

Valutazione delle prestazioni e  
Sistemi Distribuiti  
Dipartimento di Informatica  
Universita' del Piemonte Orientale

Cosimo Anglano  
cosimo.anglano@mfn.unipmn.it

Roma, 9-10 Giugno 2004

# Composizione del gruppo

---

- Cosimo Anglano
- Andrea Bobbio
- Giuliana Franceschinis
- Massimo Canonico
- Daniele Codetta-Raiteri
- Massimiliano De Pierro
- Andrea Ferrino

# Attività' di ricerca principali

---

- Tematiche sistemistiche:
  - Tecniche per la gestione delle risorse (scheduling) per cluster e piattaforme per on-demand computing
  - Tecniche per il trasferimento dati su reti best-effort
  - Sistemi Peer-to-Peer (DHT) per la memorizzazione ed il reperimento di informazioni
- Tematiche modellistiche
  - studio di formalismi (anche multiformalismo)
  - costruzione incrementale e modulare dei modelli
  - tecniche efficienti di soluzione
  - sviluppo di tools a supporto

# Progetti di Ricerca

---

- FIRB *WebMINDS* (coord. G. Chiola)
- FIRB *Perf* (coord. M. Calzarossa)
- CNR-MIUR "*Grid Computing: Tecnologie abilitanti ed applicazioni per eScience*" (coord. D. Laforenza)
- *E-Biogrid* (all'interno della Network of Excellence *EMBRACE* – attualmente in fase di negoziazione)

# Meccanismi Middleware per il Supporto alla QoS in Piattaforme per Computing On-Demand

Roma, 9-10 Giugno 2004

# Piattaforme per on-demand computing

---

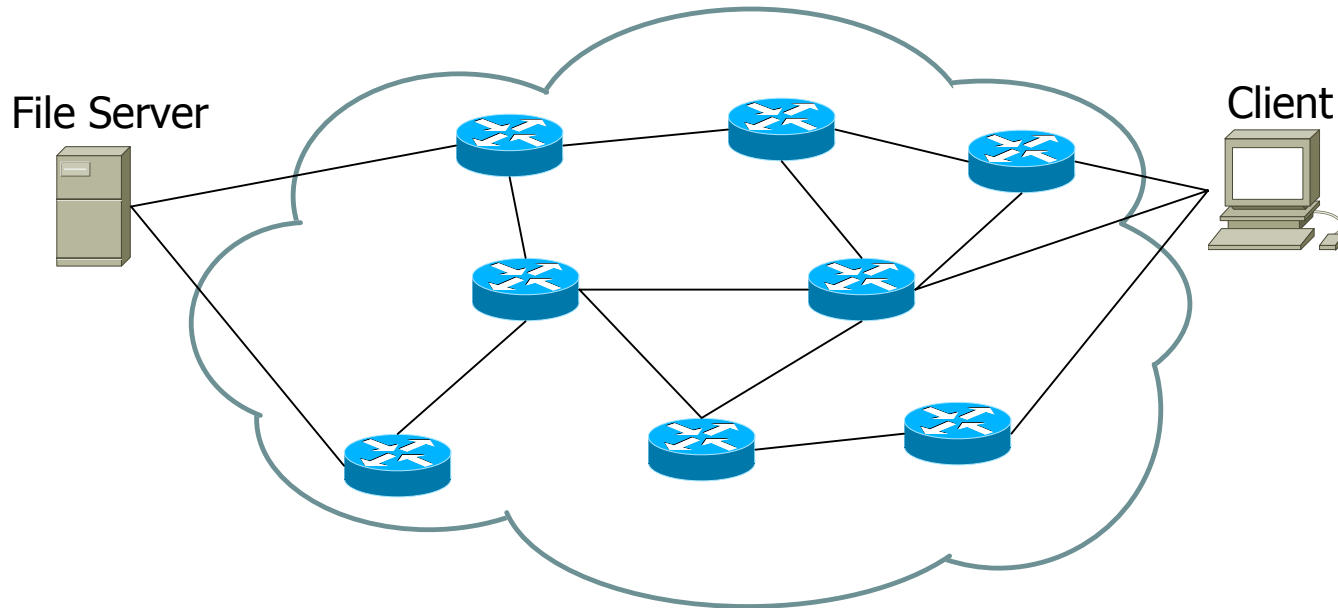
- Piattaforme distribuite basate su risorse non necessariamente proprie e reti best-effort
  - “prestito” o “noleggio” di risorse altrui per l’esecuzione di applicazioni
  - utility computing, on-demand computing, grid computing, peer-to-peer resource sharing, ecc.
- Obiettivi:
  - utilizzo efficiente delle risorse
  - garanzie (sia pur minimali) sulla qualità dei servizi erogati (es. tempo massimo di risposta/esecuzione/erogazione)
- Necessita’ di meccanismi che permettano di raggiungere questi obiettivi:
  - meccanismi efficienti e predicibili per il trasferimento di file
  - scheduling fault-aware di applicazioni

