

Alessandro Falaschi

OpenCDN: una architettura estensibile per la condivisione di contenuti Internet

*(era: OpenCDN: una architettura estensibile
per il Live Streaming Internet)*

Pisa, 10-13 Maggio 2005

Conferenza GARR_05 "la rete daPERtutto"

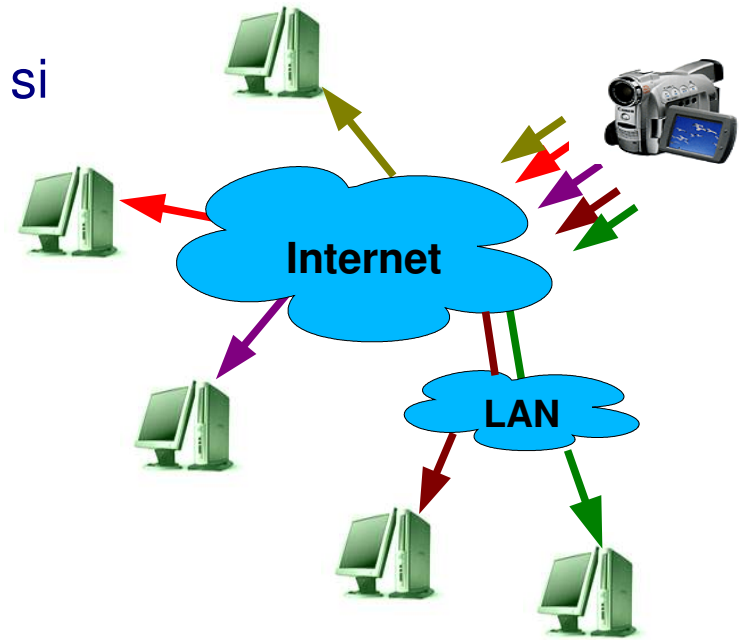
Di che si tratta ?

- Evolve nell'ambito delle Task Force **NETCAST** e **VVC** di TERENA, per problemi di configurazione e scalabilità del Live Streaming Internet, come un progetto **OpenSource**
- Generalizzabile alla distribuzione di oggetti qualunque, e permette di localizzare il nodo di replica più prossimo al richiedente (*es. sourceforge*)
- Una **Content Delivery Network** estende il concetto di **Proxy**, realizzando le componenti di
 - **Consegna:** come replicare i contenuti tra entità
 - **Distribuzione:** come sviluppare la topologia di consegna
 - **Instradamento della richiesta:** come deviare un client verso il nodo di replica più opportuno



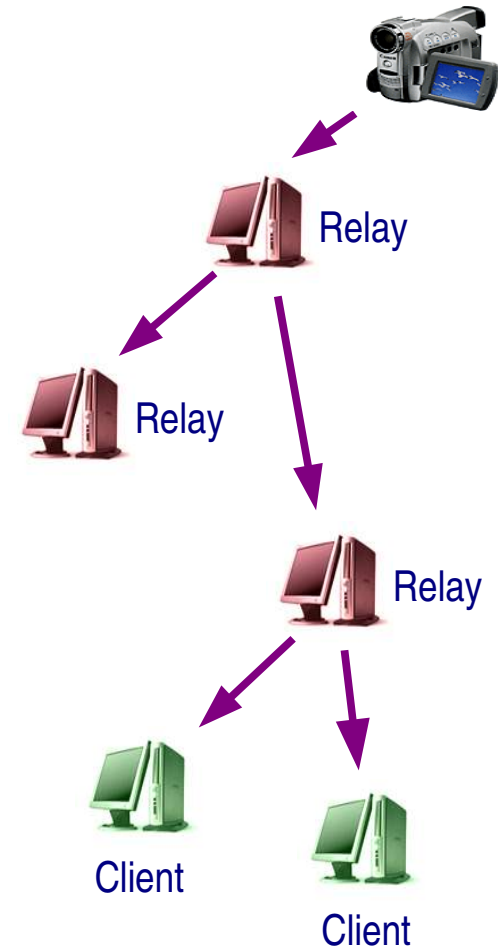
Il problema di scalabilità del Live Streaming

- nell' architettura **Client-Server** il *primo miglio* si satura presto all'aumentare del numero di clients
- se i clients in una medesima LAN sono numerosi, anche l'*ultimo miglio* può saturare
- la soluzione **Multicast** non è praticabile a causa del suo mancato supporto alla periferia, ed al mancato peering degli ISP
- il trasporto RTP non e' supportato dai Proxy (e non avrebbe senso)

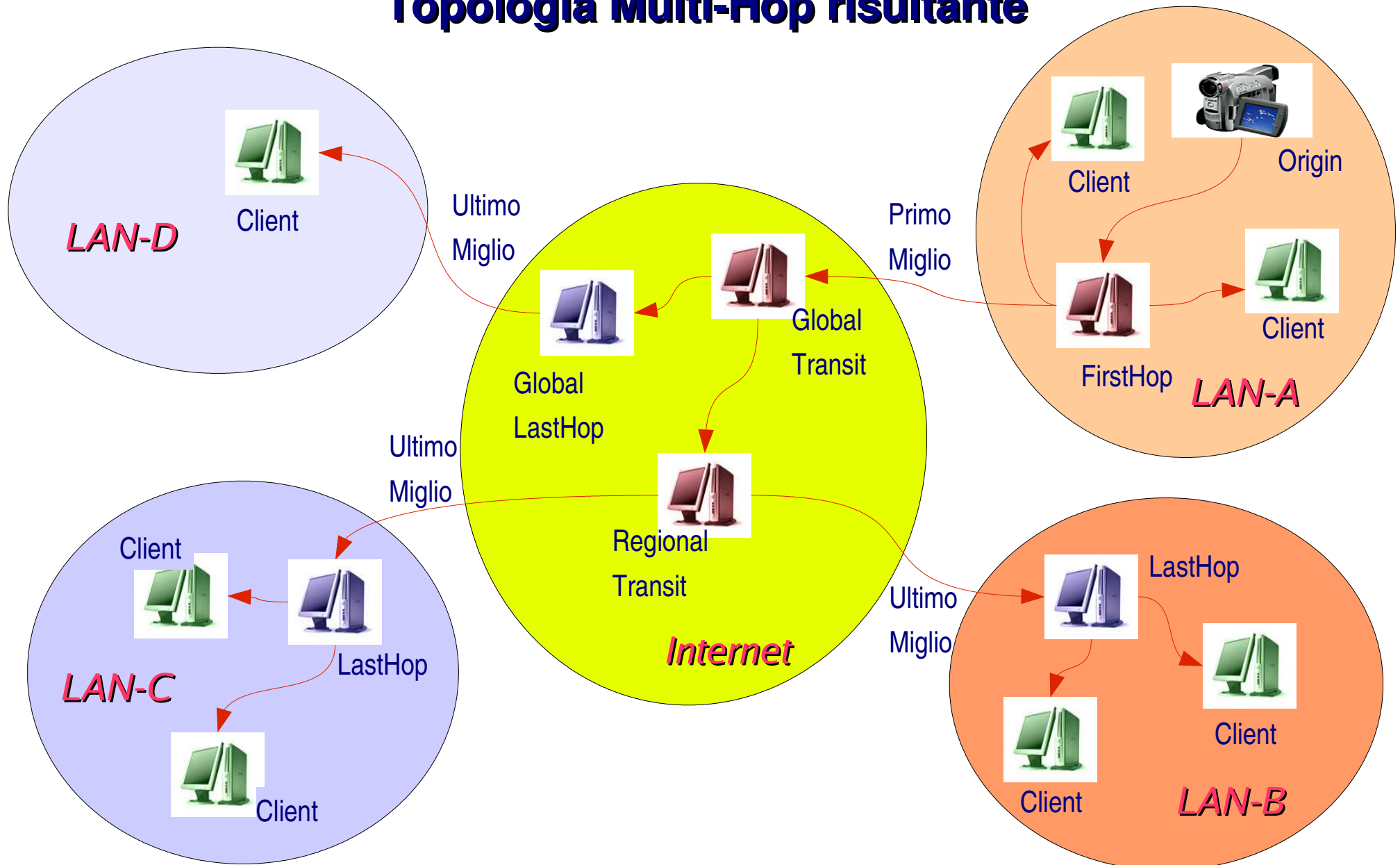


La soluzione dell'Application Level Multicast

- Al posto dei router si adottano agenti bifronte, contemporaneamente Client e Server, denominati **Relay**
- Un *piano di controllo* determina la topologia di distribuzione del contenuto, sostituendosi al **routing** (*ovvero, tunneling a livello applicativo*)
- La tecnologia dei Relay è mascherata da uno *strato di adattamento*, rendendo l'architettura **agnostica**
- La topologia risultante è determinata in modo automatico, **evitando configurazioni manuali** (*analogia al peer to peer, ma nella rete e non tra utenti*)

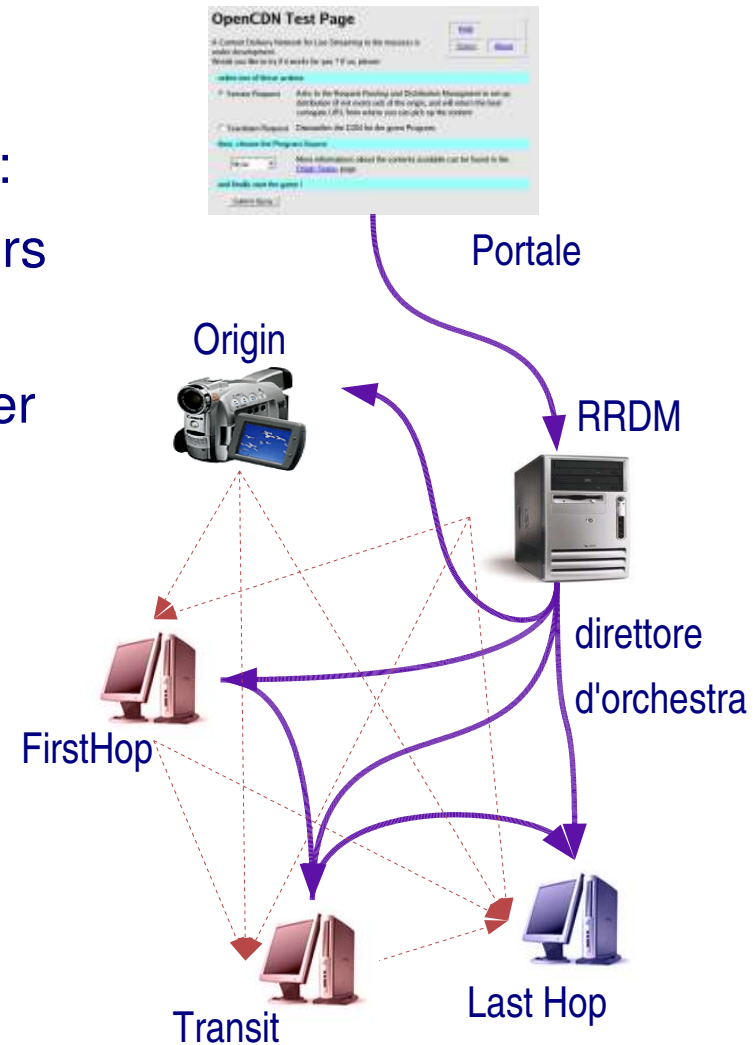


Topologia Multi-Hop risultante



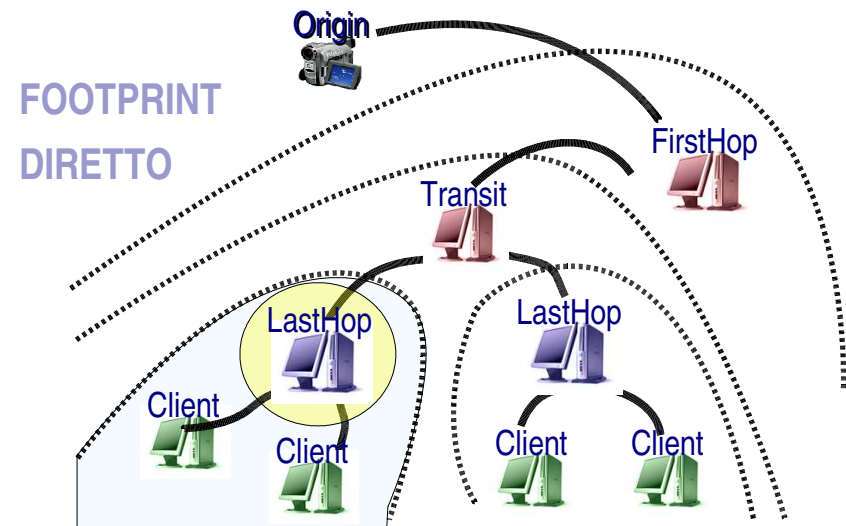
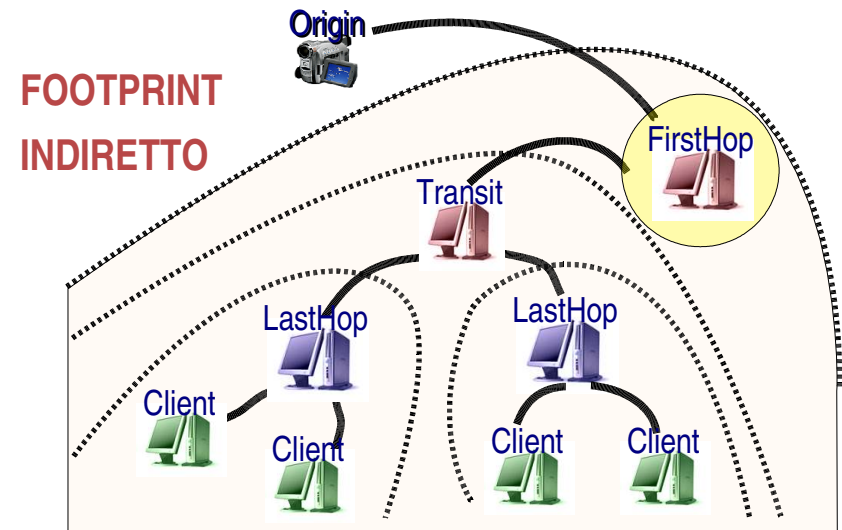
Entità & Protocolli

- OpenCDN si basa sulla esistenza di quattro entità:
 - Portale di Annuncio** – dove si rivolgono i viewers
 - Origin** – chi conferisce i contenuti
 - RRDM** – Request Routing & Distribution Manager
 - Nodo di replica** che a sua volta può essere un
 - First Hop** -> radice della distribuzione
 - Transit** -> serve altri Nodi
 - Last Hop** -> serve direttamente i viewers
- Nodi ed Origin aderiscono in modo **autonomo**, senza coinvolgimenti di terze parti
- Le entità comunicano tra loro mediante:
 - Chiamate **XML-RPC** – esecuzione distribuita
 - Probe **UDP** – misure di **RTT** (Round Trip Time) e persistenza



SetUp della Distribuzione

- La topologia dipende dai **Footprint** dichiarati dai nodi
 - individuano l'insieme dei destinatari che si intende servire
- rappresentati in alternativa come:
 - Un prefisso di rete IP
 - Un suffisso di Nome di Dominio (di prossima realizzazione)
- Un Footprint può essere:
 - Indiretto --> il nodo agisce come TRANSIT
 - Diretto --> il nodo agisce come LAST HOP
- per ogni client, la distribuzione attraversa i nodi che annunciano un footprint che ne racchiude l'indirizzo, ordinati **dal meno al più specifico**



