Università di Genova *Telematica 2*
Facoltà di Ingegneria

1. Servizi Multimediali e Qualità del Servizio (QdS) su IP

1.3. RSVP

Prof. Raffaele Bolla

dist

RSVP

- E' definito nell'RFC 2205 (e parzialmente aggiornato dal RFC 2750)
- RSVP è un protocollo di segnalazione per il controllo di rete:
  - permette di trasferire le richieste di prenotazione di risorse per traffico a QdS garantita ai *Router*.
- È pensato per gestire individualmente ognuno dei flussi di dati o "*Data Flows*" con esigenze di QdS.
- E' pensato principalmente come meccanismo di segnalazione in ambiente IntServ, ma può essere usato anche in contesti diversi (vedi MPLS)

RSVP

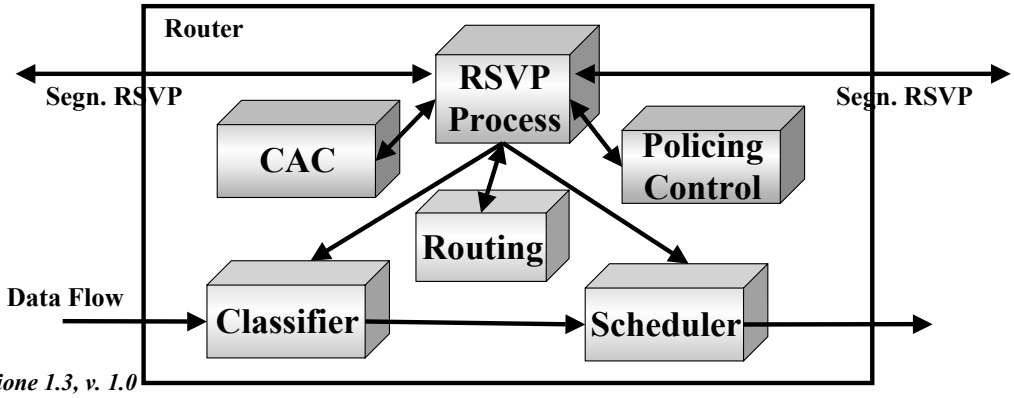
- Ha le seguenti caratteristiche principali:
  - E' stato sviluppato per operare efficientemente anche in presenza di multicast.
  - E' di tipo simplex (opera la segnalazione solo per una direzione).
  - La prenotazione è guidata dal/dai ricevitori;
  - Utilizza ***Soft State***.
  - Permette di "aggregare" secondo modalità diverse le richieste di banda.
  - Opera direttamente sopra IPv4 e IPv6.
  - Non è un protocollo di routing
  - Non definisce politiche di controllo per la QoS (non definisce CAC o Scheduling) pur sotto-intendendone la presenza.

Session

- Lo scambio di informazioni elementare in RSVP si chiama ***Session*** ed è definito come flusso di dati monodirezionale identificato da:
  - Un indirizzo IP di destinazione (anche multicast)
  - Una porta di destinazione
- La porta di destinazione potrebbe essere un qualunque identificatore di livello di Trasporto o anche applicazione.
- Per il momento si usa solo le porte TCP o UDP.
- Si osservi che l'uso dell'identificatore di porta come elemento di classificazione dei pacchetti implica che:
  - Bisogna evitare la frammentazione IP
  - In IPv6 sarebbe opportuno usare in campo Flow Spec
  - Bisogna trattare il traffico criptato in modo specifico

Funzionamento di base

- L'idea di base consiste nel supporre che ciascun nodo (router) attraversato lungo il percorso dalla sorgente alla destinazione possegga le funzionalità necessarie a garantire una qualità di servizio alle **Session**.



Funzionamento di base

- RSVP opera sostanzialmente in due fasi trasferendo in ogni passo uno specifico tipo di messaggio:
  - Fase ***Downstream*** con trasporto del messaggio di **Path**
  - Fase ***Upstream*** con trasporto del messaggio di **Resv**
- La sorgente che vuole iniziare a trasmettere un flusso (attivare una **session**) invia un messaggio di **Path** che viene instradato dai router sul percorso (**Downstream**) definito in qual momento per quella destinazione dal protocollo di routing.
- Ogni *Router* attraversato dal messaggio di **Path** memorizzi il salto (*hop*) inverso verso la sorgente per quella **Session**.
- Quando il messaggio raggiunge la destinazione questa genera un messaggio di risposta **Resv** che percorre all'indietro (***Upstream***) il percorso identificato dal messaggio di **Path**.

Messaggio di *Path*

- Il messaggio ***Path*** in *Downstream* contiene due oggetti principali:
  - **Sender_Tspec**: ossia le specifiche del traffico come definite dalla sorgente. Questo oggetto non può venir modificato dagli elementi lungo il percorso.
  - **AdSpec**: le caratteristiche del percorso e i parametri richiesti dalle funzioni di QoS (servizi disponibili per la QoS e parametri di configurazione). Viene riempito dalla sorgente e/o da tutti i router incontrati.
- Durante la fase ***downstream*** i nodi attraversati non operano nessuna vera azione se non la memorizzazione del percorso della **Session**.

Messaggio di *Path*

- Il pacchetto contenente il messaggio di **Path** ha nell'intestazione IP l'indirizzo della destinazione finale della **Session**, quindi ogni router RSVP *aware* deve intercettare tutti i pacchetti IP ricevuti che contengono messaggi RSVP.
- Questo modo di procedere permette al messaggio di **Path** di attraversare anche router che non hanno RSVP.

Messaggio di Resv

- La vera e propria azione attiva, la **prenotazione** (***reservation***), viene fatta nella fase di **upstream** per due motivi principali:
  - Permettere una efficiente/corretta gestione delle prenotazioni nel caso multicast
  - Operare l'attivazione solo dopo aver avuto l'approvazione dalla sorgente.
- Una richiesta di prenotazione di risorse RSVP consiste in un messaggio di ***Resv*** contenente almeno in una coppia di descrittori detti **FlowSpec** e **FilterSpec**.

FlowSpec e FilterSpec

- **FlowSpec**: identifica il livello di QdS richiesto ed è utilizzato per configurare i parametri degli *scheduler* nei nodi di rete. Il **FlowSpec**, come riportato nella RFC 2210, è composto da due tipologie di parametri e può essere modificato lungo il percorso:
  - » **Rspec**: il livello di QoS desiderato;
  - » **Tspec**: le specifiche del traffico.
- **FilterSpec**: definisce la composizione del ***Data Flow*** ossia specifica i parametri (l'indirizzo sorgente, il numero di porta TCP/UDP) attraverso i quali identificare i pacchetti della **Session**. Viene utilizzato per configurare opportunamente i classificatori dei *Router*.

