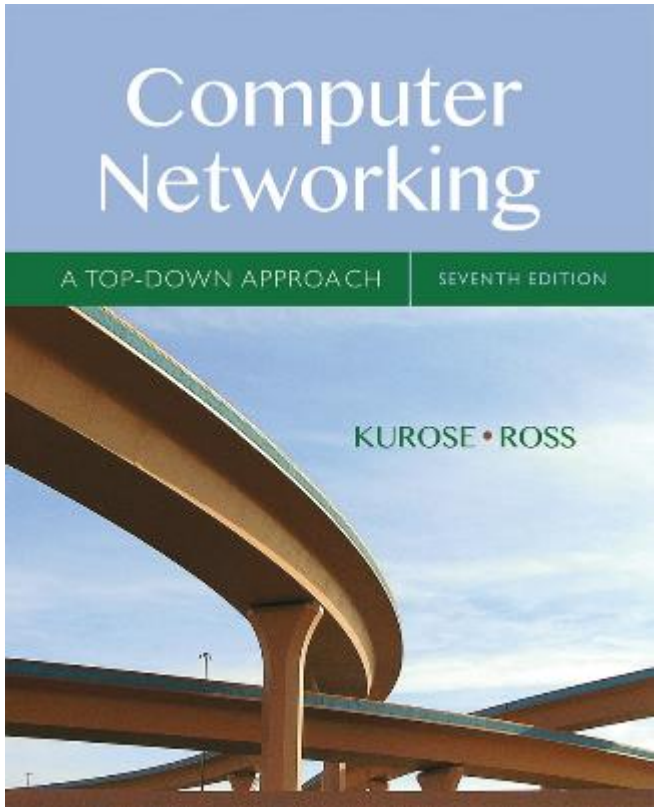
Chapter 9 Multimedia Networking



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Multimedia networking: outline

9.1 multimedia networking applications

9.2 streaming *stored* video

9.3 voice-over-IP

9.4 protocols for *real-time* conversational applications: RTP, SIP

9.5 network support for multimedia

پروتکل هایی رو معرفی می کنیم در این اسلاید که برای ریل تایم در سطح شبکه و به خصوص برای ارتباط های دو طرفه باید استفاده بشن

Real-Time Protocol (RTP)

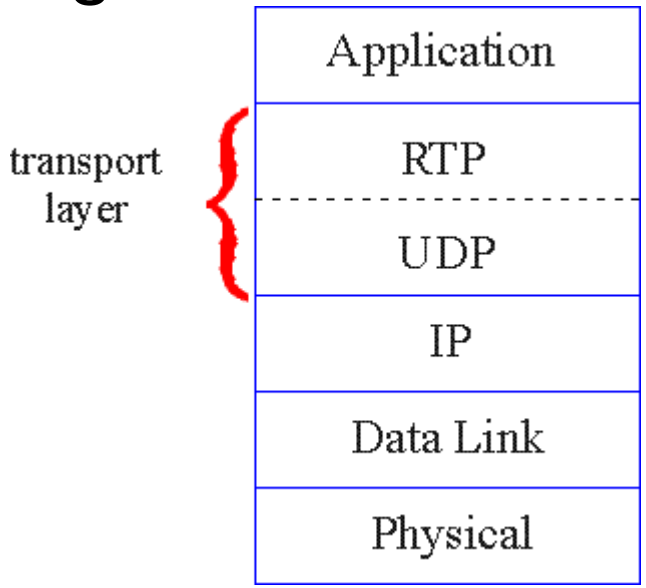
- RTP specifies packet structure for packets carrying audio, video data
- RFC 3550
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - time stamping
- RTP runs in end systems
- RTP packets encapsulated in UDP segments
- interoperability: if two VoIP applications run RTP, they may be able to work together

پروتکل RTP:
این پروتکل به طور مشخص برای اپلکیشن هایی استفاده میشه که ریل تایم هستن و تاخیر برای اون ها مهم است و نمی تونن تاخیرهای زیادی رو تحمل بکنن
یک سری اپلیکیشن هایی رو ما در لایه transport روی udp قرار میدیم چون این ها نمی تونن تاخیرهای مربوط به گم شدن بسته ها و retransmission های Tcp رو تحمل بکنن
این بسته ها ترتیب براشون مهمه و گم شدن براشون مهمه
برای تاخیر برای بسته های ویس ما لازم است که sequence number داشته باشیم روی بسته های ویس و همینطور timestamp قبلا گفتیم اینارو توی اپلیکیشن گذاشته میشه ولی ما یه پروتکلی داریم که استفاده میشه برای همچین کارهایی
همینطور مشخص کردن نوع کدینگی که استفاده میشه توی اپلیکیشن های streaming برای مالتی مدیا برای ویس مثلا کدینگ های مختلف می تونیم استفاده بکنیم و اینکه دیتای ما ویس است باید کدینگش هم مشخص بشه پهلوش --> اینا رو RTP می تونه کمک بکنه

