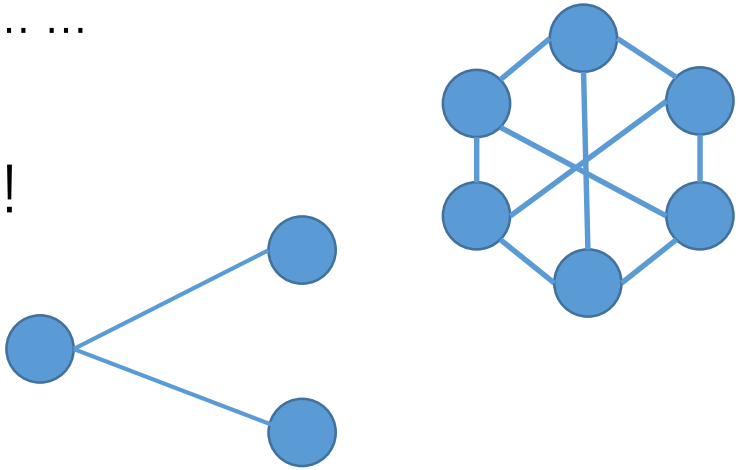
The Application Layer

The Principle of Apps

School of Software Engineering
South China University of Technology
Dr. Chunhua Chen
chunhuachen@scut.edu.cn
2016 Spring

The Ever-changing Faces of Network Usage

- We have built such complicated computer networks for **sharing of data** across the world!
- In the early age of computer network, sharing of data is needed by professors downloading scientific reports from center servers!
  - File sharing and FTP, 1971, even before TCP protocol, using NCP protocol
- But, the Network Usage has been changing it faces all the time:
  - The creator of data and people involved
  - The data format, the size, the location, the data usage … …
  - Communication/Computing models and protocols
- The network itself is also making huge progresses!
  - In its all parts, edges, core and the links



Date Back at early 1970s

- We have main servers maintained by large institutions, which can do massive computations and store huge number of documents, e.g. scientific reports.
  - What about the size of a document? Large, or small?
  - How to find the right ones? Easy ways?
- Professors work at remote offices want to use the maintain servers.
  - What about the capability computers used by these professors at their offices.
- Methods of network use:
  - direct and indirect.

Sir Tim Berners-Lee

Direct and Indirect Use of Network

- **Direct** network applications let a user access a **remote** host and use it as if it were local, creating the illusion that the network doesn't even exist (or at least, minimizing the importance of distance).
- **Indirect** network use meant getting resources from a remote host and using them on the local system, then transferring them back.
- These two methods of use became the models for the first two formalized TCP/IP networking applications: **Telnet** for direct access and the **FTP** for indirect network use.

Try Telnet & FTP Apps as house works

- Shown during lecture!

What about Today!

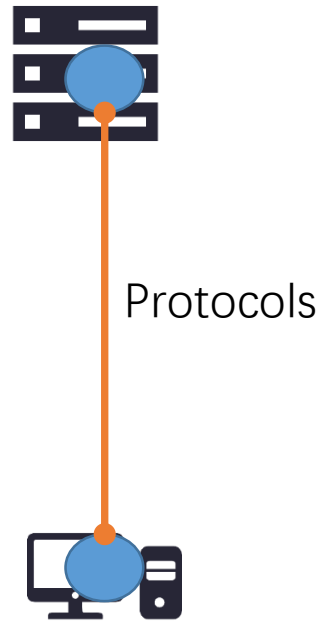
- Web is a huge success, and is a carrier for many applications, such as Search Engine, E-Commerce, online-video, … …
- **Cloud Computing**, besides enabling sharing of data, the network is becoming a computing utility!
  - What is utility means?
- Network does not just connect computers, but are gonging to connect everything!
  - The **Internet of Thing** (IoT)

amazon

YouTube

What is Network Applications?

