



دانشگاه صنعتی همدان



شبکه‌های کامپیوتری - مخابراتی

دکتر رجبی

نیمسال دوم سال تحصیلی ۹۸-۹۹

دانشگاه صنعتی همدان

گروه مهندسی برق و کامپیوتر

عنوان جلسه: پروتکل ها و لایه ها

نیاز به ساختار ماژولار

نیاز به ساختار ماژولار

✓ مثال: چک کردن ایمیل از روی گوشی

ارتباط بی سیم به BTS مخابراتی، وصل به شبکه اپراتور موبایل، یک شبکه ISP در کشور، شبکه backbone کشور، از طریق gateway بین الملل و توسط فیبر به شبکه جهانی

✓ نحوه پیاده سازی مناسب؟

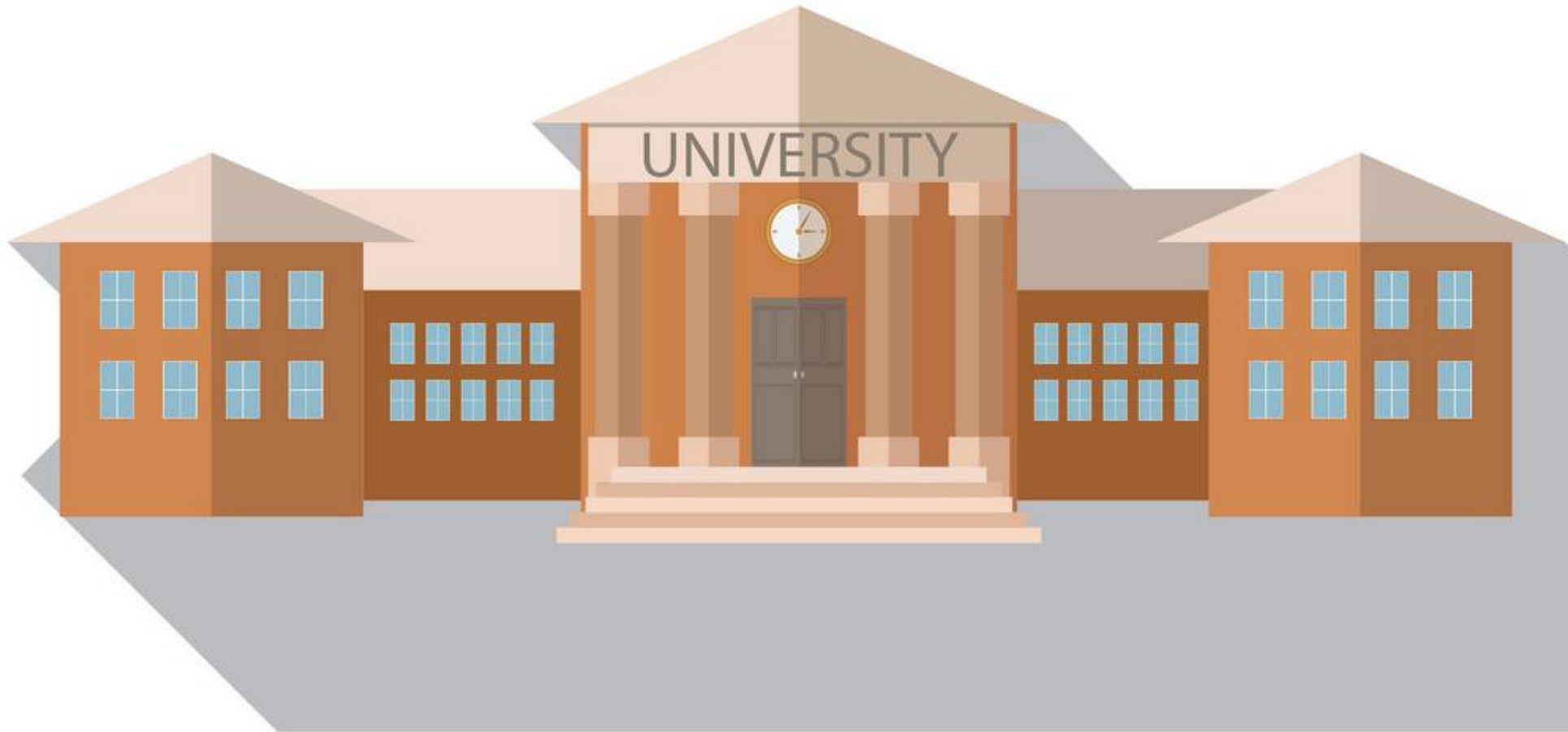
✓ پیچیدگی و وسعت زیاد کارها منجر به تقسیم بندی کارها شده است.

✓ تفکیک کار خیلی بزرگ به کارهای کوچک تر

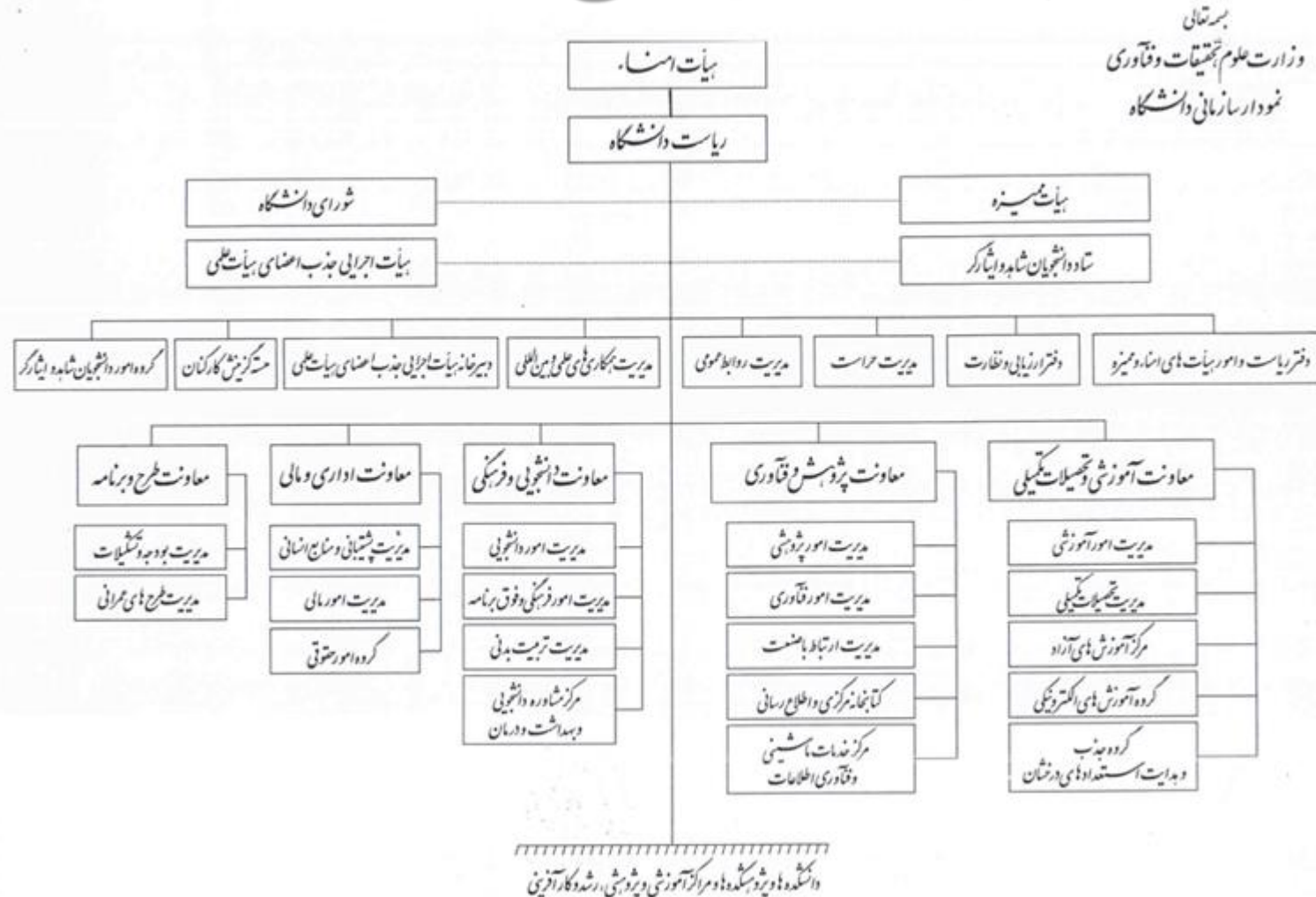
✓ تخصصی شدن کارها

✓ مجموعه فعالیتها و عملکردهایی که برای ارتباط گرهها لازم است را به صورت مفهومی می توان به لایه های مختلف تقسیم بندی کرد.

