Securing Rsvp Protocol

Diego Cilea

Tecnologie di Convergenza su IP - AA 2005-06

Introduzione

Internet:

- Servizio Best-Effort
- Indefiniti ritardi
- Frequenti congestioni di rete

Esigenza:
Maggiore supporto per applicazioni real-Time(e.g. voice,video etc.)

Soluzione:
Integrated Services, estensione dell'architettura di Internet.
La rete che supporta questo nuovo tipo di architettura è detta ISPN(Integrated Service Packet Network)

RSVP(Resource reservation protocol)

RSVP è un protocollo progettato per uno scopo:
Riservare risorse per implementare "*Integrated Services*".

RSVP fornisce una configurazione di rete:

- Scalabile
- Robusta
- Flessibile
- Eterogenea

adatta alla gestione delle prenotazioni di risorse con flussi di dati *multicast* e *unicast* per applicazioni real-time.

Si colloca nello **stack Internet**, al livello di un protocollo di trasporto,
È eseguito in background rispetto al traffico in transito.

RSVP(Resource reservation protocol)

Non è un protocollo di routing, ma utilizza i dati di routing preparati da altri protocolli.

RSVP Architecture

- **MultiPoint-to-MultiPoint communication Model(fig.1)**
  Il modello di comunicazione RSVP è una distribuzione di dati tra m sorgenti ed n destinatari, quest'ultimi, con la stessa destinazione.

- **Receiver-Initiated Reservation(fig.2)**
  Il *ricevente* richiedente il servizio sceglie di affiliarsi inviando una richiesta di prenotazione unicast o multicast al *mittente*.

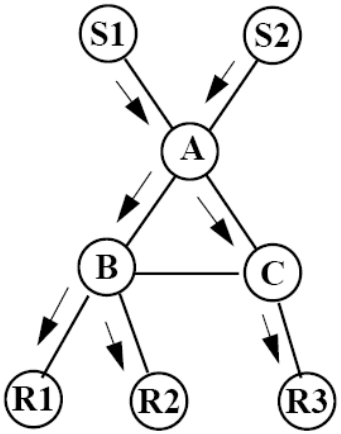


Fig.1

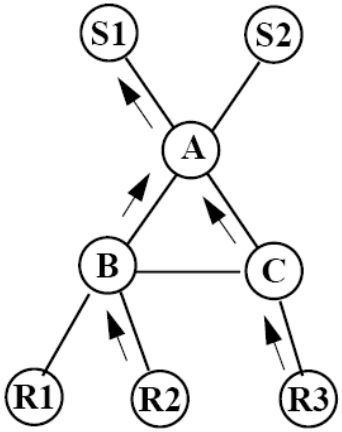


Fig.2

RSVP Architecture

- **Soft State**

  Il protocollo crea in ogni dispositivo che è attraversato dai suoi pacchetti un *soft-state.*

Soft-State riferisce a uno stato nei routers o negli end-node che deve essere periodicamente aggiornato da messaggi RSVP.
Serve per :

- Supporto ai cambiamenti dinamici delle route e alle affiliazioni
- Disaccoppiamento tra nodi interni e sorgenti del flusso
- Garantisce il mantenimento dello stato delle prenotazioni

Quindi il supporto al protocollo garantisce il mantenimento delle Informazioni sullo stato delle prenotazioni lungo la rete.

RSVP Architecture

- **Separazione della Prenotazione dal Routing**
  Questa separazione permette di rendere indipendente dall'architettura di rete (routing) l'utilizzo di RSVP.

Il minimo necessario per il routing del protocollo è la risposta alle **queries** di RSVP per :

- Next Hop discovery per un dato path di destinazione
- Pair retrieving(DESTAddr,SENDAddr)

Problema: caso in cui cambi una delle route di un determinato path:
- Ristabilire le prenotazioni su tutti i router attraversati
- La nuova route sarà adeguata a supportare il traffico per il servizio richiesto?(non tutti i router hanno il supporto hw per RSVP)

Soluzione: Tunneling con tutte le complicazioni del caso(packaging in datagram IP,interface recovery etc.. E soprattutto overhead per capire chi è l'ultimo hop RSVP che ha inviato il messaggio).

