File System Distribuiti
NFS

File System Distribuito

- Un **file system distribuito** è un file system residente su più computer che offre una ***vista integrata*** dei dati memorizzati sui diversi dischi residenti su un insieme di computer.

- Alcuni esempi di file system distribuiti

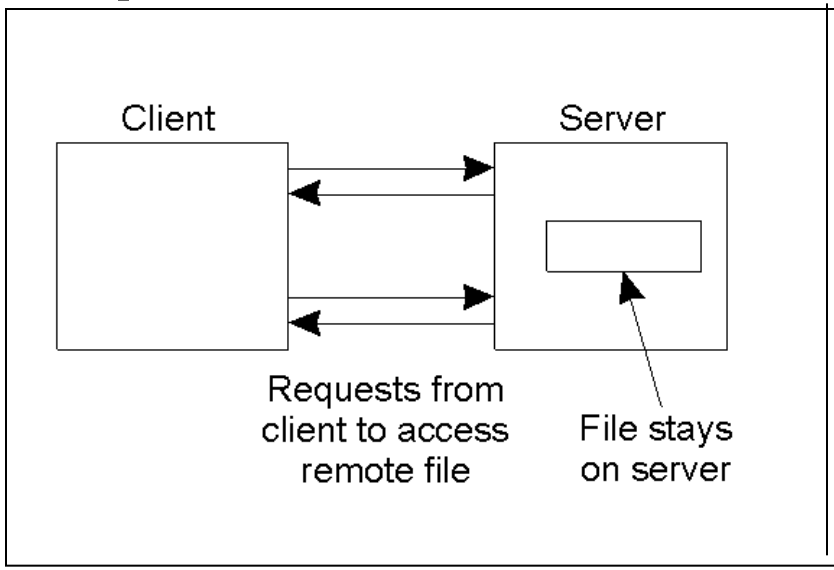
  - **NFS**
  - AFS
  - **Coda**
  - **HDFS**
  - xFS
  - Lustre
  - GFS
  - XtreemFS
  - GPFS
  - LizardFS

Network File System (NFS)

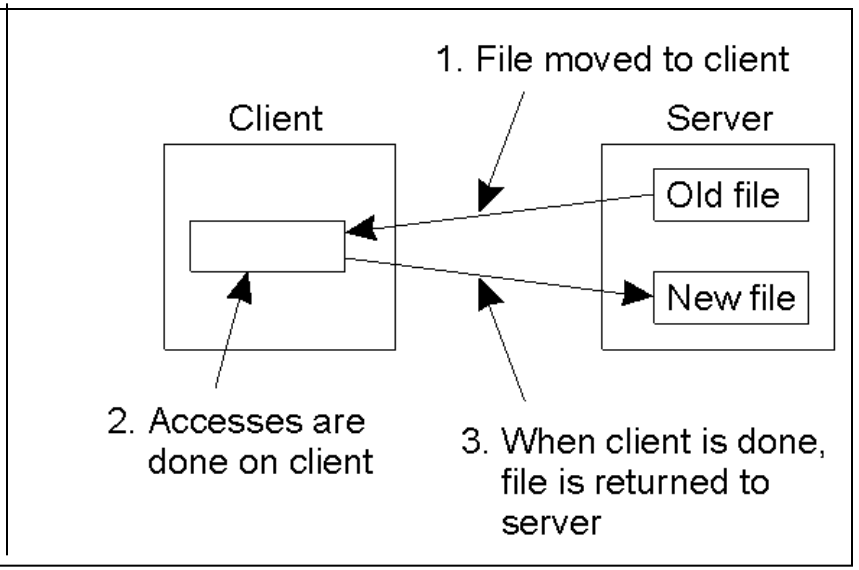
- Originariamente sviluppato alla Sun Microsystems per le workstation SUN con sistema operativo UNIX.
- E' un modello per integrare file system differenti.
- Basato sull'idea che ogni file server fornisce una vista unificata del suo file system locale.
- NFS può essere usato su gruppi **eterogenei** di computer.

Architettura di NFS (1)

- NFS nelle prime versioni usava solo il modello **remote access**:
  - I nodi clienti non conoscono le reali locazioni dei file.
  - I server esportano un set di operazioni sui file.
- Nell'ultima versione usa anche il modello **upload/dowload**

