

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni



P4P - Il protocollo ALTO: informazioni di rete finalizzate al miglioramento del processo di selezione dei peer

Relatore: Prof. Paolo Giacomazzi

Correlatore: Ing. Alessandro Poli

Tesi di Laurea di:
Mariano MOLENTINO
Matricola 750438

Anno Accademico 2012/2013

Indice

Capitolo 1 – Introduzione.....	6
Capitolo 2 – Il Protocollo ALTO.....	8
2.1 Architetture P4P.....	8
2.1.1 Ottimizzazione.....	9
2.2 ALTO Protocol.....	9
2.3 Information Resource.....	12
2.4 P2P Client: App Tracker.....	12
2.5 Scelta degli indirizzi.....	14
2.6 ALTO Informati.....	15
2.7 Linee guida A.....	16
2.7.1 Come opera ALTO.....	16
2.8 Sicurezza.....	17
2.9 ALTO Server Discovery.....	18
2.9.1 Condizioni di utilizzo.....	20
2.9.2 Ricerca indirizzo dell’ALTO Server.....	20
2.9.3 U-NAPTR.....	21
2.10 Codifica del domain name.....	22
2.10.1 Applicabilità.....	23
2.10.2 Third Part.....	23

2.11 Attacchi.....	24
2.12 RFC (Request for comments).....	24
2.13 ALTO Client Protocol.....	25
2.13.1 File Sharing.....	27
2.13.2 Live Media Straming.....	27
2.13.3 Real-Time Communications.....	27
2.13.4 Distributed Hash Tables.....	27
2.14 ALTO Service.....	27
2.15 Bandwidth.....	28
2.15.1 Colli di bottiglia.....	33
2.16 ALTO Gerarchico.....	34
2.17 ALTO con CDN (content distribution networks).....	36
2.18 Third Part.....	38
2.19 Scambi a Banda Larga.....	39
2.20 Centro Risorse.....	41
2.21 Caching ALTO.....	42
2.22 ALTO Abbonamento.....	45
2.23 Notifiche ALTO.....	45
2.24 Pianificazione dei costi ALTO.....	46
2.24.1 ALTO Multi Cost.....	47
2.25 Scelta Endpoint.....	48

2.26 Aggiornamenti incrementali.....	51
Capitolo 3 – Altre Architetture.....	52
3.1 Ono Project.....	52
3.2 CDN.....	52
3.3 DECADE.....	53
3.3.1 DECADE Discovery Server.....	53
3.3.2 Server Discovery DECADE con DHCP.....	54
3.3.3 DECADE vs CDN.....	55
3.3.4 Confronto di DECADE con CDN.....	55
3.3.5 Integrazione ALTO e DECADE.....	56
3.4 SDN.....	60
3.4.1 Interazione con ALTO.....	61
Capitolo 4 – Confronti.....	62
4.1 Confronto ALTO – altre architetture.....	62
4.1.1 Peer selection.....	63
4.1.2 Sicurezza.....	63
4.1.3 Privacy.....	64
4.1.4 Riduzione dei costi.....	64
4.1.5 Miglioramento del traffico.....	64
4.1.6 QoE.....	65
4.1.7 Memoria.....	65

4.2 Tabella riassuntiva.....	66
Capitolo 5 – Conclusioni.....	67
Capitolo 6 – Bibliografia.....	69

Capitolo 1

Introduzione

In questo lavoro di tesi ci focalizzeremo su come avviene il processo di selezione dei peer principalmente per il Protocollo ALTO che utilizza il P4P per lo scambio di dati su internet.

Parte del traffico internet è generato dal P2P, utilizzato per la condivisione dei file, per le comunicazioni in tempo reale e il live media streaming. Le applicazioni P2P scambiano grandi quantità dati e devono scegliere uno o più candidati da una selezione di peer che offrono la stessa risorsa. Uno dei vantaggi dei sistemi P2P viene dalla ridondanza delle risorse disponibili.

Non sempre l'applicazione prevede una disposizione ottimale dei peer, pertanto fornire maggiori informazioni per la selezione dei peer può migliorare le prestazioni del P2P e ridurre i costi. Si sono pertanto studiate migliorie, quali il P4P: Proactive Network Provider for P2P.

Le applicazioni P2P non hanno conoscenza della rete underlay il che comporta un uso inefficiente delle risorse di rete. Il P4P vuole essere una soluzione a questo problema e consiste nell'uso di tecniche e di architetture che hanno l'obiettivo di sfruttare le conoscenze che la rete overlay ha delle reti underlay.

In P2P il traffico segue link multipli e la maggior parte del traffico è inter-domain, ovvero esce dalla rete del provider; ciò comporta alti costi. Alcuni sistemi P2P cercano di ridurre il traffico interdomain, P4P invece sapere come è fatta la rete sottostante. Tale conoscenza avviene attraverso la comunicazione esplicita tra P2P e provider che offre come vantaggi per il provider:

- uso efficiente delle risorse
- costi inferiori

I vantaggi per il client sono invece relativi al miglioramento delle performance.

Il P4P offre:

- Scalabilità quando è coinvolto un elevato numero di peer.
- Estensibilità: può supportare protocolli P2P differenti.

Un network provider sostiene l'uso del P4P partecipando nello scambio di file e offrendo larga banda per il caching dei file.

I moderni sistemi di P2P permettono il recupero parti di uno stesso file da più fonti. La selezione della sorgente avviene in maniera random, ora, invece, vogliamo che la scelta dei peer da cui scaricare un file sia effettuata sulla base del criterio dei peer più vicini.

Application Layer Traffic Optimization è un protocollo che fornisce applicazioni che contengono informazioni atte a migliorare la selezione dei peer, offrono linee guida al posto di una scelta random.

Si focalizza su un protocollo di richiesta e risposta utilizzato per chiedere l'ALTO service e ottenere informazioni utili nella selezione dei peer; su formati di richiesta e risposta per comunicare le proprie preferenze di rete.

Nel capitolo 2 presenteremo il P4P e il protocollo in generale focalizzandoci sul principio di selezione dei peer.

Valuteremo come si utilizzano le risorse e quali possano essere continue migliorie nella selezione dei peer studiando le linee guida che ALTO fornisce passando attraverso l'ALTO server discovery e la ricerca degli indirizzi.

Passeremo alla presentazione delle informazioni ALTO e verificheremo quali sono i modi in cui opera ALTO.

Ci concentremo su come viene affrontato il problema della sicurezza.

Ci soffermeremo sulla struttura dell'ALTO Client Protocol e sulla presentazione dell'ALTO Service.

Valuteremo il concetto di bandwidth e su come evitare in tal senso colli di bottiglia.

Passeremo poi all'ALTO Gerarchico introducendo il CDN in ALTO.

Altro punto di studio sarà il centro risorse valutando come e dove reperirle, a tal proposito studieremo il caching ALTO e le notifiche inviate da ALTO.

Infine valuteremo la pianificazione dei costi e come alto si concentrerà su questo problema nella scelta degli endpoint.

Nel capitolo 3 passeremo in rassegna le nuove architetture che utilizzano il P4P: ONO, CDN, DECADE e SDN.

Nel capitolo 4 forniremo un confronto tra tutte le architetture studiate, valutando caratteristica per caratteristica quale protocollo sia migliore.

Capitolo 2

Il protocollo ALTO

2.1 Architetture P4P

I peer possono essere localizzati in diversi modi:

Client-based: ogni peer recupera informazioni da un server P4P e seleziona automaticamente il peer per il file exchange.

I client P2P possono sfruttare le informazioni in base alle policy interne; sono liberi di usare le informazioni del provider e i server P4P non hanno pieno controllo dell'uso delle informazioni.

P2P System-based: i server P2P recuperano informazioni dai server P4P per fornire adeguate risposte alle query.

Le politiche di selezione dei peer sono implementate dai server P2P / sistemi P2P; i client P2P sono costretti ad utilizzare il sotto-insieme di peer forniti.

Mixed Client – P2P System-based: ogni peer recupera informazioni di rete da un server P4P e le invia a un server P2P fornendo anche informazioni sul file da scaricare.

Le politiche di selezione dei peer sono implementate dai server P2P / sistemi P2P; i client P2P sono costretti ad utilizzare il sotto-insieme di peer forniti; i server P4P non hanno controllo sulla veridicità delle informazioni ricevute.

P4P Server-based: Client P2P invia una richiesta a un server P4P fornendo le informazioni sul file che vuole scaricare e il server P4P fornisce al client l'insieme di peer da contattare per avviare il file sharing.

I client P2P sono costretti ad utilizzare il sotto-insieme di peer forniti; le politiche di selezione dei peer sono implementate dal server P4P; il P4P controlla e limita la condivisione di alcuni file; ci possono essere colli di bottiglia sul server P4P; i provider non rivelano esplicite informazioni di rete.

2.1.1 Ottimizzazione

Si attua un algoritmo di ottimizzazione basato su:

- Statistiche di swarm
- Banda
- Posizione
- Traffico background
- Indicatori stato di rete

L'obiettivo è infatti quello di:

- Minimizzare il traffico inter-domain
- Minimizzare altre metriche di costi
- Minimizzare l'utilizzo di link
- Massimizzare il throughput

2.2 ALTO protocol^[1]

Application Layer Traffic Optimization.

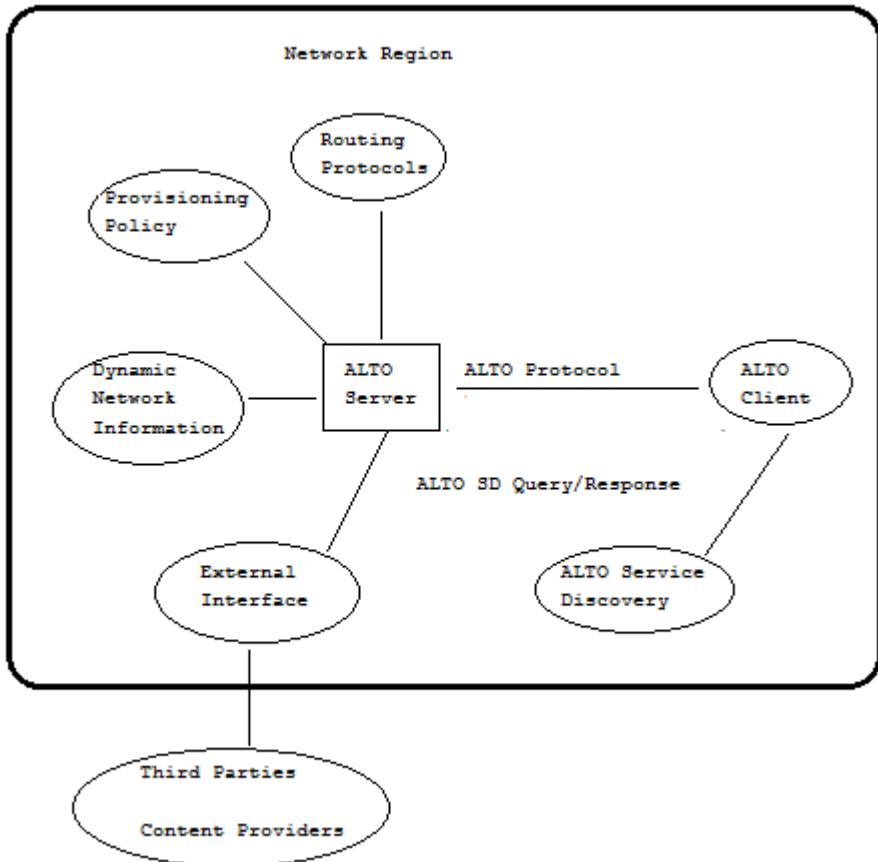
Protocollo che fornisce applicazioni che contengono informazioni atte a migliorare la selezione dei peer, offrono linee guida al posto di una scelta random.

Si focalizza su un protocollo di richiesta e risposta utilizzato per chiedere l'ALTO service e ottenere informazioni utili nella selezione dei peer; su formati di richiesta e risposta per comunicare le proprie preferenze di rete.

In questo lavoro valuteremo i requisiti del servizio ALTO individuando i tipi di informazioni di cui le applicazioni P2P hanno bisogno per ottimizzare la scelta dei peer.

ALTO (Application Layer Traffic Organization) è un protocollo di rete che presenta un meccanismo utile a fornire informazioni di rete alle applicazioni. Serve ad aumentare il traffico e a migliorare l'esperienza dell'utente. E' semplice ed estendibile. Introduce tecniche utili ad migliorare la scalabilità e i problemi legati alla privacy.

Trasmette le informazioni di rete dalla prospettiva di una Network Region



È un protocollo server/client. ALTO Server prepara le informazioni; ALTO Client usa ALTO Service Discovery per identificare l'ALTO server al quale richiede le informazioni usando l'ALTO Protocol.

Le informazioni possono essere prese da altri sistemi. Il server unisce le informazioni da tanti sistemi in modo da fornire una migliore visione di rete alle applicazioni. ALTO Server scambia informazioni con gli altri server in modo da migliorare le informazioni.

L'ALTO Protocol usa un semplice framework per trasmettere informazioni di rete.

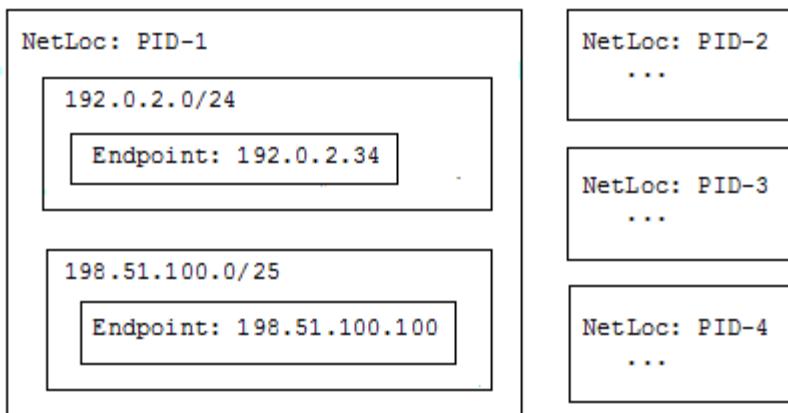
E' costruito su un protocollo di trasporto, su struttura messaggistica e codifica e su un modello di transazione.

ALTO Server usa una Network Map in cui sono raggruppati gli endpoints e nella quale viene mostrata la loro vicinanza. L'idea di vicinanza varia in base alla granularità delle ALTO informazioni. Possono essere considerati vicini, in una rappresentazione, endpoints facenti parte di una stessa subnet, in un'altra, invece, endpoints connessi allo stesso PoP (Post office Protocol) .

Ogni gruppo di endpoints è identificato dal PID (basato su topologia di rete, tipo..) grazie al quale più endpoints vengono considerati allo stesso modo. Può denotare una subnet, un insieme di subnet, un'area metropolitana, un PoP. L'unione di endpoints sotto lo stesso PID ne indica la vicinanza e migliora la scalabilità.

Endpoints aggregati sotto lo stesso PID sono identificati da una lista di prefissi IP. Una network map definisce un PID per ogni indirizzo.

ALTO Network Map



L'ALTO Server definisce le preferenze tra luoghi di rete tramite Path Costs che sono indicati nella Cost Map. La Cost Map si basa sulla sintassi e sulla semantica delle informazioni fornite dall'ALTO server. Path costs sono identificati dal tipo (cosa rappresenta il costo – viene definito un routing cost: misura dalla sorgente alla destinazione) e dal modo (come viene interpretato il costo – è un valore numerico o cardinale che indica l'indice di preferenza – meno è il costo maggiore sarà la preferenza).

La Cost map contiene una lista di sorgenti di rete e destinazioni di rete. Sono introdotte pertanto le Version Tags che permettono agli ALTO Clients di usare le informazioni anche se sono suddivise in 2 mappe. Una Version Map è una stringa associata alla Network Map.

2.3 Information Resource

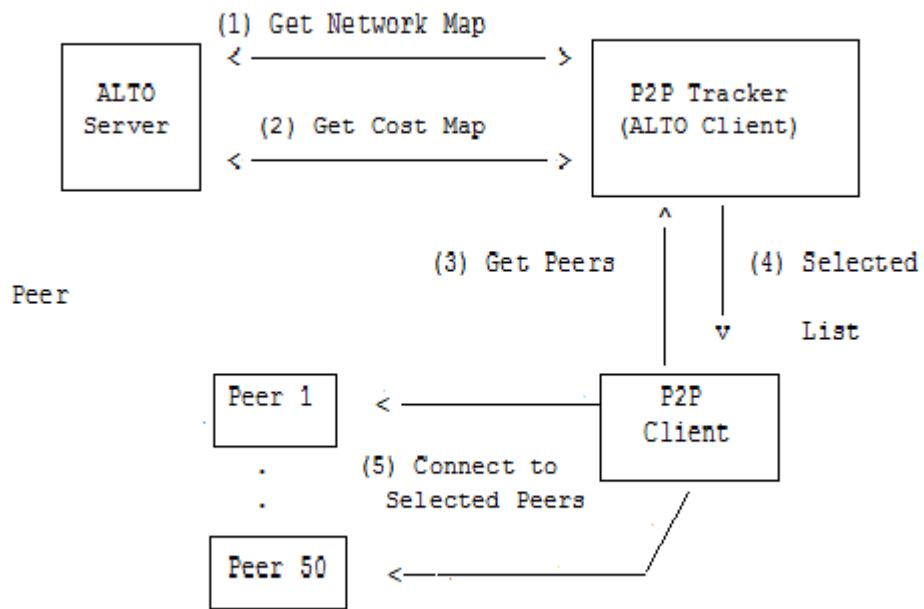
L’ALTO Protocol usa Information Resources e Information Resource Directory e per ricercare le risorse utilizza l’http.

Per ottenere le risorse disponibili, un ALTO Client richiede l’information resource directory al quale l’ALTO server fornisce un URI (uniform resource identifier) che è una stringa che può identificare un indirizzo Web, un’immagine, un file.. Appena riceve una richiesta, l’ALTO server risponde e fornisce all’ALTO client l’information resource directory indicando come raggiungere la risorsa desiderata o restituendo un errore. Se un ALTO Client non riceve l’information resource desiderata dall’ALTO Server può scegliere un altro server o va in default. Si può al limite riprovare a fare la richiesta più tardi. Nel caso di errore, l’ALTO Server deve inviare un ALTO-layer information sotto forma di ALTO Error Resource (fornisce informazioni dettagliate sul perché una particolare richiesta non sia andata a buon fine). Nel caso non fosse disponibile l’ALTO-layer information, l’ALTO Server omette l’ALTO Error resource dalla risposta, ma viene settato un http status code (indica come trattare la risposta). L’http status code correlato ad ogni ALTO Error Code è definito per indicare il corretto comportamento con gli http intermediari e clients.

