



La Sapienza

Università degli Studi di Roma

Dipartimento Info-Com

Laboratorio Telematico @ Info-Com

Alessandro Falaschi

OpenCDN: una architettura espandibile per il Live Streaming

Bologna, 26 Febbraio 2004

Primo incontro Gruppo Netcast @ CNAF

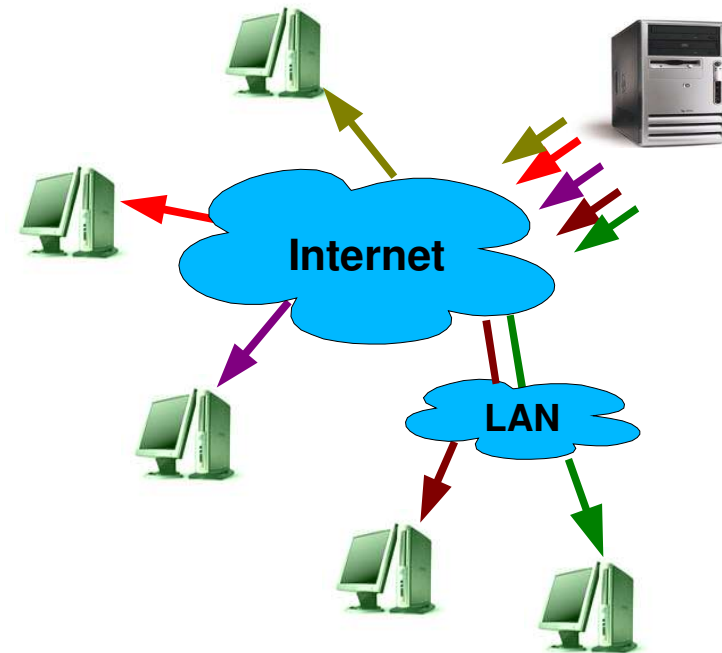
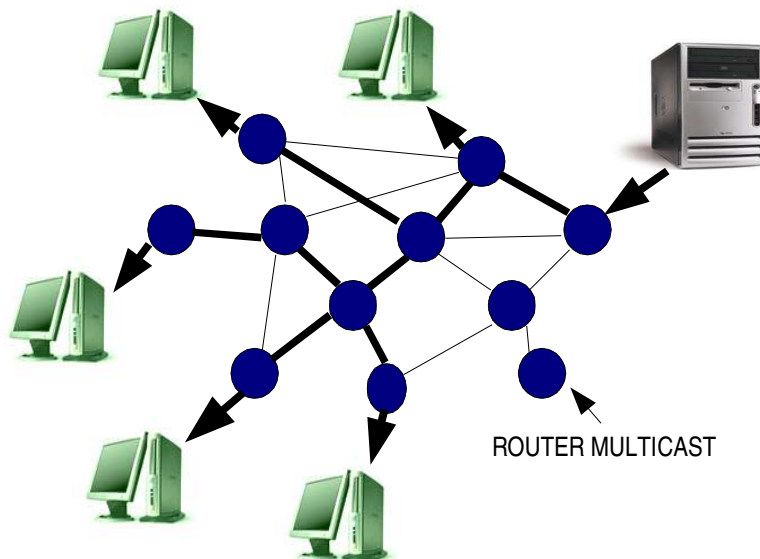
Cosa offre di preciso ?

- Gerarchica** -> Scalabile
 - Multipiattaforma** -> Diverse tecnologie coesistenti
 - Distribuita** -> Esecuzione delegata ai Nodi
 - Adattiva** -> Bilanciamento del carico
 -> Reattività alle condizioni di rete
 - Espandibile** -> Libertà di aggiungere nuovi nodi
 -> Capacità ad integrare nuove Piattaforme
 - Multipla** -> Un Nodo può ospitare tecnologie diverse
 - OpenSource** -> Aperta al contributo di tutti
-

Scalabilità del Live Streaming

■ Architettura Client-Server

- Un elevato numero di Client accede al medesimo contenuto
- Si incorre presto nell'esaurimento della CPU e della banda in uscita, od anche di quella di ultimo miglio

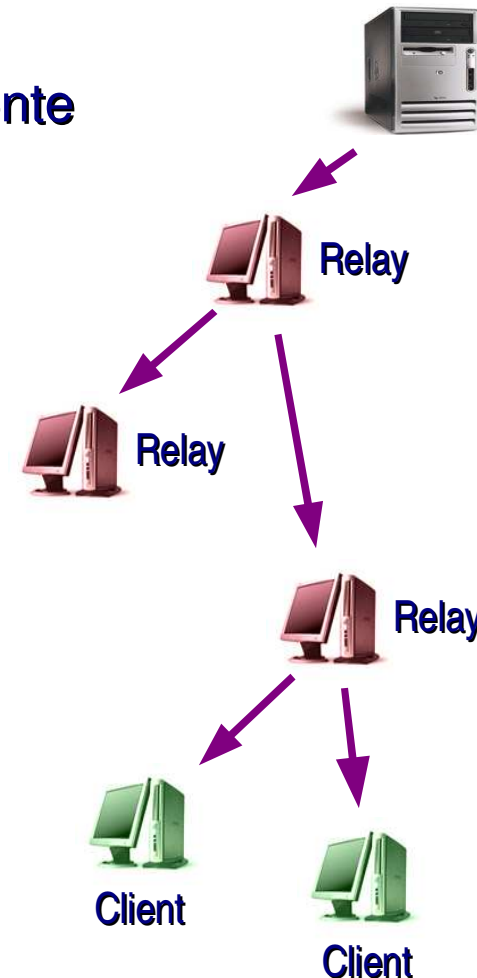


■ Soluzione Multicast

- Non è detto che i router periferici lo supportino e/o lo abilitino
- Gli ISP non fanno peering multicast
- Sussiste il problema di assegnazione degli indirizzi, che ostacola il peering

Application Level Multicast

- Si adottano agenti **bifronte**, contemporaneamente Client e Server, con funzioni di
 - Relay (ponte) ritrasmettendo quanto ricevono
 - Splitting (suddivisione) replicando il flusso multimediale per tutti i client a valle
- Una scelta appropriata dell'albero di distribuzione tra Relay ottimizza l'impegno delle risorse di rete ed il carico della sorgente
- Problemino: (-ONE) non è pensabile stabilire una distribuzione ad hoc per ogni nuovo evento !



Content Delivery Network

- Evoluzione delle Web-Cache, attuate mediante gerarchia di Proxy
 - Ora il client è IGNARO del proxy
- Valore Aggiunto: SCALE and REACH
 - SCALE: possono essere raggiunti molti più clients
 - REACH: i client prelevano i contenuti a partire da surrogati loro vicini
- Componenti del servizio CDN:
 - Delivery: la consegna dei contenuti tra Relay e verso i Clients, la tecnologia del Relay
 - Distribution: lo sviluppo della topologia dei Relay
 - Request Routing: come la richiesta di un client è instradata verso il surrogato migliore
 - Accounting: come tariffare il servizio, raccogliere i Logs

Proxy
▼
Surrogato

Necessità di una Infrastruttura Comune

- **Le diverse tecnologie di streaming (WM, QT, Real) sono spesso incompatibili dal punto di vista**
 - **dei codec (WM, MPEG, QT, REAL)**
 - **del controllo (RTSP, MMS)**
 - **del trasporto (RTP, RDT)**
- **Esiste il rischio di incorporare la tecnologia nell'architettura, favorendo la diffusione della tecnologia dominante, a discapito di tutte le altre**

Open Content Delivery Network – OpenCDN

- Realizza il piano di controllo della **DISTRIBUZIONE** in modo indipendente dalla tecnologia di Streaming
- Definita nel contesto della Task Force **NETCAST** di **TERENA**, per lo sviluppo di un canale di Live Streaming Accademico
- Si basa sulla esistenza di tre entità:
 - **PORTALE DI ANNUNCIO**
 - **RRDM** – controllo del Request Routing & Distribution Management
 - **NODO** che a sua volta può essere un
 - **LAST HOP RELAY** se serve direttamente i clients
 - **TRANSIT RELAY** se serve altri Nodi



Architettura del Nodo di OpenCDN

- OpenCDN *INGLOBA* un dispositivo (Darwin-QT, Real o WM) di Streaming in un piano di controllo che ne permette la configurazione remota
 - Il requisito per la tecnologia annidata, è che questa sia capace di realizzare un Relay PULL rispetto ad altri nodi e sorgenti compatibili
 - Uno STRATO DI ADATTAMENTO interagisce con il dispositivo, ed offre una API indipendente dalla tecnologia
 - Uno STRATO DI CONTROLLO interagisce con RRDM e con gli altri nodi, mediante messaggi XML-RPC
 - I parametri delle chiamate remote sono espressi in XML, trasportati come payload di messaggi HTTP
 - Esistono librerie e moduli che realizzano la serializzazione XML per tutti i più diffusi linguaggi
-

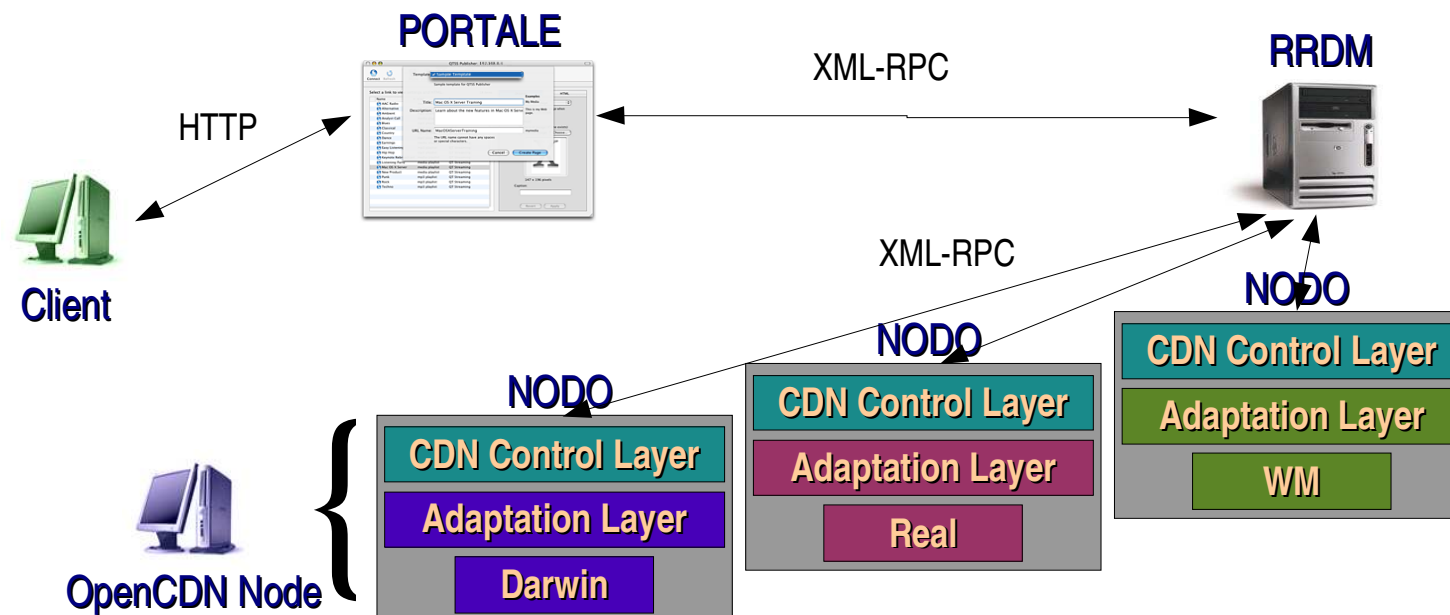
Architettura del Nodo di OpenCDN – 2

La **SEPARAZIONE** nei due strati funzionali

- **CDN Control Layer**
- **Adaptation Layer**

e la **REALIZZAZIONE** di un diverso Adaptation Layer per i diversi dispositivi

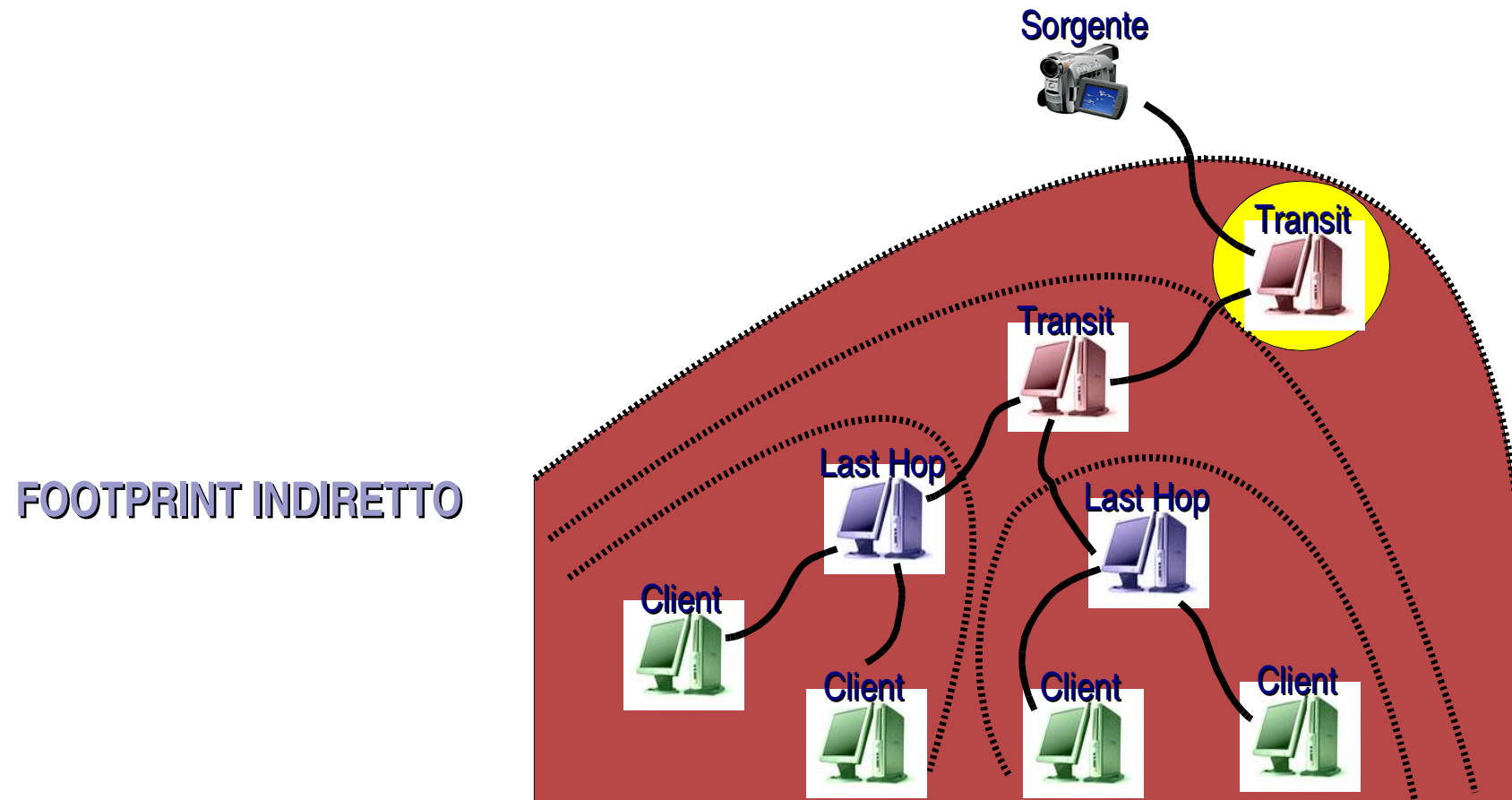
*Consentono l'uso
congiunto di differenti
tecnologie di Streaming*



