

Incentivi alla cooperazione

Sistemi di reputazione

Matteo Dell'Amico
dellamico@disi.unige.it



Sistemi Distribuiti – P2P
A.A. 2007-08
6-7 dicembre 2007

Indice

- 1 Feldman et al.
 - Introduzione
 - Esperimento
 - Risultati

- 2 Conclusioni

Feldman et al.

M. Feldman, K. Lai, I. Stoica e J. Chuang, 2004

- Simulazione della funzione di reputazione **MaxFlow** in una rete P2P.

C\S	C	D
C	7,-1	0,0

- Gioco analogo a **donatore/ricevente**, visto precedentemente.
- **Defezioni non tracciabili**: chi rifiuta di collaborare non viene riconosciuto.
- **Popolazione dinamica**: i peer entrano nella rete, ne escono, cambiano strategia.

Dinamiche temporali

Round

- In ogni round ogni peer:
 - ① agisce come **client**;
 - ② agisce come **server**;
 - ③ **sceglie la strategia** per il prossimo round.

Nuova strategia

- Con una certa probabilità, il peer sceglie una di queste possibilità:
 - muta, scegliendo una nuova strategia;
 - impara, adottando la migliore strategia;
 - lascia il sistema, e viene sostituito da un nuovo arrivato con la stessa strategia.

Dinamiche temporali

Round

- In ogni round ogni peer:
 - ① agisce come **client**;
 - ② agisce come **server**;
 - ③ **sceglie la strategia** per il prossimo round.

Nuova strategia

- Con una certa probabilità, il peer sceglie una di queste possibilità:
 - muta, scegliendo una nuova strategia;
 - impara, adottando la migliore strategia;
 - lascia il sistema, e viene sostituito da un nuovo arrivato con la stessa strategia.

Strategie

Defezionatore

Cooperatore

Storia privata Valuta la generosità di un nodo i come il **rapporto** tra servizio che gli ha fornito e ha ricevuto $g(i) = p_i/c_i$, e la normalizza rispetto alla sua stessa generosità. j coopera con i con probabilità

$$g_j(i) = \min \left(\frac{g(i)}{g(j)}, 1 \right)$$

Storia condivisa Funziona analogamente alla storia privata, valutando però l'**intera storia passata** del nodo.

Strategie

Defezionatore

Cooperatore

Storia privata Valuta la generosità di un nodo i come il **rapporto** tra servizio che gli ha fornito e ha ricevuto $g(i) = p_i/c_i$, e la normalizza rispetto alla sua stessa generosità. j coopera con i con probabilità

$$g_j(i) = \min \left(\frac{g(i)}{g(j)}, 1 \right)$$

Storia condivisa Funziona analogamente alla storia privata, valutando però l'intera storia passata del nodo.

Strategie

Defezionatore

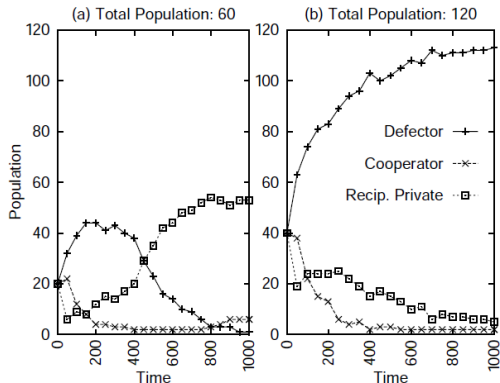
Cooperatore

Storia privata Valuta la generosità di un nodo i come il **rapporto** tra servizio che gli ha fornito e ha ricevuto $g(i) = p_i/c_i$, e la normalizza rispetto alla sua stessa generosità. j coopera con i con probabilità

$$g_j(i) = \min \left(\frac{g(i)}{g(j)}, 1 \right)$$

Storia condivisa Funziona analogamente alla storia privata, valutando però **l'intera storia passata** del nodo.

Storia privata



Parametri

- Probabilità di mutazione 0, apprendimento 0.05, turnover 0.0001.

Collusione

- Un **Sybil attack** può sovvertire il risultato calcolato dalla storia condivisa. Maxflow, invece, è sybilproof.
- Il peso degli archi nella WoT è il **numero di transazioni** andate a buon fine.

Collusione

- Un **Sybil attack** può sovvertire il risultato calcolato dalla storia condivisa. Maxflow, invece, è sybilproof.
- Il peso degli archi nella WoT è il **numero di transazioni** andate a buon fine.

Strategia soggettiva

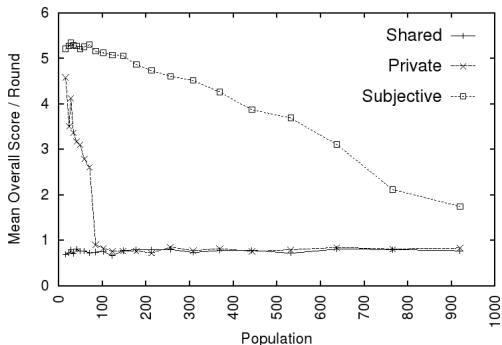
Strategia soggettiva

- Valuta la generosità di un nodo come il **flusso** che lo collega ad esso, e la rapporta al flusso nella direzione inversa. Quindi, il nodo j coopera con i con probabilità

$$g_j(i) = \min \left(\frac{\text{MaxFlow}(j, i)}{\text{MaxFlow}(i, j)}, 1 \right).$$

- MaxFlow ha **complessità** in $O(V^3)$. Per evitare un sovraccarico, se la durata media del calcolo ha superato un limite fissato, viene restituito come risultato un flusso nullo.

Risultati sperimentali



- Il punteggio medio scende al crescere della popolazione a causa del vincolo sul tempo di esecuzione di MaxFlow (in questo caso, la visita di 100 nodi).

Conclusioni

- I sistemi di reputazione sono sicuramente **promettenti** per progettare sistemi P2P su larga scala.
- Allo stato dell'arte, nessun sistema è perfetto:
 - I sistemi semplici basati sulla reciprocazione diretta non sono ottimali e funzionano solo in **casi ristretti** (per esempio, lo scambio di file di grandi dimensioni);
 - MaxFlow è un algoritmo **costoso**, e la sua efficacia decresce con la dimensione della rete su cui viene calcolata;
 - EigenTrust è in qualche maniera suscettibile al Sybil attack, e richiede dei nodi **"pre-trusted"** che sono problematici da gestire.
- Inoltre, nessun sistema di reputazione non banale è stato implementato in reti di grandi dimensioni. Vari problemi "pratici" (p. es., scheduling) sono stati al momento ignorati.