



RSVP / RTPUNIVERSITÄT DORTMUND 

Resource Reservation Protocol Realtime Transport Protocol

Seminar Multimedia-Netzwerke

Matthias Book


Lehrstuhl für Elektronische Systeme und Vermittlungstechnik
Fakultät für Elektrotechnik, Universität Dortmund



03.07.20001/27

Abstract

Dieser Vortrag stellt das Resource Reservation Protocol (RSVP) und das Realtime Transport Protocol (RTP) vor. RSVP erlaubt dem Empfänger von Datenströmen, die Reservierung von Ressourcen auf dem Datenpfad zwischen Sender und Empfänger anzufordern. RTP transportiert Echtzeitdaten in Unicast- und Multicast-Umgebungen und setzt das Kontrollprotokoll RTCP ein, um die Qualität des Transports zu überwachen. Der Schwerpunkt des Vortrags liegt mehr auf der Darstellung der Funktionsmechanismen als der exakten Spezifikation der Paketstrukturen.

RSVP / RTPUNIVERSITÄT DORTMUND 

Einleitung

- Transportprobleme bei Multimediatdaten
 - Aussetzer im Datenstrom
 - Verbindung mehrerer Teilnehmer
 - ...

- Lösungsmittel
 - Resource Reservation Protocol (RSVP)
 - Realtime Transport Protocol (RTP)

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol2/27

Protokolle wie die Komponenten der H.323-Suite definieren die Codierung von Multimedia-Daten mit dem Ziel, sie in Netzwerken zu übertragen. Die Übertragung von Multimedia-Daten unterscheidet sich jedoch in einem wesentlichen Punkt von der Übertragung „konventioneller“ Daten wie Mails oder Web-Seiten: Sie muss in Echtzeit erfolgen, d.h. der vom Sender ins Netz eingespeiste Datenstrom muss beim Empfänger **ohne Unterbrechungen** ankommen, um Aussetzer (z.B. im Videosignal) zu vermeiden. Unterbrechungen im Datenstrom werden durch Verzögerungen oder Paketverluste verursacht, die auftreten, wenn im Netz nicht genügend Bandbreite zum Transport von Paketen zur Verfügung steht. Um Echtzeitdaten störungsfrei transportieren zu können, ist daher die Reservierung entsprechender Bandbreite auf dem Übertragungspfad notwendig.

Zudem werden Multimediatdaten im Gegensatz zu „konventionellen“ Daten oft nicht Punkt-zu-Punkt übertragen, sondern (z.B. bei Konferenzen) zwischen **mehreren Teilnehmern** ausgetauscht, die unter Umständen über unterschiedlich leistungsfähige Netzanbindung und Hardware verfügen.

Die Protokolle RSVP und RTP werden eingesetzt, um diesen Problemen zu begegnen: Das **Resource Reservation Protocol (RSVP)** steuert die Reservierung der Bandbreite [RFC 2205], während der eigentliche Transport von Echtzeitdaten zwischen den Teilnehmern vom **Realtime Transport Protocol (RTP)** geleistet wird [RFC 1889].

RSVP / RTP

UNIVERSITÄT DORTMUND 


YOU ARE HERE

- Einleitung
 - RSVP
 - Funktionsweise
 - Nachrichten und Zustände
 - Reservierungs-Mechanismus
 - RTP

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol 3/27

Wir befassen uns zunächst mit dem Resource Reservation Protocol (RSVP) und gehen danach auf das Realtime Transport Protocol (RTP) ein.

Wir gehen dabei top-down vor, indem wir zunächst die grobe Funktionsweise von RSVP betrachten, dann die wichtigsten Nachrichten und Zustände kennenlernen und schließlich untersuchen, wie mit diesen Mitteln die Reservierung von Ressourcen erreicht wird.

RSVP / RTPUNIVERSITÄT DORTMUND

Motivation: Wozu RSVP?

- „Bandbreite wird fast unbegrenzt sein“
 - aber Überlastung weiterhin möglich
- „Einfache Prioritäten reichen aus“
 - aber bei vielen Strömen nichts gewonnen
- „Anwendungen sollten flexibel sein“
 - aber kritische Anwendungen existieren

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol4/27

Auf den ersten Blick scheint die Einführung eines eigenen Protokolls zur Reservierung von Bandbreiten nicht notwendig zu sein - folgende Argumente werden immer wieder ins Feld geführt [RFC 1633]:

„Die künftig nutzbare Bandbreite wird fast unbegrenzt sein.“

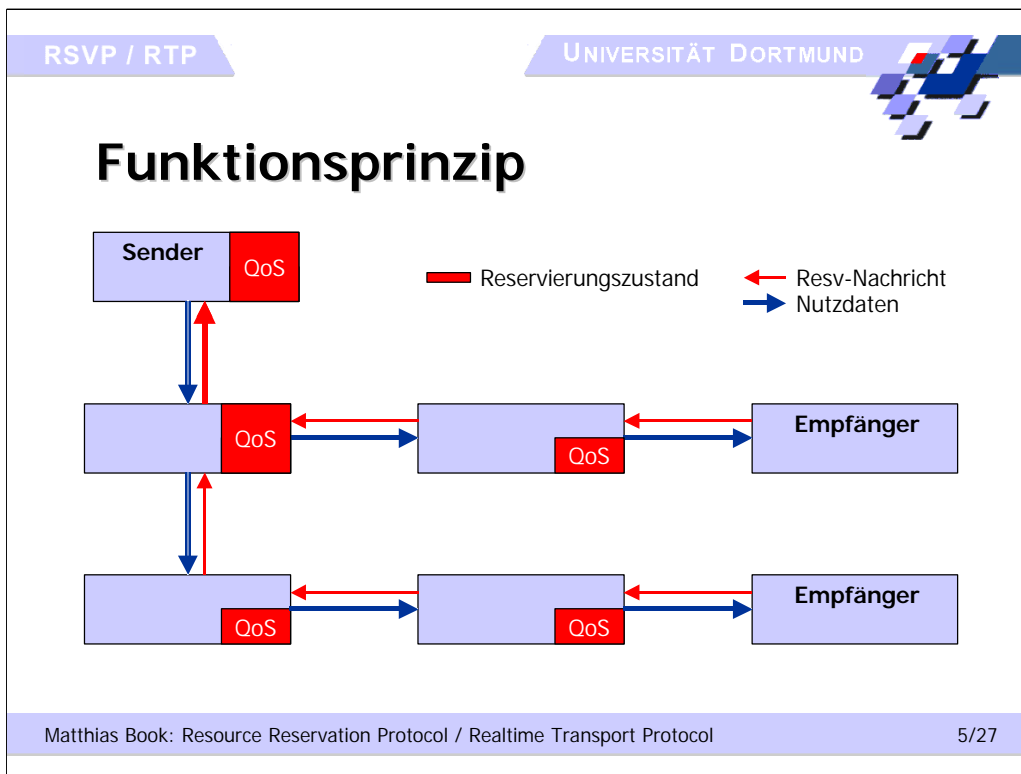
Die Bandbreite moderner Medien wie der Glasfaser ist zwar so groß, dass der Eindruck entstehen kann, sie reiche für alle Anwendungen aller Nutzer aus. Auch in Zukunft werden jedoch kaum alle beliebigen Endpunkte eines Netzes durchgängig über Hochleistungsmedien verbunden sein; an „Flaschenhälsen“ sind Staus dann vorprogrammiert.

„Einfache Prioritätsvergabe reicht aus.“

Die Einstufung von Echtzeitverkehr in eine Prioritätsstufe, die bevorzugt behandelt wird, kann nur funktionieren, wenn nicht zu viele Echtzeit-Datenströme die höhere Priorität beanspruchen. Ab einem gewissen Punkt sinkt sonst die Bandbreite, die jedem Strom zur Verfügung steht.

