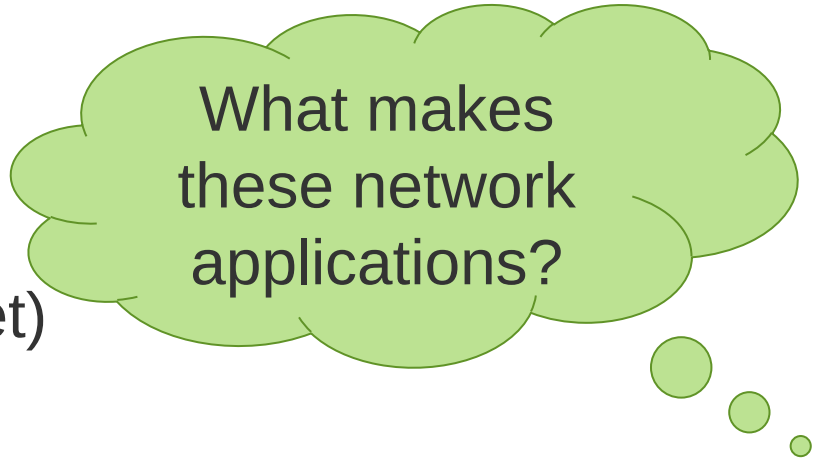
Network Applications

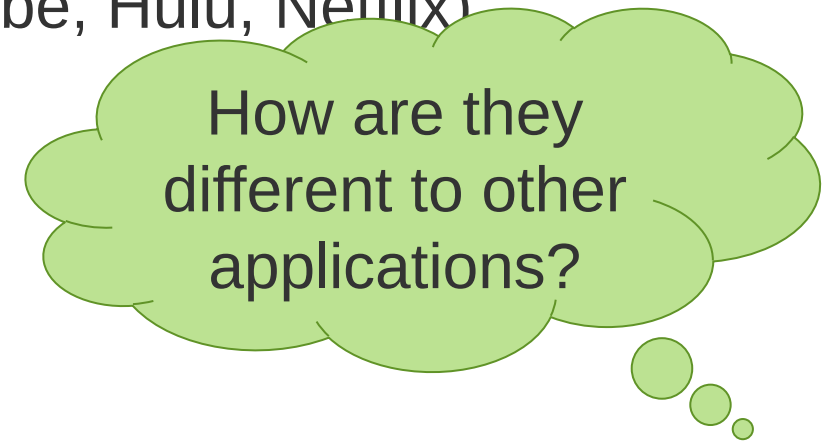
Andy Carpenter

(Andy.Carpenter@manchester.ac.uk)

