

**1. Servizi Multimediali e  
Qualità del Servizio (QoS) su IP  
1.6 RTP - SDP**

Prof. Raffaele Bolla



## **Trasporto di flussi multimediali**

---

- La rete Internet è stata concepita per il trasporto *best-effort* e quindi non in tempo reale di dati
- Non possiede meccanismi nativi atti a garantire la QoS:
  - » non ci sono limiti ai ritardi e agli jitter;
  - » i pacchetti potrebbero non arrivare a destinazione;
  - » i pacchetti dalla stessa sorgente possono essere instradati in modo diverso dalla rete verso una comune destinazione, arrivando in ordine diverso rispetto alla partenza.

## Trasporto di flussi multimediali

---

- Il protocollo di trasporto TCP realizza un servizio orientato alla connessione
  - garantisce l'arrivo dei pacchetti alla destinazione;
  - garantisce la consegna nell'ordine corretto dei pacchetti;
  - effettua un controllo di flusso;
  - rileva gli errori;
  - effettua un controllo di congestione.
- UDP realizza un servizio minimo
  - individua le entità di trasporto;
  - effettua il controllo di errore.

## Trasporto di flussi multimediali

---

- La trasmissione di flussi multimediali spesso è soggetta a vincoli sul massimo ritardo *end-to-end*
  - i meccanismi di recupero di errore e controllo di congestione e flusso del TCP non permettono di utilizzare questo protocollo;
  - il TCP può tuttavia essere utilizzato per quei flussi con requisiti meno stringenti
    - » testo (chat).

## Trasporto di flussi multimediali

---

- L'UDP è più “snello” e quindi più adatto al trasporto di dati real-time, ma manca di meccanismi essenziali, fra cui
  - individuazione delle perdite,
  - individuazione dei pacchetti fuori sequenza.
- L'UDP si presta bene per offrire il servizio di trasporto ad un ulteriore protocollo in grado di gestire flussi multimediali

### *Real Time Protocol (RTP)*

## *Real-Time Protocol (RTP)*

---

- Introduce le funzionalità che mancano a UDP per lo *streaming* di contenuti multimediali anche in multicast.
- Sviluppato principalmente da H. Schulzrinne nel periodo 1992-1996.
- **RFC 1889** definisce il protocollo mentre **RFC 1890** fornisce il profilo base per audio/video conferenza.
- RTP risiede nei sistemi terminali (*end system*).
- I pacchetti RTP vengono incapsulati in segmenti UDP.
- Per quanto riguarda l'architettura funzionale RTP è considerato una estensione del protocollo di trasporto.
- Dal punto di vista dello sviluppatore è invece parte dell'applicazione, ossia deve essere inglobato nel software applicativo (per lasciare più controllo e flessibilità all'operazione di pacchettizzazione).

## RTP

---

- In sostanza RTP aggiunge all'UDP le seguenti funzionalità:
  - **Identificazione del *payload***
  - **Numeri di sequenza**
  - *timestamp*
  - **Capacità di identificare le sorgenti per la sincronizzazione**
- Ad esempio, nel caso di voce PCM a 64 Kbit/s, ad ogni blocco audio viene aggiunta l'intestazione RTP a formare un pacchetto RTP che viene quindi passato all'UDP.
- Il pacchetto RTP permette alla sorgente di riconoscere il tipo di codifica (che la sorgente può modificare potenzialmente pacchetto per pacchetto) e compensare il  *jitter* tramite il *timestamp* ed i numeri di sequenza.

## RTP: QoS

---

- RTP non fornisce nessun meccanismo per assicurare una QoS (ritardo o perdita).
- Il pacchetto RTP è visto solo dai nodi finali e non dai nodi di rete (router) che quindi non forniscono ai pacchetti RTP un trattamento preferenziale.
- Per poter fornire una QoS bisogna utilizzare dei meccanismi aggiuntivi (*Integrated o Differentiated Services, MPLS, ...*).

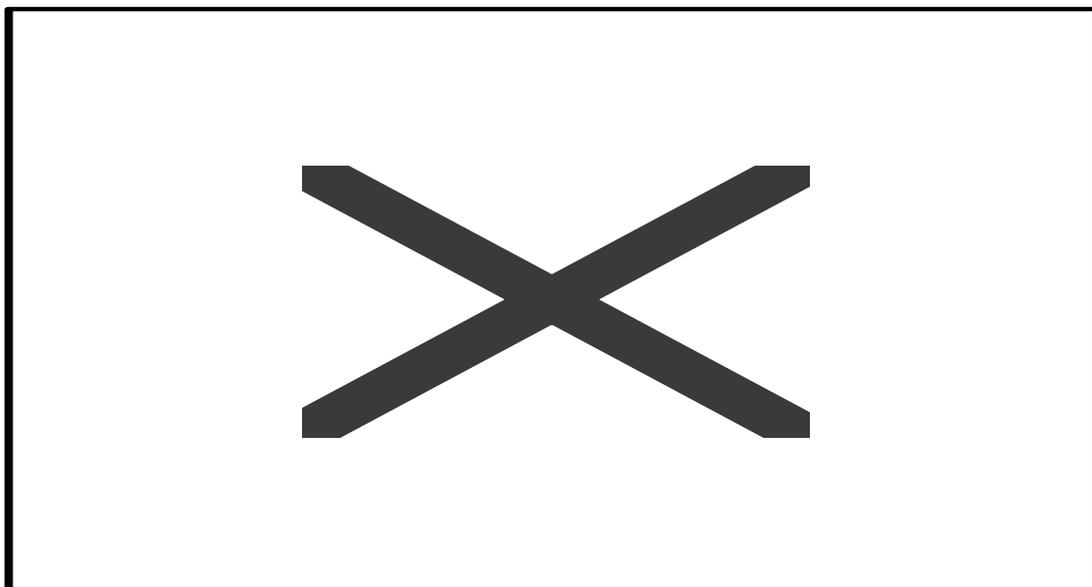
## RTP: *Streams*

---

- RTP permette di assegnare un flusso (*stream*) indipendente ad ogni sorgente (telecamera, microfono, ...).
- Alcuni standard di compressione (MPG1 e 2 ad esempio) permettono l'integrazione dell'audio e video durante il processo di decodifica; in questo caso ci sarà un solo *stream* per ciascuna direzione (comunicazioni unicast).
- Nel caso multicast ogni sorgente ha il proprio *stream* ma tutti gli *stream* usano lo stesso indirizzo multicast e lo stesso albero di distribuzione; la comunicazione tramite RTP all'interno di un gruppo multicast viene chiamata "session".

## Il pacchetto RTP

---



## Il pacchetto RTP

---

- **Version** (2 bit): è la versione, attualmente la 2.
- **Padding (1 bit)**: l'ultimo byte contiene il numero di byte da ignorare alla fine del pacchetto.
- **eXtention** (1 bit): se a 1 indica la presenza di un *header extention* alla fine dell'intestazione.
  - L'estensione è composta da tre campi:
    - » "Defined by Profile", l'interpretazione è specifica dell'implementazione;
    - » "Length", il numero di successive word da 32 bit;
    - » dati relativi all'estensione.
  - L'estensione è usata raramente, principalmente per scopi sperimentali.

## Il pacchetto RTP

---

- **Marker** (1 bit): il profilo definito nell'RFC 1890 specifica l'uso di questo bit per applicazioni voce e video:
  - voce: posto a 1 nel primo pacchetto successivo ad una fase di soppressione del silenzio
    - » durante le fasi di silenzio non sono inviati pacchetti,
    - » permette di regolare il ritardo di riproduzione nel buffer in ricezione;
  - dati: posto a 1 nell'ultimo pacchetto di uno stesso frame video.

## Il pacchetto RTP

---

- **Payload Type** (7 bits): indica il tipo di codifica usata per quel pacchetto (se la sorgente cambia codifica e tramite questo campo che il ricevitore se ne accorge)
  - statici (0-34):
    - » Payload type 0: PCM mu-law, 64 Kbps
    - » Payload type 3, GSM, 13 Kbps
    - » Payload type 4, G.723.1
    - » Payload type 7, LPC, 2.4 Kbps
    - » Payload type 26, Motion JPEG
    - » Payload type 31, H.261
    - » Payload type 33, MPEG2 video
    - » Payload type 34, H.263
  - non assegnati (35-71, 77-95) e riservati (72-76)
  - dinamici (96-127): la tendenza attuale è di usare questi.

## Il pacchetto RTP

---

- **Sequence Number** (16 bits): si incrementa di uno ogni pacchetto inviato (serve a riconoscere le perdite e i pacchetti fuori sequenza).
- **Timestamp** (32 bit): riflette l'istante di campionamento del primo campione presente nel pacchetto ed è derivato dal *clock* del campionatore.
  - Ad esempio, viene incrementato di uno ogni campione (per la voce +1 ogni 125  $\mu$ s e quindi supponendo 8 bit per campione e 160 Bytes a pacchetto, verrebbe incrementato di 160 ogni pacchetto). Il valore viene incrementato anche quanto la sorgente è inattiva ed è riferito al flusso prima della codifica.
  - Per il video, tutti i pacchetti appartenenti allo stesso frame hanno lo stesso timestamp.

## RTP e clock

---

- Tre clock sono utilizzati nella trasmissione
  - il clock del campionatore del segnale analogico;
  - il clock RTP
    - » genera il timestamp,
    - » viene incrementato linearmente e monotonicamente,
    - » presenta una precisione tale da permettere la sincronizzazione e il calcolo degli jitter in ricezione,
    - » per la voce normalmente è uguale al clock del campionatore (8, 44.1 o 48 kHz),
    - » per i dati spesso è pari a 90 kHz (multiplo di 24 Hz di HDTV, 25 Hz di PAL, 29.97 Hz di NTSC);
  - il clock di riferimento, per sincronizzare i flussi provenienti da diverse sorgenti.

## Il pacchetto RTP

---

- I rimanenti campi vengono utilizzati solo nelle sessioni multicast (anche se devono essere sempre presenti nell'intestazione).
- **CSRC Count** (4 bit): indica il numero di campi CSRC presenti nel pacchetto.
- **Synchronization Source Identifier** (32 bit): Identifica la sorgente, non è un indirizzo IP ma un identificatore generato casualmente.
- **Contributing Source** (CSRC) (32 bit): se un pacchetto contiene segnali di più sorgenti (SSRC) mescolati (tipicamente audio di più parlatori) questo campo permette di identificare le diverse sorgenti originarie (fino a 15).

## Il pacchetto RTP

---

- Il *Payload* consiste nel media da trasportare.
- Il formato del payload dipende dal media, dal codec e dal meccanismo di recupero dei pacchetti persi.
- È previsto anche un formato particolare per il *confort noise*, nel caso il codec non preveda un VAD/CNG.
- Per i codec voce tipicamente è sufficiente specificare:
  - la frequenza del clock RTP,
  - il codec utilizzato.
- È previsto un payload particolare per il recupero d'errore.

## Il pacchetto RTP

---

- Esempi:
  - il *payload* G.723.1 (PT=4) è composto da:
    - » frequenza del clock RTP, coincide con il campionatore a 8 kHz,
    - » frame G.723.1.
  - il *payload* per il *confort noise* (PT=13)
    - » frequenza del clock RTP (8 kHz),
    - » un byte, i cui bit meno significativi indicano la potenza del rumore.
  - il *payload* per T.140
    - » frequenza clock RTP (1 kHz),
    - » uno o più caratteri (dipende da compromesso tra tempi di reazione e *overhead*);
    - » il formato T.140 può essere inviato con TCP senza RTP!

## Il pacchetto RTP - Ridondanza

- Diversi *payload* ridondanti sono aggiunti al pacchetto per permettere il recupero di errore.
- L'*header* RTP si riferisce al *payload* primario
  - il PT indica comunque il formato particolare del pacchetto (*Redundancy*).
- Il *payload* del pacchetto è formato da un *header* per ogni pacchetto secondario (4 byte) e uno per il pacchetto primario (1 byte).

## Il pacchetto RTP - Ridondanza

V	P	X	CSRC count	M	PT=Redudancy	Sequene number del payload principale
Timestamp del payload principale						
SSRC						

### *Redundancy Header*

F	Block PT	Timestamp offset	Block length
		⋮	⋮
F	Block PT	Timestamp offset	Block length

### *Redundancy Payload*

F	Block PT	Primo Payload secondario
		Secondo Payload secondario
		Payload primario

## Il pacchetto RTP - Ridondanza

---

- *First* (1 bit): 0 indica l'ultimo *header* (primario).
- *Block Payload Type* (7 bit): il formato del payload.
- *Timestamp Offset* (14 bit): l'offset rispetto al timestamp indicato nell'intestazione RTP.
- *Block Length* (10 bit): la dimensione del payload descritto.

## Il pacchetto RTP - FEC

---

- L'idea è quella di generare un pacchetto FEC di correzione che
  - contenga il risultato di un XOR (o di una operazione analoga) tra i *payload* di pacchetti non FEC precedenti,
  - Venga inviato su una diversa sessione RTP
    - » in questo modo non è necessario de-moltiplicare i pacchetti RTP sulla base del PT;
- La struttura del pacchetto permette l'utilizzo di diverse tecniche di correzione (ad esempio, codici di Hamming, Reed- Solomon). Nel seguito, per semplicità, supporremo che l'operazione effettuata sia solo XOR.

## Il pacchetto RTP - FEC

---

### Header RTP

V	P	X	CSRC count	M	PT=FEC	Sequence number
Timestamp						
SSRC						

### Header FEC

Sequence number base			Length recovery			
E	PT Recovery		Mask			
Timestamp recovery						

### Payload FEC

XOR tra i payload ( + campi variabili dell'intestazione RTP) dei pacchetti non FEC						
--	--	--	--	--	--	--

## Il pacchetto RTP - FEC

---

- I campi *P*, *X*, *CSRC count*, *M* dell'intestazione del pacchetto FEC sono lo XOR dei rispettivi campi nei pacchetti non-FEC.
- Il valore del *Timestamp* si riferisce all'istante di invio del pacchetto FEC.
- *Sequence number base* (16 bit): pari al minimo numero di sequenza dei pacchetti non-FEC.
- *Length recovery* (16 bit): XOR delle lunghezze dei pacchetti non-FEC.