RTP runs on top of UDP

RTP libraries provide transport-layer interface that extends UDP:

- port numbers, IP addresses
- payload type identification
- packet sequence numbering
- time-stamping



به این ترتیب RTP یک extend است برای udp و کارهایی که انتظار داشتیم انجام بشه ولی udp نمی تونه انجام بده و البته tcp اینارو انجام میده ولی با هزینه اون تاخیری که اینجا نمیخوایم درگیر اون تاخیر بشیم در نتیجه پروتکل RTP طراحی شده ک این نواقص رو برطرف بکنه در پروتکل RTP ما پورت نامبر و ip ادرس اپلیکیشن رو می تونیم مشخص بکنیم و همینطور نوع کدینگ یا payload type و sequence numbering و time stamp رو RTP می تونه برامون انجام بده
RTP به عنوان extendشن udp در لایه transport مطرح کرده این اسلاید ولی در خیلی مراجع RTP یک پروتکلی است که در لایه اپلیکیشن مطرح میشه و بین اپلیکیشن و لایه transport در صورتیکه در لایه transport از udp استفاده بکنیم برای اپلیکیشن های ریل تایم ما روی udp از RTP استفاده میکنیم

ایا RTP در مورد کیفیت سرویس اپلیکیشنی که داره از RTP استفاده میکنه کاری می تونه انجام بده؟ نه RTP در مورد کیفیت سرویس ینی درو اقع میزان لاس و تاخیر شبکه کاری انجام نمیده بلکهRTP وسیله ای است که اگر تاخیری اتفاق افتاد یا بسته ای توی شبکه گم شد ما در انتها در لایه اپلیکیشن در مقصد بتونیم کاری براش بکنیم و سعی بکنیم اینو جبران بکنیم وگرنه خود RTP نمیتونه برای کم کردن تاخیر یا لاس کاری انجام بده --> پس RTP یک پروتکلی است که بین مبدا و مقصد اجرا میشه و RTP در نودهای میانی دیده نمیشه پس سرویسی که از شبکه میگیریم همون سرویس best effort ip است که در سطح شبکه این وضعیت وجود داره و خود لایه ip در اینترنت یک لایه best effort است و هیچ تلاشی برای کم کردن تاخیر یا لاس توی لایه ip انجام نمی گیره

RTP header

payload type (7 bits): indicates type of encoding currently being used. If sender changes encoding during call, sender informs receiver via payload type field

- Payload type 0: PCM mu-law, 64 kbps
- Payload type 3: GSM, 13 kbps
- Payload type 7: LPC, 2.4 kbps
- Payload type 26: Motion JPEG
- Payload type 31: H.261
- Payload type 33: MPEG2 video

sequence # (16 bits): increment by one for each RTP packet sent

- detect packet loss, restore packet sequence

فیلدهای هدر بسته RTP:
payload type: یک کد 7 بیتی است که مشخص میکنه که payload که توسط این بسته ها منتقل میشن encodingشون چی هست --> این یک نمونه از کدهایی است که برای encoding های مختلف استفاده میشه مثلا برای PCM هست 0 ینی 0 استفاده میشه

sequence number: یک عدد 16 بیتی است و به ازای هر بسته ای که ارسال میشه یک واحد به این اضافه میشه
sequence # به طور مشخص در مقصد می تونه برای تشخیص بسته های گم شده و همینطور برای مرتب کردن بسته ها اگر ترتیبشون بهم خورده می تونه استفاده بشه

RTP header

- *timestamp field (32 bits long):* sampling instant of first byte in this RTP data packet
  - for audio, timestamp clock increments by one for each sampling period (e.g., each 125 usecs for 8 KHz sampling clock)
  - if application generates chunks of 160 encoded samples, timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.

- *SSRC field (32 bits long):* identifies source of RTP stream. Each stream in RTP session has distinct SSRC

timestamp: هر بسته ای که ایجاد میشه در مبدا در لحظه ای که اولین بایت اون سمپل میشه و کد میشه کلاک سیستم به صورت یک عدد 32 بیتی ثبت میشه و به عنوان timestamp به اون فیلد timestamp اضافه میشه و در بسته قرار میگیره و ارسال میشه به سمت مقصد
کاربردش: در مقصد با استفاده از timestamp زمان بندی پخش بسته های مثلا ویس رو می تونیم انجام بدیم
timestamp روی بسته های RTP الزاما پریودیک نیست مگر اینکه مبدا به طور مرتب فعال بشه و بدون وقفه اون ؟ انجام بگیره و دیتا ایجاد بشه وگرنه اگر فاصله اتفاق بیوفته زمان اضافه میشه و بسته بعدی timestamp اش می تونه خیلی بیشتر فاصله پیدا بکنه با timestamp بسته قبلی پس timestamp ها الزاما پریودیک نیستن
نکته: اگر فاصله داشته باشیم توی دیتامون با Sequence number تشخیص میدیم که اینجا فاصله است و timestamp رو استفاده میکنیم برای تنظیم زمان پخش بسته ای که به مقصد می رسه