Sviluppo della distribuzione – 1

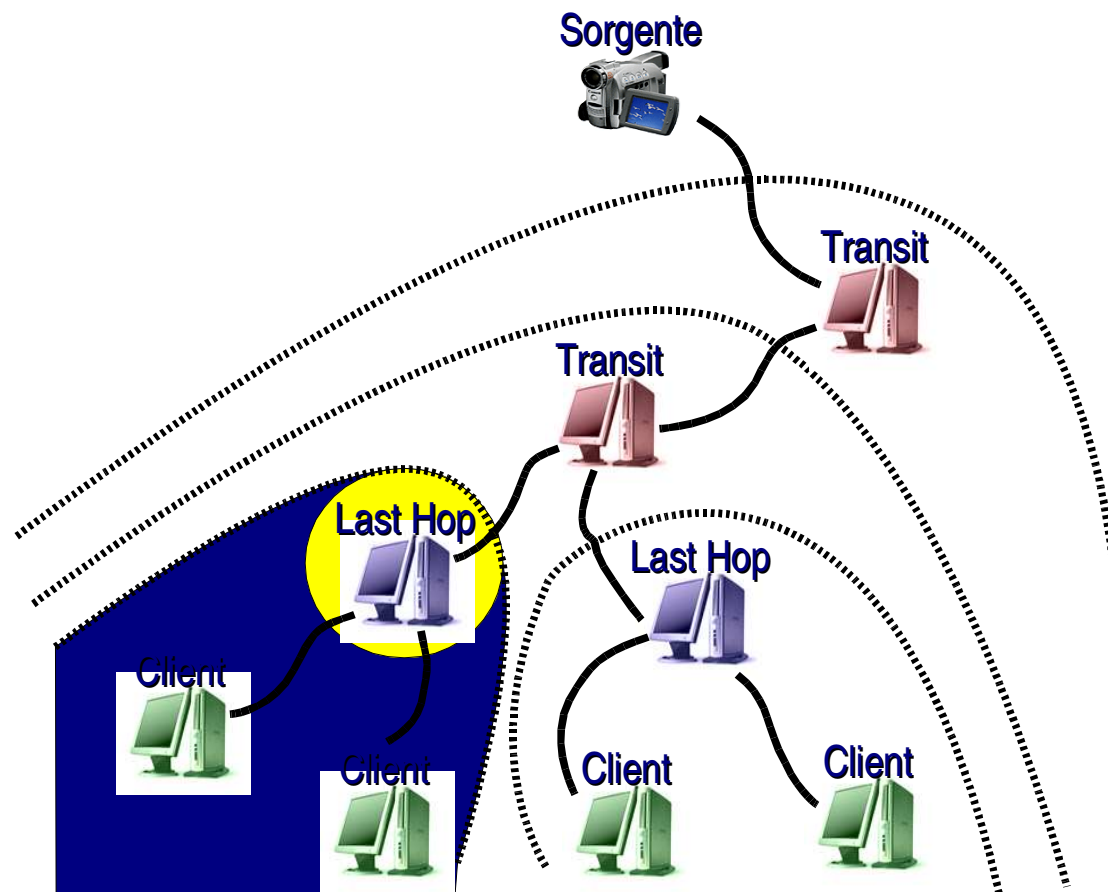
- Ogni nodo annuncia uno o più FOOTPRINT, che individuano l'insieme dei nodi che si intendono servire
 - Un Footprint è rappresentato in alternativa da:
 - Un prefisso di rete IP (simile a una riga di routing table)
 - Un suffisso di Nome di Dominio
 - Un Footprint può essere:
 - Diretto --> il nodo agisce come LAST HOP RELAY
 - Indiretto --> il nodo agisce come TRANSIT RELAY
 - I Footprint sono annidati, permettendo di individuare la topologia della distribuzione
-

Footprints – 1



Footprints – 2

FOOTPRINT DIRETTO



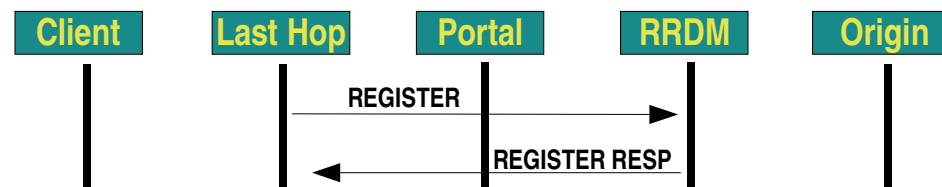
Fasi Operative - 1

- **Quando un Nodo si attiva, comunica le sue Footprint all'RRDM**
 - **Quando un client fa richiesta di un contenuto presso il Portale, questo invoca il servizio di CDN all'RRDM**
 - **L'RRDM conosce il Footprint di tutti i Nodi, e può determinare il più idoneo a servire un determinato client, e quali nodi di transito coinvolgere**
 - **Al termine della configurazione dei Nodi interessati, l'RRDM comunica l'esito al Portale, e questo al Client**
 - **Se un nuovo Client giace in una Footprint già servita, l'RRDM non esegue le fasi di configurazione remota**
-

Fasi Operative – 2

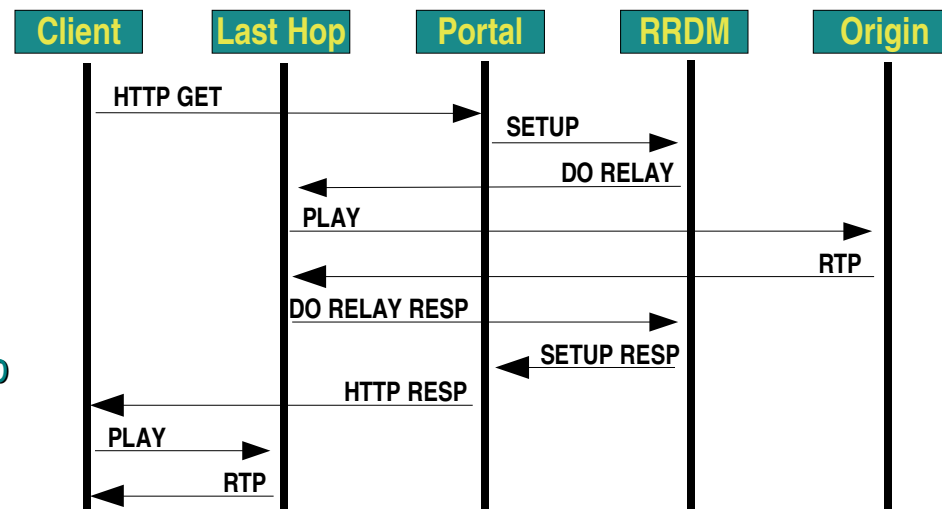
REGISTRAZIONE

- I **Nodi** contattano l'**RRDM** dichiarando:
FORMATO/PROTOCOLLO gestiti
FOOTPRINTS serviti



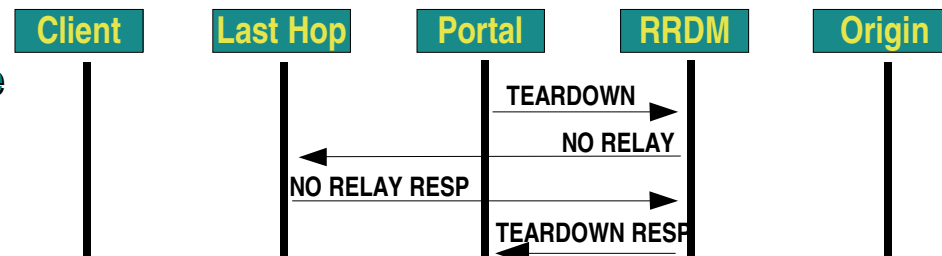
SETUP

- Il Portale invia una richiesta di servizio all'**RRDM**
- L'**RRDM** provvede alla **SODDISFAZIONE DELLA RICHIESTA**
- L'**RRDM** comunica al Portale l'indirizzo del Last Hop

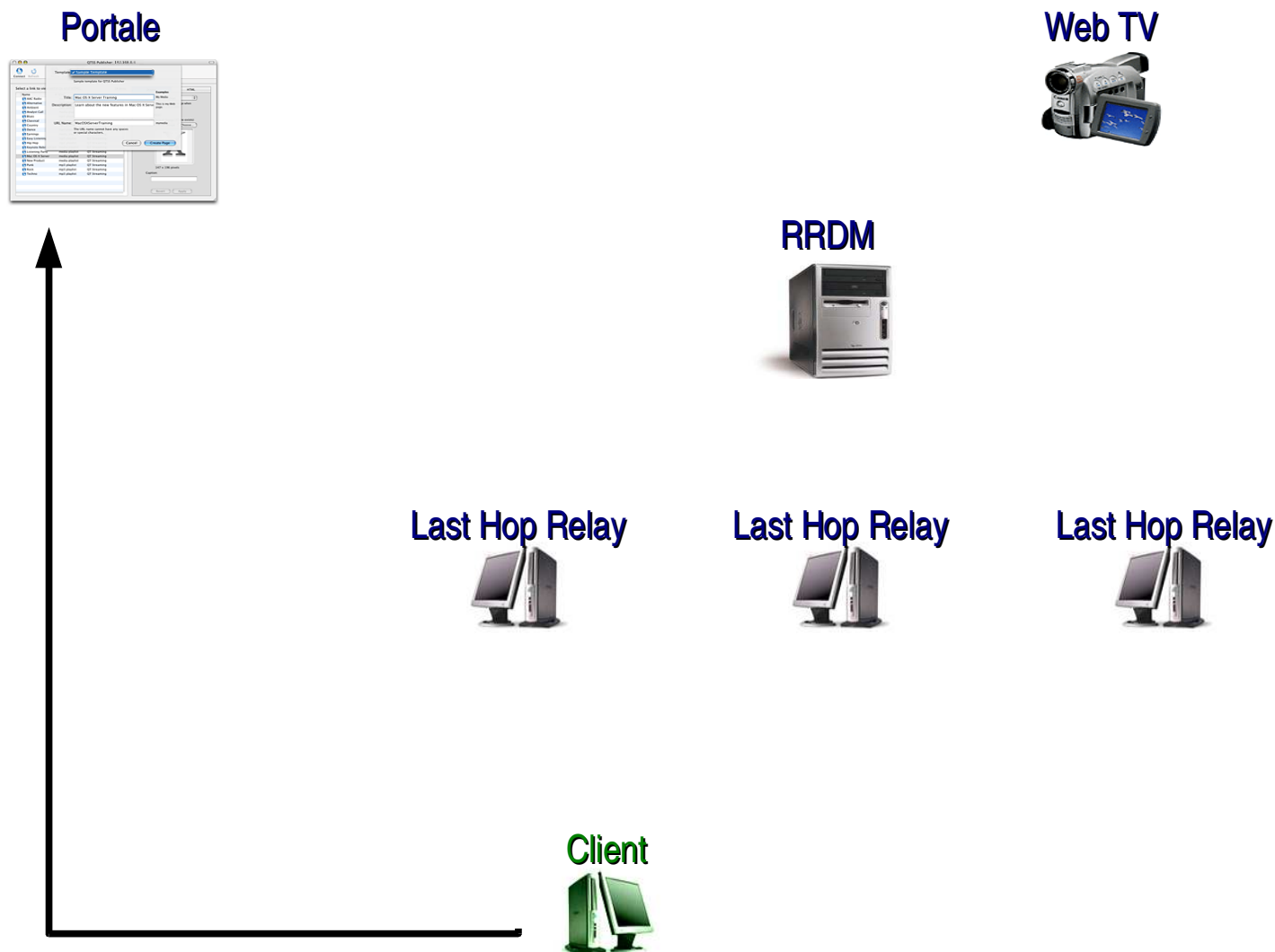


TEARDOWN

- Al termine dell'evento trasmesso, il Portale richiede all'**RRDM** di **LIBERARE LE RISORSE**