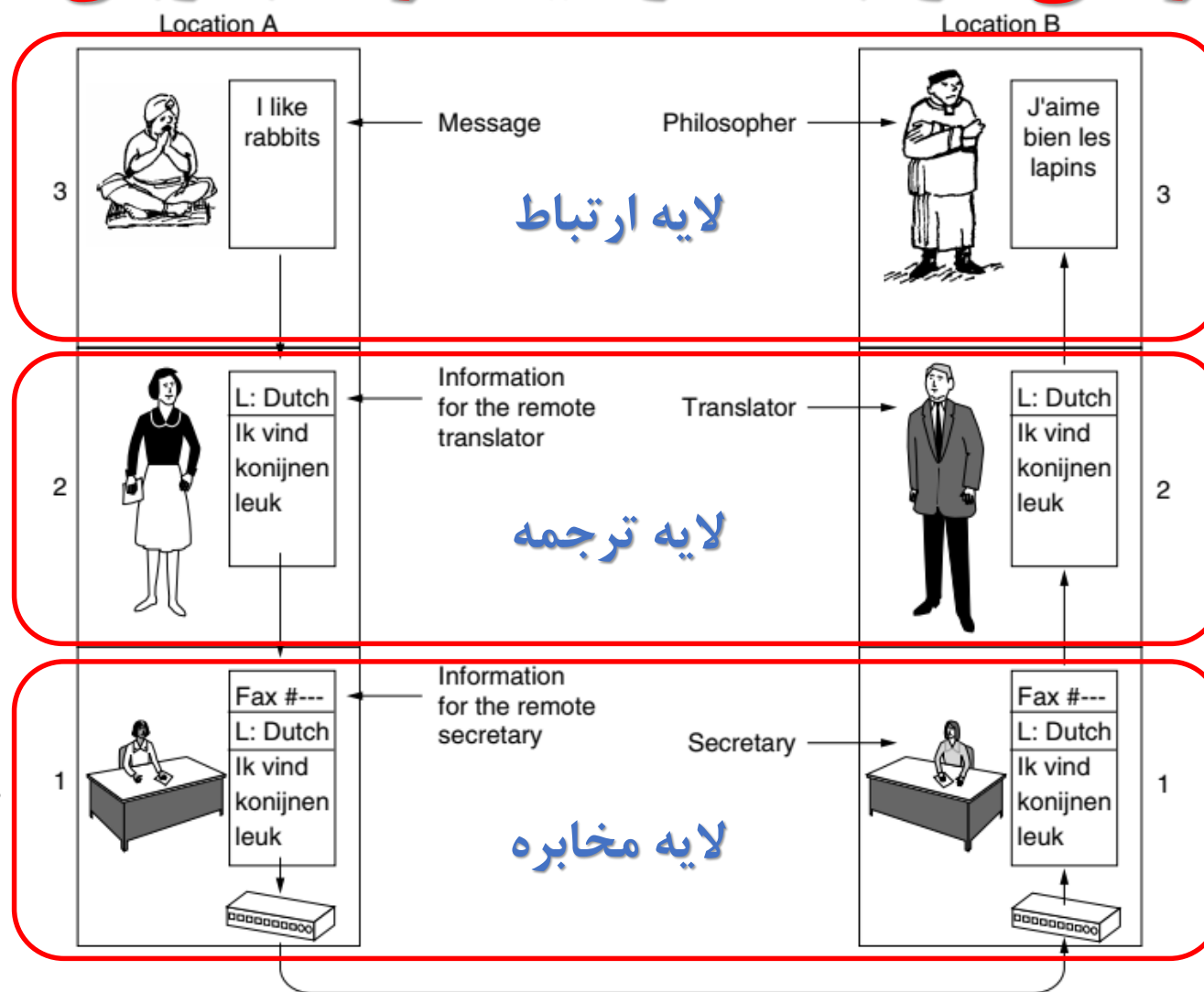
نحوه مدیریت دانشگاه



چارت سازمانی دانشگاه



مثال انگیزشی: ارتباط دو فیلسوف با زبان های مختلف

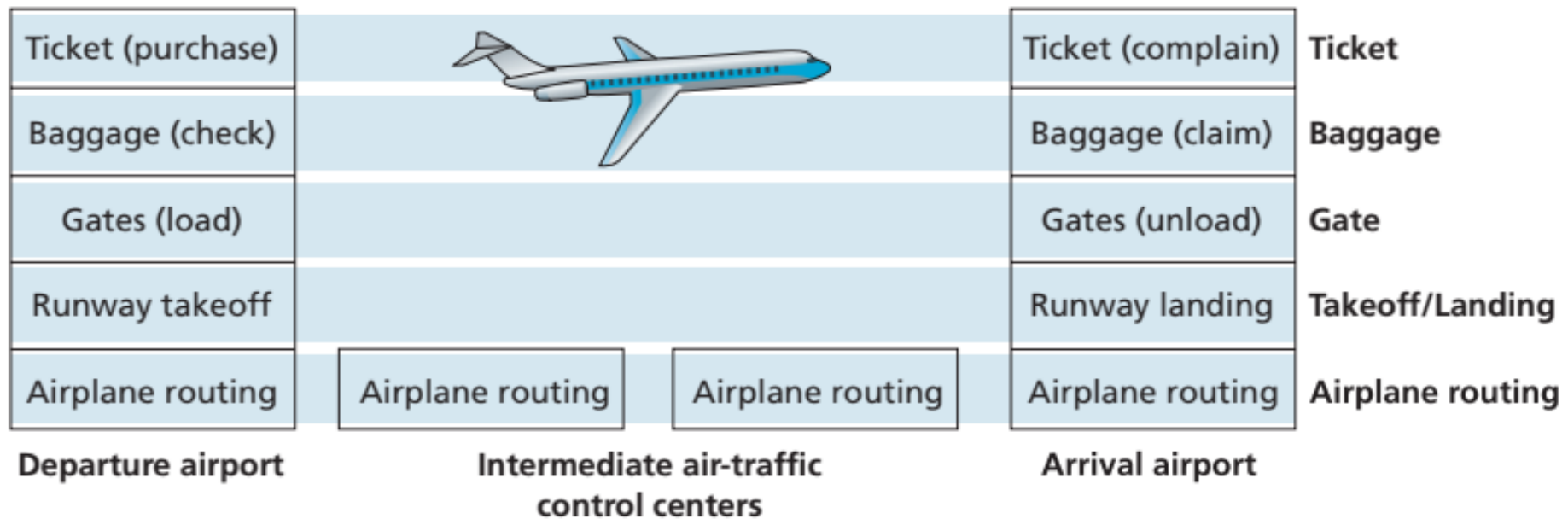


Source: A.S. Tanenbaum, D.J. Wetherall, "Computer networks", Pearson, 5th ed, 2011.

مثال انگیزشی: ارتباط دو فیلسوف با زبان‌های مختلف

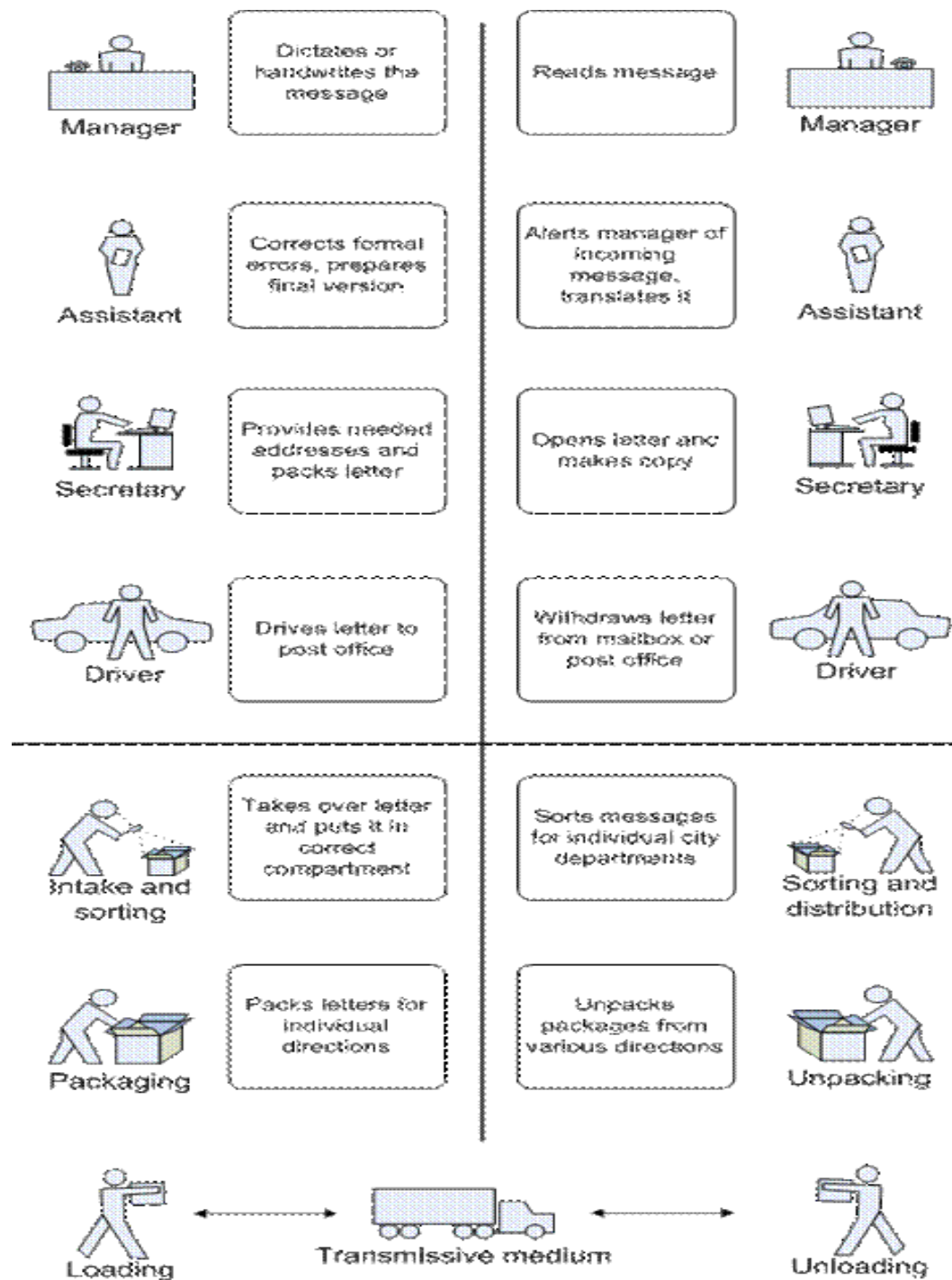
- مثال: گفتگوی فلسفی بین دو فیلسوف هندی و چینی که زبان مشترکی ندارند.
- توافق: انتخاب زبان مشترک مثلاً هلندی (Dutch) برای تعامل
- مترجم‌ها، وظیفه ترجمه حرف‌های فیلسوف‌ها را به زبان مورد توافق طرفین دارند.
- مخاطب فیلسوف چه کسی است؟ فیلسوف دیگر. اما متن خودش را دست مترجم می‌دهد.
- مترجم به فیلسوف سرویس ارائه می‌دهد.
- مخاطب مترجم کیست؟ مترجم دیگر