Se l’ALTO Server non riesce a gestire una richiesta dell’ALTO Client a causa di carico eccessivo, problemi tecnici o manutenzione ritorna un http 503 (service unavailable) o http 307 (temporary redirect) che indica un altro ALTO Server che può soddisfare la richiesta.

2.4 P2P Client: App Tracker

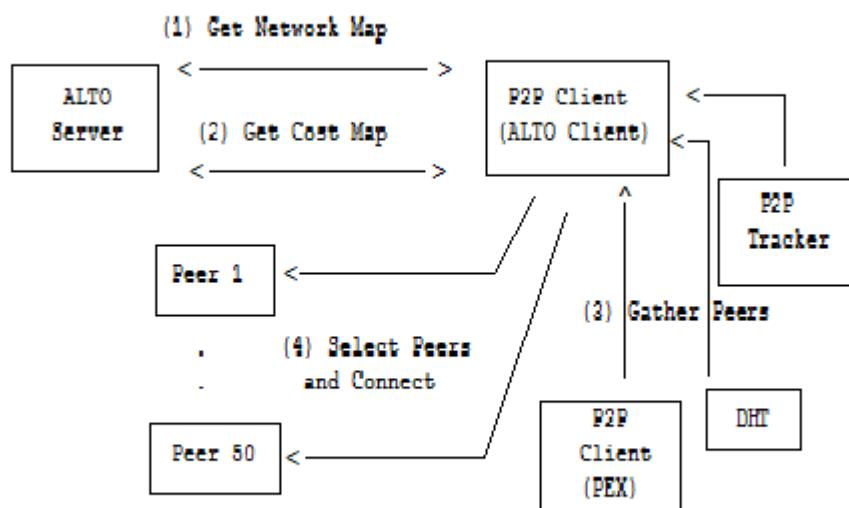
Molti sistemi P2P usano un App Tracker per migliorare il peer selection. Come ALTO Client un P2P Tracker può usare ALTO information per migliorare il traffico di rete e la performance dell’applicazione. In questo caso la scelta dei peer per il P2P client avviene:



1. Il P2P tracker richiede la network map
2. Il P2P tracker richiede la cost map
3. Un P2P Client richiede la lista dei peer al P2P tracker
4. Il P2P Tracker fornisce la lista dei peer al P2P Client
5. Il P2P Client si connette ai peer selezionati

Non tutti i sistemi P2P utilizzano un P2P tracker per ricercare e selezionare i peer.

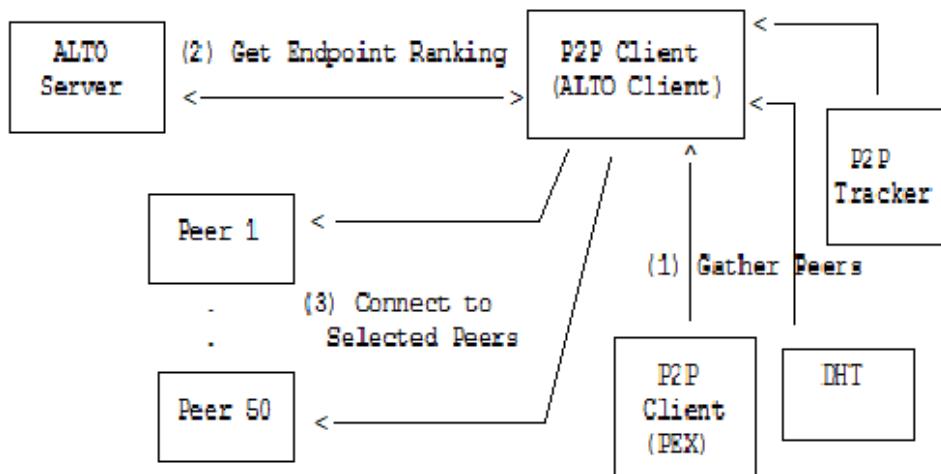
Localmente un P2P Client applica l'ALTO information per selezionare i peer come segue:



1. Il P2P Client richiede la Network Map
 2. Il P2P Client richiede la Cost Map
 3. Il P2P Client trova i peer dalla sorgente, da altri P2P Client, Distributed Hash Tables e P2P Tracker
 4. Il P2P Client usa ALTO information come parte dell'algoritmo per selezionare nuovi peer e collegarsi.

E' inoltre possibile che un P2P Client scarichi il processo di selezione e posizione dei peer da un ALTO server (come mostrato in figura):

1. Il P2P Client trova i peer dalla sorgente, da altri P2P Client, Distributed Hash Tables e P2P Tracker
 2. Il P2P Client chiede l'ALTO Server Ranking Service e indica il cost mode. La risposta contiene la posizione dei peer scelti
 3. il P2P Client si connette ai peer nell'ordine specificato dalla posizione.



2.5 Scelta degli indirizzi

Un host può essere raggiunto usando indirizzi multipli, in questo caso il percorso di rete utilizzato nell'invio di pacchetti può differire. Vi si usano pertanto indirizzi endpoint multipli. L'ALTO protocollo allora consente all'ALTO service provider la possibilità di scegliere il percorso preferito da un endpoint address di un tipo all'endpoint address di un altro tipo.

Un protocollo deve essere in grado di supportare il NAT (Network Address Translation) o deve trovare un modo affinché l'indirizzo IP sorgente passi l'informazione necessaria al server. L'ALTO Server localizza la sorgente di rete tramite l'indirizzo IP della sorgente indicato nei pacchetti richiesti dall'ALTO Client. Se è richiesto l'indirizzo IP per un'interfaccia di rete l'ALTO Client

deve usare la Session Traversal Utilities for NAT (STUN). L’ALTO Client dovrà pertanto poter identificare lo STUN Server il quale (per farsi individuare) deve fornire l’indirizzo di trasporto dell’host.

2.6 ALTO Information

Si ha inoltre bisogno di un registro dell’ALTO Cost Type, di un registro dell’ALTO Endpoint Property, dell’ALTO Address Type.

ISP deve conoscere la topologia di rete e fornire le informazioni ottenute dalle interfacce ALTO. Deve, però valutare i rischi che ciò comporta, si è infatti più vulnerabili agli attacchi riguardo la policy o le relazioni in merito al business e si hanno problemi legati all’efficienza di rete.

Le applicazioni che usano le informazioni sanno che le informazioni possono essere incorrette; questo può capitare perché le credenziali dell’ALTO server sono compromesse. ALTO Client invece devono sapere che l’ALTO server è a conoscenza del Network Location Identifiers.

Un ALTO Server può utilizzare l’autenticazione per limitare ciò che è condiviso con l’ALTO information.

Può capitare che le applicazioni ridistribuiscano un ALTO information per migliorarne la scalabilità.

Importante è anche conoscere per l’ISP il carico di lavoro, una richiesta può essere generata con un basso effort ma se il carico di lavoro è elevato si può avere un attacco di denial of service per limitare i quali si controlla l’accesso all’ALTO server. L’ALTO server può anche chiedere all’ALTO Client di verificare di verificare il costo computazionale prima di allocare risorse.

L’ALTO Protocol è basato sull’http, l’operatore deve configurare l’http server e l’ALTO information.

2.7 Linee guida ALTO

Lo scopo di ALTO è di fornire informazioni che possano aiutare le applicazioni P2P a prendere decisioni migliori nella scelta dei peer.

Il servizio ALTO è fornito da uno o più ALTO Server; può essere chiesto dall'ALTO Client. ALTO Client e ALTO Server devono implementare un ALTO Client Protocol. Questo deve trasmettere le richieste fatte dall'ALTO Client all'ALTO Server e le corrispondenti risposte.

ALTO valuta le risorse dei provider in merito a criteri che riguardano la valutazione. Le risorse dei provider sono caratterizzati da host group descrittori. Si possono avere più tipi di host group descriptor. Tra i quali devono esserci IPv4 o IPv6 address prefix. ALTO Clients e ALTO Server devono identificare il tipo di ogni host group descriptor inviato in una richiesta o risposta. ALTO Client protocol deve essere prorogabile per poter supportare anche in futuro altri tipi di host group descriptor. ALTO Client e ALTO server devono poter segnalare se il tipo di host group descriptor non è supportato.

L'ALTO Client protocol deve stabilire un criterio di valutazione utile a decidere quale sia il resource provider preferito. In più anche in futuro bisogna poter stabilire un nuovo criterio o aggiungerne altri. Tale criterio non deve far riferimento a eventuali problemi di congestione.

2.7.1 Come opera ALTO

Esistono più modi di operare:

Un ALTO Client può essere inserito nelle risorse dell'utente o si può integrare l'ALTO Client in una terza parte che può essere la resource directory. Un ALTO Client protocol deve supportare uno dei modi di operare. L'ALTO Client deve essere progettato in modo tale che deve essere fornito da entità che non siano basilari della rete IP. In un ALTO Client protocol devono poter coesistere differenti esempi di ALTO service creati da provider diversi. Deve essere specificato almeno un modo di richiesta tra i quali: target-aware e target independent. L'ALTO Client protocol deve permettere all'ALTO Server di aggiungere informazioni circa appropriati modi di riuso.

Un ALTO Client protocol deve contenere il supporto per protocolli aggiuntivi e un protocollo di supporto per nuove versioni.

ALTO Client protocol deve usare un trasporto congestion-aware. Deve essere in grado di impedire a rispondere causa overload di terminare la conversazione con l'utente, redirigere l'utente verso un altro ALTO server e richiedere al cliente di tenere basso il suo tasso di query (questo lo si fa, facendo inviare dall'ALTO server come risposta ad una query un messaggio d'errore che avvisa l'utente di riprovare più tardi, o rispondere ma aggiungere l'indicazione di non inviare più query

fino a un certo lasso di tempo). In caso in cui invece non si riesca a rispondere per problemi tecnici o di manutenzione deve: terminare la conversazione con l'utente, redirigere l'utente verso un altro ALTO server e richiedere al cliente di riprovare più tardi.

Un ALTO client protocol è supportato da uno o più meccanismi di scoperta dell'ALTO server che possono essere utilizzati da un ALTO client per scegliere l'ALTO server a cui una richiesta può essere inviata. Devono poi essere stabiliti meccanismi di sicurezza. Bisogna specificare meccanismi di autenticazione degli ALTO Server, degli ALTO client, meccanismi di crittografia, ci devono essere differenti livelli di dettaglio in domande e risposte per proteggere la privacy, e bisogna sapere rispondere a eventuali attacchi.

2.8 Sicurezza

Un problema su cui c'è molta attenzione è quello della sicurezza.

I rischi in cui può incorrere un operatore di un ALTO Server sono:

Eccessiva divulgazione dei dati di un operatore a un ALTO Client. Un operatore infatti considera alcune info confidenziali.

Divulgazione dei dati a un terzo non autorizzato. Capita se qualcuno si intromette nello scambio di informazioni e si fa trasmettere le informazioni da qualche altro ALTO Client. Eccessivo recupero e riuso di dati nella collaborazione tra ALTO Client.

I rischi in cui può incorrere un operatore di un ALTO Client sono:

Divulgazione dei dati a un ALTO server che condiziona il comportamento dell'applicazione.

Divulgazione dei dati a un terzo non autorizzato, che avviene quando un ALTO Client invia le info a un ALTO Server maligno; un terzo non autorizzato si intromette nella trasmissioni dati, un altro ALTO Server comunica al terzo non autorizzato i dati. Il problema è dannoso in quanto più dati vengono accorpati in una richiesta.

Per ovviare a questi problemi un operatore di un ALTO Server deve:

stabilire un range di indirizzi a cui inviare le risposte; richiedere schemi di crittografia, autenticazione e controllo dell'accesso all'ALTO Client protocol.

Un ALTO client invece:

cercare di preservare l'indirizzo IP dell'utente, fornendolo solo a chi strettamente necessario e nasconderlo in caso si incontrino terzi; rendere sicuro contro le intrusioni la procedura di divulgazione dell'ALTO server; crittografare l'ALTO Client protocol.

2.9 ALTO Server Discovery

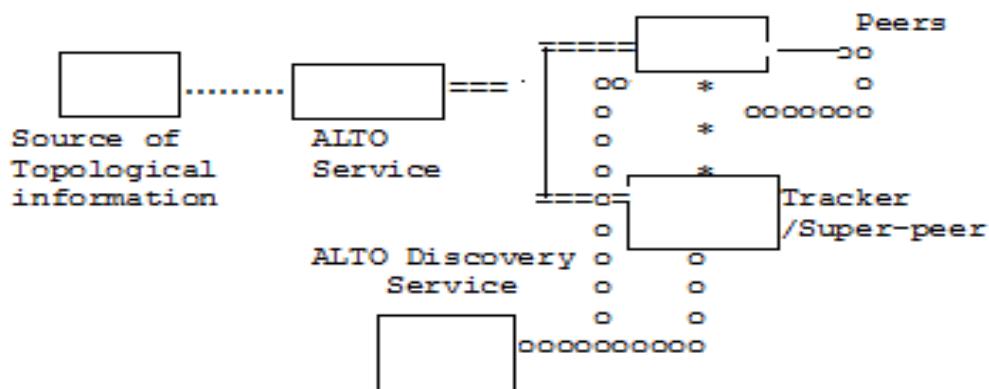
Lo scopo di ALTO è quello di fornire una linea guida alle applicazioni che devono scegliere gli host. ALTO è realizzato da un client-server protocol. L'ALTO Client invia richieste all'ALTO Server per sollecitare le linee guida. L'ALTO protocol è basato sull'http. Ci si aspetta che la procedura di scoperta ALTO produca l'http dell'Information Resource Directory dell'ALTO Server il quale fornisce maggiori informazioni circa i servizi che offre un ALTO server. Obiettivo è quello di proporre un meccanismo di scoperta dell'implementazione dell'ALTO client in ambienti non controllati che si riesca a muovere ad un ritmo veloce.

Ci sono varie opzioni di architettura dove posizionare la procedura di scoperta dell'ALTO server e dell'ALTO client:

a volte è insita nelle risorse dell'utente. La procedura dell'ALTO server è così in grado di interagire con gli utenti. In altri casi (sistemi controllati) non è necessaria la presenza della procedura in quanto se ne occupa il CDN (Content Delivery Network – Rete per la consegna di contenuti) a configurare il tutto manualmente. Un altro metodo è quello di integrare la procedura in una terza parte; si può verificare che questa esegui query al posto del resource consumer per il quale però quando invia messaggi si conosce solo l'indirizzo IP del messaggio originato e ciò può causare problemi.

Un meccanismo che redirige un ALTO Client da un ALTO server a un altro causa parecchia problemi. Primo gli ALTO Server forniscono Network maps di tutto lo spazio degli indirizzi IP e quindi può comprendere tutti i clienti che abbiano una risposta utile. Secondo gli ALTO server possono essere distribuiti in modo indipendente e quindi non essere a conoscenza uno dell'altro.

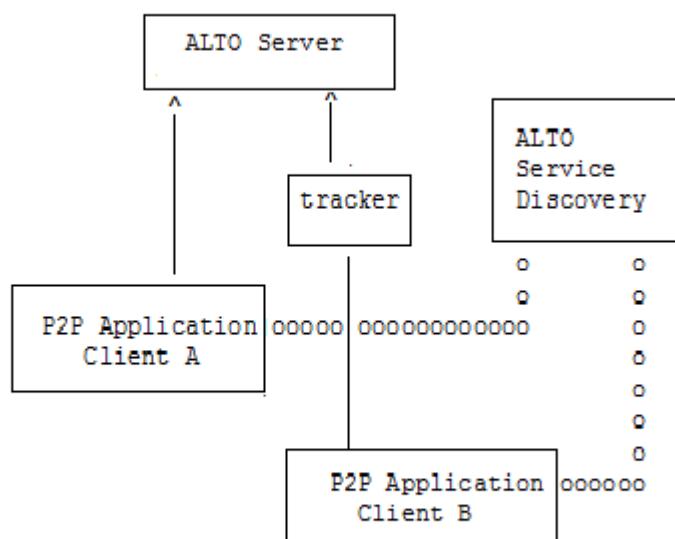
Il meccanismo di scoperta dell'ALTO Server è usato dal peer2peer per recuperare il punto di contatto dell'ALTO Service



Legend:

- ==== ALTO query protocol
- ooo ALTO service discovery protocol
- *** Application protocol (out of scope)
- ... Provisioning or initialization (out of scope)

L'ALTO service discovery in alcuni scenari deve essere eseguito dal resource consumer.



In altri casi si ha L'ALTO server discovery viene eseguito da una terza parte. Ciò causa problemi. Solitamente si ha solo l'indirizzo IP dell'ALTO Client per cercare il corrispondente ALTO Server. Ciò richiede l'uso di DNS inverso (Domain Name System). Può innanzitutto capitare che la ricerca

fallisca o che il valore riportato non sia valido. Inoltre determinare un dominio da un indirizzo IP è problematico; poi c'è il rischio di overhead e che ogni albero generi problemi, non ce n'è uno migliore di un altro. Considereremo pertanto solo il caso in cui la procedura di discovery è fatta dal resource consumer e se una terza parte vuole conoscere le info di contatto se ne occupa il resource consumer.

2.9.1 Condizioni di utilizzo

In generale l'ALTO server discovery deve essere basato sull'indirizzo IP usato per comunicare con gli altri peer. Si hanno bisogno di alcune pre-condizioni:

L'ALTO Client deve scegliere quali possibili famiglie di indirizzi IP usare. Il cambiamento dell'indirizzo IP invalida il risultato della procedura ALTO server discovery (bisogna rifarla per il nuovo indirizzo IP). Indirizzi IP multipli devono ripetere la procedura per ogni indirizzo IP.