„Anwendungen sollten flexibel mit wechselnder Bandbreite umgehen.“

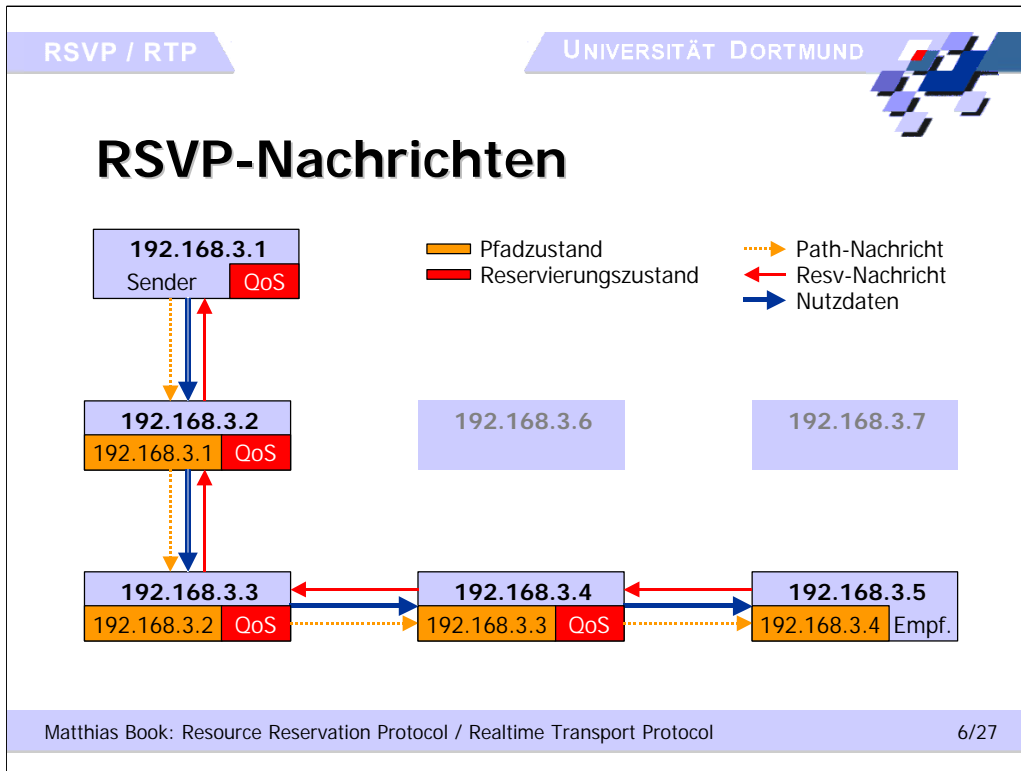
Hinreichend flexibel ausgelegte Anwendungen können kurzzeitige Aussetzer oder Bandbreitenschmälerungen zwar technisch überbrücken, ab einem gewissen Punkt ist sinnvolles Arbeiten für den Benutzers jedoch nicht mehr möglich (z.B. bei Videokonferenzen). Zudem gibt es Anwendungen, bei denen Aussetzer im Echtzeitdatenstrom schwerwiegende Konsequenzen haben können (z.B. Roboter-Telemetrie).



Zur Vorbereitung einer Echtzeit-Übertragung fordert der Empfänger für eine bestimmte Verbindung die erforderliche Dienstqualität (Quality of Service, QoS) an. Router leiten diese Anforderung an alle Netzknoten auf dem Pfad zum Sender weiter, die daraufhin entsprechende Ressourcen reservieren. RSVP ist **empfänger-orientiert** und kennt nur **Simplex-Datenströme** (d.h. Ströme in eine Richtung). Durch den gerade beschriebenen Vorgang werden daher nur Ressourcen für den Datenstrom vom Sender zum Empfänger reserviert. Sollen auch Ressourcen für den Rückkanal reserviert werden, muss der Sender diese selbst anfordern.

In einer **Multicast**-Umgebung werden die Anforderungen nur bis zum ersten Knoten weitergeleitet, in dem auch die Anforderungen anderer Empfänger eintreffen. Dort werden die Anforderungen kombiniert und eine entsprechende neue Anforderung weitergeschickt. Dadurch wird erreicht, dass der von Reservierungen erzeugte Overhead logarithmisch und nicht linear mit der Zahl der Empfänger steigt.

Da die Zahl der Teilnehmer in einer großen Multicast-Umgebung sehr wahrscheinlich variabel ist und sich Routing-Pfade verändern können, wird der Reservierungszustand in den einzelnen Knoten dynamisch gespeichert (**soft state**): RSVP sendet periodische Refresh-Nachrichten an die einzelnen Knoten. Bleiben diese an einem bestimmten Knoten aus, so wird die Reservierung dort automatisch gelöscht.

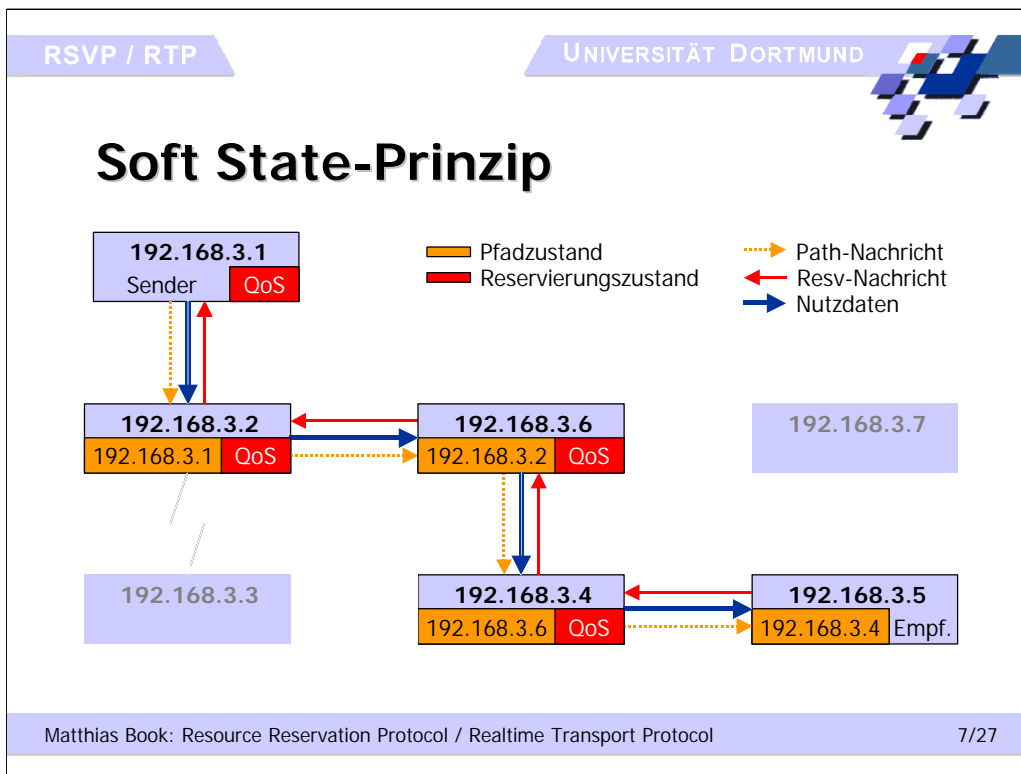


Als Kontrollprotokoll funktioniert RSVP durch den Austausch von Nachrichten zwischen beteiligten Netzknoten. Die beiden wichtigsten Nachrichten sind **Path** und **Resv** (Reservation Request).

Jeder Sender überträgt zunächst Path-Nachrichten an den oder die Empfänger, die den kommenden Datenstrom beschreiben. Diese Nachrichten sind genauso adressiert wie die Datenpakete (als Absender- und Empfängeradresse sind die Endpunkte der Übertragung gesetzt) und werden daher genauso geroutet. In jedem Knoten, den sie passieren, wird ein **Pfadzustand** gesetzt, der unter anderem die IP-Adresse des vorhergehenden Knotens enthält.

