problem* K-TREE PARTITION *is NP-hard.*

Proof. We reduce GRAPH K-COLORABILITY to K-TREE PARTITION. Given an instance of GRAPH K-COLORABILITY we construct the corresponding instance of K-TREE PARTITION as follows. For each vertex v of $G(V, E)$ we introduce an AS-path p_v. At the beginnig of the construction all AS-paths have length one, and contain the same AS_x. Then, for each edge $(u, v) \in E$, we append to the two AS-path p_u and p_v the same $AS_{(u,v)}$. It is easy to show that the construction of the K-TREE PARTITION instance can be made in polynomial time, and that a solution for the instance of the GRAPH K-COLORABILITY problem exists iff a solution for the original instance of the K-TREE PARTITION does. □

Because of Theorem 1, in BGPlay, to solve TREE PARTITION, we used the following greedy algorithm which runs in polynomial time.

The sets of AS-paths are denoted $S_0, S_1, \ldots, S_{m-1}$, where the current number of sets is m.

1. A compatibility matrix is computed in which each AS-path is compared with each other. Two AS-paths are incompatible if and only if they form a cycle.
2. Only one empty set exists at the beginning: $m = 1$ and $S_0 = \phi$
3. foreach AS-path p
   3.1 foreach set i from 0 to $m - 1$

- if p is compatible with all paths in S_i then accommodate p into S_i, skip the rest of this cycle and continue with the next path, otherwise try the next set S_{i+1} if it exists.

3.2 if p has not been accommodated in any of the available sets add a new set S_m, initialize $S_m = \{p\}$ and increment m.

For the correctness of the algorithm it is crucial to observe that all the paths have a common endpoint (target AS). The worst case time complexity of the algorithm described above is $O(n^2)$ where n is the number of paths (considered bounded in length). In fact, the number of compatibility tests performed for each AS-path is at most n.

An edge traversed by more than one tree is displayed using as many lines as the number of trees traversing that edge, where each line is colored with the color of the corresponding tree. Fig. 3 shows a drawing produced by BGPlay. It is about a prefix announced by AS702. Three levels of gray are used. AS-paths 16150 8434 3549 702 and 1853 1239 702 are drawn in gray. The other paths are grouped in two trees drawn in black and and light-gray. Only dashed lines are used since, as we shall see in Section 6, BGPlay uses solid lines to highlight paths involved in routing changes.

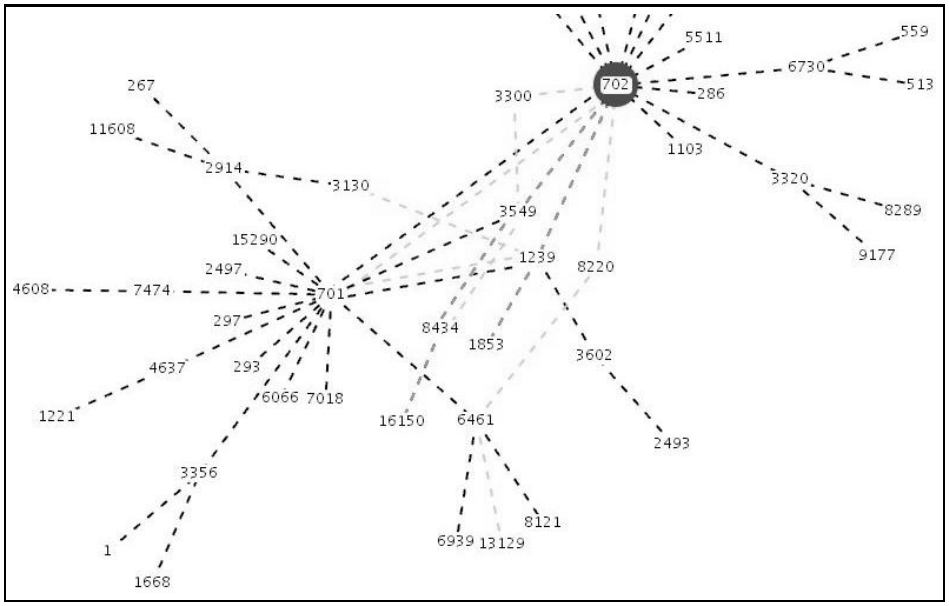


Fig. 3. A drawing produced by BGPlay. Three levels of gray are used. AS-paths 16150 8434 3549 702 and 1853 1239 702 are drawn in mid-gray. The other paths are grouped in two trees drawn respectively in black and light-gray.