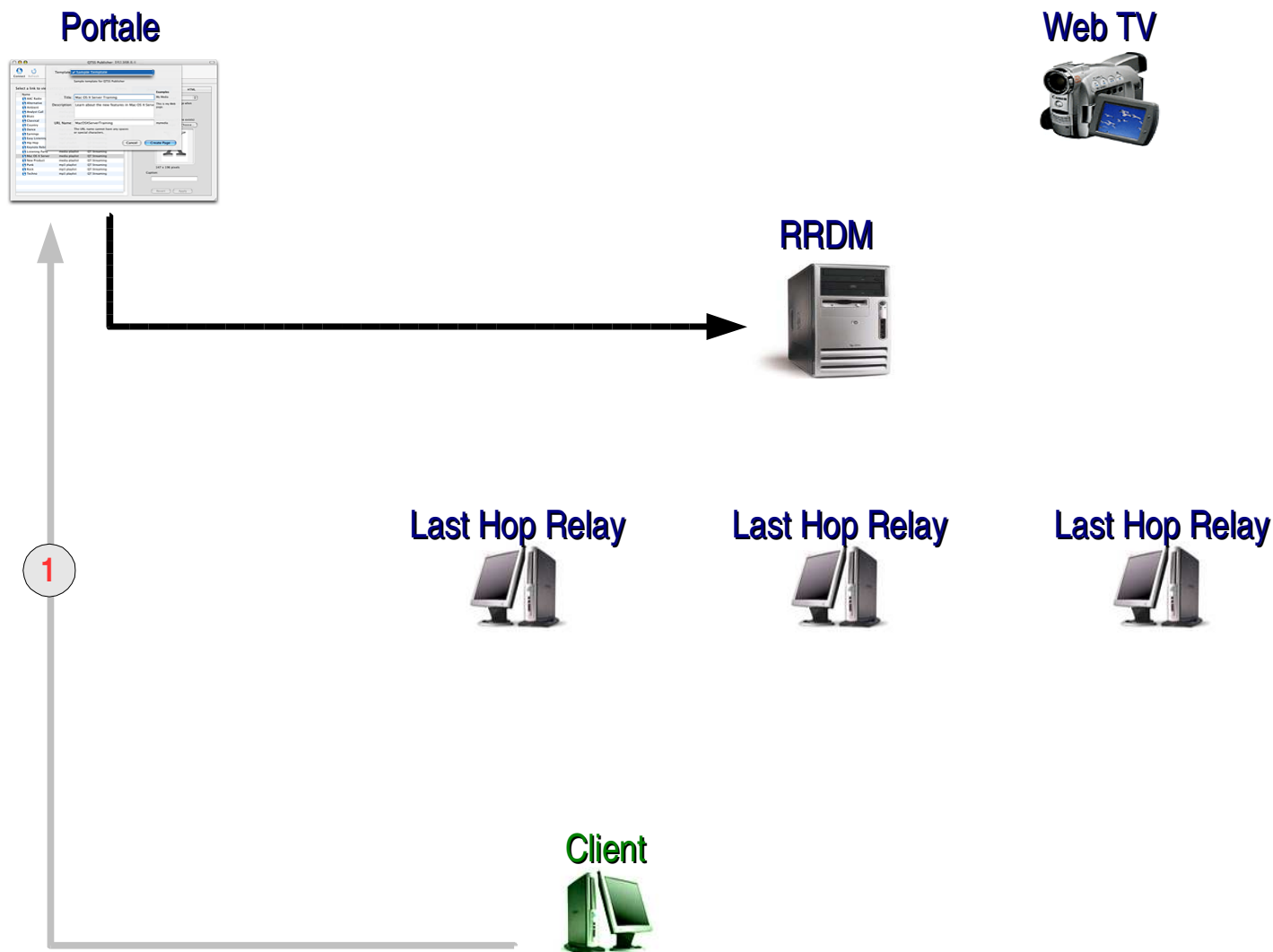


Richiesta di Servizio – 1



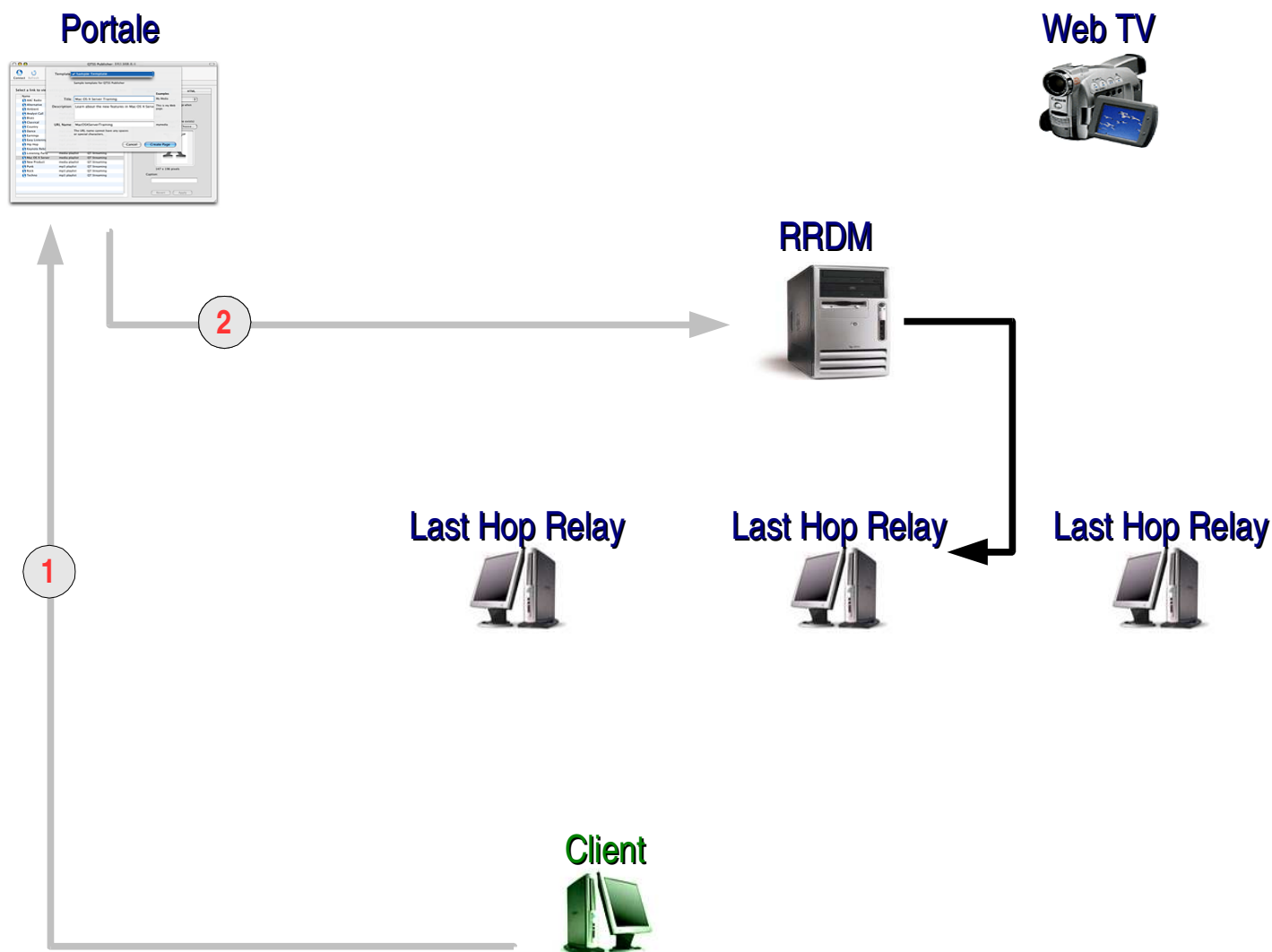
Il Client visita il Portale e richiede lo streaming della web tv

Richiesta di Servizio – 2



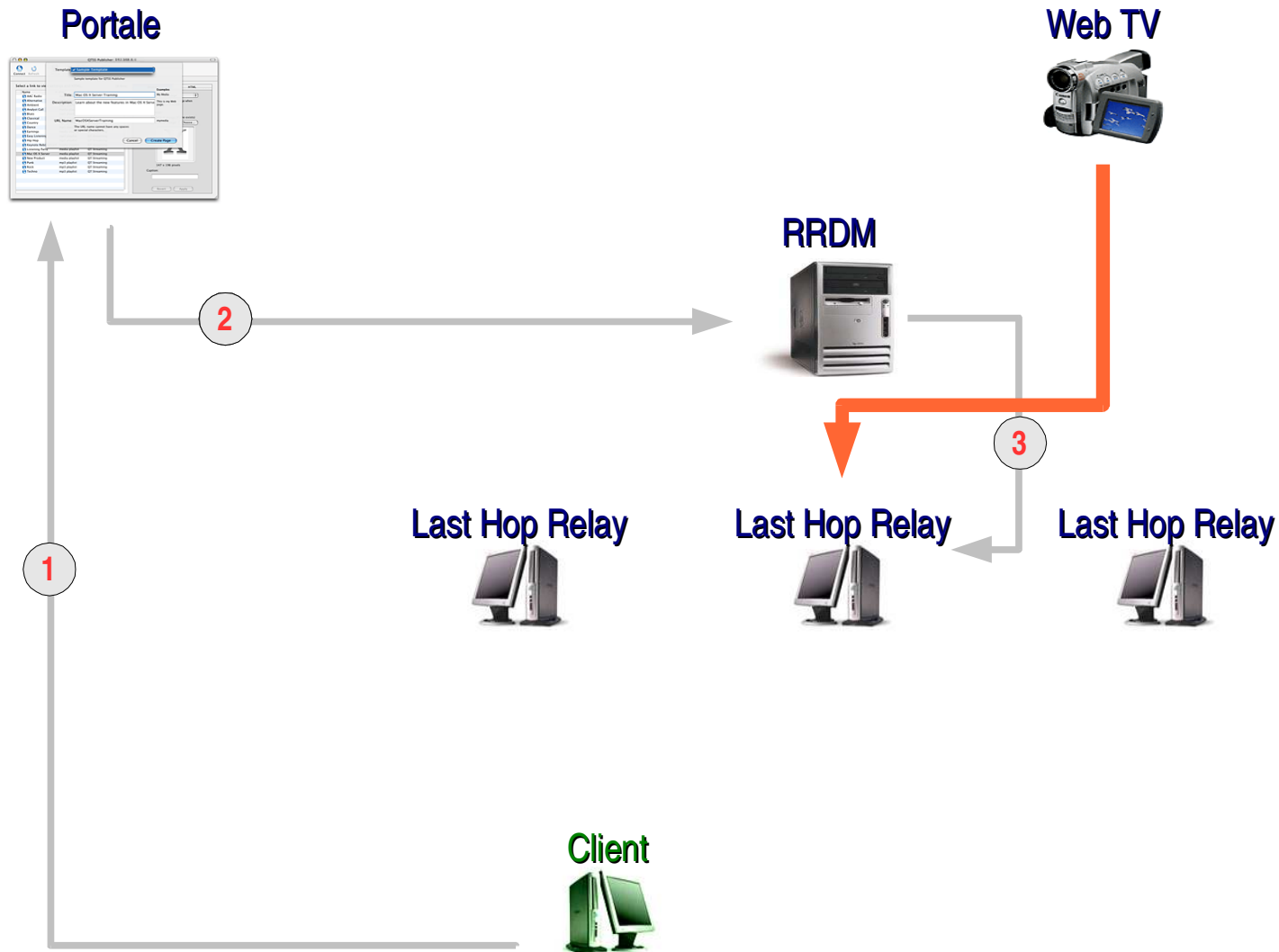
Il Portale inoltra la richiesta all'RRDM

Richiesta di Servizio – 3



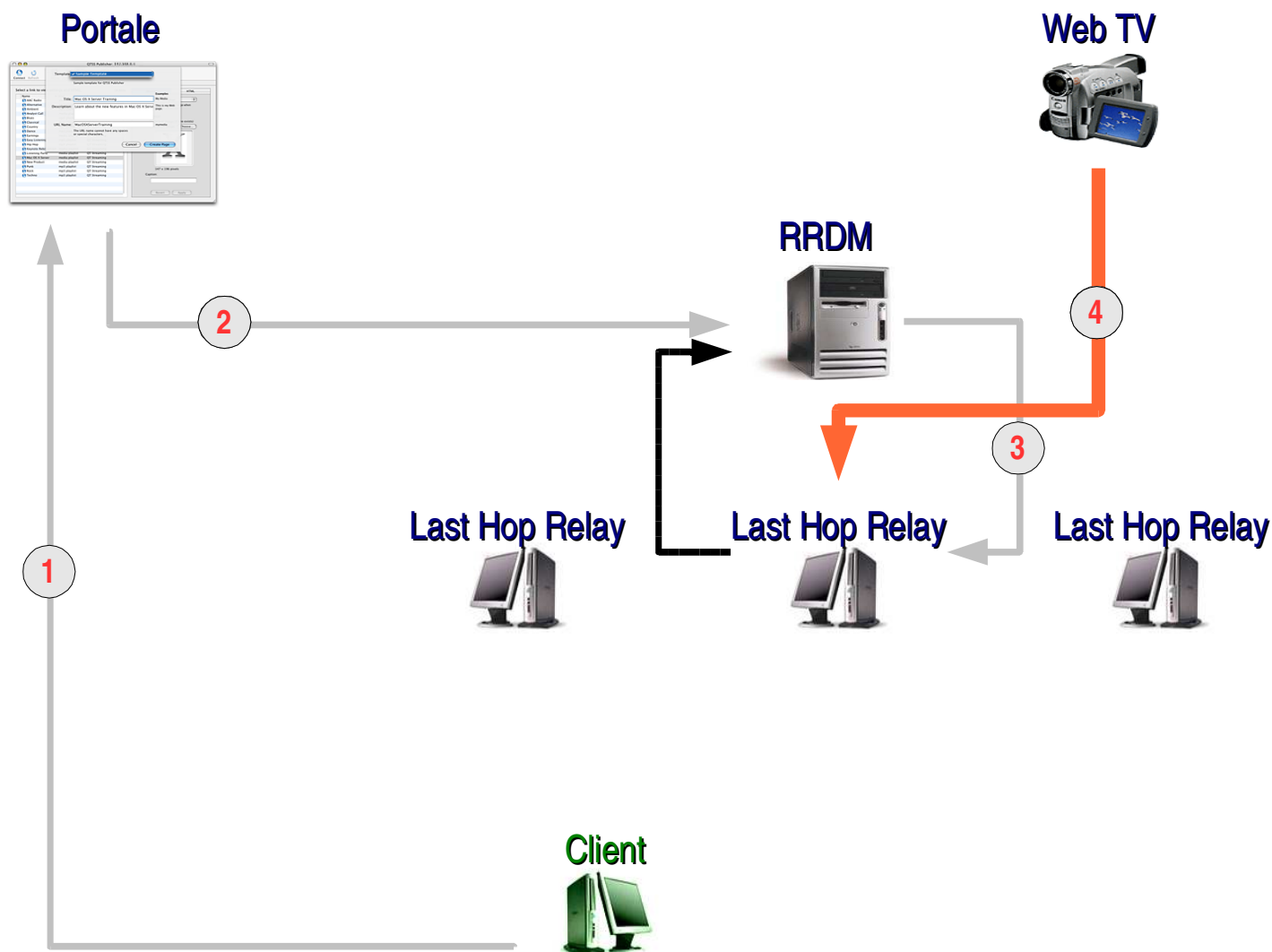
L'RRDM contatta il Last Hop pertinente indicandogli la risorsa da ricevere