Erreicht die Path-Nachricht einen Empfänger, so sendet dieser eine Resv-Nachricht zurück, in der die gewünschte Ressourcen-Reservierung definiert ist. Um sicherzustellen, dass Resv-Nachrichten auf dem gleichen Weg zurücklaufen, den später die Datenpakete nehmen, werden Resv-Nachrichten Knoten für Knoten auf dem Weg versandt, den auch die Path-Nachrichten genommen haben (mit Hilfe der im Pfadzustand gespeicherten Adresse des jeweils vorhergehenden Knotens). In jedem Knoten, den die Resv-Nachricht passiert, wird ein **Reservierungszustand** gesetzt, der die für diesen Datenstrom reservierten Ressourcen beschreibt.

Nach Empfang der Resv-Nachricht beginnt der Sender schließlich, seine Daten an den oder die Empfänger zu übertragen.




Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol 7/27

Je nach Netzbedingungen kann sich das Routing der Datenpakete mit der Zeit verändern, weil z.B. ein Knoten ausfällt. In diesem Fall würden zum einen die Datenpakete über neue Knoten ohne Ressourcenreservierung geleitet und somit die Empfangsqualität leiden; zum anderen blieben in den nicht mehr genutzten Knoten dennoch Reservierungen bestehen, die andere Übertragungen unnötig einschränken.

Um dies zu vermeiden, verwendet RSVP für die Pfad- und Reservierungszustände ein **Soft State-Prinzip**: Pfad- und Reservierungszustand müssen periodisch aufgefrischt werden, um nicht zu verfallen. Der Sender überträgt daher auch nach dem Aufbau der Verbindung periodisch Path-Nachrichten. Zum einen wird so der Path-Zustand in den zugehörigen Knoten aufgefrischt, zum anderen wird bei Routing-Änderungen auch in den Knoten des neuen Pfads umgehend für eine Reservierung gesorgt - denn auch der Empfänger überträgt periodisch weiter Resv-Nachrichten, die über die gesamte Kette weitergeleitet werden und in allen besuchten Knoten die Reservierungen auffrischen bzw. neu anlegen.

Verfällt in einem Knoten der Pfad- oder Reservierungszustand, weil keine rechtzeitige Auffrischung erfolgte, so sendet er PathTear- bzw. ResvTear-Nachrichten an seine Nachfolger- bzw. Vorgängerknoten, um die reservierten Ressourcen auch dort unmittelbar freizugeben.

Natürlich können in jedem Knoten mehrere Pfad- und Reservierungszustände für verschiedene Datenströme, die über diesen Knoten laufen, gespeichert sein.

RSVP / RTPUNIVERSITÄT DORTMUND 

Resv-Nachricht: Flow Spec

- Session, Absender, Timeout, Policy, ...
- Flow Descriptor
 - Flow Spec
 - Dienstklasse
 - Reservation Spec (RSpec)
 - Rate, Slack Term, ...
 - Traffic Spec (TSpec)
 - Peak Data Rate, Maximum Packet Size, ...
 - Filter Spec

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol8/27

Betrachten wir eine Resv-Nachricht genauer: Kern jeder Reservierungsanforderung ist der sog. **Flow Descriptor**. Er besteht aus einer Flow Spec, die eine bestimmte Dienstqualität (QoS) definiert, und einer Filter Spec, die die Datenpakete definiert, denen diese QoS zugute kommen soll.

In der **Flow Spec** werden die Dienstklasse (Controlled-Load oder Guaranteed Service), die gewünschte QoS und der Datenstrom beschrieben [RFC 2210].

In der Dienstklasse **Controlled-Load Service** bekommt der Datenstrom mit Hilfe von Zugangskontrollmechanismen auch in Überlastsituationen annähernd die QoS zugeteilt, die er auch von einem unbelasteten Netzknoten erhalten würde. [RFC 2211]. **Guaranteed Service** stellt sicher, dass die Ende-zu-Ende-Verzögerung sich innerhalb definierter Grenzen bewegt und garantiert so ein festes Maß an Verzögerung und Bandbreite [RFC 2212].

Die gewünschte QoS wird in einer **Reservation Spec (RSpec)** definiert, die von RSVP jedoch nur blind übertragen wird. Für die Auswertung der RSpec-Parameter wie Rate (zu übertragende Bytes/Sekunde) und Slack Term (Differenz zwischen gewünschter und angebotener Verzögerung) ist der Netzknoten verantwortlich. Gleiches gilt für die **Traffic Spec (TSpec)**, die den Datenstrom mit Hilfe von Parametern wie Peak Data Rate und Maximum Packet Size beschreibt.

RSVP / RTPUNIVERSITÄT DORTMUND 

Resv-Nachricht: Filter Spec

- Session, Absender, Timeout, Policy, ...
- Flow Descriptor
 - Flow Spec
 - Filter Spec
 - IPv4
 - 4-Byte-Absenderadresse, Quellport
 - IPv6
 - 16-Byte-Absenderadresse, Quellport *oder*
 - 16-Byte-Absenderadresse, Flow Label

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol9/27

Die **Filter Spec** ist der Teil der Reservierungsanforderung, der definiert, welchen Datenpaketen die gewünschte QoS zugute kommen soll.

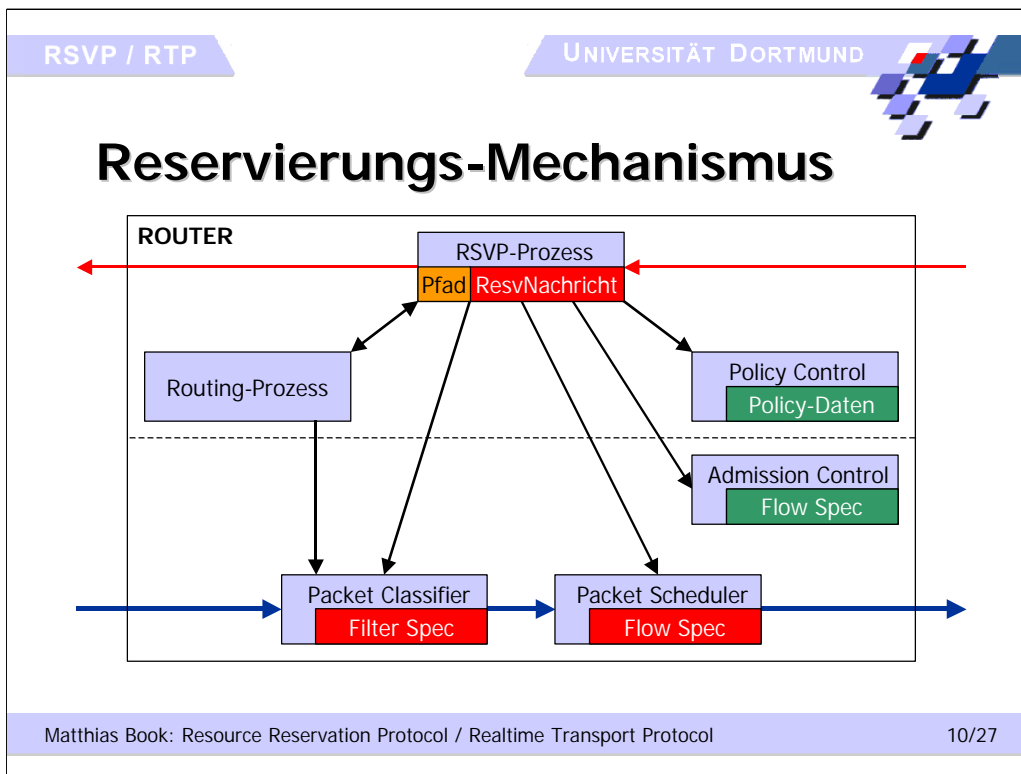
Datenpakete werden über die IP-Adresse und (optional) den Quellport ihres Senders gefiltert. Anstatt des Quellports kann bei IPv6 auch der Flow Label verwendet werden, der in jedem Paketheader enthalten ist und einen bestimmten Datenstrom vom Sender zum Empfänger eindeutig identifiziert.