## Il pacchetto RTP - FEC

---

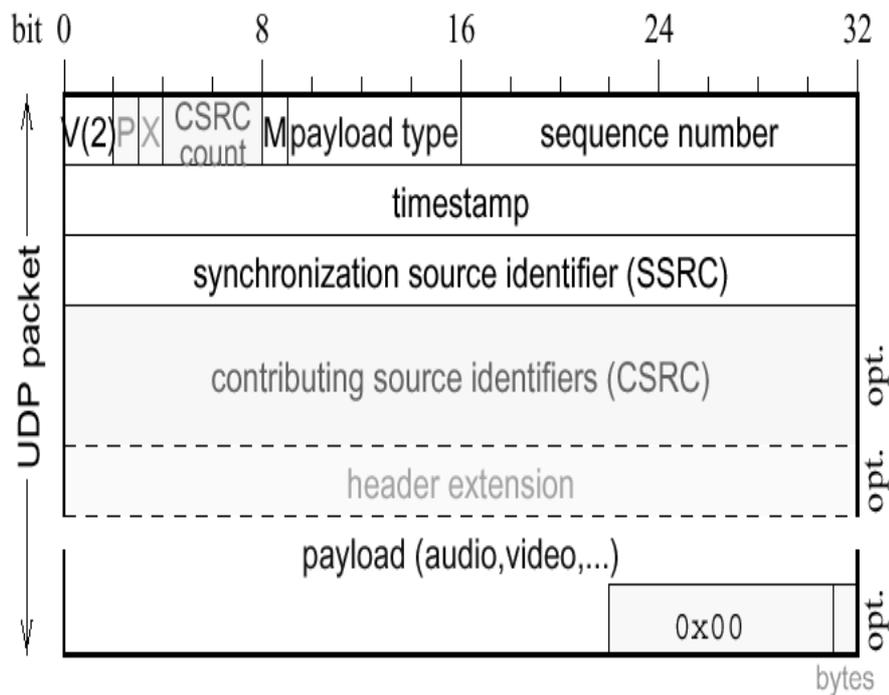
- *E* (1 bit): per usi futuri.
- *PT Recovery* (7 bit): XOR dei PT dei pacchetti non-FEC.
- *Mask* (24 bit): indica la lista dei pacchetti non-FEC che hanno generato questo pacchetto (ogni bit  $i$  a 1 indica la presenza del pacchetto non-FEC con numero di sequenza pari a *Sequence number* base +  $i$ ).
- *Timestamp recovery* (32 bit): XOR tra i *Timestamp* dei pacchetti non-FEC.

## Il pacchetto RTP - FEC

---

- Le estensioni dell'*header* e i *payload* dei pacchetti non-FEC sono posti in XOR per generare il payload del pacchetto FEC
  - tutte le stringhe sono portate alla stessa lunghezza mediante padding.

## Il pacchetto RTP - FEC (esempio)



Lezione 1.6, v.

6.27

## Il pacchetto RTP - FEC (esempio)

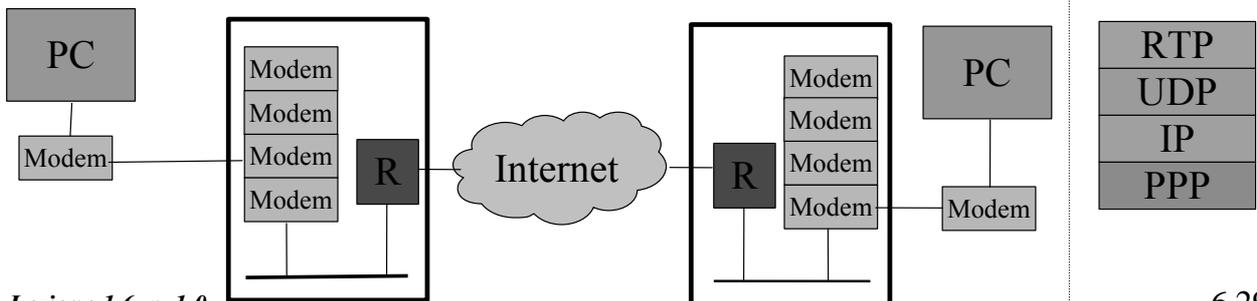
- L'esempio mostra il pacchetto di FEC ricavato da due frame (con payload statico tipo 4) G.731.1
  - il frame A a 6.3 Kbps con 24 byte di payload
  - il frame B a 5.3 Kbps con 20 byte di payload
  - Si noti che quindi il payload di B è completato con 4 byte di padding prima del XOR.
- Il pacchetto FEC aggiunge in questo caso il 41% di overhead ( $A+B+X = 116$  bytes,  $FEC(X) = 48$  bytes) e un ritardo addizionale di 30 ms più i tempi di elaborazione nella sorgente e nel ricevitore.

Lezione 1.6, v. 1.0

6.28

## Ritardi e overhead con RTP

- La trasmissione di flussi multimediali tramite RTP richiede di bilanciare due aspetti
  - ritardi end-to-end;
  - overhead.
- Si considerino due entità collegate ad Internet tramite modem:



Lezione 1.6, v. 1.0

6.29

## Ritardi e overhead con RTP

- Per calcolare il massimo jitter assorbibile:

$$j_{\text{buf}} < 300 \text{ ms} - d_{\text{coder}} - d_{\text{net}} - fs$$

- Prendendo come riferimento un codec G.723.1 si assuma:
  - $d_{\text{coder}} = 42.5 \text{ ms}$ ,
  - $fs = 30 \text{ ms}$ ,
  - ritardo Internet 40 ms,
  - ritardo dei modem 90 ms,
  - un solo frame per pacchetto RTP.

Lezione 1.6, v. 1.0

6.30

## Ritardi e overhead con RTP

---

- Il ritardo della rete è pari a 130 ms, più il tempo di pacchettizzazione, che è trascurabile:

$$d_{\text{net}} = 130 + d_{\text{pkt}} \approx 130 \text{ ms}$$

- Il buffer in ricezione potrà avere una dimensione pari a 97.5 ms.

## Ritardi e overhead con RTP

---

- Gli header influenzano notevolmente la trasmissione:
  - 12 (RTP) + 8 (UDP) + 20 (IP) + 4 (PPP) = 44 byte
  - a 5.3 kbps, un frame G.723.1 occupa  $5.3 * 30 \text{ ms} = 20 \text{ byte}$
  - l'overhead nella trasmissione è del 68.5%!
  - la banda realmente utilizzata è pari a 17.1 kbps!
- Nello scenario ipotizzato al massimo solo 2 connessioni voce possono essere presenti (38.9 kbps).

## Ritardi e overhead con RTP

---

- Accorpendo più frame G.723.1 nello stesso pacchetto l'overhead può essere ridotto
  - ad esempio, con 4 frame
    - » l'overhead diventa del 35.5%,
    - » la banda diventa di 8.3 kbps.
- Purtroppo con questa operazione il ritardo di pacchettizzazione non è più trascurabile:  $d_{\text{pkt}} = 90 \text{ ms}$ 
  - $j_{\text{buf}} < 7.5 \text{ ms}$ , impraticabile per Internet.

## Ritardi e overhead con RTP

---

- Una possibile soluzione consiste nella compressione delle intestazioni (RTP/UDP/IP)
  - si invia un pacchetto non compresso seguito da pacchetti compressi:
    - » i campi costanti sono rimossi (es. *Version* per RTP),
    - » i campi che differiscono per una costante sono rimossi (es. *Sequence number* per RTP),
    - » per i campi che differiscono per una quantità variabile viene inviata la differenza (es. *Timestamp* per RTP);
  - per la maggior parte dei pacchetti l'intestazione risultante è di soli due byte!

## Ritardi e overhead con RTP

---

- La compressione può essere applicata
  - *link-to-link*
  - *end-to-end*
    - » in questo caso ogni router intermedio deve conoscere la struttura dei pacchetti compressi per poterli instradare.
- Il vero vantaggio si ha nell'utilizzare la compressione per le linee con poca banda
  - all'interno della rete
    - » la larghezza di banda è un problema minore
    - » i router potrebbero non supportare la compressione delle intestazioni.

## Sincronizzazione dei flussi

---

- Le specifiche RTP suggeriscono di inviare flussi diversi su sessioni separate
  - questo permette
    - » in ricezione di demultiplexare i flussi a livello UDP;
    - » di richiedere diversi livelli di QoS.
- Se i flussi necessitano di essere sincronizzati al ricevitore è necessario fornire un riferimento temporale comune
  - l'orologio di sistema, se le applicazioni risiedono sullo stesso *end-point*,
  - l'NTP, se i flussi sono generati su diversi sistemi
    - » in questo caso il riferimento comune è comunicato alla destinazione tramite RTCP.

## ***Real-Time Control Protocol (RTCP)***

---

- Lavora insieme al RTP
  - permette di realizzare sessioni multicast
    - » conferenze, streaming multimediale
  - svolge una serie di funzionalità base di controllo
    - » si limita principalmente alla gestione della trasmissione
    - » tutte le altre funzionalità di controllo della conferenza sono relegate ai protocolli di segnalazione (es. H.323, SIP).
- RTCP è stato pensato per sessioni multicast su IP
  - numero di partecipanti dell'ordine delle migliaia;
  - architettura completamente distribuita.

## ***Real-Time Control Protocol (RTCP)***

---

- Ogni partecipante ad una sessione RTP invia periodicamente pacchetti di controllo RTCP a tutti gli altri partecipanti (multicast):
  - ogni pacchetto contiene informazioni statistiche del ricevitore/trasmettitore utili alle applicazioni.
    - » fra le informazioni inviate si trovano, i pacchetti inviati/ricevuti, la percentuale delle perdite, il *jitter*,...
  - questa retroazione può essere usata per verificare le prestazioni e per diagnostica; eventualmente la si può usare per modificare le caratteristiche della trasmissione (parametri della o tipo di compressione, risoluzione, ...).

## RTCP

---

- RTCP può essere usato per sincronizzare diversi *stream* all'interno di una sessione
- Si consideri una videoconferenza in cui ciascuna sorgente genera uno *stream* RTP per il video e uno per l'audio
  - i *timestamp* dei pacchetti RTP sono legati ai *clock* dei singoli campionatori, ma non hanno un riferimento temporale comune
  - i pacchetti RTCP contengono un riferimento assoluto di tempo e il *timestamp* dell'ultimo pacchetto, per cui permettono di ricostruire un sincronismo, ad esempio, fra voce e video.

## Pacchetti RTCP

---

- RTCP prevede 4 tipi di pacchetti principali
  - *Source description*;
  - *Sender Report*;
  - *Receiver report*;
  - *Goodbye*.
- Anche i pacchetti RTCP vengono trasportati tramite UDP ma usano una porta diversa rispetto a RTP.
- Per ridurre l'*overhead* diversi pacchetti RTCP possono essere combinati e trasmessi assieme (*compound RTCP packet*).

## Pacchetti RTCP

### Source Description (SDES)

---

- L'SSRC identifica in modo univoco tutti i partecipanti ad una sessione RTP.
- I pacchetti SDES forniscono informazioni più puntuali associati ad ogni SSRC:
  - identificativo univoco (comune a tutte le sessioni RTP a cui si partecipa),
  - nome utente,
  - email,
  - numero di telefono.

## Pacchetti RTCP

### Sender Report (SR)

---

- Inviati dalle sorgenti per fornire informazioni sui pacchetti trasmessi.
- Contengono:
  - SSRC della sorgente;
  - NTP *timestamp*, se diverse sorgenti non dispongono di un riferimento temporale comune i loro flussi non possono essere sincronizzati;
  - RTP *timestamp*, il riferimento utilizzato per generare i flussi RTP;

## Pacchetti RTCP

### Sender Report (SR)

---

- numero di pacchetti inviati, dall'inizio della trasmissione;
- numero di ottetti inviati, dall'inizio della trasmissione
  - » vengono ignorati le intestazioni e i byte di riempimento.
- eventuali informazioni sulla ricezione
  - » si evita di trasmettere un apposito pacchetto *Receiver Report*.

## Pacchetti RTCP

### Receiver Report (RR)

---

- Forniscono un feedback alle sorgenti sulla qualità della ricezione
  - vengono inviati anche in assenza di arrivo di pacchetti RTP.
- Contengono
  - SSRC della sorgente;
  - EHSN, il maggior numero di sequenza ricevuto (ha dimensione tale da non riavvolgersi);
  - CPL, numero totale di pacchetti persi (calcolato come differenza tra il numero di pacchetti arrivati e quello atteso;

## Pacchetti RTCP Receiver Report (RR)

---

- FPL, frazione di pacchetti persi;
- *jitter* di interarrivo, stima della varianza dei ritardi di arrivo dei pacchetti;
- LSR, *timestamp* NTP dell'ultimo SR ricevuto dalla sorgente;
- *DLSR*, tempo trascorso dalla ricezione dell'ultimo SR all'invio di questo RR.

## Pacchetti RTCP Goodbye (BYE)

---

- Informa gli altri partecipanti che uno o più utenti hanno lasciato la sessione.
- Contengono:
  - SSRC/CSRC degli utenti che abbandonano la sessione;
  - motivazione (opzionale).
- L'abbandono della sessione può essere anche implicito, attraverso il mancato invio di pacchetti RTCP per un certo periodo di tempo.

## Scalabilità RTCP

---

- Al fine di permettere la partecipazione di migliaia di utenti alle sessioni RTP, si devono affrontare due fattori di scalabilità
  - banda
    - » in genere sono attivi pochi trasmettitori alla volta,
    - » è necessario evitare che i flussi RTCP congestionino la rete;
  - memorizzazione
    - » ogni partecipante deve memorizzare tutti gli identificativi SSRC della sessione RTP.

## Scalabilità RTCP

---

- Ogni partecipante deve memorizzare in modo indipendente lo stato della sessione
  - la struttura distribuita è molto robusta agli errori;
  - lo stato consiste in informazioni riguardanti
    - » il flusso multimediale trasmesso;
    - » la qualità del flusso ricevuto;
    - » gli altri partecipanti.
  - per mantenere la consistenza delle informazioni ogni partecipante deve periodicamente aggiornare le informazioni che lo riguardano.

## Scalabilità RTCP

---

- Nel caso di sessioni multicast il traffico RTCP generato crescerebbe linearmente con il numero di utenti
  - con migliaia di utenti si congestionerebbe la rete;
  - il protocollo prevede di limitare superiormente la banda disponibile in modo indipendente dal numero di partecipanti
    - » tale limite viene fissato in funzione del traffico RTP generato.
- Sulla base del limite di banda a disposizione ogni utente calcola gli intervalli entro cui emettere i *RR*.

## Scalabilità RTCP

---

- Il traffico RTCP deve essere limitato al 5% della banda della sessione RTP
  - tale calcolo comprende l'overhead introdotto dal livello di trasporto e di rete;
  - questa banda viene utilizzata
    - » equamente tra trasmettitori e ricevitori, se i trasmettitori sono più del 25% dei partecipanti,
    - » il 25% è riservata ai trasmettitori ed il 75% ai ricevitori, se il numero dei trasmettitori è inferiore al 25% dei partecipanti.