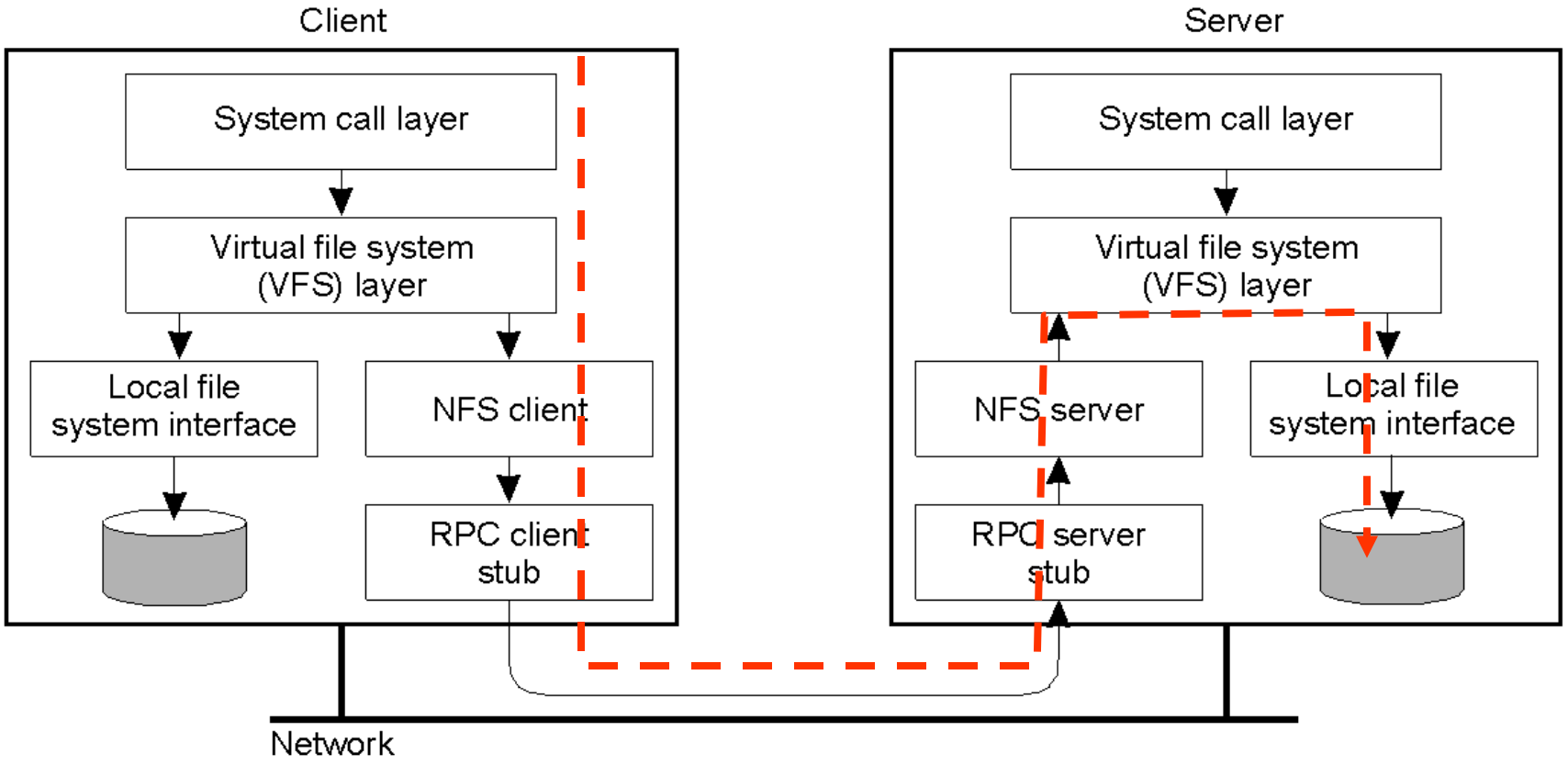


Il modello remote access.



Il modello upload/download.

Architettura di NFS (2)



L'architettura di base di NFS per sistemi UNIX.

Architettura di NFS (3)

- NFS è indipendente dall' organizzazione del file system locale.
- Integra file systems usati in UNIX, Linux, Windows, e altri sistemi operativi.
- Il modello di file system offerto all' utente è simile a quello dei file system UNIX-like, basato su files organizzati come sequenze di byte.

Modello del File System

Operazione	v3	v4	Descrizione
`Create`	Si	No	Crea un file
`Create`	No	Si	Crea un file non regolare (link simbolici, directory, file speciali)
`Link`	Si	Si	Crea un hard link ad un file
`Symlink`	Si	No	Crea un symbolic link ad un file
`Mkdir`	Si	No	Crea una subdirectory in una data directory
`Mknod`	Si	No	Crea un file speciale
`Rename`	Si	Si	Cambia il nome ad un file
`Rmdir`	Si	No	Rimuove una subdirectory vuota da una directory
`Open`	No	Si	Apre un file
`Close`	No	Si	Chiude un file
`Lookup`	Si	Si	Accede ad un file tramite il filename
`Readdir`	Si	Si	Legge Il contenuto di una directory
`Readlink`	Si	Si	Legge il pathname memoriazzato in un symbolic link
`Getattr`	Si	Si	Legge gli attributi di un file
`Setattr`	Si	Si	Modifica gli attributi di un file
`Read`	Si	Si	Legge i dati contenuti in un file
`Write`	Si	Si	Modifica i dati contenuti in un file

Una lista incompleta di operazioni sul file system supportati da NFS

Comunicazioni (1)

a) In NFS tutte le comunicazioni tra i server e clienti sono implementate tramite Remote Procedure Call (RPC).

b) Il protocollo usato é: Open Network Computing RPC.

c) ONC RPC usa anche operazioni per la trasformazione degli argomenti basate sulla specifica **eXternal Data Representation**(XDR) per codificare e decodificare I dati tra nodi NFS diversi.

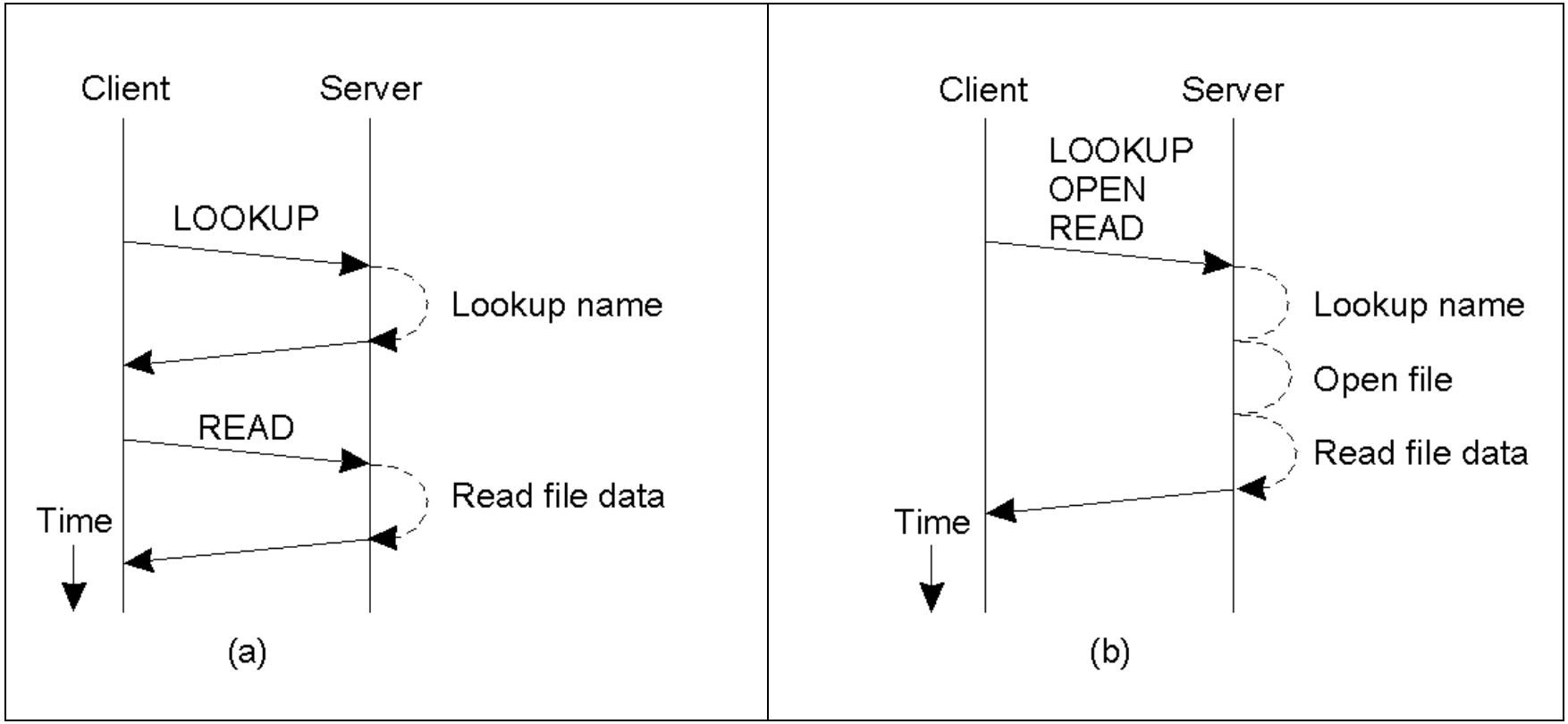
d) Prima della versione 4, NFS usava server **stateless**.

e) I clienti avevano il compito di mantenere lo stato delle operazioni correnti su un file system remoto.