Je nachdem, ob IPv4 oder IPv6 als zugrundeliegendes Protokoll verwendet wird, sind also folgende Filter Spec-Formate möglich:

- IPv4-Absenderadresse (4 Bytes) und Quellport (2 Bytes)
- IPv6-Absenderadresse (16 Bytes) und Quellport (2 Bytes)
- IPv6-Absenderadresse (16 Bytes) und Flow Label (3 Bytes)

Auch wenn in der aktuellen RSVP-Spezifikation nur diese Parameter als Filter vorgesehen werden, können prinzipiell beliebige Parameter des Übertragungs- bzw. übertragenen Protokolls verwendet werden.

Path-Nachrichten enthalten ähnliche Informationen zur Beschreibung des zu erwartenden Datenstroms: Auch sie verwenden eine Filter Spec, um die Pakete des Datenstroms zu identifizieren, sowie eine TSpec, um die Charakteristika des Datenstroms anzugeben.




Betrachten wir nun genauer, wie die Reservierung in jedem Knoten, der eine Resv-Nachricht erhält, abläuft.

Empfängt ein Knoten eine Reservierungsanforderung, so übergibt der dort laufende RSVP-Prozess die Nachricht zunächst an die **Admission Control**, die überprüft, ob der Knoten über genügend Ressourcen verfügt, um die gewünschte QoS bereitzustellen, und an die **Policy Control**, die überprüft, ob der Benutzer das Recht hat, diese Reservierung durchzuführen. Wie schon die QoS-Parameter werden auch die Policy-Parameter von RSVP nur blind übertragen, jedoch nicht interpretiert [RFC 2750].

Verlaufen beide Checks positiv, so werden **Packet Classifier** und **Packet Scheduler** konfiguriert: Der Packet Classifier ordnet jedes empfangene Datenpaket einer QoS-Klasse zu. Er wird entsprechend der Filter Spec eingestellt. Der Packet Scheduler hingegen bestimmt die Reihenfolge, in der die Datenpakete weitergeleitet werden und ist somit für die Realisierung der QoS verantwortlich. Er wird so eingestellt, dass er die in der Flow Spec definierte QoS liefert (Diese Einstellungen gelten jeweils für *einen* Datenstrom, von denen jeder Knoten natürlich mehrere handhaben kann).

Anschließend wird eine neue Resv-Nachricht an den nächsten Upstream-Knoten auf dem Pfad zum Sender weitergeleitet. Diese Reservierungsanforderung kann sich von der empfangenen unterscheiden, da in einem Knoten die Anforderungen mehrerer Empfänger zusammenlaufen können. In diesem Fall wird nur eine Anforderung weitergeleitet, die eine Kombination der Einzelanforderungen darstellt.

RSVP / RTPUNIVERSITÄT DORTMUND 

Fazit: Was macht RSVP?

- empfängerorientierte
- Reservierung von Ressourcen
- für Simplex-Datenströme
- in Unicast- und Multicast-Anwendungen
- mit flexibler Anpassung ans Routing

- transparent für RSVP-unfähige Knoten

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol11/27

Fassen wir die Funktion des Resource Reservation Protocol noch einmal zusammen:

Die Verantwortung für die Reservierung von Ressourcen liegt beim Empfänger einer Übertragung. Der Sender unterstützt ihn dabei durch die Beschreibung des zu erwartenden Datenstroms sowie durch die Markierung des Pfads, den die Datenpakete nehmen werden. Da RSVP nur Simplex-Ströme kennt, sind Hin- und Rückkanal (soweit vorhanden) getrennt zu betrachten.

Ressourcen werden in jedem Knoten reserviert, der auf dem Weg zwischen Empfänger und Sender liegt. In Multicast-Umgebungen wird der Overhead dabei durch die Kombination von Reservierungen möglichst gering gehalten.

Die Ressourcen-Reservierung passt sich durch die Verwendung von Soft States flexibel an geänderte Routing-Bedingungen an.

Für Knoten im Übertragungspfad, die RSVP nicht unterstützen, bleibt das Protokoll übrigens transparent, jedoch können in diesen Knoten natürlich keine Ressourcen reserviert werden. Abhängig von der Kapazität des Knotens wird die Dienstqualität der gesamten Verbindung dadurch mehr oder weniger stark beeinträchtigt.

RSVP / RTPUNIVERSITÄT DORTMUND 

Fazit: Was macht RSVP nicht?

- Transport von Anwendungsdaten
- Treffen von Routing-Entscheidungen
 - Verlass auf entsprechende Protokolle

- Realisierung einer Quality of Service
- Policy- und Admission Control
 - nur Transport entsprechender Daten

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol12/27


RSVP setzt auf IPv4 oder IPv6 auf und belegt im Protokollstack den Platz des Transportprotokolls. RSVP überträgt jedoch selbst keine **Anwendungsdaten**, sondern stellt lediglich ein Kontrollprotokoll dar.

Ebenso ist RSVP kein **Routing-Protokoll**, sondern arbeitet mit Unicast- und Multicast-Routing-Protokollen zusammen: Während das Routing-Protokoll bestimmt, wohin Pakete weitergeleitet werden, kümmert RSVP sich um die Ressourcen, die ihnen auf ihrem Pfad zur Verfügung stehen.

Dazu transportiert RSVP **Quality-of-Service**-Parameter, ohne diese jedoch selbst zu kennen. Die Interpretation und Durchsetzung der QoS-Parameter bleibt Aufgabe der Netzknoten. [RFC 2210]

Gleiches gilt für die Zugangskontrolle: Da RSVP einigen Anwendungen mehr Ressourcen zuordnet als anderen (auf deren Kosten!), muss ein Mechanismus existieren, der regelt, welche Anwendungen welche Ressourcen reservieren dürfen, und der sicherstellt, dass eine Ressource nur von der Anwendung genutzt wird, die sie auch reserviert hat. RSVP überträgt die **Admission-** und **Policy-Control**-Daten zwar, interpretiert sie jedoch nicht [RFC 2750].

RSVP / RTP


UNIVERSITÄT DORTMUND 

YOU ARE HERE

- Einleitung
- RSVP
- RTP
 - Überblick
 - Funktionsszenarien und Elemente
 - RTP Control Protocol (RTCP)

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol 13/27

Wie wir gerade gesehen haben, ist RSVP nicht für den Transport von Nutzdaten zuständig. Dies leistet das Realtime Transport Protocol (RTP).

RSVP / RTPUNIVERSITÄT DORTMUND 

Motivation: Wozu RTP?

- Resequencing
- Synchronisation
- Multicast-Übertragung
- Anpassung an variable QoS
- Übertragung von Meta-Daten


nicht vom Nutzdatenprotokoll zu leisten!

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol14/27

Um Multimedia-Anwendungen wie Videokonferenzen über das Netz betreiben zu können, genügt es in der Regel nicht, eine Netzverbindung aufzubauen, ggf. benötigte Ressourcen zu reservieren und die Nutzdaten ohne weitere Maßnahmen darüber zu verschicken.

Zum Transport muss der Datenstrom in Pakete aufgeteilt werden, die getrennt verschickt werden - je nach Anwendung an einen oder mehrere Empfänger (**Multicast**). An den Empfängern müssen diese Pakete in der richtigen Reihenfolge zusammengesetzt (**Resequencing**) und unter Umständen mit anderen eintreffenden Strömen **synchronisiert** werden. Zudem müssen Empfänger sich während der Übertragung an **variable Dienstqualitäten (QoS)** anpassen können und benötigen Informationen über die an der Übertragung teilnehmenden Stationen (**Meta-Daten**).

Da das Nutzdatenprotokoll dies nicht selbst leisten kann, werden die Nutzdaten in das Realtime Transport Protocol (RTP) „eingepackt“.