- A network application is an app that use the network services to achieve its goals.
- It usually contains multiple parts distributing/running at separate computers;
- These distributed components communicating with each other using network messages.
- The mechanism of component interaction is also known as **protocols**, which is the kernel of the app!



What about the factors shaping the forms of Apps

in fact, its underlying protocols

- Of course, the inherent requirements, which is fundamental!
  - What about the different requirements of normal documents and web documents?
- The network hardware, the supporting infrastructure
  - the progresses of computer power and massive possession of devices by end users, and the high-speed data link, such as Fiber Optics
- And, always the economics
  - The Dot-com bubble and Cloud Computing (before this, we have Grid Computing)

The APPs and Protocols in the Textbook

- File Transfer and FTP
- The Web and HTTP
- E-mail and SMTP, … …
- Peer-to-Peer File sharing
- Domain Name Systems

- There are many others!!!

The problems faced by Sir Tim Berners-Lee

late 80s, the problem of FTPs

- Berners-Lee worked as a software engineer at **CERN**, the large particle physics laboratory near Geneva, Switzerland.
- Scientists come from all over the world to use its accelerators, but Sir Tim noticed that they were having difficulty sharing information.

> *"In those days, there was different information on different computers, but you had to log on to different computers to get at it. Also, sometimes you had to learn a different program on each computer. Often it was just easier to go and ask people when they were having coffee…",* Tim says.

@ https://www.w3.org/People/Berners-Lee/Kids.html

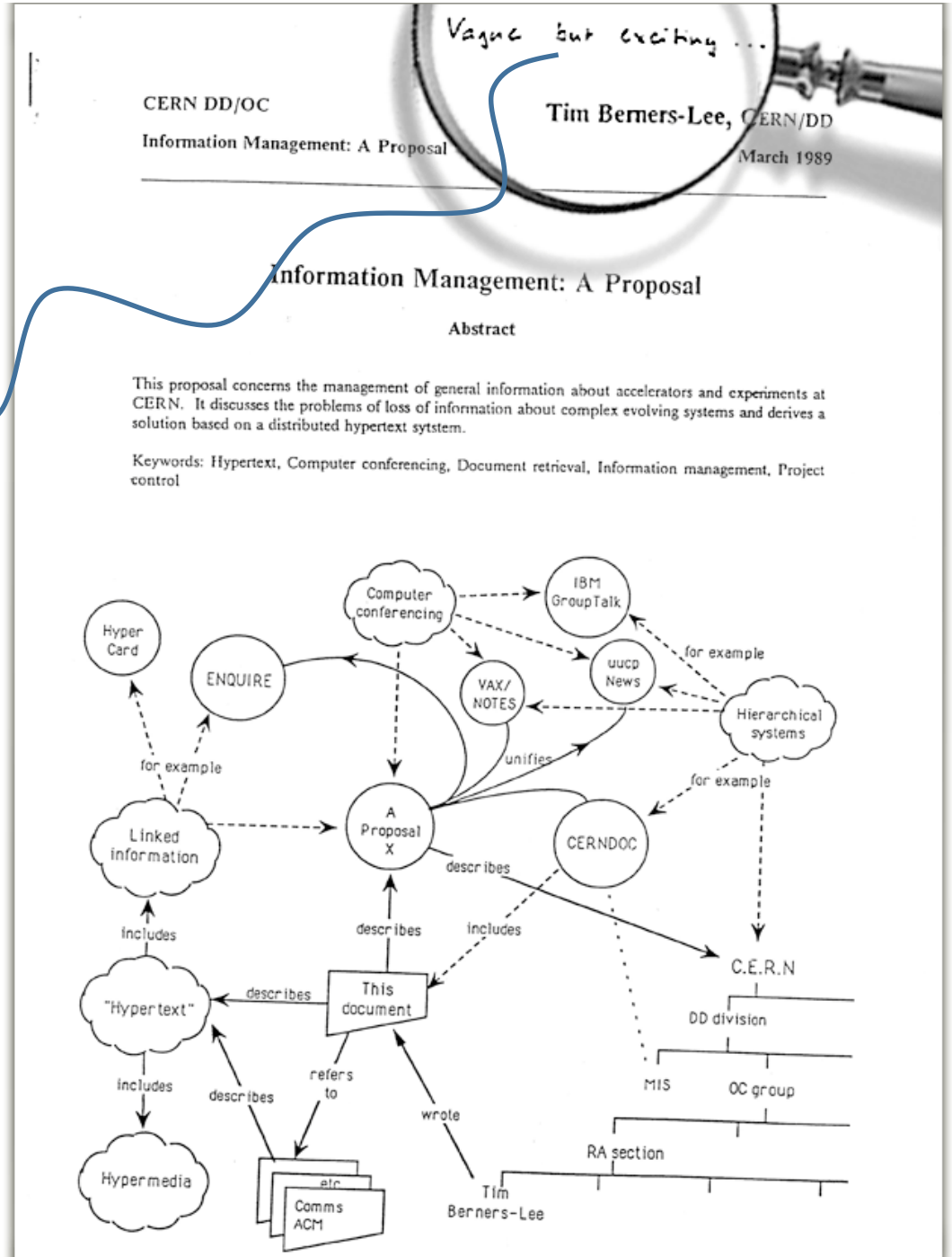
- In March 1989, Tim laid out his vision for Web, and by October of 1990, he worked it out, using a NeXT computer, one of Steve Jobs' early products.