Comunicazioni (2)

- Nella versione 4, NFS ha introdotto le **compound operations** che comprendono più richieste di operazioni in una singola chiamata.
- Usate per ridurre il numero di chiamate RPC e migliorare le prestazioni delle comunicazioni.
- Questo approccio è particolarmente adatto a **wide-area** file systems.
- Le compound operations non vengono gestite come transazioni:
  - Se una operazione in una compound procedure fallisce, le successive operazioni non vengono eseguite.
  - Viene ritornato un messaggio con le informazioni sulle operazioni eseguite e l'errore che si è verificato.

Comunicazioni (3)



(a) **Lettura di dati da un file in NFS versione 3.**

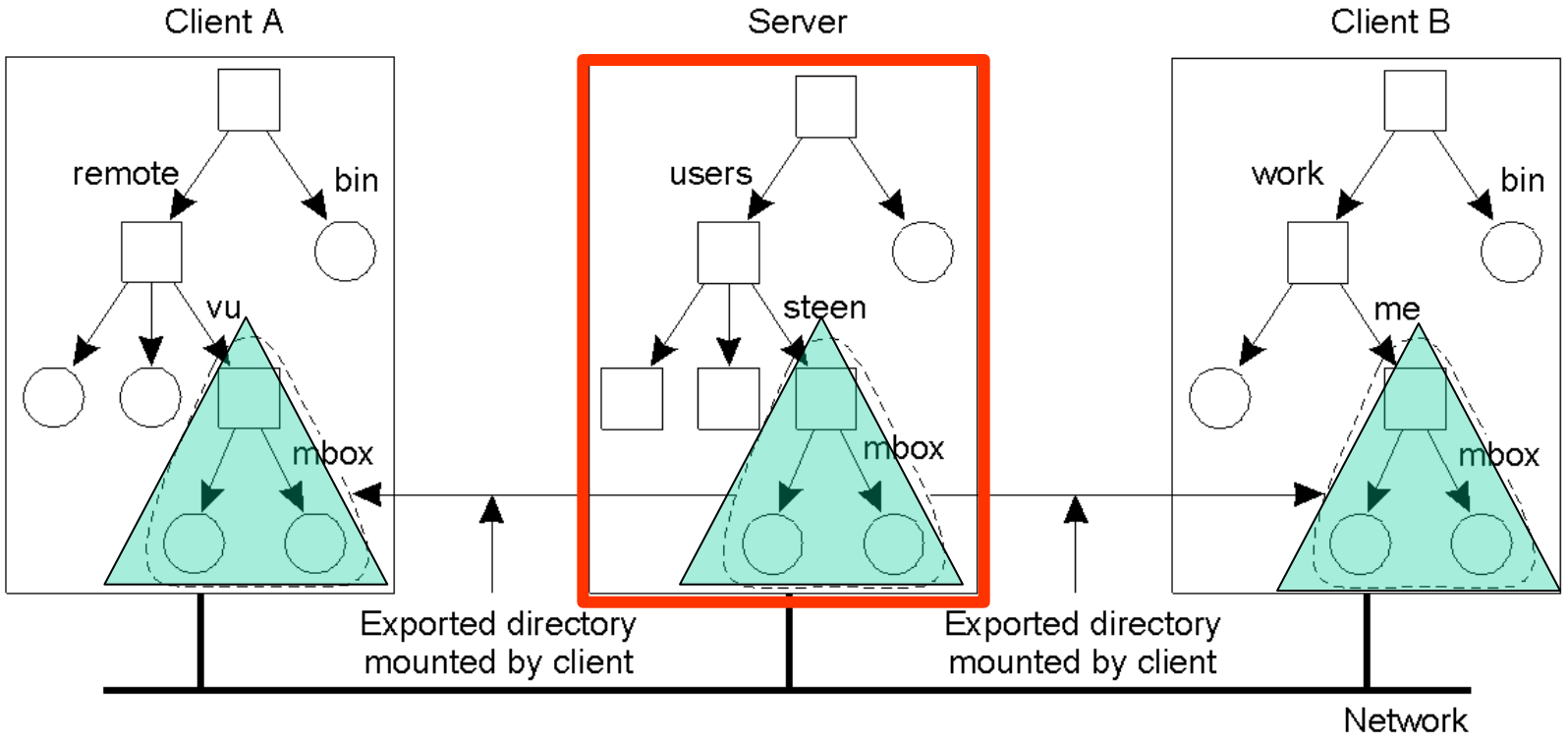
(b) **Lettura di dati da un file usando una compound procedure in NFS versione 4.**

Comunicazioni (4)

- Nella versione 4, i server NFS mantengono lo stato di alcune operazioni.
- Questo modello è stato introdotto per gestire operazioni su file systems in reti geografiche (wide-area network), come:
  - File locking,
  - Protocolli di cache consistency,
  - Callback procedures.

Naming (1)

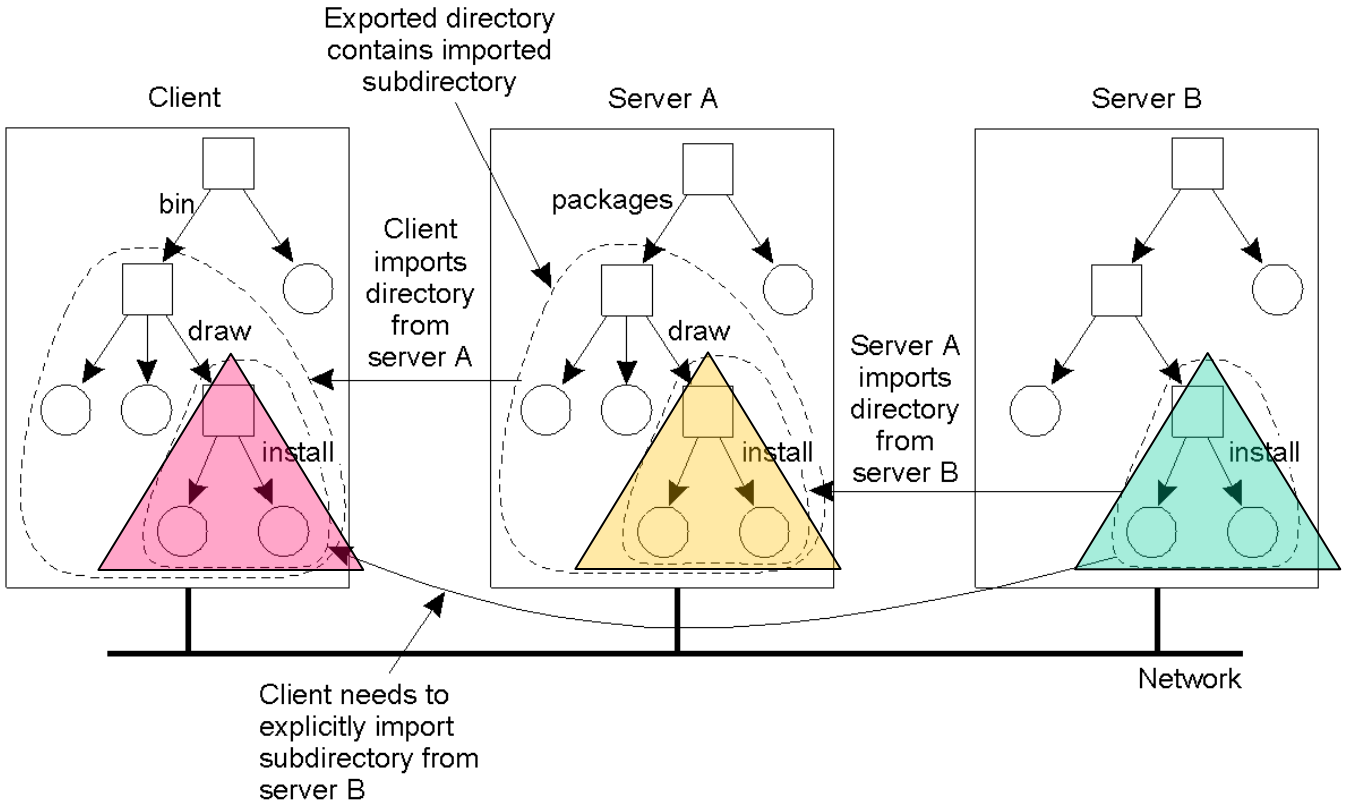
File sharing è basato su operazioni di **mounting**.