The drawing layout is computed by using a spring embedder [6]. The target AS is constrained to be in the center of the drawing area. The computation is

performed for a fixed number of steps that are much larger of the steps needed in the average case by a drawing to reach equilibrium.

What we said up to now concerns the visualization of the routing graph at a given instant of time. However, even for a single user query, BGPlay needs to visualize several routing graphs, each corresponding to the status of the routing at a specific instant.

Obviously, the visualization of the routing status before and after a BGP update occurred is sufficient to convey the information of the update. However, it is essential for two consecutive visualizations to be similar for not changing too much the users "mental map" [9, 18].

We address the above problem by computing, before any visualization or animation takes place, the layout on a graph that is the union of all the AS-paths that appear in any routing graph at any instant of the selected interval. When the animation takes place, at any instant we display only the edges that belong to the current routing graph.

This trick ensures that consecutive animation steps show drawings that are very similar, since node positions are the same. Further, nodes involved in route changes (see Section 5) are most of the times placed near. In fact, two AS-paths involved in a routes change begin and end with the same AS, hence, the spring embedder tends to draw them closely.

5 Animation of the Routing Changes

Obviously, the visualization of the routing status before and after a BGP update occurred is sufficient to convey the information of the update. However, it is essential for two consecutive visualizations to be "similar" [9, 18]. Further, the routing change should be apparent to the user.

A *routing history* is given by a starting routing status and a sequence of routing changes. From a routing history it is possible to reconstruct the routing status at each instant covered by the routing history.

A routing change happening at instant t can be one of the following types.

New route A collector-peer that is not connected at time right before t with the target AS now acquires connectivity using a specified AS-path.

Route change A collector-peer which is connected at time right before t with the target AS using AS-path p changes its connectivity by using AS-path q with $q \neq p$.

Route withdrawal A collector-peer which is connected at time right before t with the target AS, looses its connectivity.

Further, it is interesting to consider the following event that is not a routing change but can be a symptom of network problems.

Route re-announcement A collector-peer is connected at time t with the target AS and an announcement is received for the same route as if the route was new.

The collector-peers are partitioned in stable and unstable peers according to the following rules:

- if during the period of time covered by the considerd routing history a collector-peer is continuously connected with the target AS using the same AS-path, that collector-peer is considered stable,
- in all other cases the collector-peer is considered unstable.

To each stable collector-peers (and its stable AS-paths) a color is assigned according to the greedy algorithm described in Section 3. Distinct colors are assigned to all unstable collector-peers.

The union of all AS-paths that appear in the considered routing history forms a graph G in which the nodes are the involved ASes. A layout of the nodes of G is performed by using a spring-embedder algorithm as in Section 3. Such a layout remains unchanged during all the animation.

Animation is performed in the following way.

- AS-paths related to stable collector-peers are drawn dashed with their colors as in Section 3 and are not affected by any animation.
- AS-paths for new routes appear in their regular position (according to the layout of G). The attention of the user is caught by making the thickness of new AS-paths pulsing.
- Route changes are animated by means of a polyline morphing from the old AS-path to the new AS-path.
- Route withdrawals catch the attention of the user by a thickness pulse of the involved AS-path before it disappears.
- Route re-announcements catch the attention of the user by a thickness pulse of the involved AS-path.

The polyline morphing used to show route changes could be realized in several ways (see for example [4]). In BGPlay we adopted a rather simple technique which is illustrated in Figure 4. Let p be the polyline representing the old AS-path and p' be the polyline representing the new AS-path. A bijection between points of p and points of p' is defined in the following way. Polylines p and p' are consistently oriented incoming the target AS. Points in p are mapped with real numbers x in $[0 \dots 1]$ preserving the ordering. The same operation is performed on p'. Morphing between points which have the same value of x is performed linearly both in space and in time. In our case the two extremes of a polyline are always the same, namely the target AS and the collector-peer. The figure shows also the shape of the polyline in three intermediate instants. Morphing time is approximately set to one second.