مثال انگیزشی دیگر: سفر با هواپیما



Source: J. Kurose, and R. Keith, "Computer networks: A top down approach featuring the internet." *Addison Wesley*, 5th ed., (2010).

نمونه یک ارتباط پستی



نیاز به ساختار ماژولار

- کارهایی که شبکه برای برنامه‌ها انجام می‌دهد:
 - ایجاد و قطع کردن ارتباط
 - پیدا کردن یک مسیر در میان شبکه
 - انتقال قابل اطمینان اطلاعات
 - انتقال اطلاعات با طول‌های دلخواه
 - ارسال سریع، تا جایی که شبکه اجازه دهد (محدودیت سرعت شبکه).
 - اشتراک پهنای باند میان کاربران
 - ایجاد امنیت برای اطلاعات در حال تبادل
 - فراهم کردن امکان پیوستن میزبان‌های جدید
 -

نیاز به ساختار ماژولار

- کارهایی که شبکه برای برنامه ها انجام می دهد:
- ایجاد و قطع کردن ارتباط
- پیدا کردن ما به یک فرم از ماژولار بودن
- انتقال قاب برای کمک به مدیریت
- انتقال اطمینان و پشتیبانی از
- ارسال استفاده مجدد (reuse) از هد (محدودیت سرعت شبکه).
- اشتراک کارکردهای شبکه نیاز داریم.
- ایجاد امنیت برای اطلاعات در حال تبادل
- فراهم کردن امکان پیوستن میزبان های جدید
-

نیاز به ساختار ماژولار

- ✓ پیاده‌سازی‌های انعطاف‌پذیر و مدیریت‌پذیر
- ✓ مریض شدن مترجم ← عوض شدن پروتکل از زبان هلندی به انگلیسی
- ✓ چک کردن ایمیل از روی گوشی و کامپیوتر

پروتکل‌ها و لایه‌ها

- **پروتکل‌ها و لایه‌ها** روش ساختاری اصلی استفاده شده برای افراز عملکرد شبکه است.
- **پروتکل (Protocol):** زبان گفتگوی مشترک دو تا لایه متناظر هم در دو طرف
 - پروتکل لایه ۳ در مثال انگیزشی: بحث فلسفی
 - پروتکل لایه ۲ در مثال انگیزشی: زبان هلندی
 - پروتکل لایه ۱ در مثال انگیزشی: فاکس
- هر پروتکلی در واقع با همتای خود (Peer) به صورت مجازی از طریق پروتکل در ارتباط است.
- هر پروتکلی فقط از **سرویس‌های لایه پایینی** خود استفاده می‌کند.

پروتکل‌ها و لایه‌ها

- **پروتکل:** مجموعه قواعد و قوانین مشخص، به شکل یک الگو، برای برقراری ارتباط بین کامپیوترهای یک شبکه. پروتکل زبان مشترک بین کامپیوترهاست.

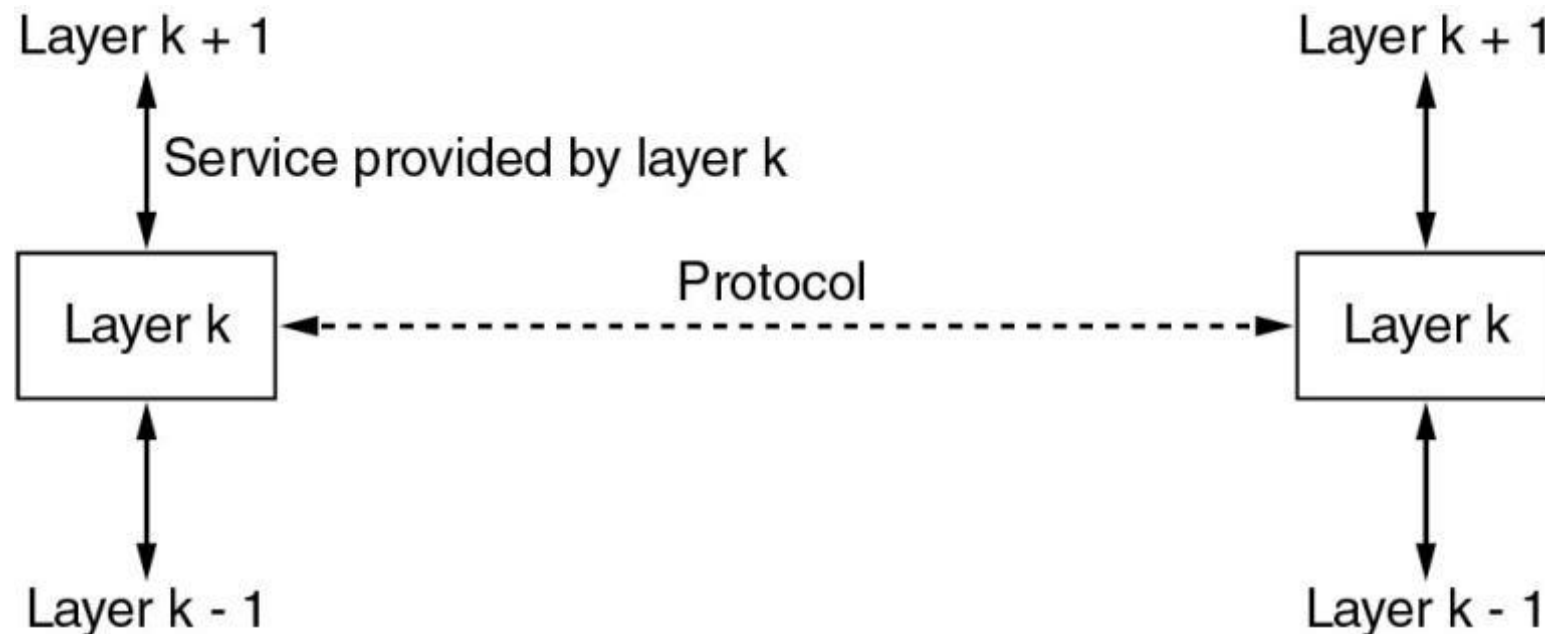
پروتکل‌ها و لایه‌ها

- **پروتکل:** مجموعه قواعد و قوانین مشخص، به شکل یک الگو، برای برقراری ارتباط بین کامپیوترهای یک شبکه. پروتکل زبان مشترک بین کامپیوترهاست.

Application Telnet, ftp, NFS, NIS	
Session e.g. RPC	
Transport Sockets / Streams - TLI	
TCP	UDP
Network IP +ARP/RARP/ICMP	
Physical protocol Ethernet/TR/FDDI/PPP	
Transmission medium Coax, Fiber, 10baseT..	