## Scalabilità RTCP

---

- L'intervallo di trasmissione teorico  $T$  per i pacchetti RTCP viene calcolato come:

Per le sorgenti

$$T = (\text{dim pacchetto}) * (\text{num. sorgenti}) / (0,25 * 0,05 * \text{Banda})$$

Per i ricevitori

$$T = (\text{dim pacchetto}) * (\text{num. ricevitori}) / (0,75 * 0,05 * \text{Banda})$$

- Esiste un intervallo minimo, pari a 5 secondi
  - per evitare burst di pacchetti nel caso di intervalli ridotti.
- L'intervallo effettivo  $T_e$  viene calcolato come:

$$T_e = T \times x, \quad x \text{ v.a. } \in [0.5, 1.5]$$

- in modo da desincronizzare le trasmissioni.

## RTP - RTCP

---

- Alcuni esempi di applicazioni che usano RTP-RTCP sono
  - strumenti di Mbone: vic/vat/rat;
  - Real Media (Real Audio, Real Video);
  - MS Netmeeting;
  - Apple Quicktime;
  - H.323.
- Attualmente le varie applicazioni non sono sempre in grado di comunicare fra loro
  - per incompatibilità slegate da RTP;
  - per piccoli scostamenti dallo standard.

## Conferenze multimediali

---

- Esistono diverse metodologie di instaurazione di conferenze multimediali
  - a chiamata
    - » uno o più partecipanti avviano la chiamata verso altri,
    - » simile all'approccio telefonico,
    - » in genere ristretta ad un numero limitato di partecipanti;
  - su annuncio
    - » la conferenza viene annunciata con un certo anticipo,
    - » gli interessati si associano,
    - » usata ad es. per *e-learning*, manifestazioni pubbliche.

Lezione 1.6, v. 1.0

6.53

## Conferenze multimediali

---

- Nel secondo caso, è necessario disseminare in anticipo l'informazione sulla conferenza
  - gli utenti vengono informati sull'inizio, le durate, i contenuti e le modalità di partecipazione;
  - gli utenti possono partecipare al momento di inizio della sessione
    - » RTP/RTCP possono essere utilizzati per trasportare i flussi multimediali e le informazioni di controllo;
  - occorre un protocollo per propagare questo tipo di informazione: *Session Description Protocol*.

Lezione 1.6, v. 1.0

6.54

## ***Session Description Protocol (SDP)***

---

- SDP describe

- le informazioni
- le codifiche

necessarie per partecipare ad una “conferenza multimediale”.

- Le informazioni possono essere distribuite attraverso

- e-mail
- web
- multicast ...

## **SDP**

---

- La struttura della descrizione è suddivisa in tre sezioni

- *sessione*: identifica e descrive i contenuti della sessione;
- *tempistica*: informazioni temporali riguardanti l'inizio e la durata della sessione;
- *trasmissione*: riguarda soltanto i media e contiene le informazioni necessarie per trasmettere e ricevere i flussi.

## SDP

---

- La descrizione SDP consiste in una serie di linee del tipo:

`<type>=<value>`

- `<type>` consiste in un singolo carattere;
- `<value>` è una stringa alfanumerica.
- Si consideri un ipotetico annuncio per la trasmissione multicast del corso di Telematica
  - il corso si svolge nelle giornate di lunedì (8-10), martedì (8-10) e giovedì (10-12);
  - le sere di sabato e domenica vengono trasmesse le repliche (3-6).

## SDP - Esempio

---

- **Session Description**

- 1) v = 0
- 2) o = raffaele.bolla 7543251234 7543251278 IN IP4 192.168.0.1
- 3) s = Corso di Telematica 2, Prof. Raffaele Bolla
- 4) i = Introduzione alle reti multimediali: H.323 e SIP
- 5) u = <http://www.reti.dist.unige.it/telematica>
- 6) e = raffaele.bolla @dist.unige.it

- **Time Description**

- 7) t = (NTP(Monday, September 27, 2004 at 8 am)) (NTP(Thursday, December 16, 2004 at 12 am))
- 8) r = 7d 120m 0 24h 74h
- 9) t = (NTP(Saturday, October 2, 2004 at 3 am)) (NTP(Sunday, December 19, 2004 at 6 am))  
r = 7d 180m 0 24h
- 10) z = (NTP(October 31, 2004 at 2 am)) -1h

## SDP - Esempio

---

- **Media Description**

- 11) m = video 3602/3 RTP/AVP 110
- 12) a = rtpmap:110 H263-1998
- 13) c = IN IP4 234.4.6.112/127/3
- 14) m = audio 4010 RTP/AVP 4  
c = IN IP4 231.40.200.3

## SDP - Esempio

---

- 1) *Version* del protocollo.  
*v = 0*
- 2) *Origin* identificativo della sessione  
*o = raffaele.bolla 7543251234 7543251278 IN IP4 192.168.0.1*  
o=<username><session><version><network type><address>
  - username la login dell'utente sul sistema di origine;
  - session id si consiglia di usare un timestamp NTP;
  - version dell'annuncio (timestamp NTP);
  - network type, IN indica Internet;
  - address type, IP4 o IP6;
  - address, nome di dominio completo o rappresentazione numerica.

## SDP - Esempio

---

- 3) *Session-name*, un nome della sessione.  
*s = Corso di Telematica 2, Prof. Raffaele Bolla*
- 4) *Information*, descrizione della sessione.  
*i = Introduzione alle reti multimediali: H.323 e SIP*
- 5) *URI*, una pagina contenente ulteriori informazioni sulla sessione.  
*u = http://www.reti.dist.unige.it/telematica*
- 6) *Email* della persona di riferimento.  
*e = raffaele.bolla @dist.unige.it*

## SDP - Esempio

---

- 7) *Time* indica l'inizio e la fine della sessione.  
*t = (NTP(Monday, September 27, 2004 at 8 am))*  
*(NTP(Thursday, December 16, 2004 at 12 am))*
- 8) *Repeat* definisce gli intervalli di ripetizione della sessione  
*r = 7d 120m 0 24h 74h*  
- *r=<repeat interval><active duration><list of offset from start-time>*
- 9) La sessione viene ripetuta negli orari notturni.  
*t = (NTP(Saturday, October 2, 2004 at 3 am)) (NTP(Sunday, December 19, 2004 at 6 am))*  
*r = 7d 180m 0 24h*
- 10) *Zone* permette di tenere conto delle modifiche dovute all'ora legale  
*z = (NTP(October 31, 2004 at 2 am)) -1h*  
- *z=<adjustment time> <offset> ...*  
- gli orari sono modificati di *<offset>*

## SDP - Esempio

---

### 11) *Media*, media trasportati

*m = video 3602/3 RTP/AVP 110*

- `m=<media><port>[</number of ports>]<transport><fmt list>`
  - » media: audio, video, application, data, control
  - » port, pari per RTP, il numero successivo dispari viene utilizzato per RTCP;
  - » number of ports nel caso di codificatori a più livelli;
  - » transport, il protocollo di trasporto (RTP/AVP si riferisce a RTP con profilo Audio/Video definito nell'RFC 1890);
  - » fmt list, una lista di payload type utilizzati, permette di definire i valori per i payload dinamici.

## SDP - Esempio

---

### 12) *Extension*, per definire i payload dinamici per RTP

*a = rtpmap:110 H263-1998*

- `a=rtpmap:<payload type><encoding name>[<clock rate>[</encoding parameter>]]`

### 13) *Connection*, il tipo di connessione da stabilire

*c = IN IP4 234.4.6.112/127/3*

- `c=<network type><address type><connection address>`

### 14) Lo stesso tipo di informazioni per l'audio.

*m = audio 4010 RTP/AVP 4*

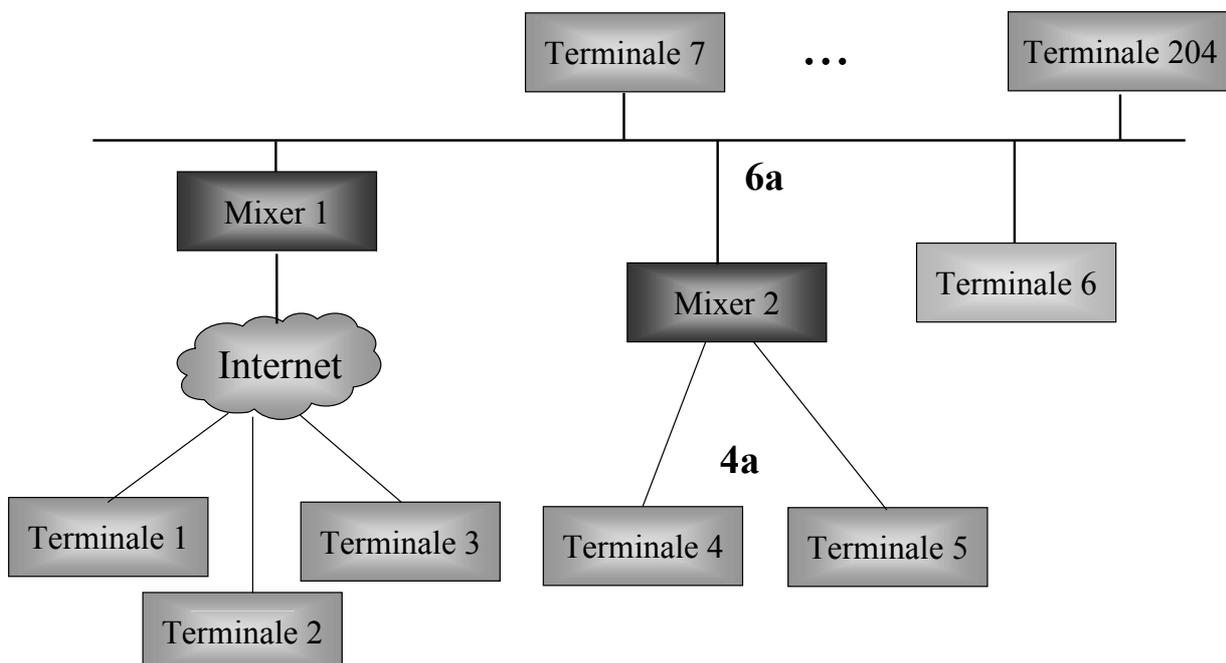
*= IN IP4 231.40.200.3*

c

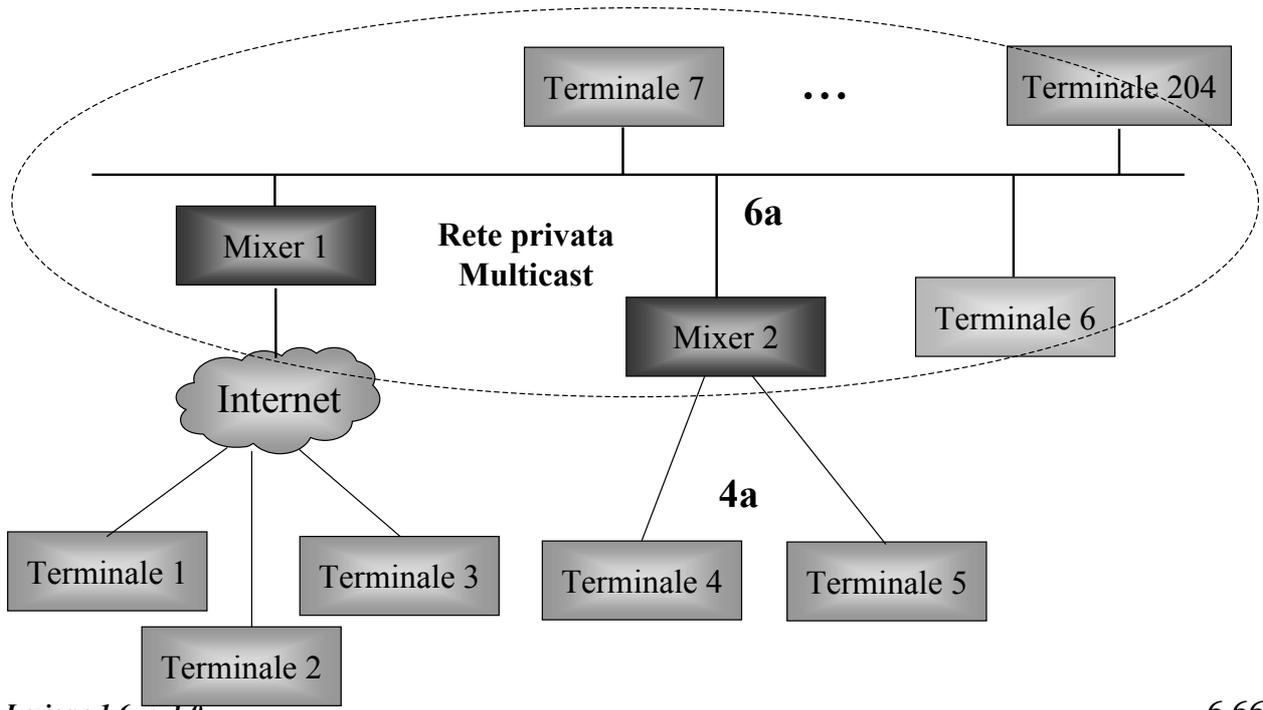
## Un esempio di conferenza multimediale

- Un esempio conclusivo può essere utile per comprendere meglio il funzionamento dei protocolli RTP/RTCP/SDP.
- Si consideri la sessione SDP annunciata in precedenza
  - audio G.723.1 a 5.3 kb/s,
    - » la banda realmente utilizzata inviando un solo frame per pacchetto è 16 kb/s;
  - video H.263+ codificato a 3 livelli,
    - » rate trasmissivi di 20, 30 e 50 kb/s.