Mounting (parte di) un file system remoto in NFS.

Naming (2)

Un NFS server può montare directory esportate da altri server, ma non può esportarle ad altri clienti.



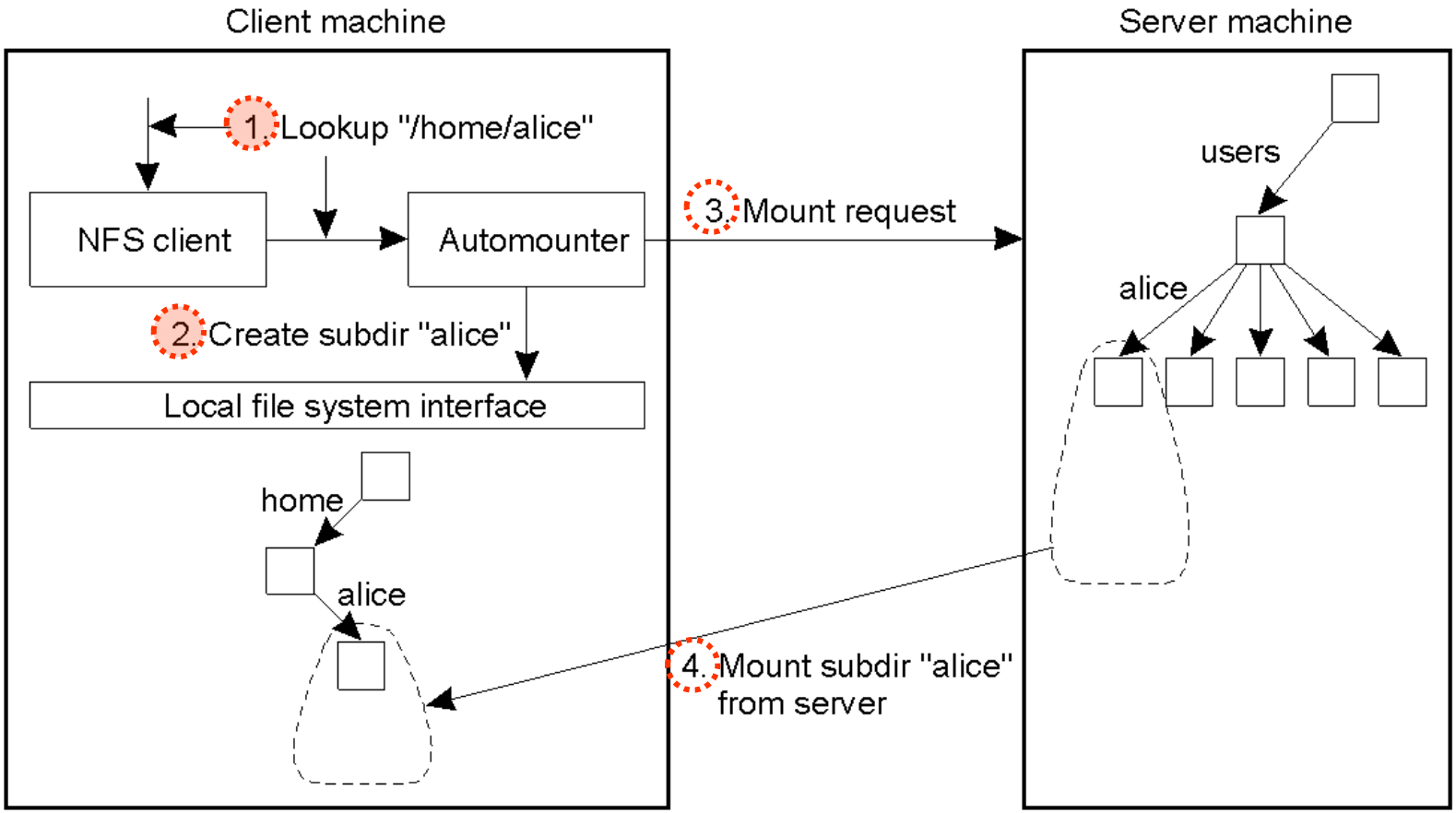
Mounting di directory innestate da più server in NFS.

Automounting (1)

- Quando un file system dovrebbe essere montato su un nodo cliente?
  - Quando un utente effettua il **login**
  - Quando viene eseguita il comando di **mount**

- Una procedura automatica è implementata da un **automounter** per NFS che
  - effettua il mount della home directory di un utente quando accede al nodo client e
  - effettua il mount di un file system on demand (quando i file sono acceduti).

Automounting (2)



Un semplice funzionamento dell' automounter per NFS.

Semantica del File Sharing (1)