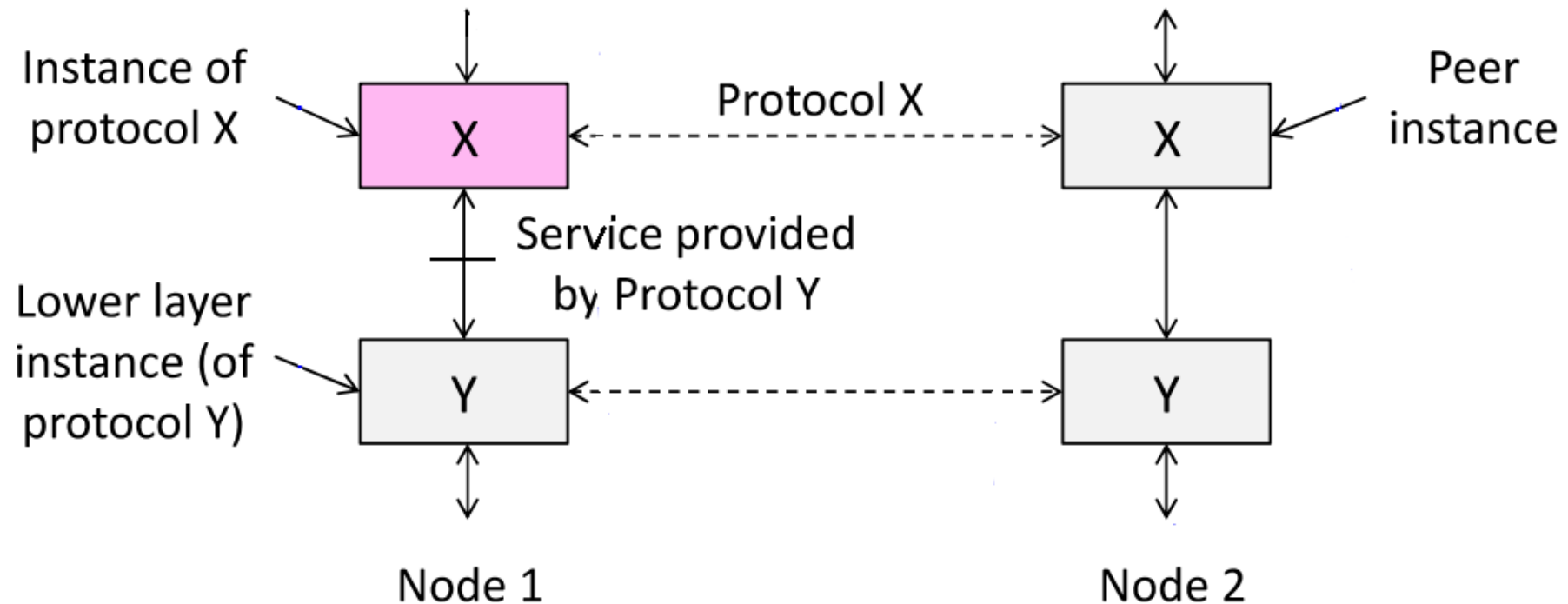
پروتکل‌ها و لایه‌ها (۲)

- **سرویس:** وظایف هر لایه از مدل OSI را مشخص میکند. (مجموعه عملکردهایی که یک لایه، در اختیار لایه بالاتر از خود قرار می دهد.)

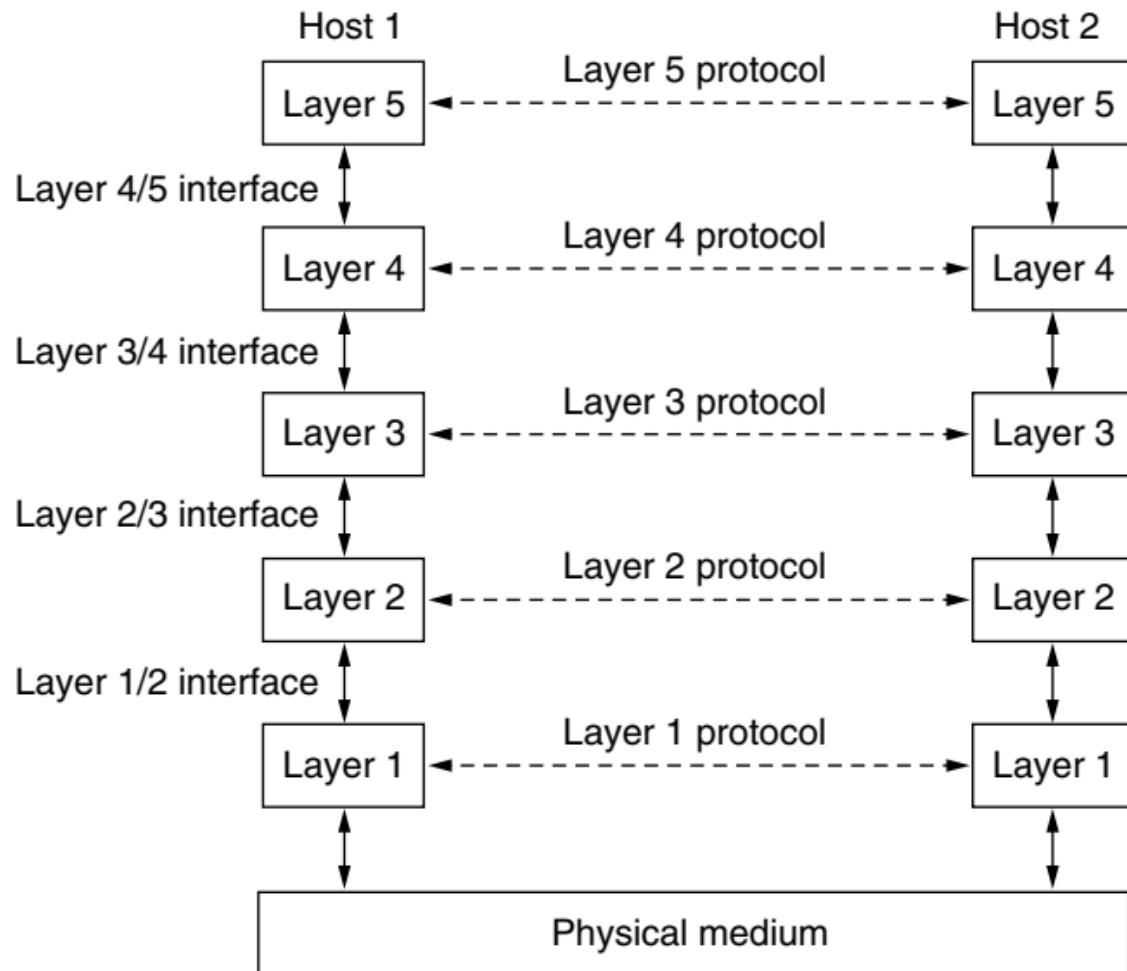


پروتکل‌ها و لایه‌ها (۲)

- پروتکل‌ها افقی و سرویس‌های لایه‌ها عمودی هستند.



پروتکل‌ها و لایه‌ها (۳)