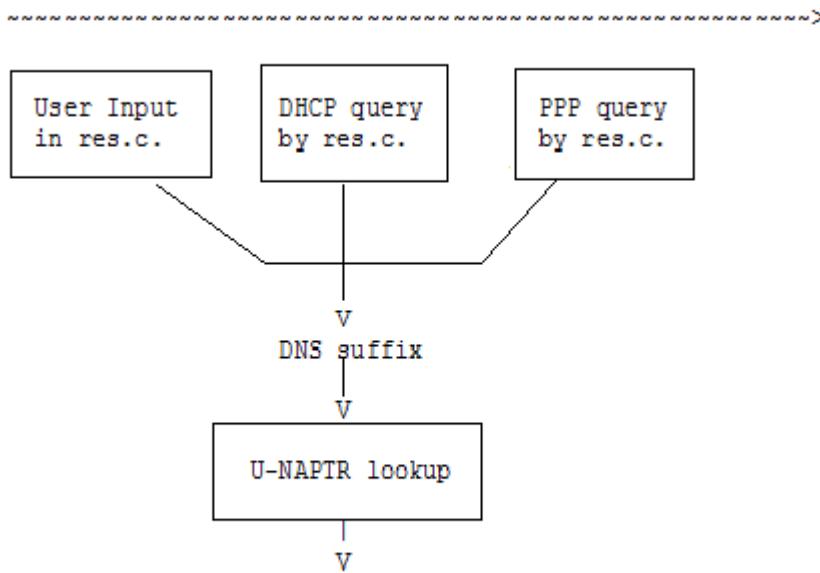
Bisogna prestare molta attenzione quando si usa il DNS, può capitare infatti che sia eseguito su un'interfaccia sbagliata e si possono avere pertanto linee guida errate. La procedura può aver bisogno di informazioni riguardo l'indirizzo IP pubblico, ciò può essere fornito dal NAT. O può aver bisogno di informazioni riguardo l'indirizzo IP di un proxy.

2.9.2 Ricerca indirizzo dell'ALTO Server

Ci sono più modi per ottenere l'indirizzo IP dell'ALTO server. La scelta è compito della distribuzione di rete locale.

Di seguito i differenti protocolli che devono essere usati per cercare l'URI di un ALTO server:

Descending Order of Preference\



ALTO Server's Information Resource Directory URI

Legend:

res.c. : resource consumer

User input: può specificare manualmente il suffisso DNS

DHCP: (dynamic host configuration Protocol) Utilizzato dal provider di rete per ottenere il suffisso DNS

PPP ICP: (point to Point Internet Protocol Control Protocol) utilizzato dal provider di rete per ottenere il suffisso DNS.

Il metodo di scoperta può essere ripetuto se il consumer resource si sposta.

Anziché usare il metodo standard di ALTO server discovery, le applicazioni possono usare un proprio metodo per trovare un ALTO server.

2.9.3 U-NAPTR

È un processo di risoluzione per ottenere l'URI.

Si ha bisogno di domain name come input che può essere determinato in 3 modi: l'utente manualmente configura una specifica istanza dell'ALTO service che vuole usare. Si definisce il DHCP o il PPP IPCP che permettano al network service provider una configurazione da remote del

client. L'ALTO server discovery deve basarsi sull'indirizzo IP Il resource consumer può avere un indirizzo IP privato e uno pubblico.

User input: un utente potrebbe voler usare una istanza dell'ALTO service esterna. L'utente è in grado di specificare il suffisso DNS. Questo è congiunto con l'indirizzo IP del resource consumer per permettere all'ALTO server esterno di indirizzare il client verso un ALTO server. Se non si è trovato il NAPTR si usano gli altri metodi.

DHCP: gli operatori di rete configurano il domain name che deve essere usato per il service discovery nella rete d'accesso che usa il DHCP. L'ALTO service discovery procedure usa il DHCP per recuperare il domain name come un input per l'U-NAPTR.

PPP IPCP è usato per effettuare la configurazione di rete dell'utente (assegnando l'indirizzo IP o il nome del server DNS). L'operatore di rete deve configurare il dominio di rete utilizzato per il server discovery tramite il PPP IPCP.

2.10 Codifica del domain name

Per codificare il domain name usato nel PPP IPCP ogni etichetta è rappresentata da un campo in cui è indicato un numero da 1 a 8 a cui segue il numero di ottetti. Nell'ultimo campo troviamo lo zero. I primi 2 bit di ogni ottetto devono essere zero e i rimanenti sei bit limitano l'etichetta a 63 ottetti o meno. La lunghezza totale del domain name è ristretto a 255 ottetti o meno.

Il dominio example.com è così codificato in 13 ottetti:

7	e	x	a	m	p	l	e	3	c	o	m	0
---	---	---	---	---	---	---	---	---	---	---	---	---

Si definisce anche un metodo per negoziare con il peer remoto il nome del domain name Access Network che deve essere usato alla fine del link:

Type	Len	Access Network Domain Name						
TBD	n	s1	s2	s3	s4	s5	...	

S1, s2, s3.. rappresentano le etichette del domain name utilizzate nella codifica.

Type: assegnato da IANA (Internet Assigned Number Authority)

Len: lunghezza dell'access network domain name in ottetti.

Access Network Domain Name: quello utilizzato dal client.

L'ALTO protocol specification si aspetta che la procedura dell'ALTO discovery produca l'http dell'Information Resource Directory dell'ALTO server il quale fornisce più informazioni riguardo le risorse e i servizi offerti dall'ALTO serve. Il primo passo è quello di produrre un U-NAPTR/DDDS nella forma del DNS name. In secondo luogo la procedura dell'ALTO server discovery usa l'U-NAPTR per ottenere l'URI per l'information resource directory dell'ALTO server.

2.10.1 Applicabilità

La procedura di discovery dell'ALTO server è eseguita dal resource consumer. Dopo ciò il peer chiede la query per conto suo o inizia a condividere con un terzo le informazioni dell'ALTO server. Per completare la procedura il resource consumer controlla se l'utente ha configurato il domain name (altrimenti si usa l'opzione DHCP o PPP). Un client ha più indirizzi IP candidati che può usare per il processo di rilevamento i quali sono ottenuti tramite il NAT o lo STUN.

2.10.2 Third Part

In caso di scenario che prevede il rilevamento effettuato da un terzo il resource consumer è un peer e l'ALTO client è una resource directory che ricerca le linee guida al posto del peer. Oppure si utilizza un applicazione che cerchi le linee guida per il resource consumer. In questo scenario si possono richiedere le linee guida tramite il PPP DHCP. Nel caso il resource consumer abbia bisogno di linee guida per un indirizzo IP differente il resource consumer deve trovare il server da solo e inviare le informazioni di contatto dell'ALTO server alla terza entità.

DHCP abilita l'operatore di rete a fornire il suffisso DNS. Ci sono alcuni problemi a riguardo quali: le opzioni del DHCP non passano attraverso alcune home gateway e per ciò vengono annullate. Inoltre un host che si trova dietro un home gateway può ricevere un suffisso locale che non si può usare per la procedura di server discovery.

Anche il PPP abilita l'operatore di rete a fornire il suffisso DNS: nelle reti residenziali PPP è interrotto nel residential gateway.

Ci sono due possibilità di fallimento dell'ALTO server discovery che possono essere causati da attacchi maligni o da problemi di configurazione. Il rilevamento può non essere in grado di trovare un ALTO server anche se esiste. In questo caso non si possono usare le linee guida. Può anche

capitare che la procedura di rilevamento trovi un ALTO server sbagliato che non può fornire informazioni al resource consumer.

2.11 Attacchi

Un attacco che si può fare all'U-NAPTR è quello di compromettere la procedura di discovery:

1. fornendo un domain name falsificato che può essere utilizzato come input per l'U-NAPTR
2. modificando il DNS
3. impersonando l'ALTO server

Il domain name usato per autenticare l'ALTO server è il domain name che si trova nell'URI. Se chi porta un attacco è in grado di modificare il DNS l'URI può essere sostituito da uno non valido. Il DNS security procura un metodo per limitare gli attacchi. Il domain name usato per l'autenticazione è il domain name nell'URI che viene fuori dall'U-NAPTR. Un ALTO server identificato da un http URI non può essere autenticato. Se un http URI è il prodotto dell'ALTO discovery si è vulnerabili agli attacchi. Per difendersi si usano i livelli bassi (livello 2 di traffico).

2.12 RFC (Request for comments)

Le applicazioni quali P2P e client/server usate per il file sharing usano parecchia capacità di rete e cicli di CPU nei router. Vengono trasmessi enormi quantità di dati grazie alla connessione che viene stabilita direttamente con i nodi.

Un vantaggio di ciò è che le risorse sono presenti in repliche multiple. Solitamente le applicazioni non hanno informazioni veritieri della rete e ha bisogno di selezionare il peer che possa fornire le repliche. Le connessioni direttamente col peer creano congestioni in quanto si attraversano i confini della rete parecchie volte. Si può ovviare a questo problema. ISP, operatori di rete o terzi possono ottenere informazioni di rete più veritieri, tra le quali informazioni riguardanti topologia o link capacity. Fornendo queste informazioni (che non cambiano spesso) alle applicazioni P2P permette queste ad una scelta migliore del peer. Ciò porta a una performance più efficiente, riduce la congestione e diminuisce il traffico tra reti differenti.

I punti principali sono quindi:

trovare un meccanismo che venga usata dal P2P per ottenere informazioni
stabilire una query che richieda le informazioni necessarie.

Definizioni:

Applicazione: un sistema di comunicazioni distribuito che usa l'ALTO service per migliorare le performance o la qualità of experience.

Peer: chi è presente nell'applicazione.

Resource: contenuto o server process alle quali le applicazioni possono accedere.

Resource Identifier: un livello dell'applicazione in cui si identifica una risorse.

Resource Provider: uno specifico peer che fornisce alcuni contenuti

Resource Consumer: uno specifico peer che ha bisogno di accedere ai contenuti.

Transport Address: indirizzi di cui un resource consumer ha bisogno per accedere ad uno specifico resource provider.

Overlay Network: una rete virtuale composta di connessioni dirette stabilita da un gruppo di peer.

Resource Directory: entità separata dal resource consumer che lo assiste nell'identificazione dei resource providers.

ALTO Service: fornisce linee guida su quale resource provider scegliere per ottimizzare la performance o la qualità of experience del client.

ALTO Server: entità logica che fornisce interfacce alle query inviate all'ALTO service.

ALTO Client: entità locale che invia ALTO queries.

ALTO Query: messaggio di richiesta di linee guida.

ALTO Response: messaggio che contiene informazioni riguardo le linee guida.

ALTO Transaction: transazione che comprende ALTO query e ALTO response.

Local Traffic: traffico presente nell'infrastruttura di rete di un ISP.

Peering Traffic: traffico internet scambiato tra due ISP le cui reti sono connesse direttamente.

Transit Traffic: traffico internet scambiato sulla base degli accordi economici tra ISP.

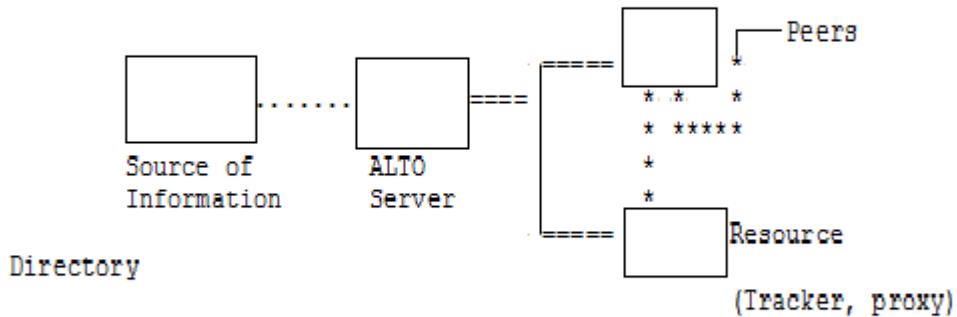
Application Protocol: protocollo usat dall'application per stabilire un rivestimento di rete tra peer che si scambiano dati.

ALTO Client Protocol: protocollo usato per inviare ALTO query e response.

Provisioning Protocol: protocollo usato per inserire informazioni nell'ALTO server.

2.13 ALTO Client Protocol

In figura l'ALTO client protocol



Legend:

- ==== ALTO client protocol
- *** Application protocol
- ... Provisioning or initialization

Il problema delle applicazioni è quello di trovare il percorso migliore per lo scambio di pacchetti tra specifici indirizzi di sorgente e destinazione. Come soluzione si possono utilizzare bassa latenza, alta affidabilità e priorità.

Applicazioni possono migliorare il traffico generato; infatti quando un applicazione P2P deve stabilire una connessione si cerca una resource, la scelta di un buon host ha un grande impatto sul traffico generato. La selezione dei peer è utile nella fase iniziale del processo, quando un peer che fa richiesta riceve la lista di candidati dove ci sono le informazioni. Da questa lista viene estratta un piccolo insieme di candidati ai quali connettersi e scambiare informazioni. Bisogna essere, pertanto, consci di ciò che c'è nella rete e utilizzare al meglio le conoscenze.

2.13.1 File Sharing

Le applicazioni che usano il file sharing permettono agli utenti di cercare i contenuti messi a disposizione dagli altri e scaricarli.

2.13.2 Live Media Streaming

Applicazioni P2P per il live streaming permettono agli utenti di ricevere contenuti multimediali prodotti da una sorgente e destinati a più destinazioni in real-time. Ciò è importante per chi (utenti o reti) non supporta il multicast. Spesso i peer fanno sia da sorgente che da destinazione. Il fine dell'ALTO solution è quello di aiutare i peer a trovare peer con cui scambiare il contenuto multimediale.

2.13.3 Real-Time Communications

Si permette agli utenti di stabilire flussi multimediali diretti per real-time audio e video. Purtroppo alcuni utenti hanno un accesso limitato a internet causa NAT, firewall o proxy. Si ha bisogno di altri elementi che trasmettano i media. ALTO deve aiutare i peer a trovare la migliore trasmissione.

2.13.4 Distributed Hash Tables

DHT sono una classe di algoritmi usati per migliorare le funzionalità di ricerca in sistemi P2P popolari. In questi sistemi un peer mantiene gli indirizzi di un insieme di altri peer presenti nella stessa DHT in base ad alcuni specifici criteri. ALTO fornisce informazioni importanti per gli algoritmi DHT.

2.14 ALTO Service

Il proposito dell'ALTO Service è quello di fornire applicazioni con informazioni che possono essere usate per avere migliori performance con il peer selection. Ci sono tante informazioni che possono aiutare a ciò. Ci sono anche informazioni che vengono utilizzate per altri scopi.

ALTO services sono forniti al massimo da 3 entità:

Network operators, terzi (hanno informazioni di rete) e comunità di utenti.

Ci sono alcuni elementi che un ALTO client può inserire in una query che può aiutare a far aumentare il livello di precisione delle risposte. Ma le applicazioni possono considerare certe informazioni private.

Caching serve per migliorare il traffico generato e richiede enormi quantità di data transfers. ALTO e caching sono ortogonali.

Per quanto riguarda la sicurezza si richiede autenticazione prima di fornire informazioni delicate. Il processo di peer selection influenza il comportamento dell'applicazione. Si può però considerare il server ostile in quanto l'operatore può:

1. usare ALTO per imporre copyright.
2. redirigere le applicazioni verso mediatori corrotti.
3. tracciare connessioni per isezioni
4. applicare policy basate su criteri diversi dell'efficienza di rete.

2.15 Bandwidth

Cloud computing, network application, software as a service, platform as a service, infrastructure as a service sono entità di calcolo che interagiscono tra loro in una rete. Quando, per via di queste interazioni, si usano parecchie risorse di comunicazione si ha un utilizzo più efficiente delle risorse di rete e di calcolo. Ai livelli bassi di rete sono presenti meccanismi di recupero e ripristino in caso di errori di rete. Si ha migliore gestione degli errori e opportunità di fornire in modo più efficiente più livelli di affidabilità. Le aspettative di affidabilità sono contraddistinte da due parametri RTO (Real Time Objective) è il tempo in cui si ottiene il backup e RPO (Recovery Point Objective) che quantifica in termini di tempo l'ammontare di dati persi che può essere tollerata quando capita un guasto (può andare da millisecondi a 12 ore). Con l'obiettivo di avere tempi di recupero di circa un'ora, si riesce a diminuire il costo complessivo raggiungendo alti livelli di affidabilità, per ottenere ciò si ha bisogno di cooperazione tra applicazione e livello di rete.

General multi-protocol label switching (GMPLS) fornisce una topologia dinamica di rete e resource information e la capacità di allocare risorse dinamicamente. All'ottimizzazione del percorso del traffico ci pensa il Path computation element (PCE). Ma né GMPLS, né PCE riescono a fornire interfacce appropriate.

Mentre cloud è abbastanza fumoso, il livello fisico di risorse di rete è reale e delimitato da vincoli fisici e logici. Consideriamo le reti connesse centri di dati per via della connettività di rete che è significativa rispetto alla larghezza di banda. Centri di dati molto grandi dispongono di fibra di accesso multiplo e consumano parecchio.

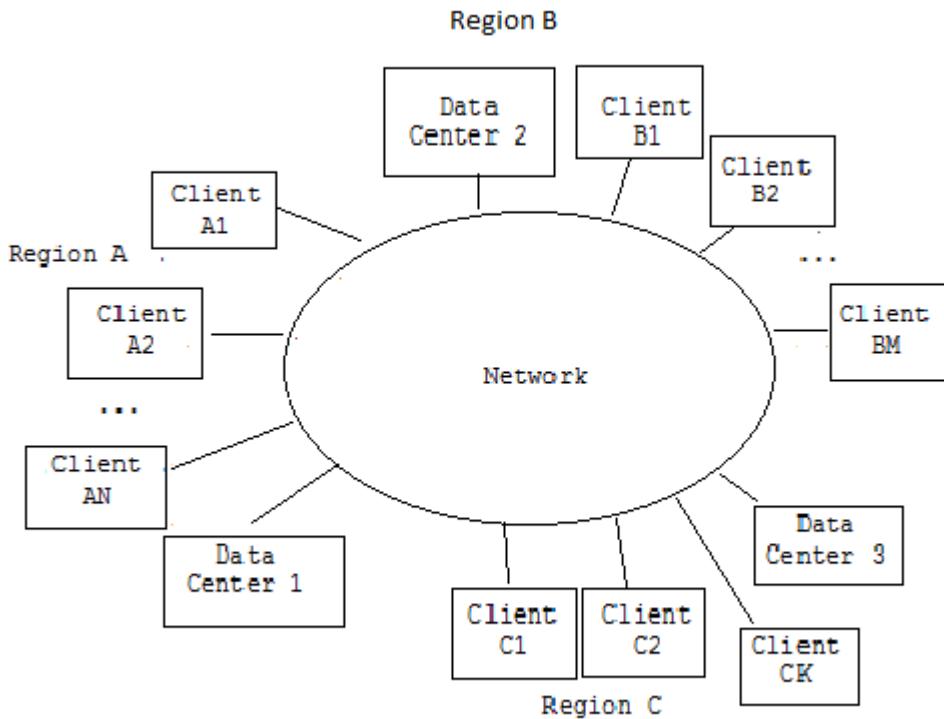


Figure 1. End system to data center communications.

Clients sono end system con piccolo accesso alla larghezza di banda (paragonabile a una lunghezza d'onda). Clients sono partizionati in 3 reti di regioni (A, B, C). Ogni client in una regione è associato a un data center.