Richiesta di Servizio – 4

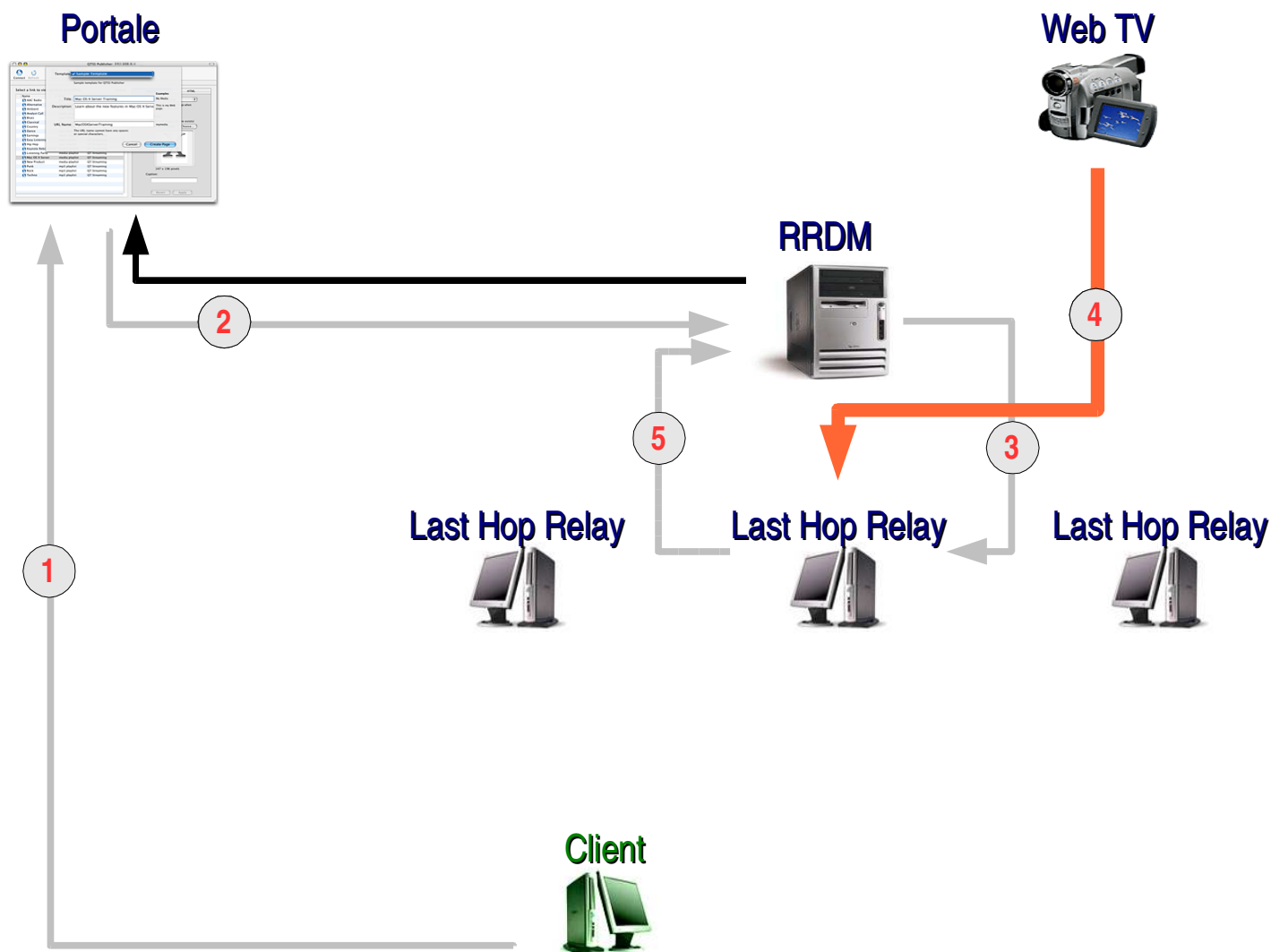


Lo streaming raggiunge il surrogato Last Hop

Richiesta di Servizio – 5

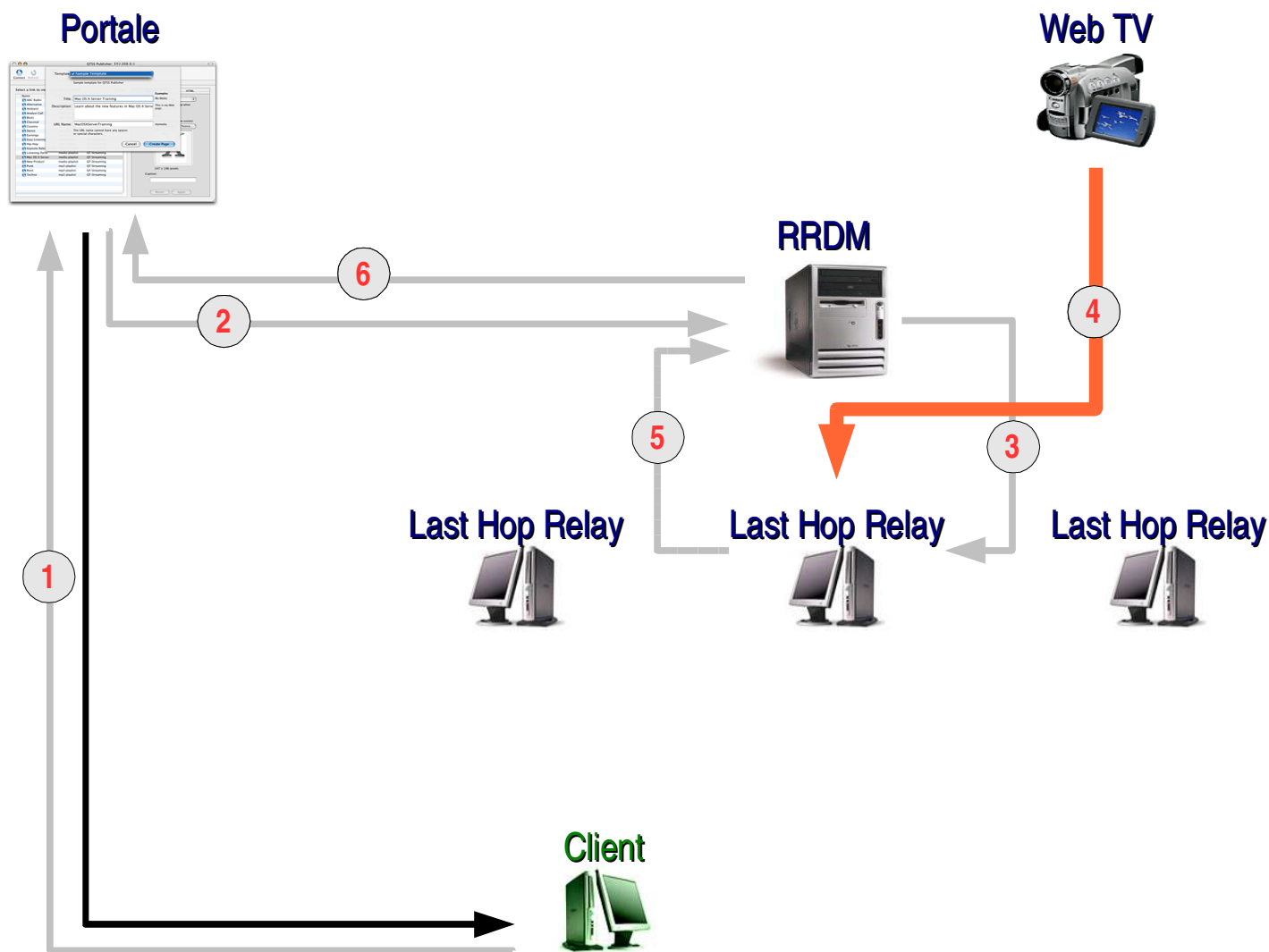


Richiesta di Servizio – 6



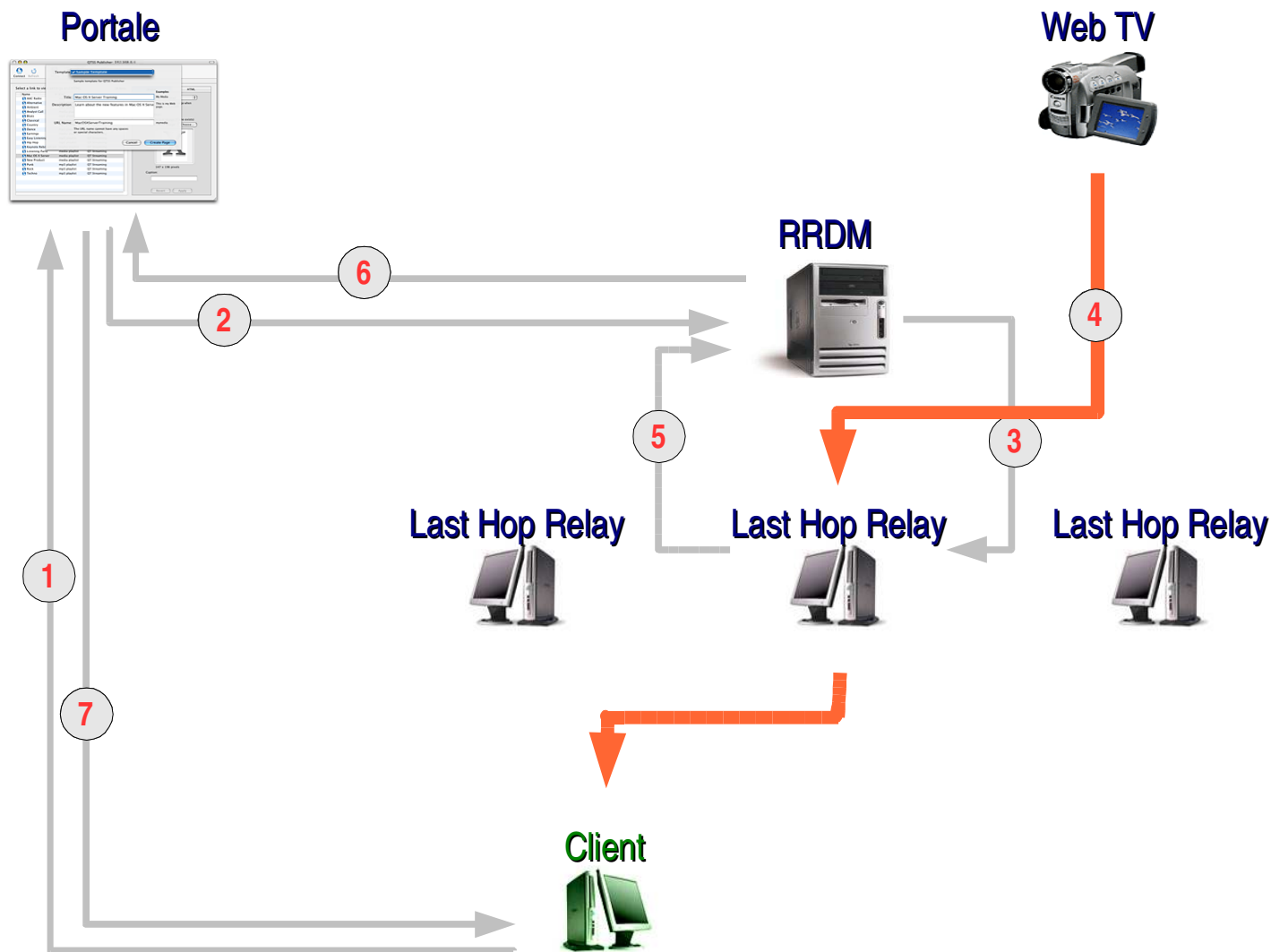
L'RRDM inoltra la risposta al Portale

Richiesta di Servizio – 7



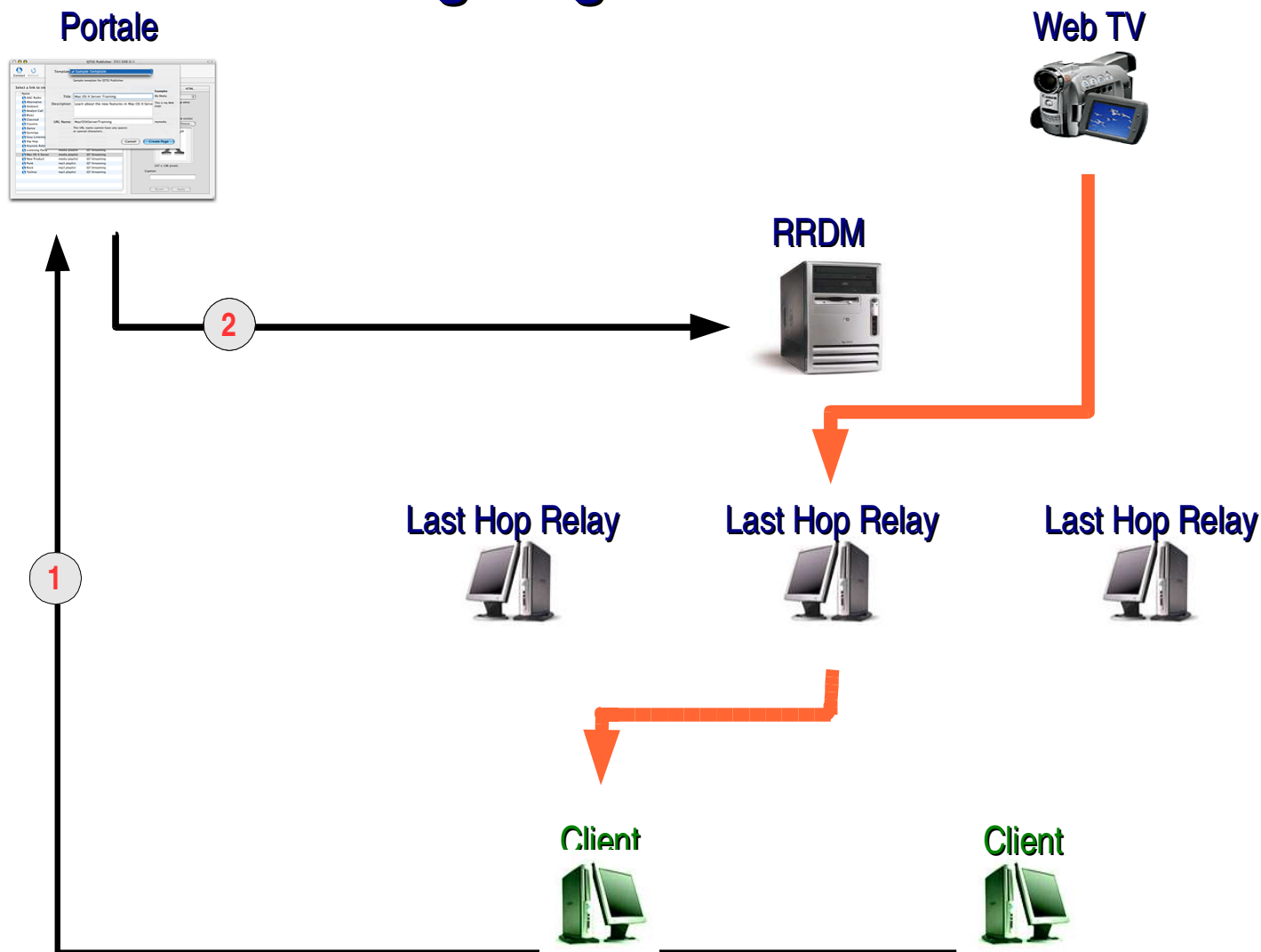
Il Portale comunica al Client l'indirizzo della risorsa presso il Surrogato