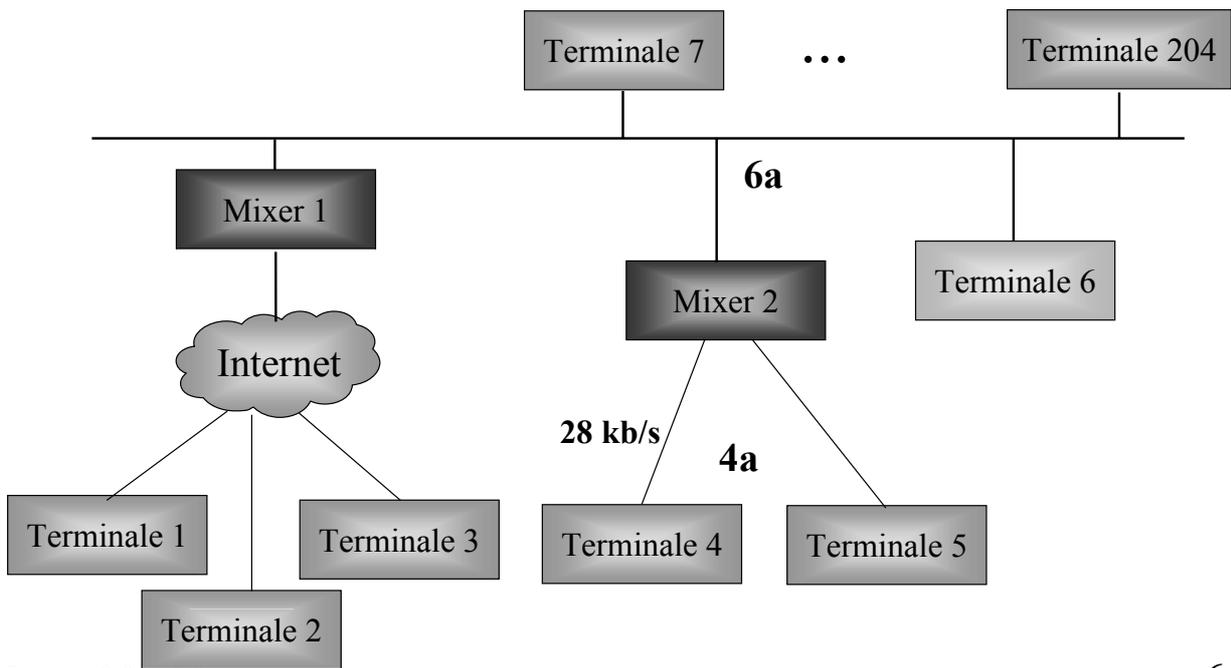
## Un esempio di conferenza multimediale I partecipanti



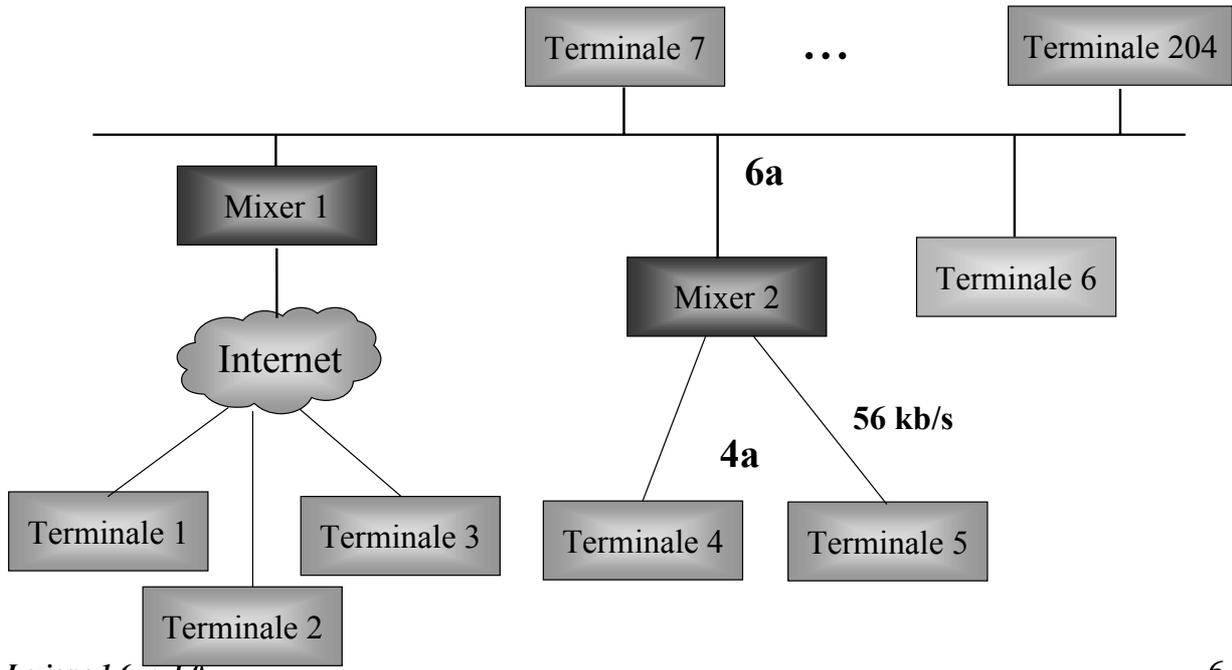
# Un esempio di conferenza multimediale I partecipanti



# Un esempio di conferenza multimediale I partecipanti



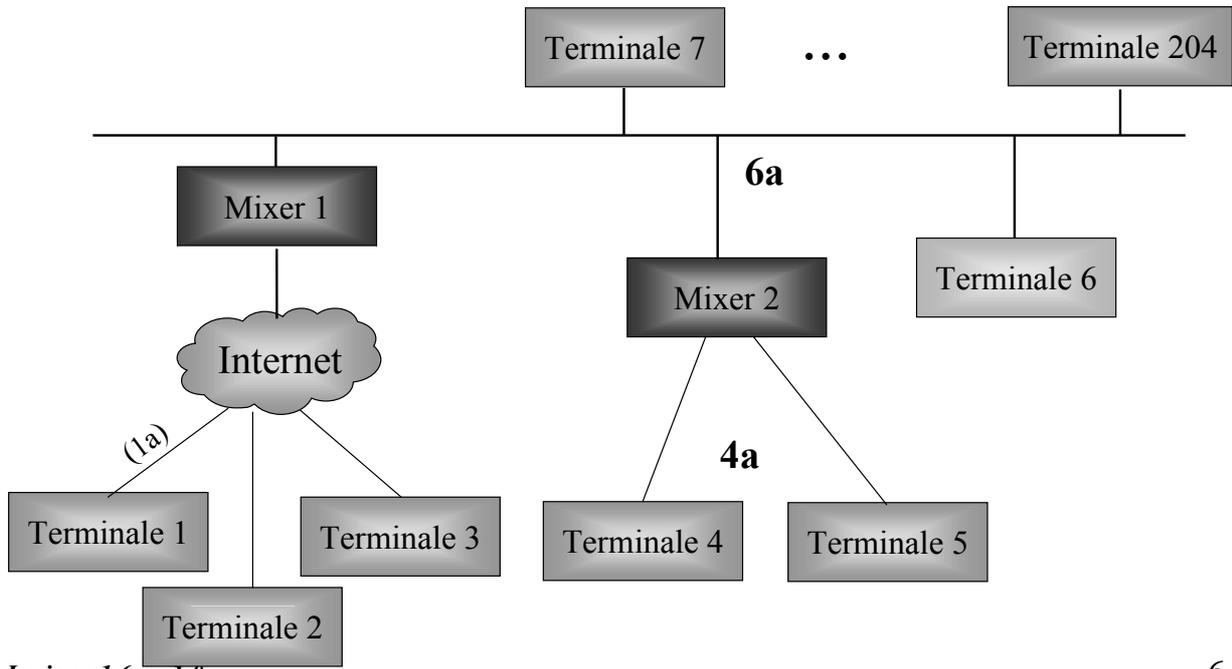
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

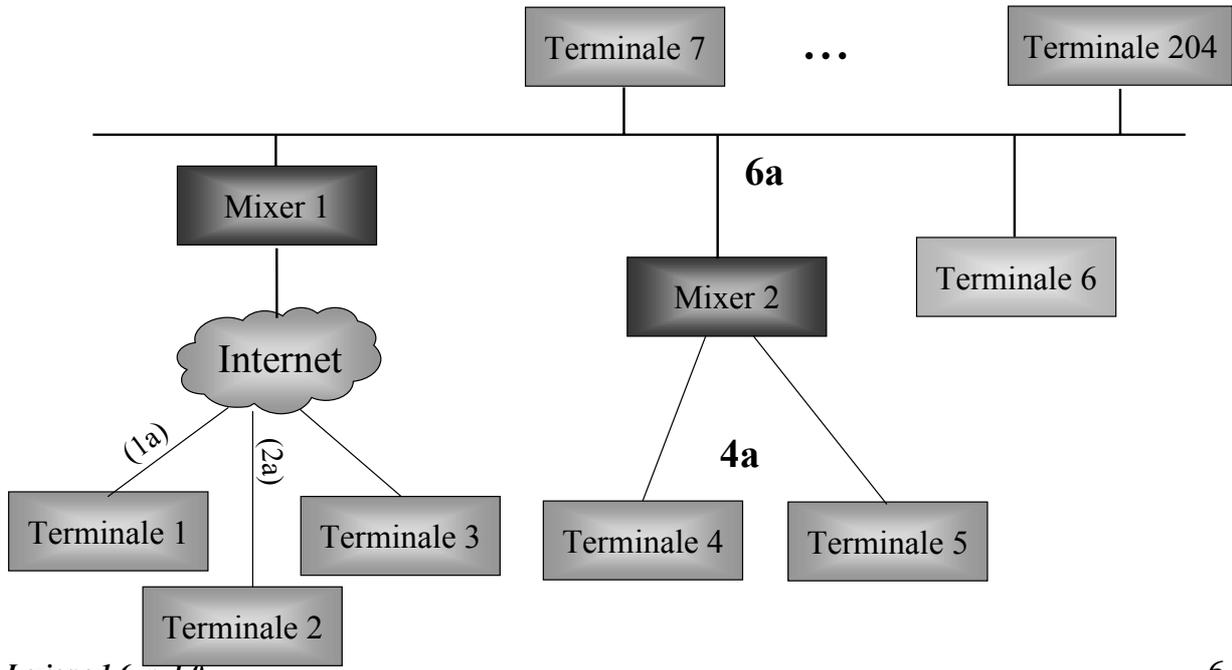
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

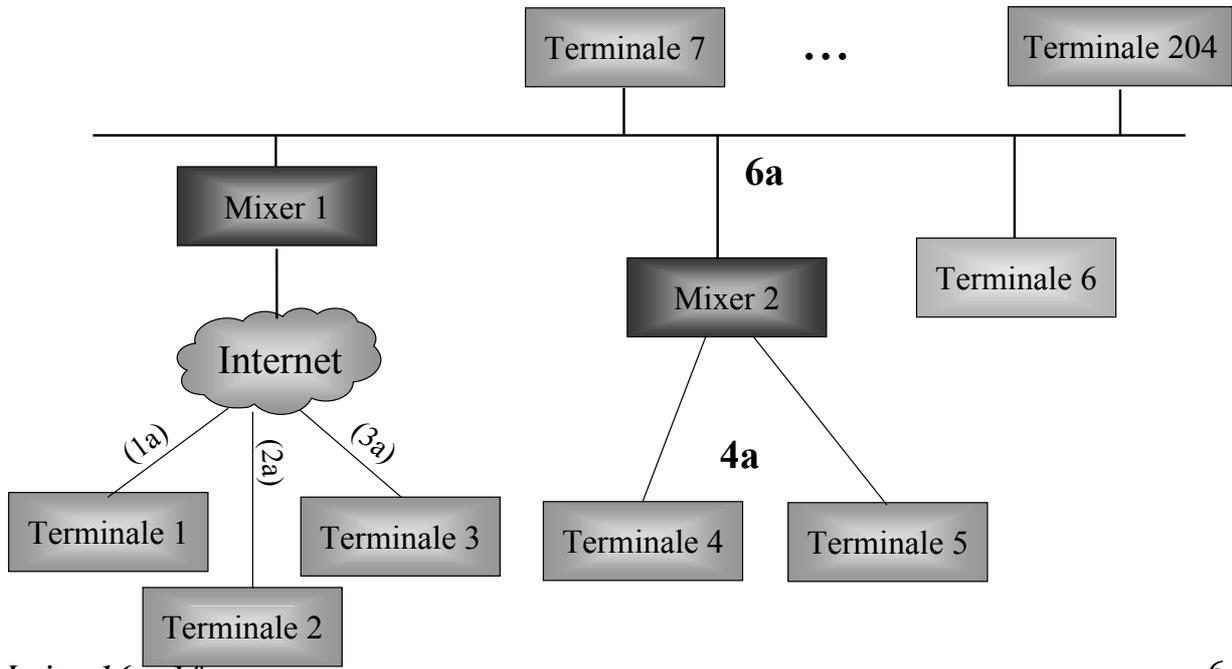
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

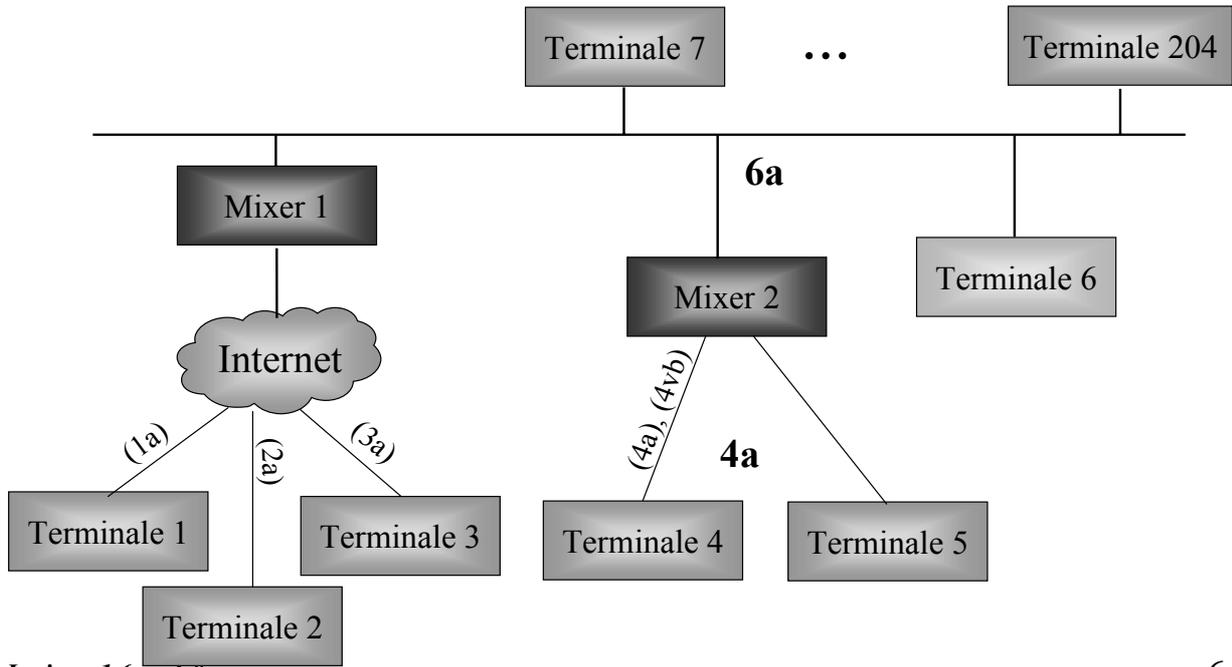
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