Nel caso di VoD (video on demand) è utile per la rete l'aggregazione della larghezza di banda degli end system. Qui la trasmissione avviene solo tra data center e client.

Data center e reti hanno vincoli e costi marginali che variano col carico o col tempo. Si passa al cross stratum optimization (CSO). Si vuole verificare se si può accogliere una nuova richiesta da una regione utilizzando un data center diverso in quanto il locale pieno. Per rispondere bisogna conoscere la capacità del data center a cui si richiede, i costi marginali di richiedere servizio a un altro data center, l'abilità della rete di fornire larghezza di banda tra la regione e il data center e i costi incrementali che ciò comporta.

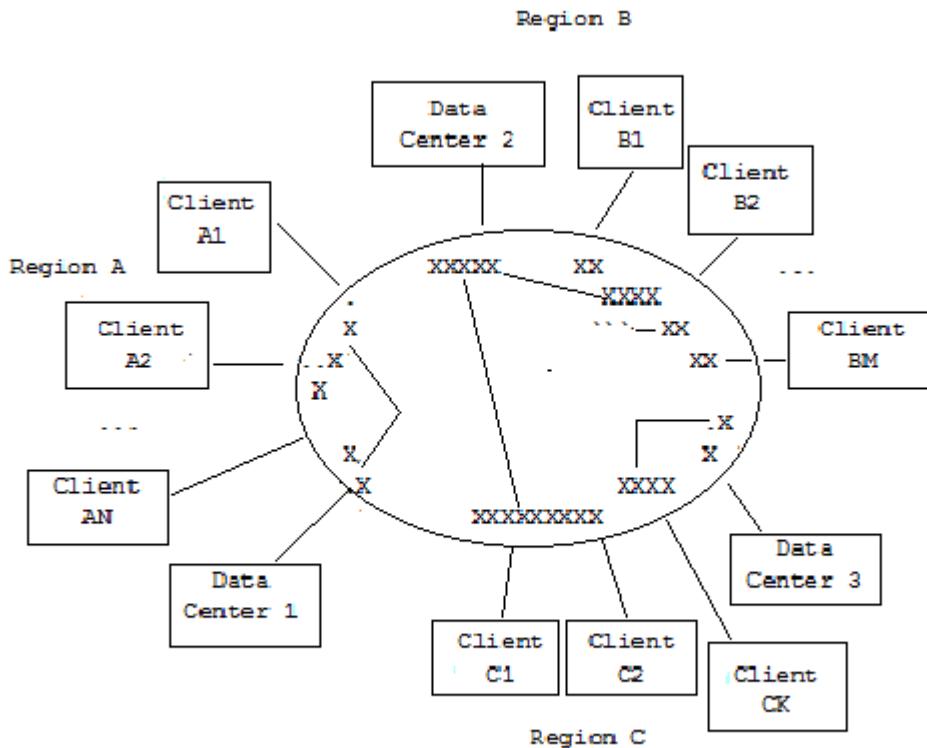


Figure 2. Aggregated flows between end systems and data centers.

Fallimenti dei data center hanno maggiori impatti sul VoD. Se un altro data center ha capacità in più i client vengono rediretti verso questi se la rete in cui si trovano ha anche a sua volta capacità. MPLS e GMPLS sono in grado di reindirizzare il traffico molto velocemente preservando il QoS. La comunicazione tra data center avviene per scambiarsi parecchie informazioni: espansione di capacità (cloud bursting), backup di dati etc...

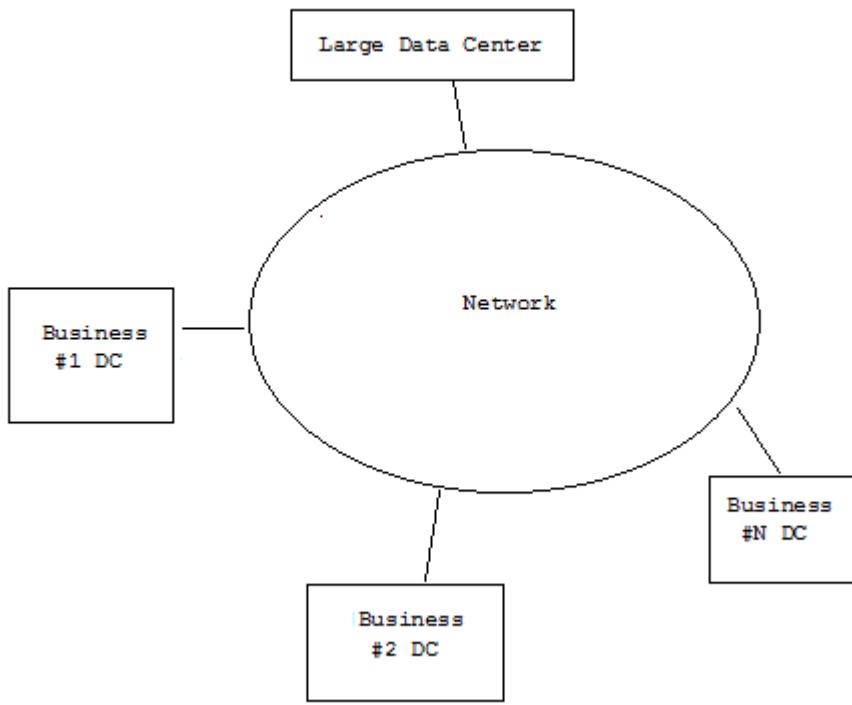


Figure 3. Basic data center to data center networking.

In figura un esempio di come vengono contattati data center esterni per richiedere capacità aggiuntiva. Bisogna comunque considerare i vincoli computazionali, i costi fissi e marginai e i costi di larghezza di banda di rete per i data center sia locale sia remoto.
 Per applicazioni di rete che hanno bisogno di livelli alti di affidabilità si possono avere data center esterni e business location in esubero.

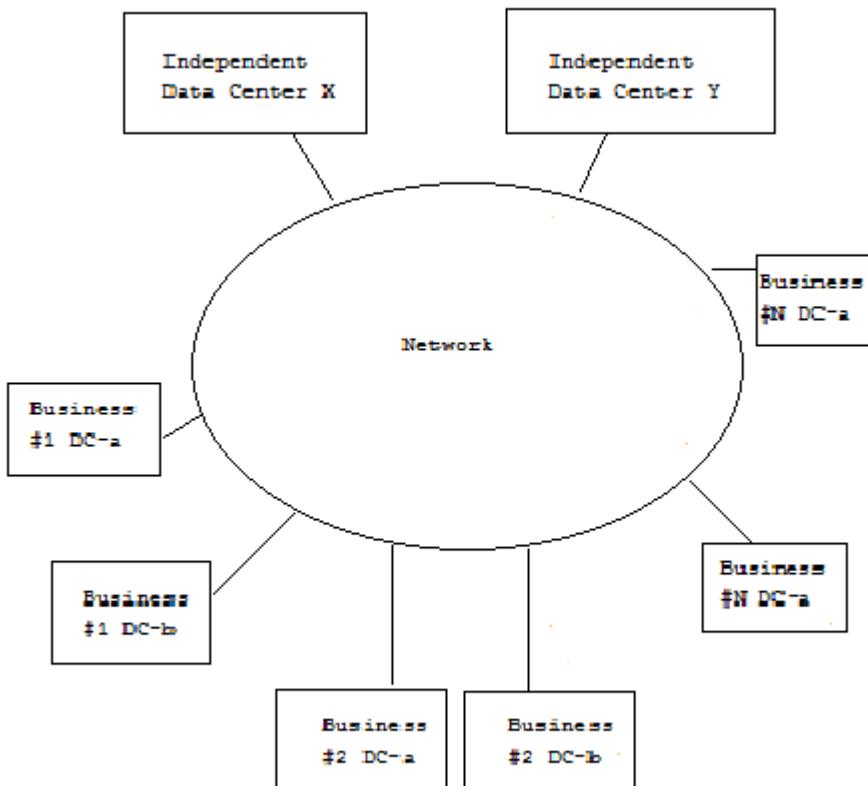


Figure 4. Data center to data center networking with redundancy

È il caso dei database geografici che parlano con siti che hanno alta disponibilità.

Per bilanciare il carico si utilizzano due tecniche:

bilanciamento all'interno del data center (locale)

bilanciamento globale usato per assegnare i clienti a particolari data center che sono fuori portata.

Sistemi che usano bilanciamento globale hanno scarsa conoscenza della rete, usano, pertanto, informazioni geografiche basate sugli indirizzi IP. Nel CSO si può avere:

1. Una network query interface: per informarsi sulla disponibilità di larghezza di banda tra client di regione e data center.
2. Una network resource reservation interface: per fare richieste esplicite di banda tra client di regione e data center.
3. Una fault recovery interface: per richiedere il reindirizzamento di traffico da un data center a un altro. O per i livelli di rete di fare richiesta di aiuti in casi di errori di rete.

In merito all'applicabilità dell'ALTO protocollo e la sua estensione per supportare una network query interface bisogna considerare le proprietà di scenari con elevata larghezza di banda. In primo luogo portata e scala. Solitamente l'utente è uno che controlla le applicazioni nei data center. Gli utenti che richiedono le informazioni riguardo la larghezza di banda sono pochi rispetto a quelli della rete peer 2 peer. Infatti poiché queste informazioni sono riservate un network provider inserisce un limite a queste entità pre- registrate. Inoltre in merito alla scalabilità per la network resource queries si è interessati alla larghezza di banda disponibile al momento della richiesta, in futuro e in un trasferimento dati.

Un altro obiettivo è quello di minimizzare il bisogno di ogni strato di conoscere troppo riguardo ciò che fanno gli altri.

Le estensioni principali ALTO utili ad interrogare la rete circa le applicazioni a largo consumo di banda sono:

- (a) Bandwidth Constraint Information
- (b) Constraint Based Filtering
- (c) Multi-cost information [MultiCost]
- (d) Endpoint Access Bandwidth Capacity

Ciò che importa di più alle applicazioni è la larghezza di banda disponibile tra due entità che comunicano tramite una WAN, una metro area, una LAN o all'interno di un cluster.

Tra le varie richieste di banda si può avere:

1. Banda disponibile tra una coppia di sorgente e destinazione.
2. Banda disponibile tra una sorgente e più destinazioni.
3. Banda disponibile tra un insieme di sorgenti e un insieme di destinazioni.

Se la richiesta di banda riguarda flussi simultanei c'è la possibilità che i flussi condividano network resources. Questo è il caso di problema di flusso multi-prodotto. In generale si lascia libera scelta del percorso e la divisione delle richieste attraverso percorsi multipli. Formulazioni alternative si hanno quando la scelta del percorso tra una sorgente e una destinazione è ristretta ad una lista .

2.15.1 Colli di bottiglia

Per una ottimizzazione del traffico, la presenza e la capacità di questi colli di bottiglia devono essere trattati da applicazioni con parecchia banda.

Dal punto di vista del provider si ha:

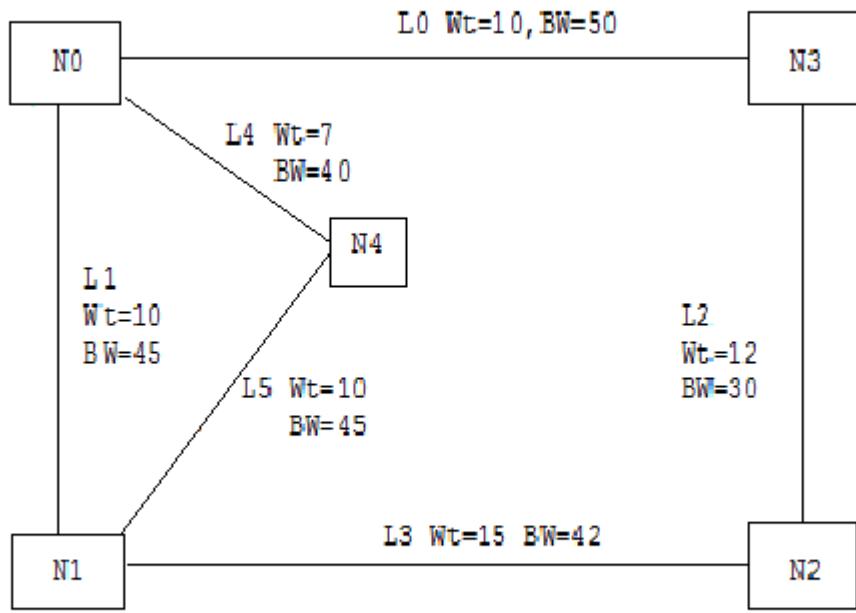


Figure 1 Generic Constrained Network Example

Sono rappresentati 5 nodi e 6 link. Consideriamo un singolo costo e un singolo limite di banda per link visti dalla prospettiva del network provider che possono o non possono essere le uniche linee guida nella scelta del percorso. Comunque in un percorso tra una sorgente e una destinazione si sommano tutti i singoli costi di ogni singolo link attraversato.

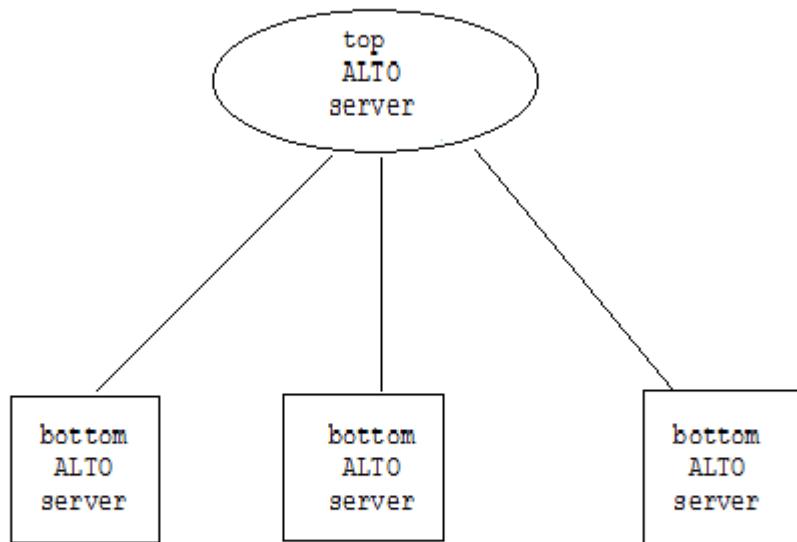
Consideriamo il controllo dati e le tecnologie per dedurre quali siano i percorsi disponibili.. Tali metodi influenzano l'efficienza con cui la rete riesce a trasmettere costi e limiti alle applicazioni. Le tecnologie che permettono la sistemazione arbitraria includono: WDM, TDM, MPLS, ATM e Frame Relay. In questi casi un provider di rete fornisce una rappresentazione grafica (a volte accompagnata da una lista dei percorsi possibili) della rete utile alle applicazioni di ottimizzare la scelta dell'instradamento. Ethernet offre la più ampia varietà di scelta di percorsi.

Il provider di rete fornisce informazioni circa il percorso utili a ricercare informazioni nascoste o per minimizzare il carico di informazioni che si ha bisogno di trasferire attraverso ALTO.

2.16 ALTO Gerarchico

L’ALTO Service permette alle applicazioni di ottenere informazioni di rete per ottimizzare il traffico. L’ALTO protocol definisce la comunicazione tra un ALTO client e un ALTO server. Un ALTO server ha bisogno di mantenere le informazioni ALTO e fornire i servizi ALTO ai client. In reti grandi si potrebbero avere colli di bottiglia, si propongono pertanto meccanismi per migliorare

la scalabilità quali http caching, redistribuzione di informazioni ALTO, e impegno di più ALTO server. Per migliorare la scalabilità di ALTO si propone un'architettura gerarchica che permetta di collezionare le ALTO information e fornisce servizi ALTO in modo distribuito. Questo meccanismo non tiene in conto soltanto dell'ALTO service ma anche delle ALTO information.



Notiamo 2 livelli dove a livello più basso ci sono più server mentre a livello alto solo uno. A livello basso si hanno le mappe o si forniscono ALTO services.

I dati ALTO nell'architettura di raccolta sono collezionati e generati in maniera distribuita mentre gli ALTO services sono forniti dagli ALTO client in maniera centralizzata. A basso livello vengono generate le mappe e inviate al livello superiore che è l'unico che possiede tutte le mappe e fornisce ALTO services agli ALTO client. Nell'architettura distribuita i dati sono generati in maniera centralizzata mentre gli ALTO services sono forniti agli ALTO client in maniera distribuita. L'ALTO server a livello superiore raccoglie tutte le topologie di dati, genera le mappe e le distribuisce ai server a livello inferiore. Tutti i server forniscono ALTO services agli ALTO client. Nell'architettura di riflesso i dati sono generati in una maniera distribuita e ALTO services sono forniti ai client allo stesso modo. Ogni server di livello inferiore raccoglie alcune topologie di dati e genera una parte delle mappe. Tutti poi inviano queste parti all'ALTO server di livello superiore il quale poi fornisce la mappa intera a tutti i server sotto. Gli ALTO services sono forniti da tutti i server. Capita che i server del livello inferiore invino update delle mappe al server a livello superiore, il quale invia l'update come riflettore dei dati.

2.17 ALTO con CDN (content distribution networks)

Un CDN consiste in una rete di server attaccata a Network Service Provider (NSP). Il punto di attacco è spesso il più vicino agli utenti al fine di ridurre il traffico da caricare sulla dorsale NSP e per avere un throughput più elevato. Per ottenere il benefit ottimale da un CDN è preferibile inviare contenuti dai server più vicini agli utenti che li richiedono. Se i CDN sono posizionati fuori o in profondità rispetto a una rete NSP diventa necessario avere una conoscenza dettagliata della topologia di rete e dei costi.

CDN: infrastruttura di rete nella quale gli elementi di rete collaborano a livello 4 attraverso il livello 7 per una più efficace consegna del contenuto ai software.