6 Monitoring the Routes of the Traffic Incoming a Given AS with BGPlay

In this section we show how a user (e.g. the NOC of an ISP) can exploit the BGPlay service to monitor the evolution of the routing around an AS manager by the ISP.

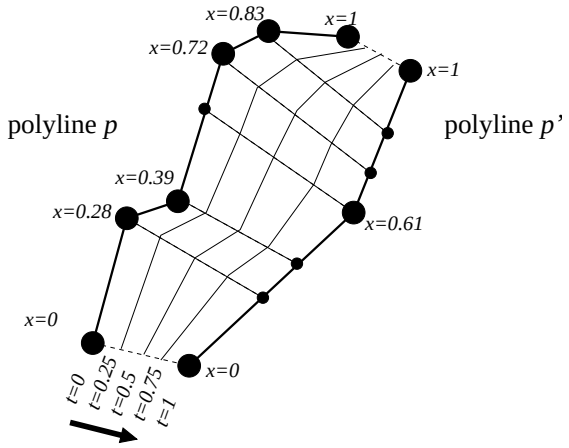


Fig. 4. Route changes are animated using a linear polyline morphing.

Suppose that the NOC is interested in monitoring the routing evolution concerning the prefix 193.0.0.0/21 from May 21, 2003 to May 23, 2003, because in that period some network instabilities have been somehow perceived. She/He fills the form of Fig. 5. Observe that the user is filling out the form selecting all the available sources of information (checkbox RRC00, RRC01, etc.).

BGPlay shows (Fig. 6) a routing graph that represents the routing status at the time of the first event in the selected time interval. Observe that the AS originating the prefix is 3333 and it is placed in the center of the window. The paths that represents stable routes are drawn dashed while the paths associated to unstable routes are drawn solid.

The buttons on the bottom allow the user to move through the sequence of events that happened in the specified time interval. Both forward and backward moves are possible. A time panel (on the left) shows the distribution of the routing events in the interval and an arrow indicates the time of the event just displayed.

Now, the user steps through the events. BGPlay shows the the evolution of the routing. The status bar (on top) indicates the event identifier, a timestamp, and the type of the event (see Section 5). Depending on the type, additional information is displayed:

New route The AS-path of the new route.
Route change Both the old and the new AS-path.
Route withdrawal The AS-path that is no longer valid.
Route re-announcement The AS-path that is re-announced.

References

1. Hermes. `http://www.dia.uniroma3.it/~hermes/`.

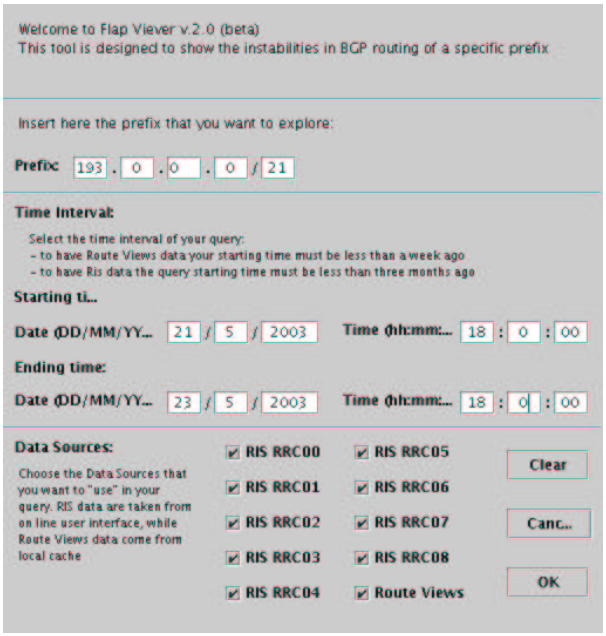


Fig. 5. The starting form.