RSVP Architecture

In generale il disaccoppiamento avviene in tre passi.:

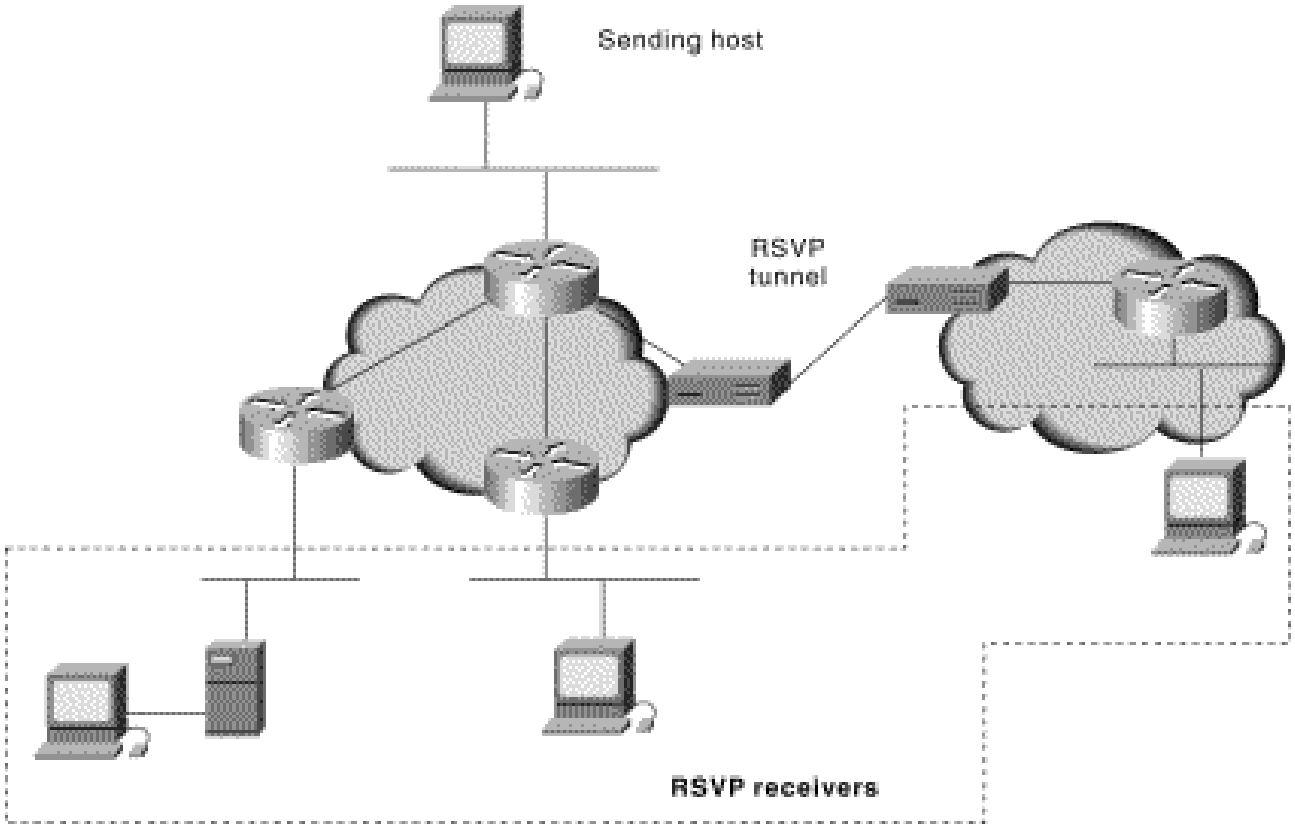
1. **Il protocollo di routing** è responsabile dell'identificazione di una route che abbia la possibilità di soddisfare i requisiti del servizio richiesto.

2. **Il protocollo di prenotazione** verifica la disponibilità delle risorse lungo la route selezionata e prova a riservarle per l'utente.

3. Se questo tentativo **fallisce** il protocollo di routing tenta un'altra route.

Una volta che l'operazione di prenotazione è riuscita il protocollo di routing manterrà la route per l'utente(receiver).