# Meccanismi per file transfer

---

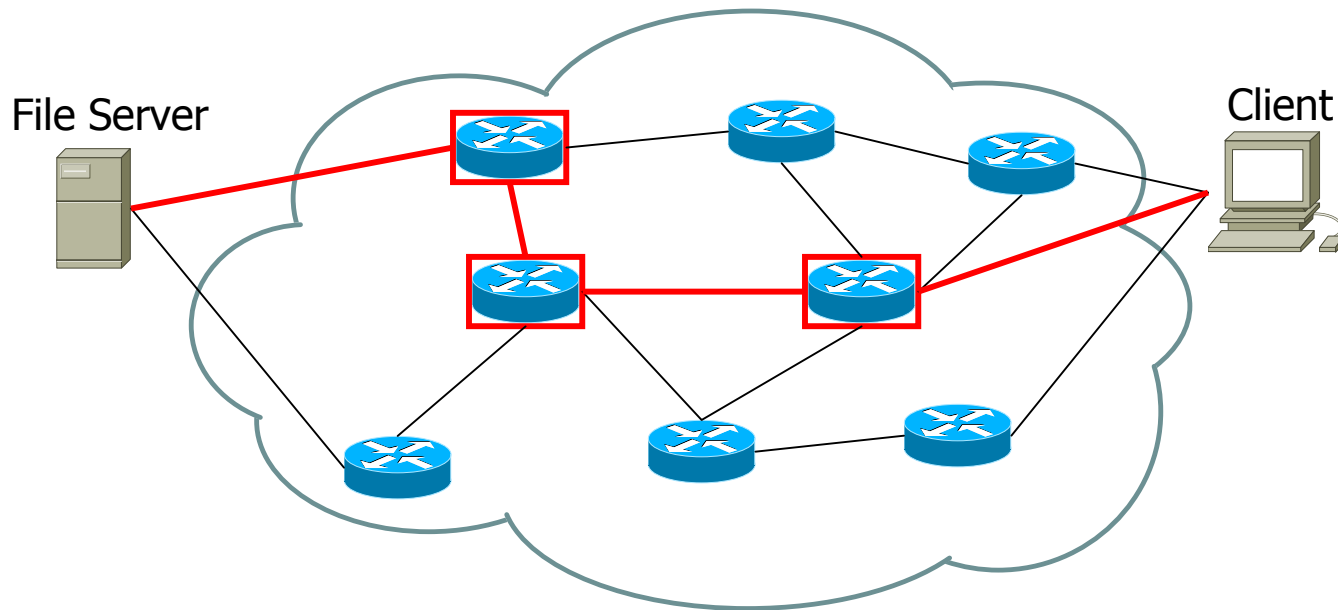
- Tecniche per il trasferimento di file
  - efficienti, per ridurre il tempo di trasferimento
    - ... e di conseguenza il tempo di esecuzione di applicazioni e/o di erogazione di servizi
  - che permettano la stima dei tempi di trasferimento
    - gestione oculata delle risorse
    - calcolo di stime attendibili dei tempi di completamento di applicazioni e/o di erogazione di servizi
- Tecniche attuali basate su protocolli di trasporto (end-to-end) inadatte