Richiesta di Servizio – 8



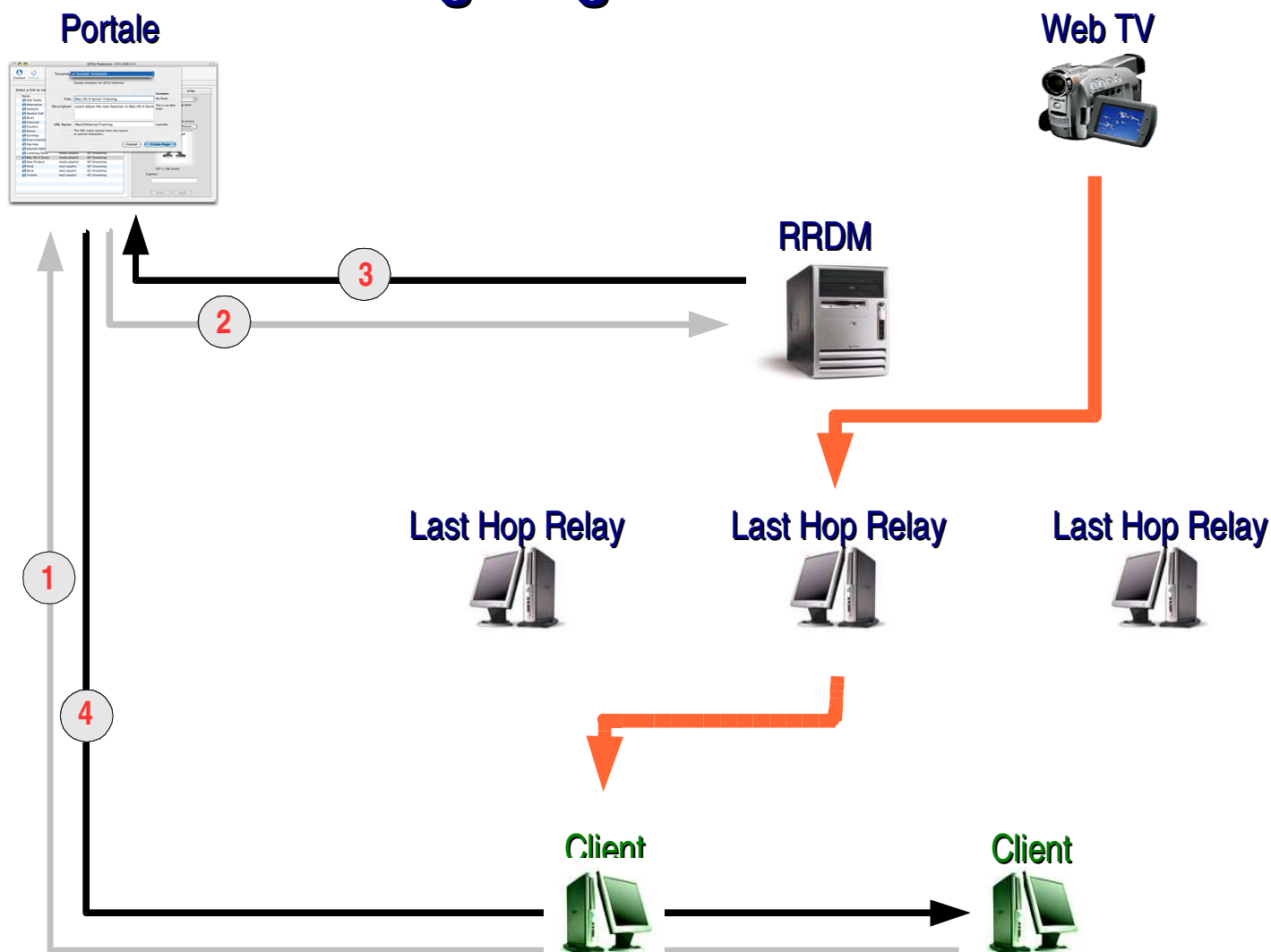
Lo streaming raggiunge il Client

Surrogato già attivo – 1



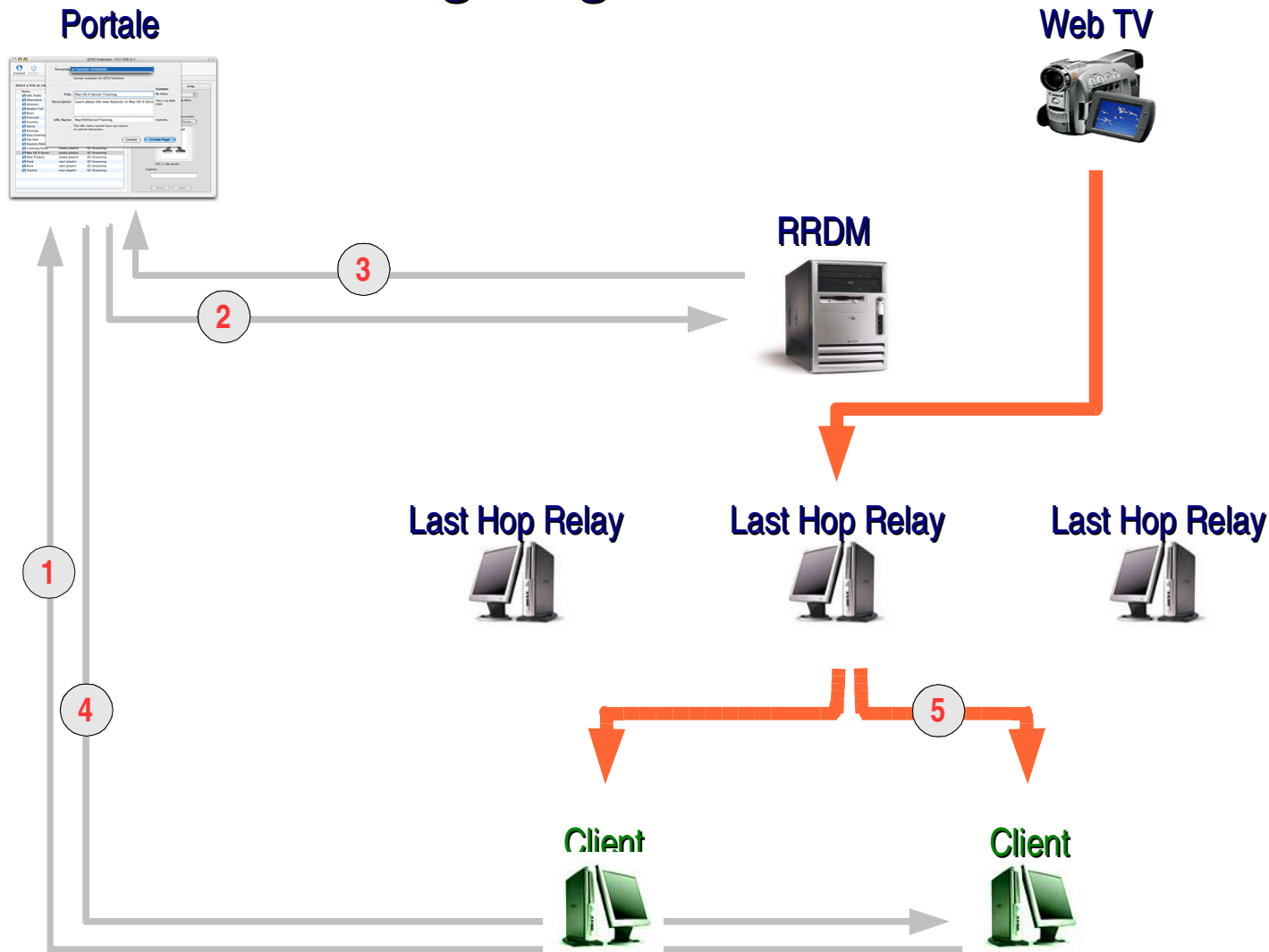
Il Portale inoltra la richiesta del Client all'RRDM

Surrogato già attivo – 2



L'RRDM risponde con l'indirizzo del Surrogato già attivo

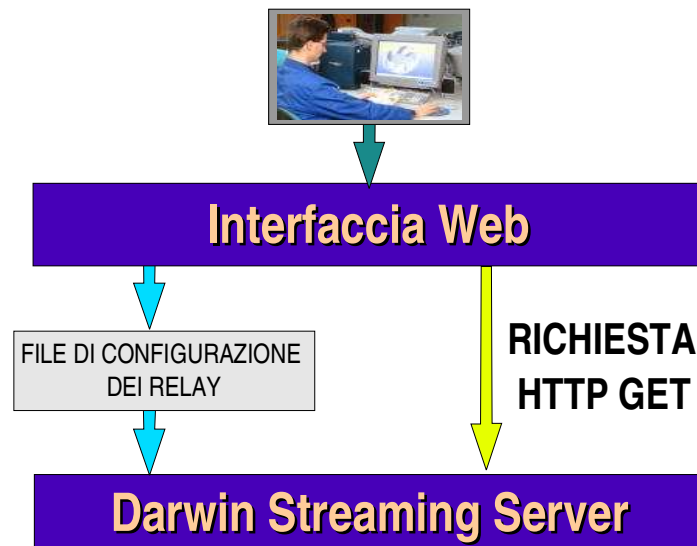
Surrogato già attivo – 3



Lo streaming raggiunge il secondo Client

Darwin Streaming Server

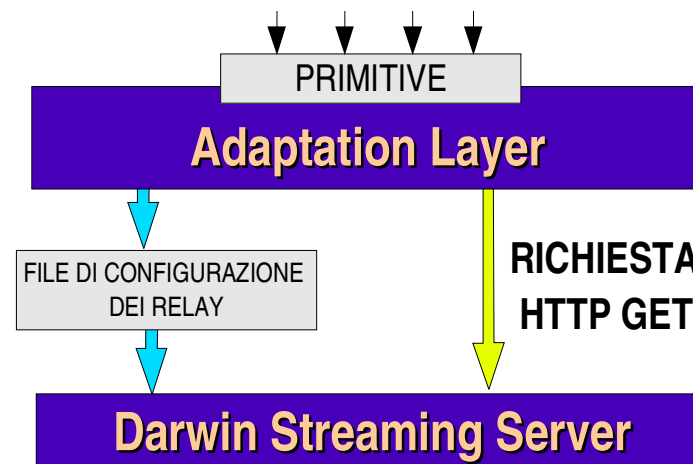
- Si utilizza Apple Darwin Streaming Server (DSS)
 - Streaming in formato MPEG4 con MPEG4IP
 - Codice Open Source
 - RTP per il trasporto e RTSP per il controllo
 - Può erogare in MULTICAST (es. LAN di destinazione)
 - E' gestito mediante una INTERFACCIA WEB
- L'Interfaccia è realizzata come uno SCRIPT PERL che invia dei comandi HTTP a DSS
- I comandi sono stati analizzati con lo sniffer *ETHERREAL*
- Il controllo dei relay attivi si basa su un FILE XML generato dal Perl e letto dal Server



Adattamento per DSS

- L'Adaptation Layer per DSS EMULA il funzionamento della sua Interfaccia Web
- Controlla Darwin modificando il FILE XML e generando richieste HTTP GET
- è realizzato in PERL ed offre una API che implementa le primitive sottoriportate

DO RELAY	}	GESTIONE DEI MEDIA
NO RELAY		
QUERY BAND	}	CONFIGURAZIONE RELAY
SERVER STATUS		
SERVER ADDRESS		
CLIENT STATS	}	STATISTICHE D'USO
RELAY STATS		