Resv

- Quando un *Router* sull'albero di distribuzione riceve messaggi di ***Resv***:
  - Interroga il processo di CAC (ossia verifica la disponibilità di risorse) e il processo di *Policing* (i diritti a richiedere un certo servizio)
  - In caso caso positivo configura il classificatore e lo *scheduler*, altrimenti rifiuta la prenotazione.
  - Provvede se possibile all'aggregazione delle richieste in caso di flussi multicast.
  - Inoltra i messaggi di **Resv** verso la sorgente (conosce l'*hop* grazie al messaggio di **Path** precedente).

Soft State

- E' evidente che se l'algoritmo di instradamento decidesse eventualmente di cambiare i percorsi le prenotazioni di risorse perderebbero senso.
- Cioè quello che accadrebbe è che, durante la fase di funzionamento del servizio, i pacchetti non verrebbero più fatti fluire lungo il percorso su cui sono state fatte le prenotazioni e configurati i router ma su altri privi di prenotazioni, con due effetti:
  - Le sessioni non avrebbero più una QdS assicurata
  - Sprecherei risorse sui percorsi non più usati.
- La soluzione di RSVP a questo problema sono i così detti **Soft State**.

Hard e *Soft State* (1/2)

- La RFC 1633 definisce due differenti approcci, *Hard State* e *Soft State,* per memorizzare le caratteristiche dei flussi a QdS garantita ai cui sono state riservate risorse su un *Router*.
- Nell'approccio *Hard State*, lo stato è creato, modificato o distrutto in modo completamente deterministico tramite la cooperazione tra *Router*.
  - Quando un *host* richiede una nuova sessione, è la "rete" a gestire esplicitamente la creazione e la successiva distruzione dello stato.
  - I protocolli *Hard State* dovranno provvedere a meccanismi di segnalazione affidabili (riscontri e ritrasmissioni).

Hard e *Soft State*. (2/2)

- Nell'approccio *Soft State:*
  - Ogni stato è associato ad *timeout*.
  - Per essere mantenuto deve essere periodicamente rinnovato.
  - Gli stati inutilizzati sono cancellati allo scadere dei *timeout* ad essi associati.
- L'approccio *Soft State* risulta essere particolarmente adatto ad ambienti *connectionless*:
  - Ad esempio se cambia l'instradamento i messaggi di *refresh* provvedono all'allocazione di risorse (creazione di nuovi ***soft state***) sul nuovo percorso, mentre gli stati sui *Router* del vecchio percorso vengono invalidati dai *timeout*.

Soft State in RSVP

- L'RSVP utilizza un approccio di tipo *Soft State*.
- In sostanza la configurazione dei router in corrispondenza di una **Session** non permane per un tempo indefinito, ma viene rimossa dopo un timeout (***cleanup timeout***).
- Per questa ragione il meccanismo di prenotazione viene ripetuto periodicamente (*refresh*)
- Il periodo di ripetizione è più corto di ***cleanup timeout*** per evitare che eventuali perdite estemporanee di messaggi RSVP di ***refresh*** causino la rimozione di uno stato.

Rifiuto di una prenotazione

- Nel caso in cui, durante la fase di prenotazione, uno dei router incontrati rifiuti la prenotazione stessa avviene quanto segue:
  - Le prenotazioni avvenute sugli eventuali *router* in *downstream* rimangono valide
  - Il nodo in questione porta il soft-state relativo a quella **Session** in un ***blockade state***
  - Viene emesso un nuovo messaggio **ResvErr** verso la sorgente che
    - » Attiva un ***blockade state*** in ogni nodo attraversato fino alla sorgente
    - » Avvisa la sorgente del fallimento (parziale) della richiesta.

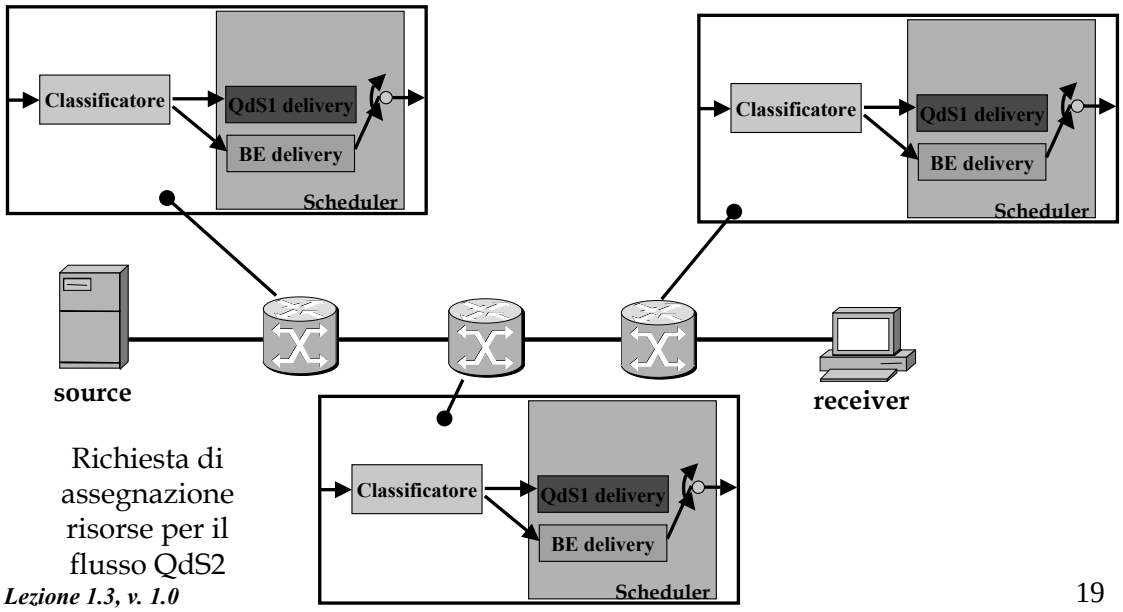
Blockade State

- Nel ***Blockade state*** il nodo in sostanza memorizza il soft state senza attivare la prenotazione
- In corrispondenza dell'attivazione di un ***Blockade state***, viene attivato un timer allo scadere del quale tale stato viene definitivamente eliminato.
- La presenza dello "stato blocco" ha varie ragioni fra le quali, nel caso *unicast*, si ha
  - Permettere ad una sorgente di attivare il servizio anche con una prenotazione parziale
    - » In questo caso in corrispondenza dei messaggi di ***refresh*** periodici i nodi in blocco se si decongestionano onorano la prenotazione
  - Velocizzare il processo di prenotazioni nel caso di futuri tentativi.