“Information Management: A Proposal”

the paper invented Web

Vague and exciting …

- **HTML**: HyperText Markup Language. The markup (formatting) language for the Web.
- **URI**: Uniform Resource Identifier. A kind of “address” that is unique and used to identify to each resource on the Web. It is also commonly called a URL.
- **HTTP**: Hypertext Transfer Protocol. Allows for the retrieval of linked resources from across the Web.

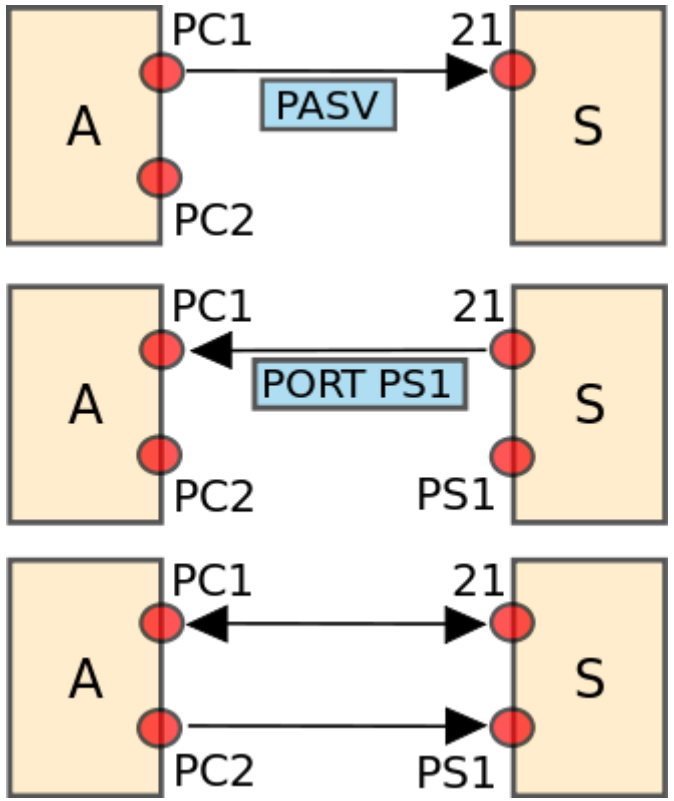


Let us have a deep foray of FTP and Web

- The different natures of FTP documents and Web document
  - FTP documents: 1971, Abhay Bhushan
    - usually normal files, such as pdfs;
    - files stored in the same server, users decides which ones to download
  - FTP: separate stateful control connection and data connection
  - Web documents/pages: 1989, Sir Tim Berners-Lee
    - Typical web pages provide **hypertext** that includes a navigation bar to other web pages via **hyperlinks**, often referred to as links. (HTML)
    - Web browsers coordinate the various web **resource elements** for the written web page, such as style sheets, scripts, and images, to present the web page.
    - Resource elements contained in one web page might not be stored in the same server.
  - stateless and multiplexes control and data over a single connection