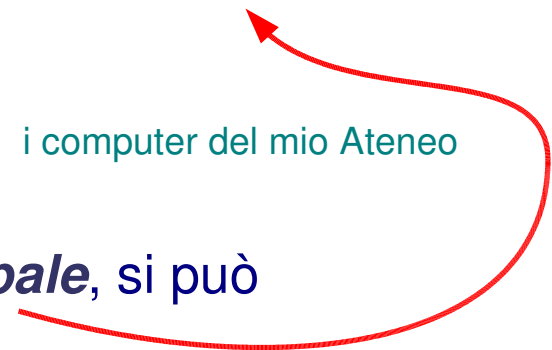
Scelta del surrogato periferico

- L'entità di controllo individua il nodo con Footprint più specifico rispetto al client e lo comunica la portale

oppure

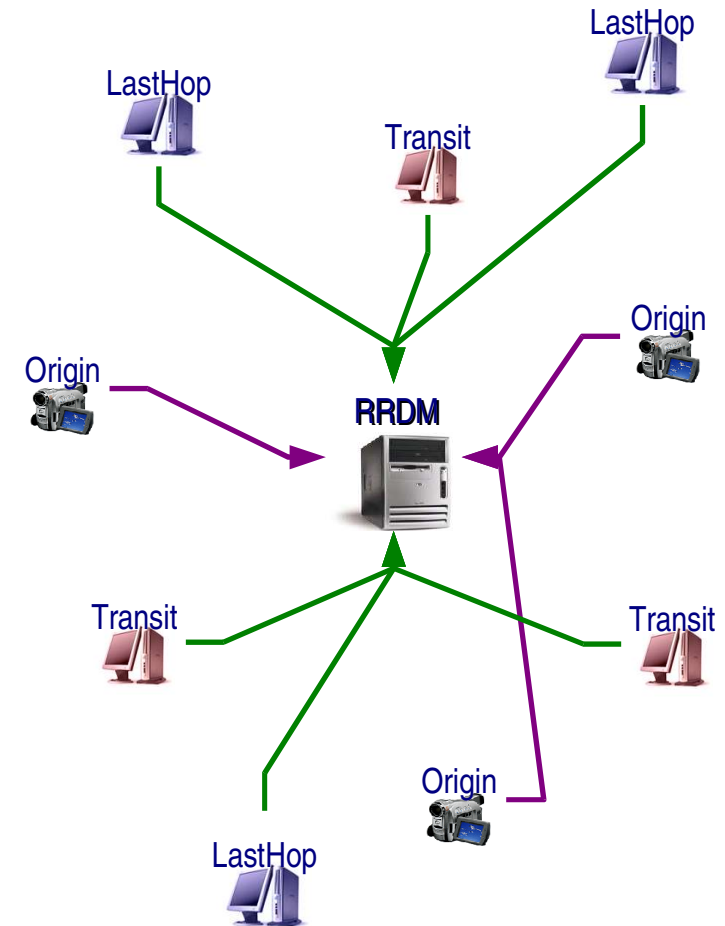
- se diversi nodi hanno annunciato una FootPrint **globale**, si può applicare un derivato di **PYTE** (CINECA), e cioè:
 - Il portale *include mini-immagini* che il client scarica *dai surrogati globali*
 - I surrogati misurano il tempo di download delle immagini, ossia il **RTT**
 - prima dell'ultimo Hop, *i tempi sono confrontati*, e si sceglie il surrogato più prossimo al client

Esempi di Footprint:

- 151.100.122.0/24 = i computer del mio Dipartimento
 - 0.0.0.0/2
 - 64.0.0.0/2 = FootPrint globale di 2° ordine
 - 128.0.0.0/2
 - 192.0.0.0/2
 - .uniroma1.it = i computer del mio Ateneo
- 

Operazioni ricorrenti

- i **Relay** rinnovano periodicamente l'informazione dei propri **Footprint**
- le **Origin** pubblicano ripetutamente i **metadata** relativi ai contenuti offerti
- il periodo di rinfresco *aumenta* con il numero delle entità
- la rete torna presto operativa anche in caso di *reboot* dell'entità di controllo



Pagina di Richiesta

- Permette di invocare i metodi di **SETUP** o di **TEARDOWN** →
- e di scegliere tra i contenuti pubblicati dalle **Origin** →

Open CDN Test Page

A Content Delivery Network for Live Streaming to the masses is under development.
Would you like to try if it works for you ? If so, please

[Help](#) [Fake CDN](#)

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

select one of these actions:

- Service Request: Asks to the Request Routing and Distribution Management to set up distribution (if not exists yet) of the origin, and will return the best surrogate URI, from where you can pick up the content
- Teardown Request: Dismantles the CDN for the given Program.

then, choose the Program Source:

More informations about the contents available can be found in the [Origin Status](#) page

and finally start the game !

- restituisce l'indirizzo ove prelevare il contenuto richiesto:

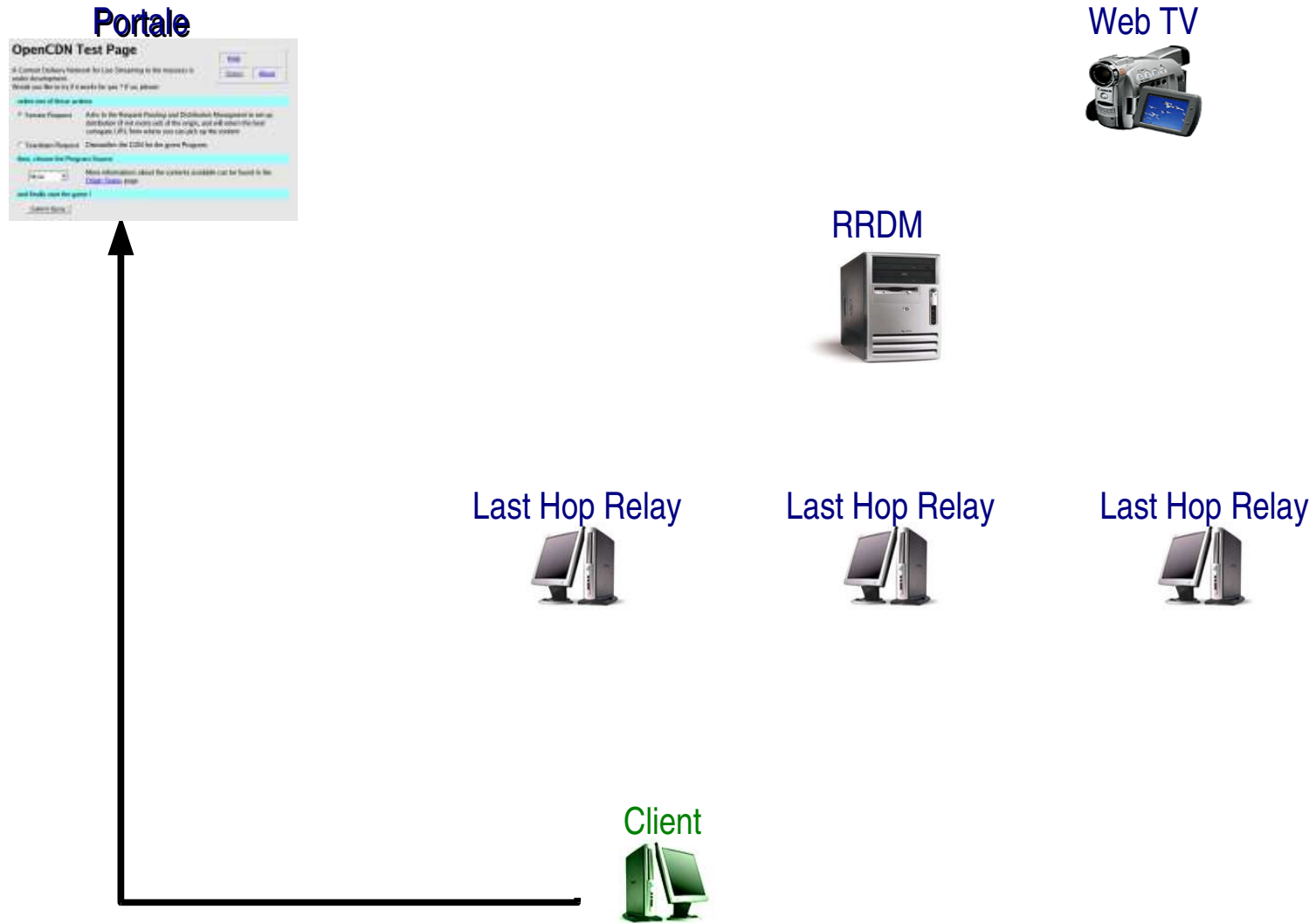
OpenCDN Response Page

Please receive the requested content at

rtsp://193.206.139.37/151.100.4.60_151.100.122.171_v-0.6.9-qenni.ing.uniroma1.it_WhatsItt1955.mp4.sdp

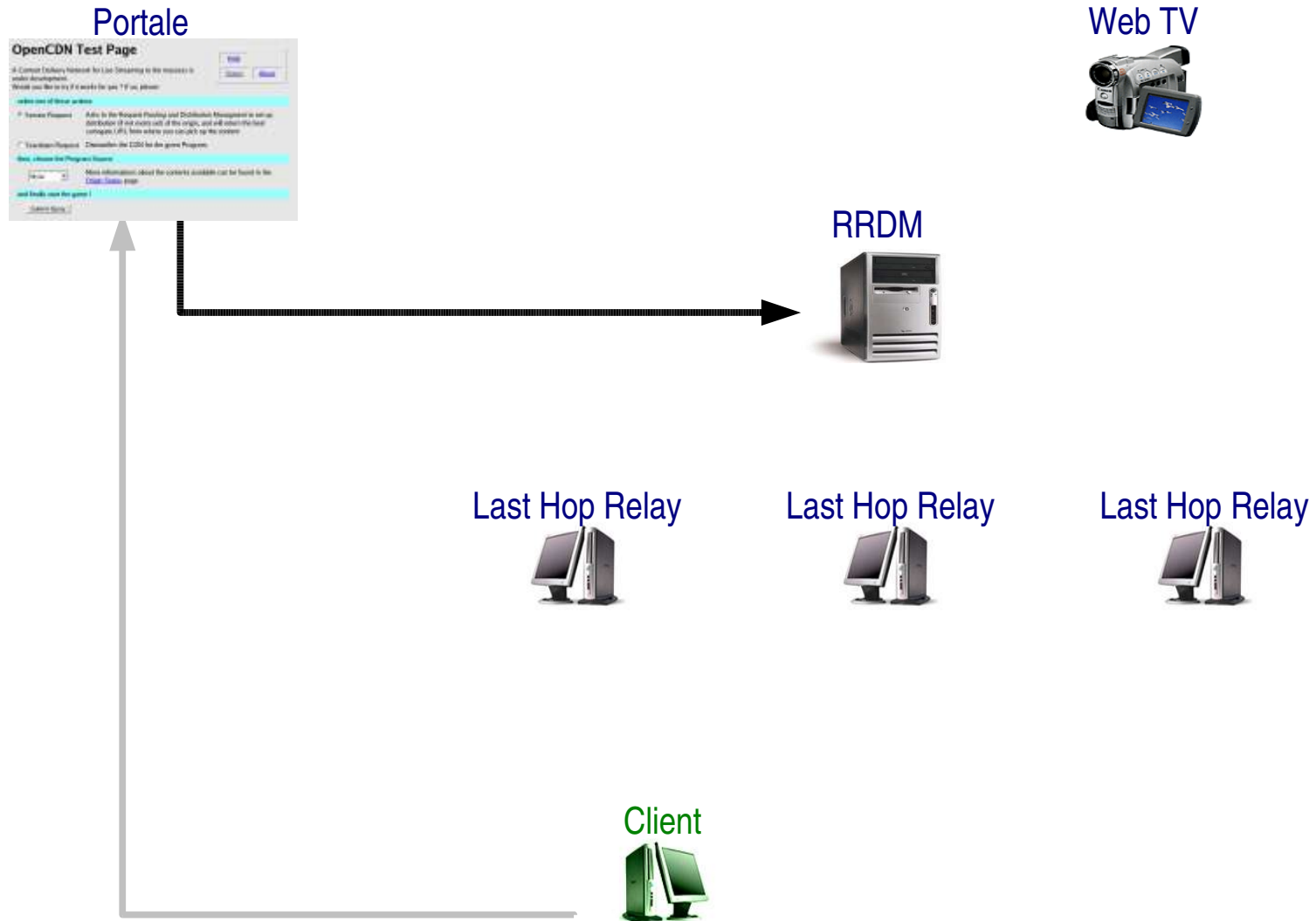
Note: at the first SetUp, you may need to wait some seconds before you start to receive media packets, because of buffering at each intermediate hop relay involved

Richiesta di Servizio



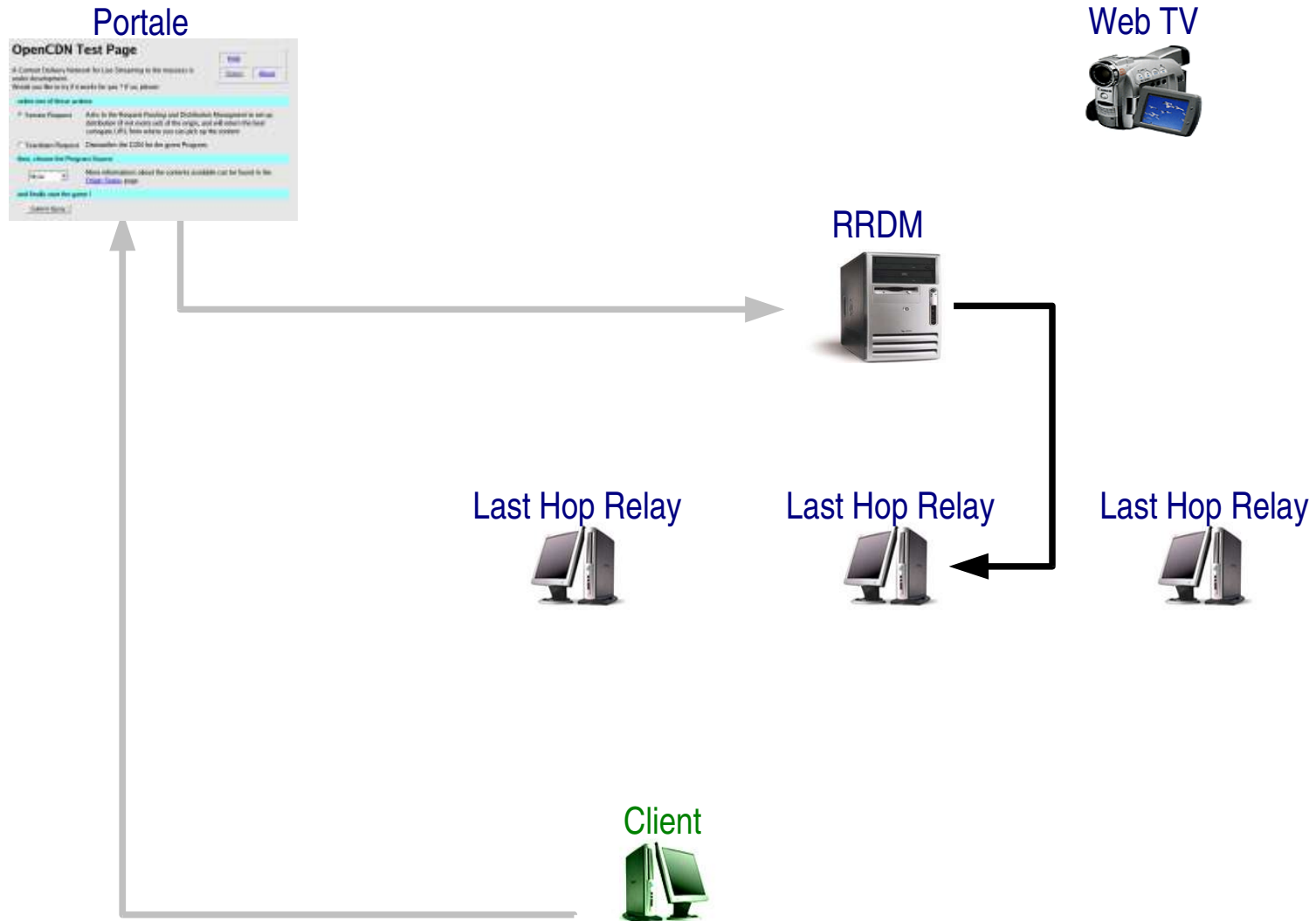
Il Client visita il Portale e richiede lo streaming della web tv

