

# Specifiche del Progetto per il Corso di Complex Adaptive Systems (CAS) A.A. 2005-2006

Ozalp Babaoglu  
Stefano Arteconi  
Università di Bologna  
Dip. Scienze dell'Informazione

April 6, 2006

Utilizzando il simulatore Peersim, modificare i meccanismi di *drop* e *rewire* dell'algoritmo Slacer[1] e valutare le proprietà topologiche (clustering coefficient, average path length) e di cooperazione (livello di cooperazione, tempo necessario, LCC, CCP) ottenute partendo da un overlay arbitrario (star, scalefree, lattice,...). Le dimensioni della rete da considerare sono nell'intervallo [1000 : 10000] nodi<sup>1</sup>.

## Slacer

Slacer è un algoritmo in grado di produrre alto livello di cooperazione ed una topologia di tipo small world in una rete peer-to-peer.

Ogni nodo esegue una qualche applicazione che a sua volta definisce una strategia (es: velocità di upload e download in un sistema di file-sharing) ed un livello di utilità (es: numero di file scaricati).

In Slacer i nodi sono caratterizzati dalla propria vista (lista dei vicini) e, dalla strategia usata a livello applicazione (ad esempio: "cooperate" o "defect" in *Prisoners' Dilemma*) e da una misura di utilità ottenuta dall'esecuzione dell'applicazione.

Come semplice applicazione di test che rappresenti la contraddizione tra interesse egoistico e globale, viene usato *Prisoners' Dilemma*. Ogni nodo sceglie ad ogni ciclo un vicino casuale e gioca un singolo round di prisoner dilemma, ricevendo payoff relativo, come illustrato nella tabella 1.

	C	D
C	R,R	S,T
D	T,S	P,P

Table 1: Prisoners' Dilemma payoff table.  $T > R > P > S$  e  $2R > T + S$ . I singoli nodi sono spinti a scegliere la strategia D, a livello globale il caso migliore è quando entrambi i giocatori scelgono C.

Nel caso specifico la strategia consiste in "C" o "D" (rispettivamente comportamento cooperativo o egoistico), e l'utilità è la media dei payoff ricevuti nelle interazioni passate.

<sup>1</sup>Scegliere almeno due valori compresi nell'intervallo.

Il passo base dell'algoritmo Slacer eseguito periodicamente da ogni nodo consiste nel cercare di aumentare la propria utilità copiando i nodi con utilità maggiore. Questa azione, eseguita periodicamente da ogni nodo, descrive il passo base dell'algoritmo, che avviene secondo le seguenti azioni:

- Scelta di un nodo casuale  $j$
- Confronto dei valori di utilità (ad esempio  $U_i < U_j$ )
- **DROP**: distacco di ogni link di  $i$  con probabilità  $W$
- **REWIRE**: copia dei link di  $j$  da parte di  $i$
- Copia della strategia di  $j$  da parte di  $i$

In Slacer sono definite due diverse topologie overlay indipendenti: una topologia random (ottenuta tramite Newscast[2]) usata per confrontare l'utilità con gli altri nodi, ed un'ulteriore topologia (ottenuta tramite rewiring) che definisce le interazioni tra i nodi.

I metodi da implementare per ridefinire le azioni di DROP e REWIRE devono essere generati estendendo la classe *casproject.Slacer* contenuta nel package *casproject* reperibile all'indirizzo <http://www.cs.unibo.it/~arteconi>. Inoltre i seguenti parametri devono essere configurabili dal file di configurazione di peersim:

- **VIEWSIZE**: dimensioni della vista (o cache) locale del nodo; scegliere i valori nell'intervallo [10:40];
- Altri possibili parametri dipendenti dal meccanismo di rewire implementato (ad esempio  $W$  nella versione originale di Slacer)

I valori di tali parametri sono da considerare statici e non variabili durante la simulazione.

L'overlay di partenza può essere definito da una qualsiasi topologia di inizializzazione del simulatore tra quelle definite nel package *peersim.dynamics* (*WireKOut*, *WireRingLattice*, *WireStar*, *WireWS*...).

Scegliere ed utilizzare 2 inizializzazioni diverse. L'utilizzo di inizializzazioni diverse da quelle presenti sono opzionali.

Valutare le seguenti proprietà per ogni inizializzazione di partenza scelta, per ogni dimensione di rete e per 2 valori sia di VIEWSIZE che di ulteriori parametri di configurazione:

#### 1. Proprietà topologiche

- Clustering Coefficient (tramite *peersim.reports.GraphStats*)
- Average Path Length (tramite *peersim.reports.GraphStats*)

#### 2. Proprietà di cooperazione

- Livello di cooperazione raggiunto (tramite *StrategyObserver* contenuto nel package *casproject*, almeno 5 esperimenti)
- Tempo necessario per ottenere alto livello di cooperazione (tramite *StrategyObserver* contenuto nel package *casproject*, almeno 5 esperimenti)
- LCC e CCP (tramite *CCPObserver* contenuto nel package *casproject*).

3. (opzionale) Introdurre il churning<sup>2</sup> nel modello usando la classe: *DynamicNetwork* contenuta nel package *peersim.dynamics*. Considerare un solo scenario usato precedentemente (cioé una sola combinazione di NETSIZE, inizializzazione, VIEWSIZE e parametri di rewiring) e variando la quantità dei nodi che vengono sostituiti (3 prove, ad es: 10%, 20% e 40% ) rivalutare i punti 1 e 2.

## Visualizzazione Dati

Per visualizzare i dati raccolti dagli observer per i punti 1,2 e 3 si può utilizzare Gnuplot: un programma a linea di comando con cui é facile estrarre valori da file di testo. Un tutorial ed una breve guida introduttiva sono disponibili online. Brevemente, se si dispone di un file i cui valori sono organizzati in colonne (separate da spazi, tab o quant'altro), il comando:

```
> plot <nome-file> using 1:3 every 4 with lines
```

genererà una linea continua usando i dati della colonna 1 (asse X) e della colonna 2 (asse Y) estratti ogni 4 righe.

Si consiglia di consultare <http://www.gnuplot.info> per ulteriore documentazione.

## Modalità di consegna

Spedire via email, con oggetto "Progetto CAS 06", un file zip contenete il package *casproject* comprendente tutti i file sorgente creati o modificati, ed un elaborato di (max) 5 pagine di testo (scelte di design e motivazioni, descrizione dell'algoritmo, eventualmente supportato da spezzoni di codice, e commenti ai risultati ottenuti e ai relativi grafici generati) + lo spazio necessario alle figure all'indirizzo:

`arteconi@cs.unibo.it`

L'elaborato deve contenere cognome, nome ed indirizzo email del candidato. Il formato del file può essere pdf (preferibile), postscript o rtf.

Consegnare entro: 30 giugno 2006.

## References

- [1] D. Hales and S. Arteconi Friends for Free: Self-Organizing Artificial Social Networks for Trust and Cooperation *Technical Report UBLCS-2005-20*, Università di Bologna, Dipartimento di Scienze dell'Informazione, December 2005, Available at: <http://www.cs.unibo.it/pub/TR/UBLCS/2005/2005-20.pdf>
- [2] M. Jelasity and W. Kowalczyk and M. van Steen Newscast Computing *Technical Report IR-CS-006*, Vrije Universiteit Amsterdam, Department of Computer Science, November 2003, Available at: <http://www.cs.vu.nl/globe/techreps.html#IR-CS-006.03>

---

<sup>2</sup>Forma di dinamismo tipica dei sistemi P2P. Consiste in un flusso continuo di peer che entrano ed escono dalla rete.