SSRC: یک فیلد 32 بیتی است که مشخص میکنه که دیتایی که داره توسط این بسته های RTP میاد stream میشه سورسش چی هست
یکی از کارهایی که با RTP می تونیم انجام بدیم میکس سیگنال های مدیاهای مختلف است در مقصد مثلا تصویر از یه جایی باید و صدا از یه جایی دیگ و اینا میکس بشن در مقصد در این صورت ما توی مقصد بسته های RTP از سورس های مختلف داریم دریافت می کنیم و توی بسته های RTP این سورس ها با یک id مشخص میشن و یکی از این ها به عنوان رفرنس کلاک بقیه سورس ها استفاده میشه

Real-Time Control Protocol (RTCP)

- works in conjunction with RTP
- each participant in RTP session periodically sends RTCP control packets to all other participants

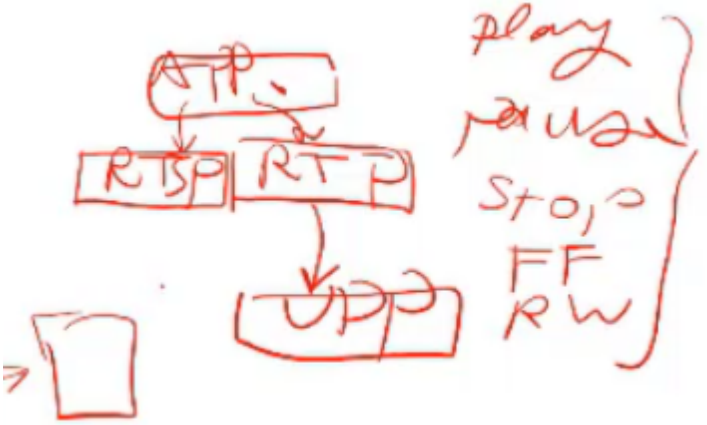
- each RTCP packet contains sender and/or receiver reports
  - report statistics useful to application: # packets sent, # packets lost, interarrival jitter
- feedback used to control performance
  - sender may modify its transmissions based on feedback

در کنار RTP و RTSP باید از پروتکل RTCP هم استفاده بکنیم
:RTCP
مثلا اگر برای اپلیکیشن های streaming در کنار RTP ما RTSP رو داریم اصولا برای هر نوع اپلیکیشنی که داره RTP رو استفاده میکنه RTCP هم باید استفاده بکنیم --> عکس:
مجددا دیتا توسط RTP منتقل میشه و روی udp قرار میگیره و بعد می ره توی شبکه و برای کارهای کنترلی (اگر کارهای کنترلی مربوط به streaming مدیا باشه از پروتکل RTSP استفاده میکنیم) و یکسری کارهای کنترلی دیگه رو RTCP انجام میده
RTCP پروتکلی است که برای کنترل ارتباط ریل تایم استفاده میشه
RTCP باز بسته های پروتکلی خودش رو داره و هدر خودش رو داره و هدرش هم فیلدهای مختلفی دارن که برای کارهای مختلف استفاده میشه --> روال کار این است که در یک ارتباط مالتی مدیا تمام نودهایی که مرتبط هستن؟؟
براساس پروتکل RTCP هر یک از این ها یک گزارش هایی از وضعیت ارتباط اماده می کنن و به بقیه می دن که اینارو به عنوان sender report , receiver report می شناسیم پس sender report , receiver report مثل چاه RTCP هستن که شامل اطلاعات مربوط به اپلیکیشن ها هستن و مالتی مدیاهایی که در حال پخش است برای مثال receiver اگر اینو برای سندر بفرسته توی حالت نقطه به نقطه بعد سندر براساس این اطلاعات می تونه تشخیص بده که کیفیت ارتباط الان چجوری است و اگر لازمه کاری براش انجام بده برای مثال اطلاعاتی که توسط RTCP ارسال میشن می تونن برای مدیریت کیفیت سرویس streaming مالتی مدیامون استفاده بشن و حتی ممکنه براساس این اطلاعات مبدا تشخیص بده که کدینگش مناسب نیست و یک کدینگ با بیت ریت کمتر رو استفاده بکنه مثلا
مثال--> فیلم 14 دقیقه 18:53 تا 19:53 دوباره ببین!!
اطلاعاتی که در receiver report گذاشته میشه --> اماری از time stamp ها و براساس اون ها تاخیرهایی که بسته ها توی شبکه پیدا میکنن تا به مقصد برسن و تغییرات این تاخیر یا jitter و اماری از بسته هایی که گم شدن و به مقصد نرسیدن در نهایت در قالب یک report این هارو برای مبدا ارسال میکنه و برای مبدا به طور مثال از پکت لاس می تونه تشخیص بده که مثلا ریتی که اطلاعات رو داره برای اون می فرسته نسبت به عرض باند شبکه چه وضعیتی داره و اگر بالا است و باعث drop بسته ها میشه ریتشو میتونه بیاره پایین همینطور سندر می تونه sender report رومی تونه بفرسته با استفاده از RTCP که اطلاعاتی در مورد Streaming که در حال انجام هست و زمان در خود نود و تعداد بسته هایی که ارسال میشه و... می تونه توی بسته گذاشته باشه