Example Network Applications

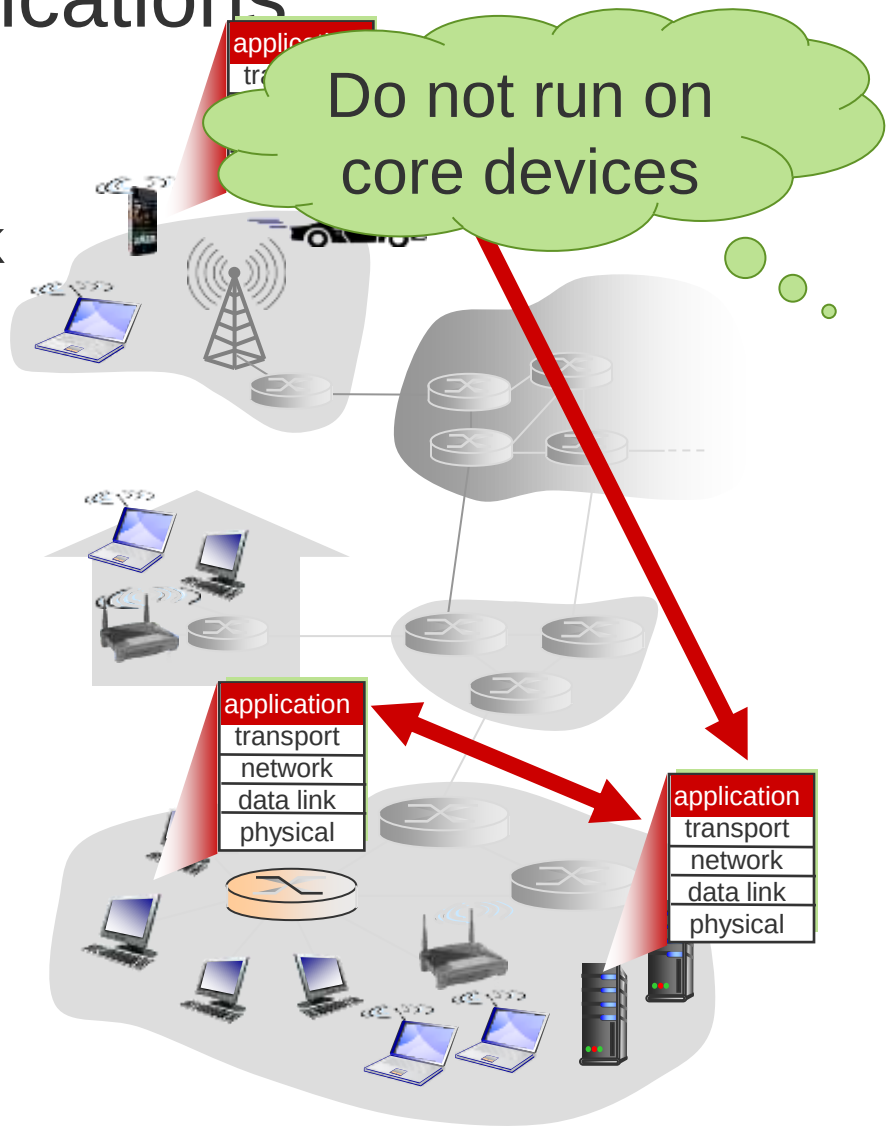
- e-mail
- Web
- Instant messaging
- Remote login (ssh and Telnet)
- P2P file sharing
- Multi-user network games
- Streaming stored video (YouTube, Hulu, Netflix)
- Voice over IP (e.g. Skype)
- Real-time video conferencing
- Social networking