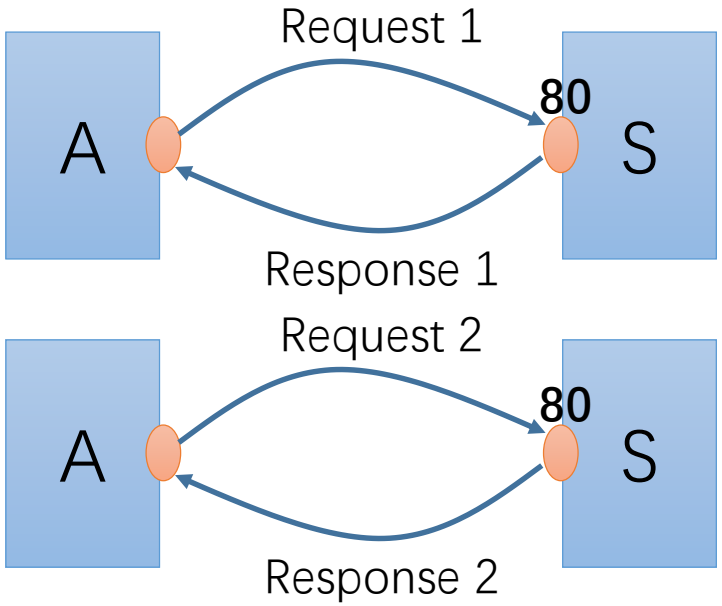
FTP and Web/HTTP

FTP



A stateful control connection
Data transfer connections

HTTP



More @
https://en.wikipedia.org/wiki/File_Transfer_Protocol#Differences_from_HTTP

Some Other Revolutionary Ideas

big thinks behind the Web

- The early Web community produced some revolutionary ideas that are now spreading far beyond the technology sector:
  - **Decentralisation**: No permission is needed from a central authority to post anything on the Web, there is no central controlling node, and so no single point of failure ··· and no “kill switch”! This also implies freedom from indiscriminate censorship and surveillance. (无中心)
  - **Non-discrimination**: If I pay to connect to the internet with a certain quality of service, and you pay to connect with that or a greater quality of service, then we can both communicate at the same level. This principle of equity is also known as Net Neutrality. (网络平等)