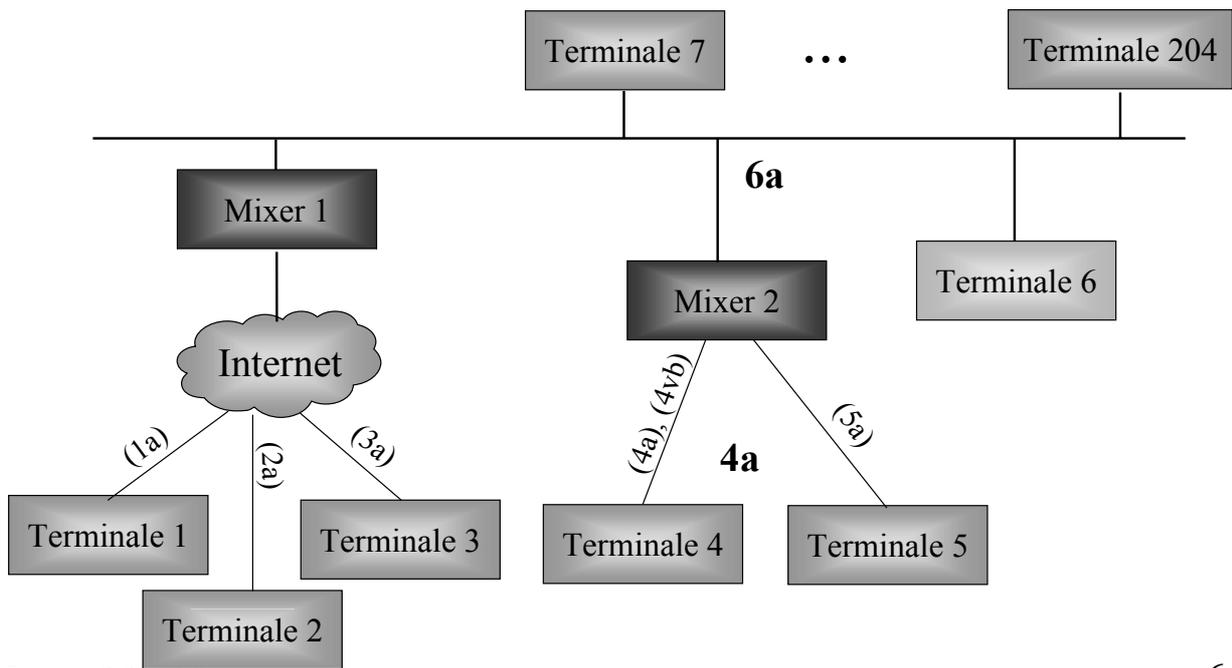
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

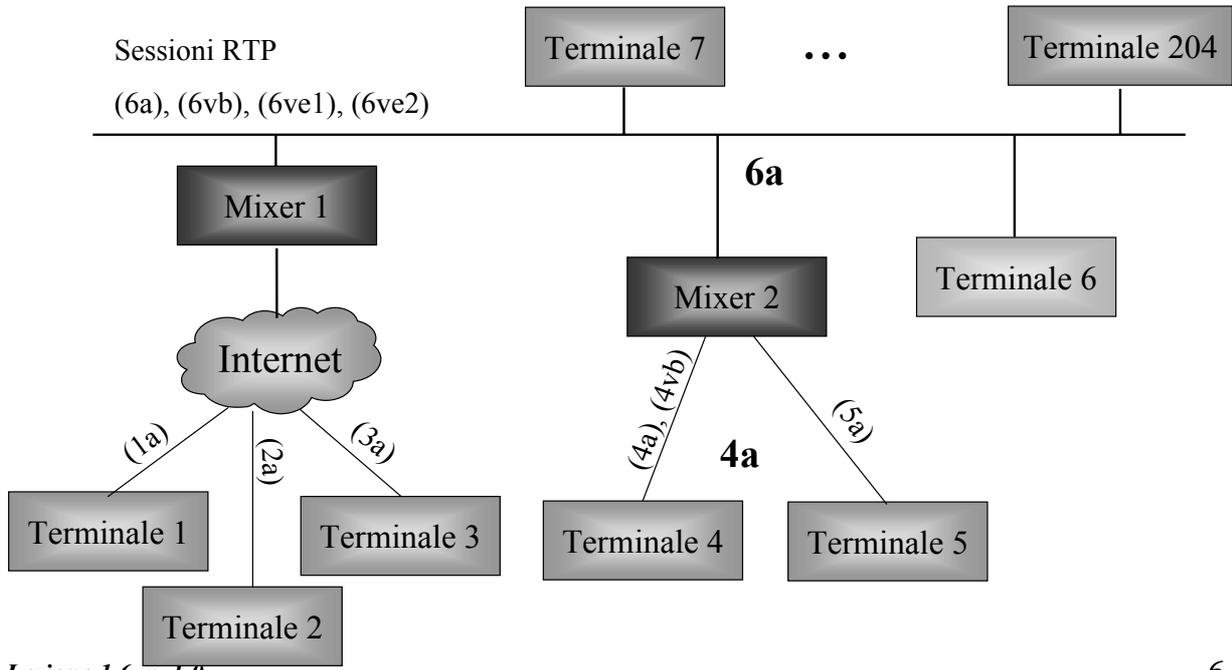
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

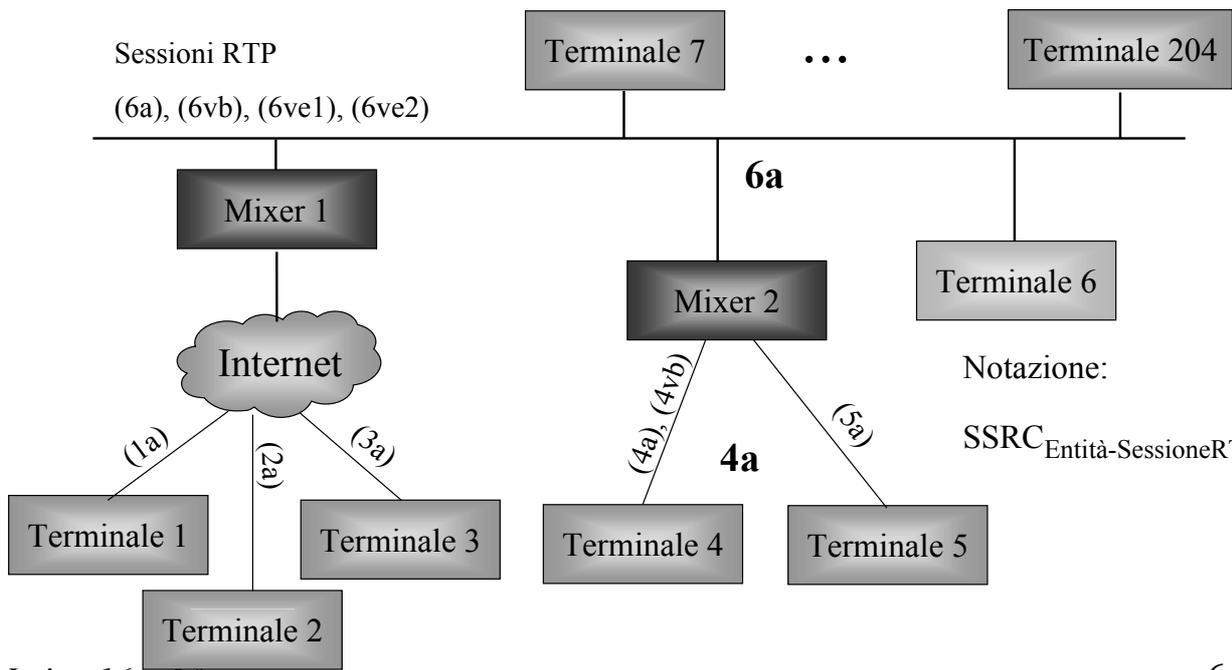
6.66

# Un esempio di conferenza multimediale I partecipanti



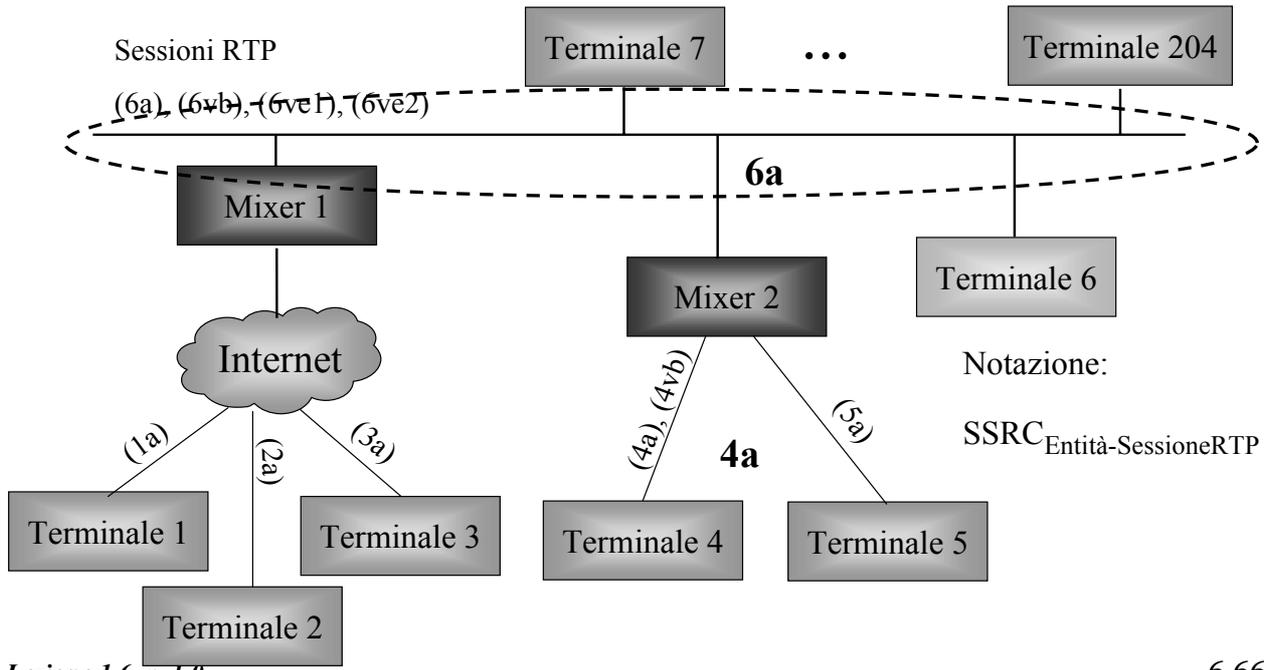
6.66

# Un esempio di conferenza multimediale I partecipanti



6.66

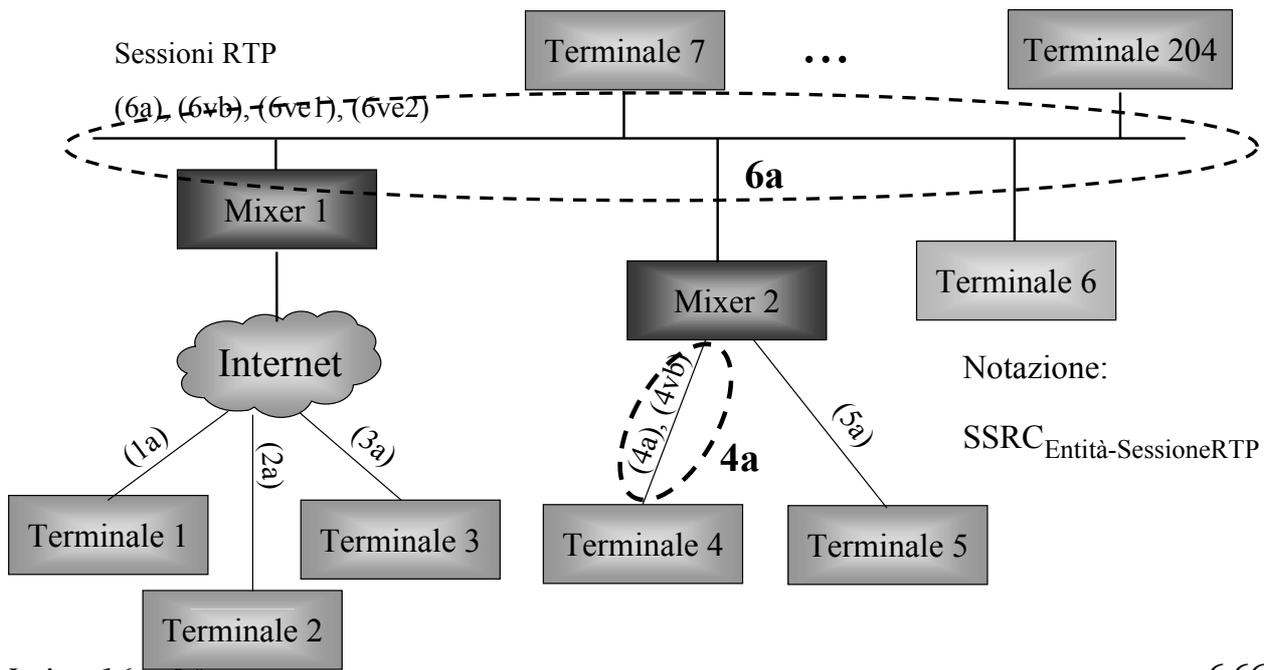
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

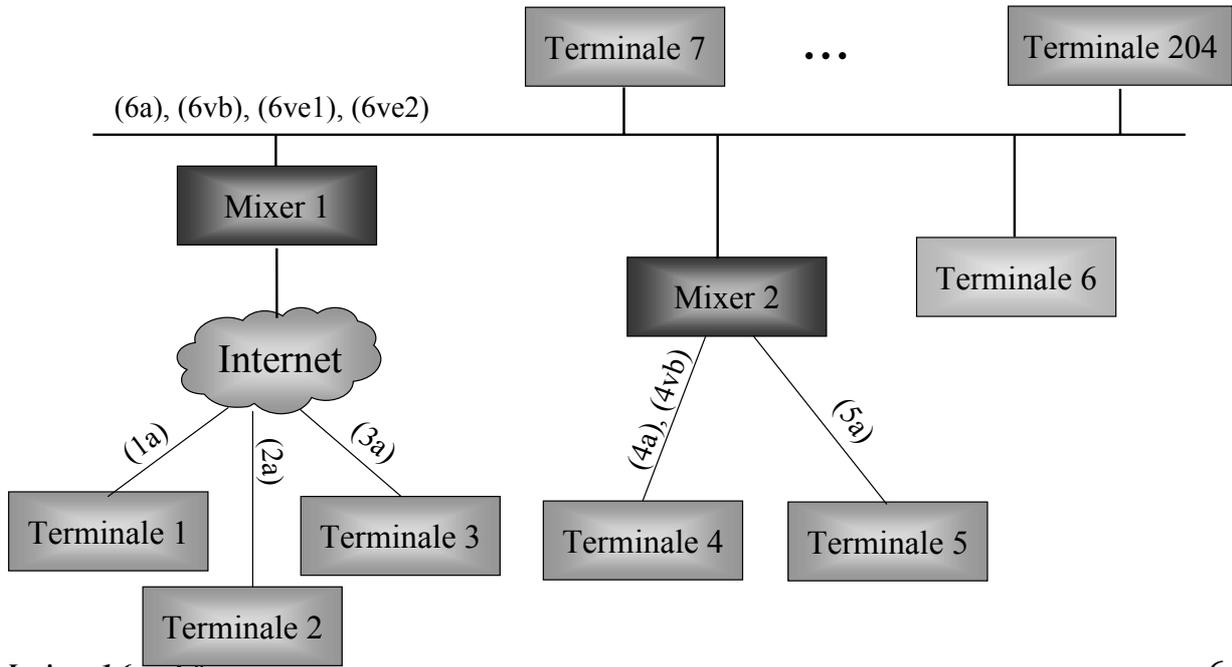
# Un esempio di conferenza multimediale I partecipanti