Blockade State

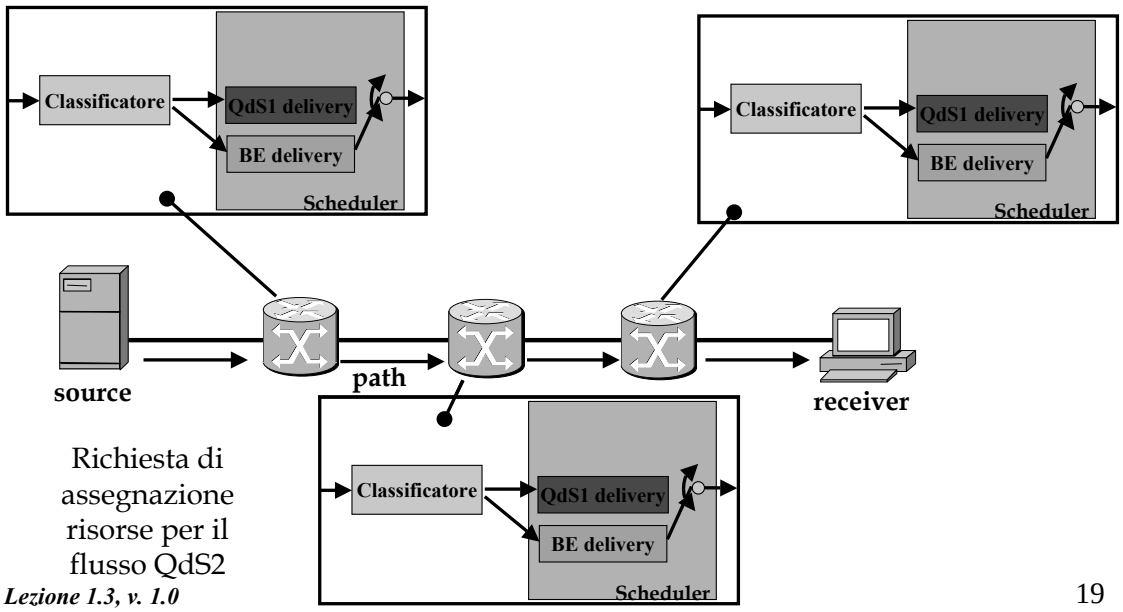
- Nello stato blocco si finisce come stato di transito anche allo scadere del timeout di *refresh*
- Questo per permettere una più rapida riattivazione della prenotazione
  - Per esempio, nel caso il percorso cambi e venga mosso su nodi congestionati, se ritorna rapidamente nella configurazione precedente la prenotazione può venir riattivata rapidamente.

Esempio di Funzionamento nel caso Unicast



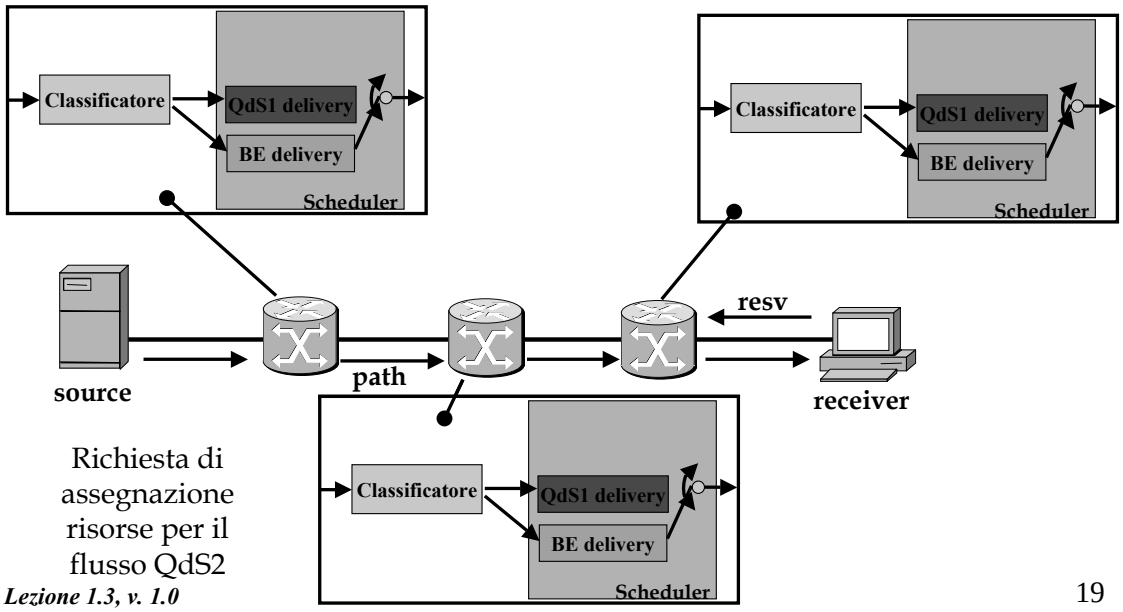
Richiesta di assegnazione risorse per il flusso QdS2

Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

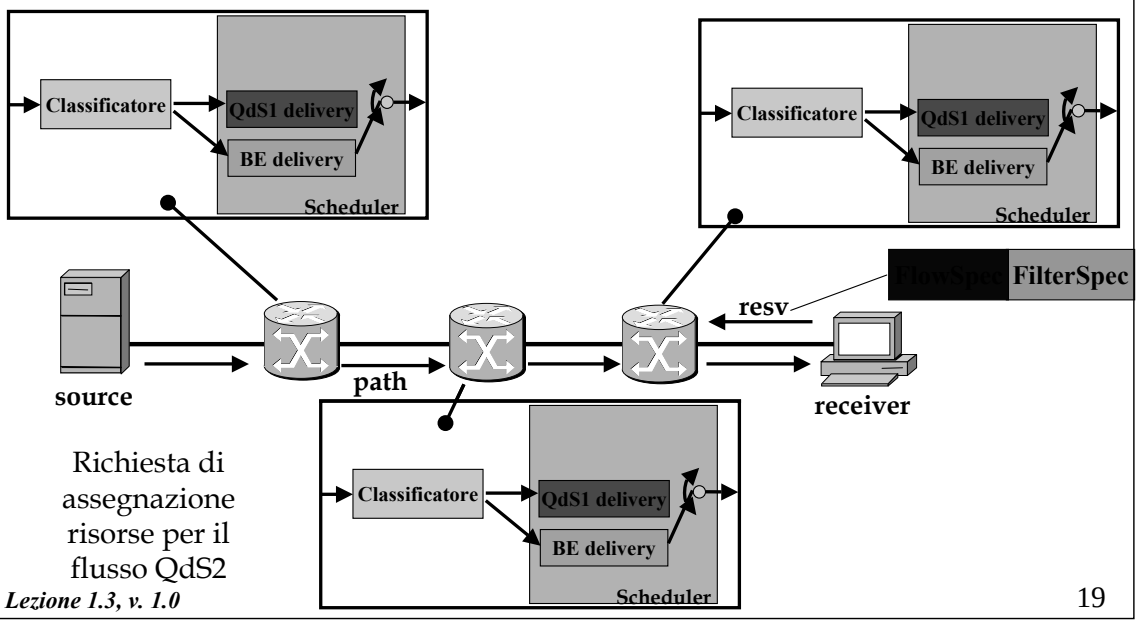
Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