RSVP / RTPUNIVERSITÄT DORTMUND 

Aufgaben von RTP

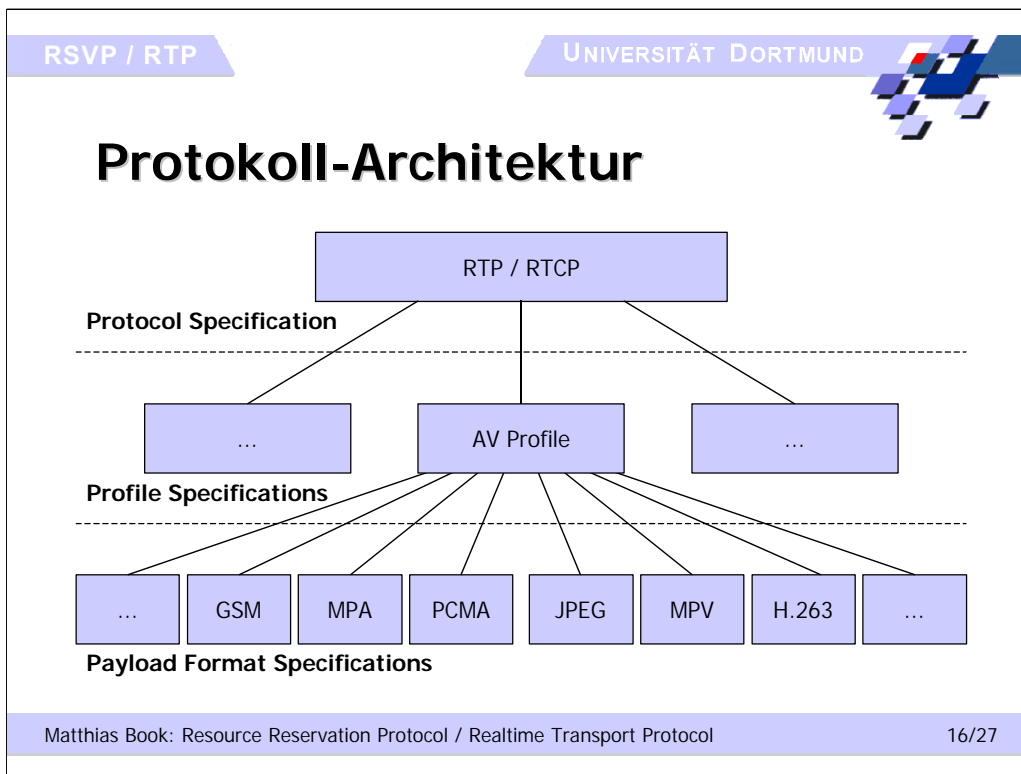
- RTP
 - Nutzdaten-Übertragung/Kennzeichnung
 - Quellen-Identifizierung
 - Sequenznummern/Zeitstempel-Vergabe
- RTCP
 - QoS-Kontrolle
 - Metadaten-Übertragung
- keine Transport/QoS-Garantien

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol15/27

RTP stellt Dienste zur Verfügung, um Echtzeitdaten zwischen den Endpunkten einer Unicast- oder Multicast-Umgebung zu übertragen. Zu diesen Diensten gehören die Kennzeichnung der übertragenen Nutzdaten und ihrer Quellen, die Vergabe von Sequenznummern und Zeitstempeln an Datenpakete, die Kontrolle der zur Verfügung stehenden Dienstqualität (QoS) sowie die Übertragung von Informationen über die Teilnehmer.


Die letzten beiden Dienste werden dabei nicht vom Datentransferprotokoll RTP selbst, sondern von dem zugehörigen Kontrollprotokoll **RTCP (RTP Control Protocol)** bereitgestellt.

Dabei ist zu beachten, dass RTP nicht für die Durchsetzung einer bestimmten Quality of Service verantwortlich ist und auch nicht garantiert, dass Pakete in der richtigen Reihenfolge, synchron oder überhaupt am Empfänger eintreffen. Ersteres kann z.B. durch Einsatz von RSVP erreicht werden, letzteres muss die Anwendung anhand der von RTP bereitgestellten Daten (Sequenznummern und Zeitstempel) selbst handhaben.



Um ein Protokoll erweiterbar zu halten, wird es üblicherweise sehr allgemein spezifiziert, enthält Möglichkeiten zur Definition optionaler Felder oder ähnliche Mechanismen. Diese Flexibilität erfordert jedoch aufwändiges Parsen der Protokollheader - ein Overhead, der bei Echtzeitdaten unerwünscht ist.

Die Architektur des RTP-Protokolls unterscheidet sich darum von der anderer Protokolle: Die Veränderung oder Erweiterung der RTP-Header zur Anpassung an bestimmte Anwendungsklassen ist ausdrücklich erlaubt. Solche Änderungen des RTP-Kerns werden in **Profile Specifications** definiert, die für eine bestimmte Klasse von Anwendungen gelten (z.B. Audio- und Videokonferenzen [RFC 1890]). Eine Profile Specification bildet zugleich das Bindeglied zwischen dem RTP-Kern und den **Payload Format Specifications**, die definieren, wie konkrete Nutzdatenformate (z.B. H.263-Videostreams [RFC 2190]) zum Transport in RTP „verpackt“ werden.

RSVP / RTP UNIVERSITÄT DORTMUND 

Beispiel: Kapselung von H.263

<ul style="list-style-type: none">■ RTP-Header<ul style="list-style-type: none">■ Flags, Counter■ Nutzdatentyp=H.263■ Sequenznummer■ Zeitstempel■ eigene Datenquellen-ID■ beitragende Datenquellen■ H.263-Nutzdatenheader<ul style="list-style-type: none">■ PB-Frames Mode Option■ Start/End Bit Position■ Source Format	<ul style="list-style-type: none">■ Picture Coding Type■ Unrestricted Motion Vector Option■ Syntax-Based Arithmetic Coding Option■ Advanced Prediction■ Differential Quantization Parameter■ Temporal Reference B-Frame/P-Frame■ H.263-Bitstrom<ul style="list-style-type: none">■ ...
---	--

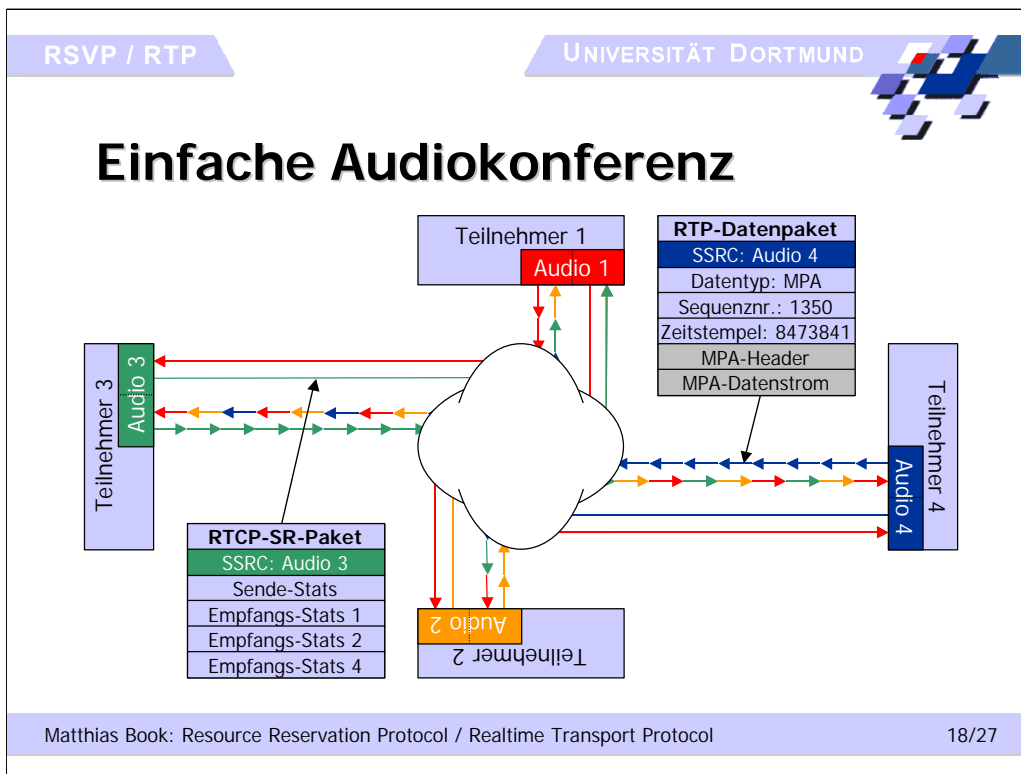
Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol 17/27

Als Beispiel betrachten wir, wie H.263-Videodaten in RTP-Paketen gekapselt werden:

Nach dem RTP-Header mit den bereits besprochenen Feldern folgt ein Header, der die übertragenen Nutzdaten (hier den H.263-Videostrom) beschreibt. Die Felder dieses Headers werden in der entsprechenden Payload Format Specification [RFC 2190] definiert.

