

# Ingegneria del Web 07/08

Simone Notargiacomo

Università di Roma Tor Vergata

June 29, 2008

**TORTELLA**

## 1 Sommario

## 2 Protocollo trasporto

HTTP

Incapsulamento dati

## 3 Servent

Server e Client

BootStrap

## 4 Flooding

Search e SearchHits

Join e Leave

# Definizione header

- Utilizzati header protocollo versione "1.1".
- Realizzati metodi POST e GET.
  - I POST trasferiscono tutti i pacchetti TorTella.
  - I GET servono per richiedere il trasferimento di file.

## Pacchetto di richiesta:

```
POST * HTTP/1.1
User-Agent: TorTella/0.1
Connection: Keep-Alive
Content-Length: <num>
...data...
```

## Pacchetto di risposta:

```
HTTP/1.1 200 OK
Server: TorTella/0.1
Content-Type:
application/binary
Content-Length: <num>
```

# Motivazioni

- Utilizzati header strettamente necessari al protocollo.
- Scelto metodo POST perchè consigliato per l'invio di dati.
- *Keep-Alive* per consentire una connessione persistente fra i peer.
- Vari tipi di pacchetti di risposta per segnalare l'avvenuto invio o eventuali errori.

# Dati TorTella

## Pacchetti TorTella incapsulati nel campo dati dell'HTTP

- Dati convertiti in forma binaria ed inseriti dopo l'header HTTP.
- Lunghezza dati specificata nell'header HTTP.
- Pacchetto risposta al GET può incapsulare parte di un buffer.

# Server e Client

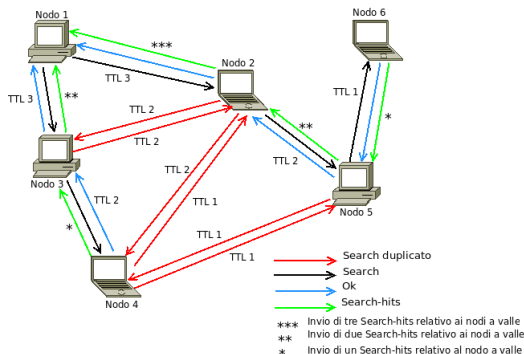
- Creazione di due Thread per ogni peer.
- Il Server Thread gestisce i pacchetti ricevuti.
- Il Client Thread gestisce l'invio dei pacchetti.
- I Thread rimangono attivi fino alla disconnessione dal peer.

# BootStrap

- Necessario per la connessione alla rete TorTella.
- Necessario reperire gli indirizzi di alcuni peer.
  - Passaparola tra utenti.
  - Uso di un server centralizzato (non implementato).
- Connessione ai peer attivi della lista.
- Per ogni peer si lancia un nuovo client thread.

# Descrizione

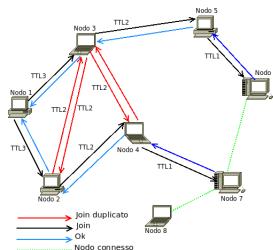
- La ricerca viene effettuata in flooding.
- Coinvolti tutti i peer nel raggio del TTL.
- I risultati sfruttano il backward routing.
- Realizzate tabelle di routing apposite.





# Join

- Il Join viene inviato a tutti i vicini.
- Evita inconsistenza dei dati.
- Diminuisce la probabilità di ricerca nulla.
- Tutti gli utenti connessi alla chat aggiungeranno l'utente.



# Leave

- Anche il Leave viene inviato a tutti i vicini.
- Tutti devono sapere che un peer sta lasciando la chat.
- Evita di inviare risultati errati dopo una ricerca.