Richiesta di Servizio



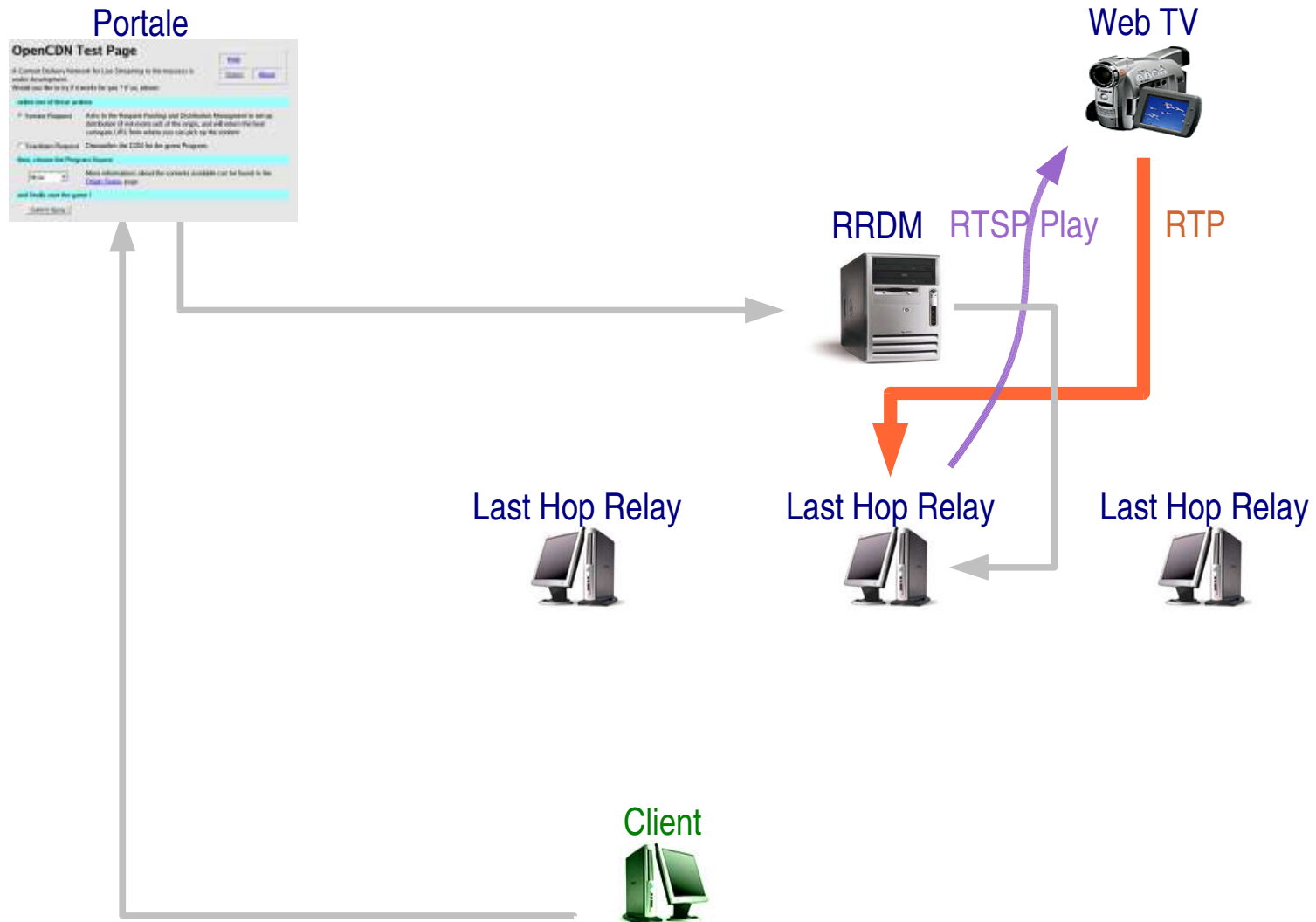
Il Portale inoltra la richiesta all'RRDM

Richiesta di Servizio



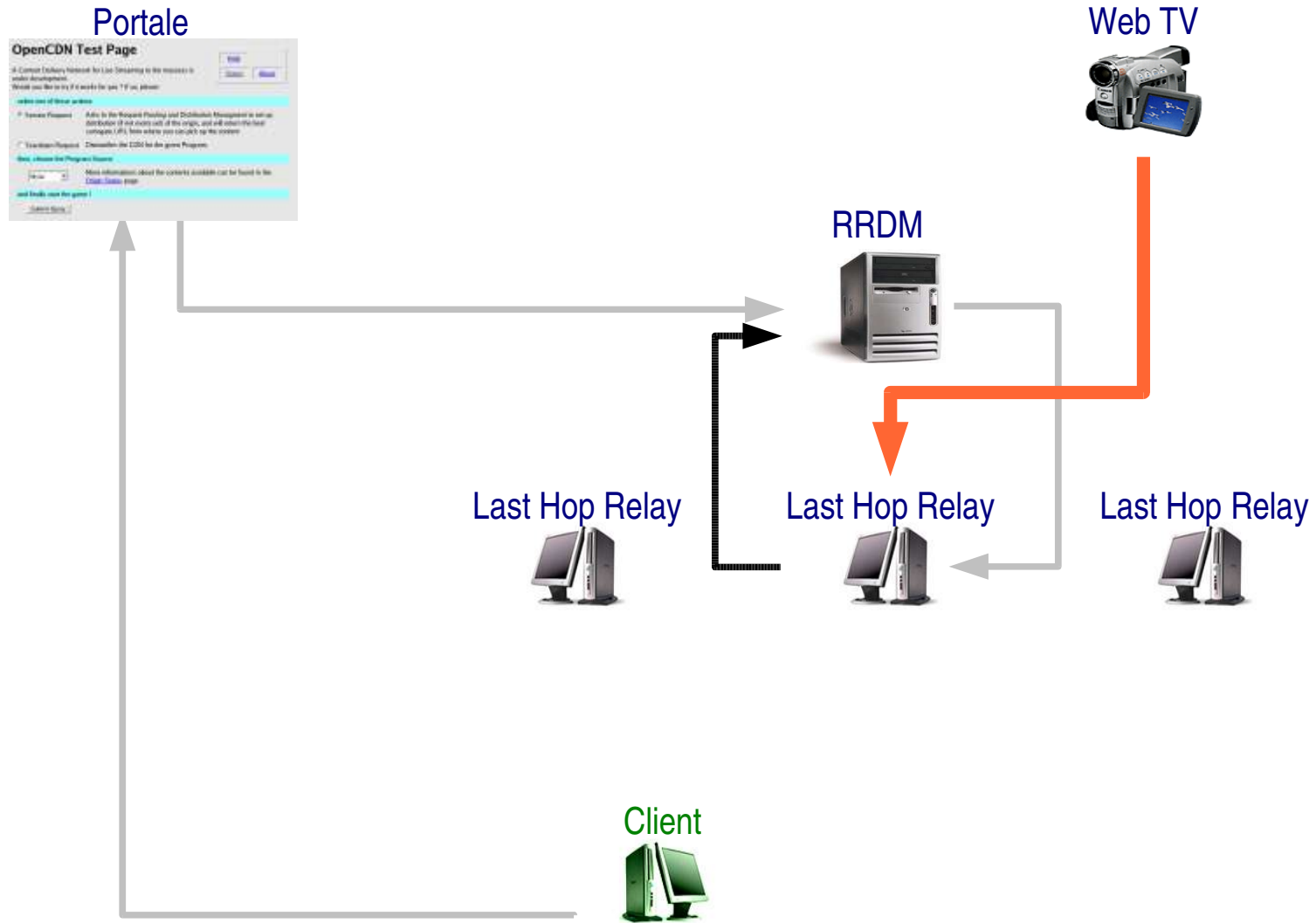
L'RRDM contatta il Last Hop pertinente indicandogli l'Origin da contattare

Richiesta di Servizio



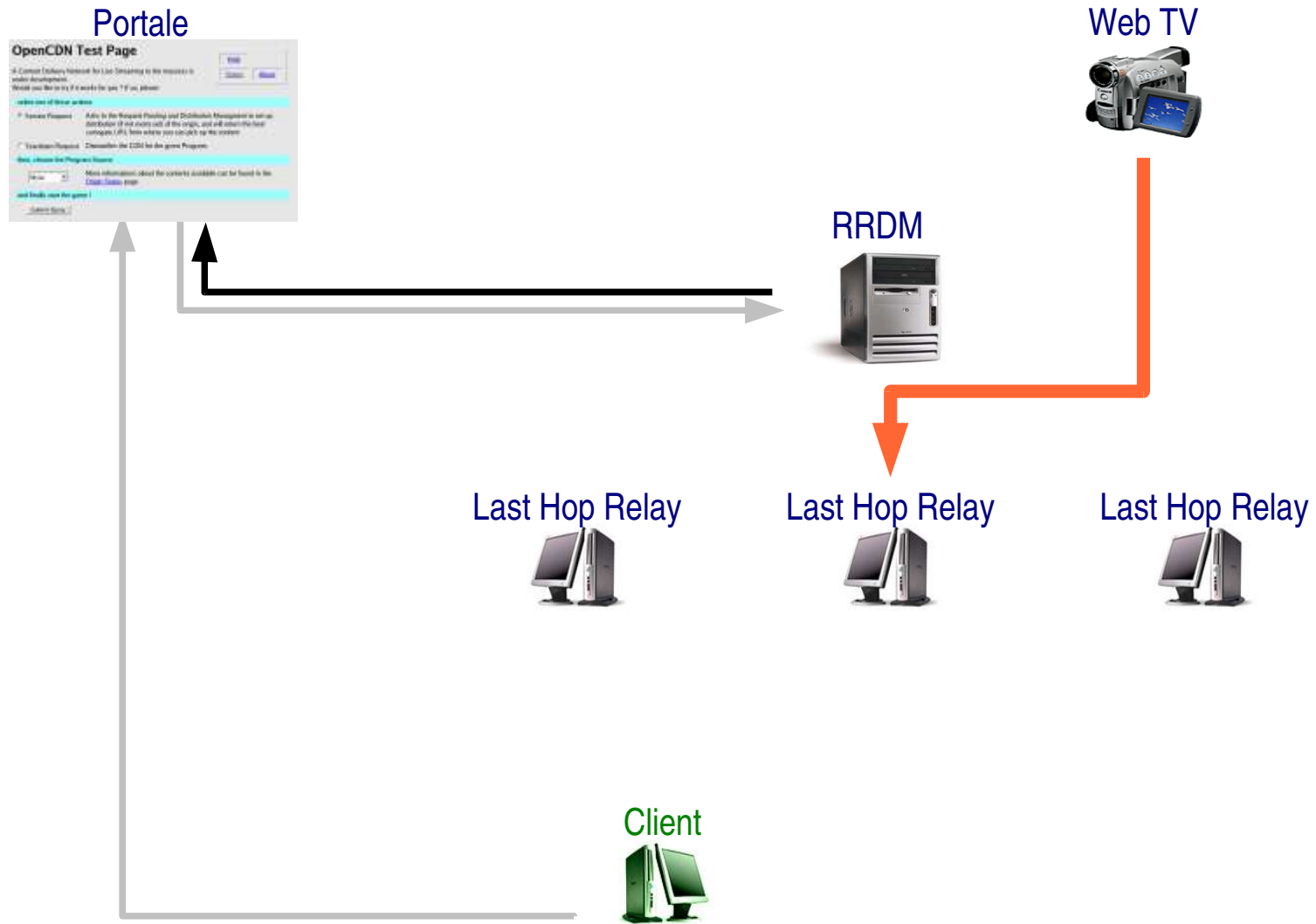
Lo streaming raggiunge il surrogato Last Hop

Richiesta di Servizio



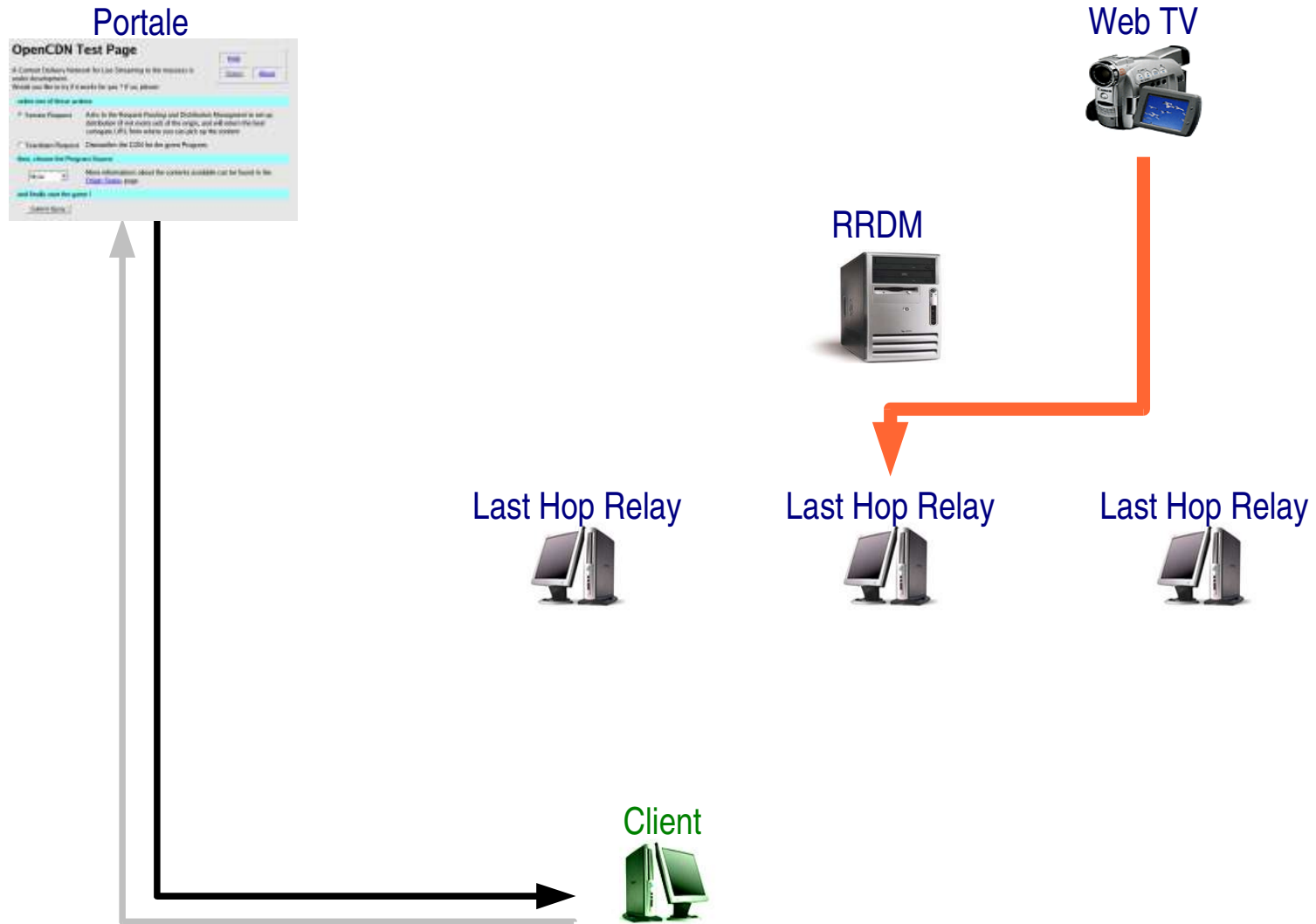
Il Last Hop conferma all'RRDM l'esito dell'operazione