# File Transfer: soluzioni end-to-end



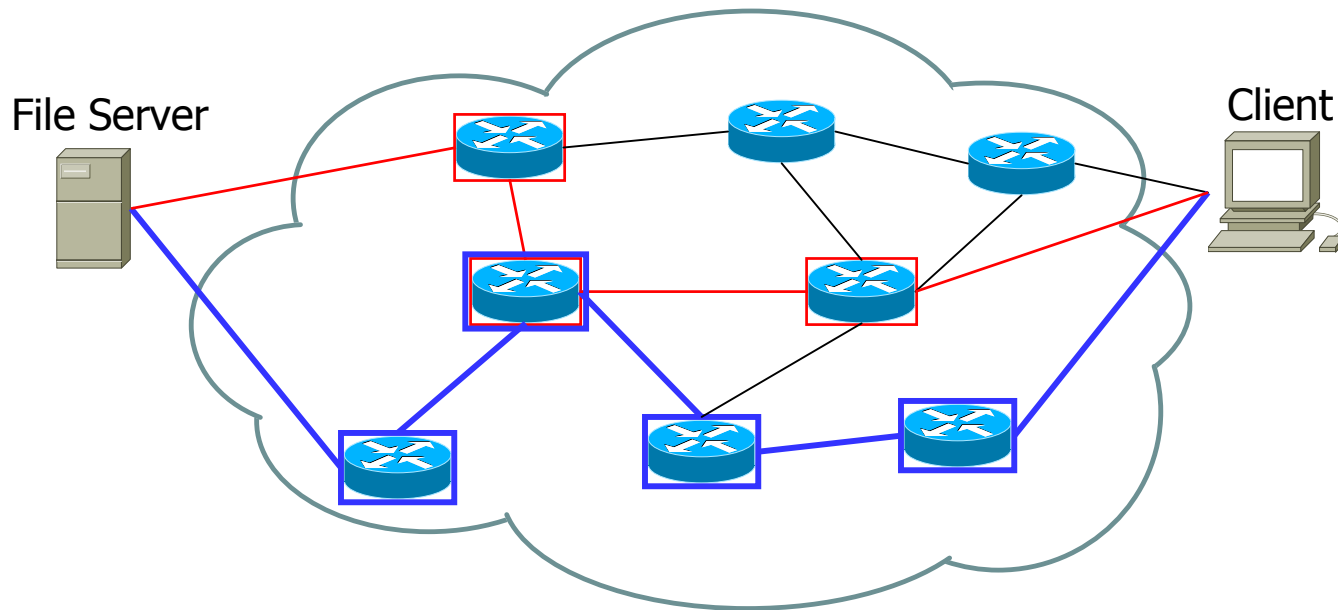


# File Transfer: soluzioni end-to-end



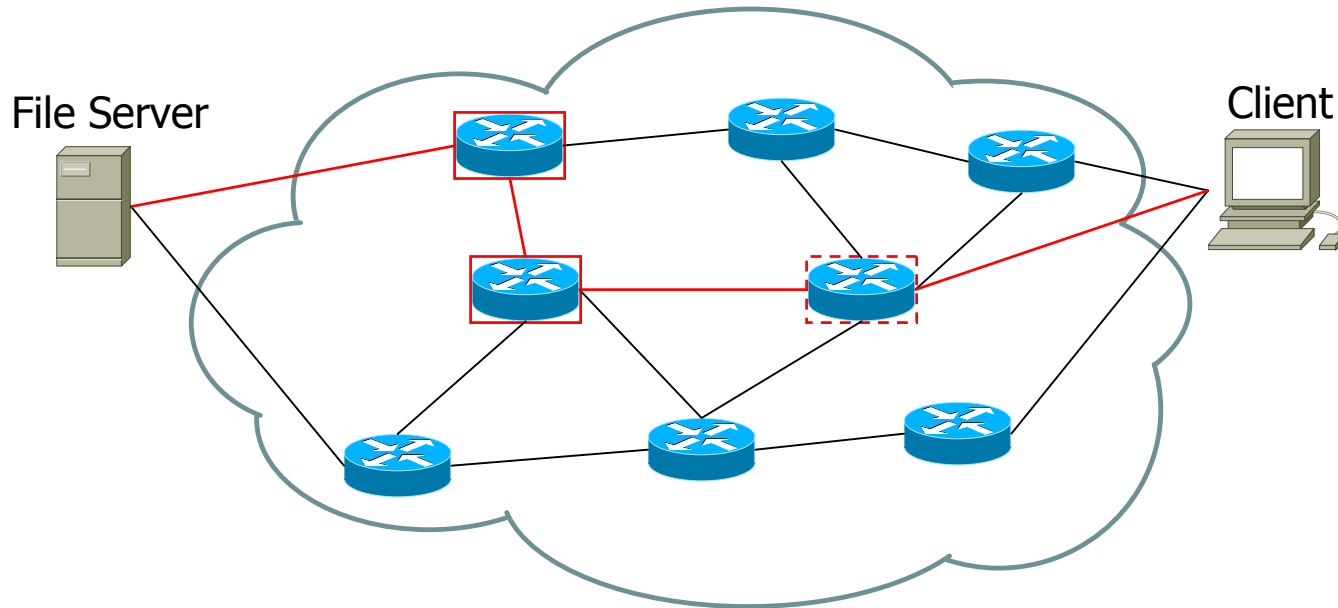
- Scelta del cammino utilizzato per il trasferimento effettuata a livello IP

# File Transfer: soluzioni end-to-end



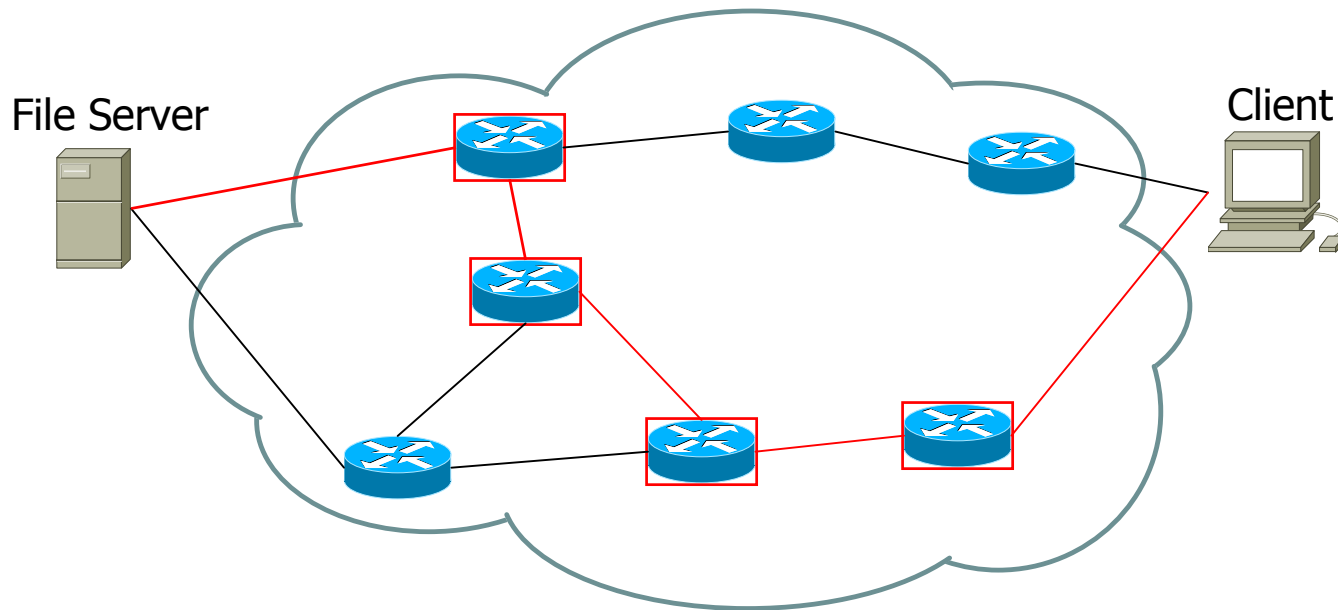
- IP non e' in grado di sfruttare eventuali cammini alternativi caratterizzati da migliori prestazioni