R. Bolla Telematica 2, n. o.

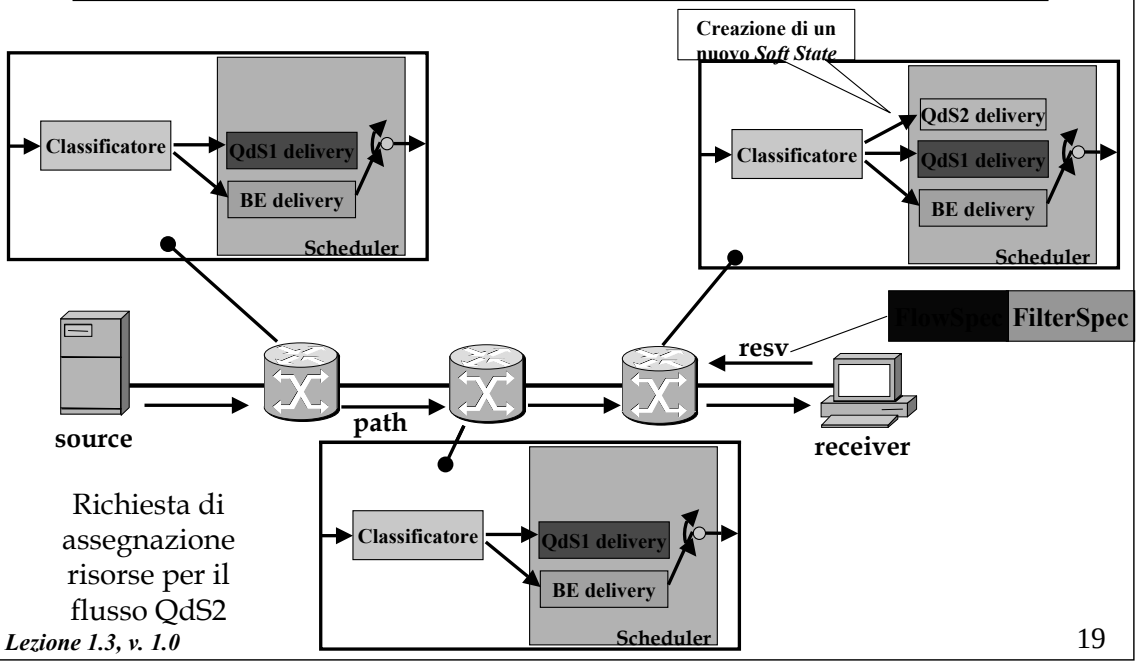
Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

Lezione 1.3, v. 1.0
19

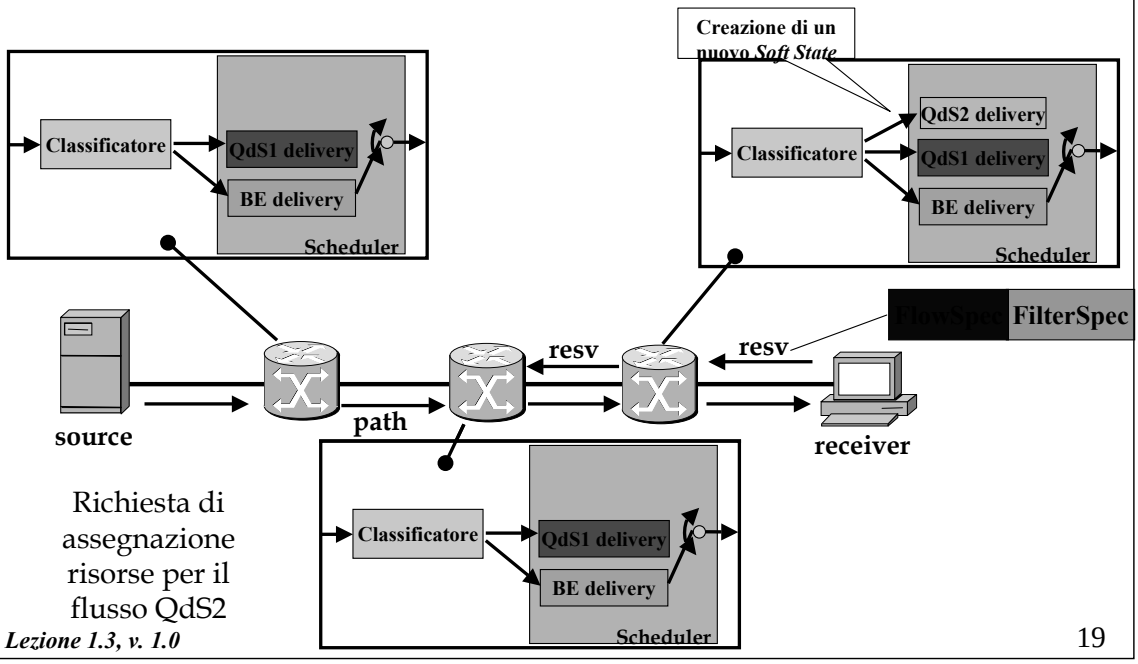
Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