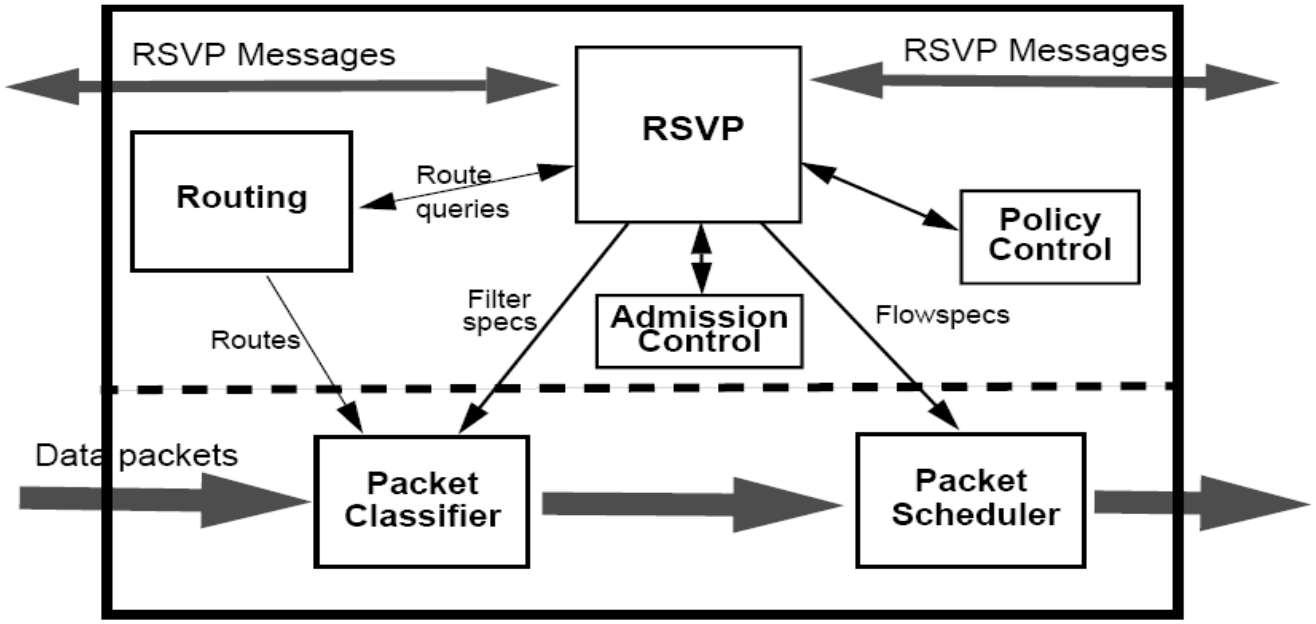
Protocol Overview



Protocol Overview

- Un host usa RSVP per richiedere uno specifico QoS dalla rete sfruttando un application data stream.
- RSVP trasporta la richiesta attraverso le rete visitando ogni nodo che la rete usa per dirigere lo stream.
- Per ogni nodo RSVP prova a riservare risorse per lo stream.

Nel generico router che supporta RSVP avremo tali componenti:



Protocol Overview

Procedura di computazione dei **pacchetti RSVP**:

Per fare una prenotazione di risorse locale al router,il demone RSVP comunica con due moduli decisionali:

- **Admission control**-----> controlla disponibilità risorse per il flusso
- **policy control**----->check se l'utente ha abbastanza diritti per l'op.

OnSuccess: Rsvp daemon scrive i parametri dentro il *packet classifier* e dentro il *packet scheduler* per ottenere il desiderato QoS(descritto da un object ***flowspec***).
Il packet classifier determina la classe QoS per ogni pacchetto e lo Scheduler ordina la trasmissione.

OnFail: Rsvp daemon ritorna un messaggio di errore all'applicazione che ne ha fatto richiesta(e.g. un altro Rsvp daemon).

RSVP Messages

RSVP supporta **quattro** tipi base di messaggi:
- **Reservation-Request** messages(RESV)
- **Path** messages(PATH)
- **Error/Confirmation** messages
- **TearDown** messages

Reservation message(RESV):
- Composto da un oggetto ***flowspec*** e ***filter spec***

- È inviato da ogni receiver in upstream verso i senders per creare un *Reservation State* in ogni nodo.

- Il messaggio segue a ritroso la route che è usata dai data-packets.

- E' necessario inviare questo tipo di messaggio al sender host per permettergli di stabilire e mettere a punto i parametri necessari al controllo del traffico.

- Viaggia su datagram IP

RSVP Messages

Path message(PATH):

- È inviato da ogni **sender** lungo ogni unicast/multicast routes fornite dal protocollo di routing.
- Viaggia **downstream**
- Crea in ogni nodo attraversato un ***path state*** per la gestione del flusso RSVP.
- In ogni nodo RSVP fa **queries** al protocollo di routing per forwardare i pacchetti lungo il percorso adeguato.

Informazioni contenute:

Previous RSVP Hop ----> (Ip | interfaceNumber)

Salvato nel path state dell'Hop attuale per stabilire una tabella di routing dei pacchetti RSVP

TSpec----> descrive il traffico inviato dal sender

È utilizzato per prevenire un esubero di prenotazioni nei links per lo stesso sender

AdSpec----> Misura le proprietà del path

Valore aggiornato in ogni Hop del path,serve al receiver per le prenotazioni

RSpec----> Definisce il traffico QoS desiderato(dentro il *flowspec*)

RSVP Messages

Error and confirmation messages:

- **Path-error-messages**
- **Reservation-request-error-messages**

Passati Hop by Hop per segnalare errori di computazione,failure o risorse insufficienti

- **Reservation-request acknowledges**

Passati hop by hop per comunicare la conferma delle prenotazioni ai singoli router fino al receiver.