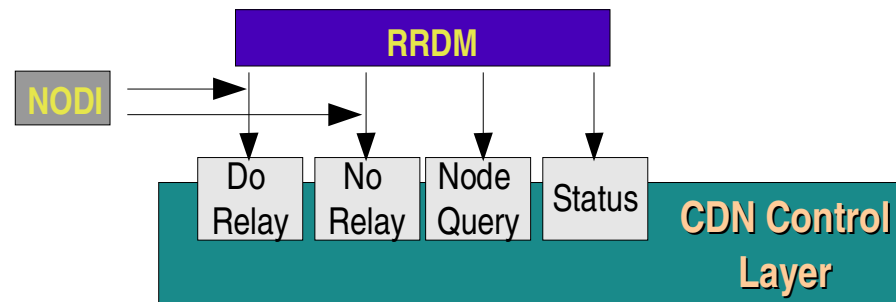


CDN Control Layer

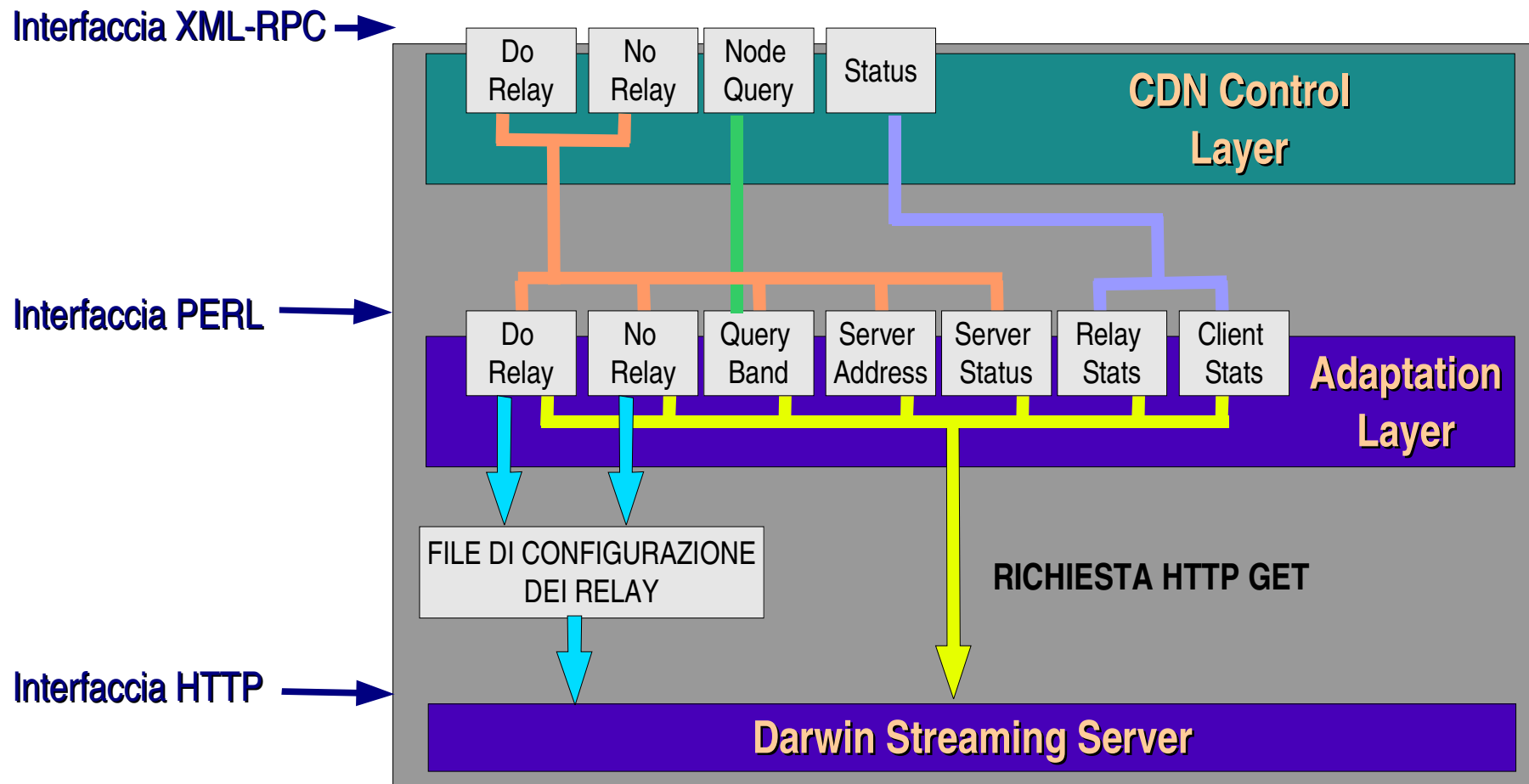
- Anch'esso realizzato in PERL
- Comunica con RRDM e con gli altri nodi con chiamate XML-RPC
- Usa le API dei Adaptation Layer per controllare lo SS
- Gestisce la recursione dei Transit
- Mantiene memoria dei propri nodi DownStream
- Implementa i metodi seguenti, invocabili dalle altre entità:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>NoRelay</methodName>
  <params> <param>
    <value>
      <struct>
        <member>
          <name>Program</name>
          <value>
            <string>rtsp://prova.mp4</string>
          </value>
        </member>
      </struct>
    </value>
  </param> </params>
</methodCall>
```

DO RELAY	}	GESTIONE DEL MEDIA
NO RELAY		
NODE QUERY	}	STATISTICHE D'USO
STATUS		

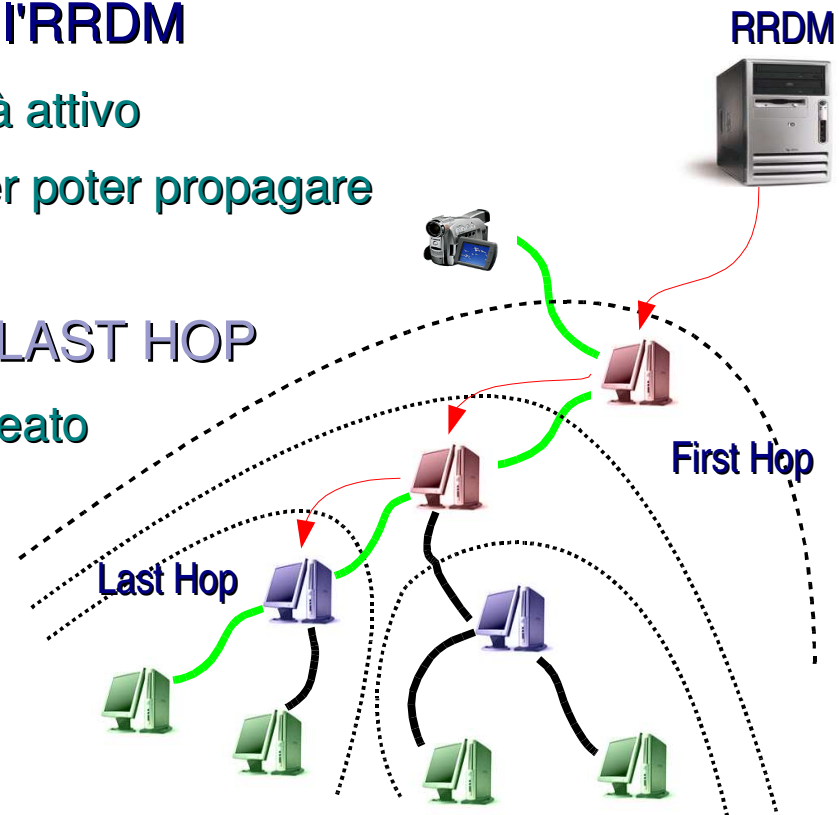


Struttura di Insieme del Nodo



Recursione per i TRANSIT RELAY

- L'RRDM dialoga con un solo nodo, il FIRST HOP, relativo alla Footprint più ampia per i client da servire
- Il FirstHop riceve un elenco di possibili downstreamers, ed assume per loro il ruolo che aveva l'RRDM
 - Può essere il più specifico Nodo già attivo
 - Ricorda il downstream prescelto per poter propagare il TearDown/NoRelay
- Quando la recursione raggiunge il LAST HOP
 - l'esito risale lungo il path che si è creato
 - l'RRDM è notificato di tutti i nuovi surrogati attivi

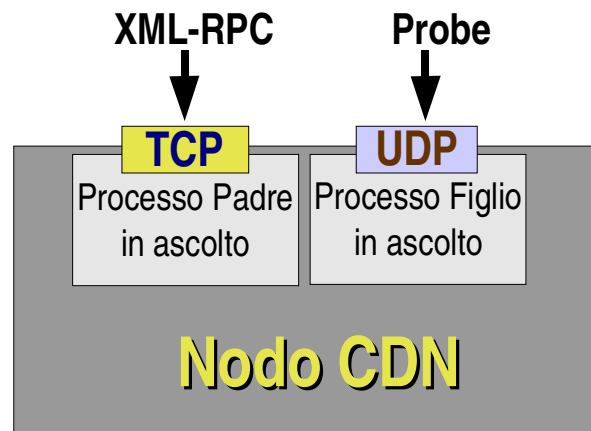


Adattamento alle Condizioni di Carico

- **OpenCDN può realizzare un bilanciamento di carico tra i Nodi, e rispondere alle variazioni delle condizioni della rete**
 - **Le chiamate XML-RPC viaggiano in TCP**
 - **Se un nodo è DOWN, occorre attendere il timeout!**
 - **Prima di contattare un Nodo, questo viene sondato via UDP**
 - **Più sondaggi sono svolti in parallelo, ed i nodi ordinati in base alla latenza, in modo da privilegiare**
 - **il meno carico**
 - **quello con migliori condizioni di rete**
-

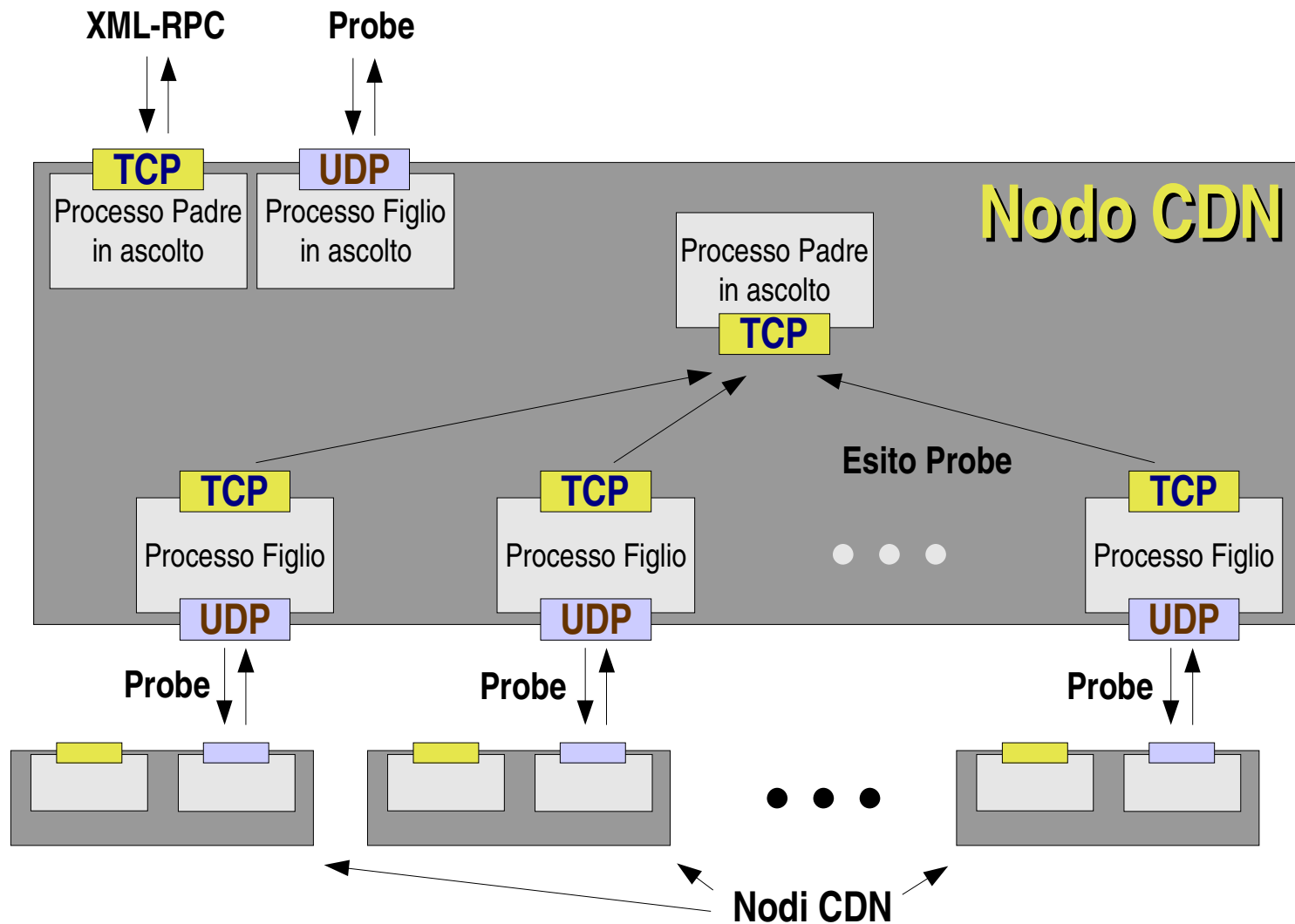
UDP Probe

- Il nodo è in ascolto anche su di un socket UDP, che re-invia al mittente i pacchetti ricevuti

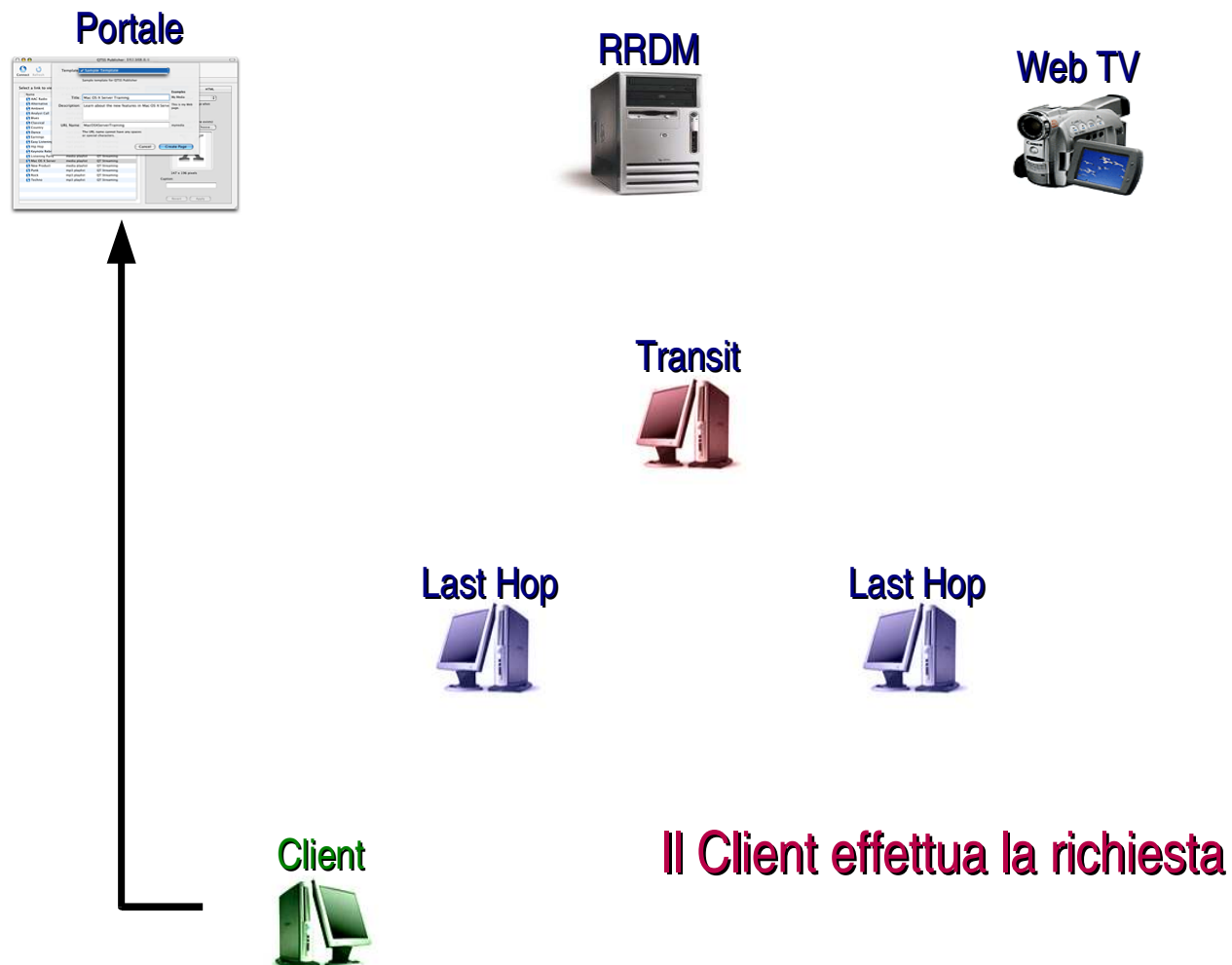