R. Bolla Telematica 2, n. o.

Esempio di Funzionamento nel caso Unicast

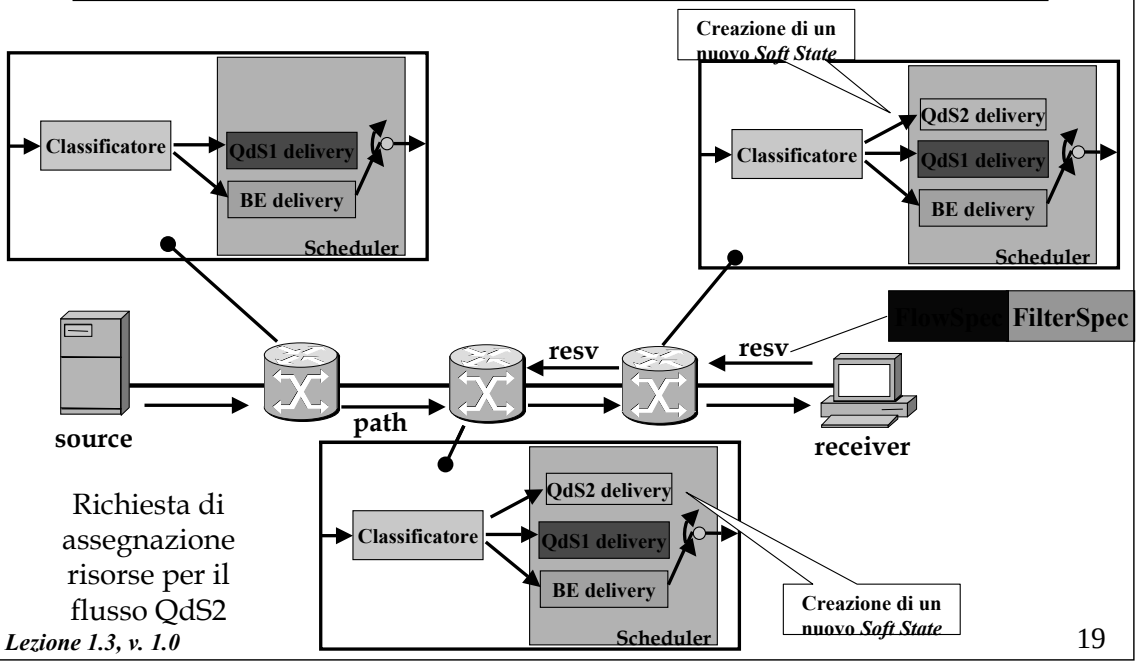


Richiesta di assegnazione risorse per il flusso QdS2

Lezione 1.3, v. 1.0

19

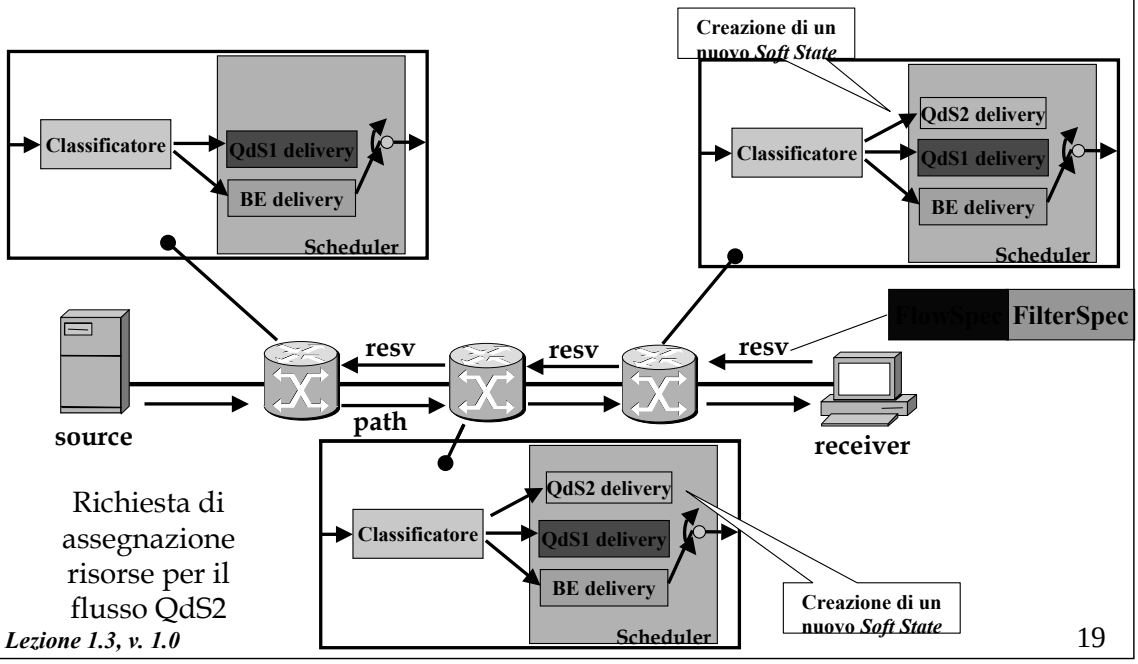
Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