Lezione 1.6, v. 1.0

6.66

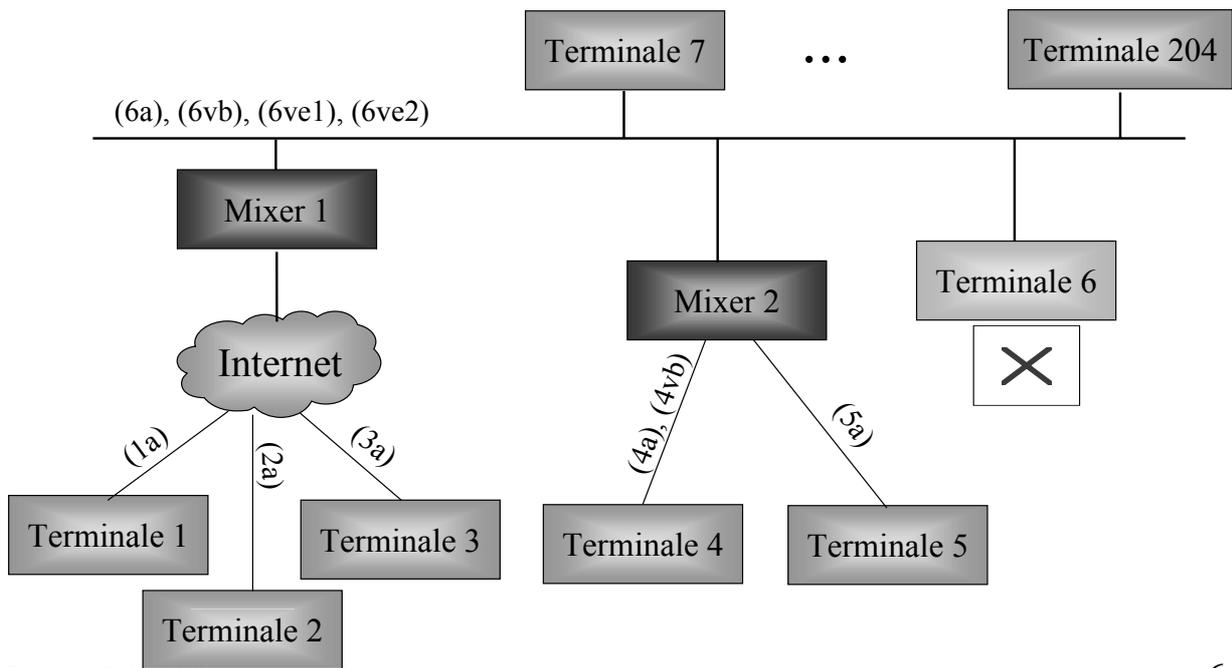
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.67

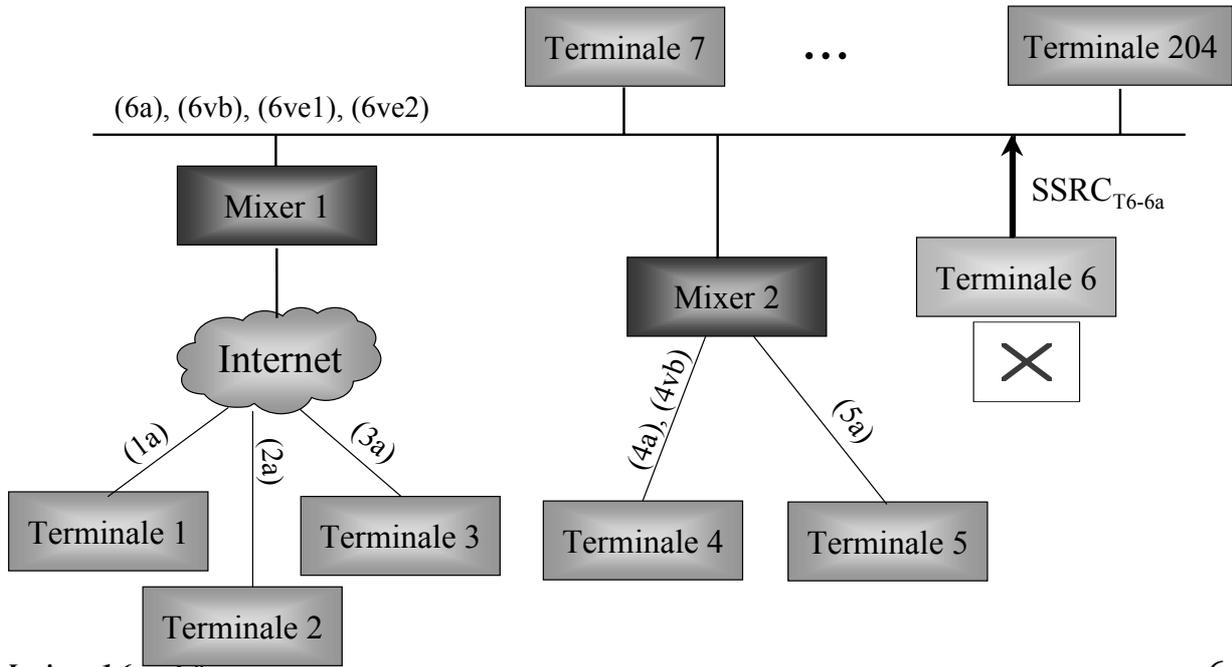
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.67

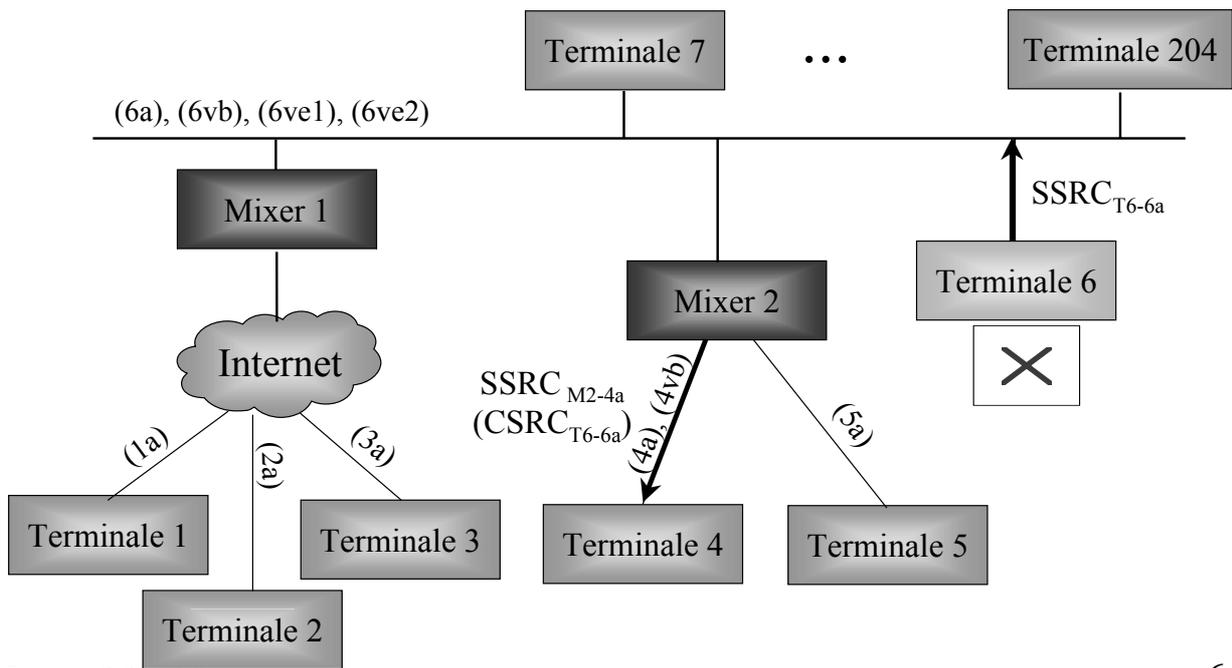
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.67

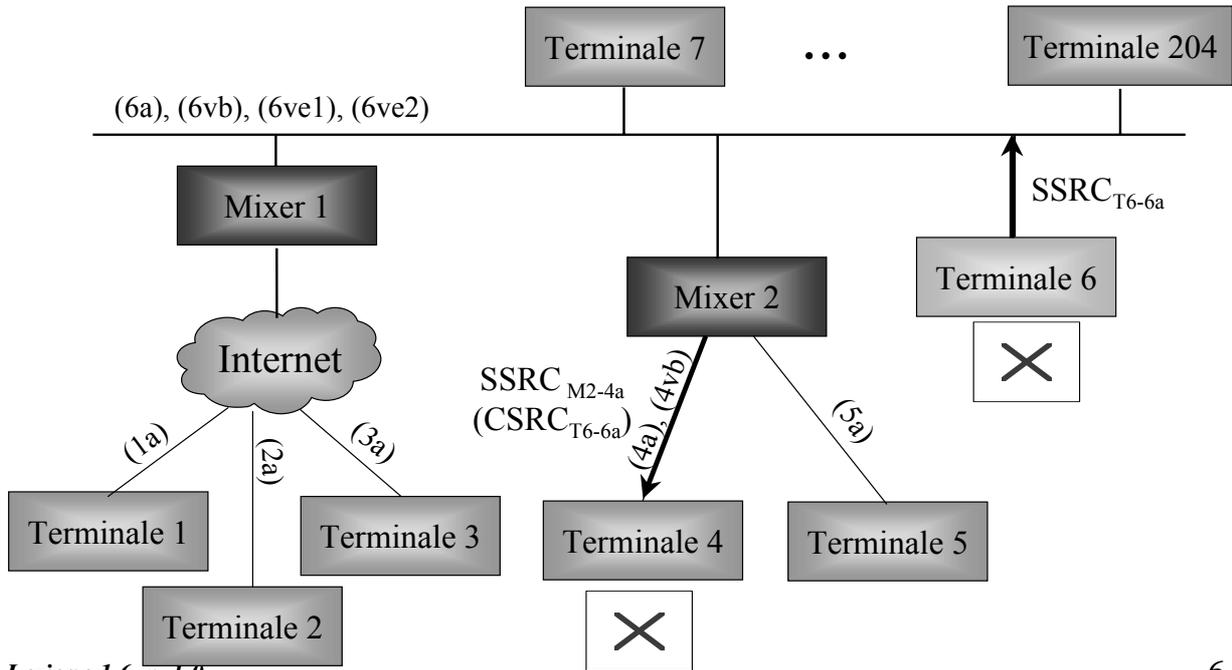
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.67

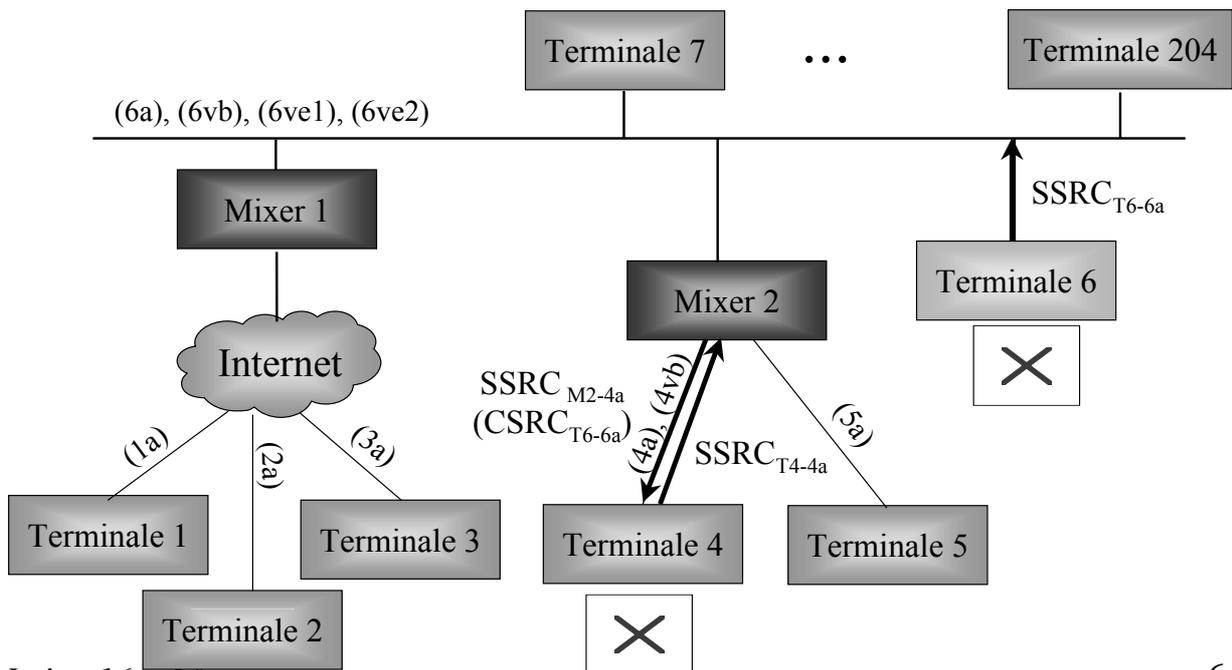
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.67