RSVP Messages

TearDown Message

- **Rimuovono** gli stati di path e prenotazione lungo la route prenotata
- Avvengono allo scadere del **timeout** di refresh della prenotazione
- Possono essere invocati in caso di **fallimento** o di attacco al protocollo

Si comportano come normali messaggi PATH o RESV.

Securing QoS - Analysis

L'emergere di applicazioni moderne come VoIP, teleconferenze etcc. porta ad una nuova concezione di attacco al core di tali tecnologie, cioè Il loro protocollo.

Tali nuovi network services diverranno quindi prossimamente il nuovo target di attacchi di DoS ed essendo suscettibili alla velocità del traffico,

si passerà da attacchi di **Denial Of Service**(DoS) a
Denial of Quality of Network Services(DoQoNS)

Securing QoS - Analysis

Attackers

In sicurezza delle reti sono classificati come :

- Insider the network
- Outsider the network

Gli outsider sono più facili da controllare attraverso dei meccansimi d'accesso.
Gli insider no in quanto possiedono dei privilegi d'accesso.

In RSVP definiamo tre classi d'attackers:

INSIDER RSVP , cioè un router abilitato RSVP sul path prenotato
OUTSIDER ONPATH, router non RSVP sul path prenotato
OUTSIDER OTHER, router esterno al path

L'ordine di pericolosità è lo stesso con cui sono elencati.

Securing QoS – Attacking Targets

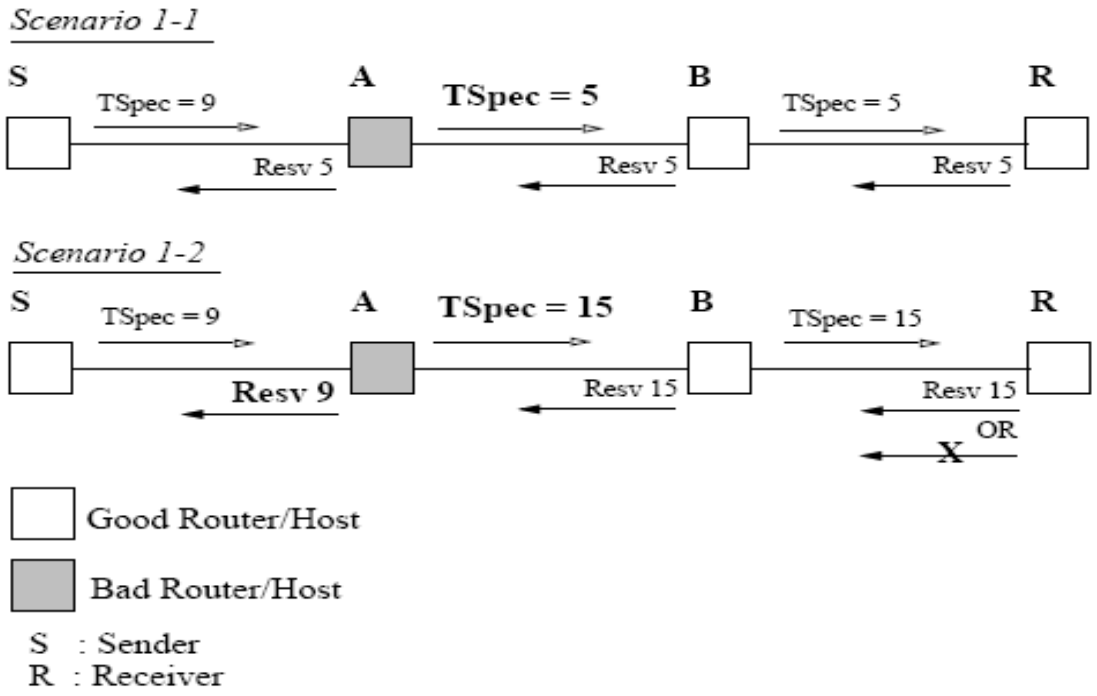
L'obiettivo dell'attacco al protocollo RSVP sono 7 oggetti o messaggi del protocollo:

- TSpec(PATH)
- AdSpec(PATH)
- Rspec(RESV)
- TSpec(RESV)
- Filter_spec(RESV)
- TearDown(PATH)
- TearDown(RESV)

Gli attacchi consistono nell'aumentare o diminuire il valore di questi campi o crearli.