R. Bolla Telematica 2, n. o.

Esempio di Funzionamento nel caso Unicast

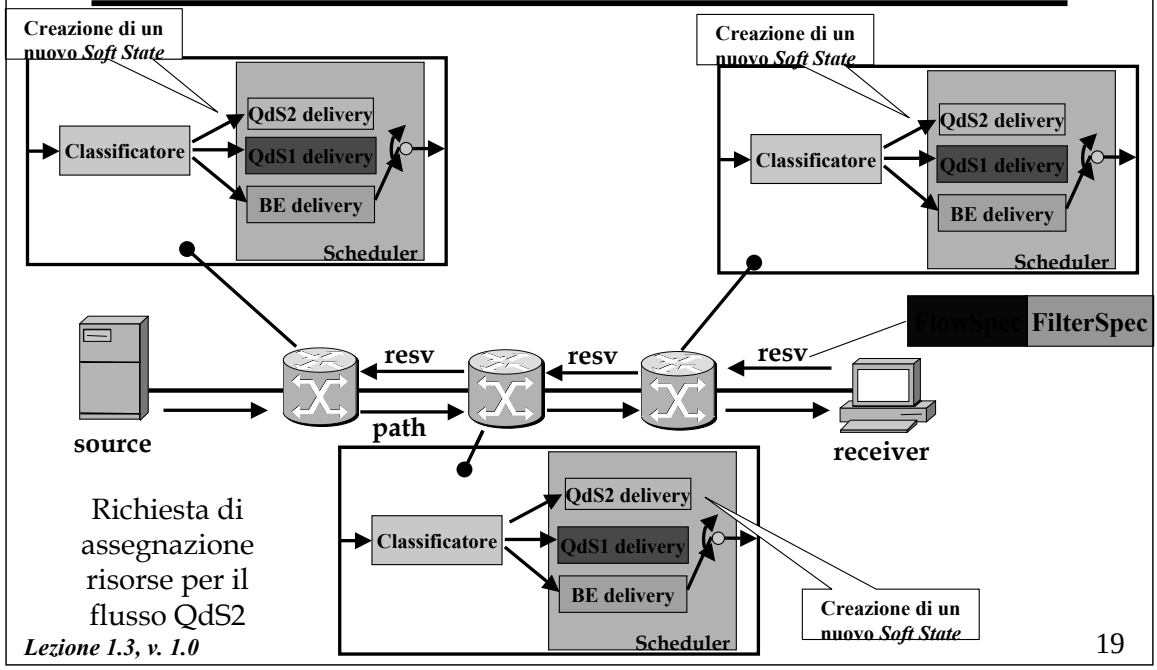


Richiesta di assegnazione risorse per il flusso QdS2

Lezione 1.3, v. 1.0

19

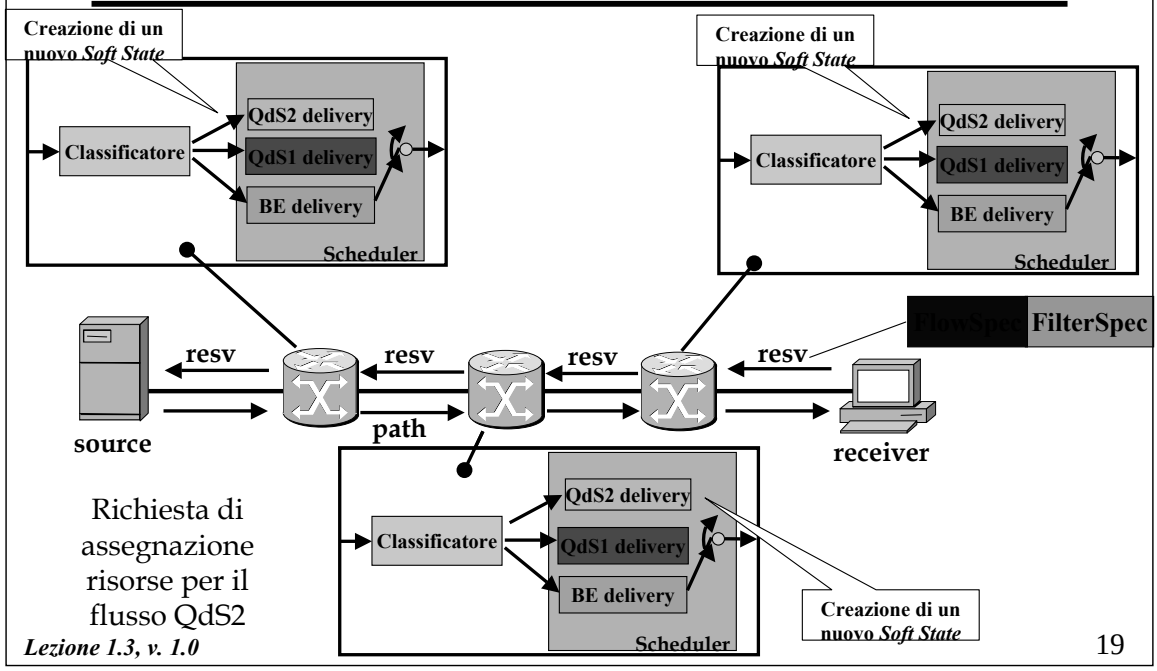
Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

R. Bolla Telematica 2, n. o.

Esempio di Funzionamento nel caso Unicast

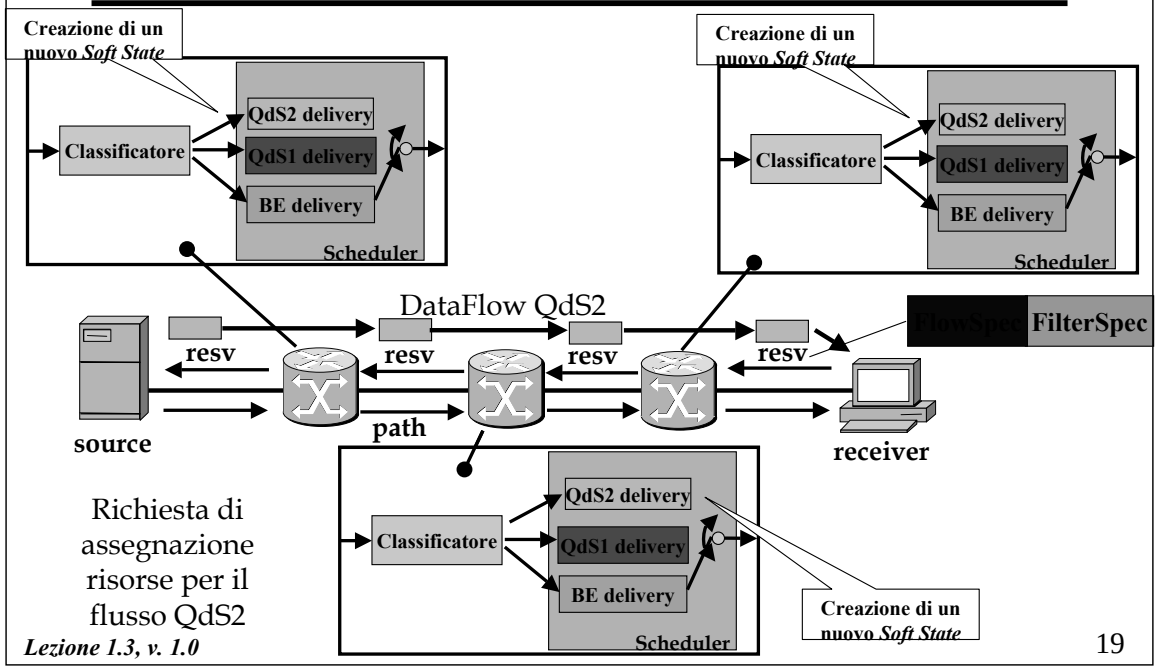


Richiesta di assegnazione risorse per il flusso QdS2

Lezione 1.3, v. 1.0

19

Esempio di Funzionamento nel caso Unicast



Richiesta di assegnazione risorse per il flusso QdS2

Reservation Style

- Se il *Router* è nel punto di incontro di più rami dell'albero di distribuzione del *multicast*, opera effettuando una aggregazione delle richieste.
- L'RSVP consente di utilizzare diverse opzioni (dette "***style***") di aggregazione delle richieste di prenotazione provenienti dagli *host* di un gruppo *multicast* verso uno o più *sender*.