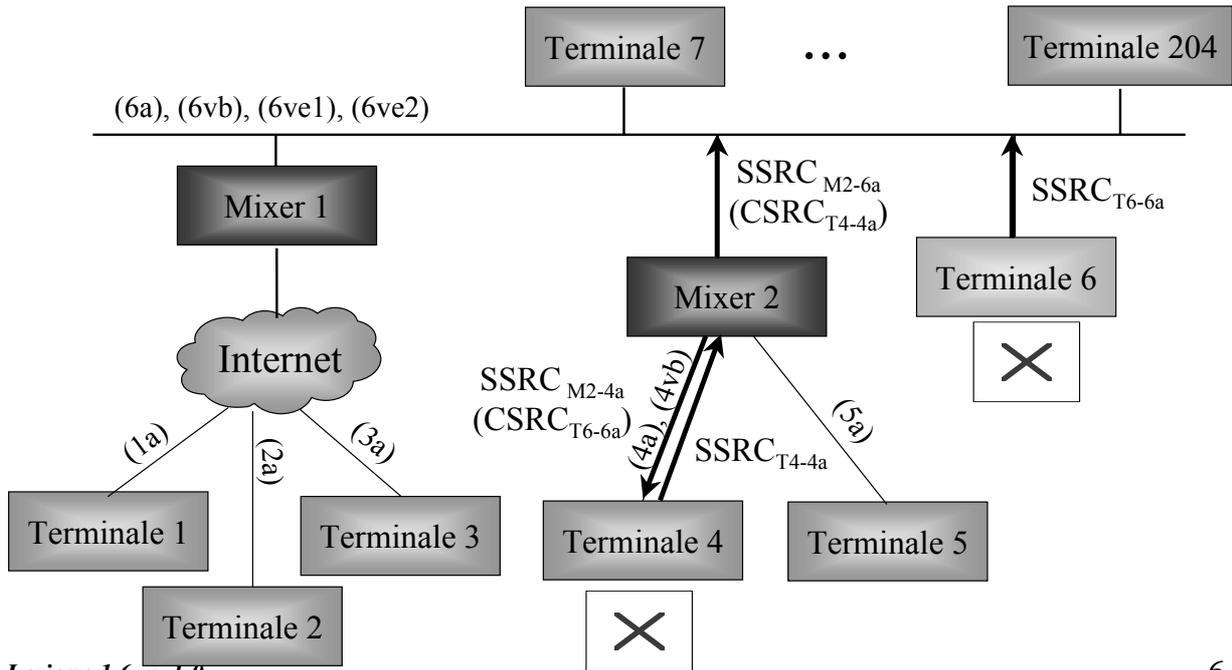
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

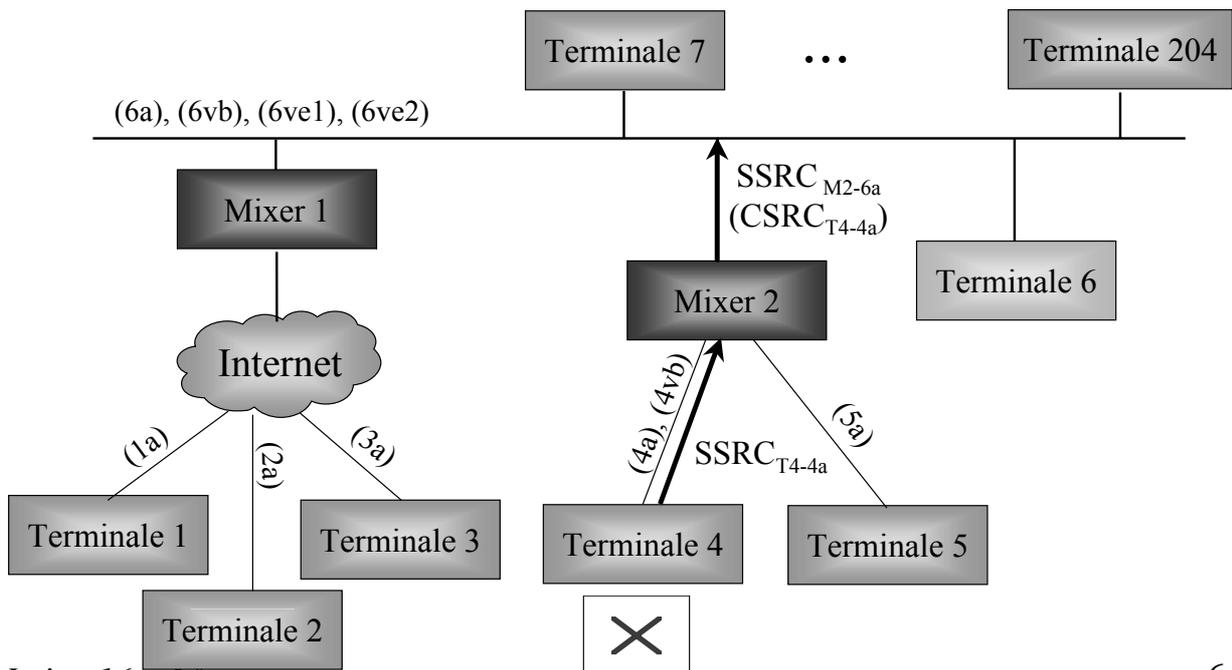
6.67

# Un esempio di conferenza multimediale



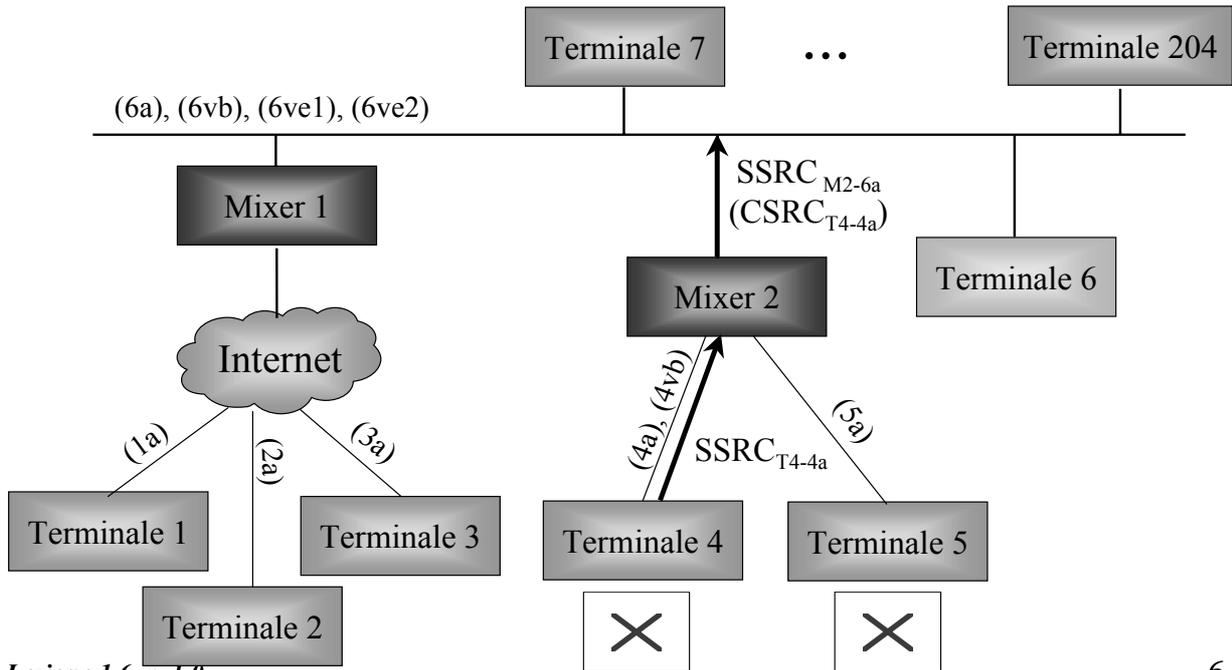
Lezione 1.6, v. 1.0

# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

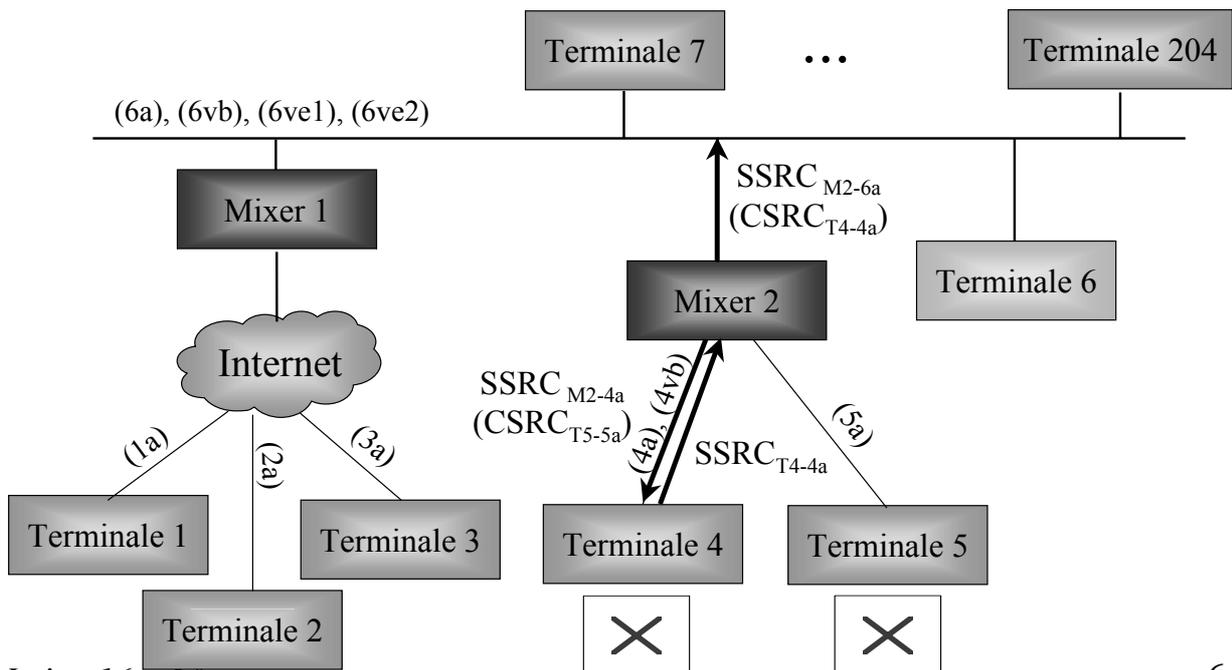
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.68

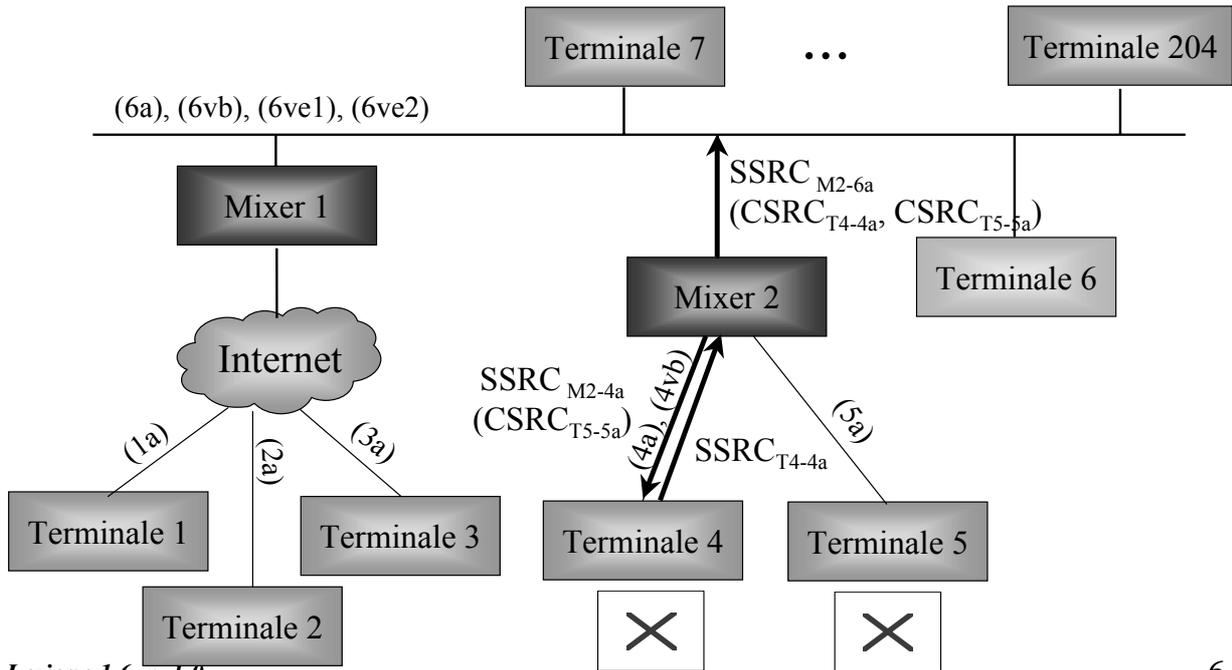
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.68

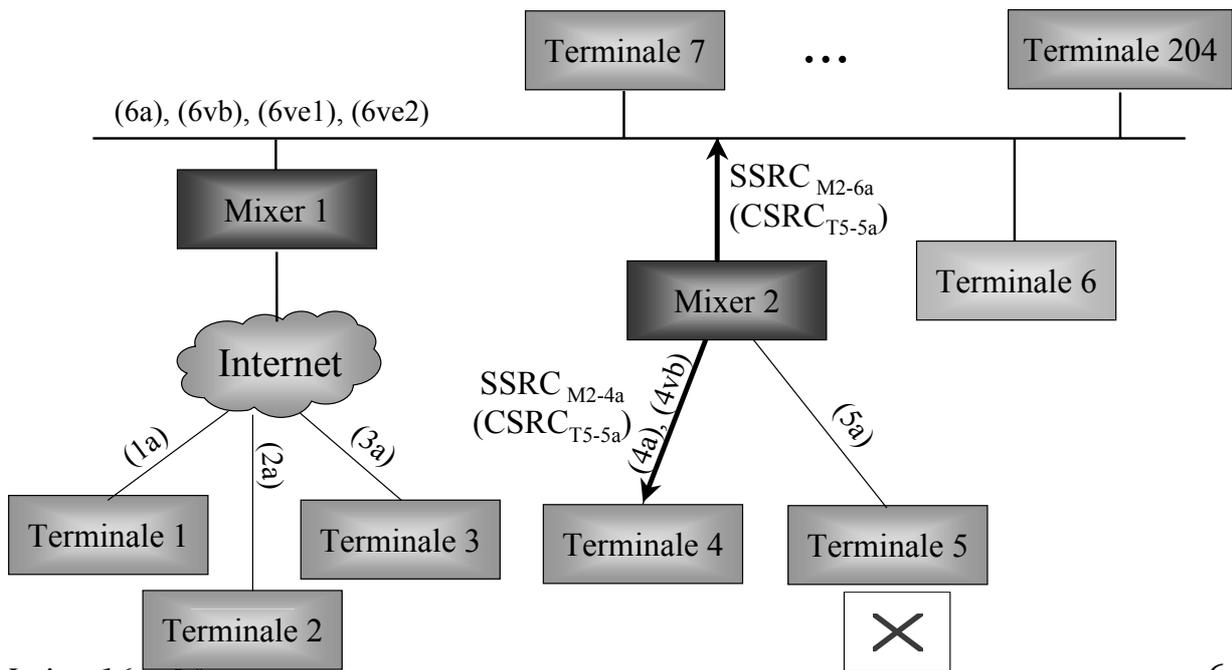
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.69