CSP (content service provider): fornisce servizi agli utenti

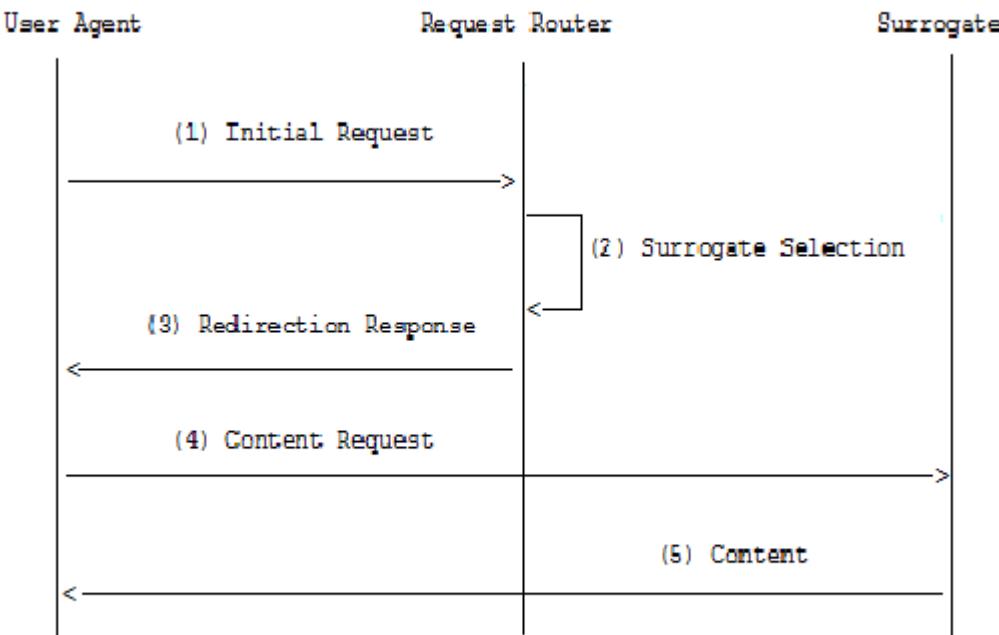
NSP: fornisce servizi di rete basati sulla connettività.

Surrogate: una funzione che interagisce con gli altri elementi del CDN per il controllo e la distribuzione dei contenuti e per la consegna degli stessi.

Un CDN consiste in 3 componenti: request routing function, surrogate function, origin function.

Request routing è responsabile di ricevere il contenuto richiesto dal software, ottenendo informazioni riguardo i surrogati e indirizzando i software ai corrispondenti surrogati.

Surrogate interagisce con altri elementi per il controllo e la distribuzione dei contenuti e interagisce con i software per spedire il contenuto.



In figura una chiamata:

1. L'user Agent fa la richiesta al CDN (applicazione o una richiesta di DNS).
2. Il Request Router seleziona un appropriato surrogate basato sull'IP dell'user agent e sulla conoscenza della topologia di rete.
3. Il request router risponde alla richiesta iniziale indicando la redirezione alla cache selezionata.
4. L'user agent usa queste informazioni per connettersi direttamente al surrogate e richiedere il contenuto desiderato.
5. Se la policy del CDN permette agli user agent di ricevere il contenuto richiesto il surrogate spedisce il contenuto agli user agent.

L'uso principale di ALTO in un contesto CDN è di migliorare la selezione del CDN surrogate o origin. Si può ottenere la mappa della rete e l'informazioni dei costi ascoltando il protocollo di instradamento dell'NSP. L'NSP può essere messo in grado di mantenere una separazione funzionale tra il piano di instradamento e il calcolo della mappa di rete, ciò è interessante perché:

l'ALTO Service può fornire una visione filtrata della rete relativa al luogo in cui sono collocati i CDN e la loro vicinanza agli utenti.

L'ALTO Service può applicare ulteriori policy alla mappa di rete e costi per fornire ai CDN una visione specifica della rete.

In generale l’ALTO service dovrebbe essere applicato da entità il cui scopo è quello di ottimizzare l’instradamento.

Il request router ha bisogno di avere informazioni di ogni utente. La granularità delle informazioni circa la collocazione degli utenti dipende dalla posizione dei CDN. Queste informazioni sono spesso ottenibili tramite un protocollo di instradamento dinamico.

In alcuni casi un CDN non riesce a raggiungere tutti gli utenti di cui un CSP ha bisogno. In questi casi la mappa di rete può essere utilizzata per mostrare al CSP quali utenti siano raggiungibili attraverso un particolare CDN. Quando un CDN è gestito da un NSP, l’NSP fornisce servizi di broadband. È comune per utenti di broadband ottenere il proprio indirizzo IP dinamicamente. Inoltre in alcuni casi CDN surrogati potrebbero essere sovraccarichi indicando ai CDN che scelgono altri surrogati in differenti regioni della rete. NSP influenza questa scelta preferendo il vicino migliore (in termini di costi). Un CDN perciò deve essere in grado di scegliere i giusti surrogati. Inoltre un NSP usa gli ALTO service per mostrare i costi tra un la rete IP di un utente e i punti di collegamento per influenzare la posizione dal quale i CDN offrono i contenuti.

Prima che un surrogato di un CDN possa spedire il contenuto ad un utente deve averne una copia. Il contenuto può essere ottenuto prima che venga richiesto dall’utente (pre-positioned) o in risposta all’utente (on-demand). La fonte ultima è l’origine del contenuto ma spesso i CDN utilizzano qualche gerarchia grazie alla quale non per forza si ha bisogno dell’origine. Ciò porta parecchi vantaggi.

Inoltre per un contenuto di un particolare servizio CSP l’Origine può essere utilizzata per la flessibilità e può essere installata in maniera distribuita su più aree geografiche.

Un surrogato può scegliere da quale parte ottenere un contenuto, anche se può essere influenzato dal CDN.

2.18 Third part

L’obiettivo di ALTO è quello di fornire linee guida alle applicazioni nella scelta degli host. Il client per richiedere queste linee guida deve essere associato ad un ALTO server.

Nelle applicazioni che prevedono una directory centralizzata l’efficienza è migliore se il client è incorporato nella directory delle risorse. Il client interroga il server per conto dei consumatori, ha bisogno quindi di sapere quali sono i server che possono fornire linee guida. Questa è l’individuazione dei server ALTO da parte di terzi. Il client dovrebbe conoscere il proprio indirizzo IP.

La procedura di individuazione dei server da parte di terzi viene eseguita in 2 passi:

Un suffisso DNS è prodotto, tramite ricerca DNS sull’indirizzo IP del consumatore; questo indirizzo IP di origine dei messaggi di protocollo arriva alla directory delle risorse.

Questo suffisso DNS è utilizzato per un U-NATPR per poter ricercare l’URI.

Il suffisso DNS è il nome di dominio in cui il cliente si trova.

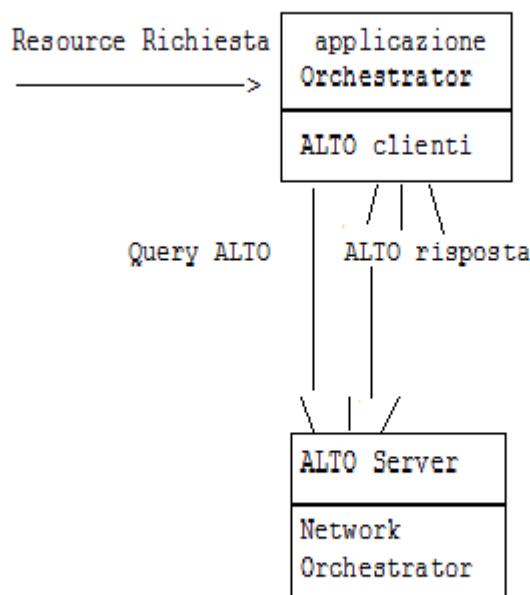
2.19 Scambi a Banda Larga

Consideriamo lo scambio di informazioni per applicazioni a banda larga. Si possono verificare tre casi, l'uso di risorse che diventa significativo a causa di molte richieste da parte dei client, il caso in cui c'è un enorme scambio di dati per i quali si usano corsie di passaggio rapido e il caso in cui il provider inter-CDN può esporre una risorsa per rendere necessario l'utilizzo delle applicazioni.

Utilizzeremo un trade off per l'ottimizzazione che consideri i costi di rete e i vincoli. In particolare si tengono presenti i vincoli per ridurre gli scambi di informazioni estranei alla rete e per nasconderli e i costi per avere ottimizzazione del traffico.

Una delle principali preoccupazioni dei fornitori di rete è dettagliare le informazioni, bisogna considerare quali siano quelle utili all'applicazione, nascondendo dettagli sulla rete. Bisogna quindi diminuire la quantità di informazioni e aumentarne l'utilità.

In figura è rappresentato come avviene lo scambio di informazioni: il client ALTO si interfaccia con il server. Il Client interagisce con l'applicazione Orchestrator che ha conoscenza dell'utente finale, della posizione del Data Center e delle risorse di informazione.



Client e server non hanno bisogno di essere negli stessi confini. L'architettura può essere inter operatore o multi-ambiente, cioè il client può interagire con più ALTO server o client o con un ALTO server rappresentante.

Un punto critico nello scambio di informazioni è la quantità di informazioni scambiate tra l'applicazione e gli strati di rete. Al fine di ridurre informazioni estranee si fa un filtro riducendo i risultati ad un insieme specificato di sorgente e destinazione senza contare tutta la mappa di costo. Si può avere un vincolo di filtraggio sui percorsi o grafici riducendo i risultati a solo quelli che soddisfano il client nei limiti delle spese. In un ambiente controllato l'ottimizzazione viene migliorata dalla condivisione dei dati relativi alla larghezza di banda e alle misure di costo.

In figura una rete dove DC indica data Center, ER la regione dell'utente finale, N un nodo di commutazione e L un link. Il collegamento riporta la capacità e i costi nonché un costo mappa (tra ER1, ER2 e DC1, DC2, DC3). Consideriamo che la capacità massima disponibile tra zona terminale e data center sia di 5 unità ($L1=L2=L5=L6=5$), $L3$ (che è di 8 unità) rappresenta collo di bottiglia a tutto il data center in quanto presente in tutti i percorsi e le regioni.

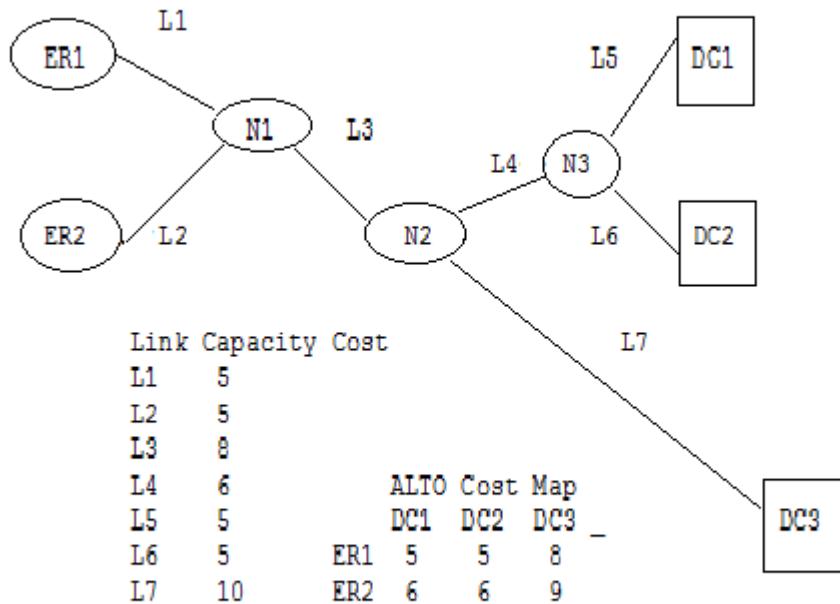


Figure 2. Example network illustrating bottlenecks

Il percorso di ER1 sarebbe su DC1 o DC2 data la struttura della mappa dei costi ALTO. Tuttavia con i cost map citati la connessione da ER1 a DC3 potrebbe essere migliore.

Affinchè lo strato di rete fornisca informazioni, l'applicazione deve fornire il punto finale della mappa dei costi che deve includere:

L'indirizzo finale dell'utente (posizione)

L'indirizzo di destinazione dell'end point.

2.20 Centro Risorse

Quando le risorse sono distribuite geograficamente in centri di dati, il Centro decide dove ospitare l'applicazione richiesta che deve valutare la disponibilità, i costi di rete, le risorse e la struttura. Consideriamo i dati nel contesto delle infrastrutture per applicazioni di scambio valutando l'estensione del protocollo ALTO come parte dell'infrastruttura.

In una rete applicazione ambiente si ha:

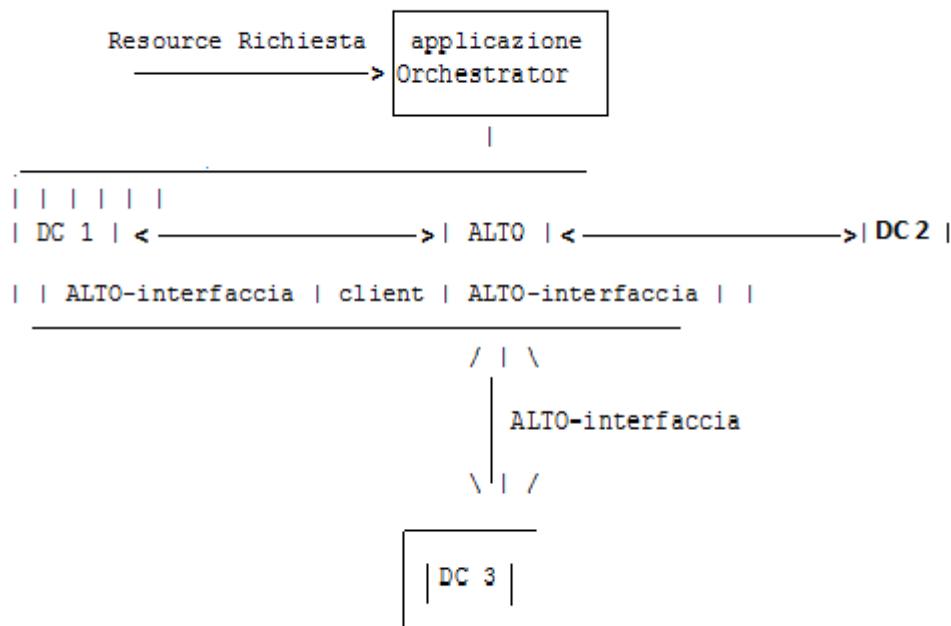


Figura 1 Architettura ALTO in reti distribuite di data center

Come si nota dalla figura gli ALTO client possono stabilire un ALTO interfaccia con ogni data center per raccogliere le informazioni e selezionare dati ottimali in merito alla posizione centrale.

La risorsa richiesta arriva all'orchestratore dell'applicazione che è un'entità separata dal Client ALTO. La raccolta dei dati viene inviata all'orchestratore di applicazione. Le informazioni riportate riguardano capacità, costi di prestazioni, struttura e reti associati al data center.

In una tipica infrastruttura il software applicativo viene eseguito all'interno di una macchina virtuale assegnata all'hardware fisico sotto la supervisione di un hypervisor virtuale. Per ottenere un determinato livello di prestazioni di una macchina virtuale all'interno di un'applicazione si ha bisogno di una certa quantità di risorse di calcolo quali memoria, CPU virtuali, e un dedicato storage sul disco.

Un'applicazione scalabile viene in genere implementata in modo che possa essere eseguita su un numero di macchine virtuali variabile a seconda del carico. Diverse macchine virtuali possono assumere ruoli diversi nell'applicazione. Ciò comporta l'uso di istanze differenti. Bisogna però fare attenzione perché il centro dati può fornire un numero limitato di istanze e le risorse sono una quantità finita. Sono utilizzati due approcci: fornire informazioni in base alle capacità disponibili e rispondere positivamente o negativamente alle richieste di risorse o fornire limite sulla memoria, sul numero di CPU virtuali e sulla memoria locale.

I fornitori di servizi hanno introdotto modelli di pricing e forniscono tre modelli basati sui concetti di casi riservati, su richiesta di istanze e su istanze in loco.

ALTO Client raccoglie le informazioni estratte da ciascun data center utili ad avere una visione astratta di stato di ogni data center

2.21 Caching ALTO

L'obiettivo di ALTO è quello di fornire orientamenti per le applicazioni che devono selezionare uno o più host da un insieme di candidati che sono in grado di fornire le risorse desiderate. ALTO utilizza ALTO per la scoperta di informazioni disponibili tramite approcci map-based e non map based. Inoltre utilizza meccanismi per memorizzazione dei contenuti nella cache.

ALTO utilizza un framework estensibile per trasmettere le informazioni ed è così strutturato:

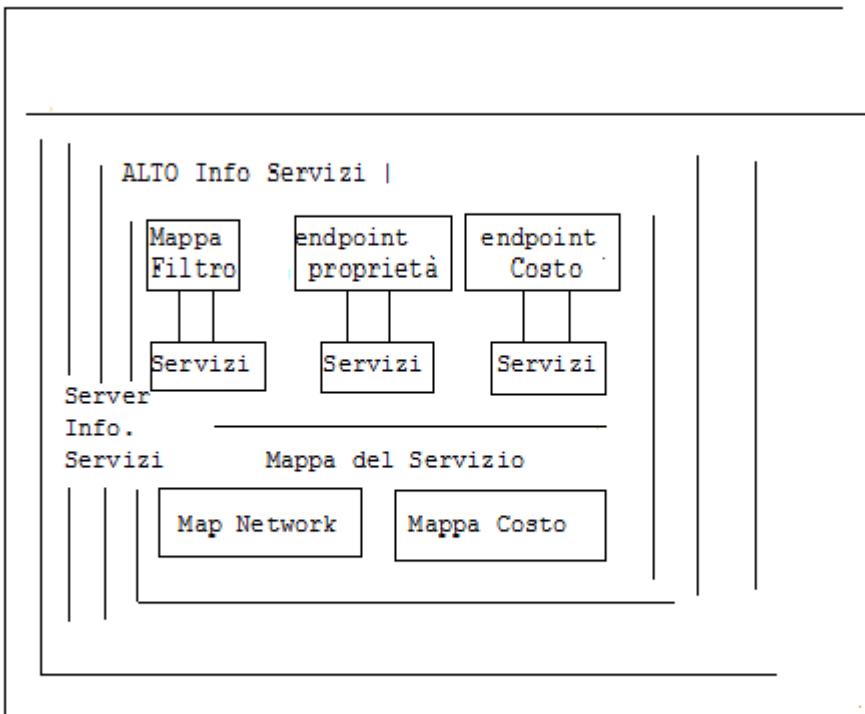


Figura 1: Struttura del protocollo ALTO

Il server ALTO invia informazioni di mappa che forniscono dettagli circa la scadenza. I messaggi scambiati con il client possono essere:

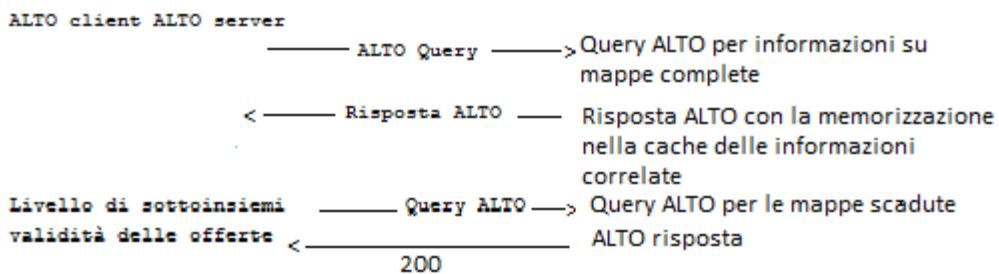


Figura 2: Flusso di messaggi

ALTO cache: si memorizzano nella cache le risposte ALTO per ridurre le ripetute richieste da parte del client. Può essere all'interno del client o in un nodo separato.