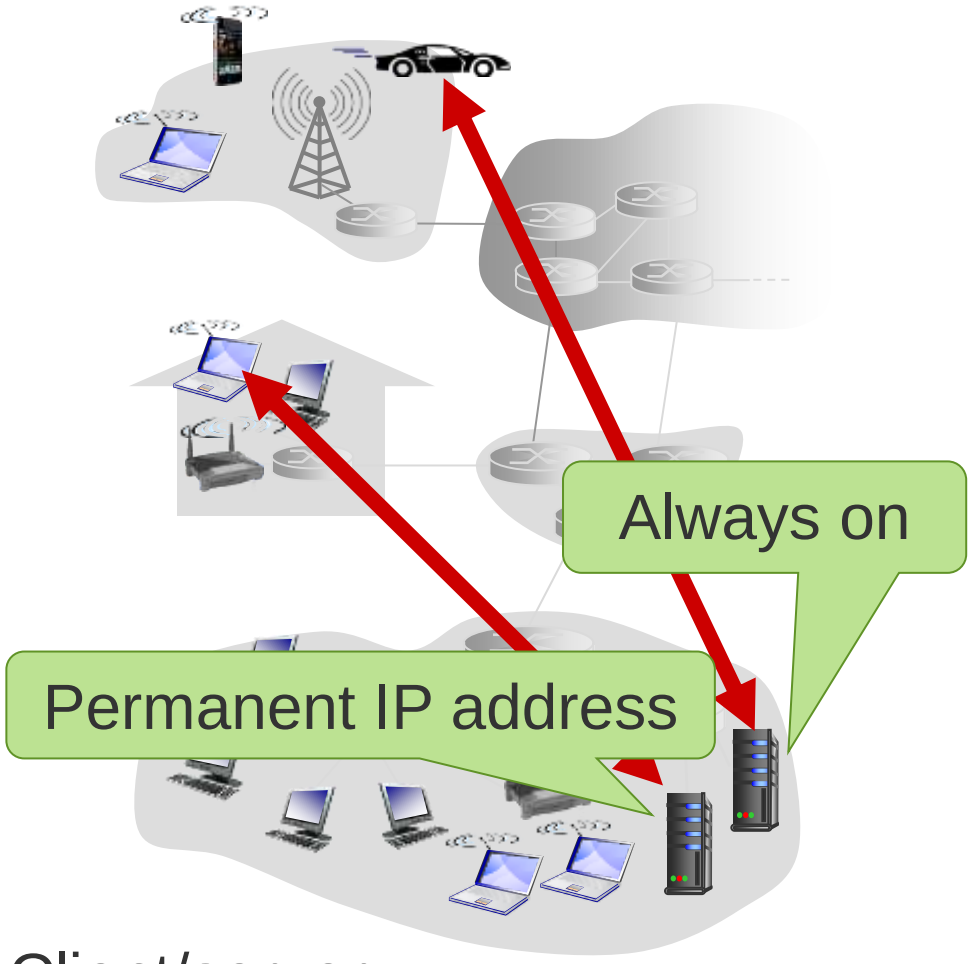


Network Applications

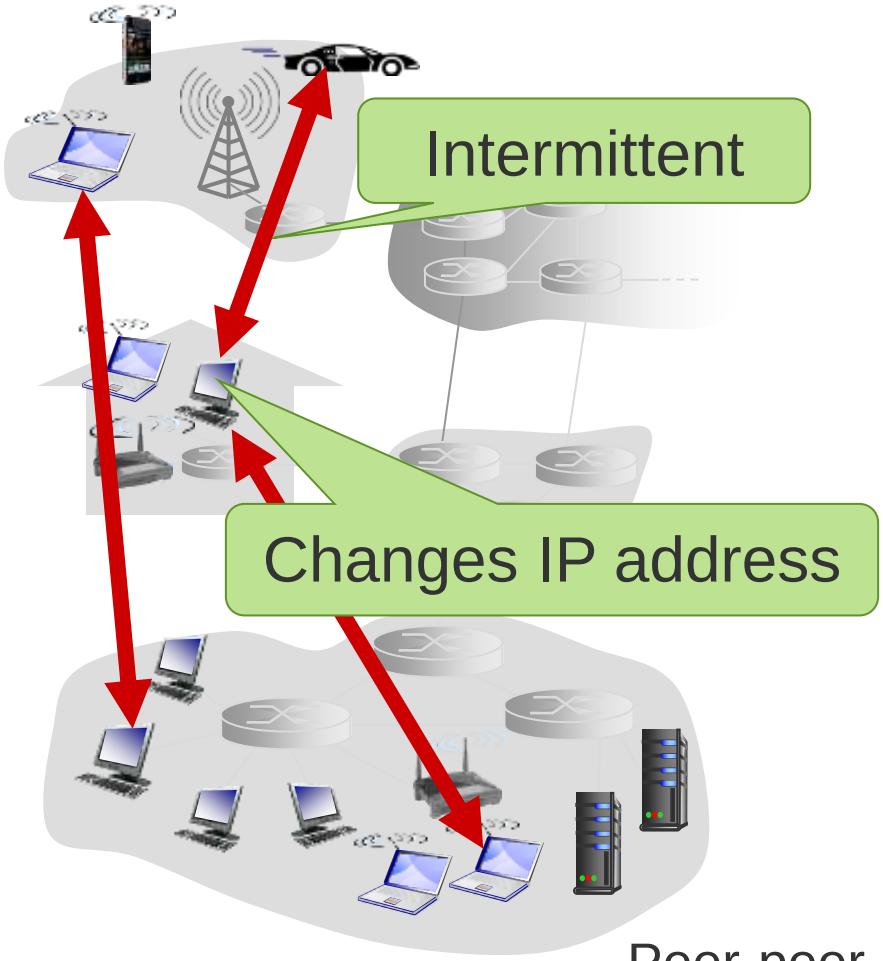
- Programs that:
  - communicate over network
  - run on end systems
- Issues:
  - architecture
  - QoS
  - protocols, addressing
  - understanding data
  - control vs. data
  - extensibility, scalability
  - buffering, state