Securing QoS – Attacking Scenarios

1. Tspec(PATH) Tampering Attack



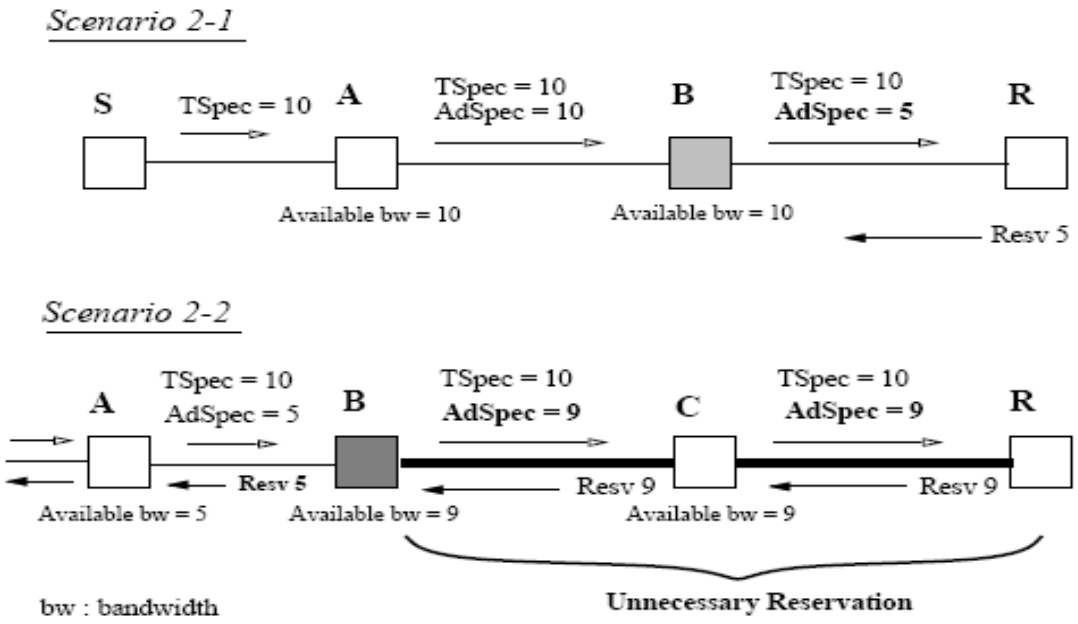
Tspec è utilizzato per prevenire un esubero di prenotazioni di risorse.

In questo caso il receiver, a seconda della manomissione, richiede più o meno risorse del necessario.

Attacco non eseguibile dal di fuori del path route(OUTSIDEROTHER)