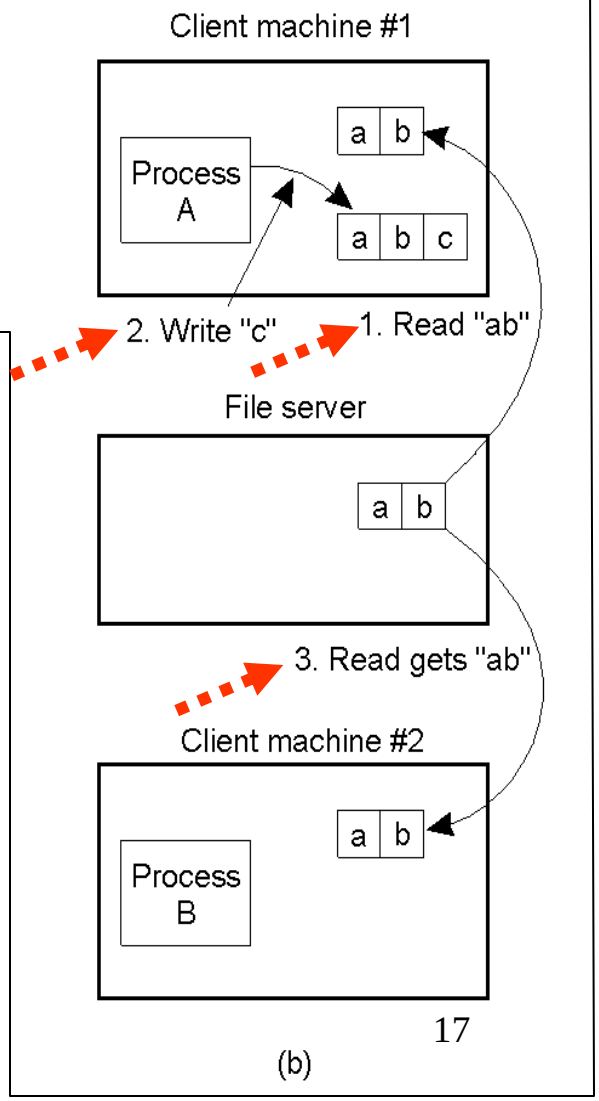
- Secondo la **semantica UNIX** in un file system sequenziale che permette di condividere files:
  - una read dopo una write, ritorna il valore scritto
  - Dopo due successive write una read ritorna il valore dell' ultima scrittura.

- In un sistema distribuito, la semantica UNIX può essere garantita solo se vi è un solo file server e i clienti non mantengono i file nella cache.

Semantica del File Sharing (2)

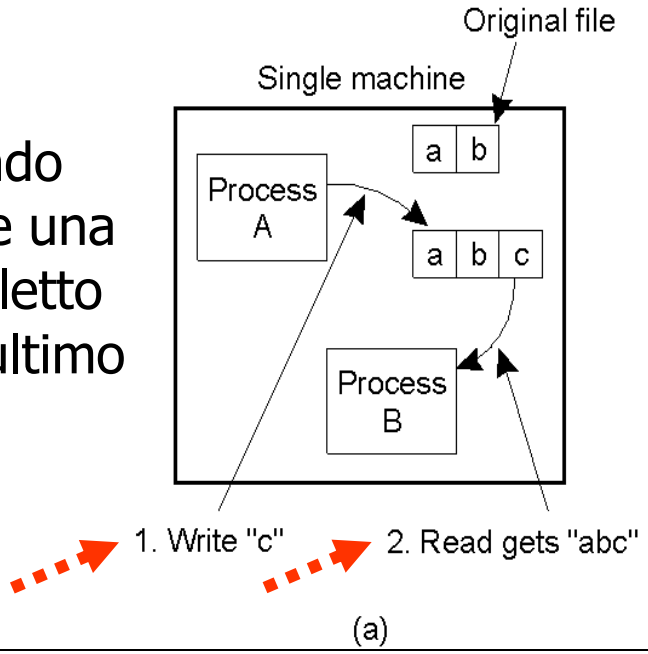
2

In un sistema distribuito con uso di cache, possono essere letti valori obsoleti.



1

Su un singolo computer quando una *read* segue una *write*, il valore letto dalla *read* è l'ultimo valore scritto.



Semantica del File Sharing (3)

- Sebbene NFS in principio usa il modello *remote access*, molte implementazioni usano cache locali, che in pratica corrisponde ad usare il modello *upload/download*.

- NSF implementa la **session semantics**:

  Le modifiche ad un file aperto sono inizialmente visibili solo al processo che ha modificato il file. Quando il file viene chiuso tutte le modifiche sono visibili agli altri processi (computer).

Semantica del File Sharing (4)

Cosa accade quando due processi memorizzano localmente un file e lo modificano?

Metodo	Commento
UNIX semantics	Ogni modifica su un file è instantaneamente visibile a tutti i processi
Session semantics	Nessuna modifica è visibile ad altri processi prima che il file venga chiuso
Immutable files	Non sono permesse modifiche; una modifica crea un nuovo file
Transaction	Tutte le modifiche sono atomiche

Quattro differenti modelli per gestire file condivisi in un sistema distribuito.

File Locking in NFS

NFS versione 4 usa uno schema di file locking.

- I Read lock non sono mutualmente esclusivi (solo più read sono permesse).
- I Write lock sono esclusivi (solo la scrittura di un processo è permessa).

NB: In NFS v3 si può usare un Network Lock Manager (NLM) per il file locking.

Operation	Description
Lock	Crea un lock (r o w) per un blocco di bytes
Lockt	Controlla se un lock in conflitto è stato acquisito
Locku	Rimuove un lock per un blocco di bytes
Renew	Rinnova un lease sull'uso del lock specificato

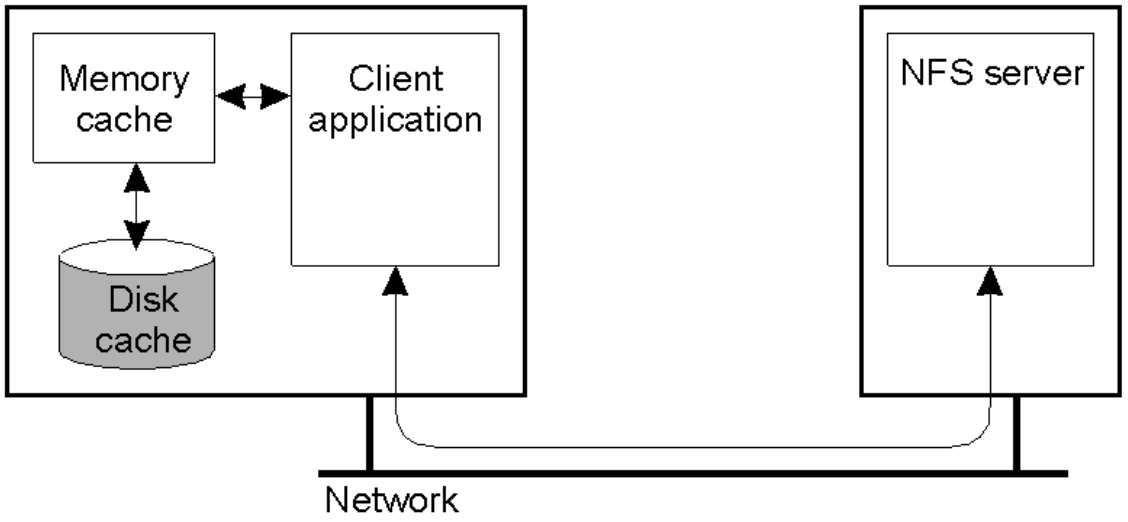
Le operazioni di NFS vers. 4 per il file locking.

NFS implementa anche una **modalità implicita** per il lock di un file detta **share reservation** che indica il tipo di accesso e il tipo di diniego

NFS Client Caching (1)

- Mentre la versione 3 non usa una cache, dalla versione 4 NFS implementa un sistema di caching sul lato client che include una Memory cache e una Disk cache.

- Per un file: i dati, gli attributi, gli handle, e le directory possono essere memorizzati nella cache.