Richiesta di Servizio



L'RRDM inoltra la risposta al Portale

Richiesta di Servizio



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

Richiesta di Servizio



RRDM



Last Hop Relay



Last Hop Relay



Last Hop Relay



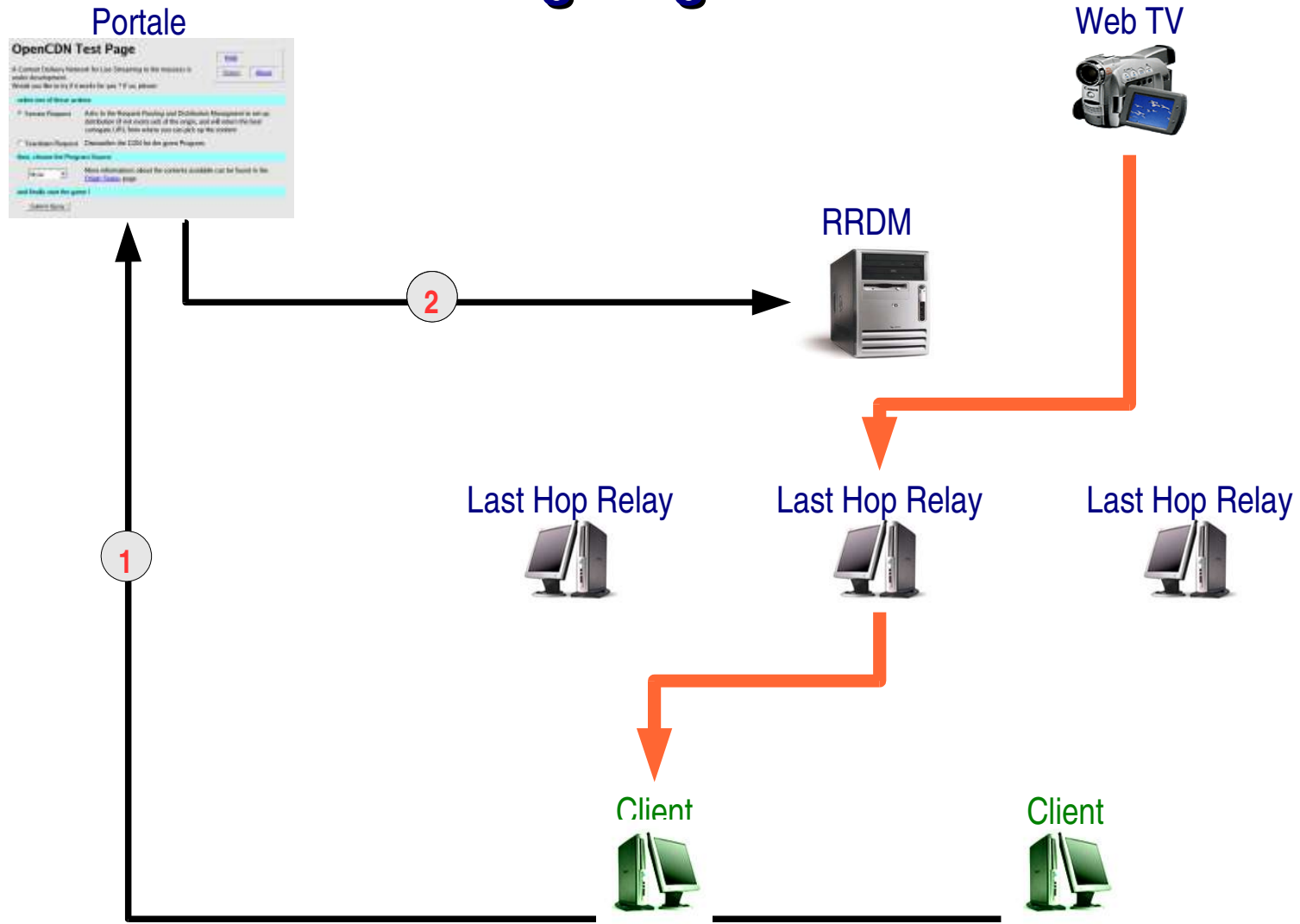
Client



Play

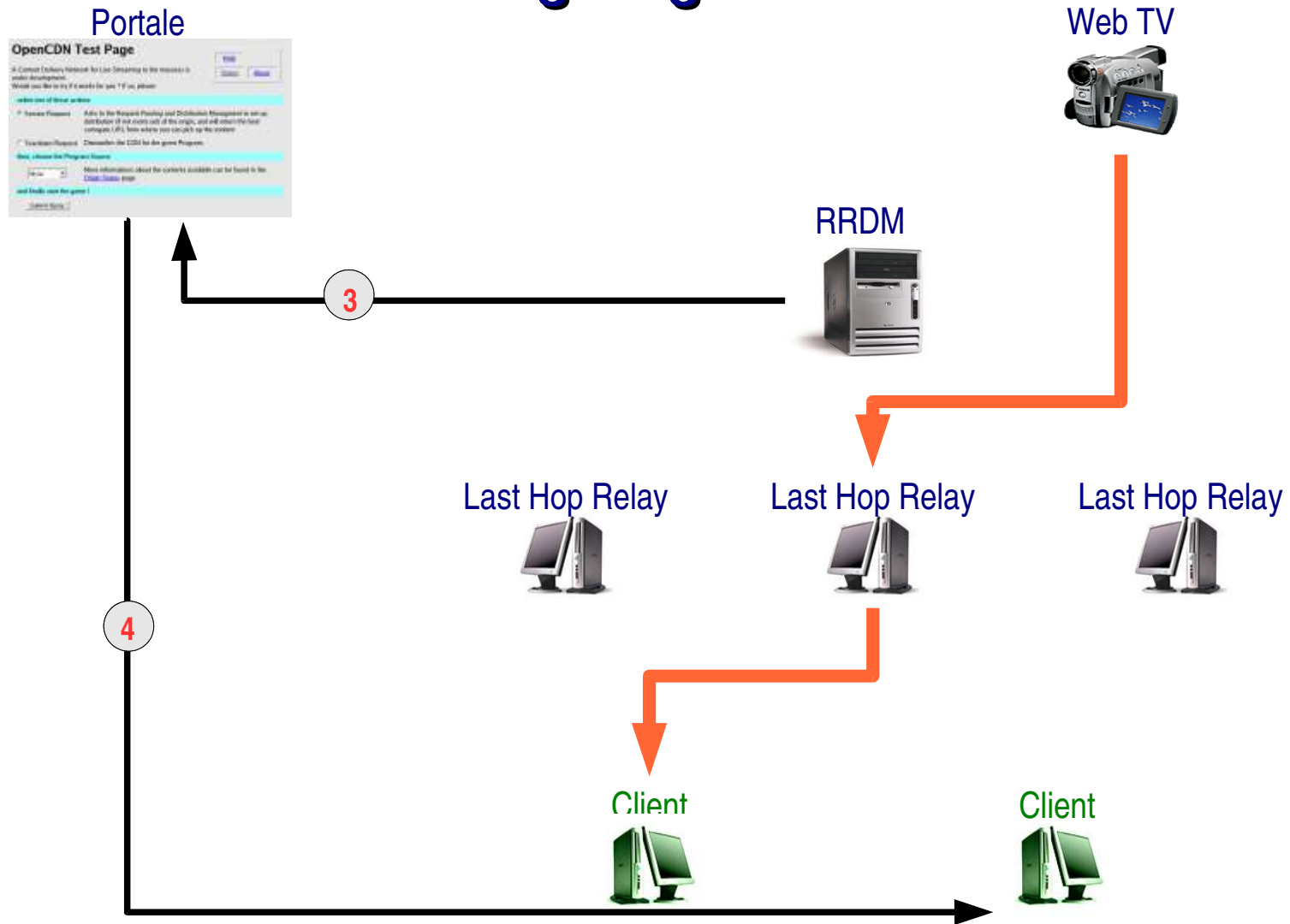
Lo streaming raggiunge il Client

Surrogato già attivo



Il Portale inoltra all'RRDM la richiesta del nuovo Client

Surrogato già attivo



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

Surrogato già attivo



Web TV



RRDM



Last Hop Relay



Last Hop Relay



Last Hop Relay



Client



Play

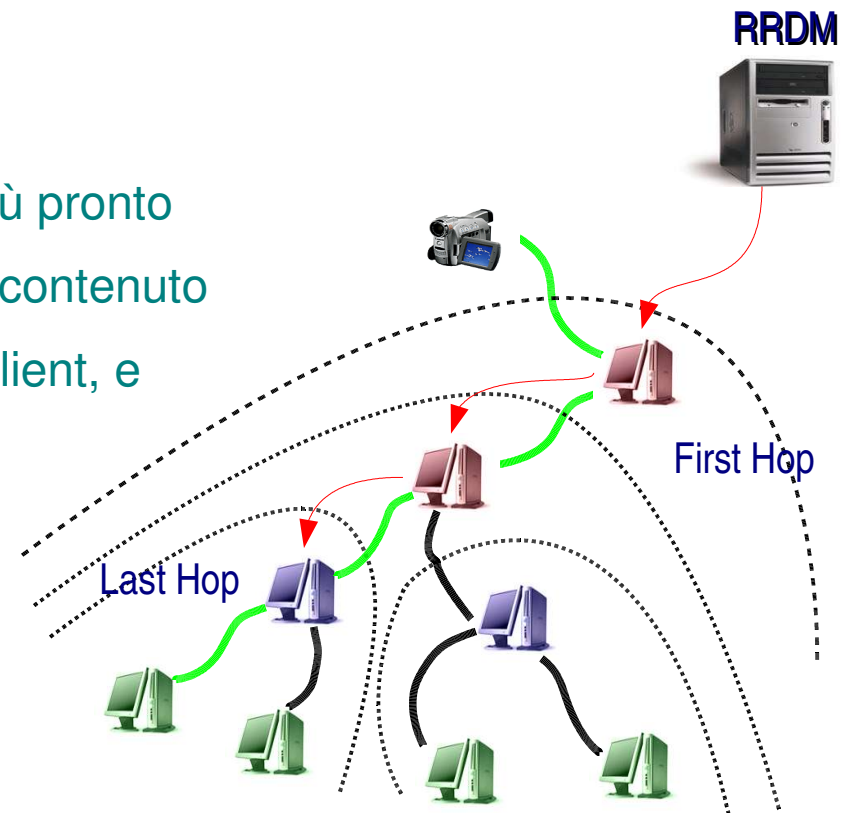
Client



Lo streaming raggiunge il secondo Client

Recursione e sondaggio

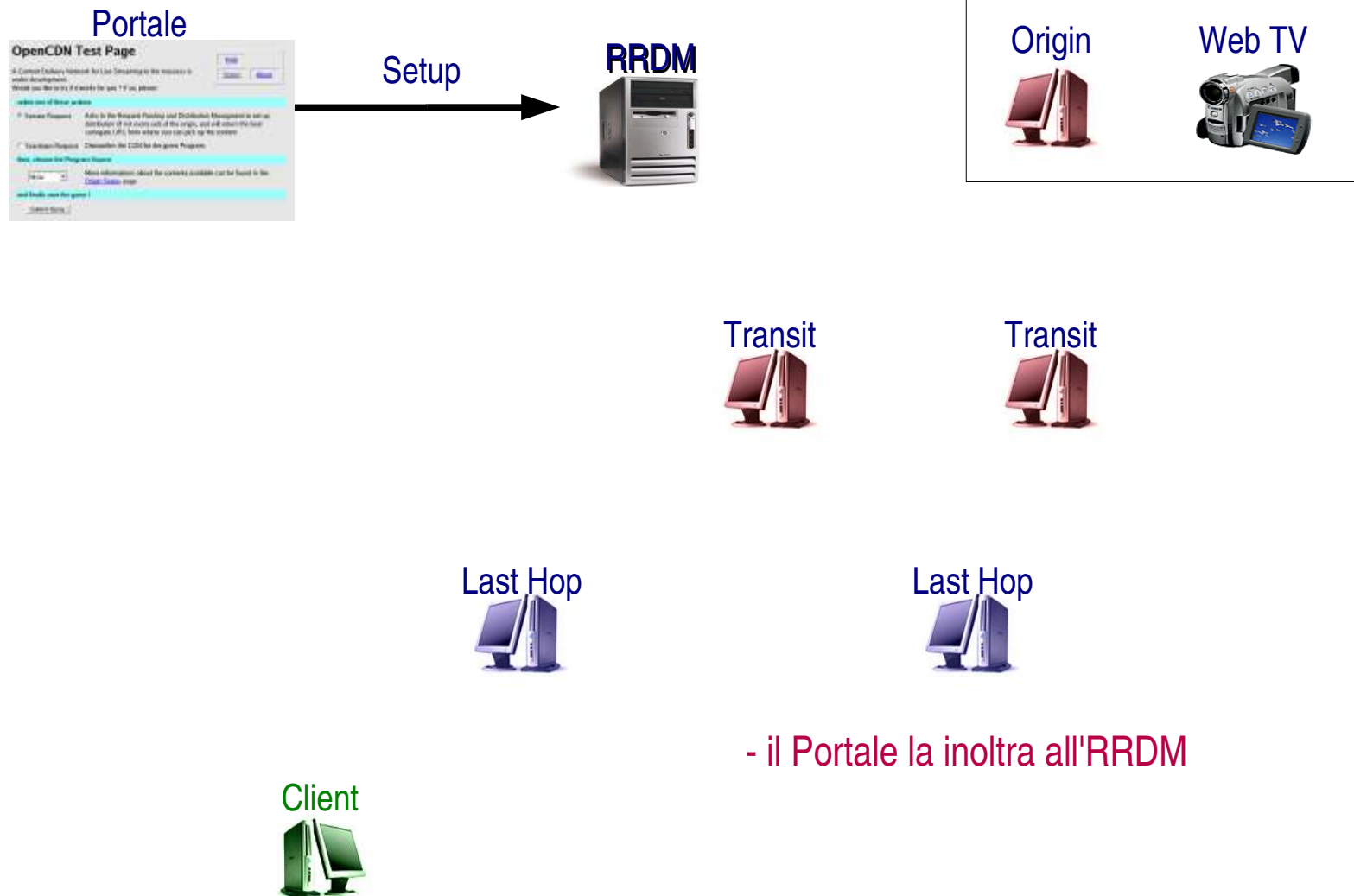
- RRDM contatta il FIRST HOP, comunicando i possibili discendenti
- per ogni HOP avviene un **sondaggio**, ed il vincitore riceve un nuovo elenco, ristretto, di possibili discendenti
- il sondaggio è svolto in **UDP**, e mira a
 - verificare l'operatività dei nodi
 - individuare il NextHop più prossimo e/o più pronto
 - interrogare se un nodo detiene tuttora un contenuto
 - conoscere l'**RTT** misurato tra LastHop e client, e confrontarlo tra diversi LastHop
- raggiunto il LAST HOP
 - l'esito risale lungo il path che si è creato
 - l'RRDM è notificato di tutti i nuovi surrogati attivi



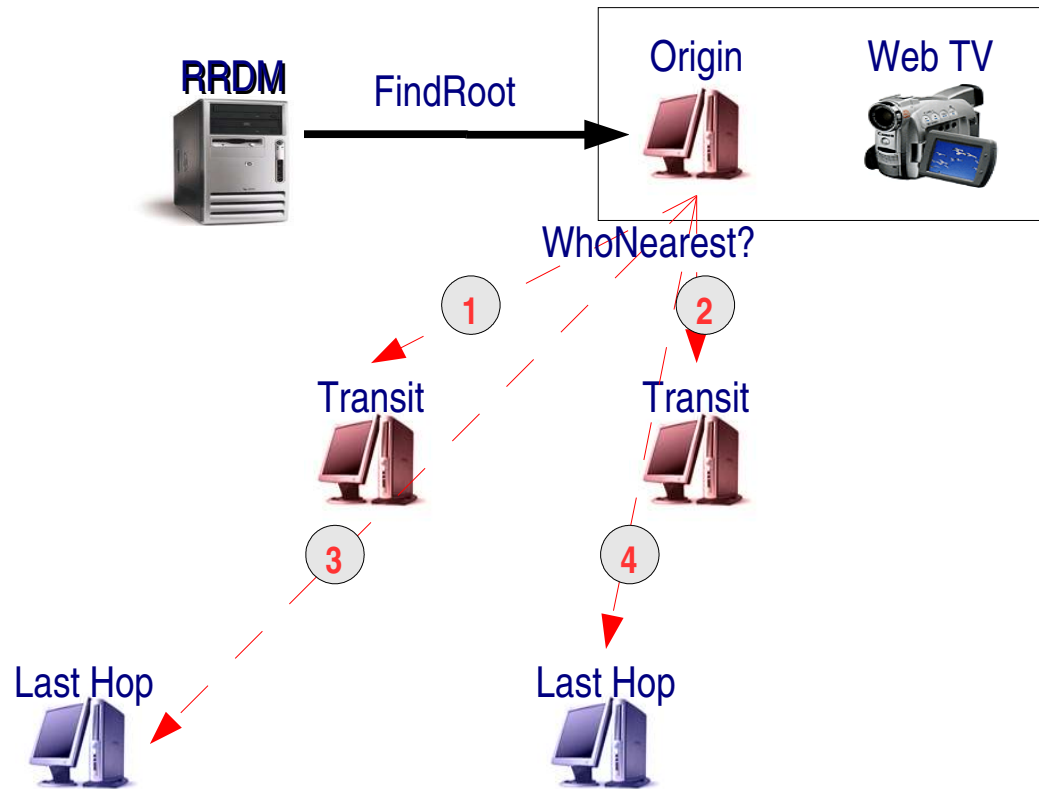
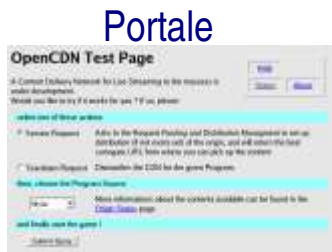
Topologia Gerarchica e Adattativa