Der Videostrom selbst folgen dann nach dem H.263-Nutzdatenheader. Auch sein Format ist in der Payload Format Specification beschrieben. Im Fall von H.263 kann der Ausgabestrom des Encoders z.B. direkt in Pakete aufgeteilt werden. Für jedes Einzelbild wird der H.263-Bitstrom selbst inclusive seines Bildheaders etc. im RTP-Nutzdatenbereich gespeichert.

Betrachten wir nun die Funktionsweise von RTP anhand verschiedener Anwendungsfälle. Ausgehend von einer einfachen Konferenz steigern wir dabei die Komplexität der Szenarien.

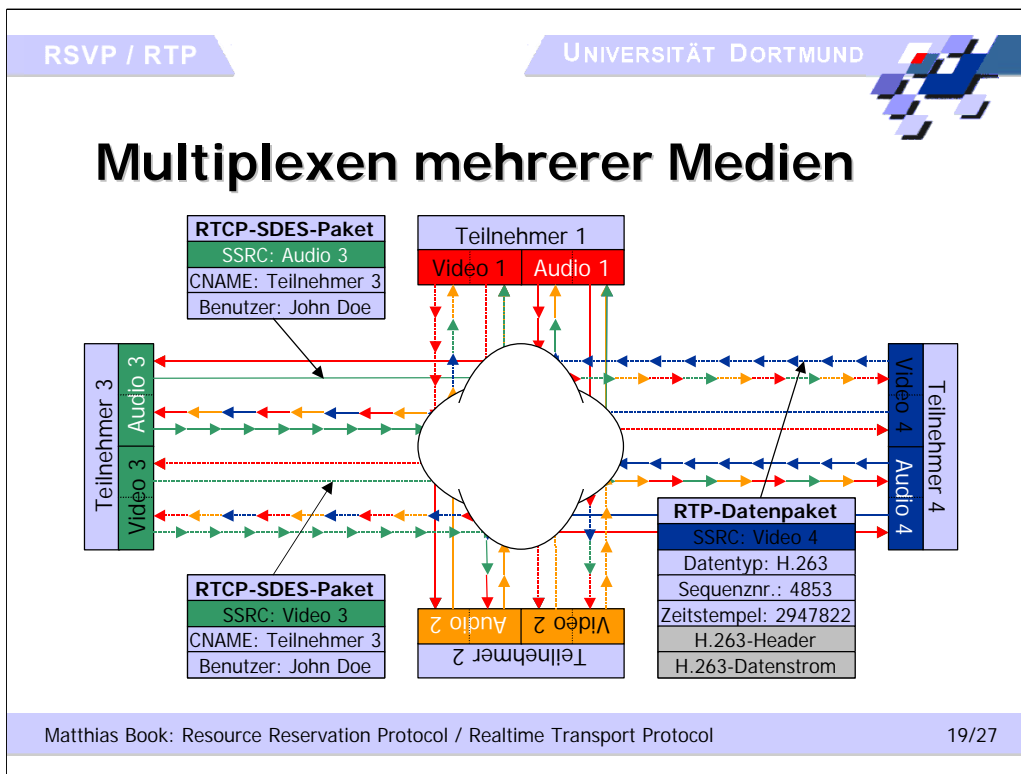


Um eine Audiokonferenz aufzubauen, müssen die Teilnehmer sich zunächst auf eine Multicast-Adresse und zwei Port-Nummern für die RTP- und RTCP-Pakete einigen.

Während der Konferenz verpackt jeder Teilnehmer den von ihm erzeugten Audio-Datenstrom in RTP-Pakete fester Größe. Die RTP-Pakete werden an die Multicast-Adresse geschickt und vom Netz an alle Teilnehmer weitergeleitet. Anhand der Zusatzinformationen, die jedes RTP-Paket neben den Nutzdaten enthält, bringt der Empfänger die Pakete in die richtige Reihenfolge, synchronisiert sie, identifiziert das gekapselte Datenformat und baut daraus die Nutzdatenströme für die Wiedergabe zusammen. Jede Datenquelle wird dazu im RTP-Paket über eine eindeutige Kennung (**Synchronization Source, SSRC**) identifiziert.

Da während der Konferenz die für bestimmte Teilnehmer zur Verfügung stehende Bandbreite schwanken kann, verschickt jede Station periodisch RTCP-Pakete mit Statistiken über ihre Empfangsqualität.

Jeder Teilnehmer sendet außerdem in zeitlichen Abständen Meta-Informationen über sich (eindeutige Teilnehmer-Kennung, Realname, e-Mail-Adresse, Standort, Anwendungsprogramm etc.). Dies dient zum einen der Verknüpfung von Datenquellen und Teilnehmern, zum anderen erkennen auf diese Weise alle Teilnehmer, wenn ein neuer Teilnehmer der Konferenz beitrifft oder ein anderer sie verläßt.

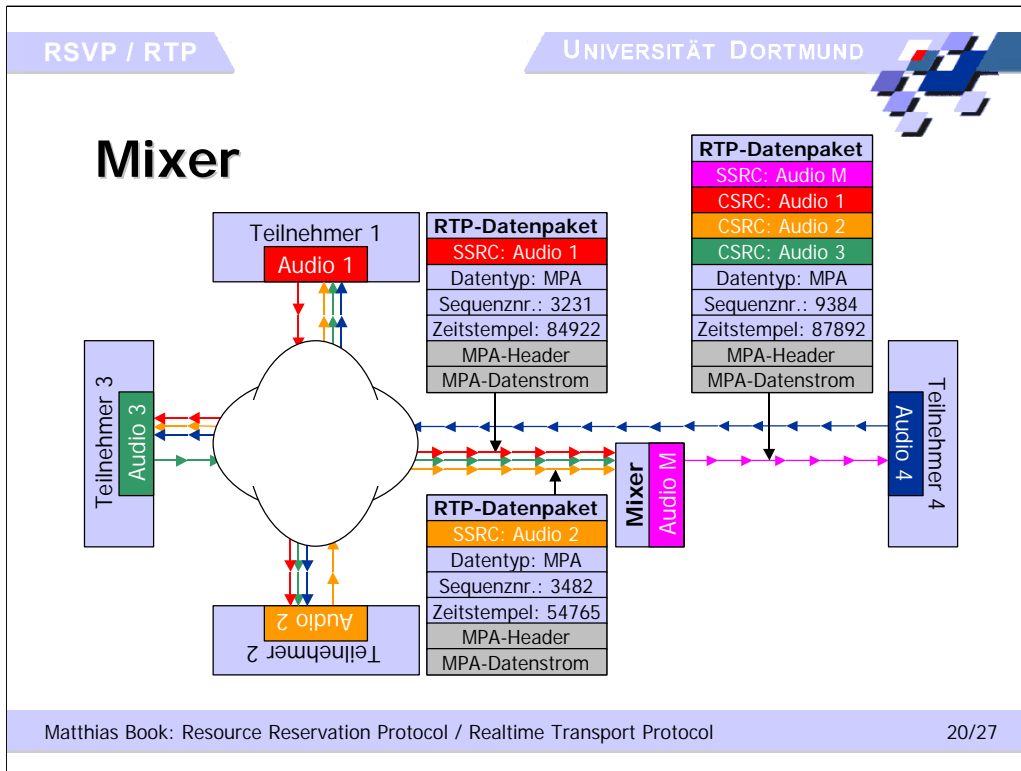


Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol 19/27

Wenn jeder Teilnehmer Datenströme verschiedener Medien generiert (in einer Videokonferenz z.B. Audio- und Videosignal), so werden diese in **getrennten** RTP-Sessions übertragen. Die Ströme werden nicht über ein Feld im RTP-Header wie „Nutzdatentyp“ oder „Synchronisation Source“ gemultiplext, sondern über ihre Zieladresse (IP-Adresse und Port). Für den zweiten Strom wird folglich ein neues Port-Paar (einer für RTP und einer für RTCP) benötigt.