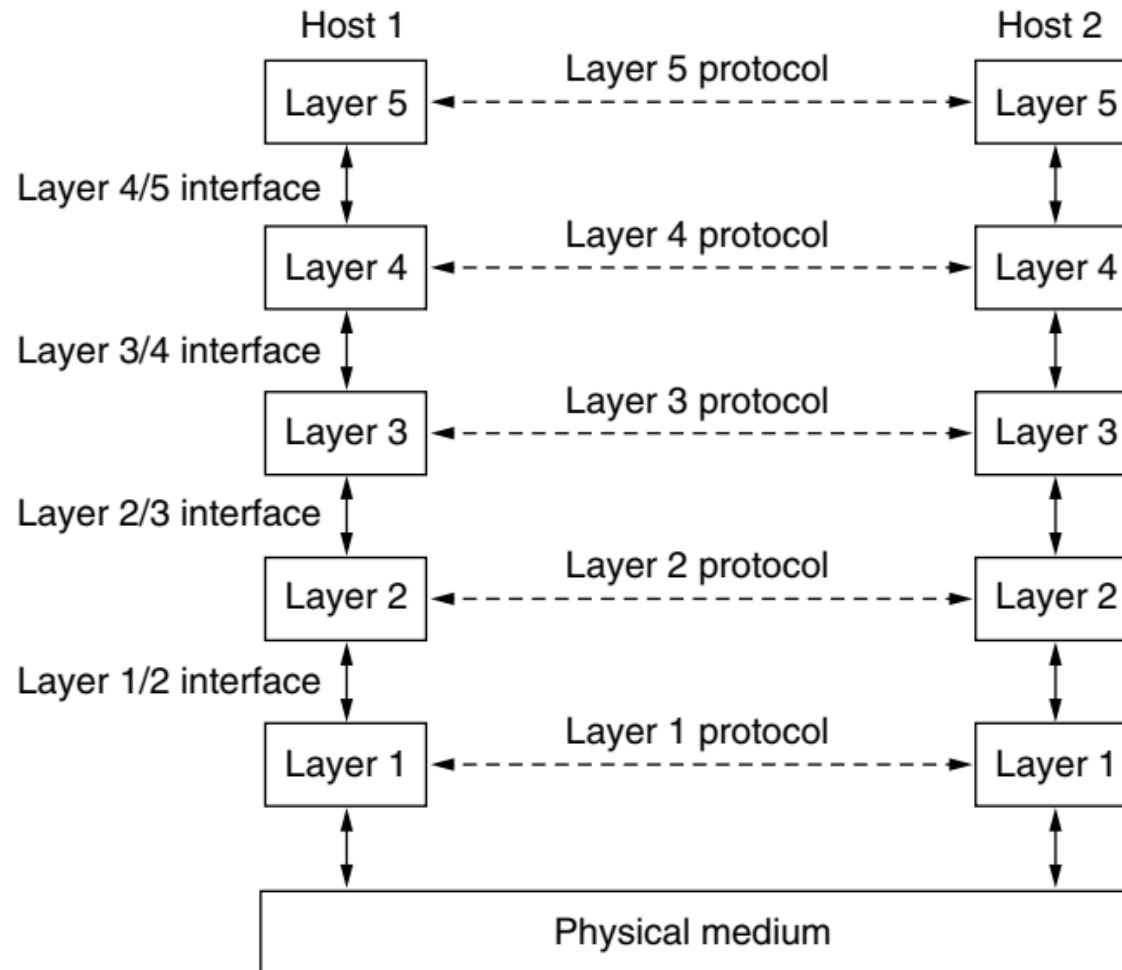
- **پشته پروتکل (Protocol Stack):**

چینشی از پروتکل‌های (یک پروتکل برای هر لایه) در حال استفاده لایه‌های مختلف سیستم

- **واسط (Interface):**

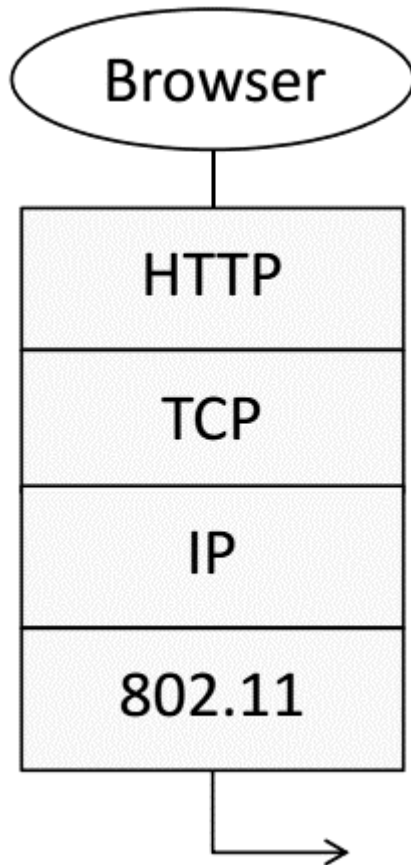
واسط ارتباطی بین یک لایه با لایه‌های بالایی و پایینی. نحوه تعامل لایه‌ها با همدیگر از طریق واسط صورت می‌پذیرد. می‌تواند فیزیکی و یا نرم‌افزاری باشد.

پروتکل‌ها و لایه‌ها (۴)



- هر لایه یک سری وظایفی دارد.
- مشخص می‌کنیم که هر لایه چگونه با لایه‌های بالایی و پایینی خود صحبت کند.
- چه سرویسی را باید به لایه بالاتر تحویل دهد و چه سرویسی را باید از لایه پایین‌تر تحویل بگیرد.
- تقسیم وظایف برای هر لایه یک مسئله طراحی است. جواب یکتای بهینه ندارد. برای هر کاربردی، تقسیم وظایف متفاوت است.

پروتکل‌ها و لایه‌ها (۵)



- پروتکل‌هایی که احتمالاً تاکنون شنیده‌اید:

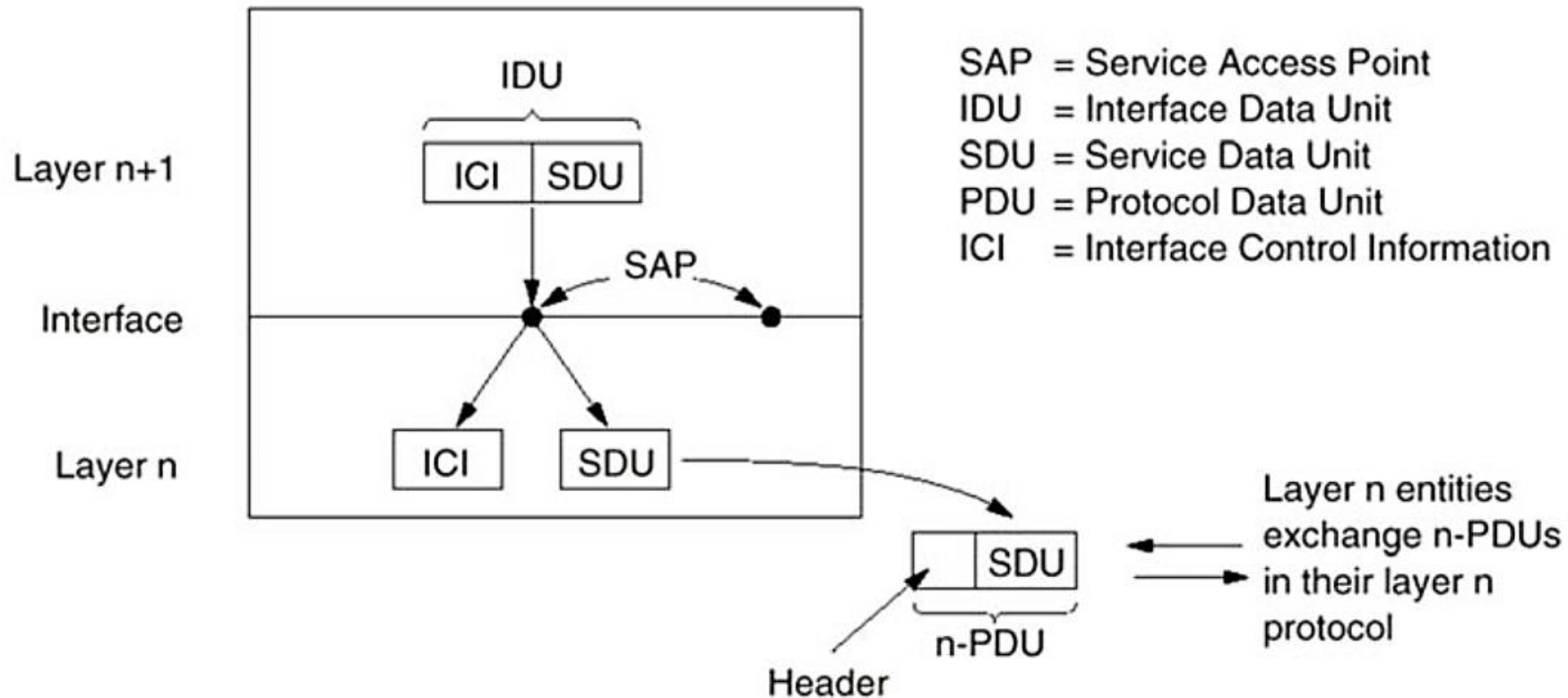
– TCP , IP , 802.11 , HTTP , SSL , DNS , Ethernet ,

- یک مثال از پشته پروتکل

– توسط یک مرورگر وب بر روی یک میزبان که به صورت بی‌سیم به اینترنت متصل شده است.

پروتکل‌ها و لایه‌ها (۶)

- رابطه بین لایه‌ها در واسط
- معرفی واژگان موجود در تعامل بین لایه‌های مختلف در استانداردها



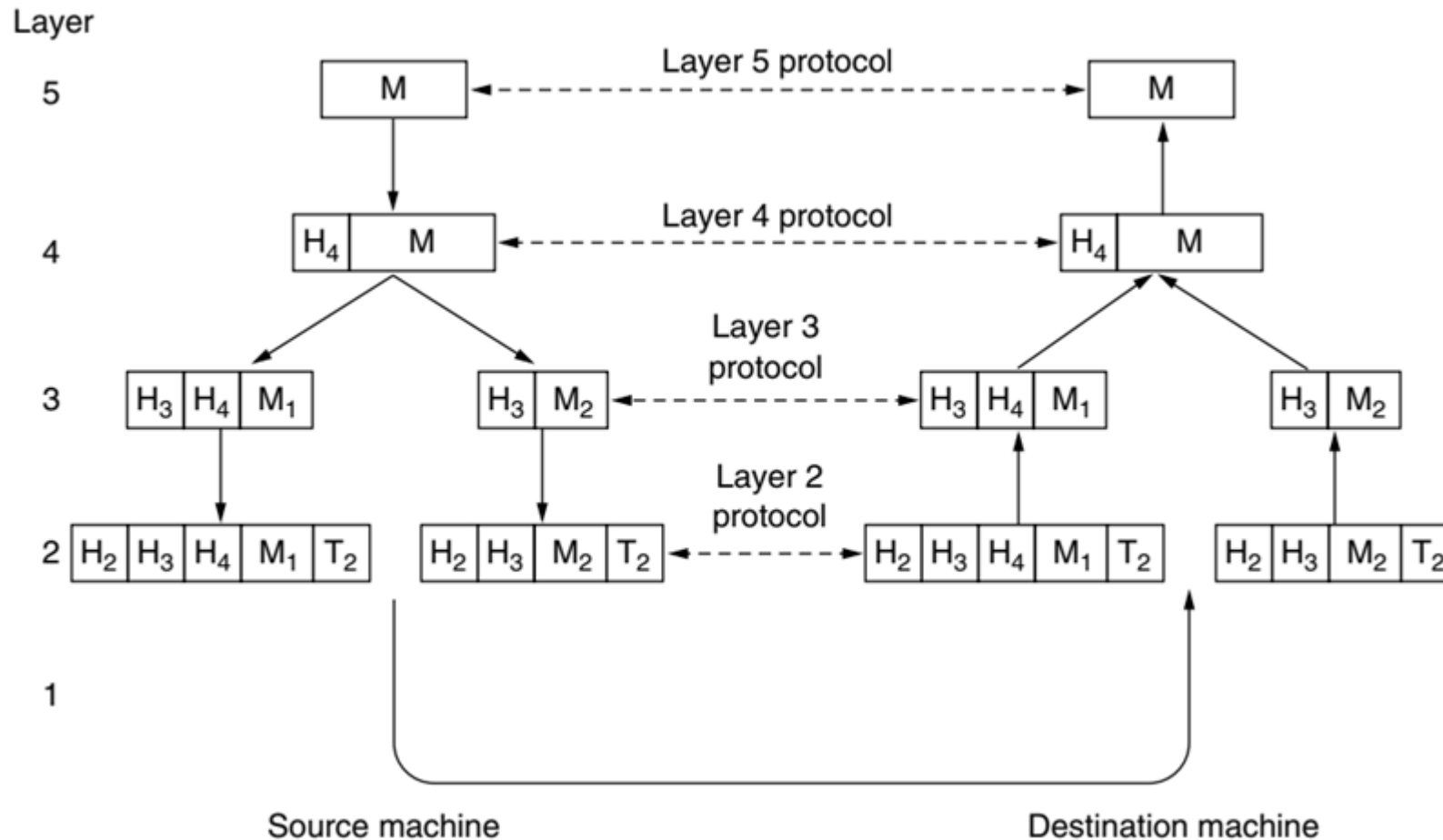
پروتکل‌ها و لایه‌ها (۷)