- L'invio e la ricezione degli **UDP PROBE** è realizzata da un secondo processo figlio, che comunica con il padre mediante un socket TCP interno al nodo (*figura seguente*)
-

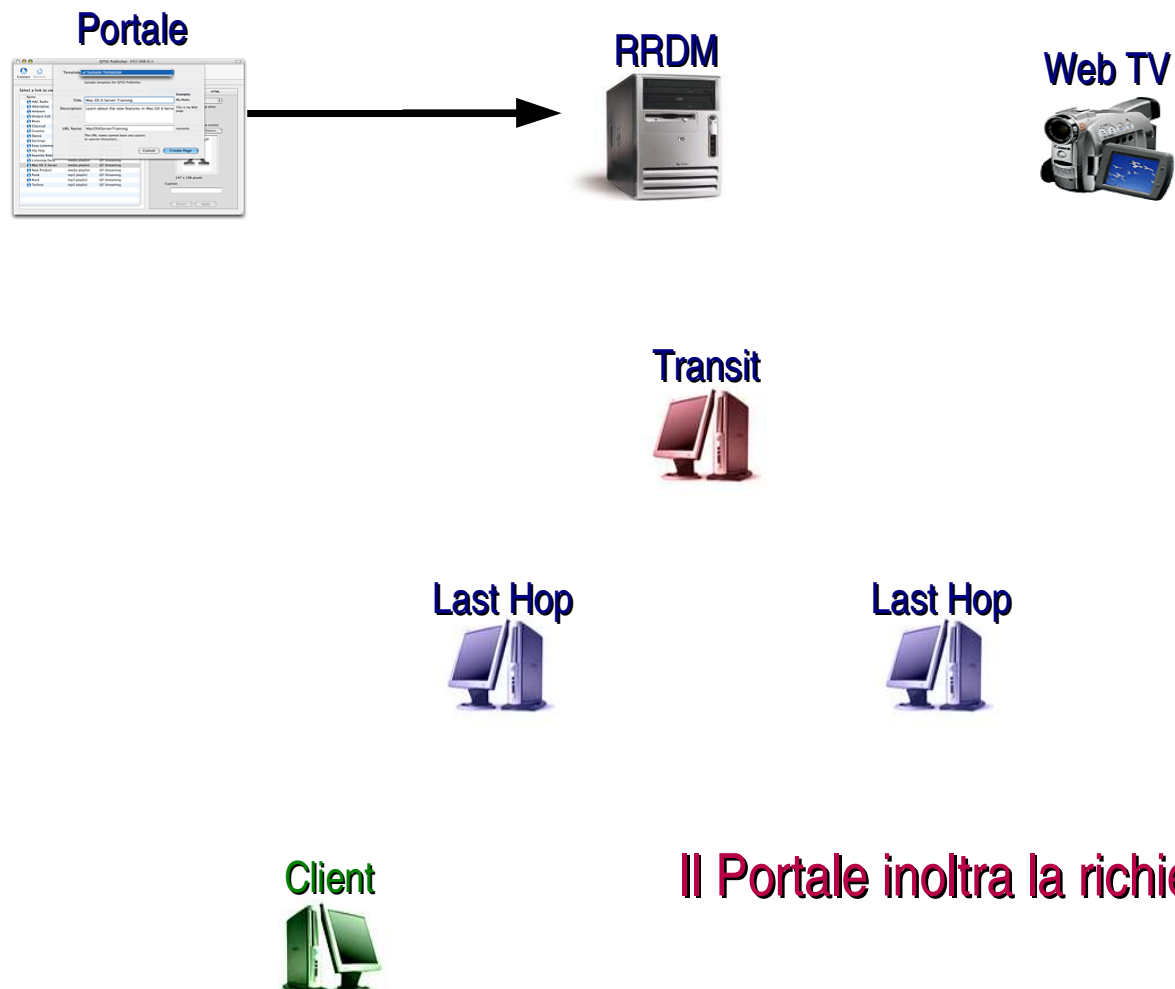
UDP Probe – 2



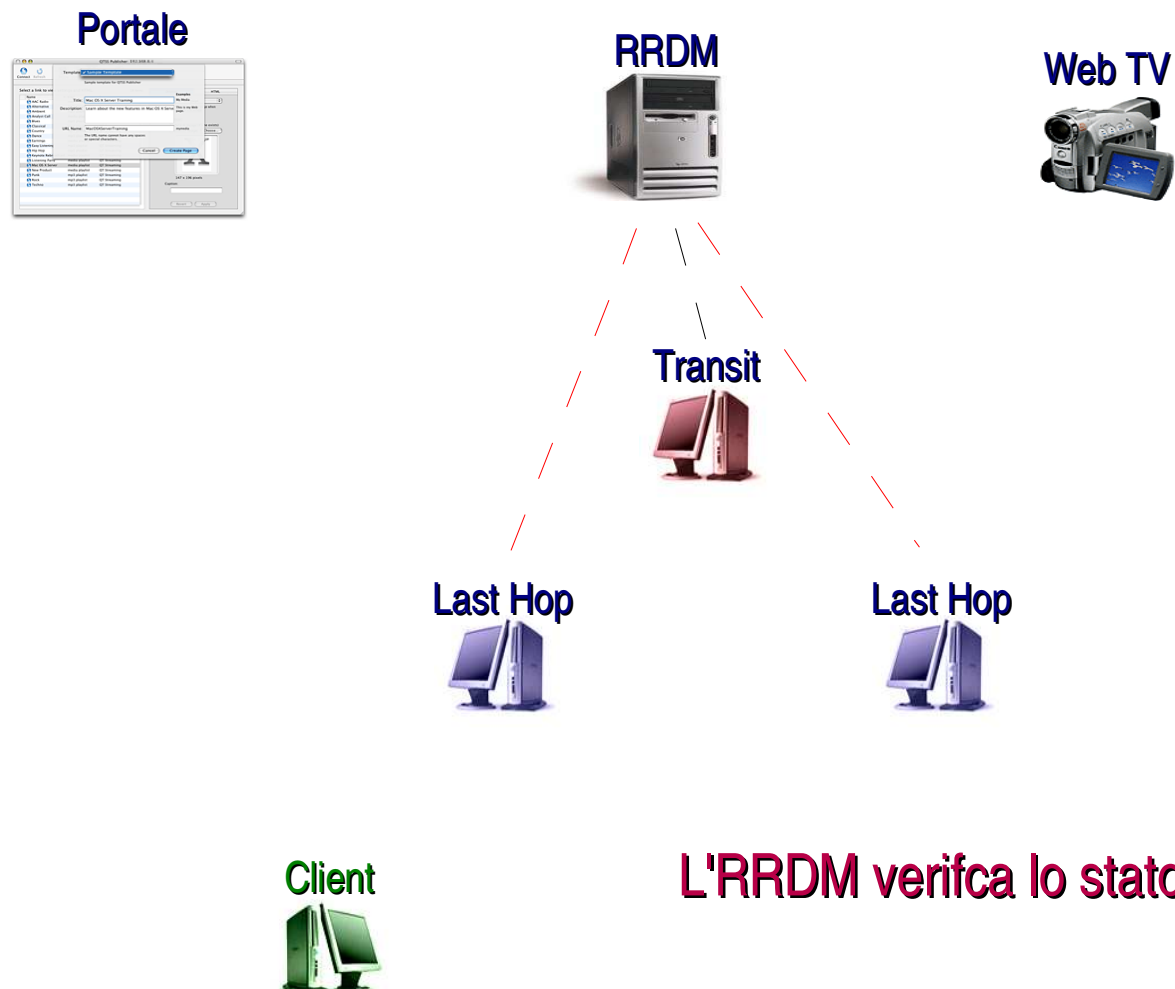
Topologia Gerarchica e Adattativa – 1



Topologia Gerarchica e Adattativa – 2



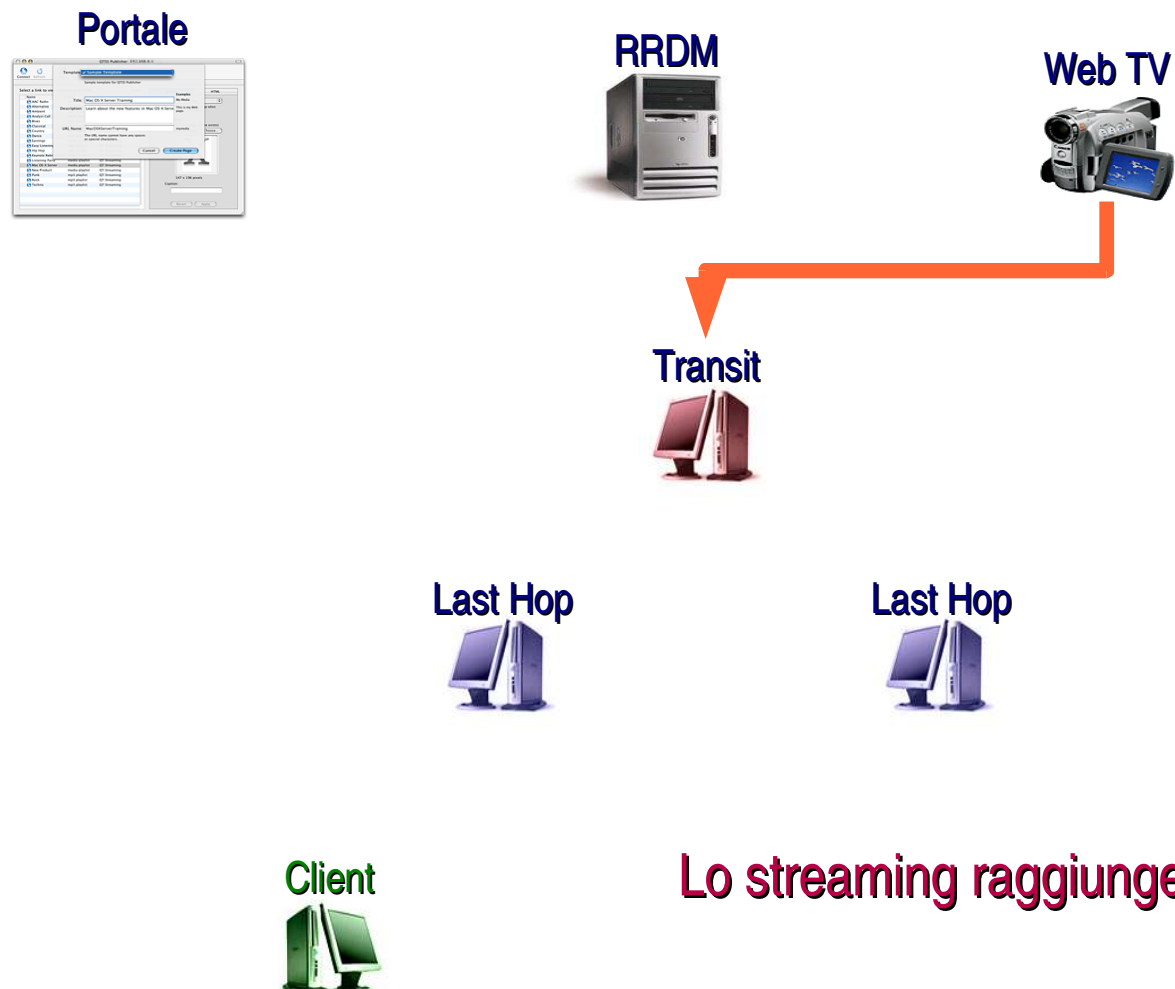
Topologia Gerarchica e Adattativa – 3



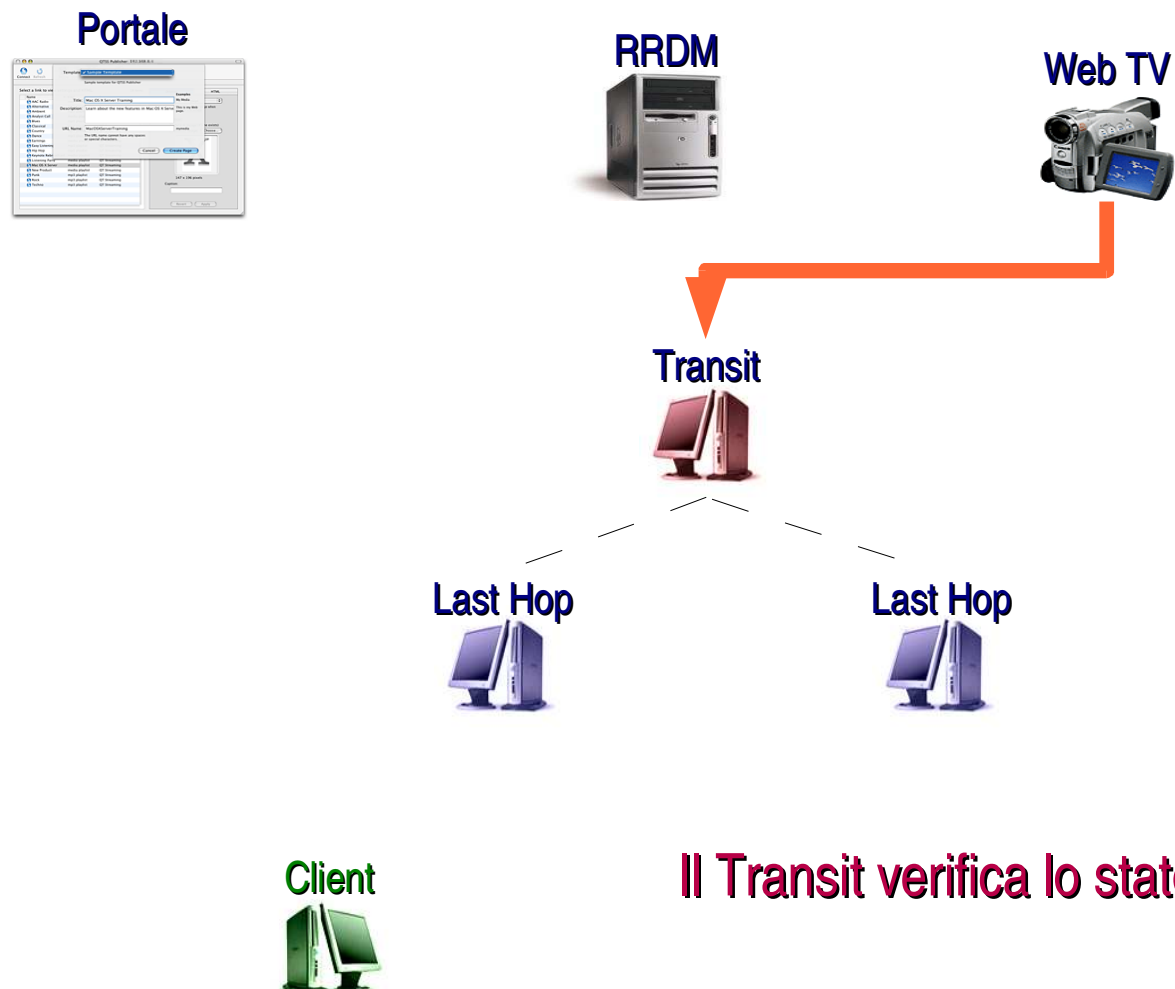
Topologia Gerarchica e Adattativa – 4



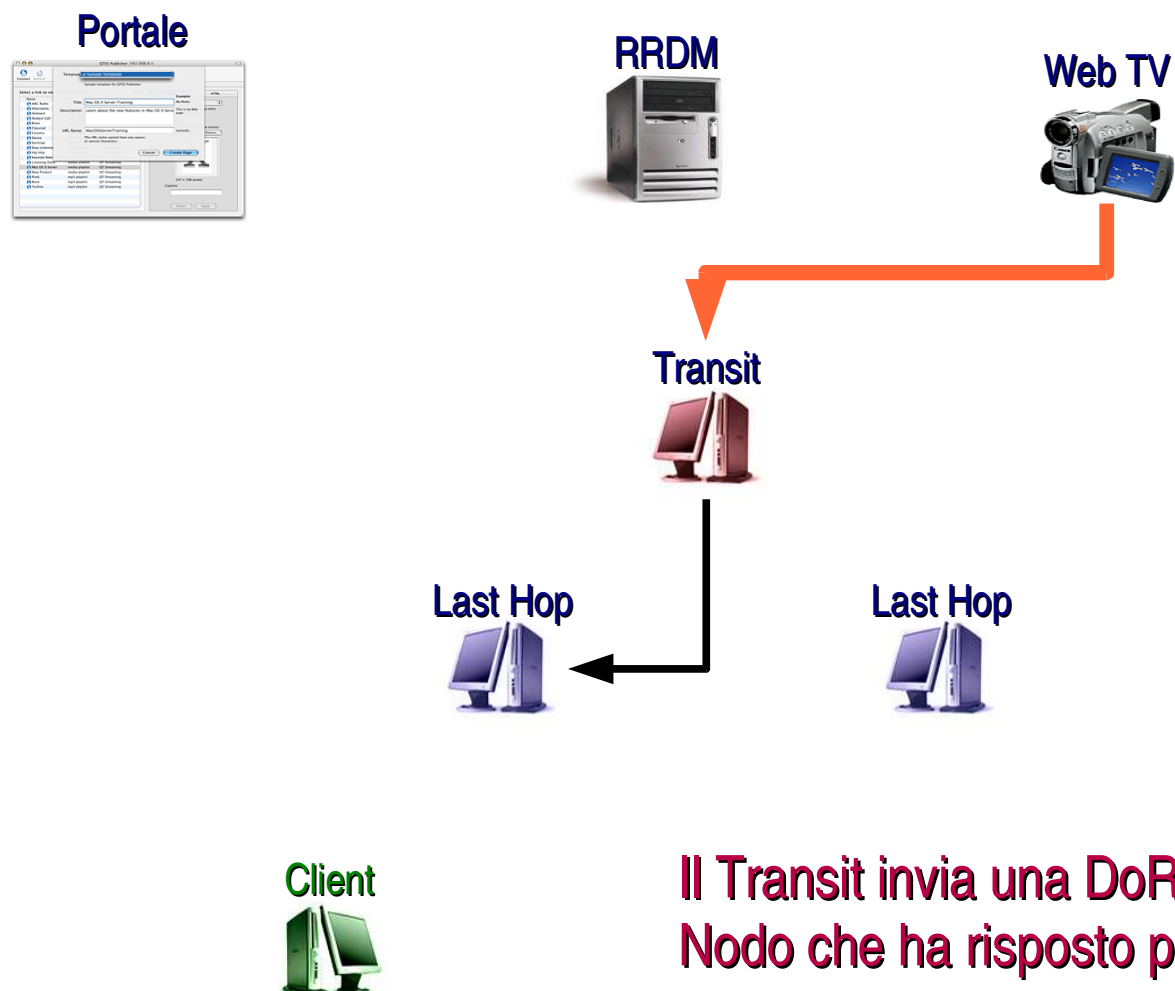
Topologia Gerarchica e Adattativa – 5



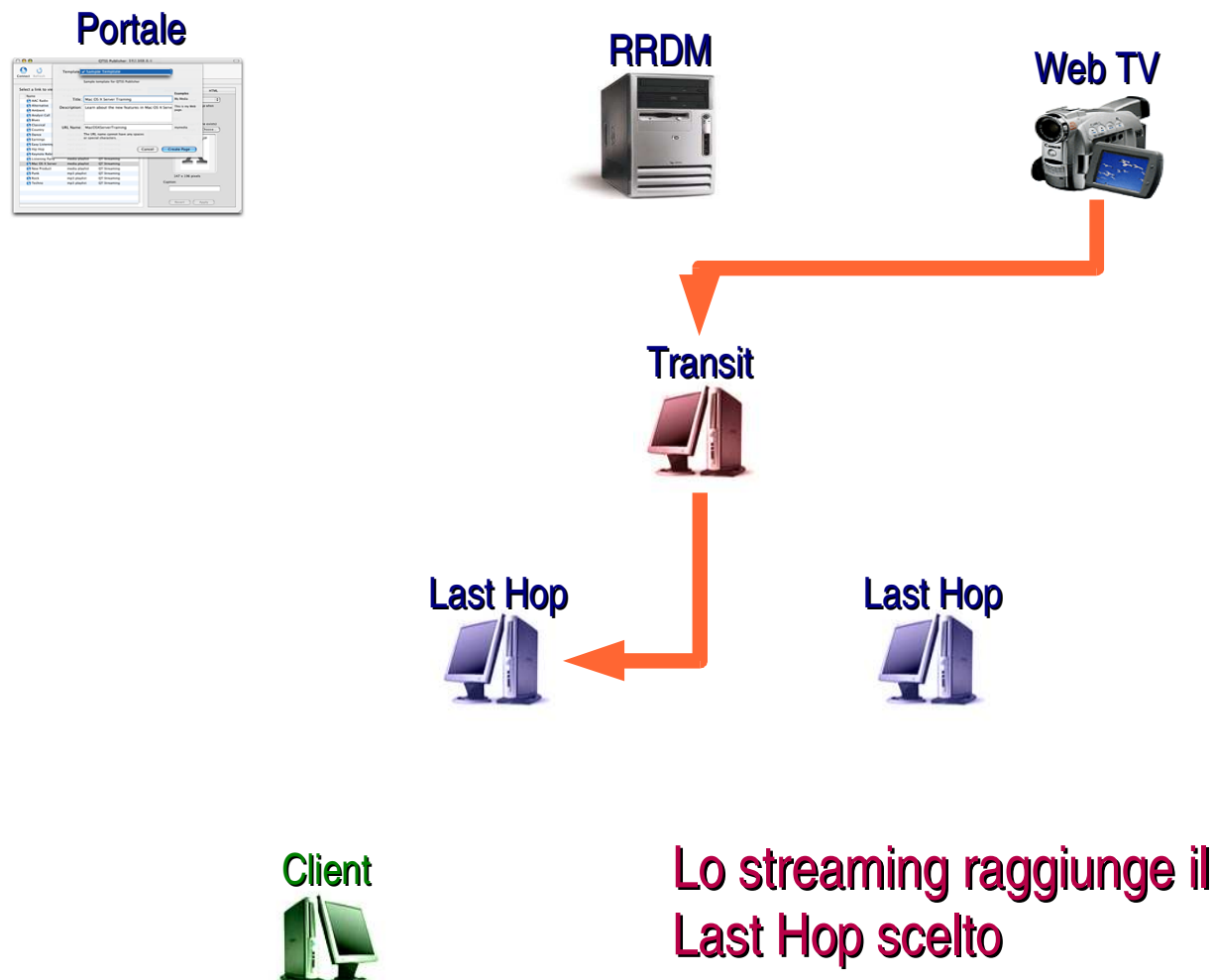
Topologia Gerarchica e Adattativa – 6



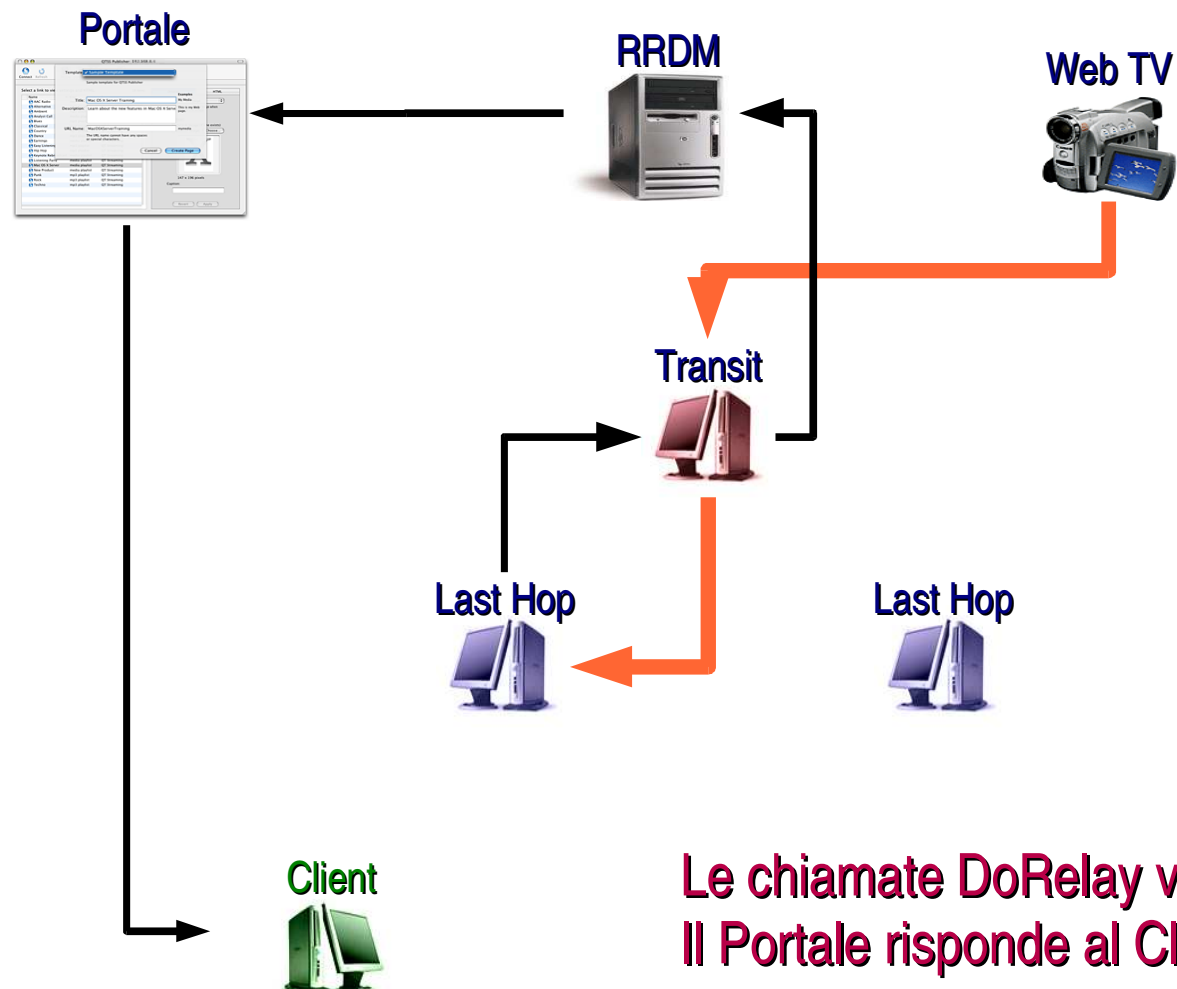
Topologia Gerarchica e Adattativa – 7



Topologia Gerarchica e Adattativa – 8

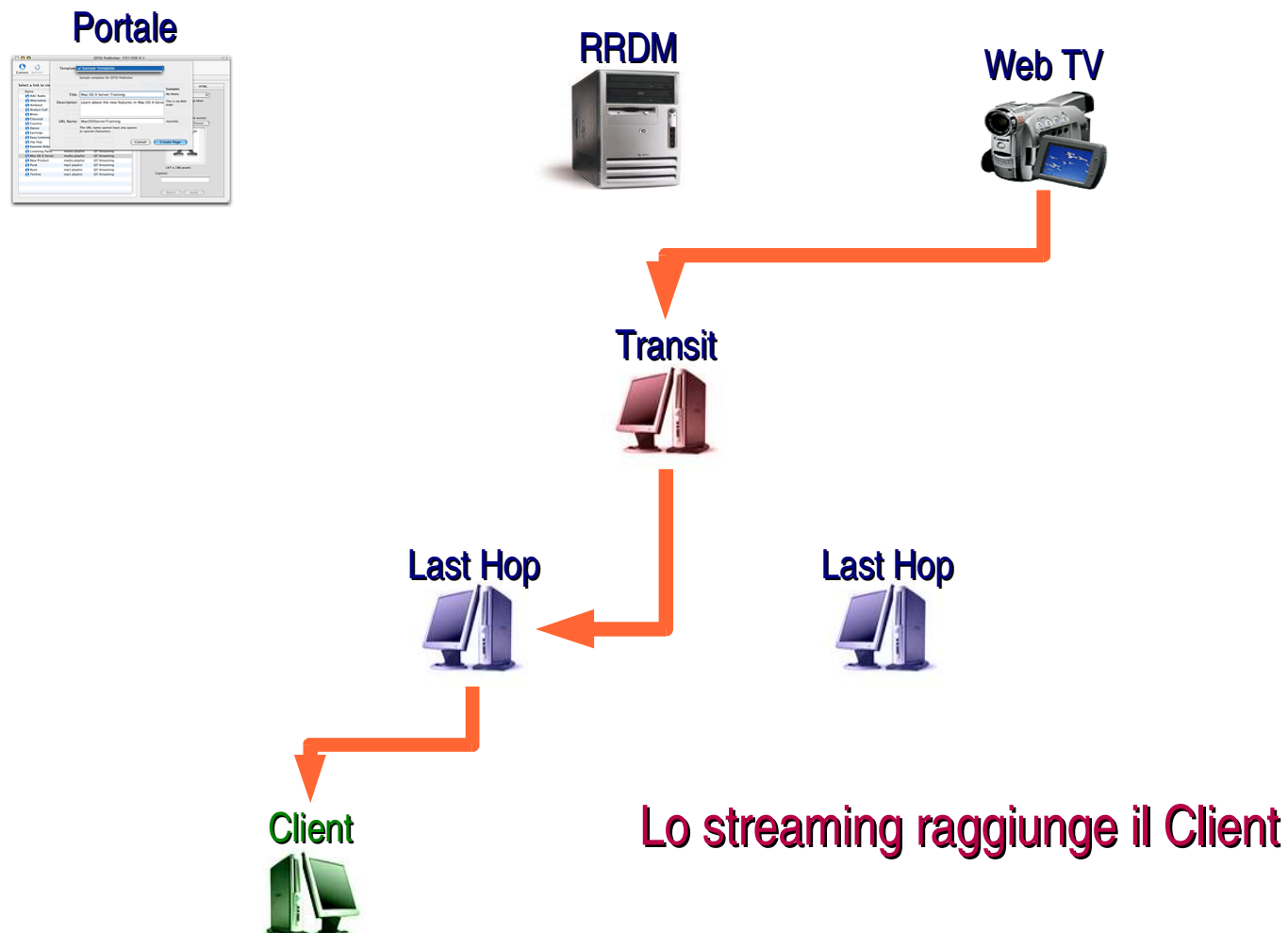


Topologia Gerarchica e Adattativa – 8



**Le chiamate DoRelay vengono chiuse
Il Portale risponde al Client**

Topologia Gerarchica e Adattativa – 9



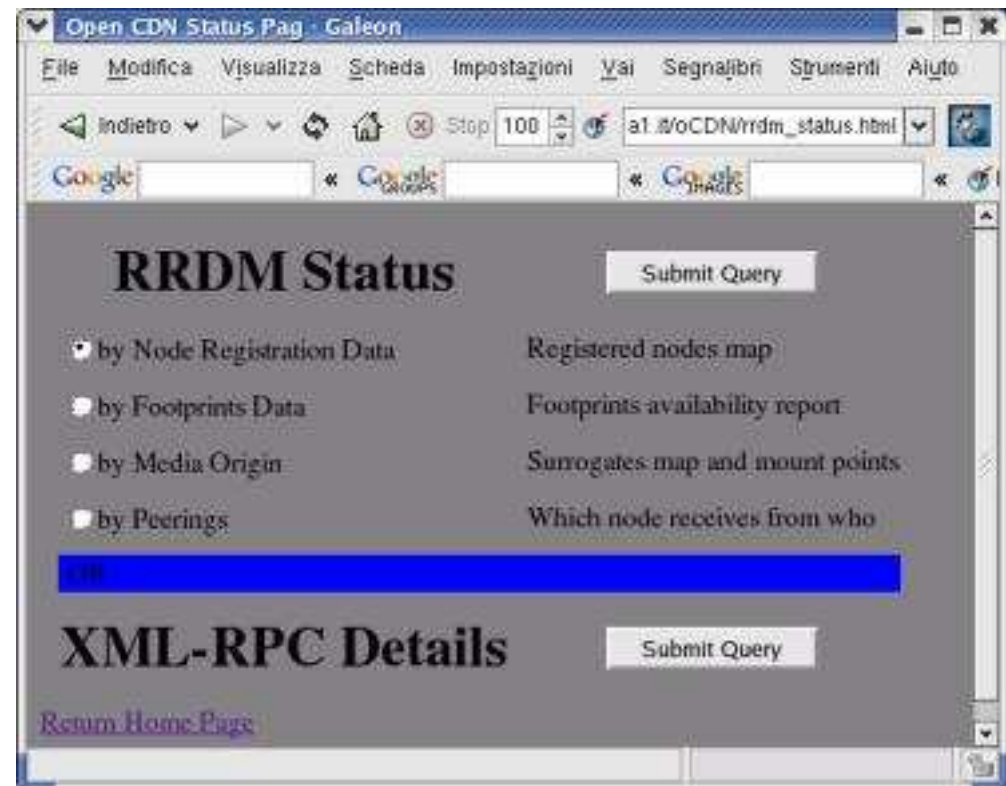
Request Routing and Distribution Managment

- Realizzato anch'esso in PERL
- Memorizza le Footprint annunciate dai Nodi
- Stabilisce la gerarchia di preferenza dei Nodi, per un dato indirizzo del client
- Memorizza i Relay attivi per i diversi Programmi, ed i DownStream diretti a cui propagare il TearDown
- Implementa i metodi XML-RPC

REGISTER	}	GESTIONE CDN
SETUP		
TEARDOWN		
STATUS	}	ACCESSO AI DATI
IPC		

Esecuzione concorrente

- L'RRDM può usare il modulo Net::Server di Perl, per rispondere in modo concorrente a più richieste contemporanee di SetUp
- Un secondo processo non forkato mantiene i dati delle registrazioni, dei Relay attivi e dei Downstream
- La comunicazione interprocesso è attuata a sua volta mediante chiamate XML-RPC
- I contenuti delle strutture dati sono accessibili dall'esterno



Installazione e Configurazione

- **Al raggiungimento di una versione stabile, sono prodotte delle Milestone Release Tarball**
 - **E' attivo un server CVS per lo sviluppo cooperativo ed il reperimento della versione corrente**
 - **La distribuzione contiene un README con le istruzioni di installazione e configurazione**
 - **Files di configurazione permettono di indicare il valore dei parametri operativi delle entità**