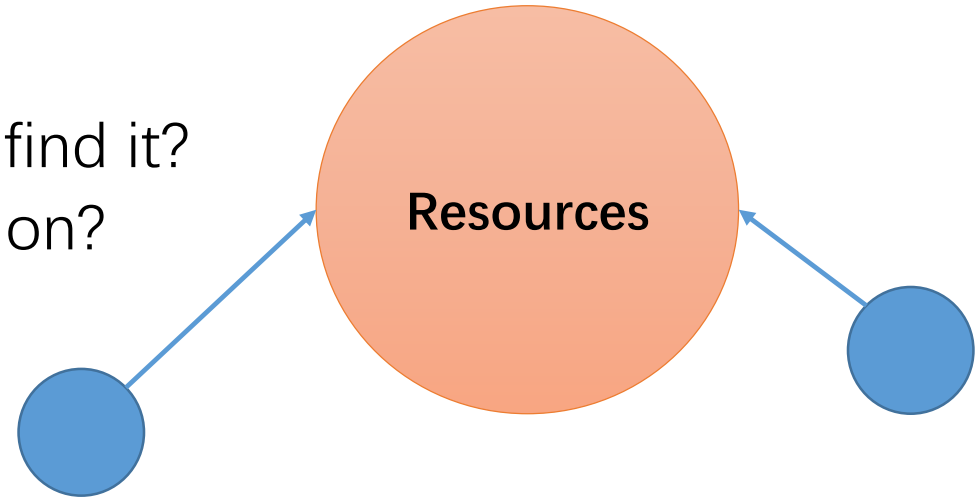
Some Other Revolutionary Ideas

big thinks behind the Web

- The early Web community produced some revolutionary ideas that are now spreading far beyond the technology sector:
  - **Universality**: For anyone to be able to publish anything on the Web, all the computers involved have to speak the same languages to each other, no matter what different hardware people are using; where they live; or what cultural and political beliefs they have. In this way, the Web breaks down silos while still allowing diversity to flourish. (普遍性)
  - **Consensus**: For universal standards to work, everyone had to agree to use them. Tim and others achieved this consensus by giving everyone a say in creating the standards, through a transparent, participatory process at W3C. (一致性)

The principles of Apps

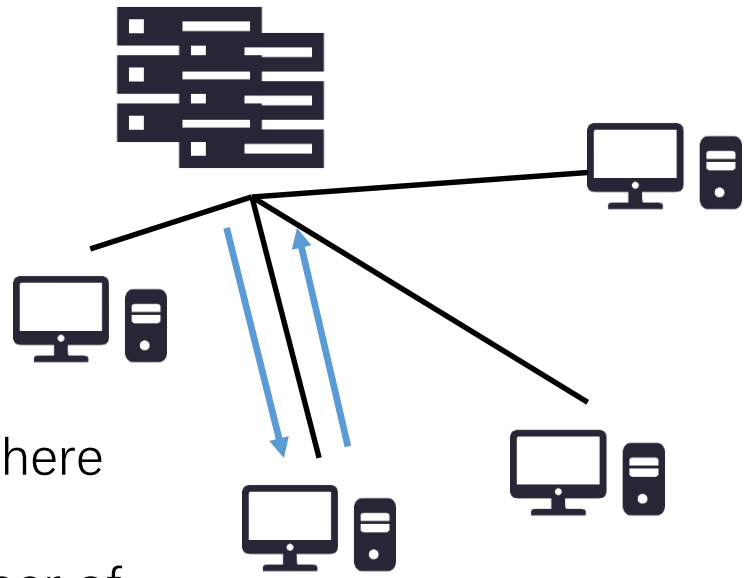
- The very first principle is applications use services provided by Transport Layer!
  - For data delivery/transfer services
- But, before that lets look at a principle which is irrelevant to transport services.
- Questions?
  - Where is the resources? How can you find it?
  - Who should initialize the communication?



The Application Architecture

- Architecture:
  - The scopes, how many nodes, where are they?
  - How these nodes interact with each other?
    - Directly communication with any of two, or
    - There is a center proxy?
  - Does these nodes are all equal or not? In term of capability! (CPU, Main Memory, NIC, …)
- Client-Server model, Peer-to-Peer (P2P) model and hybrid model

The Client-Server Model



- Nodes are Asymmetric, two types:
  - Servers:
    - High capabilities, containing the resources
    - Always on, hence can be found anytime, from any where (Well-known Fixed Global IPs)
    - Server farm in a data center for handling huge number of requests
  - Clients:
    - Use servers on demand, occasionally on
    - Clients do not communication with other clients directly
- Apps: Web ~~ Google, Amazon, Facebook
- What is Browser-Server model, B/S model?