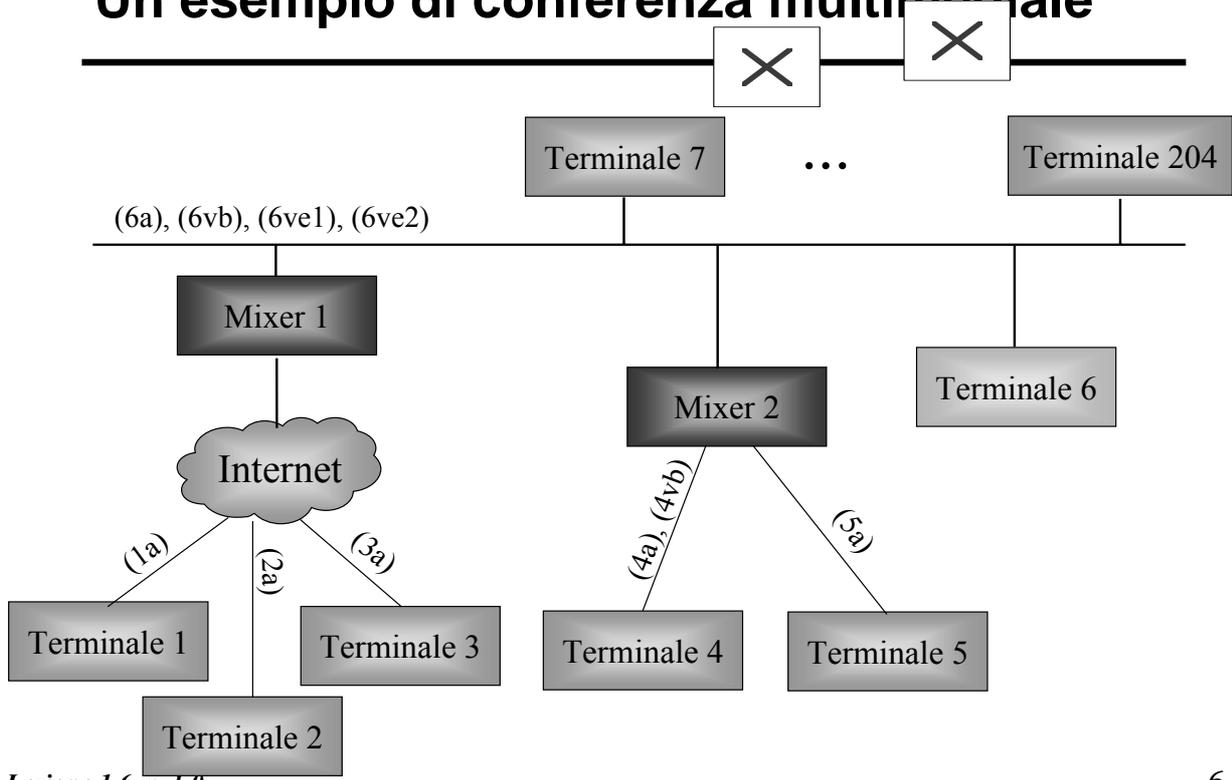
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.70

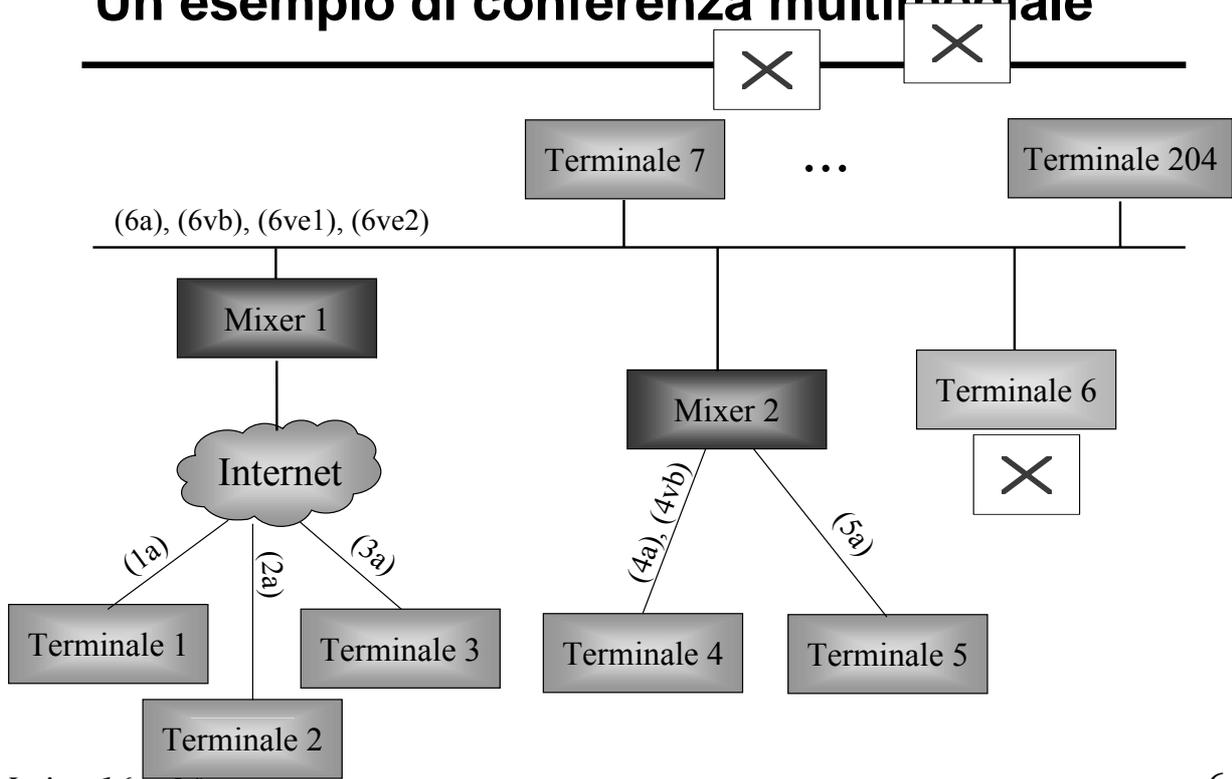
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.71

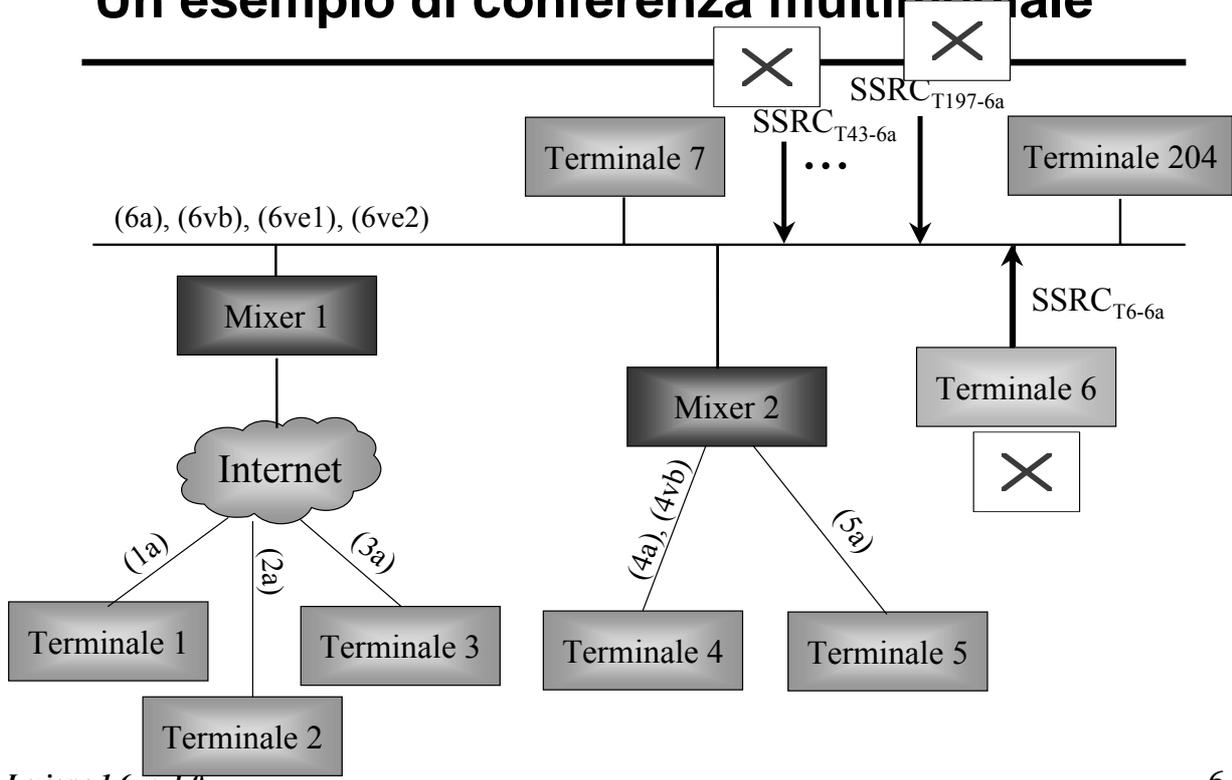
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.71

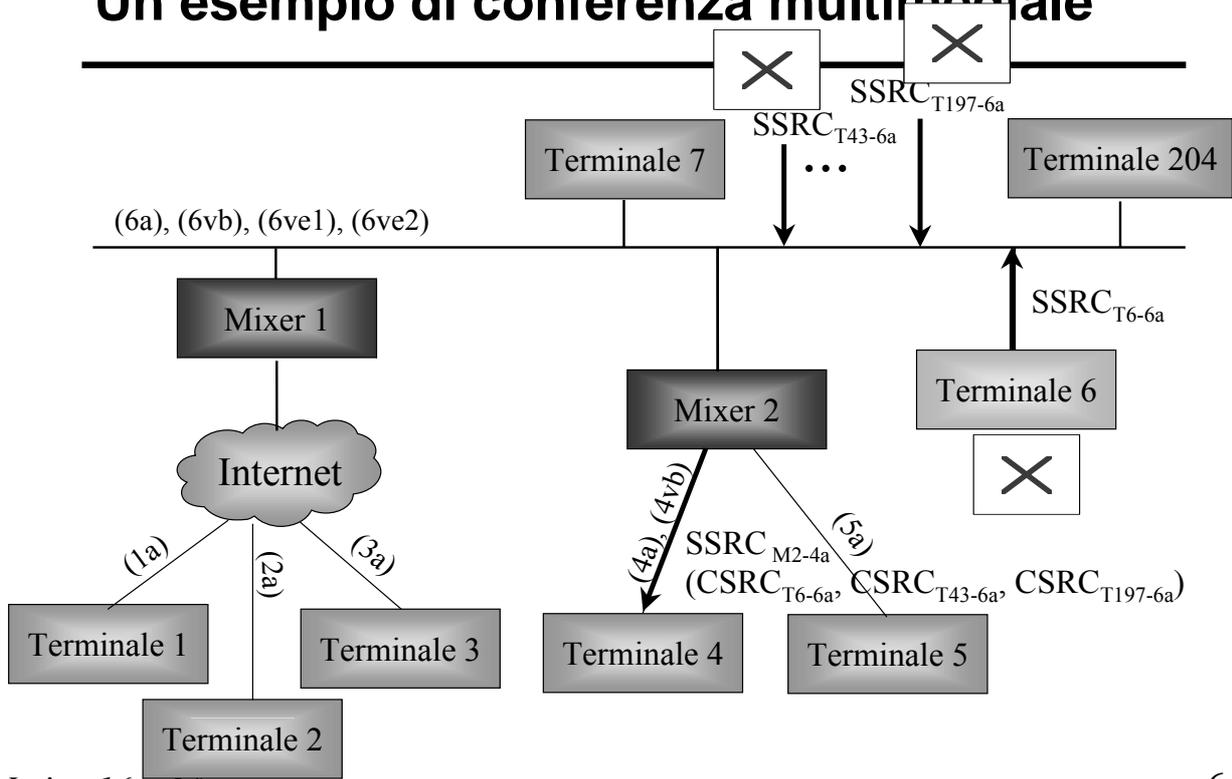
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.71

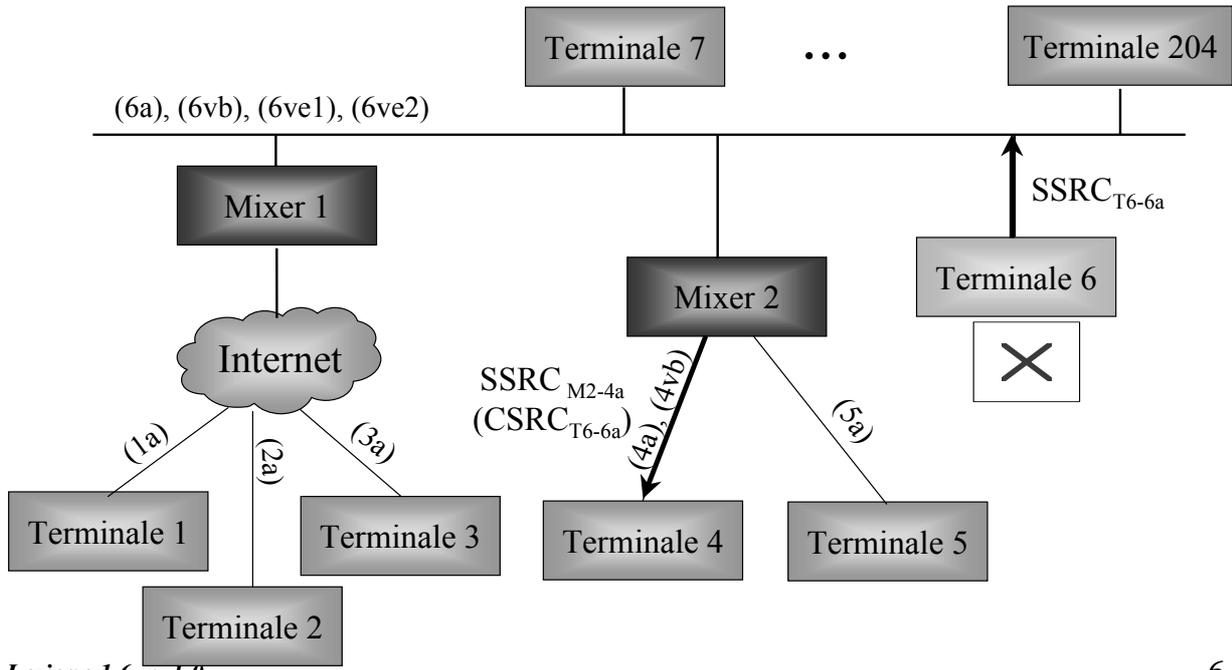
# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0

6.71

# Un esempio di conferenza multimediale



Lezione 1.6, v. 1.0