Securing QoS – Attacking Scenarios

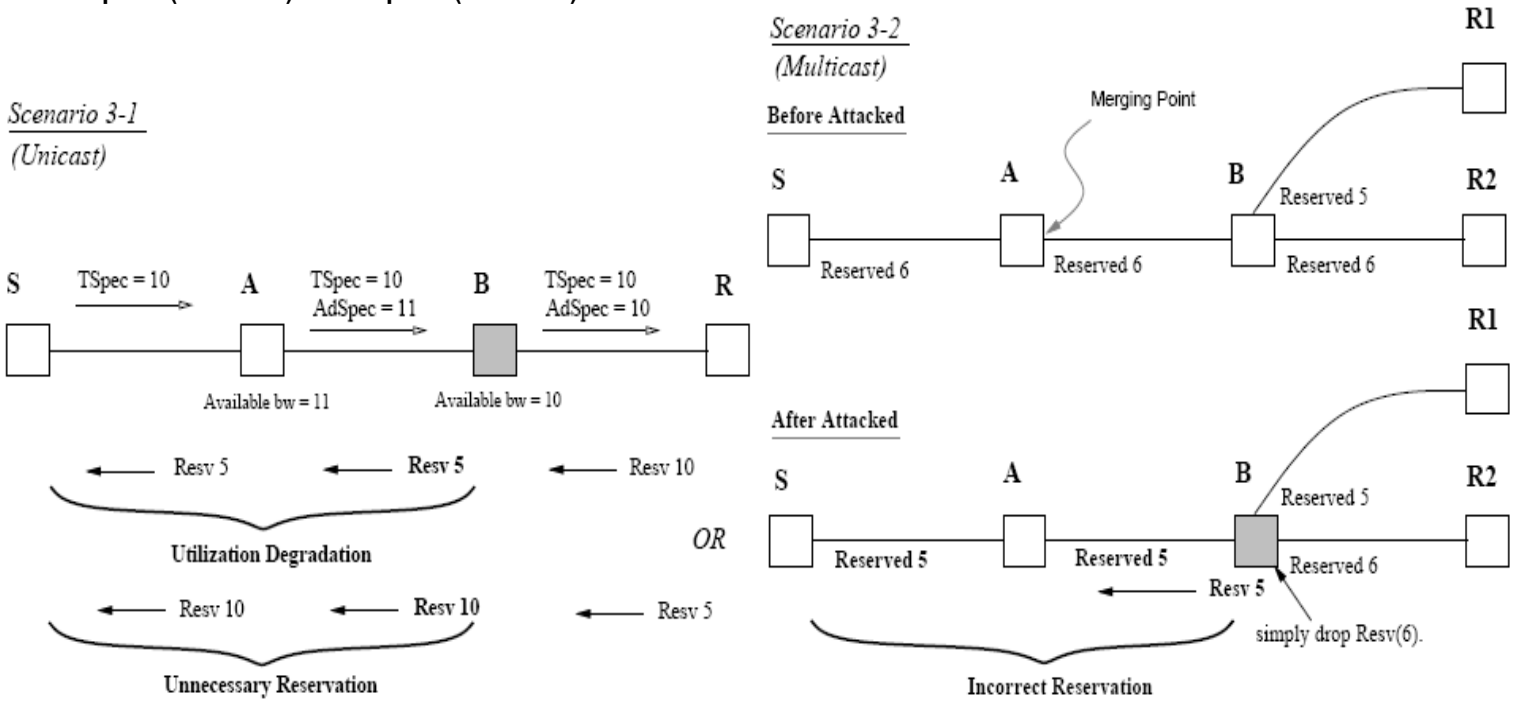
2. AdSpec(PATH) Modification



Concettualmente l'oggetto AdSpec rappresenta il minimo di risorse disponibili lungo il path RSVP.
Una modifica di tale valore porterebbe alla eccessiva o insufficiente richiesta e stanzaizoione di banda da parte del receiver.

Securing QoS – Attacking Scenarios

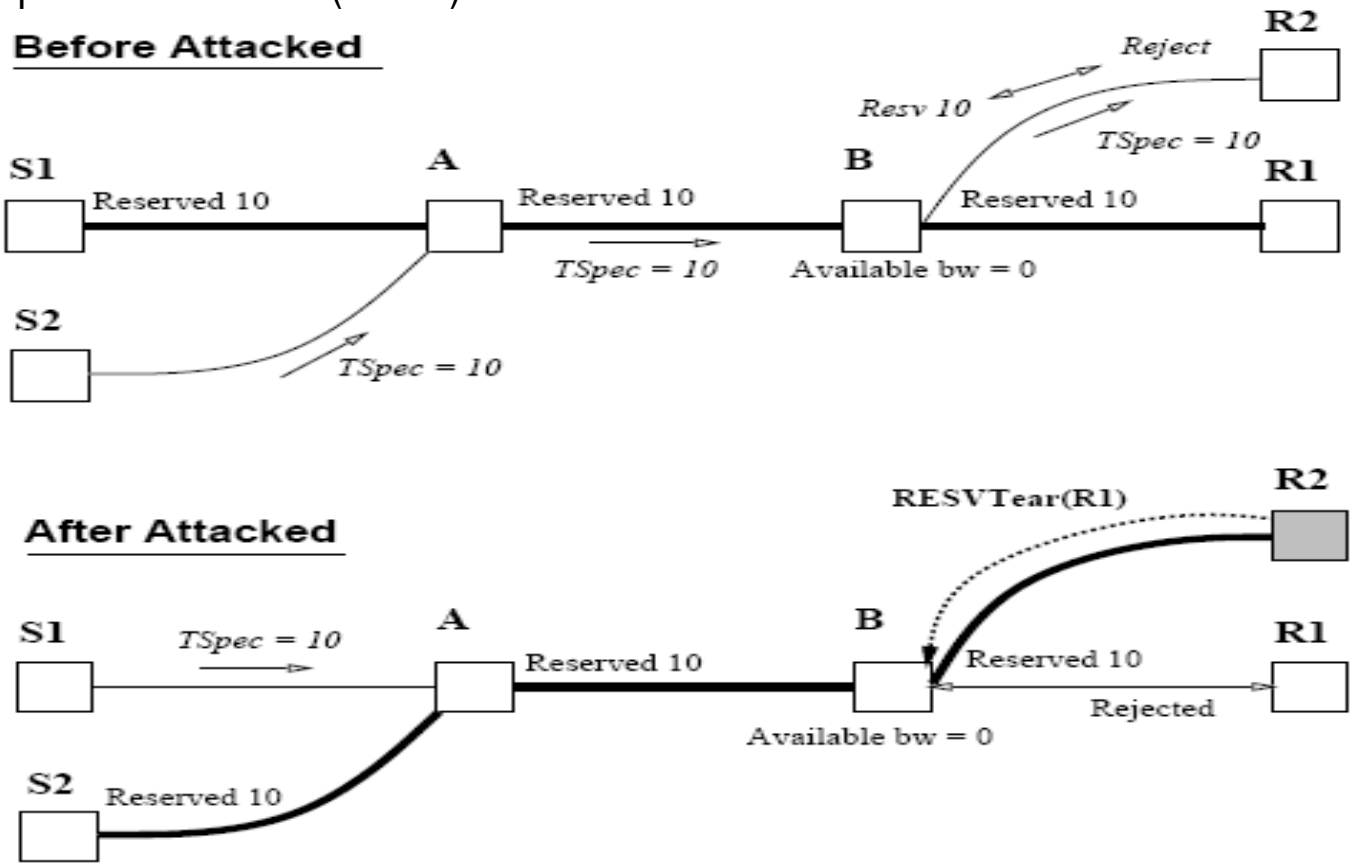
3.Rspec(RESV) e Tspec(RESV) modification



Il router attaccante B(Outsider o Insider) aumenta o diminuisce i valori degli oggetti Rspec o Tspec in modo tale che i router vicini facciano delle prenotazioni di risorse errate e causino degradazione del QoS.