# File Transfer: soluzioni end-to-end



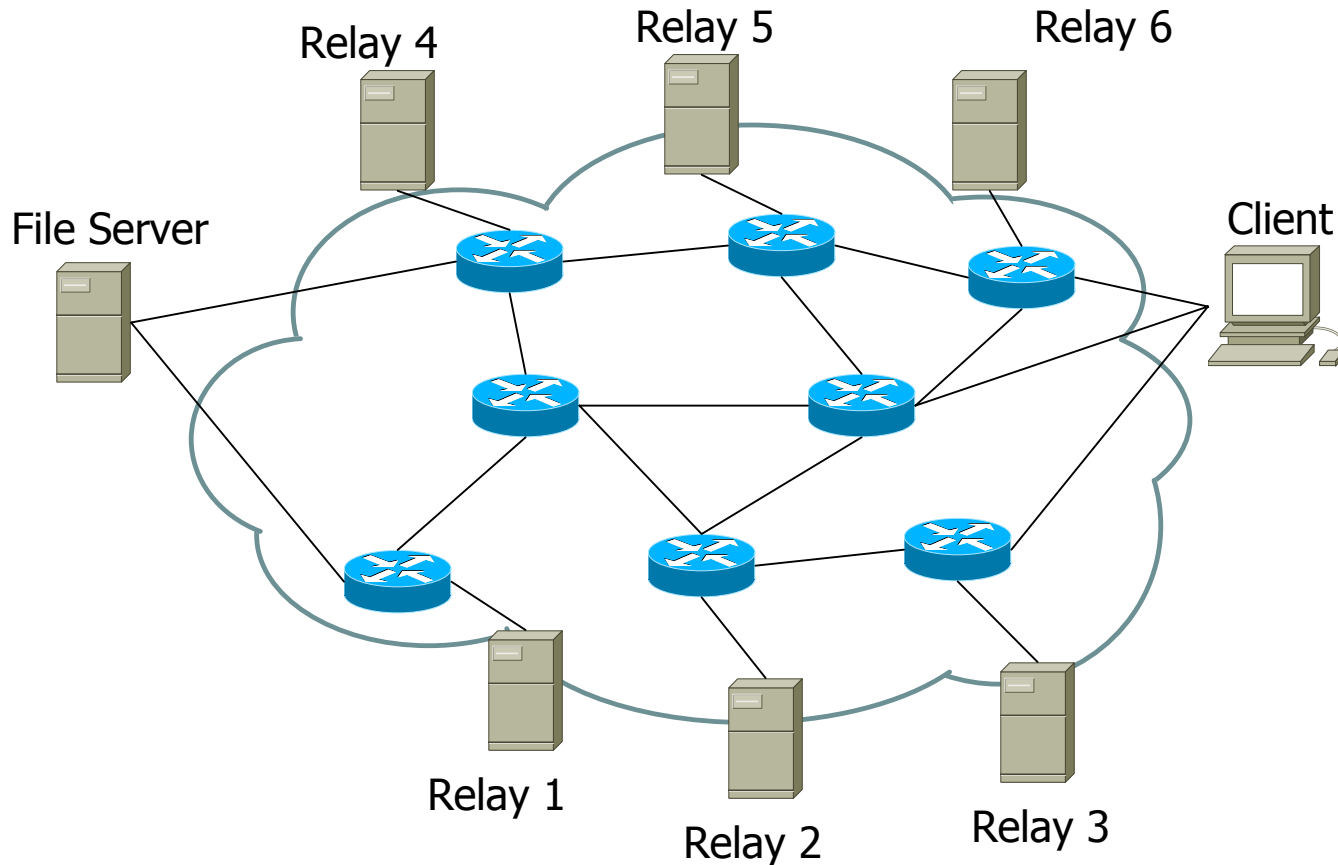
- In caso di guasti sul cammino prescelto ...