Topologia Gerarchica e Adattativa

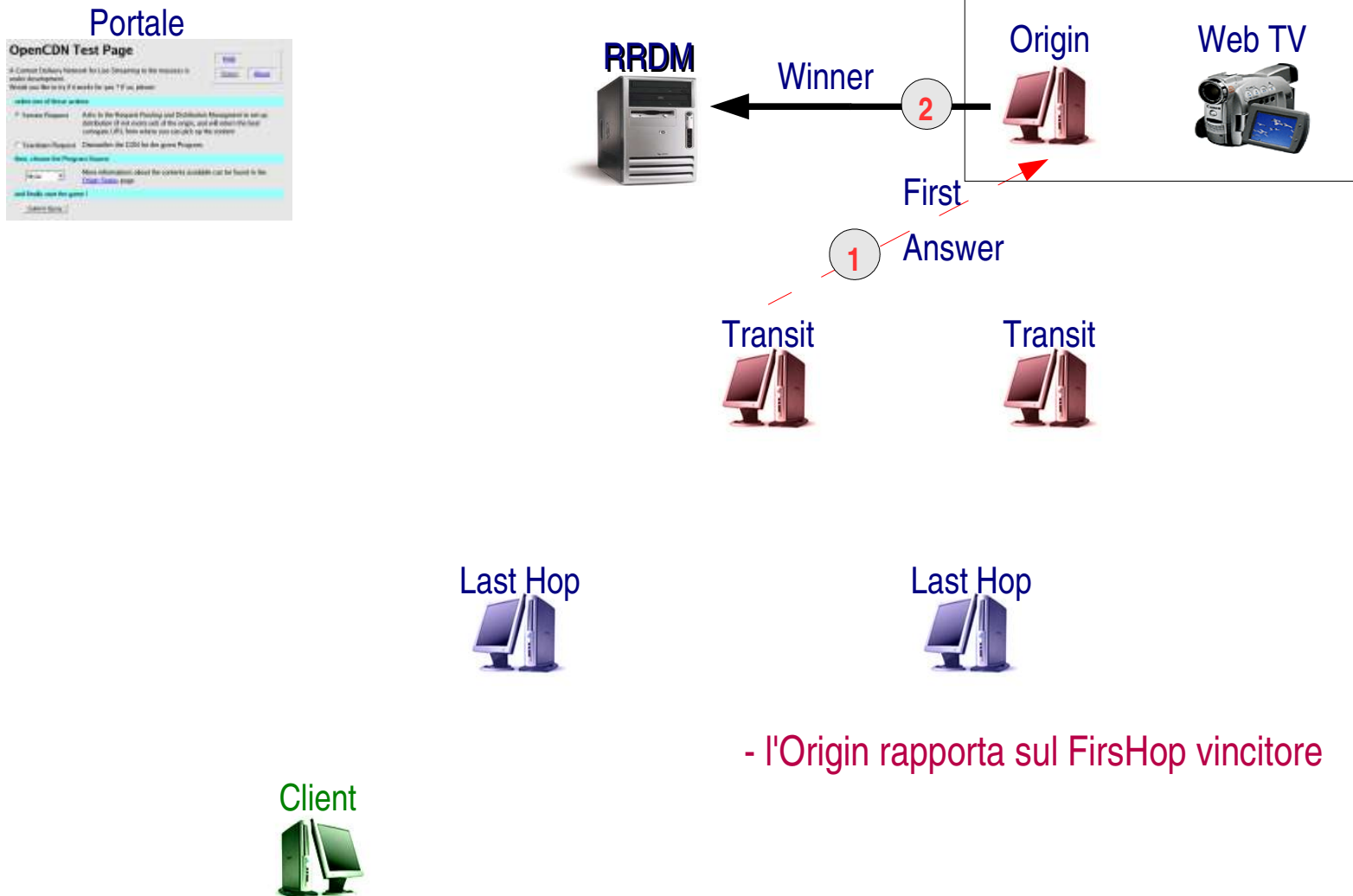


Topologia Gerarchica e Adattativa

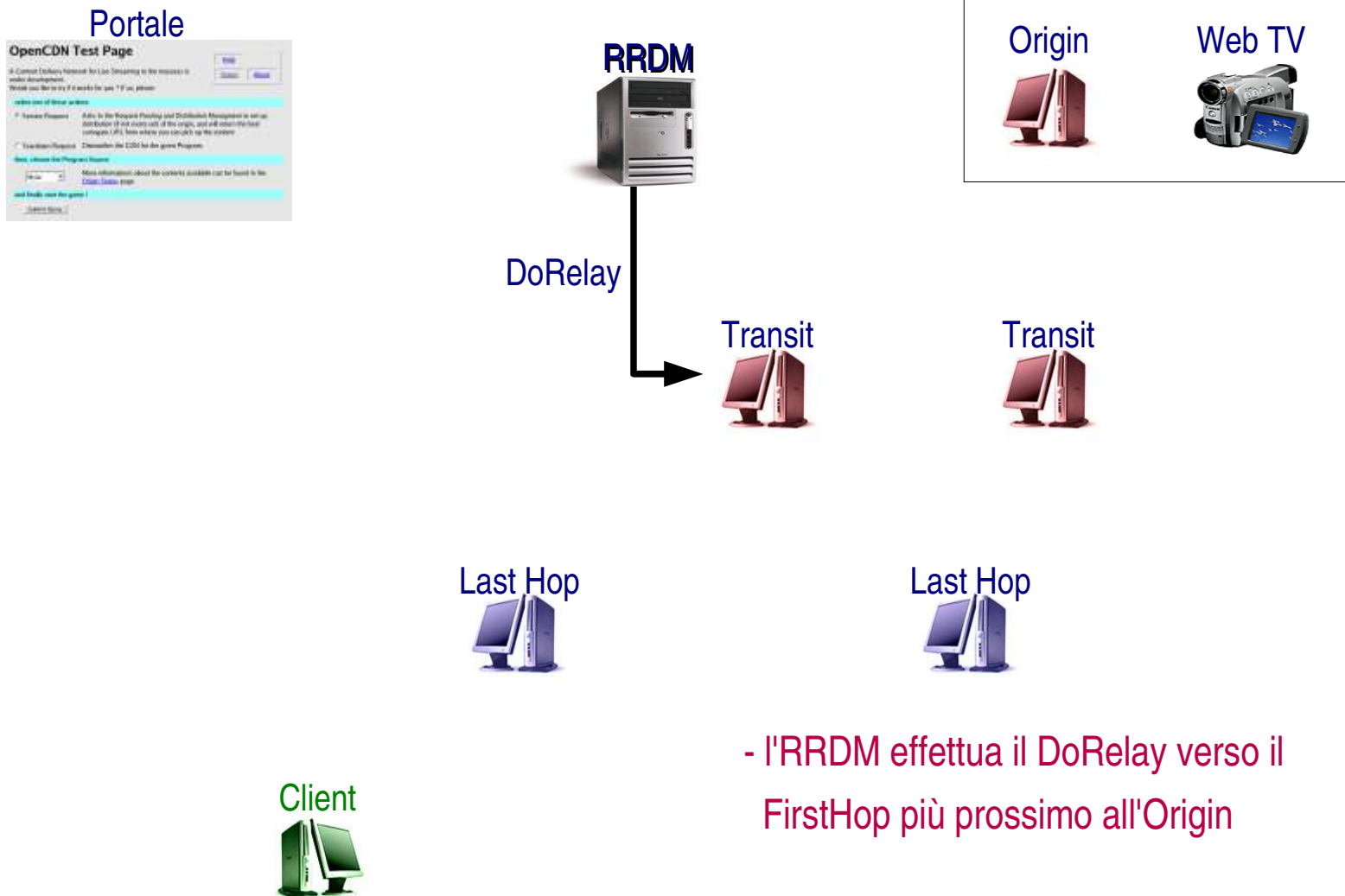


- l'Origin del contenuto sonda i nodi
- l'ordine rispetta le preferenze dei FootPrint dal meno al più specifico

Topologia Gerarchica e Adattativa



Topologia Gerarchica e Adattativa



- l'RRDM effettua il DoRelay verso il FirstHop più prossimo all'Origin

Topologia Gerarchica e Adattativa



Portale



RRDM



Origin

Web TV



Transit



Transit

AreYouThere?

1

2



Last Hop



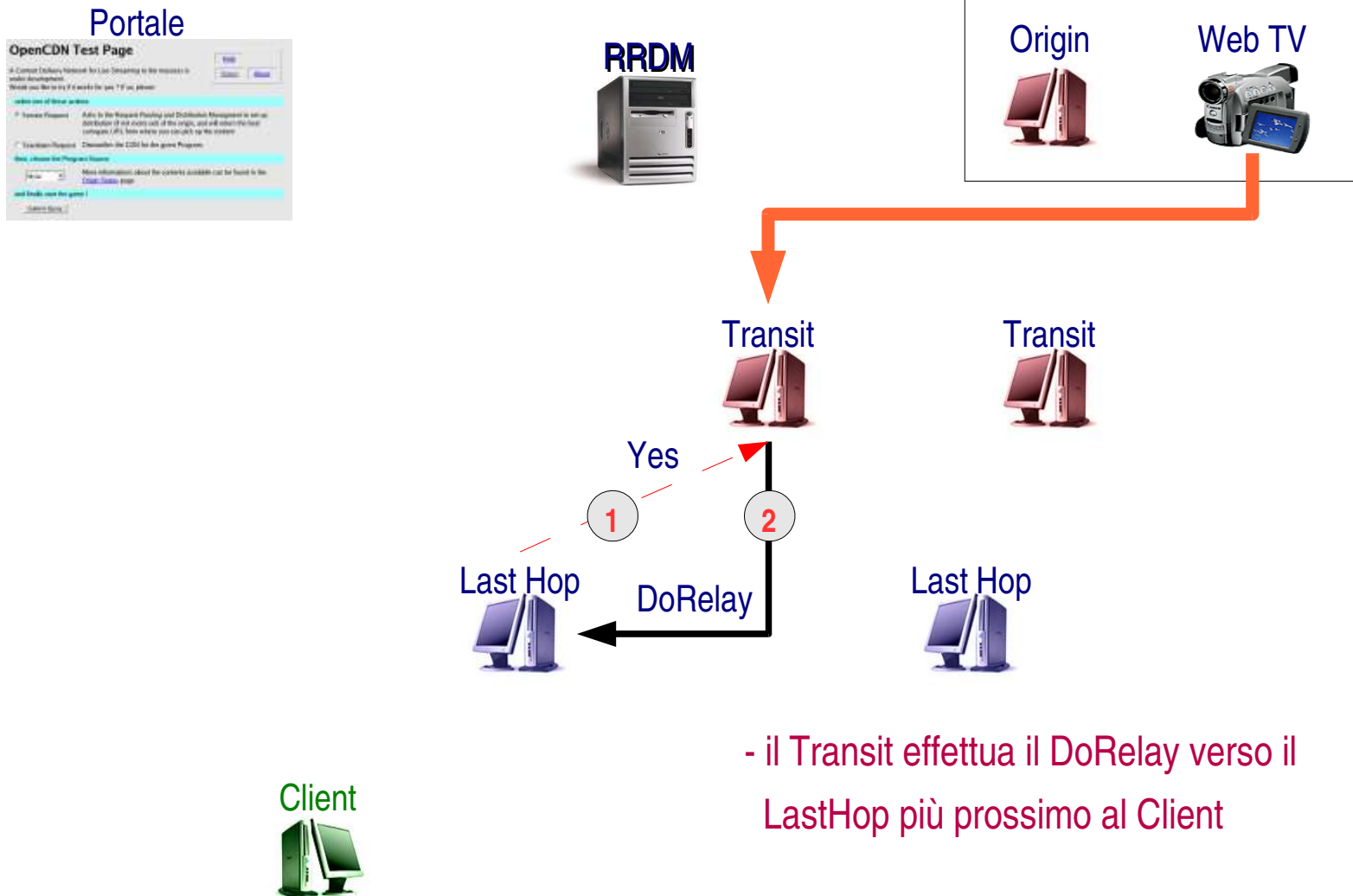
Last Hop



Client

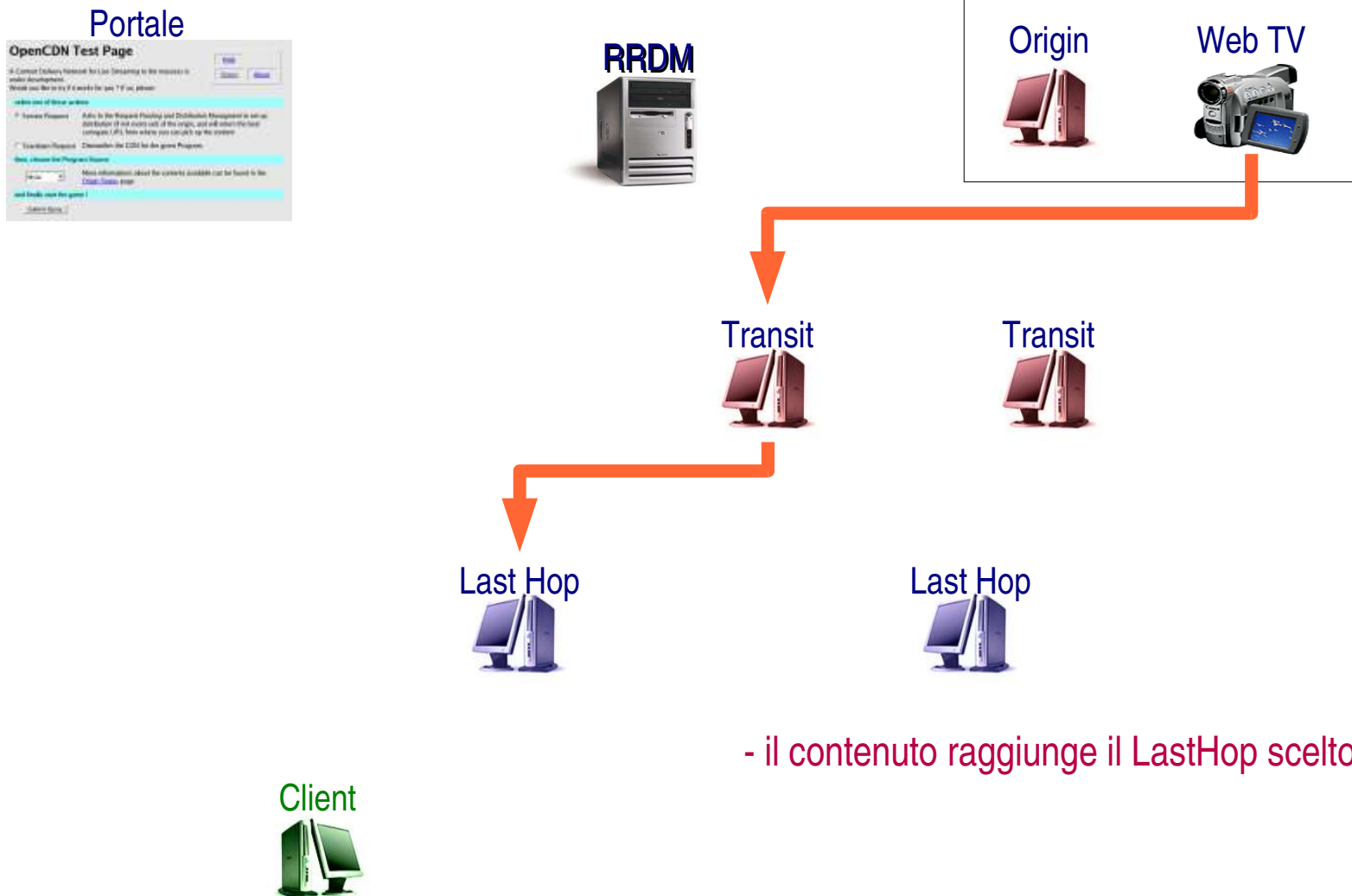
- il contenuto raggiunge il Transit
- i LastHop sono sondati per ordine di FootPrint dal più al meno specifico