Real Time Streaming Protocol (RTSP)

- RFC 2326
- Remote media playback control:
  - VCR-like user control of display: play, rewind, fast forward, pause, resume, etc…
- Similar to HTTP but RTSP servers maintain state
- TCP or UDP can be used for the control channel

در اپلیکیشن های مالتی مدیا در کنار پروتکل RTP پروتکل دیگری استفاده میشه به نام RTSP
RTSP: برای اپلیکیشن --> RTP رو برای دیتای اپلیکیشنمون استفاده میکنیم که روی udp قرار میگیره و ارسال میشه منتها همزمان ما ما کارهای کنترلی مربوط به این ارتباط رو از طریق RTSP روی مدیا می تونیم انجام بدیم و یکسری از کارهای کنترلی که لازم است برای مدیا انجام بگیره مثلا برای حالت streaming --> ما کنترل هایی مثل play , paus, stop .., باید بتونیم انجام بدیم --> عکس
از طریق شبکه هم اگر ما داریم یک فیلمی رو به طور مثال از طریق شبکه که از سروری داره پخش میشه دریافت میکنیم دوست داریم بتونیم اینو paus اش بکنیم برای مثال چجوری می تونیم این کارو بکنیم؟ پروتکل RTSP در کنار RTP این کارو برای ما انجام میده
پس پروتکل RTSP می تونه این کارهای کنترلی رو انجام بده بین مبدا و مقصد



RTSP Requests (methods)

- **Major methods**
  - **SETUP:** allocate resources for a stream and start an RTSP session
  - **PLAY:** start data transmission on a stream
  - **PAUSE:** temporarily halt a stream
  - **TEARDOWN:** free resources and end the stream
- **Additional methods**
  - **OPTIONS:** get available methods
  - **DESCRIBE:** get low level description of media object
  - **RECORD:** start recording a stream
  - **REDIRECT:** redirect client to new server

SIP: Session Initiation Protocol [RFC 3261]

long-term vision:

- all telephone calls, video conference calls take place over Internet
- people identified by names or e-mail addresses, rather than by phone numbers
- can reach callee *(if callee so desires)*, no matter where callee roams, no matter what IP device callee is currently using

پروتکل SIP:
دیدگاه کلی این است که ارتباطات تلفنی یا تصویری یا ... از طریق اینترنت انجام بگیره
وقتی ارتباطات تلفنی رو می بریم روی شبکه دیتا ما علاوه بر سرویس مکالمه تلفنی سرویس های ارزش افزوده زیادی رو می تونیم به دست بیاریم و خیلی از محدودیت هایی که توی شبکه تلفنی معمولی داریم در شبکه دیتا دیگه اون محدودیت ها وجود نداره