Dies hat zum einen technische Gründe - die Implementierung des Protokolls würde aufwändiger, wenn es einen eigenen Multiplex-Mechanismus bereitstellen müsste. Zum anderen ermöglicht es die Übertragung in getrennten Sessions, beim Transport verschiedene Netzwerkpfade und Ressourcen zu verwenden sowie beim Teilnehmer die Verarbeitung in getrennten Prozessen zu implementieren. Zudem kann ein Teilnehmer, der nur über geringe Bandbreite verfügt, sich entscheiden, vollständig auf ein Medium zu verzichten und z.B. nur Audio übertragen.

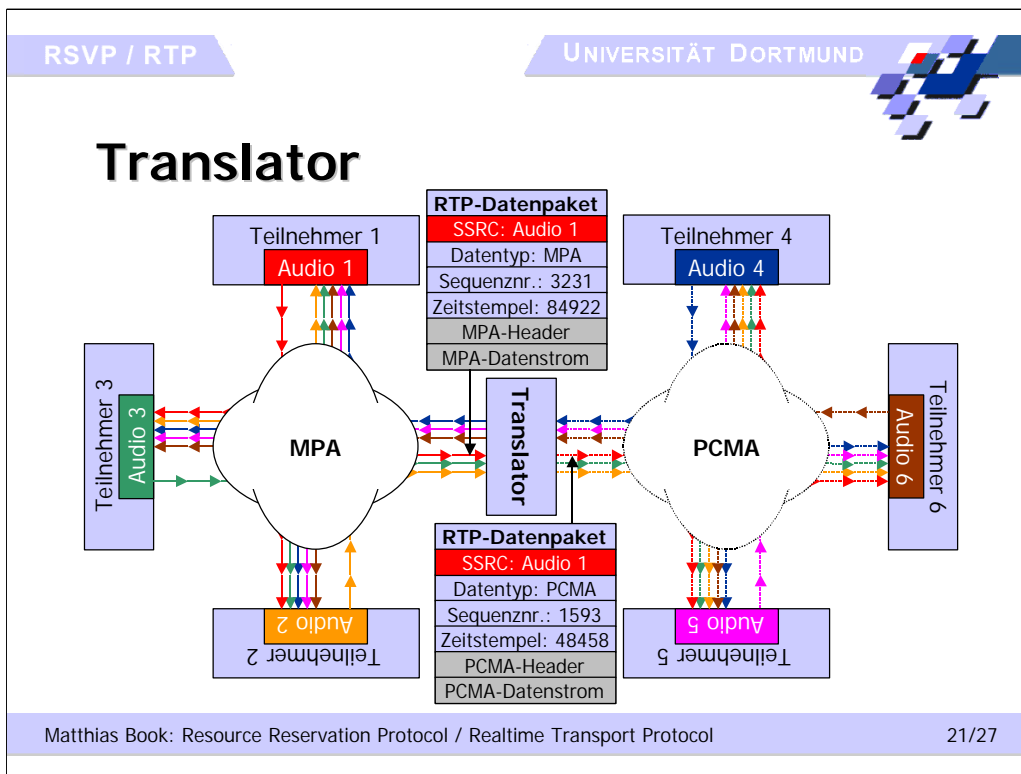
Obwohl in unserem Beispiel Video- und Audiosignal eines Teilnehmers in getrennten Sessions übertragen werden, müssen sie für die Wiedergabe natürlich verknüpft werden. Die Zuordnung zu einem bestimmten Teilnehmer geschieht über die eindeutige Teilnehmerkennung, die regelmäßig via RTCP übertragen wird; die Synchronisation erfolgt über die Zeitstempel in jedem RTP-Paket.



In einigen Fällen kann es sinnvoll sein, Datenströme von mehreren Teilnehmern zu kombinieren und als einzelnen Datentrom weiterzuleiten. Diese Aufgabe wird von einem sog. **Mixer** übernommen.

Ist ein Teilnehmer einer Audiokonferenz z.B. mit geringerer Bandbreite angebunden, so kann ein Mixer an den Anfang des betreffenden Netzabschnitts gesetzt werden. Er kombiniert die bei ihm eintreffenden Ströme, indem er die Audiosignale decodiert, zusammenmischt und das Ergebnis mit einem stärker komprimierenden Verfahren wieder codiert, um es als neuen Strom weiterzuleiten. Der Empfänger des neuen Stroms kann entweder ein einzelner Teilnehmer oder eine weitere Multicast-Gruppe sein.


Da der Mixer somit selbst zur Datenquelle wird, trägt er sich selbst als Synchronisation Source im RTP-Header jedes Pakets ein. Die ursprünglichen Pakete gehen verloren, um den Empfängern dennoch mitzuteilen, von welchen Sendern die zusammengemischten Daten stammen, ergänzt er jedes Paket um eine Liste von **Contributing Sources (CSRC)**.



Ein weiteres Element, das in RTP-Netzen eingesetzt werden kann, ist der Translator: Er kann RTP-Pakete auf verschiedene Arten verändern, bewahrt dabei jedoch im Unterschied zum Mixer ihre Identität.

Translatoren können z.B. das Nutzdatenformat von Strömen paketweise ändern, ohne sie zusammenzumischen, um Verbindungen mit geringerer Bandbreite oder Teilnehmer mit anderer Technologie zu bedienen.

Ein anderes Anwendungsgebiet ist die Überbrückung von Firewalls: Lässt eine Firewall z.B. keine Multicast-Pakete durch, so kann ein Translator auf der Außenseite die eingehenden Pakete per Unicast an einen Translator auf der Innenseite übertragen, der sie per Multicast an die inneren Teilnehmer weiterleitet.

RSVP / RTPUNIVERSITÄT DORTMUND


RTP Control Protocol (RTCP)

- Feedback über QoS der Verbindungen
 - zur Anpassung der Datenrate nutzbar
 - hilfreich bei der Fehler-Lokalisierung
- Teilnehmer-Identifikation (CNAME)
 - zur Verknüpfung von Datenströmen
- Meta-Informationen über Teilnehmer
- Einstellung der RTCP-Datenrate

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol22/27

In den vorhergehenden Anwendungsfällen wurde bereits mehrfach zu RTP gehörige Kontrollprotokoll RTCP angesprochen, das wir nun näher betrachten wollen. RTCP hat vier Aufgaben:

1. Hauptzweck ist es, den Teilnehmern laufend aktuelle Informationen über die Qualität der Datenverteilung im Netzwerk zu liefern. Dieses Feedback ermöglicht den Teilnehmern, den von ihnen generierten Datenstrom an die Netzbedingungen anzupassen (z.B. durch Reduktion der Datenrate bei geringer QoS) und Fehler einzugrenzen. RTCP überträgt dazu periodisch sog. **Sender Reports (SR)** und **Receiver Reports (RR)**.
2. Die zweite wichtige Aufgabe ist es, für jeden Teilnehmer eine konstante, eindeutige Kennzeichnung (**Canonical Name, CNAME**) zu übertragen. Diese Kennzeichnung wird vom Empfänger dazu verwendet, mehrere Datenströme eines Teilnehmers (z.B. Audio und Video), die in getrennten Sessions übertragen werden, wieder zusammenzuführen.
3. Diese Kontrolldaten müssen zwar periodisch von allen Teilnehmer übertragen werden, dürfen bei großen Teilnehmerzahlen jedoch nicht die Bandbreite für den Nutzdatentransport einschränken. Da jeder Teilnehmer jedoch alle RTCP-Pakete erhält, kennt er die Gesamtzahl der Teilnehmer und kann daraus die Rate berechnen, mit der Kontrollpakete gesendet werden dürfen.
4. Optional kann jeder Teilnehmer via RTCP grundlegende Informationen über sich selbst versenden (**Source Description, SDES**), die z.B. in der Benutzungsoberfläche der Anwendung angezeigt werden können.