ALTO deve specificare l'ora di scadenza della mappa completa o di ogni livello delle mappe nella risposta; il formato tiene in considerazione l'ora GMT.

ALTO dovrebbe conoscere la scadenza della mappa nella cache e far scattare una nuova richiesta al server per aggiornarla.

Le cache sono utili quando viene mantenuta la consistenza della cache stessa, ci si base su coerenza debole e coerenza forte. Nella coerenza debole i documenti vecchi possono essere restituiti all'utente mentre nella coerenza forte i documenti non aggiornati non vengono mai restituiti all'utente. Ci devono però essere meccanismi di invalidazione forti o le richieste potrebbero essere mal indirizzate.

In tabella confronti di approcci di convalida diversi (Consideriamo che le richieste e le modifiche del file avvengono col seguente ordine: RRRMMRMRRMRRMMRM dove R è il numero di volte in cui una nel nodo arriva una risorsa e RI il numero di intervalli in cui il client chiede una risorsa mentre la risorsa è cambiata). La notifica viene inviata:

```
Messaggi | Polling | Notifica | approccio basato TTL |
| Ogni | approccio | |
Approccio | | |

Richieste GET | 1 | 1 | 1 |
If-Modified-Since | R-1 | 0 | TTLmissed - 1 |
Risposta 304 | R-RI | 0 | TTLmissed - |
| | | TTLmissedreschange |
Notifica Msg | 0 | 1 Sub + RI * | 0 |
| | Non |
Trasferimenti di file | RI | RI | RI - |
| | | Stalehitintervals |
Documento Stale | No | No | Sì |
```

Il contenuto del messaggio per gli approcci di consistenza

Tabella 1: Cache compromesso la coerenza

2.22 ALTO Abbonamento

L'obiettivo di ALTO è quello di fornire orientamenti per le applicazioni le quali devono selezionare gli host in grado di fornire una risorsa desiderata, a tal fine ALTO client deve interrogare il server per la mappa dei costi e di rete.

Distinguiamo alcuni casi:

- gli ISP allocano subnet all'interno della loro infrastruttura per assicurare un uso efficiente di IPv4
- si ha bisogno di notifiche per modifiche di informazioni: notifica per garantire la coerenza della cache forte.

Il funzionamento generale si basa sul client ALTO che interroga la mappa informazioni. Il server invia le informazioni di mappa le quali forniscono informazioni circa la scadenza.

Definiamo alcuni termini:

Evento: modifica di una risorsa che attiva una notifica.

Abbonamento: un rapporto che si stabilisce in cui una risorsa è interesse indicato in un evento

Abbonato: colui che si sottoscrive a un nodo ALTO.

Notifica: metodo con cui i nodi inviano informazioni in merito agli eventi.

Notificante: nodo che genera un messaggio di notifica.

Questo servizio consente ai client di sottoscrivere eventi in modo che il server può restituire mappe. Una directory information resource indica ai client quali risorse sono messe a disposizione da un server ALTO.

2.23 Notifiche ALTO

ALTO è stato progettato per consentire alle imprese con conoscenze sull'infrastruttura di rete di esportare informazioni alle applicazioni che devono scegliere gli endpoint per la connessione.

Considerando i casi in cui le informazioni non cambiano di frequente, tuttavia si ha sempre bisogno di informazioni un po' più fresche, pertanto ALTO deve essere dotato di un meccanismo di aggiornamento informazioni quale il Websocket. Il websocket è destinato a fornire biderizionalità all'HTTP. Una risorsa che scaturisce una notifica di aggiornamento può essere identificata da un URI websocket.

Quando un ALTO client desidera recuperare un'informazione messa a disposizione da un server deve:

recuperare l'IRD dell'ALTO service a cui sta per accedere;

trovare l'URI della risorsa interessata;

recuperare una copia della risorsa;

trovare l'URI del websocket per la notifica di aggiornamento del servizio associato a tale risorsa; stabilire una connessione websocket con l'URI del servizio notifica;

indicare il tag riguardante la versione della risorsa recuperata;

elaborare aggiornamenti successivi nel websocket.

Valuteremo come definire un meccanismo per indentificare l'URI del websocket, come far comunicare client e server circa la versione della risorsa memorizzata localmente e per la codifica degli aggiornamenti delle informazioni.

L'indicazione della versione può avvenire in due modi:

- Dopo che la connessione è stata stabilita col websocket con un annuncio ad hoc del client al server
- Nella connessione al websocket.

2.24 Pianificazione dei costi ALTO

Consideriamo i parametri che influiscono sulle prestazioni e sull'efficienza della trasmissione di dati. In questo modo si riesce a migliorare la Quality of Experience.

ALTO specifica una mappa che definisce gruppi di endpoint in una regione della rete. Il costo del servizio e l'endpoint servizio classifica sono in grado di fornire classifiche per i collegamenti tra le aree di rete. In tal modo ALTO evita la fornitura di informazioni in tempo reale, ma le applicazioni interrogano i servizi per recuperare i valori di costo attualmente in vigore.

Consideriamo applicazioni che hanno un grado di libertà per il trasferimento di dati in un periodo di tempo pianificato, non avendo quindi bisogno di avviare un trasferimento istantaneo su una richiesta di risorsa recuperata.

Fornire i valori di costo ALTO in modalità programma produce parecchi benfici. Un programma costo ALTO può essere utilizzato in diversi modi:

il server ALTO può fornire i valori su periodi precedenti.

Fornire valori in tempo presente o futuro come previsioni.

Fornire valori basati su periodi di tempo che coprono passato presente e futuro.

Alcune CDN sono cache precompilate con i contenuti disponibili agli utenti, vi è quindi un grado di libertà su quando il contenuto viene trasmesso dal server al nodo di caching. Per programmare trasferimenti si deve avere una buona conoscenza della domanda e dei collegamenti tra i centri dati diversi e la rete. La fornitura dei dati diventa più complessa con il numero di nodi CDN.

Applicazioni limitate dalla loro connettività hanno bisogno di sincronizzare il proprio stato periodicamente. Per le applicazioni non in tempo reale è conveniente fornire agli ALTO client il routing dei costi di connessione ai nodi e avere interazione con endpoint pianificata a intervalli regolari, quest'ultimo caso è utile quando si hanno risorse sparse in tutto il mondo. La pianificazione dei costi ALTO è un beneficio per l'ottimizzazione delle operazioni ALTO.

L'SDN (il software defined networking) è un modello che prova a gestire reti e a riconfigurarle in modo flessibile per affrontare meglio il traffico. Raccoglie informazioni sullo stato della rete e li presenta in forma astratta alle applicazioni.

Il protocollo ALTO consente di eseguire i seguenti servizi SDN:

- Astrazione
- Valutazione topologia di rete
- Funzionalità del dispositivo

SDN inoltre fornisce applicazioni con informazioni che permettono di valutare la QoE prevista. Uno degli obiettivi principali di un controllore SDN è la capacità di bilanciare il traffico delle applicazioni, per raggiungere questo obiettivo può:

ottenere la storia dello stato della rete

derivare una previsione dei valori su periodi di tempo indicati

conservare la loro astrazione in un ALTO server

fornire valori alle applicazioni SDN tramite gli endpoint.

Così le applicazioni possono ottenere il miglior QoE possibile e il controllore SDN raggiunge il bilanciamento del carico.

Si possono infine valutare costi in base al tempo, alla programmazione, alla capacità di Ambito.

2.24.1 ALTO Multi Cost

ALTO fornisce una guida per applicazioni di sovrapposizione. Questa previsione si basa su parametri che influiscono sulle prestazioni e sul rendimento della trasmissione di dati tra gli host. ALTO fornisce inoltre una mappa di rete e una mappa di costi e la classifica degli endpoint in base ai loro costi di routing.

Si considerano inoltre reti di contenuti, centri di dati e applicazioni che hanno bisogno di informazioni per selezionare gli endpoint. Ciò comporta l'introduzione di nuovi costi, per recuperare i quali bisogna inviare diverse richieste separate ai server.

Si rende utile ottenere tutti i costi con una singola query/risposta.

Studieremo tre parti: l'introduzione di nuovi tipi di costo, estensione per sostenere risposte sui tipi di costi multipli e il multi-cost ALTO.

Consideriamo il client ALTO incorporato in un application client, presumendo che le applicazioni che utilizzano ALTO possono effettuare la scelta di endpoint.

Parleremo di

Endpoint: peer

Endpoint discovery: i diversi tipi di processi utilizzati per scoprire i peer ammissibili.

Network Service Provider: fornitori di servizi Internet che forniscono mezzi per il trasporto dati e di Content delivery Network (CDN) che si prendono cura della diffusione dei contenuti.

Transazione ALTO: richiesta/risposta di scambio tra ALTO server e ALTO client.

Application Client: cliente di un'applicazione in esecuzione su un server virtuale.

Il protocollo ALTO prova a ottimizzare i costi di routing in reti di NSP. I tipi di costi dipendono da requisiti di QoE specifici per le applicazioni.

Tipi di costi ALTO e proprietà endpoint sono registrati in registri. Il registro di tipo di costo ALTO assicura che i tipi di costi che sono rappresentati da una mappa di costi sono univoci. Il registro endpoint ALTO garantisce unicità di identificatori degli endpoint.

Le continue richieste di trasferimento file hanno creato congestione del traffico a bassa velocità di uplink, allineando le topologie secondo il routing cost gli strati beneficiano di riduzione dei costi e migliore QoE.

Un altro caso riguarda il router di richiesta CDN prende una decisione sul nodo surrogato o sulla cache a cui inoltrare un contenuto di richiesta. In genere questa richiesta è basata sulla località.

Tramite il registro tipo di costo ALTO guida il CDN nella scelta cercando di selezionare il nodo migliore della cache.

In aggiunta alle cache CDN, per segnalare il carico degli endpoint possiamo avere le proprietà degli endpoint e i costi, quali:

EP capacity: la capacità che può essere nominale di memoria (memrizzazione del punto finale) e nominale di larghezza di banda (capacità delle risorse di calcolo).

Costo capacità endpoint occupato: che riflette le risorse disponibili e può essere diviso in memoria (capacità di archiviazione rimanente) e larghezza di banda occupata (risorse di calcolo rimanenti).

Un costo pathoccupationcost (POC): la banda utilizzata nel percorso utilizzato. Questo può avere valori che cambiano di frequente.

Diversi tipi di costo sono adatti per diverse applicazioni. Consideriamo pertanto nel servizio ALTO multi-cost che:

includono diversi tipi di costo quali costo mappa e costo di richiesta endpoint; forniscono una serie di valori del tipo di costo in una risposta anziché uno solo.

Tutto ciò per ottimizzare il tempo e la larghezza di banda.

Il servizio Costo endpoint (ECS) fornisce informazioni sui costi degli endpoint risorse delle applicazioni e delle risorse di rete utilizzate per accedere a tali endpoint. L'ECS può essere utilizzata anche in situazioni a breve termine ossia per richieste frequenti e per richieste di informazioni quasi istantanee.

I tipi di costo si aggiornano ciclicamente e nei tipi costo inoltre possono essere presenti più parametri. L'application client sfrutta al meglio i costi quando richiede mappe di costo filtrate per tipo di costo o per un insieme di coppie di peer.

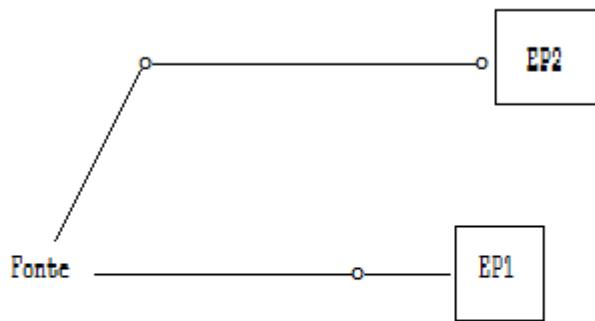
Eseguire query su tutti i valori di costo degli endpoint occupa più tempo che farlo in modo sequenziale, anche se diventa necessaria in caso di rapide modifiche.

2.25 Scelta Endpoint

Possiamo valutare un esempio in cui abbiamo due endpoint candidati EP1 e EP2. A T=1 si sceglie EP1 nonostante il maggior costo di instradamento perché ha un valore migliore (minore) di hopcount, e a T=2 si ha l'aggiornamento di routingcost e hopcount. Avendo entrambi lo stesso valore di hopcount ma EP2 un costo di instradamento maggiore l'applicazione si rende conto che è più conveniente interagire con EP1 e passa ad EP2 (che ha costo inferiore di routing):

T = 1: EP1: routingcost = 40, hopcount = 2
EP2: routingcost = 30, hopcount = 3

EP1 è stato selezionato perché l'applicazione è sensibile al tempo e metrico 'hopcount' ha un peso maggiore



$T = 2$: EP1: routingcost = 40, hopcount = 3

EP2: routingcost = 30, hopcount = 3

- Percorso a EP1 è cambiato. Hopcount è ora 3

\Rightarrow EP2 è stato selezionato perché routingcost è inferiore a quello EP1, con lo stesso valore hopcount

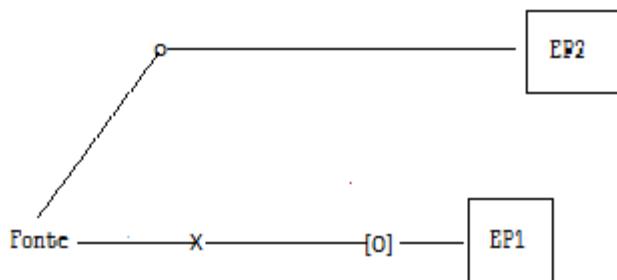


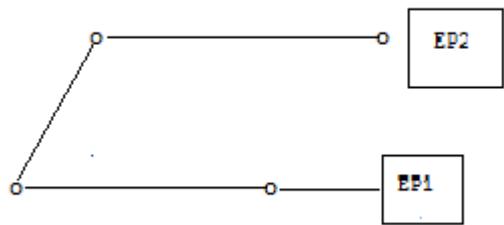
Figura 1: Endpoint ri-selezione utilizzando Multi-Cost su richiesta ALTO valori di costo aggiornati, su chnage nel percorso.

Un altro esempio:

$T = 1$: EP1: routingcost = 30, hopcount = 2

EP2: routingcost = 30, hopcount = 3

\Rightarrow EP1 è stato selezionato perché l'applicazione è sensibile al tempo e metriche hopcount ha maggiore peso



T = 2: EP1: routingcost = 40, hopcount = 2

EP2: routingcost = 30, hopcount = 3

Routingcost a EP1 è aumentato. Hopcount è la stessa.

==> Applicazioni ritardo sensibili che vogliono ridurre al minimo hopcount

rimanere con EP1, mentre altre applicazioni possono restare
con EP2, che ora ha un routingcost inferiore.

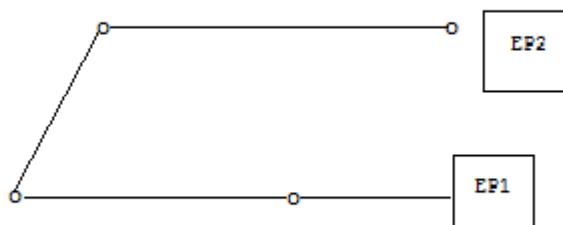


Figura 2: Endpoint selezione utilizzando due tipi di costo con la richiesta congiunta
valori di costo aggiornati e per le applicazioni sensibili ritardi.

Per consentire l'esecuzione del multi-cost il servizio ALTO deve effettuare alcune modifiche nel protocollo di base.

Il costo singolo qualifica le voci in base al progetto del protocollo. Il cambiamento fondamentale per il multi-cost è quello di codificare il costo con diversi di costo.

Consideriamo costi-mode e costi-type che elencano in quali tipi di costo ci troviamo e in quale modalità questo tipo di costo è rappresentato.

Il costo di ciascuna sorgente/destinazione deve essere fornito nello stesso ordine nella matrice tipi di costo. La matrice di cost-mode deve essere della stessa dimensione della matrice cost-type.

Nei multi-costo il valore numerico rappresenta il valore del costo.

Ci sono anche estensioni del protocollo utili a sostenere il servizio ALTO multicosto e forniscono ulteriori ALTO informazioni. Se un Client desidera informazioni sui tipi di costi diversi può chiedere e ricevere tutti i tipi di costo desiderati in una singola transazione.

Alcune estensioni sono:

informazioni o risorse di repertorio

Cost map service

Mappa o costo filtering service

Servizio Endpoint o costo di ricerca.

2.26 Aggiornamenti incrementali

L'obiettivo di ALTO è quello di colmare il divario tra la rete e le applicazioni di provisioning correlate. Ciò consente alle applicazioni di fare decisione informate.

Un server ALTO fornisce servizi che guidano gli ALTO client nelle loro decisioni. Il property service endpoint consente ai clienti di cercare le proprietà degli endpoint. Il costo Endpoint consente di classificare gli endpoint del server tra loro rispetto ai costi. Il servizio mappa e la mappa filtering service consente di recuperare ai client le mappe di rete e le mappe dei costi associate.

L'ALTO map Network contiene gruppi di endpoint ottenuti aggregando endpoint vicini tra loro che forniscono una maggiore scalabilità. A ogni gruppo di endpoint è associata una posizione identificatore di rete chiamata PID. La dimensione della rete ci dà informazioni riguardo la granularità della mappa che un server ALTO prevede per i suoi client. Per evitare la trasmissione delle stesse informazioni in ogni richiesta del cliente si è studiato un meccanismo che consente a un server di inviare aggiornamenti incrementali. Valutiamo le varie opzioni che consentono tali aggiornamenti incrementali dei costi di rete e delle mappe.

Il protocollo ALTO utilizza mappe di rete e dei costi per consentire ai server ALTO di specificare la propria posizione nella rete. Essenzialmente la mappa di rete contiene informazioni su come gli endpoint sono raggruppati. La mappa dei costi contiene i costi dei percorsi tra le aree di rete. Modifiche alla mappa di rete ha un impatto sulla mappa dei costi non è invece vero il contrario. Consideriamo un client e un server ALTO che possono sincronizzare il loro stato senza trasmettere l'intera mappa e come possono essere inviati aggiornamenti parziali al client.