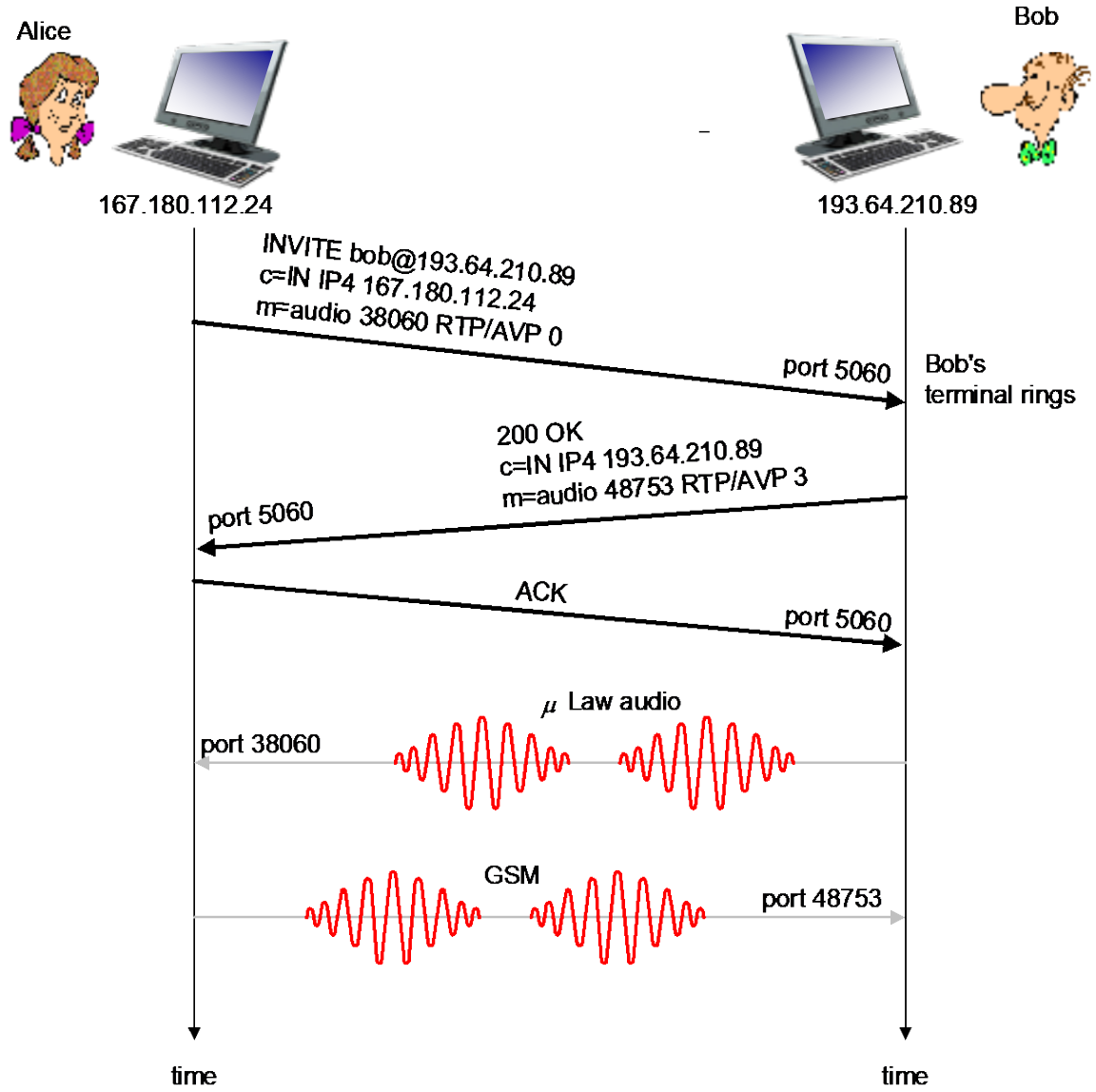
توی شبکه اگر ما از پشت یک کامپیوتری وصل شدیم به شبکه و ip اون شبکه هم از شبکه محلیش گرفته میشه چگونه ما شناسایی میشیم و کسی که میخواد به ما زنگ بزنه از کجا می فهمه ما کجا هستیم و اون اطلاعات تماسشو برای ما بفرسته؟ این مستلزم یک پروتکلی است که یک چنین مسئله ای رو حل بکنه و پروتکل SIP برای همچین منظوری طراحی شده توی این پروتکل برای شناسایی لازم نیست ما یک شماره تلفن داشته باشیم بلکه یک id لازم داریم که این id می تونه ایمیل باشه یا اسم یا می تونه شماره تلفن هم باشه

SIP services

- SIP provides mechanisms for call setup:
  - for caller to let callee know she wants to establish a call
  - so caller, callee can agree on media type, encoding
  - to end call
- determine current IP address of callee:
  - maps mnemonic identifier to current IP address
- call management:
  - add new media streams during call
  - change encoding during call
  - invite others
  - transfer, hold calls