Securing QoS – Attacking Scenarios

4. Spoofed TearDown(RESV)



I messaggi TearDown(RESV) sono creati dai receiver o da qualsiasi Nodo dove una prenotazione è scaduta e viene inviato in upstream Verso gli altri nodi.

Securing QoS – Attacking Scenarios

Questo tipo di attacco è realizzato quando più utenti si competono le Poche risorse disponibili su un router.

- R1 e R2 sono affiliati allo stesso gruppo multicast.
- R2 tenta di riservarsi delle risorse che però sono occupate da R1.
- R2 effettua un attacco “spoofed TearDown” buttando giù le risorse di R1 e una volta caduto si accaparra il maltolto.
- R1 una volta rifatta richiesta non potrà più usufruirne finchè R2 non chiude la sua prenotazione.

Securing QoS – Difesa

Il *soft state* non garantisce sicurezza in special modo da INSIDER RSVP.

Alcuni oggetti sono costanti end to end(TSpec) altri invece vengono cambiati in ogni hop della route legittimamente(Adspec,RSpec)

Il passaggio di pacchetti con informazioni critiche di QoS modificabili è l'oggetto delle soluzioni da trovare.

Securing QoS – Soluzioni

Algoritmo SDS/CD(Selective Digital Signature and Conflict detection)

(Wu,Gong et al.)

- Concettualmente semplice
- Distinzione tra oggetti **costanti** end2end e **modificabili**.
- Gli oggetti costanti sono segnati con una chiave del sender/receiver
- Gli oggetti modificabili sono utilizzati da protocollo

L'utilizzo della crittografia non compromette la performance in quanto
Gli oggetti passano sulla route e vengono controllati solo
da sender/receiver.

Come si garantisce la sicurezza per gli oggetti modificabili?

- Generazione di **pacchetti "storici"** che informano della ricezione i router
- I router mantengono una serie di indicatori di ciò che hanno mandato e lo confrontano con gli "history packets".(e.g.i valori del campo AdSpec)
- Se è riscontrato un **mismatch** viene generato un avviso(e.g. TearDown)

Securing QoS – Soluzioni

Hop by Hop Authentication(Internet Draft IETF,1999)

- Autenticazione e firma dei messaggi con chiave privata.
- Il messaggio crittografato è inviato sulla rete insieme ad un oggetto

INTEGRITY

L'oggetto INTEGRITY è taggato con un numero di sequenza "one-time-use" crittografato anch'esso e con una Key ID per identificare la chiave e l'algoritmo hash da usare.

Il sender/receiver accetterà solo pacchetti con numero di sequenza più alto dell'ultimo ricevuto(previene replay-attacks)

Controllo di integrità dei pacchetti fatto su ogni Hop.

Semplicemente TROPPO costoso in termini di tempo e di Hw.