 - **Script CGI permettono di verificare l'operatività di RRDM e di realizzare una infrastruttura di Test**
 - **Disponibile presso <http://labtel.ing.uniroma1.it/opencdn>**
-

Pagina di Test

- Invoca i metodi di SETUP o di TEARDOWN
- Due sorgenti disponibili
- CDN di Test con 15 nodi virtuali su indirizzi privati, e client address fittizio

Open CDN Test Page

The multi-level CDN is under development.
Do you like to try if it works for you ? If so, please

[Help](#)
[Status](#)
[About](#)

select one of these actions:

- | | |
|--|--|
| <input type="radio"/> Service Request | It asks to the RRDM to set up distribution (if not exists yet) of the origin, and will return the best surrogate URL, from where you can pick up the content |
| <input type="radio"/> Teardown Request | This will dismantle the content distribution network for the given origin. |

then, choose the origin:

<input type="text" value="LiveTV"/>	After a timeout of 30 minutes, the LiveTV source stops. In order to restart it, please take again the ServiceReq action.
-------------------------------------	--

and finally start the game !

(check this ☒ if a forked RRDM is in use)

Comments to alef@infocom.uniroma1.it

Alternative client address

You can change the client IP address from where oCDN will receive your request. This gives you the chance of verify the outcome of request routing.

Actually, a four level test network is in place, whose nodes advertise *footprints* ranging from 127.0.0.0/16 to 127.0.224.0/19.

For this reason, you may simulate one the following client addresses:

127.0. .1 ☐ Yes, use it!

Futuro

- **Aggiunta di**
 - nuove sorgenti alla test page
 - ulteriori Nodi-Darwin presso altre sedi
 - nuovi RRDM presso altre sedi
 - **Sviluppo dell'Adaptation Layer per altre piattaforme**
 - **Sviluppo delle features mancanti**
 - Uso di thread anziché processi figli
 - Generazione del grafico della topologia risultante
 - Footprint come suffissi di Dominio
 - Probe tra sorgenti e FirstHop
 - Impiego di informazioni sulla *salute* dei Nodi
 - Autenticazione dei messaggi tra le entità
 - Componente di Accounting
 - **Sperimentazione di nuove soluzioni**
 - *Binning* per individuare i Transit indipendentemente dai Footprint
-