سرویس هایی که پروتکل SIP تامین می کنه برای یک ارتباط تلفنی از طریق اینترنت عبارتند از:
اونی که زنگ می زنه می تونه براساس پروتکل SIP به اونی که می خواد زنگ بزنه رو پیدا بکنه و با استفاده از این پروتکل این ها بهم وصل میشن
بعد براساس همین پروتکل اونا می تونن در مورد کدینگ اون مدیایی که بینشون رد و بدل خواهد شد توافق بکنن و این ها می تونن برای اتمام مکالمه اقدام بکنن
این ها می تونن با استفاده از همین پروتکل ip ادرس همدیگر رو به دست میارن --> این مسئله مهمی است ک ما الان ip ادرسمون چی هست چون توی شبکه انتقال اطلاعات به ما براساس ip ادرس صورت می گیره --> چون وقتی که ما پروتکل SIP رو استفاده میکنیم لایه های پایین شبکه ما درست نخوردن همون کارهای قبلی است و چیز جدیدی نداره داخلش
همینطور پروتکل SIP یکسری مدیریت هایی رو برای مکالمه ای که برقرار است امکان پذیر میکنه از جمله اطلاعات مکالمه و اطلاعات طرفین می تونه منتقل بشه و می تونن افراد دیگر رو دعوت بکنن به مکالمه و حتی می تونن حین مکالمه encoding سیگنال رو تغییر بدن و...

Example: setting up call to known IP address



- Alice's SIP invite message indicates her port number, IP address, encoding she prefers to receive (PCM μlaw)
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP
- default SIP port number is 5060

پروتکل SIP چجوری کار میکنه؟
اساس کار پروتکل SIP و کار اصلیش به این صورت است که اونی که میخواد زنگ بزنه مسیج INVITE رو می فرسته و این مسیج INVITE رسونده میشه به اونی که میخواد بهش تلفن بشه و اون تلفنش زنگ می خوره و اگر گوشی رو برداشت و پاسخ داد مسیج OK می فرسته و اگر این OK توسط اونی که زنگ زده دریافت بشه این ACK می فرسته و به این ترتیب ارتباط برقرار میشه و بعد دیگه با هم حرف می زنن
مسائلی که اینجا داریم:
اولا اینی که داره زنگ می زنه و INVITE رو می فرسته باید اینو به چه ادرسی بفرسته؟ توی این مرحله فرض میکنیم ادرس IP باب رو داریم پس این بسته می رسه به باب و توی این بسته چه اطلاعاتی است؟ توی این بسته اطلاعاتی این که چه کسی داره کال می کنه و IP ادرسش چی هست ینی IP ادرس الیس اینجا می تونه توی این ارسال بشه و در نتیجه باب IP ادرس الیس رو به دست میاره و بعدش باب IP ادرس خودش رو هم می فرسته (حالا ما اینجا فرض کردیم که الیس از قبل اینو داشته) بعد از این مرحله پکت های ویس که روی RTP و روی UDP قرار داده میشن اینا به لایه IP که می رسن برای این IP ادرس ها بین دو طرف رد و بدل میشن ینی الیس برای IP ادرس باب می فرسته و باب برای IP ادرس الیس
پروتکل SIP باز port number مشخصی داره و از این پورت نامبر استفاده میکنه و البته بسته های پروتکل SIP می تونه هم روی tcp ارسال بشه و هم روی udp و البته وقتی که به مرحله دیتا رسیدیم دیگه دیتا روی udp ارسال میشه
قابلیتی که SIP در اختیار دو طرف قرار میده --> سیگنالینگ است ینی SIP امکان سیگنالینگ رو بین این ها فراهم میکنه و اون رد و بدل کردن اطلاعات کنترلی است مثلا قطع ارتباط یا گذاشتن گوشی از طریق همین پروتکل SIP انجام میگیره
در عین حال اگر گوشی رو برنداره کال ریجکت میشه --> این ها سیگنالینگ هایی هستن که با مسیج هایی مختف SIP بین مبدا و مقصد رد و بدل میشن
تمام کارهایی که توسط پروتکل SIP انجام می گیره به صورت نرم افزاری است و ما می تونیم به صورت انعطاف پذیری این هارو گسترش بدیم برای مثال اگر گوشی رو برنداریم می تونیم پیغام busy رو برگردونیم
هر کس می تونه یه شرایطی رو بذاره که اون کالی که براش میاد رد بشه یا قبول بشه یا ذخیره بشه یا .. --> همه اینا با نرم افزار قابل انجام است

Example of SIP message

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:

- HTTP message syntax
- sdp = session description protocol
- Call-ID is unique for every call

- Here we don't know Bob's IP address
  - intermediate SIP servers needed
- Alice sends, receives SIP messages using SIP default port 506
- Alice specifies in header that SIP client sends, receives SIP messages over UDP