- **SAP:** محل ارتباط لایه بالایی با لایه پایینی برای گرفتن سرویس است. لزوماً فیزیکی نیست که بتوان دنبال آن گشت. یک مفهوم است.
- به کل بسته اطلاعاتی که لایه بالاتر در اختیار لایه پایین‌تر قرار می‌دهد، **Interface Data Unit (IDU)** گفته می‌شود. این اطلاعات دو بخش هستند:
 - یک بخش از آن، بخشی است که مصرف کننده آن همان لایه پایینی است. به عبارت دیگر مخاطب این بخش از پیام همان لایه پایینی است.
- مثال: مترجم زمانی که اطلاعات را دست منشی می‌دهد، شماره فاکسی که منشی باید از آن استفاده کند را نیز به او می‌دهد. این شماره فاکس بخشی از پروتکل بین مترجم‌ها نیست. بخشی از مکالمه فلسفی دو فیلسوف هم نیست. داده‌ای است که به لایه پایینی داده می‌شود تا لایه پایینی کارش را درست انجام دهد.
 - به این بخش از پیام **Interface Control Information (ICI)** گفته می‌شود.
- بخش دیگر هم همان داده‌ای است که در اختیار لایه پایینی قرار داده می‌شود تا سرویس انجام پذیرد. محتوای نامه ترجمه‌شده در مثال فیلسوف

پروتکل‌ها و لایه‌ها (۸)

- ممکن است لایه‌ها برای انجام وظایف خودشان در خیلی از موارد نیاز به گفتگو با طرف مقابل خودشان داشته باشند.
- برای مثال: ممکن است لازم باشد که مترجم‌ها به همدیگر چیزی بگویند که برای کار ترجمه لازم باشد، اما بخشی از آن مکالمه فلسفی نباشد. به عبارت دیگر، پروتکل این لایه برای انجام درست وظایف خودش، جابجاشدن اطلاعاتی را لازم دارد که اضافه بر مکالمه فلسفی است.
- یکی از متداول‌ترین روش‌ها افزودن سرآیند (Header) است.

پروتکل‌ها و لایه‌ها (۹)



➤ افزوده شدن سرآیند (Header) و پیش آیند (Trailer) به پیام در هر لایه
 ➤ امکان تقسیم به فریم‌های کوچک‌تر

پروتکل‌ها و لایه‌ها (۱۰)

- مثال‌هایی از مشکلات طراحی لایه‌ها
 - آدرس‌دهی
 - قوانین جابجایی داده‌ها
 - یک‌طرفه، نیم دوطرفه، دوطرفه کامل
 - کانال اختصاصی باشد یا نه
 - اولویت‌بندی پیام‌ها
 - کنترل خطا
 - مسیریابی

سرویس

- مجموعه خدماتی است که یک لایه در اختیار لایه بالاتر قرار می دهد.

سرویس های مختلف لایه ها

بی اتصال

اتصال گرا

پست کردن
نامه

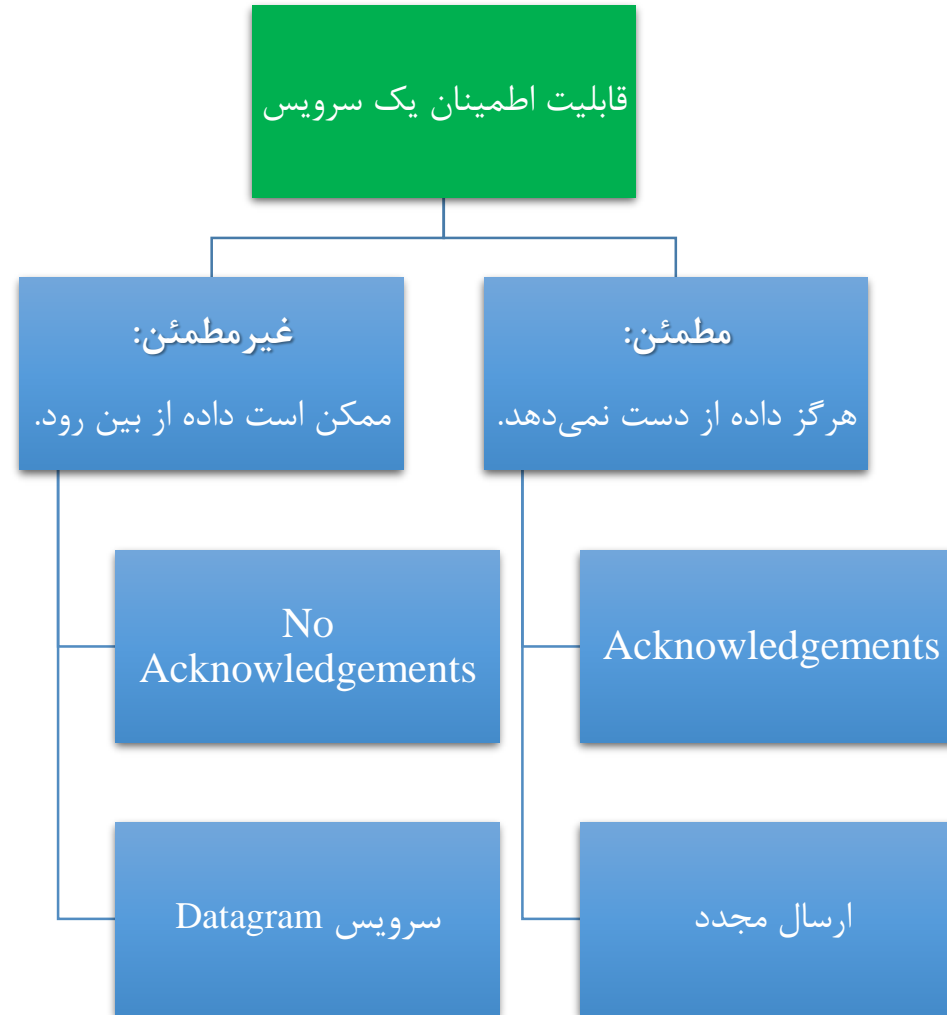
ایمیل

FTP

لاگین از راه
دور

سرویس
مکالمه تلفنی

سرویس (۲)

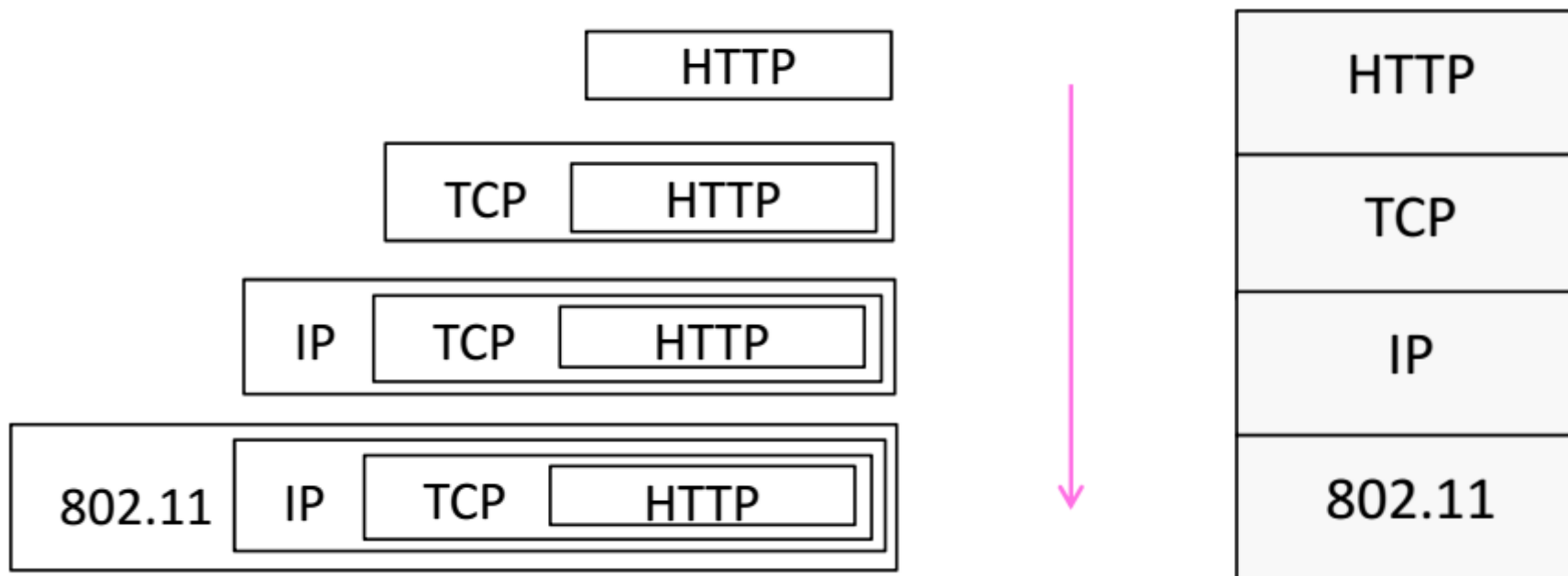


کپسوله سازی (Encapsulation)