Reservation Style

- Gli stili di prenotazione sono definiti in base a due caratteristiche
- ***Reservation Attribute***: che permette di realizzare la prenotazione di risorse come:
  - ***Distinct***: una prenotazione distinta per ogni sorgente specificata;
  - ***Shared***: una prenotazione complessiva per tutte le sorgenti selezionate.
- ***Sender Selection***: che permette di controllare il modo di selezione delle sorgenti permettendo una selezione di tipo:
  - ***Explicit***: il *receiver* fornisce una lista esplicita dei *sender* selezionati;
  - ***Wildcard***: il *receiver* seleziona implicitamente tutte le sorgenti

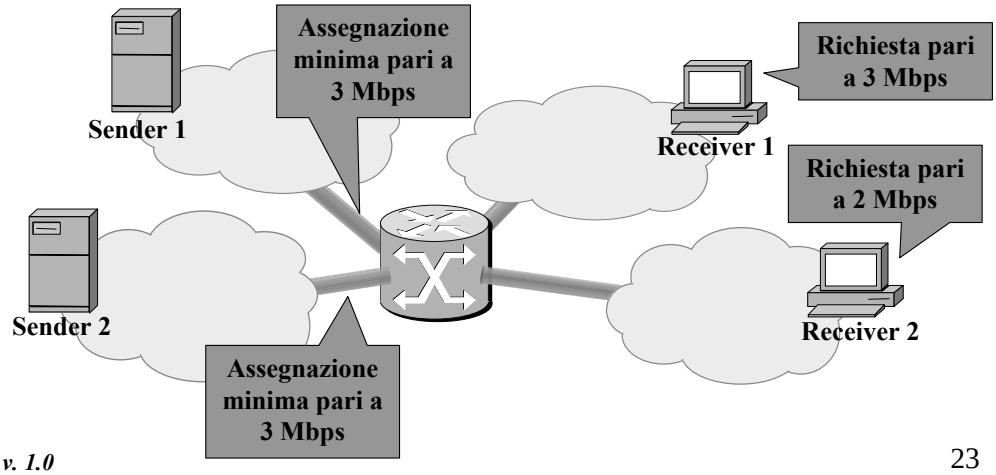
Reservation Style

		Reservation Attribute	
		Distinct	**Shared**
Sender Selection	**Explicit**	**Fixed-Filter Style (FF)**	**Shared-Explicit Style (SE)**
	Wildcard		**Wildcard-Filter Style (SE)**

- L'RSVP definisce tre ***Reservation Style*:**
  - ***Wilcard-Filter Style***: Il WF *Style* è specificato da una richiesta di risorse condivisa da un numero non precisato di sorgenti.
  - ***Fixed-Filter Style***: l'FF *Style* è specificato da una richiesta di risorse distinta per una lista nota di sorgenti.
  - ***Shared-Explicit Style***: l'SE *Style* è specificato da una richiesta di risorse condivisa da una lista nota di sorgenti.

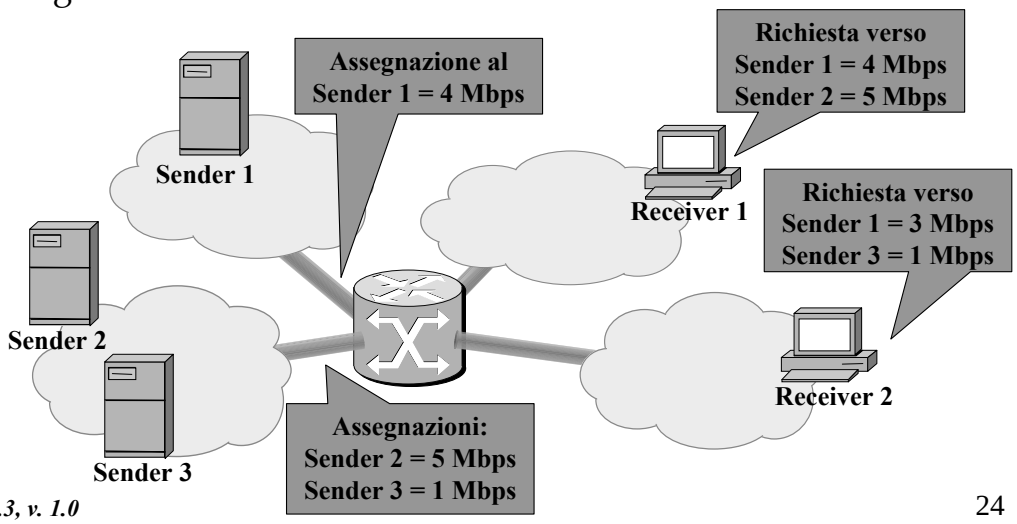
Wildcard-Filter Style

- La risorsa (banda nell'esempio) riservata sulle linee verso i *sender* deve essere maggiore o pari della più alta richiesta registrata dal Router RSVP considerato.



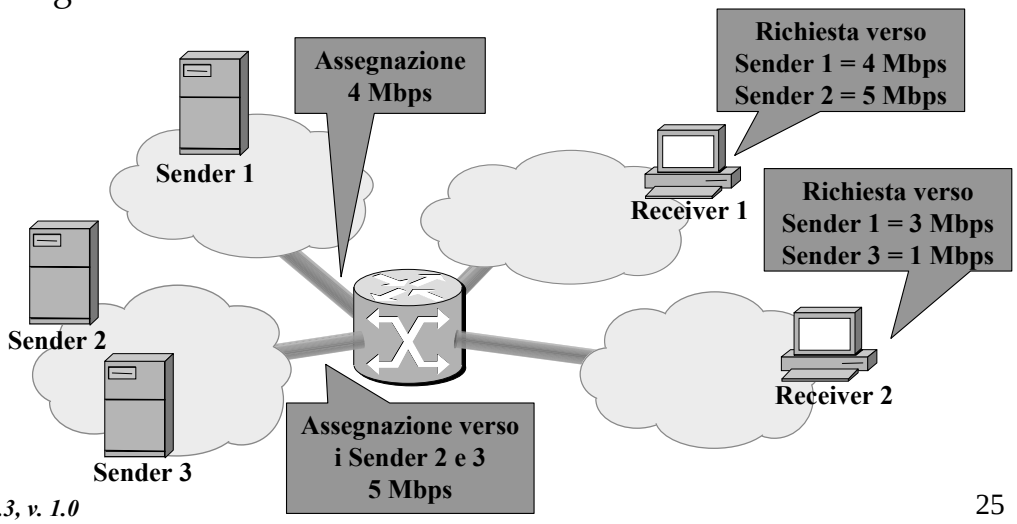
Fixed-Filter Style

- La banda riservata sulle linee verso un *sender* deve essere maggiore o pari della più alta richiesta per quel *sender* registrata dal Router RSVP considerato.



Shared-Explicit Style

- La banda riservata sulle linee verso i *sender* deve essere maggiore o pari della più alta richiesta per ciascun *sender* registrata dal Router RSVP considerato.