RSVP / RTPUNIVERSITÄT DORTMUND 

Sender/Receiver Report

<ul style="list-style-type: none">■ Header<ul style="list-style-type: none">■ Flags, Paketlänge■ # Report Blocks■ SSRC des Senders■ Sender Info<ul style="list-style-type: none">■ realer Zeitstempel■ RTP-Zeitstempel■ # gesendete Pakete■ # gesendete Bytes	<ul style="list-style-type: none">■ Report Block<ul style="list-style-type: none">■ SSRC der Quelle■ % verlorene Pakete■ # verlorene Pakete■ höchste Sequenznr.■ Empfangs-Jitter■ Zeit des letzten SR■ Verzögerung seitdem■ Profile-spez. Daten
---	--

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol23/27


Der Sender bzw. Receiver Report (SR bzw. RR) wird periodisch von jedem Teilnehmer gesendet, um den anderen Teilnehmern Informationen über die Qualität der Datenübertragung mitzuteilen.

Neben den technischen Informationen wie einigen Flags und Zählern enthält der **Header** die eindeutige Kennung der Datenquelle (Synchronization Source, SSRC), die diesen Report erzeugt hat.

Hat diese Quelle seit dem letzten Versand eines Sender bzw. Receiver Report auch Nutzdaten versandt, so folgt nach dem Header der **Sender Info-Block** (in diesem Fall heißt das Paket „Sender Report“; fehlt der Sender Info-Block dagegen, so heißt es „Receiver Report“). Er enthält zum einen Zeitstempel mit der realen Uhrzeit, an dem der Report versandt wurde, und zum anderen einen äquivalenten Zeitstempel in der von RTP für diesen Strom verwendeten Einheit. Außerdem wird die Anzahl der seit Übertragungsbeginn gesendeten Pakete bzw. Bytes angegeben.

Für jede externe Datenquelle, von der dieser Teilnehmer seit dem letzten Versand eines Sender bzw. Receiver Reports Nutzdaten erhalten hat, folgt anschließend ein Report Block. Er enthält zu jeder (durch ihre SSRC identifizierte) Datenquelle den Prozentsatz und die Anzahl der nicht empfangenen Nutzdatenpakete, die höchste bereits empfangene Sequenznummer, die Varianz des Zeitintervalls zwischen dem Eintreffen zweier Datenpakete, den Zeitstempel des letzten empfangenen Sender Reports sowie das seitdem verstrichene Zeitintervall.

Weitere anwendungsspezifische Felder können in Profile Specifications definiert werden.

RSVP / RTPUNIVERSITÄT DORTMUND 


Source Description

- Header
 - Flags, Paketlänge
 - # Quellen
- Quelle 1
 - SSRC (eindeutige Quellenkennung)
 - CNAME (eindeutige Teilnehmerkennung)
 - Benutzername
- Quelle 2
 - e-Mail-Adresse
 - Telefonnummer
 - geogr. Standort
 - Anwendungsname
 - Kurznachricht/Status
 - spez. Erweiterungen
- ...

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol24/27

Jeder Teilnehmer sendet periodisch **Source Description (SDES)**-Pakete, die Zusatzinformationen über ihn enthalten können. Um Overhead einzusparen, können mehrere Datenquellen (Synchronization Sources, SSRCs) in einem Paket beschrieben werden.

Die eindeutige Teilnehmerkennung (**Canonical Name, CNAME**) muss für jede Quelle angegeben werden, da sie wichtig für die Verknüpfung verschiedener Datenströme eines Teilnehmers ist. Die übrigen Elemente sind jedoch optional und sollten, um Bandbreite zu sparen, lediglich in größeren zeitlichen Abständen übertragen werden. Ihre Bedeutung ist im wesentlichen selbsterklärend. Über das Kurznachricht-Element kann ein Teilnehmer kurze Mitteilungen an alle anderen verschicken, zudem steht ein Element für applikations-spezifische Erweiterungen bereit.

RSVP / RTPUNIVERSITÄT DORTMUND 

Fazit: Was macht RTP?


- RTP
 - Kapselung von Echtzeit-Datenformaten
 - zur Uni- bzw. Multicast-Übertragung
 - unterstützt durch Mixer und Translatoren
- RTCP
 - Feedback über QoS
 - Teilnehmer-Identifikation
 - Metadaten-Übertragung

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol25/27

Fassen wir die Leistungen von RTP und RTCP noch einmal zusammen:

Das **Realtime Transport Protocol (RTP)** transportiert Echtzeit-Nutzdaten in Unicast- und Multicast Umgebungen. Zur Handhabung besonderer Netzwerkeigenschaften (geringe Bandbreite, Firewalls etc.) können dabei Mixer und Translatoren zum Einsatz kommen, die mehrere Datenströme zu einem zusammenfassen (Mixer) bzw. Datenströme paketweise verändern (Translator). Die Kapselung der Nutzdaten erfolgt durch Erweiterungen bzw. Veränderungen des RTP-Protokolls, die in Profile Specifications und Payload Format Specifications definiert sind.

RTP wird durch das **RTP Control Protocol (RTCP)** ergänzt, das Informationen über die Verbindungsqualität zwischen den Teilnehmern transportiert, Teilnehmer eindeutig identifiziert und so die Zuordnung mehrerer Datenquellen zu einem Teilnehmer ermöglicht, und Metadaten über die Teilnehmer transportiert, die je nach Anwendungsgebiet interessant sein können.

RSVP / RTPUNIVERSITÄT DORTMUND 

Fazit: Was macht RTP nicht?

- Multiplexen von Datenströmen
- Durchsetzen einer bestimmten QoS
 - reservierte Bandbreite
 - feste Übertragungszeiten
- Transportgarantien
 - Pakete in korrekter Reihenfolge
 - keine Paketverluste

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol26/27

RTP ist jedoch nicht verantwortlich für das Multiplexen von Datenströmen verschiedener Quellen. Da diese in getrennten Sessions übertragen werden, muss der Empfänger anhand des Ports, auf dem die Pakete eintreffen, entscheiden, zu welchem Datenstrom sie gehören.

Desweiteren reserviert RTP keine Bandbreite oder garantiert feste Übertragungszeiten. Via RTCP wird lediglich ein Mechanismus bereitgestellt, Informationen über die zur Zeit von den Teilnehmern erfasste QoS zu verteilen, sodass die Anwendungen darauf ggf. reagieren können.

Ebenso garantiert RTP nicht, dass Pakete in korrekter Reihenfolge oder überhaupt am Empfänger ankommen, sondern stellt der Anwendung lediglich Daten zur Verfügung, mit deren Hilfe diese die korrekte Paketreihenfolge wiederherstellen bzw. verlorene Pakete erkennen kann.

RSVP / RTPUNIVERSITÄT DORTMUND 

„Noch Fragen, Kienzle?“

;-)

Matthias Book: Resource Reservation Protocol / Realtime Transport Protocol27/27

Literatur

- [RFC 1633] Integrated Services in the Internet Architecture: an Overview
 - [RFC 1889] RTP: A Transport Protocol for Real-Time Applications
 - [RFC 1890] RTP Profile for Audio and Video Conferences with Minimal Control
 - [RFC 2190] RTP Payload Format for H.263 Video Streams
 - [RFC 2205] Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification
 - [RFC 2210] The Use of RSVP with IETF Integrated Services
 - [RFC 2211] Specification of the Controlled-Load Network Element Service
 - [RFC 2212] Specification of Guaranteed Quality of Service
 - [RFC 2215] General Characterization Parameters for Integrated Service Network Elements
 - [RFC 2750] RSVP Extensions for Policy Control
- Alle RFCs sind unter www.rfc-editor.org verfügbar.*