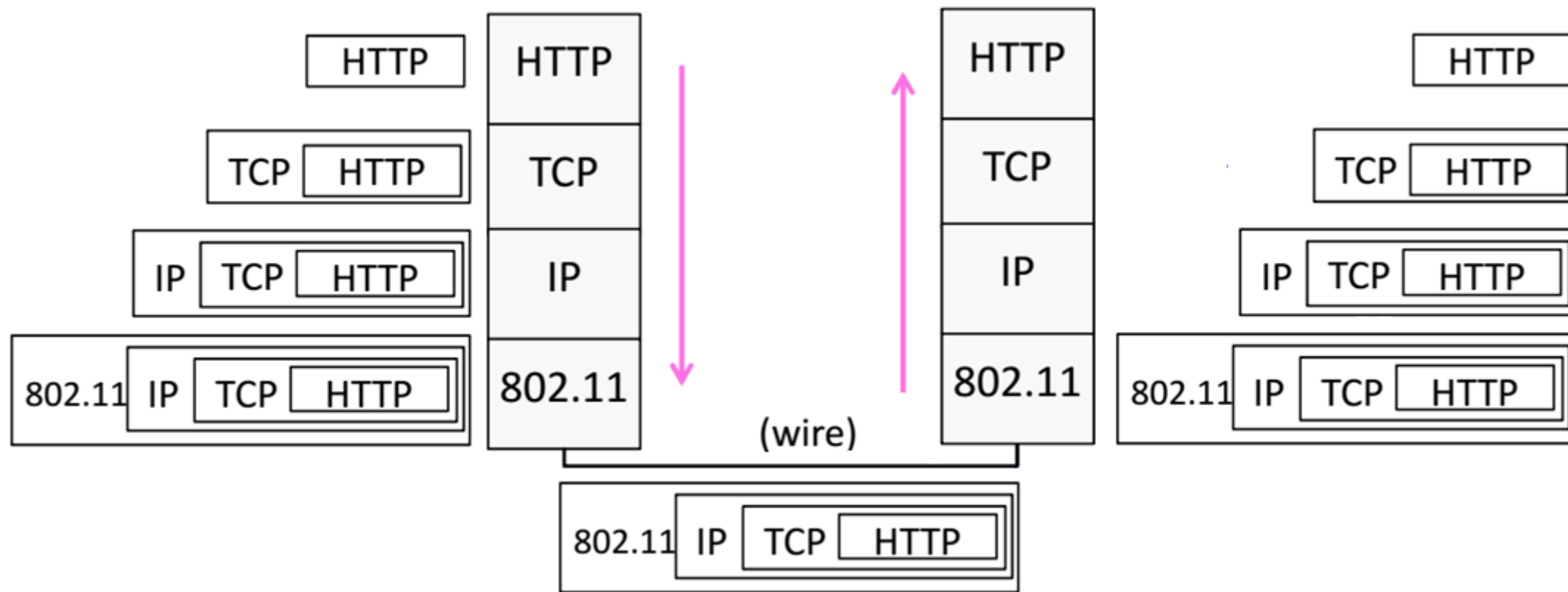
- کپسوله سازی یک مکانیزم است که برای اجرای لایه بندی پروتکل استفاده می شود.
 - لایه پایینی محتوای لایه بالایی را می پوشاند و اطلاعات خودش را به آن اضافه کرده یک پیام جدید برای تحویل می سازد.
 - مانند ارسال یک نامه درون پاکت، اداره پست درون آن را نگاه نمی کند.
 - پاکت نامه همانند لایه پایین است که اطلاعات آدرس گیرنده به صورت سرآیند (Header) بر روی آن نوشته می شود. اداره پست فقط این اطلاعات سرآیند (اطلاعات لایه پایین) را نگاه کرده و نامه را به دست گیرنده می رساند.
 - ممکن است پروتکل های دیگری نیز وجود داشته باشد. برای مثال، نامه هایی که به یک مقصد ارسال می شوند، همگی با هم ارسال شوند. برای مثال نامه هایی که از اردبیل به تهران ارسال می شود. اداره پست فقط نامه ما را ارسال نمی کند، بلکه بسته های زیادی همزمان با استفاده از هواپیما ارسال می شود.

کپسوله سازی (Encapsulation) (۲)

- یک پیغام "روی سیم" ساختار لایه‌ای مشابه پیاز دارد.
– لایه‌های پایینی، خارجی تراند.

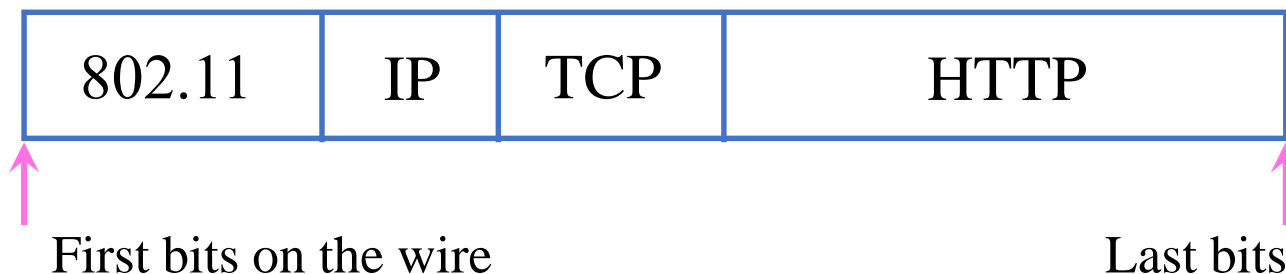


کپسوله سازی (Encapsulation) (۳)



کپسوله سازی (Encapsulation) (۴)

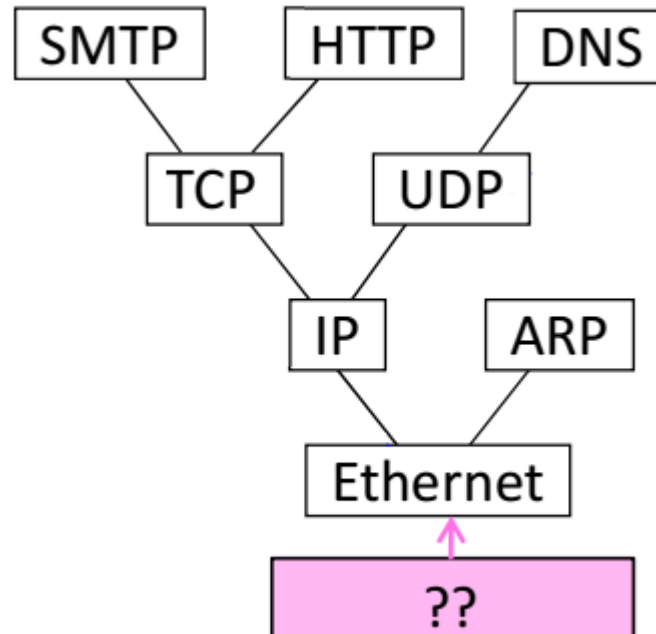
- معمولاً پیام را به این صورت رسم می کنیم.
– هر لایه سرآیند (Header) خود را اضافه می کند.



- پیچیدگی بیشتر در عمل
– پیش آیندها (Trailers) نیز مانند سرآیندها، رمزنگاری (Encrypt) و فشرده سازی (Compress) محتوا
– قطعه قطعه سازی (تقسیم پیام های طولانی) و بازسازی آن در گیرنده.