# File Transfer: soluzioni end-to-end



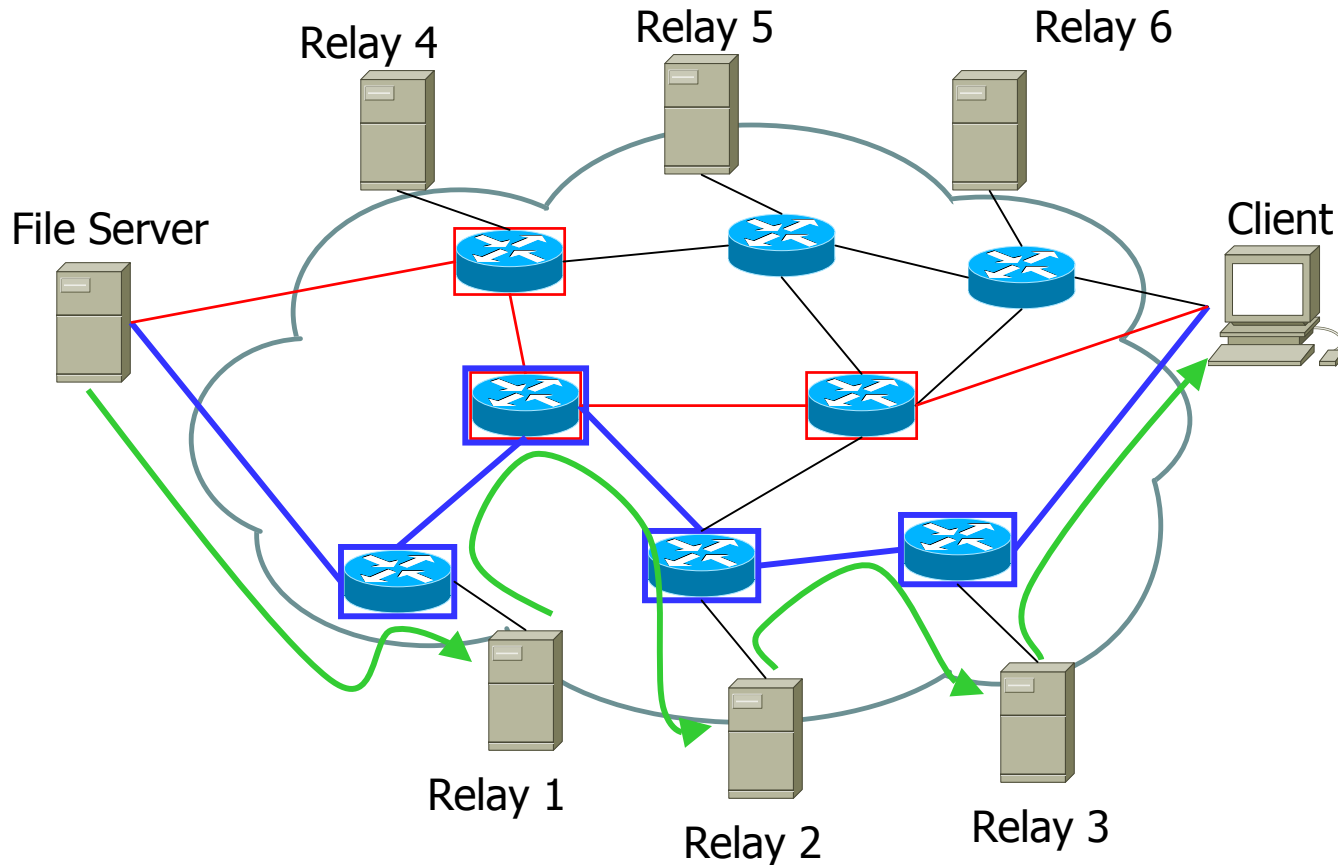
- ... possono trascorrere parecchi minuti prima che le tabelle di routing siano aggiornate

# Il File Mover: idea di base



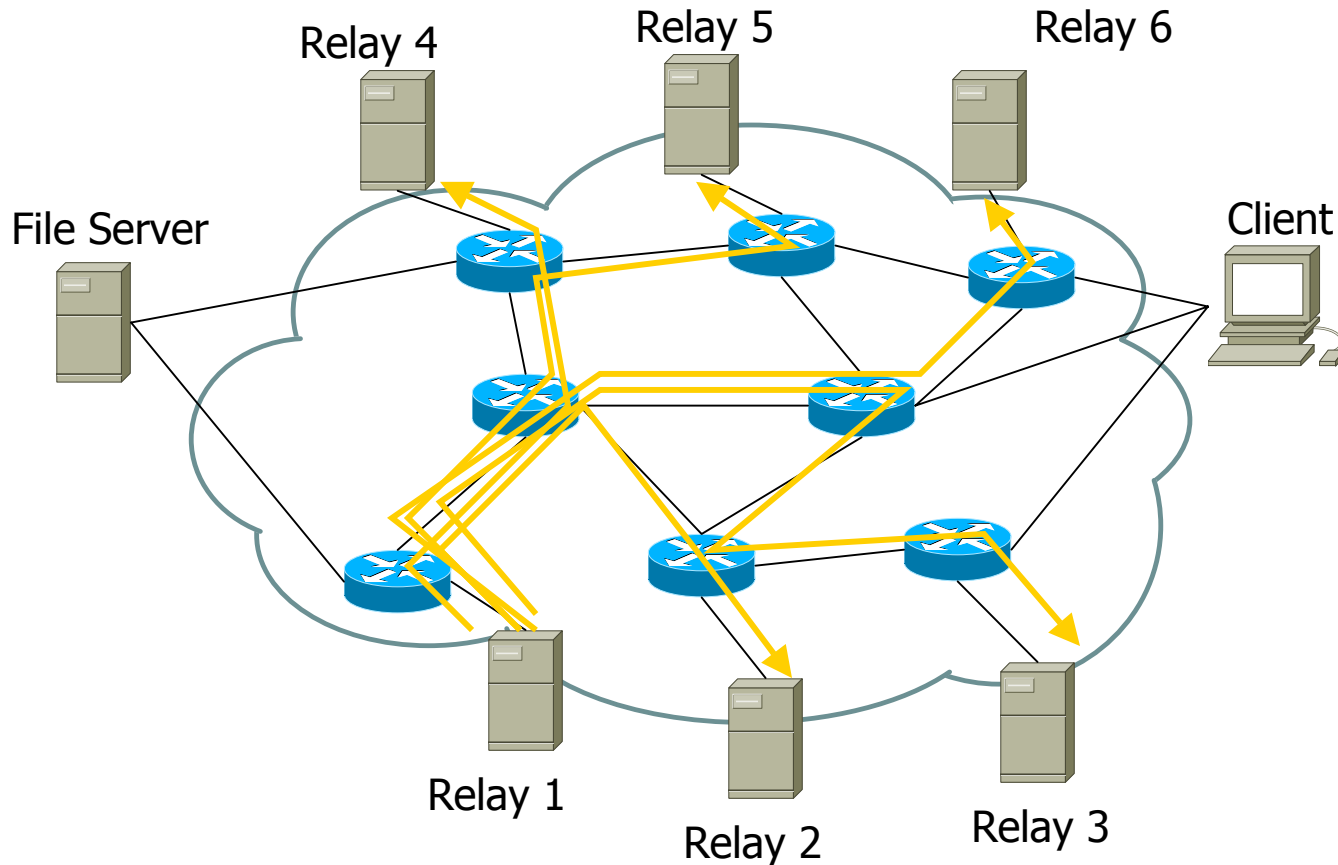
- Utilizzo di calcolatori (File Relay) che cooperano nel trasferimento di file