Opzioni per la convalida degli aggiornamenti incrementali si basano sull'HTTP che viene utilizzato per inviare le richieste.

Mentre per quanto riguarda gli aggiornamenti parziali ci sono diversi modi per farlo. Un modo banale è quello di inviare l'intera mappa. Un server che conosce la mappa che un client ha già nota le differenze tra le mappe. Gli aggiornamenti parziali si usano per ridurre il tempo di trasmissione riducendo la quantità di dati che deve essere trasferita al client. Ciò si può fare attraverso l'utilizzo di applicazioni di compressione.

Un'altra opzione è quella di offrire un servizio dedicato per gli aggiornamenti parziali. Si può interrogare il servizio per recuperare l'aggiornamento parziale.

Capitolo 3

Altre Architetture

3.1 ONO Project

In ONO Project il plugin contatta un server CDN; si ottiene il server CDN più vicino per cui quando il peer deve scegliere il peer più vicino fa cui scaricare confronta il server assegnatogli dalla rete CDN con il server di ogni peer (che ha nel suo elenco) per cui se i server sono uguali i peer sono vicini (i più vicini).

È un servizio software che permette il file sharing tramite identificazione efficiente di peer nelle vicinanze. L'utilizzo di peer vicini localmente aumenta la velocità di download di file contemporaneamente (producendo un aumento tra il 31% e il 207%).

Utilizza un plugin che apprende quali peer si trovino nelle vicinanze da reti di distribuzione di contenuti (CDN).

3.2 CDN

Content Delivery Network rappresenta un sistema di computer collegati in rete attraverso internet che collaborano sotto forma di sistema distribuito per distribuire contenuti agli utenti finali ed erogare servizi di streaming audio e video.

I nodi CDN sono distribuiti geograficamente e collaborano tra loro per soddisfare le richieste di contenuti trasferendoli al fine di ottimizzare il processo di consegna. Vantaggi sono riduzione dei costi per ampiezza di banda o il miglioramento delle prestazioni.

L'obiettivo di un CDN è di instradare una richiesta di contenuto sul nodo che viene individuato come ottimale, bisogna valutare se quello ottimale sia il nodo meno costoso o quello più vicino.

3.3 DECADE^[2]

Un sistema DECADE fornisce meccanismi di ricerca che permettono una configurazione automatica dei client DECADE con tutte le informazioni necessarie utili a contattare i server DECADE appropriati della rete.

Queste informazioni includono il dominio o l'indirizzo IP di ogni server DECADE. Valuteremo come un client DECADE possa ottenere informazioni sulla posizione del server tramite DHCP.

Il Decoupled Application Data Enroute (DECADE) architecture definisce un sistema comprendente applicazioni di distribuzione di contenuti in grado di migliorare lo scambio di dati in Internet.

Le applicazioni all'interno di DECADE sono in grado di memorizzare e recuperare dati nella rete e utilizzare DECADE server per la loro gestione.

Tale sistema può comprendere numerosi server distribuiti in diverse parti della rete. In questo sistema un client deve essere in grado di individuare tutti i server che può utilizzare. Per applicazioni endpoint se un dato viene caricato in un certo server si è in grado di notificare ai peer la posizione del server in cui sono stati memorizzati i dati fornendo tutte le credenziali necessarie per accedervi.

Definiremo il meccanismo di determinazione del server da utilizzare per un client DECADE. Si introduce un server DECADE di scoperta che specifica come un client può ottenere informazioni sulla posizione del server tramite DHCP.

3.3.1 DECADE Discovery Server

Un sistema DECADE deve includere un meccanismo di rilevamento del server. I client DECADE possono ottenere informazioni di configurazione del server tramite altri mezzi, per esempio può essere preconfigurato per accedere a un particolare server attraverso al suo domain name (FQDN).

In aggiunta l'amministratore di rete può richiedere al sistema DECADE informazioni riguardo i noti autorizzati a collegarsi alla rete. Si possono infatti avere server raggiungibili solo da nodi di un particolare dominio. La distribuzione del server basato su indicazioni geografiche può portare vantaggio migliorando prestazioni e sicurezza, nonché la gestione di eventuali attività di manutenzione. Il client potrebbe essere accantonato con un backup delle informazioni DECADE circa le posizioni del server per bilanciare il carico del traffico.

La configurazione automatica del client DECADE è utile per utenti che viaggiano, infatti gli abbonati a un provider di servizi internet godono di connettività a banda larga residenziale usando un server ISP DECADE se connessi da casa e se in vacanza possono usufruire di un altro server a metà strada ovunque si trovino. L'amministratore di rete è in grado infatti di offrire DECADE servizi a tutti i peer connessi alla propria rete, magari chiedendo agli utenti di immettere manualmente le informazioni sulla posizione. Tuttavia ciò è soggetto a errori. Sarebbe meglio se il client DECADE fosse in grado di rilevare automaticamente informazioni sulla posizione del server nella rete.

Studieremo ora come un client DECADE riesce ad ottenere il domain name del server DECADE e/o l'indirizzo IP in base alle preferenze dell'amministratore di rete. Il client DECADE utilizza il sistema DECADE per decidere quali server impiegare sulla base di diversi fattori quali le preferenze dell'utente, la fiducia, la privacy, la sicurezza.

3.3.2 Server Discovery DECADE con DHCP

Un client DECADE può utilizzare DHCP per ottenere informazioni sulla posizione del server nella rete.

Si definiscono due opzioni DHCP: uno ordinato per ottenere un elenco ordinato il domain name dei server DECADE e un altro per ottenere gli indirizzi IP degli stessi.

Il DHCP può essere attivato all'avvio del dell'applicazione o in qualsiasi momento del suo funzionamento. Un client DECADE può indicare il client DHCP sul nodo del server configurato.

In un nodo con supporto per protocolli DECADE è possibile scoprire server DECADE all'avvio così come quando cambia il punto di collegamento di rete o quando si attivano nuove interfacce.

Un client DHCP utilizza le due opzioni e le può inviare al client DECADE compatibile. Il client DHCP configura le informazioni del server DECADE e dispone il client DECADE sul nodo.

Un server DHCP può inviare l'elenco del domain name e degli indirizzi IP anche se il client DHCP non ne fa esplicita richiesta; deve inviare l'elenco del domain name e può inviare gli indirizzi IP se il client richiede al server DECADE la lista del domain name; e deve inviare la lista degli indirizzi IP e può inviare il domain name se il client richiede al server DECADE gli indirizzi IP.

Il server DHCP può utilizzare l'identificatore univoco DHCP del client DHCP per determinare il contenuto del server DECADE che deve essere restituito.

Domain name e indirizzi IP compaiono più di una volta in ogni messaggio DHCP, e nello specifico contengono nei messaggi di sollecito, pubblicità, richiesta informazioni, rinnovo e risposta.

L'implementazione del server DHCP è in grado di eseguire la ricerca del domain name per conto del client DHCP e restituire l'indirizzo IP del server DECADE quindi riducendo il carico di segnalazione in termini di risoluzione. Tuttavia è più probabile che un client richieda quale server DECADE utilizzare basandosi sul domain name in quanto ciò è preferibile.

In base alle informazioni ricevute un client DECADE può decidere di ignorare il server DECADE suggerito dall'amministratore di rete, se invece lo utilizza deve prendere in considerazione tutti i domain name della lista ricevuta prima di avviare le connessioni al server DECADE.

Un client DECADE dovrebbe stare in guardia quando riceve le informazioni che vengono fornite perché si rischia di contattare un server dannoso e gli operatori potrebbero ottenere l'accesso a informazioni sensibili o private e inquinare i dati. DHCP supporta un meccanismo di autenticazione che contrasta l'attacco di man in the middle.

DHCP viene utilizzato per la fornitura di informazioni ai nodi di applicazione. Poiché il percorso del server può essere utilizzato da qualsiasi applicazione, DHCP può essere utilizzato per fornire ulteriori opzioni del server DECADE alle applicazioni compatibili. Tuttavia la disponibilità delle informazioni sul server non significa che il client ha accesso per archiviare e recuperare i dati dal server. Bisogna autenticarsi e ottenere l'autorizzazione.

3.3.3 DECADE vs CDN

CDN sono utilizzati per fornire file generati dal Content Service Provider agli utenti finali; tuttavia un singolo CDN è vincolato in copertura geografica e per quanto riguarda il volume delle risorse. Il gruppo DECADE si propone di introdurre una memoria di rete utile ad alleviare colli di bottiglia della larghezza di banda. Questa memoria può essere usata dal CSP per fornire i contenuti utilizzando un CDN. Un CSP può infatti inviare i propri contenuti al portale DECADE fornire dall'ISP il quale distribuirà il contenuto ai server DECADE in più sedi. Quando viene richiesto il contenuto a CSP, questo viene inviato al portale DECADE che poi seleziona il server ottimale in cui inoltrarle. In questo modo DECADE fornisce un servizio simile ad un CDN in ISP.

3.3.4 Confronto di DECADE con CDN^[3]

DECADE e CDN possono utilizzare ALTO come servizio per prendere decisioni. In DECADE, ALTO aiuta DECADE per decidere qual è il miglior server per gli utenti finali. In CDN, un CDN può utilizzare ALTO per decidere a quale CDN inviare le richieste del client. DECADE inoltre può essere utilizzato da CDN per lo scambio di messaggi di controllo, acquisizioni di oggetti contenuto in CDN diversi e per la distribuzione di contenuti all'interno di un CDN dato.

Grazie alla memoria di rete fornita da DECADE gli ISP possono distribuire facilmente i server DECADE nelle loro reti. Quando si hanno più ISP, un CSP può solo sottoscrivere il servizio DECADE fornito da un ISP e può fornire i contenuti ai server DECADE attraverso questi ISP multipli. Ci dovrebbe essere un modello di caricamento grazie al quale gli ISP possono caricare i contenuti che servono agli altri ISP. Il lato negativo è che gli ISP dovranno implementare i server

DECADE anche per provider che non usufruiscono e non hanno il servizio DECADE. CDN invece possono sfruttare ciò senza implementare nuovi server il che comporta un costo di distribuzione inferiore.

3.3.5 Integrazione ALTO e DECADE

L’architettura DECADE fornisce un quadro generale dell’interfaccia standard per il controllo delle risorse e dei dati di trasporto.

ALTO server: entità logica che fornisce interfacce per la query al servizio ALTO

Server DECADE: archivia i dati DECADE all’interno della rete e poi gestisce sia i dati memorizzati e l’accesso a tali dati.

DECADE Client: memorizza e recupera i dati

App tracker: server che memorizza l’elenco dei colleghi che condividono le risorse di rete.

In figura è mostrata la struttura di base dell’architettura di integrazione ALTO e DECADE. Indica che le due integrazioni sono in grado di fornire traffico di rete

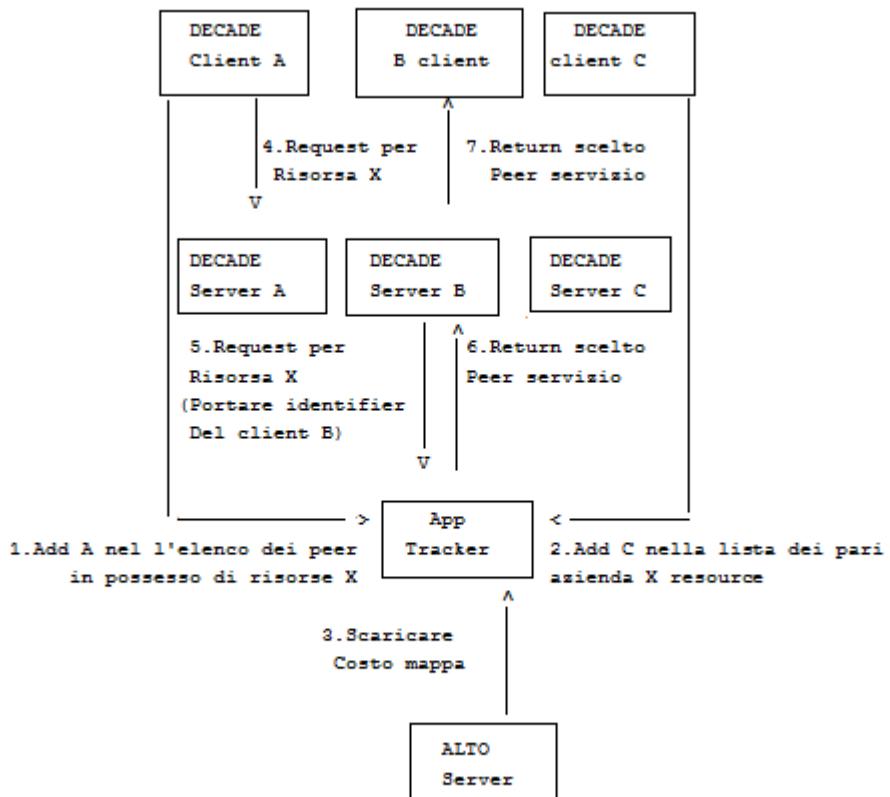


Figura 1 ALTO e Integrazione quadro DECADE Architettura

Il client DECADE chiede determinate risorse di rete e se il server le contiene la richiesta sarà soddisfatta celermente. Altrimenti il server interroga il tracker App chiedendo quale peer contenga tali risorse. App tracker riceve la richiesta e interroga tutti i peer, dopo dichè guarda la mappa di rete per controllare se la risorsa sia stata già scaricata dall'ALTO server. L'interazione tra ALTO e DECADE porta parecchi vantaggi quali il risparmio di larghezza di banda della rete.

La figura 2 mostra il flusso di processo di integrazione DECADE e ALTO.

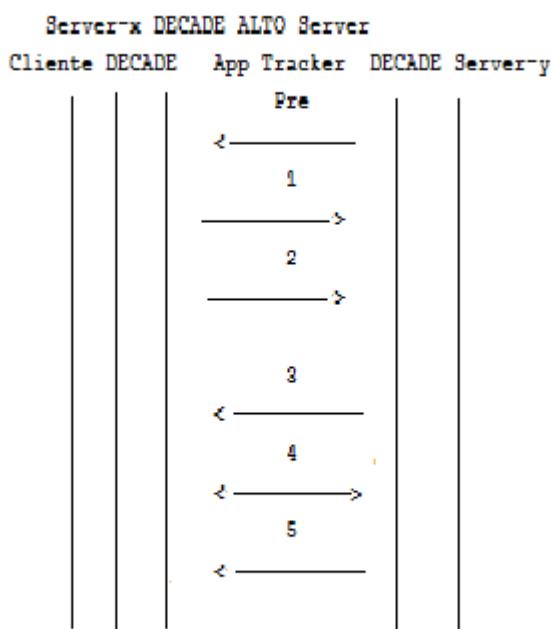


Figura 2 Flusso del processo di integrazione e di DECADE ALTO

Pre: App tracker scarica la mappa dal server ALTO in anticipo

1. Client DECADE richiede risorse al suo server
2. Se il server non le contiene chiede suggerimento all'App tracker
3. App Tracker guarda tutti i peer che contengono tali risorse e calcola il costo.
4. Il server scarica i dati da un altro server
5. Il client scarica i dati dal suo server.

La figura 3 mostra un esempio in cui il client DECADE B vuole ottenere un file contenuto nel server DECADE A e C. Senza l'integrazione ALTO potrebbe ottenere il file casualmente tramite il suo server da uno dei due server. Ma in questo caso il Server consulta l'App che indica il peer che può offrire il servizio più adatto. ALTO server fornisce la mappa dei costi e la mappa di rete, in base alle quali può essere calcolato il costo e viene selezionato il peer che server al minor costo:

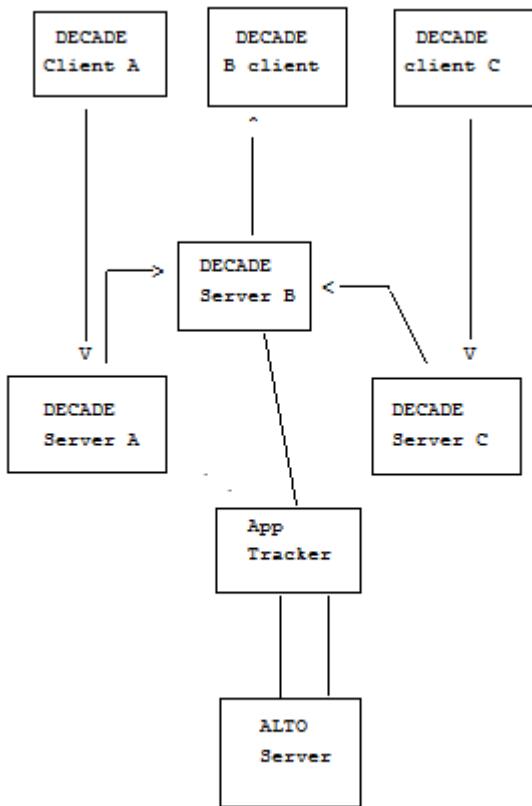


Figura 3 La condivisione di file P2P

In figura 4 l'esempio in cui il client B sta trasmettendo uno streaming in diretta. Un client è in linea e guarda il live e il Server DECADE A scarica il trasferimento di dati dal Server DECADE B in tempo reale. Client C è in linea e vuole vedere il flusso in un secondo momento, può pertanto scaricarlo da uno degli altri due server dopo il confronto dei costi tramite la consultazione di App Tracker e ALTO server. Dopodichè il Server C conterrà i dati che serviranno al client C che una volta on line potrà ottenerli:

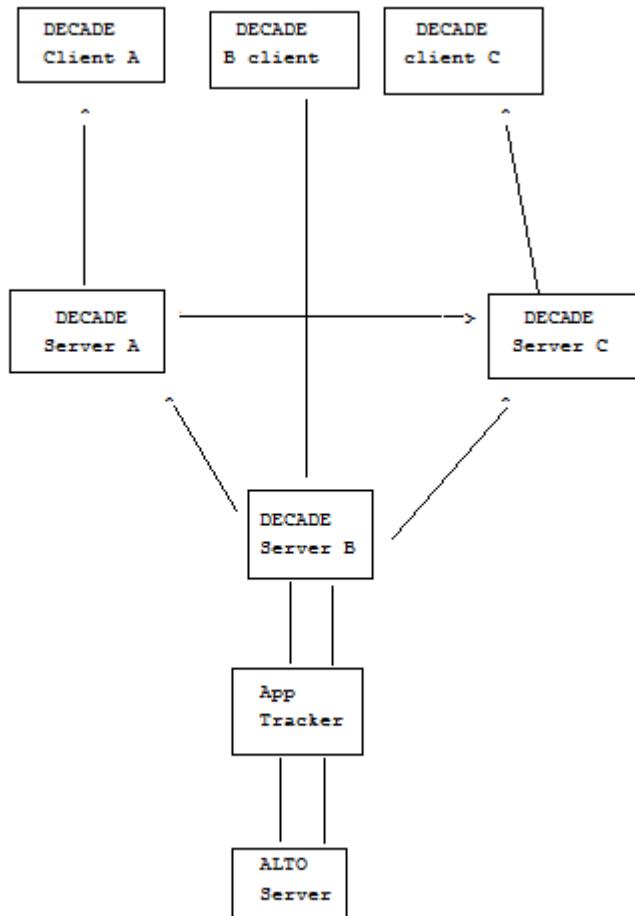


Figura 4 Offline Streaming P2P in diretta

App tracker può avere bisogno di informazioni per eseguire la selezione tra i peer. La figura 5 mostra le interazioni tra le entità principali dell'integrazioni in cui l'App tracker è un client ALTO:

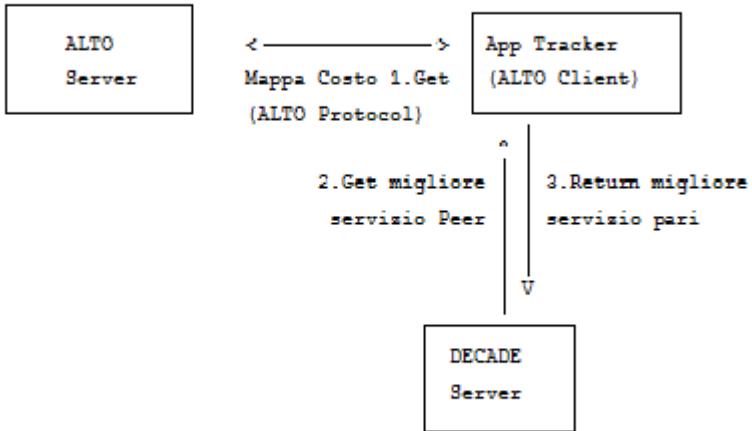


Figura 5 ALTO client integrato in App Tracker

1. L'App tracker richiede la mappa dei costi di tutti i PID dell'ALTO server
2. Un Server DECADE serve i peer al minor costo
3. L'App tracker restituisce il peer col miglior servizio.