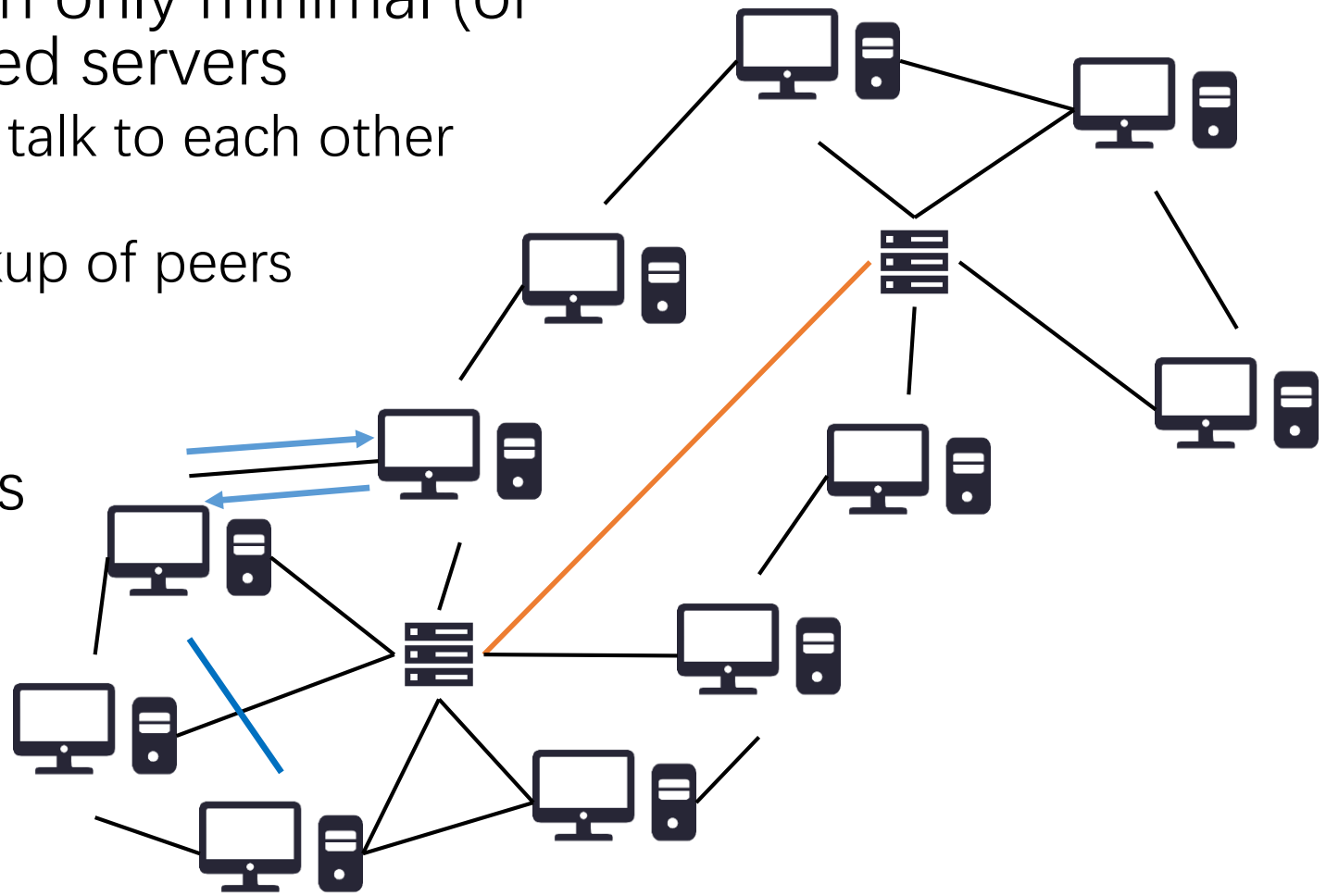
Downlink and Uplink speed

The P2P Model

- All nodes are equal, with only minimal (or no) reliance on dedicated servers
  - Nodes are called Peers, talk to each other directly
  - Servers needed for lookup of peers
  - Self-scalability, why?

- All peers have resources to offer.