# Il File Mover: trasferimento di file



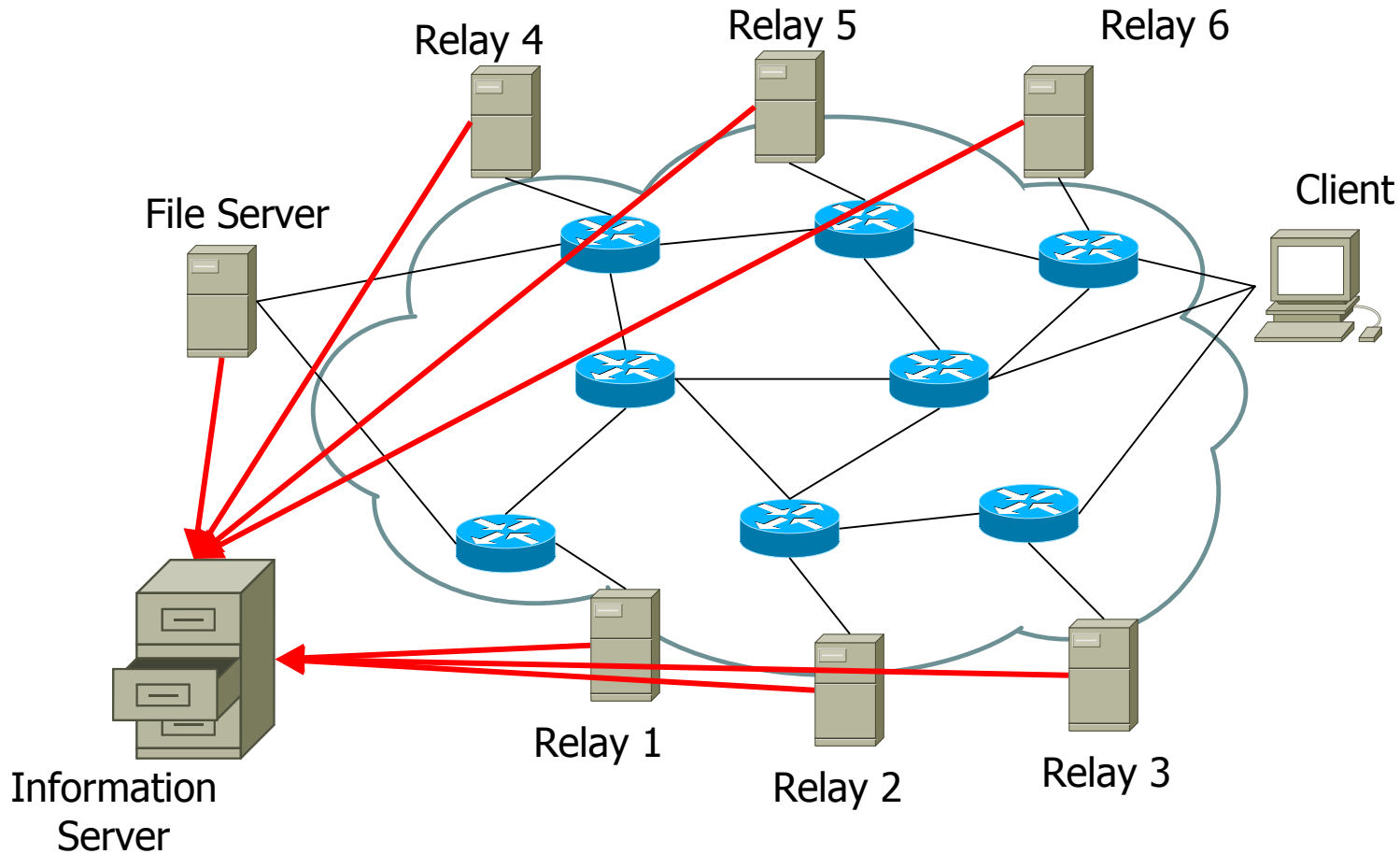
- I Relay fungono da router a livello applicativo
  - routing cut-through

# Il File Mover: il routing



- Raccolta periodica di misurazioni della bandwidth disponibile tra ogni coppia di Relay ...