یک نمونه از پیغام است که وارد جزئیاتش نمی شیم فقط در حد یک معرفی میگیم:
INVITE: مشخص میکنه که با پروتکل SIP می خوایم با کسی که ID اش اینجا مشخص شده مکالمه بکنیم
در این مثال ما IP باب رو نداریم و فقط IDاش رو داریم در نتیجه در پیغام INVITE مون ایدی اون رو قرار میدیم و توی این حالت ما پیغام INVITE رو می فرستیم به یک پروکسی سرور برای سرویس VOIP توی شبکه خودمون و اون پروکسی سرور کارش این است که این ایدی رو ترجمه بکنه و IP باب رو پیدا بکنه شبیه DNS است الان این پس در کل واسطه اینجا در کار است و که از طریق این واسه این کار انجام خواهد شد
همین طور توی پیغام مشخص میکنیم این پیغام از کی داره میاد و برای کی هست و برای این کال که میخواد با این پیغام برقرار بشه یک ID براش تعریف میکنیم و content type رو مشخص میکنیم و یک پروتکل sdp هم داریم که روی SIP این سوار میشه و توی دل SIP قرار میگیره و این پروتکل یک پروتکلی برای ارائه اطلاعات بین مبدا و مقصد است

Name translation, user location

- caller wants to call callee, but only has callee's name or e-mail address.
- need to get IP address of callee's current host:
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, smartphone, car device)
- result can be based on:
  - time of day (work, home)
  - caller (don't want boss to call you at home)
  - status of callee (calls sent to voicemail when callee is already talking to someone)

در حالتی که ip شخصی که می خوایم بهش زنگ بزنیم رو نداریم از id اش استفاده میکنیم و این ایدی توی یک سیستمی شبیه DNS که در سطح شبکه توزیع شده ترجمه میشه به ip ادرس ینی ip ادرسی که الان اون شخص از طریق اون ip ادرس به شبکه وصل است

مکانیزیمی که SIP فراهم میکنه این امکان رو فراهم میکنه که ما براساس مثلا چه ساعتی از روز است یا چه روزی از هفته است و براساس این که ما الان کجا هستیم این امکان به وجود بیاد ما تماس هایی که بهمون میشه رو فیلتر بکنیم یا جواب ندیم یا رد بکنیم یا بفرستیم برای ایمیل یا...

SIP registrar

- one function of SIP server: registrar
- when Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server

register message:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

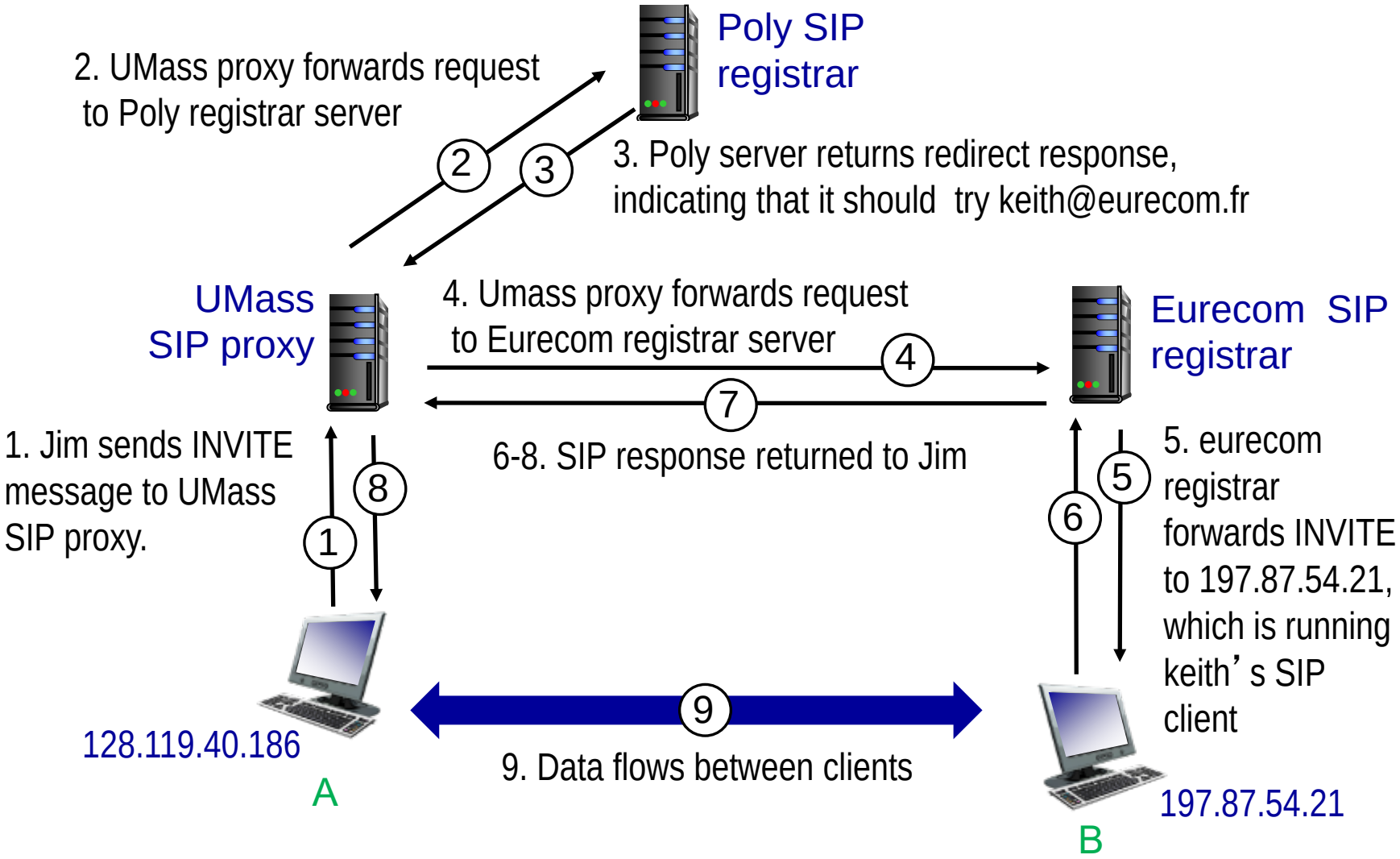
یکی از سرورهایی که برای اجرای SIP در سطح شبکه لازم است SIP registrar است و SIP registra عملا یک دیتابیسی است که هر کلاینت SIP هر جا که به اینترنت وصل میشه از اونجا به registrar خودش وصل میشه و ip ادرس محلی خودش رو در اختیار registrar قرار میده به این ترتیب اگر ما پشت کامپیوتر بشینیم و اپلیکیشن voip رو ران بکنیم که اینجا SIP client اون اپلیکیشنه است این اپلیکیشن براساس کانفیگی که روش قرار داده شده که توش ادرس یک registrar محلی بهش داده شده میاد خودش رو و ip ادرس خودش رو توی اون registrar ثبت میکنه به این ترتیب ما یک دیتابیسی داریم که توی هر شبکه ای کلاینت های SIP در اون شبکه ip فعلیشون اونجا registrar میشه و این registrar کاملا قابل اپدیت شدن به صورت داینامیک است مثلا اگر شخص جاشو عوض بکنه و بره پشت یک کامپیوتر دیگه بشینه و می تونه دوباره وصل بشه به registrar و جای جدیدشو ثبت بکنه