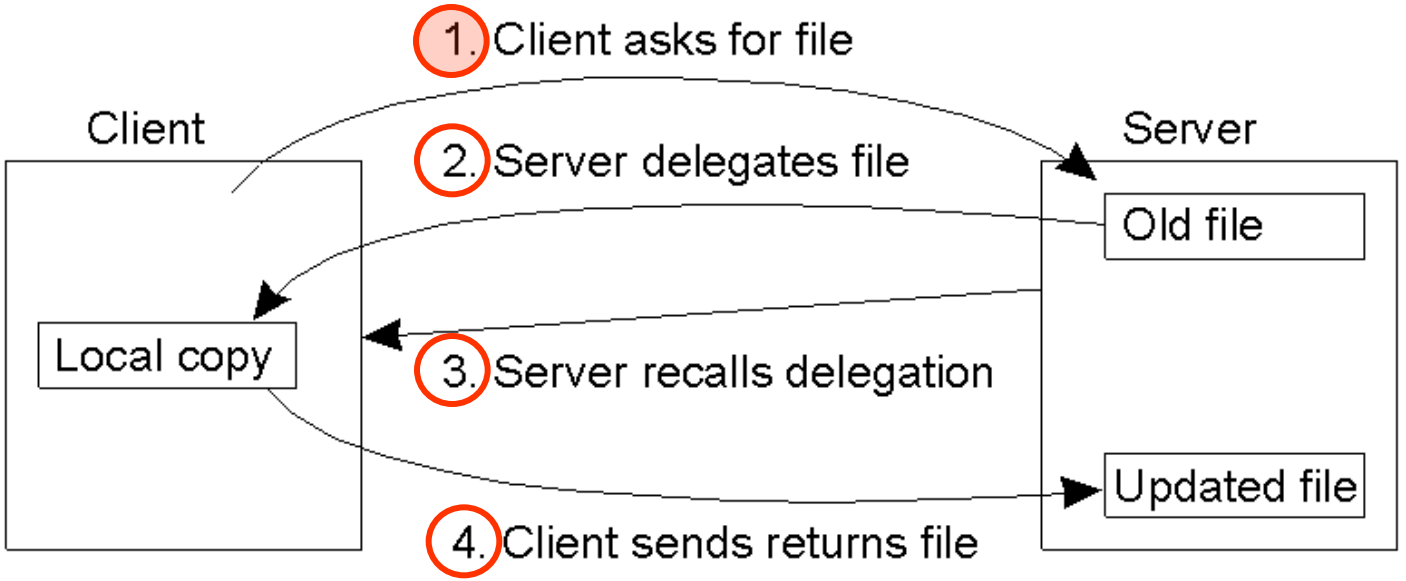


NFS Client Caching (2)

- Il sistema di caching dei dati di un file usa la *session semantics* : le modifiche dei dati nella cache vengono spostate sul server quando un client chiude il file.
- I dati possono essere mantenuti nella cache, ma se il file sarà riaperto, dovranno essere *rivalidati*.
- NFS usa anche la **open delegation** per delegare alcuni diritti al client che ha aperto il file.
- Il client può prendere decisioni relative al proprio nodo senza chiedere al server. Altre decisioni rimangono al server.

NFS Client Caching (3)

- Un server può aver bisogno di ritirare una delega quando un altro client su una macchina differente chiede i diritti di accesso per un file.
- Il meccanismo di **callback** è usato che ritirare una *file delegation*.



NFS Client Caching (4)

- Attributi, file handle, e directory possono essere memorizzati nella cache, ma le modifiche di questi valori devono essere inviate al server.

- Le informazioni nella cache sono automaticamente invalidate dopo un dato intervallo di tempo. Questo obbliga i clienti di ri-validarli prima di poterle riusare.

- NFS v4 offre un supporto per la **replicazione** di un file system tramite una **lista di locazioni** dove il file system può essere memorizzato (su diverse macchine del sistema distribuito).

NFS Fault Tolerance

- Poichè NFS v4 implementa **server stateful** (e gestisce file locking, open delegation, ecc.), sono necessari meccanismi di fault tolerance e di recovery per gestire eventuali fallimenti delle RPC.

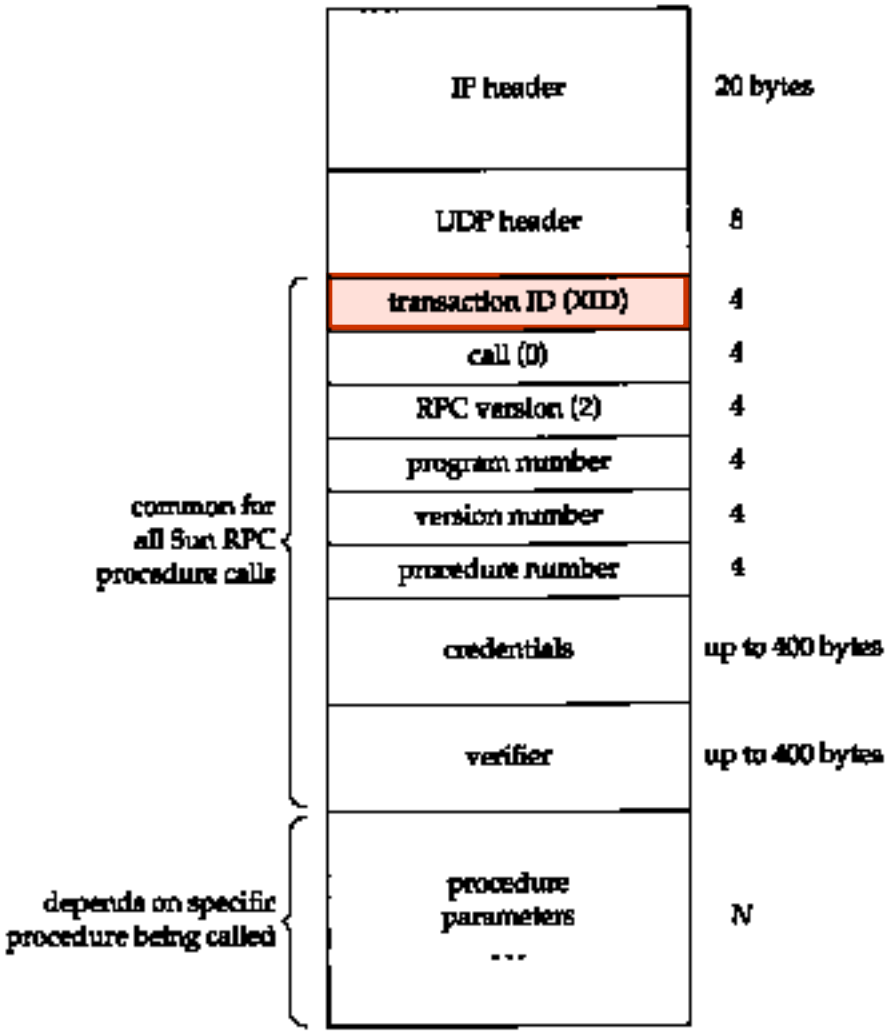
- Le RPC di NFS usano protocolli TCP e UDP che hanno diversi livelli di affidabilità.