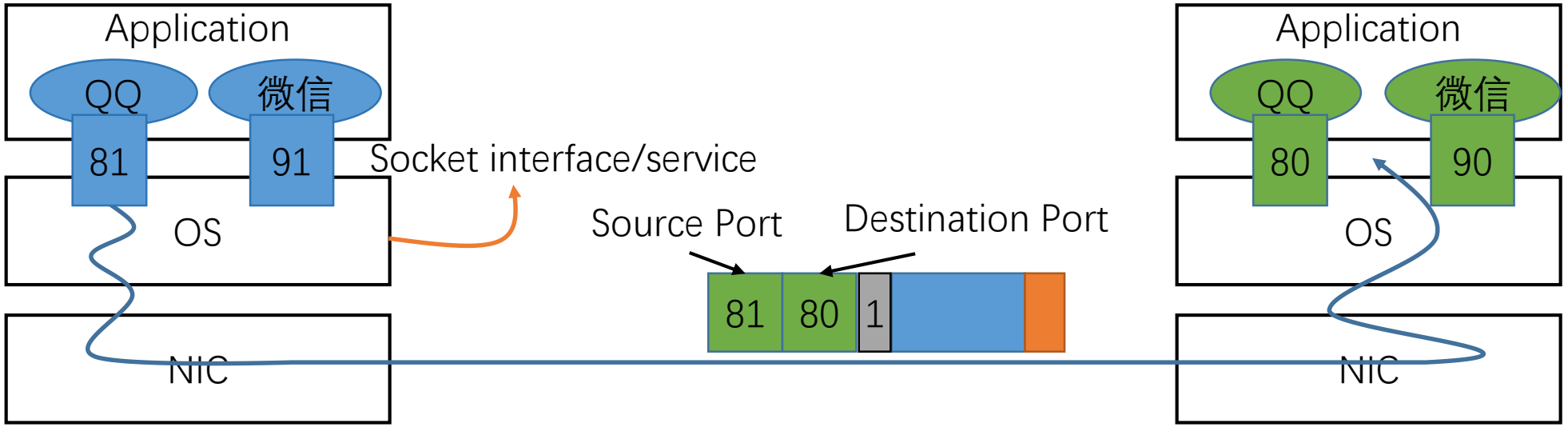
Downlink and Uplink speed in this case!



How Apps use the Network Services

the process communication model

- App are run as processes in OS
- App can fork multiple processes; a process can use multiple Sockets.
- Port number is used to address a pair of communicating processes.



What Transport Services can be provided?

- Measure service in terms of:
  - Reliability:
    - reliable (what A send what B get) vs loss-sensitive
    - unreliable (loss of data) vs loss-tolerant
  - Throughput:
    - guaranteed throughput (at least *r* bps) vs bandwidth-sensitive
    - unguaranteed throughput (fluctuate with time) vs elastic applications
  - Timing:
    - timing guarantees (no more than 100 msec delivery) vs interactive real-time applications

Requirements of selected network applications

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
Email	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio /video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Transport Services Provided by the Internet

TCP and UDP Services of Internet

- The Internet (and, more generally, TCP/IP networks) makes two transport protocols available to applications, UDP and TCP.
- TCP services:
  - Connection-oriented service; handshake for TCP Connection
  - Reliable data transfer service
- UDP Services (the no-frills, Best Effort Service)
  - Connectionless
  - Unreliable data transfer service
- Throughput or Timing guarantees—services not provided by Internet transport protocols!!

Popular Internet applications and its Protocol

Application	Application Layer Protocol	Underlying Transport Protocol
Email	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary UDP or TCP (e.g., Skype)	TCP

Application Layer Protocols

- In particular, an application-layer protocol defines:
  - The **types of messages** exchanged, for example, request messages and response messages
  - The **syntax** of the various message types, such as the fields in the message and how the fields are delineated
  - The **semantics** of the fields, that is, the meaning of the information in the fields
  - Rules for determining when and how a process sends messages and responds to messages (**actions**)