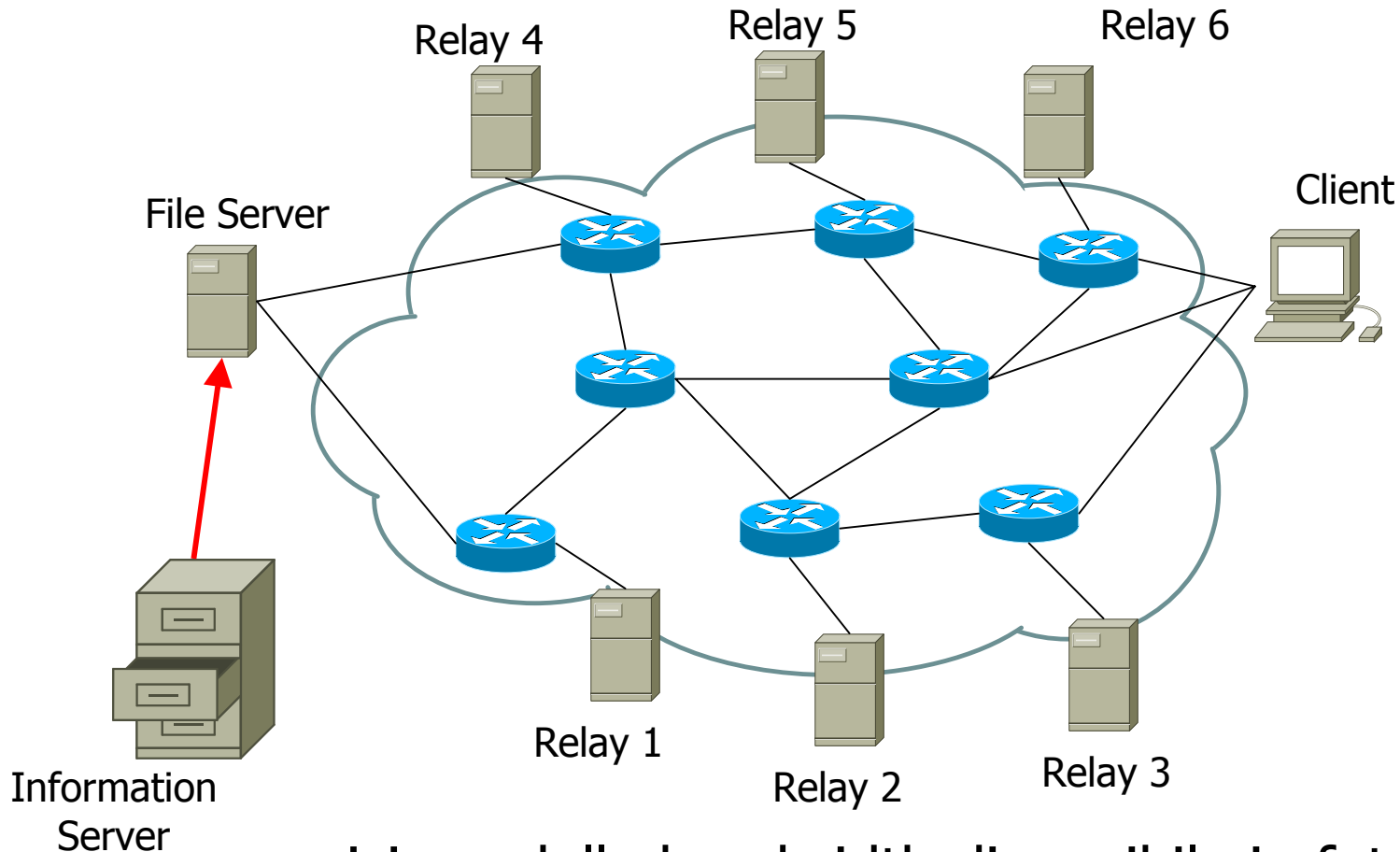
# Il File Mover: il routing



- ... memorizzazione in un database ...



# Il File Mover: il routing



- ... e previsione della bandwidth disponibile in futuro per i vari virtual link
  - mediante il sistema Network Weather Service

# Il File Mover: altre caratteristiche

---

- Riconfigurazione dinamica di un cammino se cambiano le condizioni della rete
- Gestione di guasti/indisponibilità di Relay

# File Mover: risultati preliminari

- Sperimentazione compiuta su testbed distribuiti
  - Emulab della University of Utah
  - attualmente in corso su PlanetLab
  - C. Anglano, M. Canonico. The File Mover: An Efficient Data Transfer System for Grid Applications. Proc. of Int. Workshop on Grid and Advanced Networks, April 2004.

Experiment	Source	Destination	File Mover				FOBS		
			Path	Est. Thr.	Transfer time	Eff. Thr.	Est. Thr.	Transfer time	Eff. Thr.
1	utah	seal	utah → digitalwest → seal	11.5	47.24	8.48	5.46	61.67	6.49
2	utah	syrah	utah → digitalwest → syrah	32.9	17.12	24.32	5.73	42.9	9.36
3	mit-main	seal	mit-main → nyu → intel → seal	3.65	125.9	3.2	2.58	330.9	1.21
4	utah	syrah	utah → syrah	5.73	40.5	9.88	5.73	41.15	9.49
5	mit-main	syrah	mit-main → nyu → intel → seal	3.77	123.3	3.26	3.08	41.42	9.69
6	mit	seal	mit → nyu → intel → seal	3.76	125.2	3.21	2.73	68.9	5.85
7	nortel	seal	nortel → nyu → intel → seal	3.64	118.6	3.37	2.6	83.28	4.8
8	nortel	syrah	nortel → nyu → intel → syrah	3.77	116.9	3.42	3.1	69.3	5.77
9	aros	syrah	aros → syrah	7.1	69.3	5.77	7.1	42.1	9.5
10	cornell	seal	cornell → nyu → intel → seal	3.75	118.2	3.38	2.91	unavail.	unavail.
11	cmu	seal	cmu → nyu → intel → seal	3.63	117.5	3.4	2.95	unavail.	unavail.