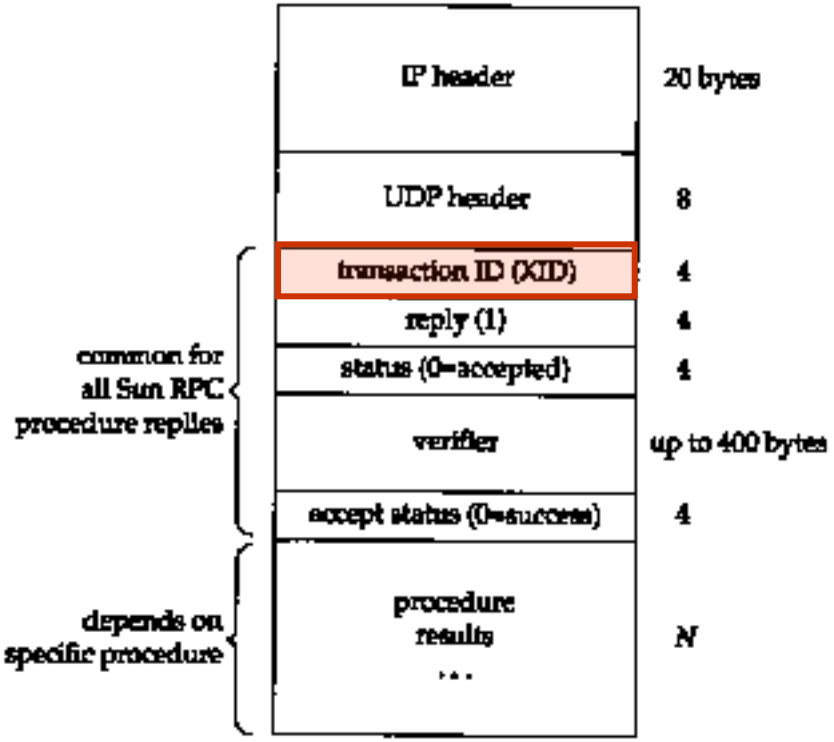
- Ad esempio: RPC può generare richieste duplicate quando si ha la perdita di una chiamata di procedura remota. Questo può portare il server ad effettuare più volte una richiesta.

- E' necessario gestire la duplicazione di chiamate.

Duplicate-Request Cache

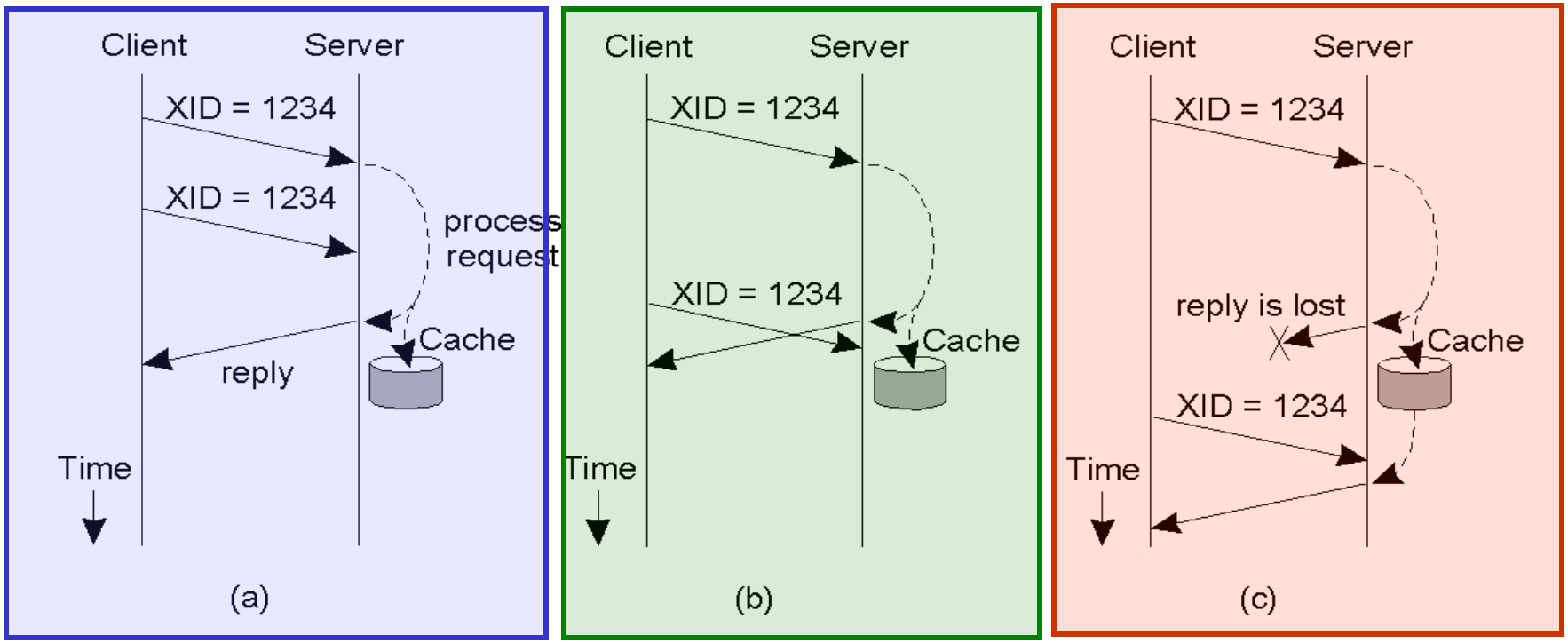
Il problema si risolve usando un *transaction id* (***XID***) nella richiesta del client e nella risposta del server.





Duplicate-Request Cache

Ogni richiesta di RPC di un client contiene un *transaction id* (***XID***) e viene memorizzata dal server insieme alla risposta.



Tre situazioni in cui è necessario gestire ritrasmissioni.