App tracker funge da client ALTO.

3.4 SDN

Il concetto di Software Defined Network è nato ed è diventato fondamentale per le reti che consentono flessibilità di controllo, separazione dei piani funzionali e continua evoluzione.

È necessario che ci sia la separazione dei piani funzionali, di controllo e di inoltro dati, ma deve esserci anche interazione tra i due piani che ha il compito di eseguire specifici protocolli consentendo una combinazione libera dei dispositivi di rete e controller.

Alcuni vantaggi dell'SDN sono la virtualizzazione di rete, un maggiore controllo flessibile della rete, reti personalizzate. Alcuni SDN sono stati utilizzati in DCNS Data Center Networks.

Con SDN, ALTO deve affrontare problemi di privacy, conservazione, granularità della raccolta di informazioni e di scambio, ottimizzazione.

Un dominio SDN è una porzione di una infrastruttura di rete costituito da numerosi dispositivi di rete collegati che sono in grado di implementare il controllo/inoltro dei piani.

Le applicazioni nelle reti di software sono SDN-aware o inconsapevoli di SDN:

Se un'applicazione è SDN-aware l'applicazione preferisce una comunicazione diretta con i controllori SDN.

Se un'applicazione è inconsapevole di SDN l'applicazione non comunica direttamente con i controllori SDN ma tramite datagrammi di applicazione con formati specifici.

Un operatore di rete può decidere di dividere la rete in sotto-reti, il che porta numerosi vantaggi:

Scalabilità: per gestire un numero di dispositivi che non possono essere messi sotto controllo di un singolo SDN controller.

Gestibilità: A livello di rete al fine di ridurre la complessità di gestione.

Privacy: ogni sotto-rete può implementare politiche di privacy differenti.

Distribuzione: ci deve essere una porzione di infrastruttura legacy e una non-legacy.

3.4.1 Interazione con ALTO

Per la coesistenza di ALTO e SDN si adottano due architetture:

architettura verticale che permette migliore riportazione, gestione, flessibilità, controllo della privacy ed evoluzione a lungo termine della rete.

Architettura orizzontale che semplifica l'implementazione di estensioni ALTO per SDN.

L'architettura verticale è costituita da un'architettura gerarchica di 3 livelli: nel primo livello ci sono gli SDN controllori, nel terzo domini SDN.

L'interazione tra controllori SDN e ALTO possono essere suddivise in due categorie:

verso l'alto: ogni controllore SDN raccoglie le informazioni dai dispositivi di rete e le riferisce al server ALTO.

Verso il basso: ogni controllore SDN è un client ALTO e recupera le informazioni rilevanti dal server ALTO.

L'architettura orizzontale il controllore SDN può agire come un ALTO client.

Capitolo 4

Confronti

4.1 Confronto ALTO – altre architetture

Il principale vantaggio di ALTO è dato dal principio di selezione dei peer che ha, innanzitutto, rappresentato una svolta nel passaggio dal P2P al P4P ed è stato il principale obiettivo di tutte le architetture in fase di studio che utilizzano il P4P per il trasferimento di file via internet.

La selezione dei peer avviene basandosi sia sulla misura della distanza che va dalla sorgente alla destinazione e dall'indice di preferenza. Utilizza una memoria (cache) in cui sono inserite le risorse/i documenti.

Si evitano colli di bottiglia, si ha più larghezza di banda disponibile.

Migliora il traffico tramite la conoscenza delle informazioni di rete e l'esperienza dell'utente data dal miglioramento prestazioni e dall'efficienza della trasmissione dei dati.

Problemi di sicurezza a causa informazioni incorrecte. Eccessiva divulgazione dati (per la quale si utilizza un filtro) – invio dati a terzi non autorizzati. Si prova a ovviare stabilendo un range di indirizzi a cui inviare le risposte o richiedere schemi di crittografia.

ONO aumenta la velocità di download grazie all'utilizzo di peer vicini geograficamente e costi ridotti ma abbiamo un impatto limitato, le performance sono ridotte e si ha un degrado della robustezza della rete e in più un conflitto di interessi^[4] in quanto i costi sono ridotti solo per alcuni ISP.

CDN migliora il trasferimento dei file in rete ottimizzando la consegna portando come vantaggi la diminuzione dei costi per ampiezza di banda o il miglioramento delle prestazioni. Abbiamo però nella scelta dei peer venga preferito una delle ottimizzazioni piuttosto che l'altra in quanto CDN è vincolato in copertura geografica.

DECADE bada al miglioramento dello scambio dei dati tramite applicazioni di distribuzioni di contenuti. Si utilizza una memoria in cui si inseriscono i dati. Sono i peer che vanno alla ricerca del server. I rischi sono quelli di contattare server dannosi e si potrebbero ottenere informazioni private. Viene utilizzato un sistema di autenticazione e la crittografia utilizzata resiste agli attacchi di man in the middle.

SDN grazie alla separazione dei piani di controllo porta come vantaggi la virtualizzazione di rete, controllo flessibile delle reti e reti personalizzate. C'è però da affrontare il problema legato alla conservazione, alla granularità delle informazioni e all'ottimizzazione anche per il fatto che tali algoritmi cambiano di sotto-rete in sotto-rete. Anche se quest'ultimo punto può portare al miglioramento della gestione e ai processi di privacy.

Nel dettaglio vediamo:

4.1.1 Peer selection:

ALTO: usa una Network Map in cui sono raggruppati gli endpoints e nella quale viene mostrata la loro vicinanza e definisce le preferenze tra luoghi di rete tramite Path Costs che sono indicati nella Cost Map. L'unione delle due fa stabile al protocollo un indice di preferenza attraverso il quale vengono scelti i peer.

ONO: un server seleziona i peer a cui fornire risorse valutandone la vicinanza geografica. Infatti due peer vengono considerati vicini se associati allo stesso server.

CDN: valuta quale sia preferibile tra un minor costo del percorso per ampiezza di banda o la vicinanza geografica.

DECADE: si utilizza DHCP per avere informazione della posizione dei server tramite il domain name e gli indirizzi IP.

SDN: si creano sotto-reti per le quali i peer fanno riferimento ad unico server.

Per quanto riportato ALTO è il protocollo che ci offre una prospettiva migliore in quanto presenta una scelta oculata e ben studiata, stesso discorso dicasi per DECADE. Le scelte di ONO e SDN sono piuttosto limitate geograficamente; mentre CDN non sempre riesce ad attuare la scelta ottimale, ma deve valutare quale sia conveniente.

4.1.2 Sicurezza

ALTO: Si può incorrere in eccessiva divulgazione dei dati e di informazioni confidenziali; a ciò si porta rimedio stabilendo un range di indirizzi a cui inviare le informazioni.

ONO: si incappa nella divulgazione di dati errati che incidono sul degrado della robustezza della rete.

DECADE: si rischia di contattare server dannosi ottenendo informazioni private o si potrebbero inquinare i dati.

SDN: attua un meccanismo di controllo delle sotto-reti.

Il problema della sicurezza è molto importante, il rischio maggiore è quello di ottenere informazioni delicate e divulgarle in giro. ALTO prova a stabilire range di indirizzi, DECADE lascia che il client "senta" il suo server e acceda solo alle informazioni da esso tramandate. SDN fornisce, invece, una maggiore efficienza e grazie alla separazione dei piani di controllo e inoltro dati riesce ad avere un controllo flessibile della rete.

4.1.3 Privacy:

ALTO: terzi potrebbero inserirsi nello scambio di informazioni, si richiedono autenticazione e sistemi di crittografia all'accesso del client e bisogna crittografare il client.

DECADE: viene utilizzato un sistema di autenticazione e crittografia resistente agli attacchi di man in the middle.

SDN: il problema di privacy viene affrontato in maniera autonoma da ogni sotto-rete che può utilizzare sistemi differenti, risultando pertanto molto sicura.

La privacy è un altro punto su cui c'è massima attenzione, si utilizzano in generale sistemi di crittografia. SDN ci offre maggior sicurezza.

4.1.4 Riduzione dei costi:

ALTO: è l'obiettivo principale del protocollo che valuta attentamente la distanza tra una sorgente e una destinazione ottimizzando il processo di invio dati basato su percorsi che hanno indice di preferenza migliore.

ONO: impatto limitato, si ha un conflitto di interessi in quanto i costi sono ridotti solo per alcuni ISP.

CDN: si hanno minori costi per ampiezza di banda, ma spesso si preferisce migliorare il traffico a discapito dei costi.

DECADE: insito nella scelta dei server, anche se il client preferisce guardare la sicurezza, la privacy o le preferenze dell'utente.

Il protocollo ALTO è quello che sotto questo profilo fornisce maggiori garanzie grazie all'algoritmo di peer selection. ONO e CDN offrono soluzioni limitate, DECADE valutando gli indirizzi IP dovrebbe aver utilizzato percorsi a minor costo ma a volte si preferiscono altre vie.

4.1.5 Miglioramento del traffico:

ALTO: utilizza applicazioni con parecchia banda che evita colli di bottiglia.

ONO: anche se teoricamente si dovrebbe avere un aumento della velocità dei file nello specifico si hanno performance ridotte.

CDN: si ha miglioramento delle prestazioni ma a volte viene scelto di badare più ai costi rallentando di fatto il processo.

DECADE: viene utilizzato il DHCP che migliora la distribuzione dei contenuti.

SDN: le sotto-reti garantiscono aumento delle prestazioni, ogni sotto-rete risulta efficiente.

Il punto principale è quello di evitare lunghi ritardi nelle distribuzioni dei file. ALTO aumenta l'ampiezza di banda, DECADE fa gestire la distribuzione al DHCP, SDN utilizza una virtualizzazione di rete. CDN lo pone come obiettivo nel peer selection ma non sempre ciò viene

rispettato e infine ONO nonostante punti a velocizzare il tutto incappa spesso in un rallentamento del processo.

4.1.6 QoE:

ALTO: viene migliorata l'efficienza della trasmissione dati aumentando la quality of experience dell'utente.

CDN: si soddisfano le richieste di contenuti andando ad ottimizzare quello che è il processo di consegna dei dati.

SDN: si hanno reti personalizzate utili agli utenti che fanno parte di sotto-reti.

Ci si concentra a migliorare la QoE grazie ad un miglioramento delle trasmissioni dei dati. ALTO, CDN e SDN utilizzano metodi differenti ma utili a implementare ciò.

4.1.7 Memoria:

ALTO: viene fornita per inserire i dati che poi vengono inviati agli utenti, non c'è bisogno di ricercarli in caso di nuova richiesta. È utilizzata per recuperare le risorse, ma grazie ad essa si ha una migliore conoscenza delle informazioni di rete.

DECADE: i dati vengono qui inseriti per evitare nuove richieste, velocizza lo scambio dati.

SDN: utilizzata per migliorare il processo, si incappa nel problema della conservazione delle informazioni e questa deve essere refreshata di volta in volta.

La memoria è un punto in favore dei protocolli che la utilizzano; abbiamo però notato che SDN ha bisogno di un continuo aggiornamento della stessa.

4.2 Tabella riassuntiva

protocolli caratteristiche	ALTO	ONO	CDN	DECADE	SDN
selezione dei peer	basato sui costi	peer vicini	vincolata in copertura geografica	peer alla ricerca dei server	
sicurezza	informazioni a volte incorrette	degrado robustezza di rete		divulgazione informazioni private	controllo delle reti
privacy	necessita crittografia			autenticazione e crittografia	ottima - diversa per ogni sotto-rete
riduzione dei costi	basata su distanza sorgente e destinazione	impatto limitato	meno costi per ampiezza di banda		
miglioramento del traffico	no colli di bottiglia	performance ridotte	velocità trasferimento file	applicazioni di distribuzione contenuti	virtualizzazione della rete
aumento QoE	efficienza		ottimizzazione consegna		reti personalizzate
memoria	recupero risorse conoscenza info di rete			utile per lo scambio dati	problema conservazione informazioni

Si evince che non c'è un protocollo migliore per eccellenza, ma ci potrebbe essere un'interazione tra tutte le architetture. Partendo da ALTO infatti potremmo utilizzare nel peer selection anche la ricerca dei server indicata dal DECADE e per sicurezza e privacy potremmo fare affidamento sull'SDN. E in più si potrebbero unire i vantaggi di ogni protocollo per avere sempre un minor costo e un aumento delle prestazioni.

Capitolo 5

Conclusioni

Le soluzioni P4P, per essere pienamente operative, richiedono la cooperazione tra fornitori e utenti P2P.

La cooperazione è fattibile se i benefici sono multi-laterali

La cooperazione tra gli utenti può essere raggiunta solo fornendo vantaggi significativi, ad es
velocità di download più elevate

Problemi di privacy e la censura sono particolarmente importanti per molte comunità degli utenti
P2P.

Abbiamo visto come ALTO valuti gli indirizzi IP dei peer afferenti alle rete scegliendo quali preferire e quali evitare, valuti la scelta dei peer fornendo elenchi ordinati come richiesto dal client resi disponibili dalla vicinanza topologica e geografica delle posizioni.

Al fine di interrogare i server ALTO, i client devono prima conoscere uno o più server che sono in grado di fornire informazioni. Ci siamo focalizzati su come vengono individuati i servizi in uso.

Per quanto riguarda la sicurezza delle reti abbiamo visto come ALTO chieda autenticazione e servizi di crittografia al fine di evitare la diffusione di informazioni private.

Il concetto di bandwidth ci è utile nel miglioramento del traffico, infatti aumentando la larghezza di banda si evitano colli di bottiglia.

Abbiamo valutato il processo di recupero risorse che vengono inserite in una memoria grazie alla quale ALTO non deve ricercare le informazioni se già chieste.

I meccanismi di ricerca possono essere riutilizzati, in tal caso si utilizza una specifica per individuare i server che permettono ciò.

Abbiamo valutato come possono comunicare tra loro applicazioni e server ALTO. I servizi ALTO possono essere utili in ambienti client-server ma anche in ambienti P2P.

Altro punto principale di studio è stato il meccanismo di riduzione dei costi, come già visto nel peer selection ALTO tramite specifici algoritmi tiene bassi i costi senza portare un degrado della prestazioni.

Abbiamo infine presentato nuove architetture, quali ONO, che utilizza un plugin; CDN, che si basa sulla vicinanza geografica dei peer; DECADE, che utilizza il server discovery, in cui i client ricercano il loro server tramite il domain name o gli indirizzi IP che gli vengono forniti; e infine SDN che suddivide la rete in sotto-reti separando i piani funzionali. Abbiamo valutato i vantaggi e

gli svantaggi di ognuno e anche come possa essere possibile una loro interazione col protocollo ALTO.

Si possono valutare eventuali siluppi futuri. Un primo passo è l'interazione, la coesione dei protocolli visti, che già in parte avviene.

Ci si potrebbe poi concentrare su alcuni aspetti non trattati in questo lavoro.

Quando le informazioni che il server ALTO può fornire sono standardizzate si valutano alcuni criteri per assicurare la reale fattibilità del protocollo ci si possono porre le seguenti domande: il servizio ALTO può realmente scoprire le informazioni?

La distribuzione di quelle informazioni ha permesso agli operatori di utilizzare quel servizio?

Quali informazioni sono utili per il client?

Può un client ottenere le informazioni senza avere timore per la privacy?

Ci sono altri modi per trovare le informazioni?

Una volta soddisfatti tali criteri può essere valutata l'importanza dei dati per definire la standardizzazione delle priorità.

Non abbiamo inoltre tenuto conto delle norme su come viene segnalata la congestione o gli interventi di bonifica né si provvede a inoltrare informazioni che rappresentano un'istantanea della rete.

Si potrebbe inoltre valutare l'utilizzo di un nuovo meccanismo di rilevamento dei servizio o modificare altri protocolli per l'effettiva implementazione degli ALTO servizi.

Bibliografia

[1] ALTO Working group <http://datatracker.ietf.org/wg/alto/charter/>

[2] K. Pentikousis

DECADE Server Discovery

DECADE Working Group, 2012

[3] P. Zhang

Comparison of DECADE with CDNi

DECADE, 2012

[4] Michael Piatek, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, Thomas Anderson

Pitfalls for ISP-friendly P2P design

United States, 2010