Topologia Gerarchica e Adattativa

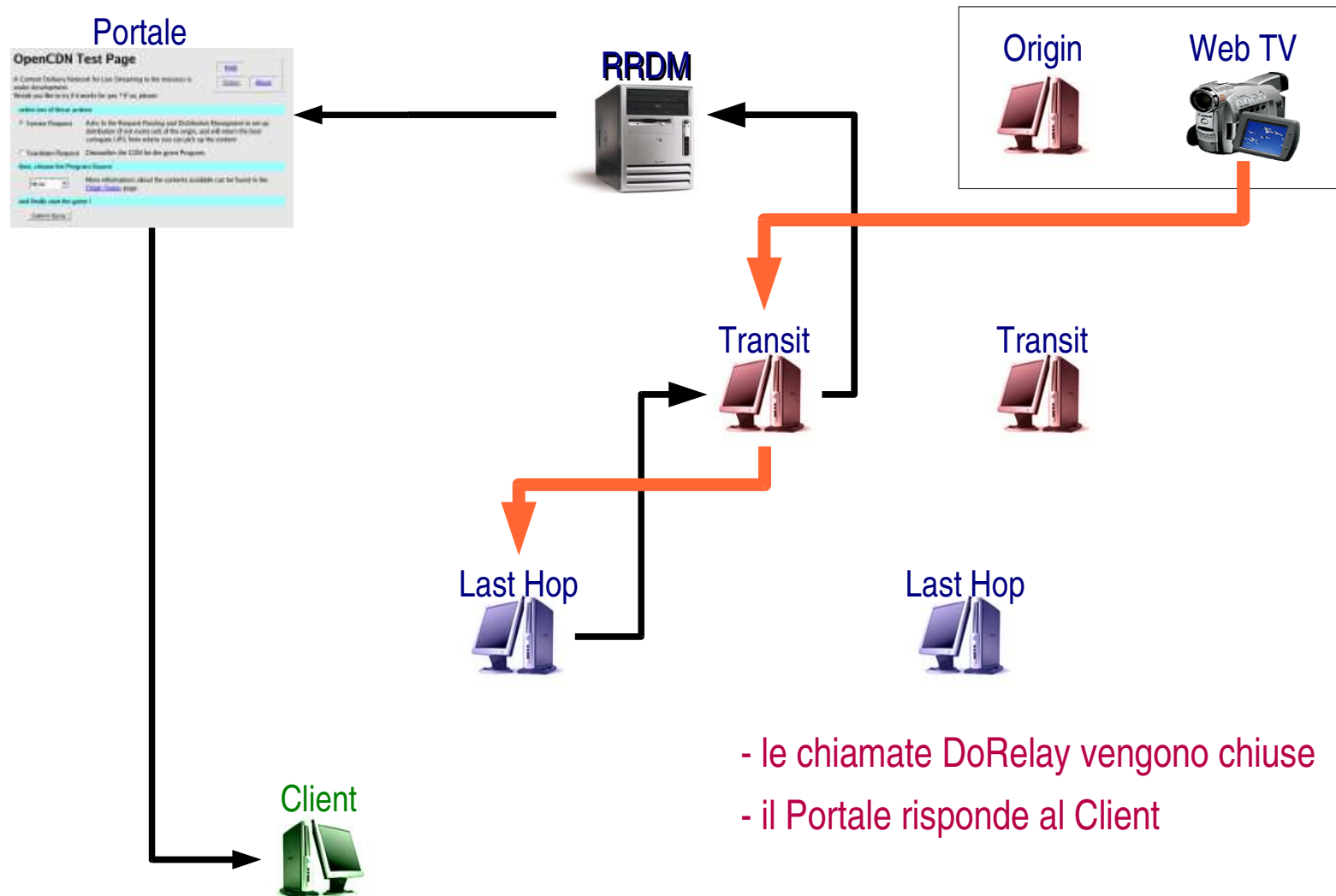


- il Transit effettua il DoRelay verso il LastHop più prossimo al Client

Topologia Gerarchica e Adattativa

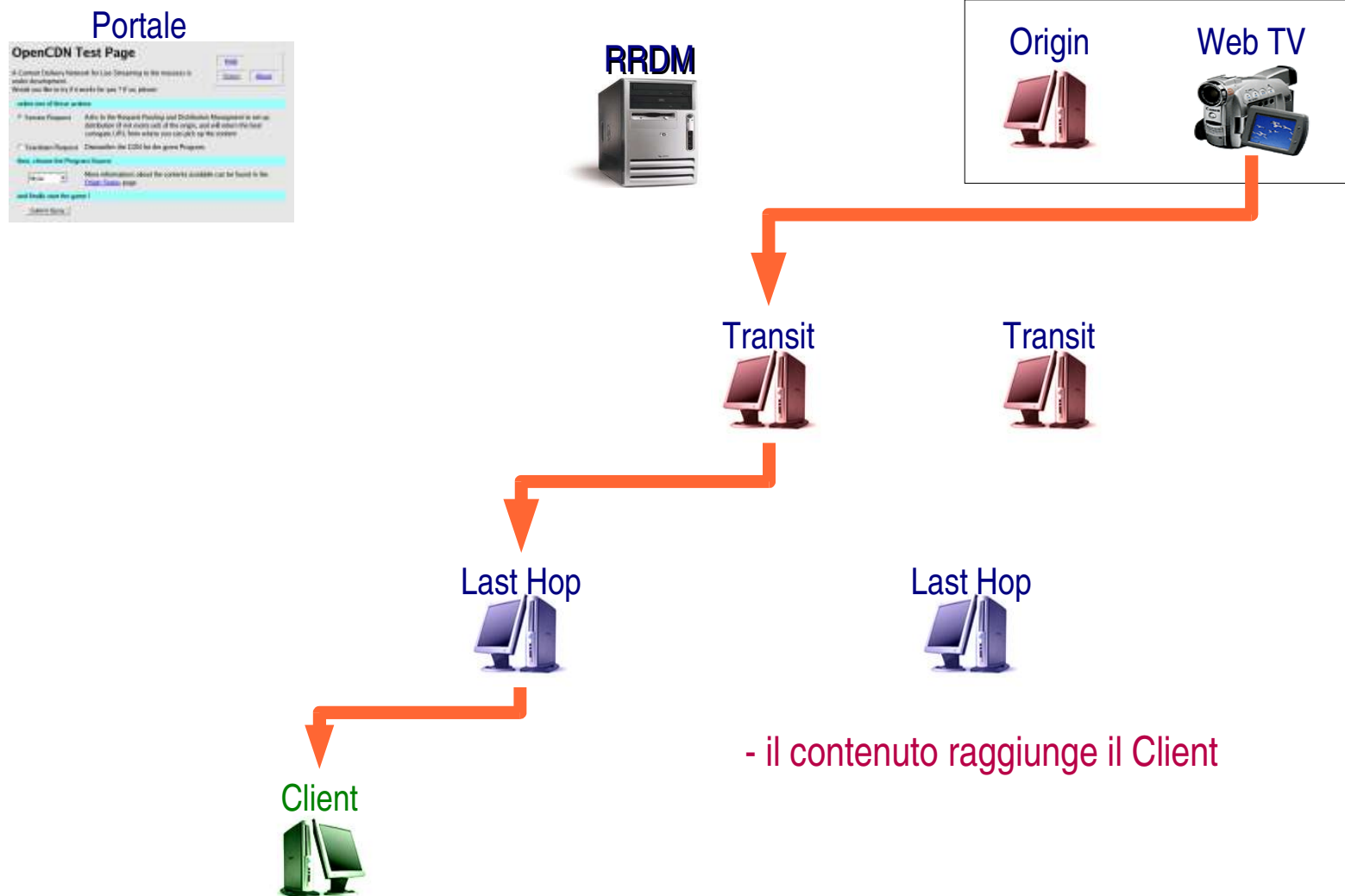


Topologia Gerarchica e Adattativa

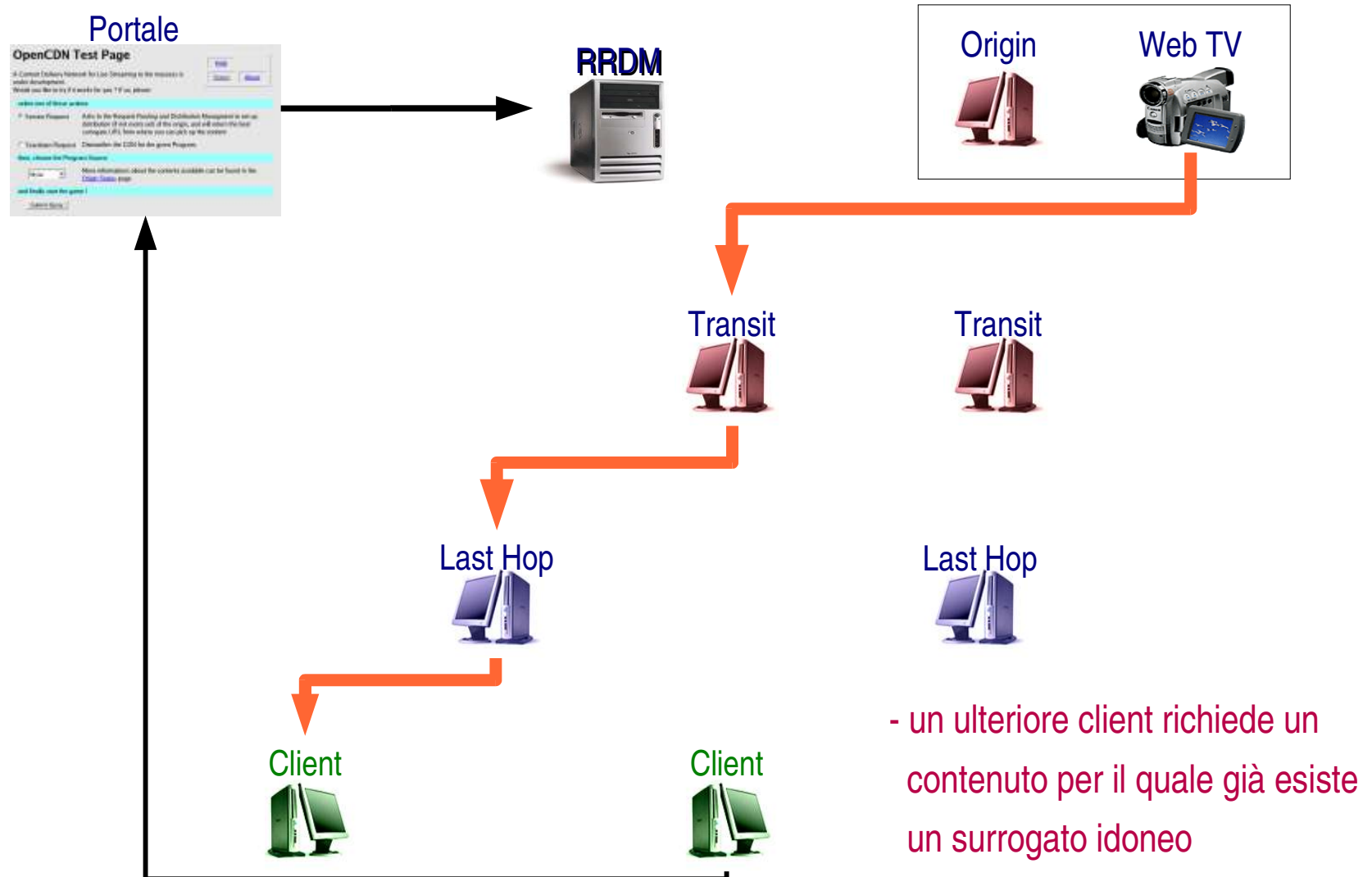


- le chiamate DoRelay vengono chiuse
- il Portale risponde al Client