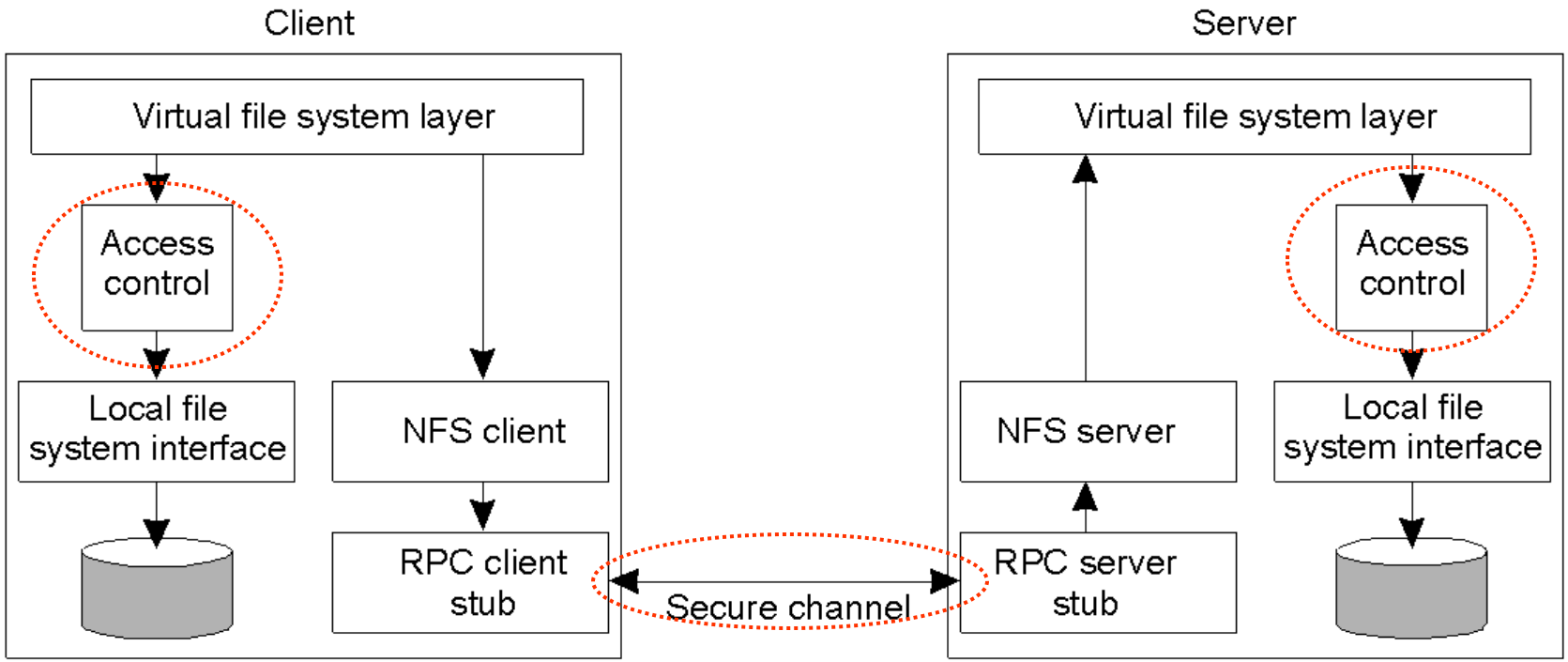
(a) La richiesta è ancora in fase di gestione.

(b) La risposta è stata da poco inviata.

(c) La risposta era stata inviata da tempo ed era stata

Sicurezza in NFS

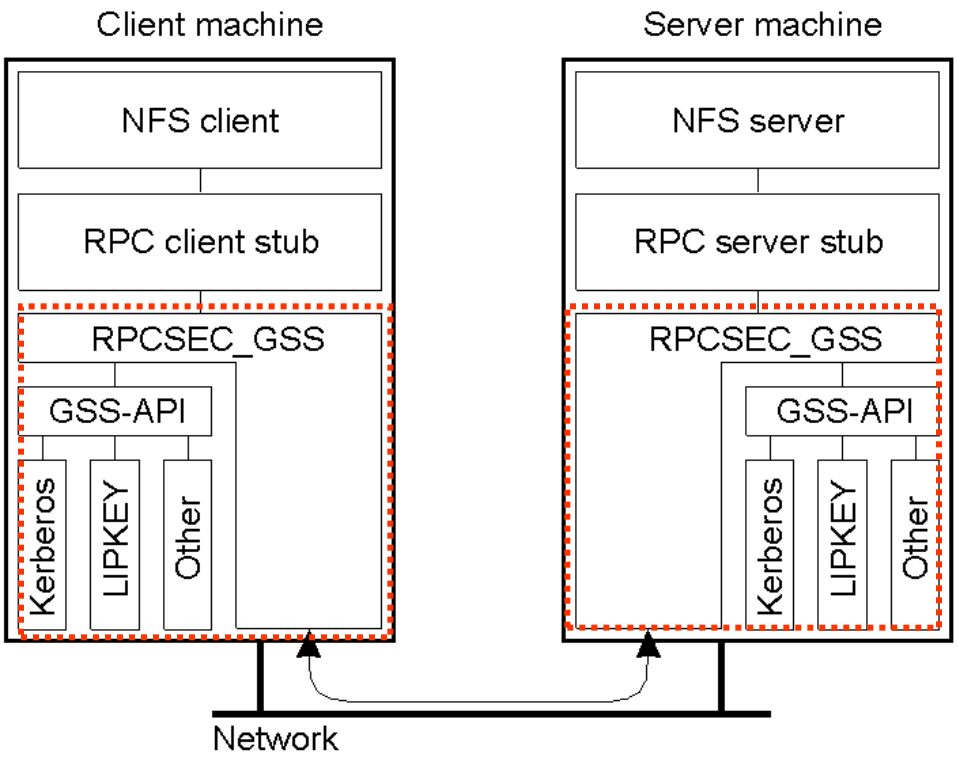
La sicurezza in NFS è basata su canali di comunicazione sicuri (**secure channels**) e meccanismi di controllo degli accessi (**file access control**).



La NFS security architecture.

RPC Sicuro

- RPC Sicuro in NFS v4 è basato su RPCSEC_GSS.
- RPCSEC_GSS - Generic Security Service authentication protocol per ONC RPC basato su Generic Security Services API (GSS API)



Controllo degli Accessi

Valori degli attributi ACL

Operazione	Descrizione
Read_data	Permission to read the data contained in a file
Write_data	Permission to to modify a file's data
Append_data	Permission to to append data to a file
Execute	Permission to to execute a file
List_directory	Permission to to list the contents of a directory
Add_file	Permission to to add a new file t5o a directory
Add_subdirectory	Permission to to create a subdirectory to a directory
Delete	Permission to to delete a file
Delete_child	Permission to to delete a file or directory within a directory
Read_acl	Permission to to read the ACL
Write_acl	Permission to to write the ACL
Read_attributes	The ability to read the other basic attributes of a file
Write_attributes	Permission to to change the other basic attributes of a file
Read_named_attrs	Permission to to read the named attributes of a file
Write_named_attrs	Permission to to write the named attributes of a file
Write_owner	Permission to to change the owner
Synchronize	Permission to to access a file locally at the server with synchronous reads and writes

La classificazione di operazioni riconosciute da NFS rispetto al controllo degli accessi.

Tipi di Utenti NFS

Tipo di utente	Descrizione
Owner	Proprietario del file
Group	Gruppo degli utenti associato al file
Everyone	Qualsiasi utente di un processo
Interactive	Qualsiasi processo che accede il file in modalità interattiva
Network	Qualsiasi processo che accede il file dalla rete
Dialup	Qualsiasi processo che accede il file tramite una connessione lenta
Batch	Qualsiasi processo che accede il file come parte di un job batch
Anonymous	Qualsiasi utente che accede il file senza autenticazione
Authenticated	Qualsiasi utente autenticato
Service	Qualsiasi processo di servizio del sistema che accede il file

Differenti tipi di utenti e processi distinti da NFS rispetto al controllo degli accessi.