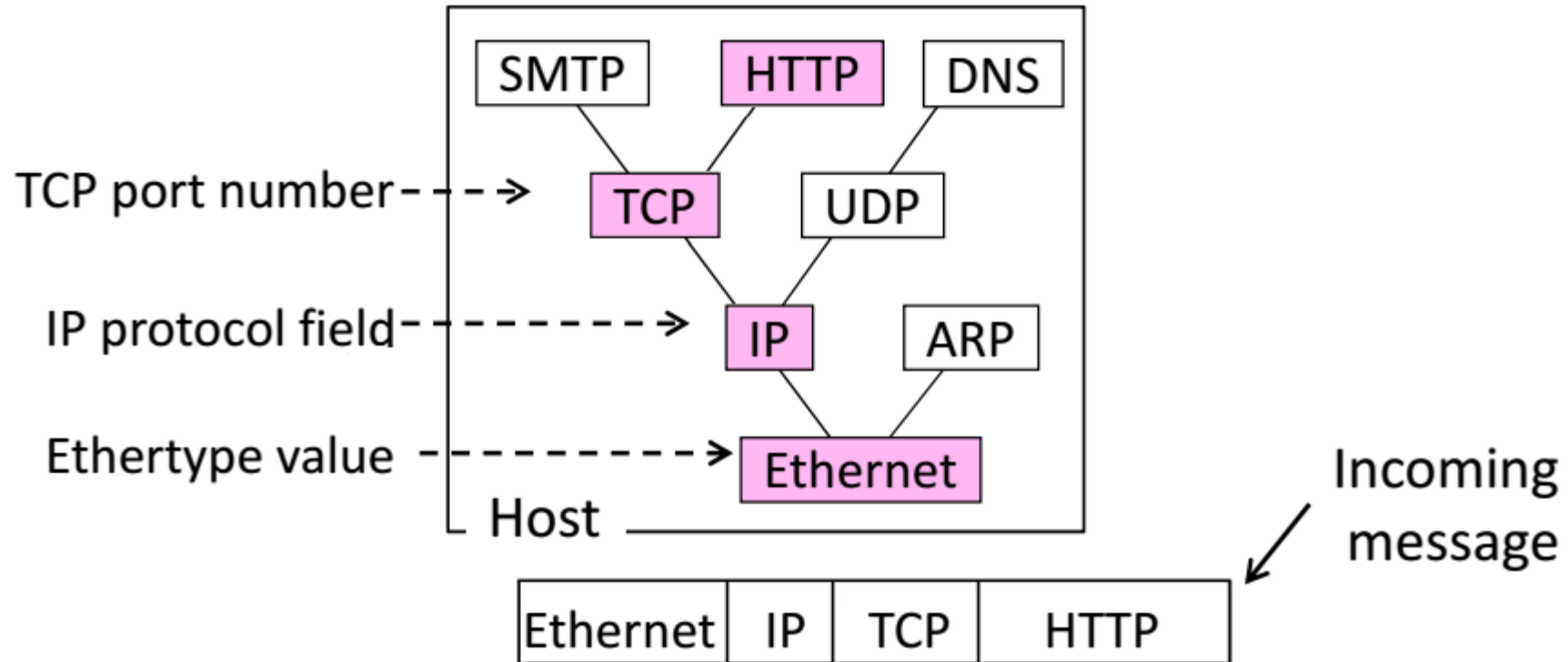
Demultiplexing

- پیام دریافتی باید از پروتکلی که استفاده کرده گذر کند.



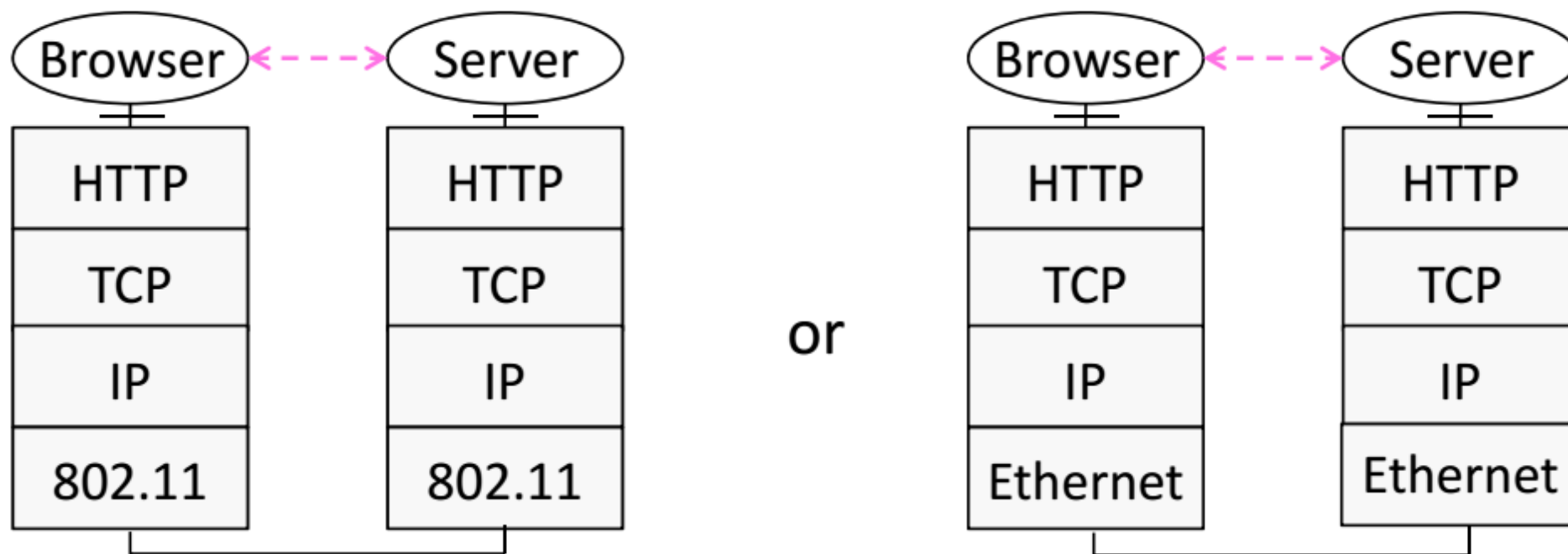
Demultiplexing (2)

- Done with demultiplexing keys in the headers



مزایای لایه بندی

- پنهان کردن و استفاده مجدد از اطلاعات



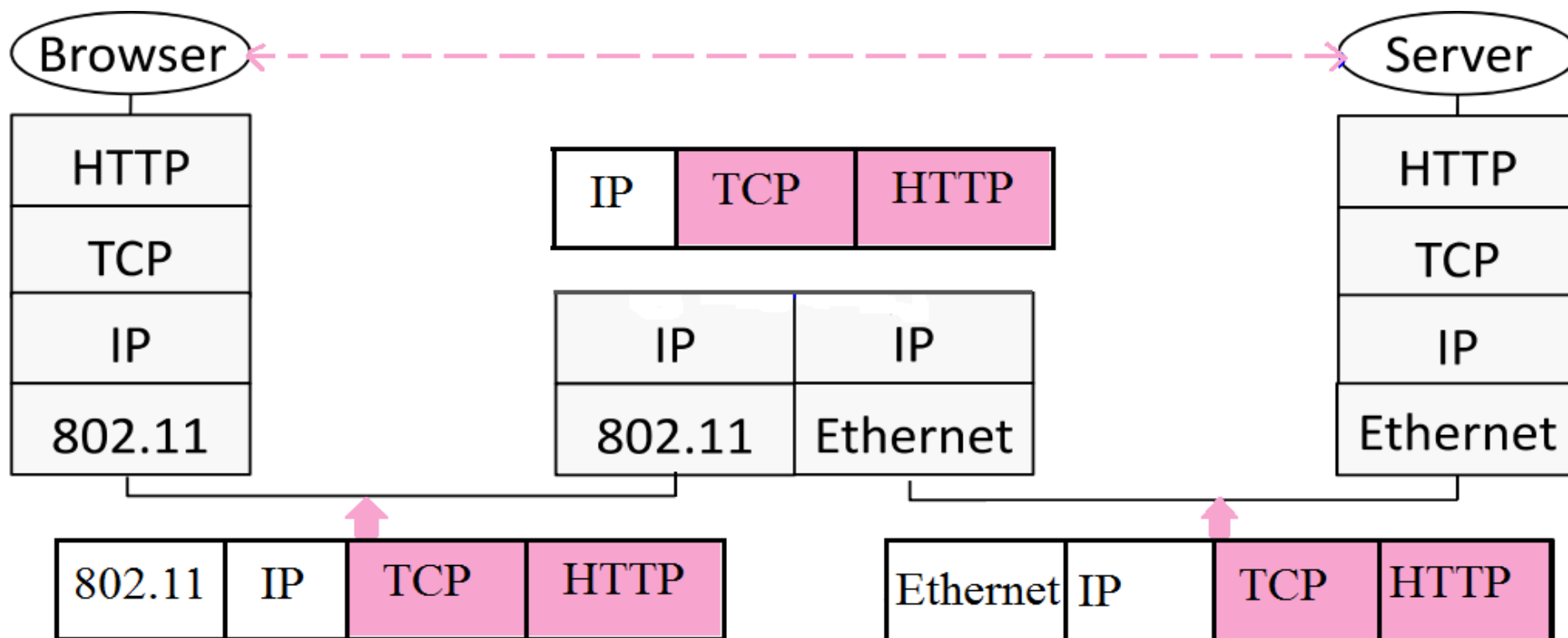
مزایای لایه بندی (۲)

- استفاده از پنهان کردن اطلاعات برای اتصال سیستم های متفاوت



مزایای لایه‌بندی (۳)

- استفاده از پنهان کردن اطلاعات برای اتصال سیستم‌های متفاوت



معایب لایه‌بندی

- اضافه کردن overhead

– Overhead برای پیام‌های طولانی کمتر است.

- پنهان کردن اطلاعات

– برنامه ممکن است برایش مهم باشد که در محیط سیمی در حال اجرا است و یا در محیط بی‌سیم