Formato del pacchetto: RSVP header

Vers.	Flags	Msg Type	RSVP Checksum
Send_TTL	(Reserved)	RSVP Length	

- ***Vers.*** (4 bit): versione del protocollo;
- ***Flags*** (4 bit): non definiti;
- ***Msg Type*** (8 bit): tipologia di messaggio RSVP, può assumere i seguenti valori:

1 = Path	2 = Resv	3 = PathErr	4 = ResvErr
5 = PathTear	6 = ResvTear	7 = ResvConf	

- ***RSVP Checksum*** (16 bits): controllo di parità calcolato sull'intero messaggio.
- ***Send_TTL*** (4 bit): il valore di TTL nell'header IP all'istante di invio (serve a capire se il pacchetto attraversa *router* non RSVP).
- ***RSVP Length*** (16 bits): lunghezza totale del messaggio in byte.

Message Type

- ***PathTear***:
  - Messaggio utilizzato per cancellare esplicitamente il ***soft-state*** di una *session*.
  - Tali messaggi possono essere creati dai ***sender*** (o dai nodi di transito allo scadere dei ***timeout***) e viaggiano in ***downstream*** fino ai ***receiver***.
    - » Nel caso unicast non viene inoltrato solo dai nodi che hanno un diverso *previous hop* nel ***path-state*** di quella sessione***;***
    - » nel caso multicast, invece, la decisione di ***forwarding*** di questo messaggio è più articolata.
  - La cancellazione di un ***path-state*** può comportare una ri-allocazione delle risorse al nodo.

Message Type

- ***ResvTear:***
  - Fa sostanzialmente la stessa operazione del ***PathTear*** ma nella direzione opposta**.**
  - Il ***soft-state*** candidato alla cancellazione è identificato dai parametri SESSION, FILTER_SPEC e STYLE.
  - Può essere generato dai ***receiver*** o dai nodi allo scadere di un ***reservation timeout***; viene propagato in ***upstream*** fino ai ***sender***.
  - L'inoltro del messaggio di *ResvTear* può essere interrotto solo in corrispondenza dei nodi in cui vengono aggregate le richieste di prenotazione.

Message Type

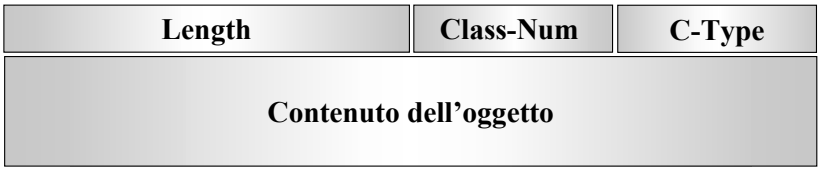
- *PathErr*:
  - Riporta la descrizione di eventuali problemi nell'elaborazione dei messaggi di ***Path***.
  - Vengono inoltrati in ***upstream*** fino ai ***sender***.
  - Non modificano in alcun modo lo stato dei nodi attraversati.
- ***ResvErr:***
  - Riporta la descrizione di eventuali problemi nell'elaborazione dei messaggi di Resv.
  - Vengono inoltrati in ***downstream*** fino ai ***receiver***.
  - Tali messaggi devono essere inoltrati verso tutti ***receveir*** che possono aver provocato l'errore.

Message Type

- ***ResvConf:***
  - Trasporta il riscontro per una richiesta di prenotazione di risorse conclusa positivamente.
  - È inoltrato in ***downstream*** ai ***receiver***.

Formato del pacchetto: RSVP payload

- Il payload del messaggio RSVP è costituito da uno o più oggetti.
- Ogni oggetto è composto da una riga di intestazione e una o più parole (32 bit)



- ***Length*** (16 bit): lunghezza dell'oggetto in byte.
- ***Class-Num*** (8 bit): identifica la classe dell'oggetto.
  - I due bit più significativi di ***Class-Num*** sono utilizzati per determinare quale azione deve compiere il nodo se non riconosce la classe dell'oggetto.
- ***C-Type*** (8 bit): assieme a ***Class-Num*** identifica univocamente il tipo di oggetto.

Classi

- RSVP definisce le seguenti classi di oggetto:
  - ***NULL***: identificato dal campo ***ClassNum*** posto a 0. Il suo contenuto viene ignorato dai ricevitori.
  - ***SESSION***: obbligatorio in ogni messaggio RSVP. Contiene i dati necessari ad identificare una specifica sessione a cui applicare le informazioni degli oggetti successivi. In particolare contiene l'indirizzo IP del destinatario, l'IP ***protocol*** ID ed una descrizione dei valori utilizzati per il numero di porta di destinazione.
  - ***RSVP_HOP***: contiene l'indirizzo IP del router RSVP che ha inviato il messaggio.
  - ***TIME_VALUES***: obbligatorio nei messaggi di ***Path*** e ***Resv***; contiene il valore del periodo di ***refresh*** utilizzato dal ***sender.***

Classi

- ***STYLE***: necessario nei messaggi di ***Resv***; definisce il ***Reservation-Style*** e contiene informazioni aggiuntive non presenti in ***Filter-Spec*** e in ***FlowSpec.***
- ***FLOWSPEC***: nei messaggi ***Resv***; definisce il livello di QdS desiderata.
- ***FILTER-SPEC***: nei messaggi ***Resv***; definisce un sotto-insieme di pacchetti della sessione che devono ricevere la QdS desiderata.
- ***SENDER_TEMPLATE***: obbligatorio nei messaggi di ***Path***; contiene l'indirizzo IP e altri parametri utili per riconoscere il ***sender***.
- ***SENDER_TSPEC***: obbligatorio nei messaggi di ***Path***; contiene i parametri descrittivi del ***data flow*** di un ***sender.***
- ***ADSPEC:*** nei messaggi di ***Path***; contiene informazioni sui servizi disponibili.

Classi

- ***ERROR_SPEC***: specifica l'errore nei messaggi di PathErr e di ResvErr; nei messaggi ResvConf specifica la conferma.
- ***POLICY_DATA***: trasporta informazioni utilizzabili dal modulo di ***policy*** locale per modificare la propria configurazione; può apparire nei messaggi di Resv, Path, ResvErr o PathErr.
- ***INTEGRITY***: trasporta informazioni crittografate per autenticare il nodo sorgente.
- ***SCOPE***: contiene una lista esplicita dei ***sender*** a cui il messaggio deve essere recapitato. Può comparire nei messaggi di Resv, ResvErr e ResvTear.
- ***RESV-CONFIRM:*** contiene l'indirizzo IP del receiver che ha richiesto la conferma. Può comparire nei messaggi di Resv, e ResvConf.

Messaggio di Path

- Si riporta il contenuto come esempio la struttura del messaggio di Path:
- <Path Message> ::=<Common Header>
  [<INTEGRITY>]
  <SESSION>
  <RSVP_HOP>
  <TIME_VALUES>
  [<POLICY_DATA> ...]
  [<sender descriptor>]

  <sender descriptor>::= <SENDER_TEMPLATE>
  <SENDER_TSPEC>
  [<ADSPEC>]