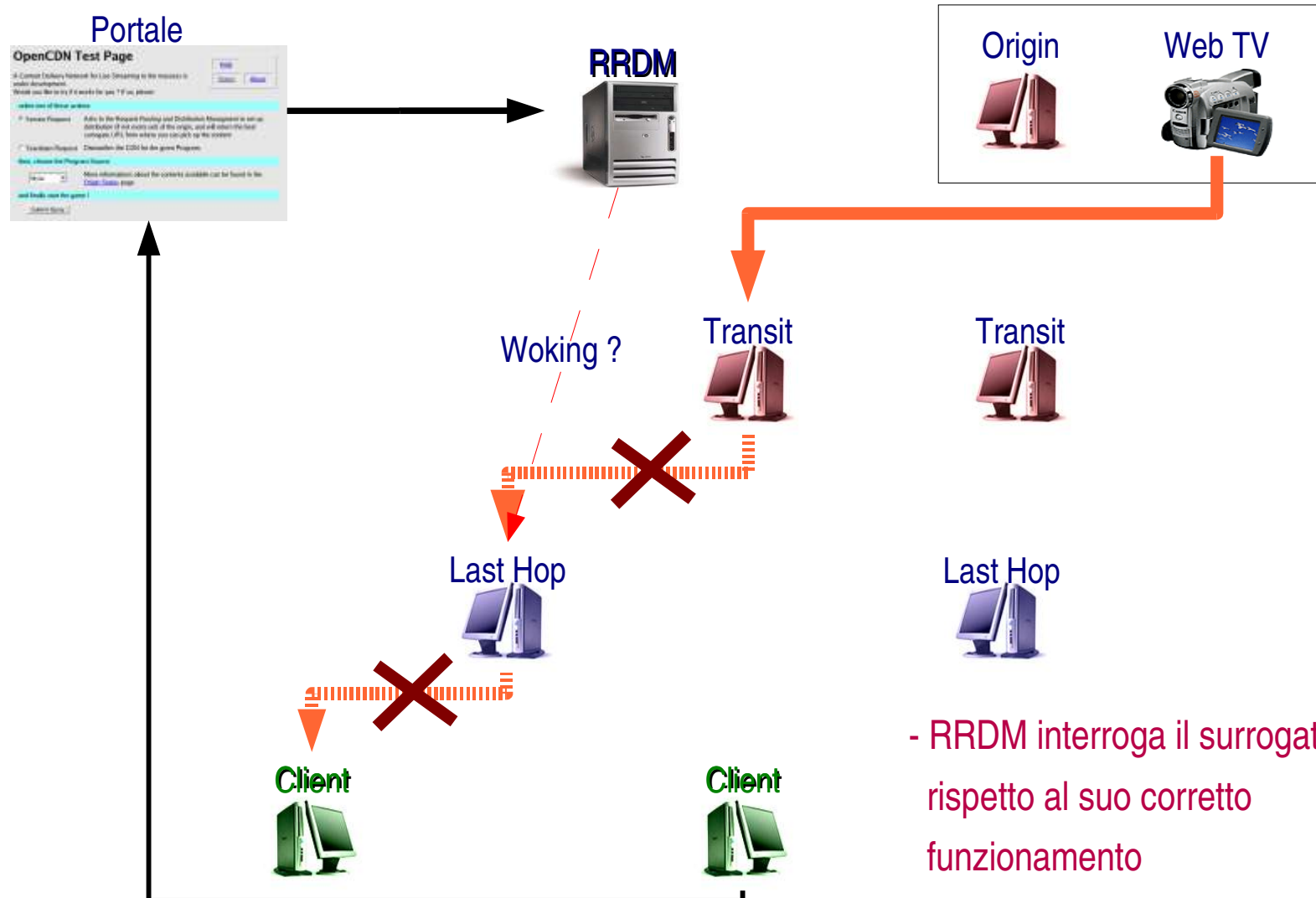
Topologia Gerarchica e Adattativa



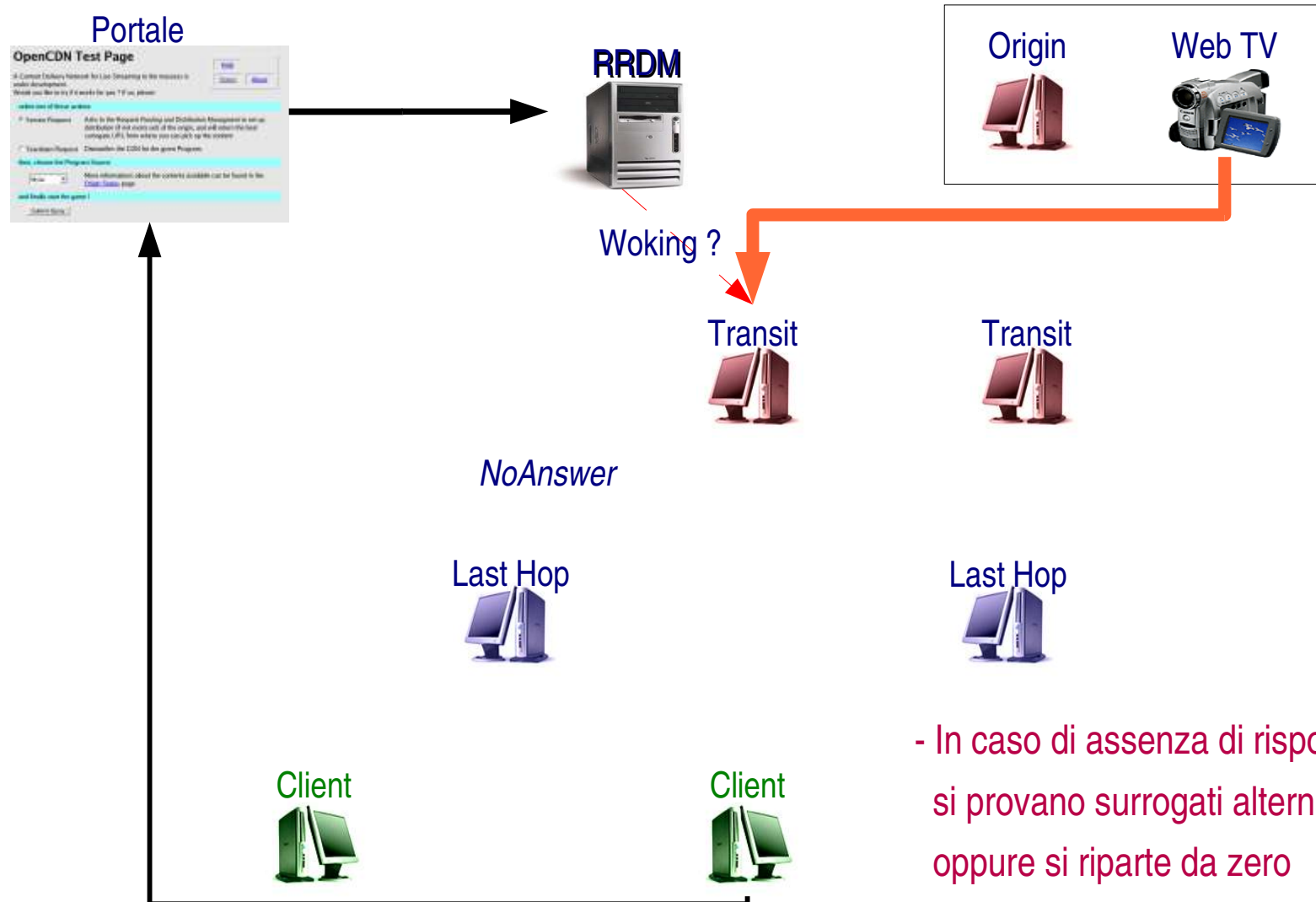
Resilienza a Failure e Outage



Resilienza a Failure e Outage

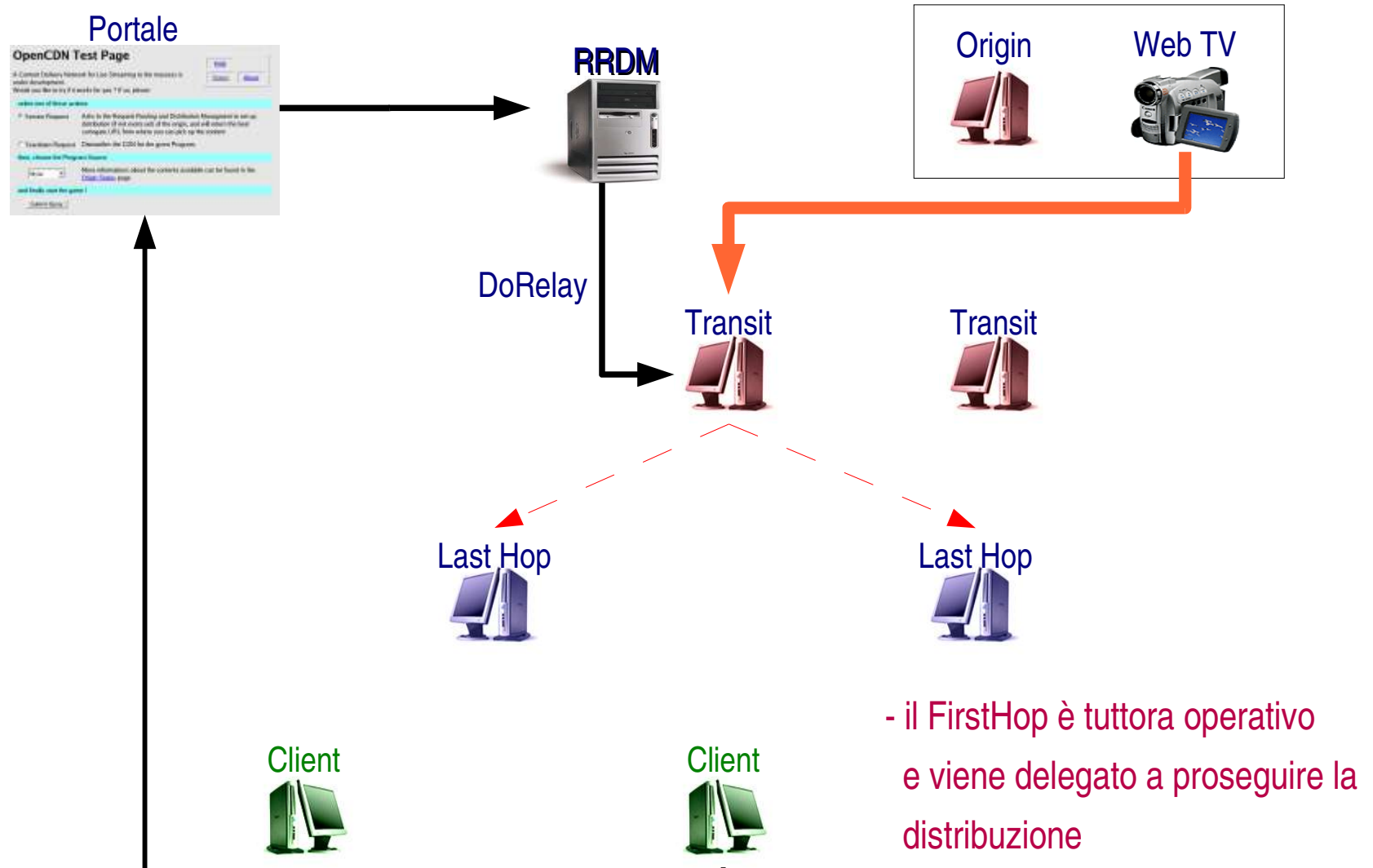


Resilienza a Failure e Outage

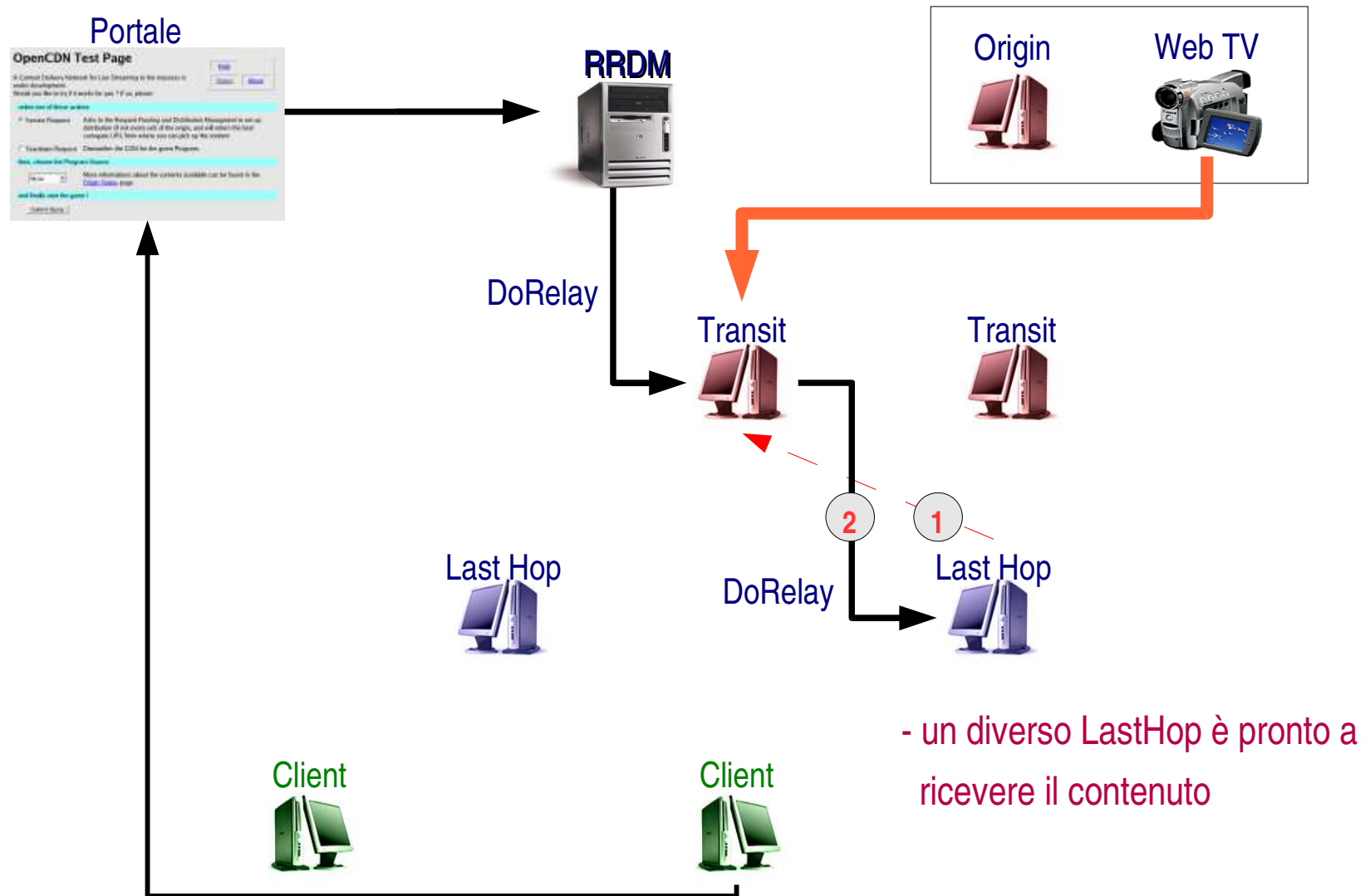


- In caso di assenza di risposta si provano surrogati alternativi oppure si riparte da zero