# File Mover: sviluppi in corso

---

- Incremento delle prestazioni
  - impiego di tecniche di trasferimento dati di tipo rate-based basate su UDP
  - tecniche di caching sui Relay e trasferimenti "striped" da più Relay verso lo stesso client
  - sfruttamento di percorsi multipli tra un server ed un client
- Incremento dell'accuratezza delle stime dei tempi di trasferimento
  - attualmente solo stime puntuali dei tempi di trasferimento
  - applicazione di tecniche di stima dei quantili in modo da determinare la probabilità che un trasferimento termini entro un dato istante di tempo

# Fault-aware scheduling

---

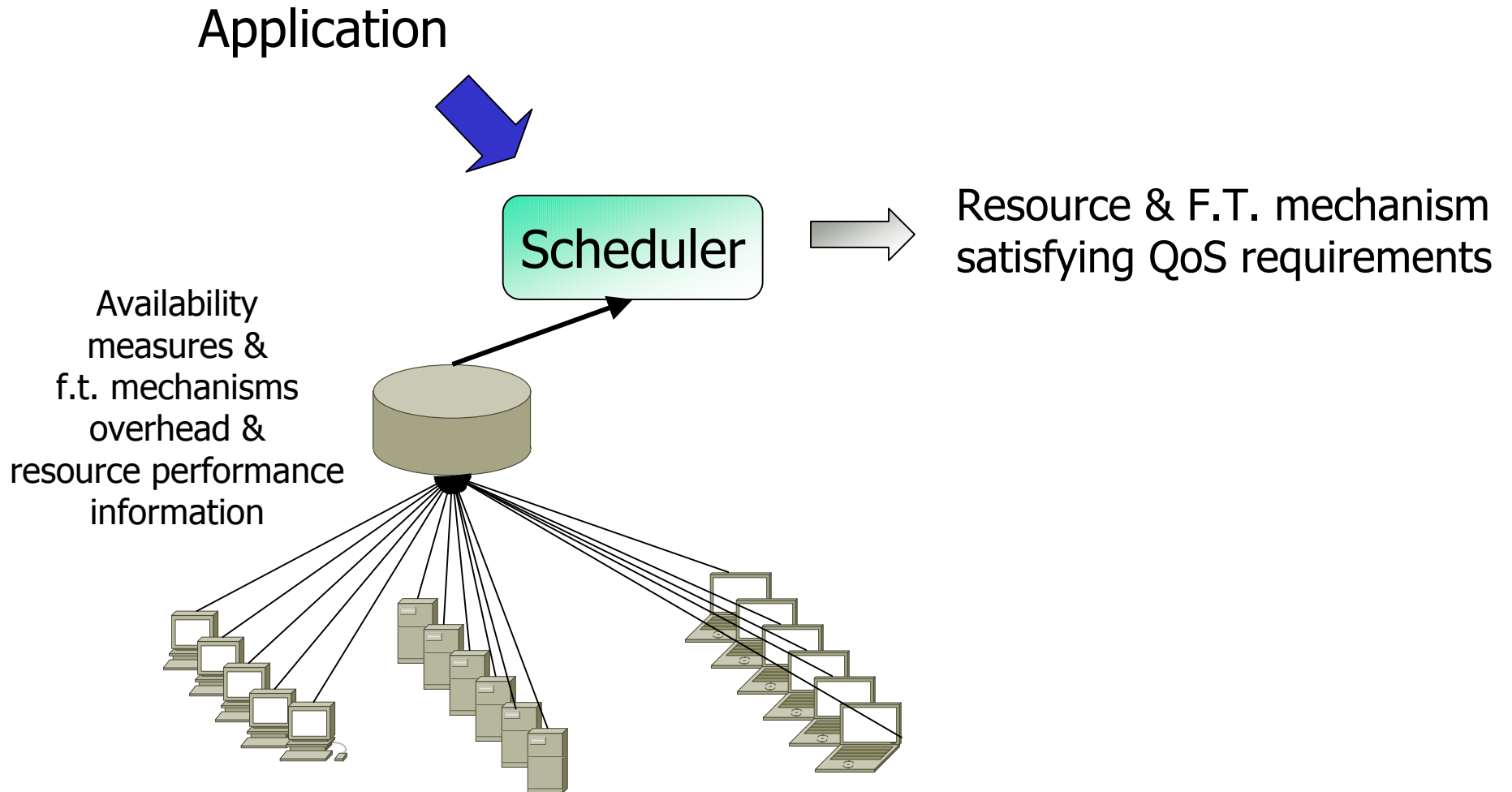
- Nei sistemi per computing on-demand le risorse possono essere molto “volatili”:
  - rimozione dal sistema da parte del proprietario
  - alta probabilità di indisponibilità (temporanea o permanente) dovute a guasti (hw e/o sw) delle risorse e/o dell’infrastruttura di rete
- La scelta delle risorse su cui eseguire un’applicazione deve tenere conto non solo delle prestazioni, ma anche della disponibilità delle stesse
  - una risorsa più lenta ma più disponibile di un’altra più veloce può rappresentare una scelta migliore

# Fault-aware scheduling

---

- Sviluppo di un framework caratterizzato dalle seguenti proprietà:
  - QoS espressa in termini di prestazioni e disponibilità
    - es. "terminare entro 3 ore con probabilità del 99.9%" oppure "servizio disponibile al 99.99% su un periodo di 2 mesi con tempi massimi di erogazione di 2 sec."
  - Disponibilità di vari meccanismi per l'affidabilità di applicazioni (replicazione di processi, checkpoint e rollback)
  - Scelta automatica delle risorse e del meccanismo di affidabilità che permettano di soddisfare i vincoli di QoS dell'applicazione

# Fault-aware scheduling



# Fault-aware scheduling

---

- Sviluppo di modelli per la caratterizzazione dell'overhead dei meccanismi di fault tolerance
- Sviluppo di algoritmi di scheduling fault-aware per alcune tipologie di applicazioni (bag-of-task)
- Sviluppo di librerie di checkpointing in grado di misurare automaticamente l'overhead da esse indotto sulle applicazioni
- Sistema distribuito per la memorizzazione ed il reperimento di checkpoint