SIP proxy

- another function of SIP server: *proxy*
- Alice sends invite message to her proxy server
  - contains address sip:bob@domain.com
  - proxy responsible for routing SIP messages to callee, possibly through multiple proxies
- Bob sends response back through same set of SIP proxies
- proxy returns Bob's SIP response message to Alice
  - contains Bob's IP address
- SIP proxy analogous to local DNS server plus TCP setup

کار اصلی پروکسی سرور SIP عبارت است از:
پیغام این INVITE رو دریافت بکنه از کلاینت و بعد این رو براساس اون ایدی که توش است با استفاده از registrar این رو ترجمه بکنه و بعد به صورت دست به دست به از طریق پروکسی های دیگر اینو برسونه به اون شبکه ای که به اونی که میخوایم زنگ بزنیم اونجا است و اونجا هم پروکسی محلی وظیفه اش این است که درخواست کالی که اومده رو منتقل بکنه به اون کلاینتی که مورد نظر است

SIP example: jim@umass.edu calls keith@poly.edu



مثال:
از A می خوایم به B یک درخواست مکالمه بفرستیم و ما نمی دونیم ip ادرس B الان چی هست کاری که می کنیم این است که پیغام INVITE رو می فرستیم با ایدی اون شخص مثلا jim@umass.edu می خواد با keith@poly.edu حرف بزنه
پس پیغام INVITE رو به پروکسی UMass <-- SIP می فرستیم ینی به شبکه محلی خودش می فرسته این INVITE رو
و پروکسی SIP اولین کاری که باید کنه اینه که باید ببینه این شخصی که میخوایم باهاش صحبت بکنیم این کجاست و جاشو پیدا بکنه پس این ایدی رو می فرسته به registrar و registrar مشابه DNS ادرس پروکسی سروری که می تونه این یوزر رو ترجمه بکنه رو برمی گردونه و ما درخواست رو می فرستیم برای اون پروکسی سرور و اون حالا ادرس ip کلاینت رو داره و اونو می شناسه و Eurecom در واقع پروکسی SIP همون شخصی است که می خوایم باهاش تماس بگیریم و اینو می فرسته برای ماشین اون شخص بعد اینجا INVITE می رسه به طرف مقابل و در پاسخ این، این می تونه اینو اوکی بکنه و این اوکی به پروکسی سرور شبکه ما برگردونده میشه و بعد به خود A رسونده میشه و بعد ACK میشه دوباره و بعد دیتا می تونه رد و بدل بشه