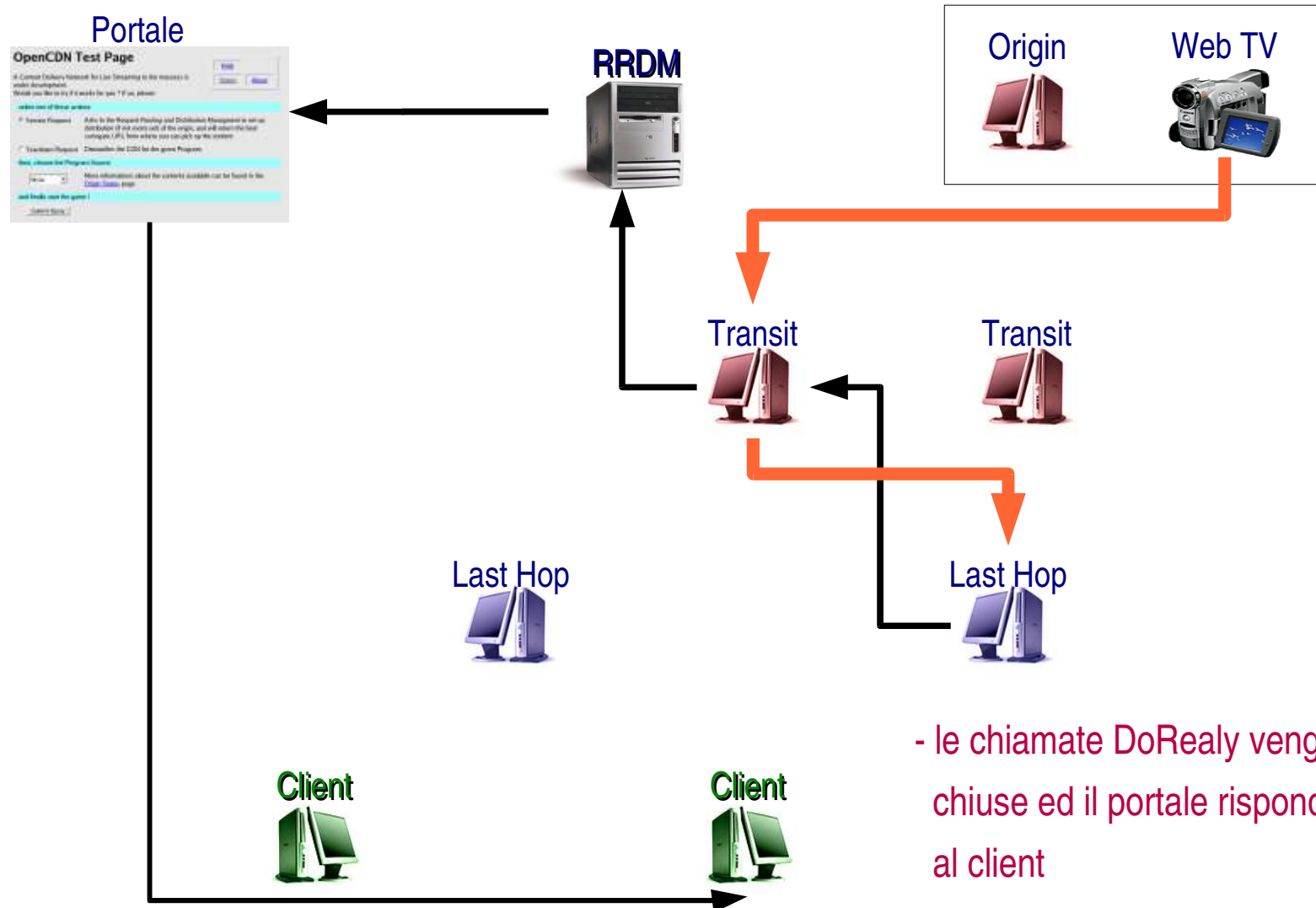
Resilienza a Failure e Outage



Resilienza a Failure e Outage

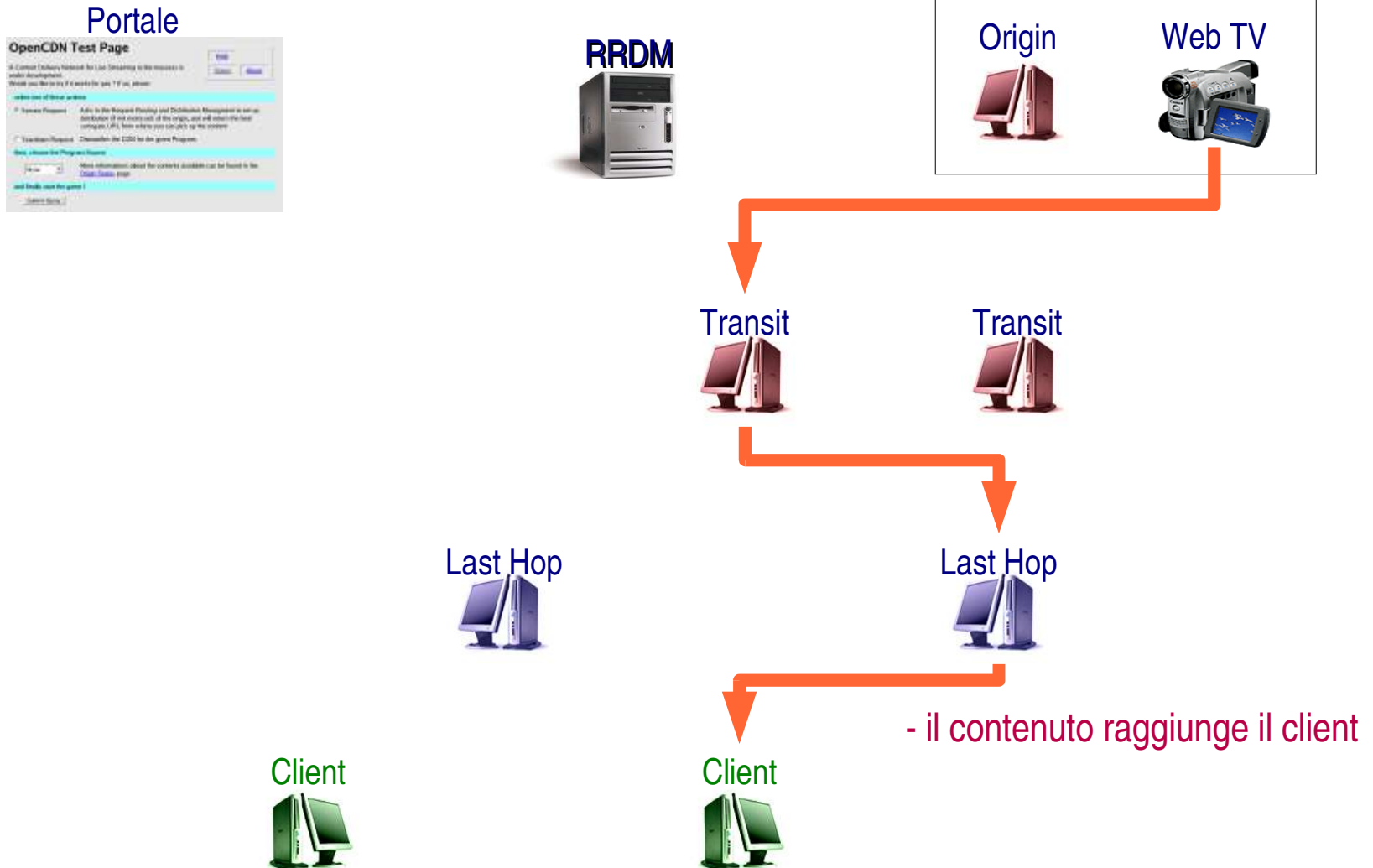


Resilienza a Failure e Outage

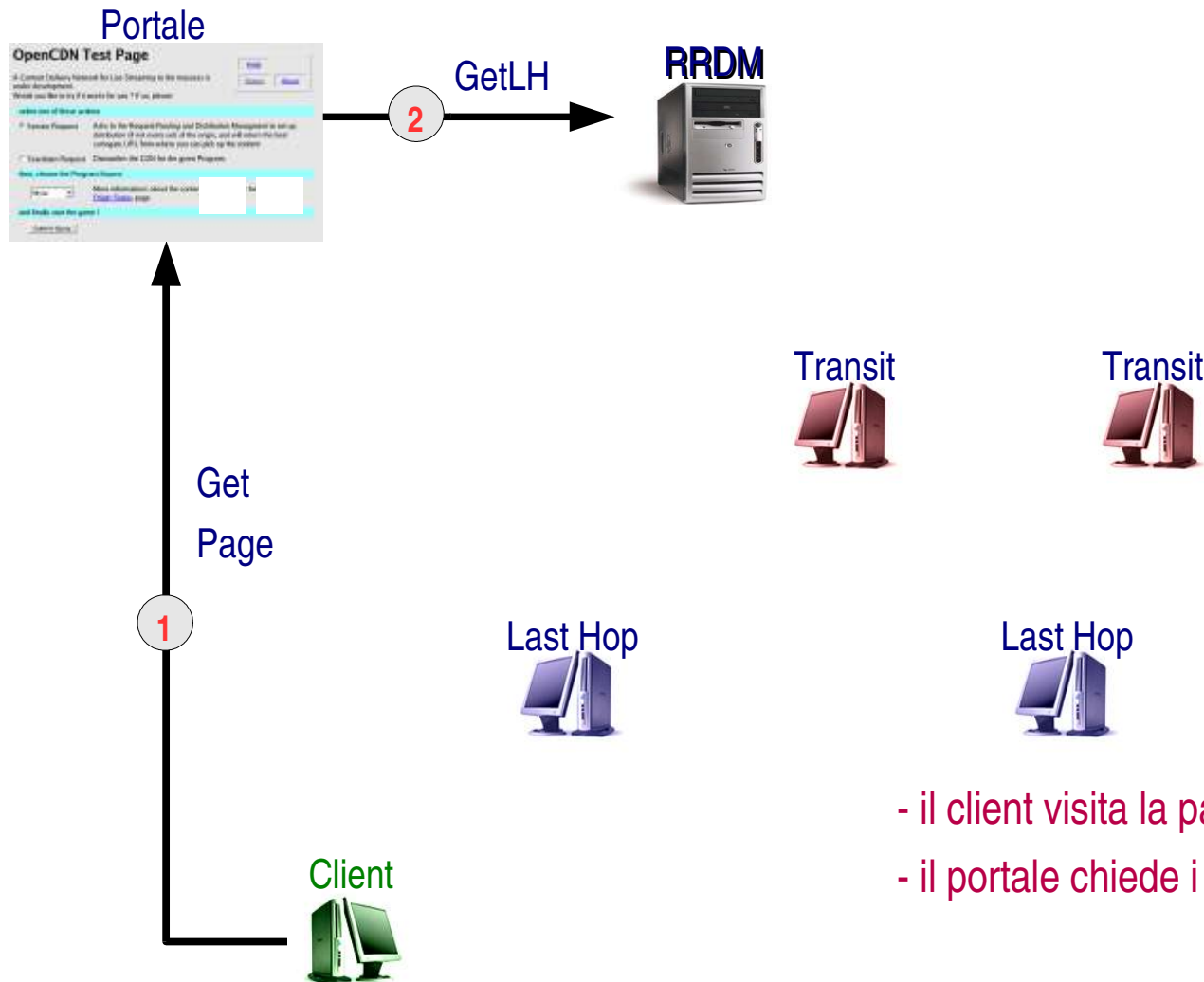


- le chiamate DoRealy vengono chiuse ed il portale risponde al client

Resilienza a Failure e Outage

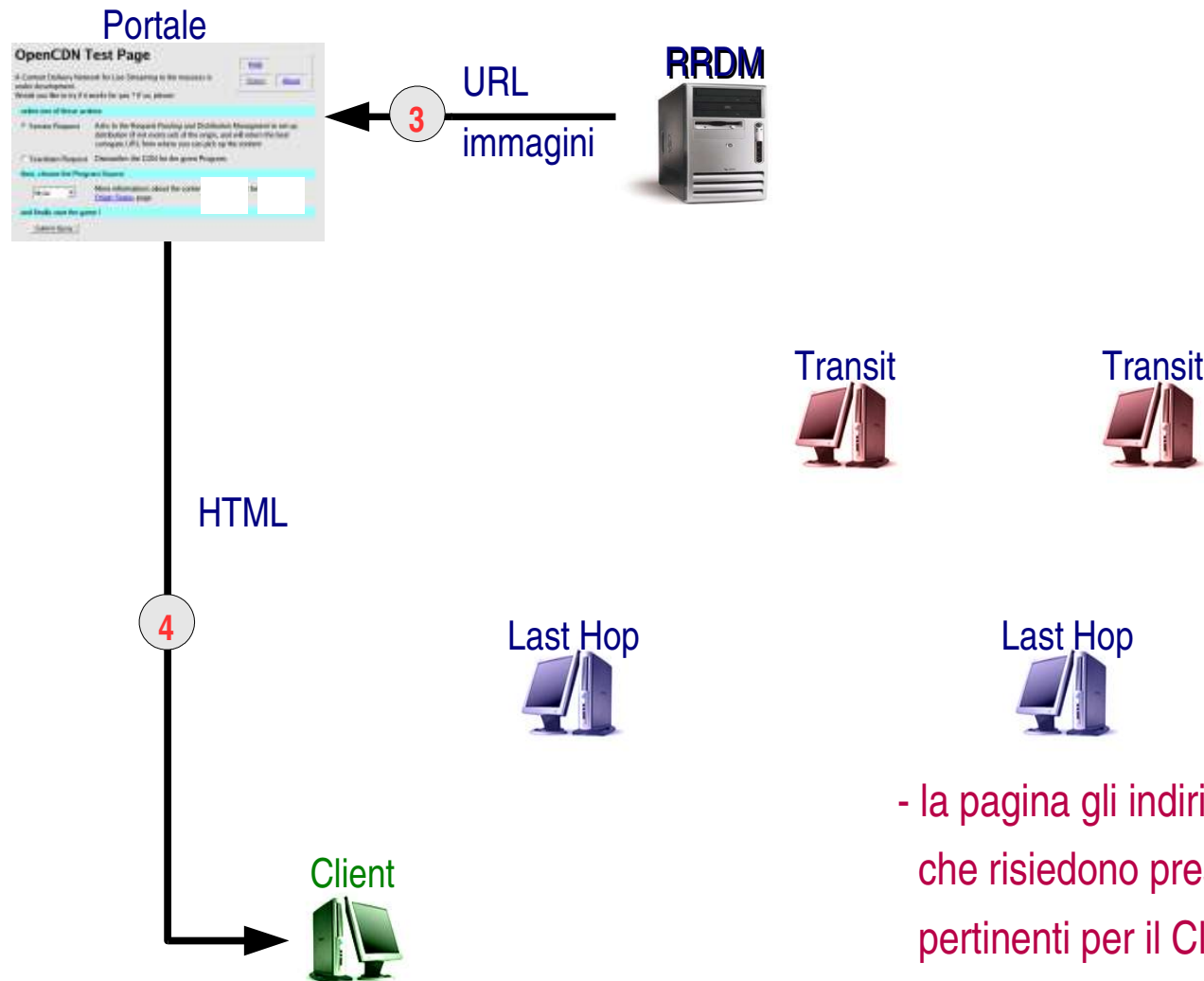


Scelta LastHop – metodo *PYTE*



- il client visita la pagina di richiesta
- il portale chiede i LH disponibili

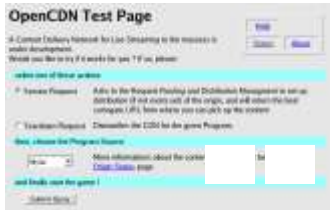
Localizzazione LastHop – metodo *PYTE*



- la pagina gli indirizzi delle immagini che risiedono presso i LastHop pertinenti per il Client

Localizzazione LastHop – metodo *PYTE*

Portale



RRDM



Transit



Transit



Last Hop



Last Hop



Get

Get

Client



- il client richiede le immagini ai LastHop

Localizzazione LastHop – metodo *PYTE*

Portale



RRDM



Transit



Transit



Last Hop



save RTT

Last Hop



save RTT

LH1

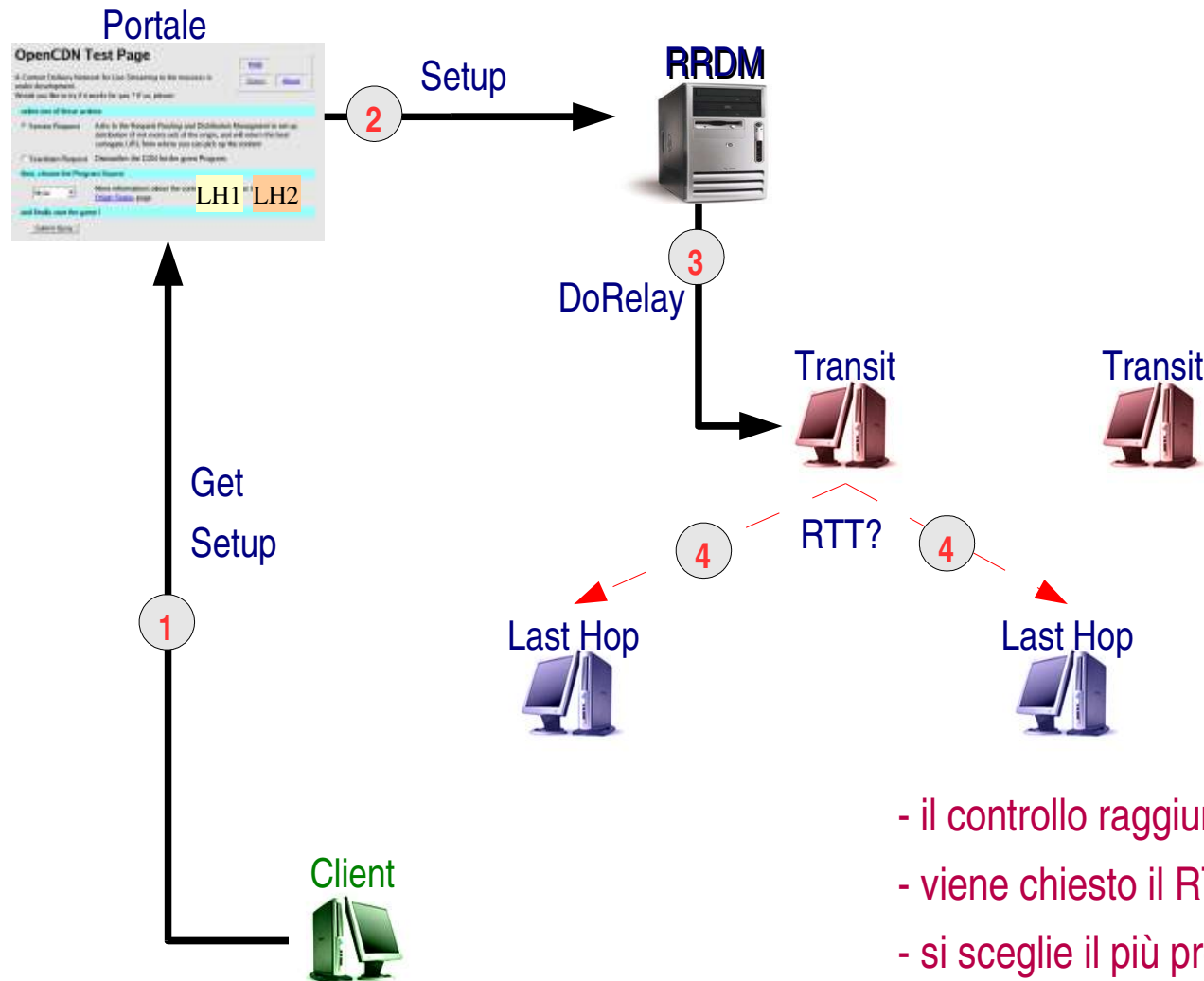
LH2

Client



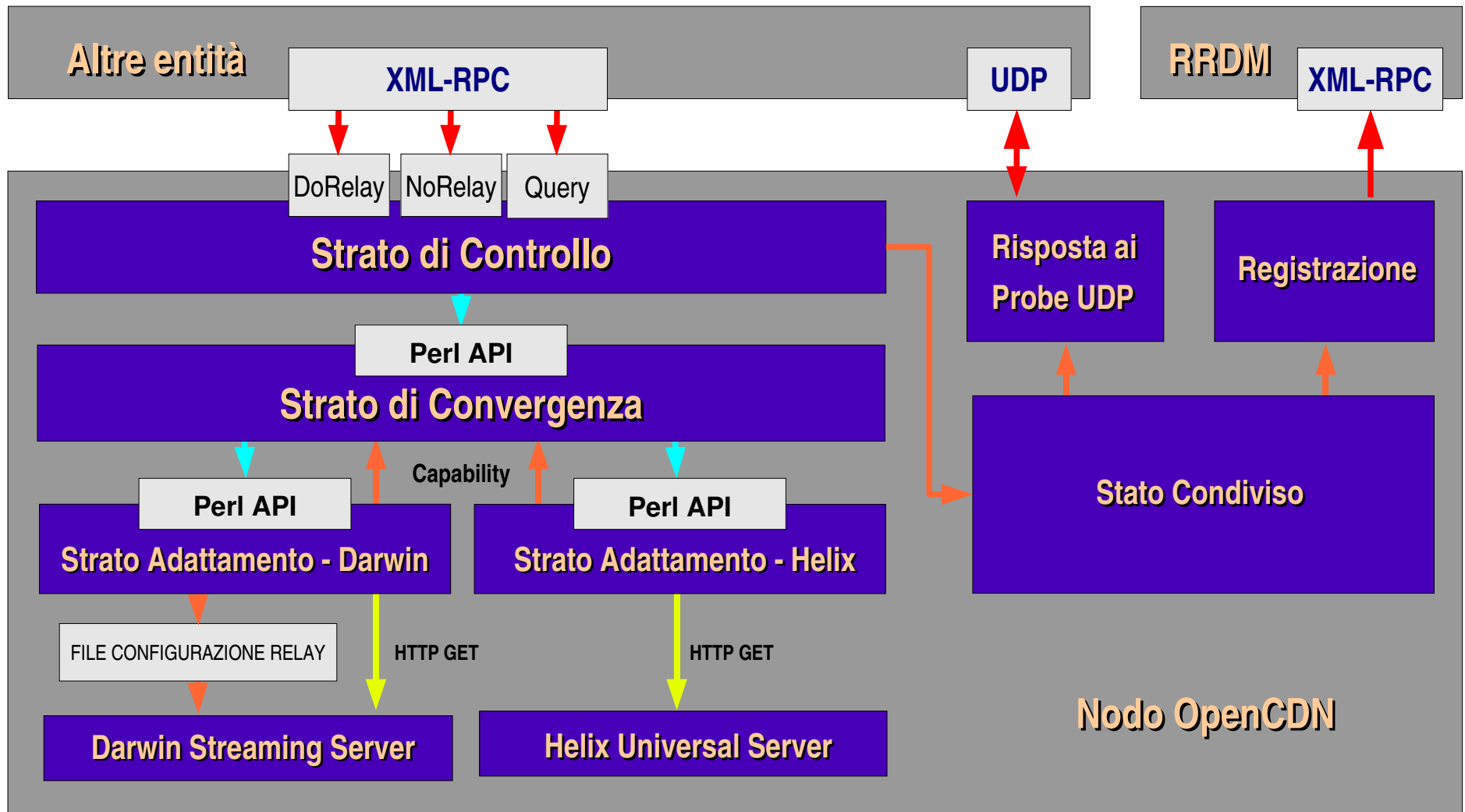
- nella pagina compaiono le immagini
- i LastHop determinano il Round Trip Time nei confronti del Client

Localizzazione LastHop – metodo *PYTE*



- il controllo raggiunge l'ultimo Transit
- viene chiesto il RTT verso quel Client
- si sceglie il più prossimo al Client

Architettura di un Nodo OpenCDN



Riassunto delle caratteristiche

- Estensibilità:
 - nuove Origin e Nodi aderiscono in modo autonomo
 - supporto di contenuti e tecnologie differenti
 - Filosofia agnostica:
 - inconsapevolezza della natura e della tecnologia del dispositivo di replica
 - Controllo centralizzato e accesso Web:
 - facilità di amministrazione e fruizione
 - Resilienza: l'architettura resiste a
 - partizionamenti della rete
 - disattivazione dei nodi
 - reboot dell'entità di controllo
 - Topologia ottimizzata
 - FirstHop vicino all'Origin
 - LastHop vicino al Client
 - Autenticazione
 - Per aderire occorre conoscere un segreto condiviso
-



Utilizzi potenziali e Attesi

- Dibattiti socio politici
 - Eventi culturali e sportivi
 - Musica dal vivo, teatro, danza
- } piccole audiences per i canali Sat e TV,
ma troppo ampie per lo streaming
tradizionale
- Home netcasting *(datemi un'ADSL e vi predicherò al mondo!)*
 - Teledidattica *(studenti a casa => LastHop presso gli ISP)*
 - Streaming di videoconferenza *(da cui nascerà il talk show internet)*
 - Localizzazione della cache più prossima
 - l'utilizzo di OpenCDN per la distribuzione di contenuti generici è possibile solo dopo lo sviluppo di un appropriato strato di adattamento per il caching
 - in un progetto OpenSource, gli interessati possono iniziare a rimboccarsi le maniche ;-)

Requisiti e Disponibilità

- Per aderire ad OpenCDN occorre:
 - una Linux box (*... o compatibile*)
 - uno Streaming Server (*Helix di Real, o Darwin di Apple*)
 - connettività commisurata alla audience che si intende coprire
 - Per apportare contenuti Live Streaming occorre:
 - un encoder (*Real producer, QT broadcaster, MPEG4IP live...*)
 - un minimo di struttura di produzione (*videocamera, microfoni, mixer*)
 - una scheda di acquisizione (*Pinnacle, Osprey...*)
 - connettività ADSL
 - E' operativo un Hub nazionale con 100 Mbit in uscita, grazie alla disponibilità di GARR e CILEA
 - Chiunque è benvenuto alla sperimentazione, e può contribuire con contenuti, LastHop per la propria rete, o Transit sia nazionali che internazionali. Rivolgersi presso <http://opencdn.sourceforge.net/>
-