Securing QoS – Soluzioni

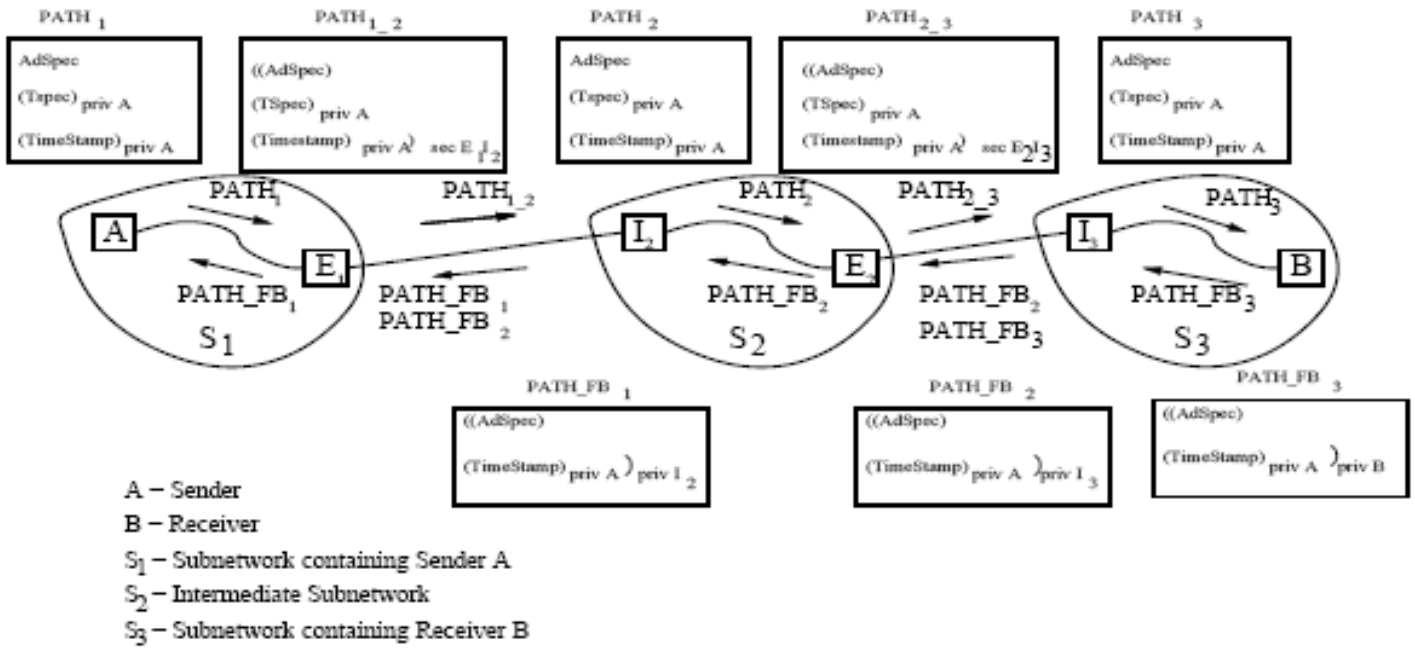
RSVP con protezione QoS scalabile(SQOS)(Talwar,Nath et al.)

- Suddivisione della rete in tante sottoreti(S1…Sn)
- Utilizzo dei Router di frontiera per le comunicazioni *unsecure:*
  Egress node , router di uscita dalla sottorete Si
  Ingress node, router di entrata dalla sottorete Si

Tre fasi della trasmissione per rendere sicuro RSVP:

Fase 1 : Per i messaggi PATH
Fase 3 : Per i messaggi RESV
Fase 2 : Per i messaggi PATH/RESV tra router di frontiera di due sottoreti

Securing QoS – Soluzioni



Fase 1(Intra-Domain PATH Protocol)

Sender segna con la sua chiave le parti costanti del messaggio RSVP
Invia verso Ek il msg
I nodi interni si passano il messaggio fino ad arrivare ad Ek.
Ek manda un messaggio di FeedBack(PATH_FB) cifrando con la sua chiave l'Adspec che sarà controllato dai nodi interni con il loro ultimo AdSpec generato fino al Receiver che controllerà tutto il pacchetto.

Securing QoS – Soluzioni

Fase 2 **Inter Domain PATH/RESV protocol**:

- Egress e Ingress Node delle sottoreti vicine concordano una chiave condivisa a coppie.
- Nello scambio di messaggi viene attaccata al msg la cifatura del PATH msg(eventualmente viene cifrato la parte del msg modificabile).
- Alla ricezione del messaggio viene controllata l'integrità e in caso di successo viene forwardato.

E' l'unica fase dove avviene la crittazione e decrittazione di dati sulla Route.

Fase 3 (**Intra Domain RESV protocol**)

Identico alla fase 1 differisce solo il tipo di messaggio(RESV) e il trattamento del dato che porta, cioè il numero e il tipo di risorse QoS richieste.

N.B.

Nei messaggi, per prevenire l'attacco "Replay" è inserito un timestamp crittografato dal sender/receiver.

Il messaggio di feedBack è locale alle sottoreti,cioè è generato dagli E_k/I_k

Conclusione

L'algoritmo migliore e sicuro è RSVP SQOS per :
- Scalabilità
- Uso minimale di crittografia

In pratica migliora l'algoritmo SDS dal punto di vista di scalabilità e di numero di pacchetti generati.

Esistono molte implementazioni di quest'algoritmo su device di rete
Ad esempio Juniper e Cisco forniscono nativamente il supporto
per tale tipo di algoritmo di sicurezza di Rsvp.

In generale comunque si può dire che al momento una sicurezza dal punto di vista di QoS per RSVP è molto difficile da applicare con reti Best-Effort,e quelle applicabili comportano un overhead di calcolo voluminoso.

References:
RSVP-SQOS, Talwar - http://citeseer.ist.psu.edu/wu99securing.html
Cisco Sys. RSVP - http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/rsvp.htm
IETF – rfc4495,rfc2205,draft md5-08 1999.