2. Routing Information Service of the RIPE (RIS). `http://www.ripe.net/ripencc/pub-services/np/ris/`.
3. University of Oregon RouteViews project. `http://www.routeviews.org`.
4. S. Bespamyatnikh. An optimal morphing between polylines. *International Journal of Computational Geometry & Applications*, 12(3):217–228, 2002.
5. A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the high level structure of the internet with hermes. *J. of Graph Algorithms and Applications*, 6(3):281–311, 2002.
6. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.
7. G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia. Archives of bgp updates: Integration and visualization. In *Proceedings of IPS 2003, International Workshop on Inter-domain Performance and Simulation*, pages 123–129, 2003. online.
8. C. A. Duncan, A. Efrat, S. G. Kobourov, and C. Wenk. Drawing with fat edges. *Lecture Notes in Computer Science*, 2265:162–177, 2002.
9. P. Eades, R. F. Cohen, and M. L. Huang. Online animated graph drawing for web navigation. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 330–335. Springer-Verlag, 1997.
10. L. Gao and J. Rexford. Stable internet routing without global coordination. In *Measurement and Modeling of Computer Systems*, pages 307–317, 2000.
11. R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *IEEE INFOCOM 2000*, pages 1371–1380, Tel Aviv, Israel, March 2000.
12. T. Griffin and G. T. Wilfong. An analysis of BGP convergence properties. In *SIGCOMM*, pages 277–288, 1999.

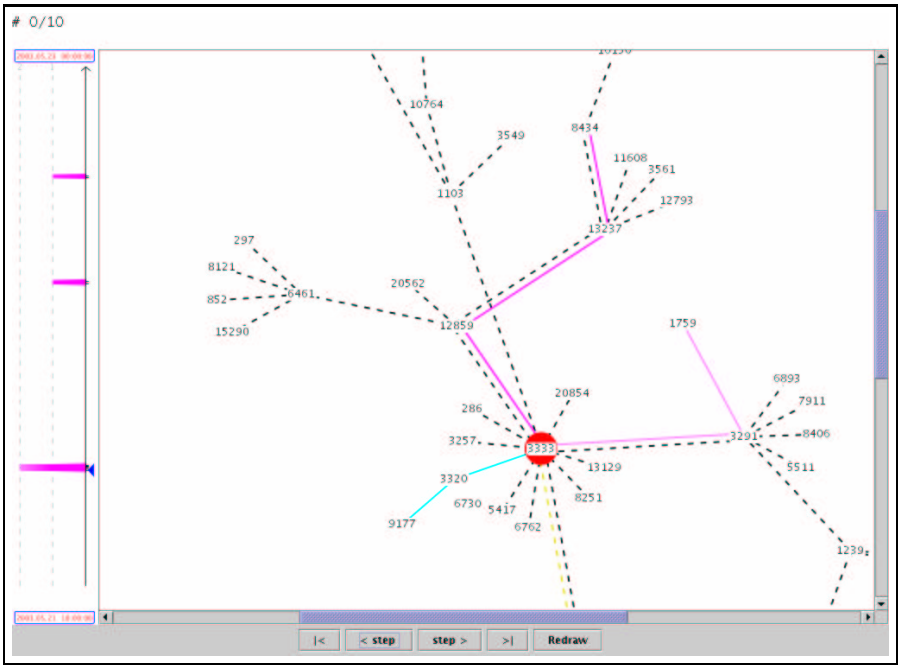


Fig. 6. A routing graph shown by BGPlay.

13. B. Huffaker, D. Plummer, D. Moore, and k claffy. Topology discovery by active probing. Technical report, Cooperative Association for Internet Data Analysis - CAIDA, San Diego Supercomputer Center, University of California, San Diego, 2002.
14. G. Huston. Interconnection, peering and settlements – part 1. *Internet Protocol Journal*, 2(1):2–16, 1999.
15. G. Huston. Interconnection, peering and settlements – part 2. *Internet Protocol Journal*, 2(2):2–23, 1999.
16. C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *SIGCOMM*, pages 175–187, 2000.
17. C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, 1998.
18. K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Lang. Comput.*, 6(2):183–210, 1995.
19. Y. Rekhter. A border gateway protocol 4 (BGP-4). IETF, RFC 1771.
20. N. Spring, R. Mahajan, and D. Wetherall. Measuring isp topologies with rocketfuel. In *Proceedings of ACM/SIGCOMM '02*, Aug. 2002.
21. J. W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, Reading, MA, 1999.
22. A. S. Tanenbaum. *Computer Networks*. Prentice-hall International, Inc., 1996. ISBN: 0-13-394248-1.