Architecture: Options

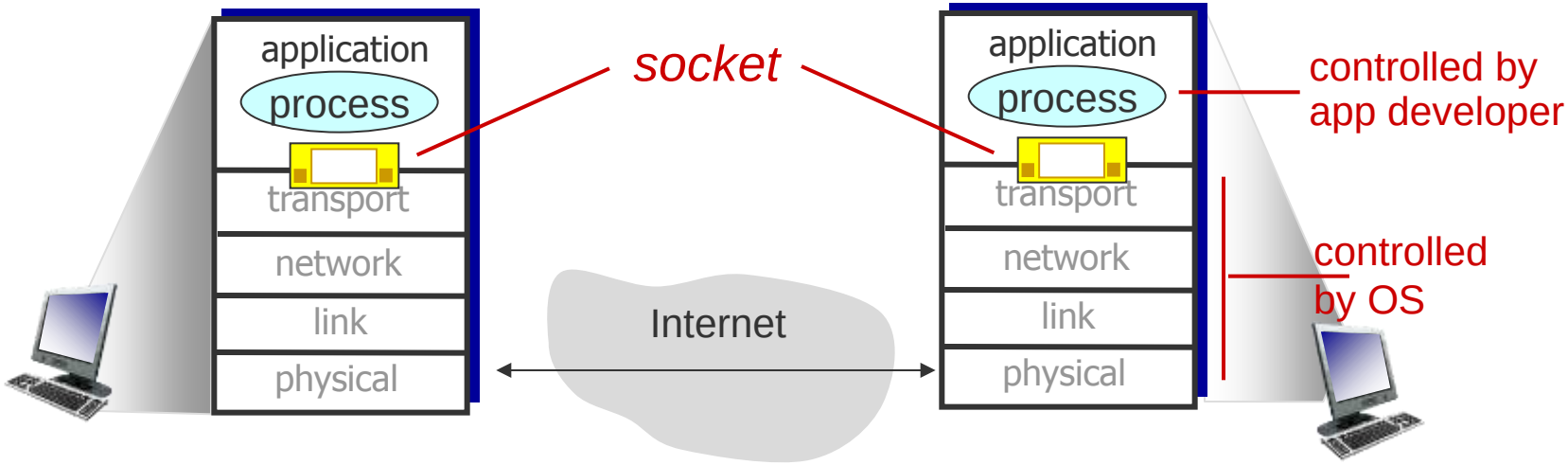


Client/server



Peer-peer

Architecture: End-points



- Application end-point is a process
- Communicate by exchanging messages
- Messages sent/received via socket

Application QoS: (Some) Params

Data loss

- some apps (e.g., audio) can tolerate some loss
- other apps (e.g., file transfer, telnet) require 100% reliable data transfer

Timing

- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

Throughput

- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

Security

- Encryption, data integrity, …

Application QoS: Requirements

Application	Data loss	Throughput	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
web	no loss	elastic	no
real-time audio/video	loss-tolerant	Audio: 5kbps-1Mbps Video: 10kbps-5Mbps	yes, 100's msec
stored audio/video	loss-tolerant	Same as above	yes, few secs
interactive games	loss-tolerant	few kbps upwards	yes, 100's msec
instant messaging	no loss	elastic	yes and no

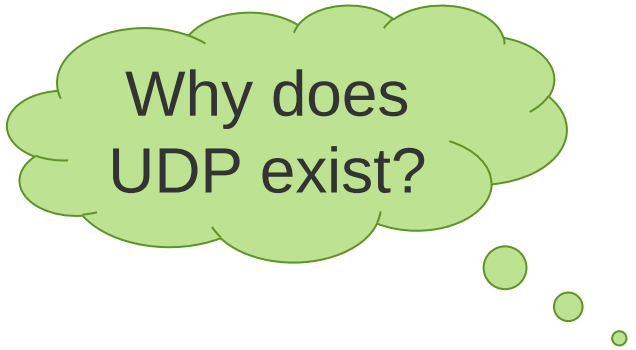
Internet Transport Service Models

TCP service:

- *connection-oriented:* setup required between client and server processes
- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum throughput guarantees, security

UDP service:

- unreliable data transfer between sending and receiving process
- does not provide: connection setup, reliability, flow control, congestion control, timing, throughput guarantee, or security



Application Protocols

- Application protocols enhance transport service model to precise communication service needs of application
- Define:
  - types of message exchanged; e.g. request
  - message syntax;
    - fields present and their delineation
  - message semantics; meaning of fields
  - message exchange rules
- Ways protocols are defined:
  - RFCs, open-standards allowing interoperability
  - proprietary implementations, e.g. Skype

Application Protocol Examples

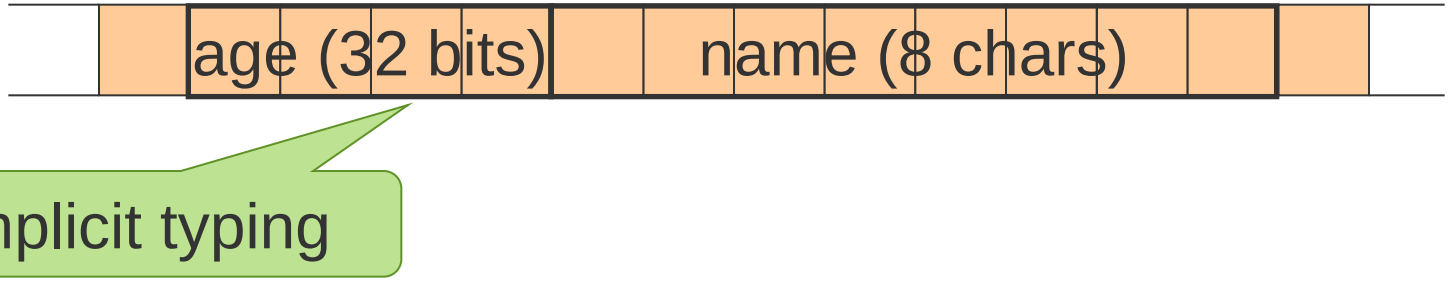
Application	Application layer protocol	Transport layer protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
web	HTTP [RFC2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g. Youtube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP proprietary (e.g. Skype)	Usually UDP

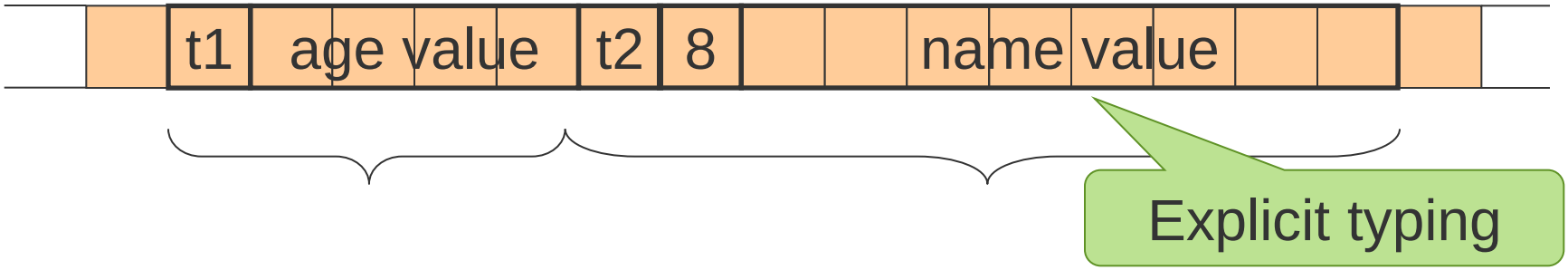
Why UDP?

Application Data

- What does this decimal byte sequence mean?
  - 72 101 108 108 111 32 99 108 97 115 115 32
- Application source and destination must:
  - each make same interpretation
- Also want efficient transmission (encoding) of data
- Compression minimises size on cable; not considered further
- Issues:
  - type and meaning of data (understanding)
  - representation for data in transit
  - when presentation encoding/decoding performed

Data: Implicit/Explicit Typing



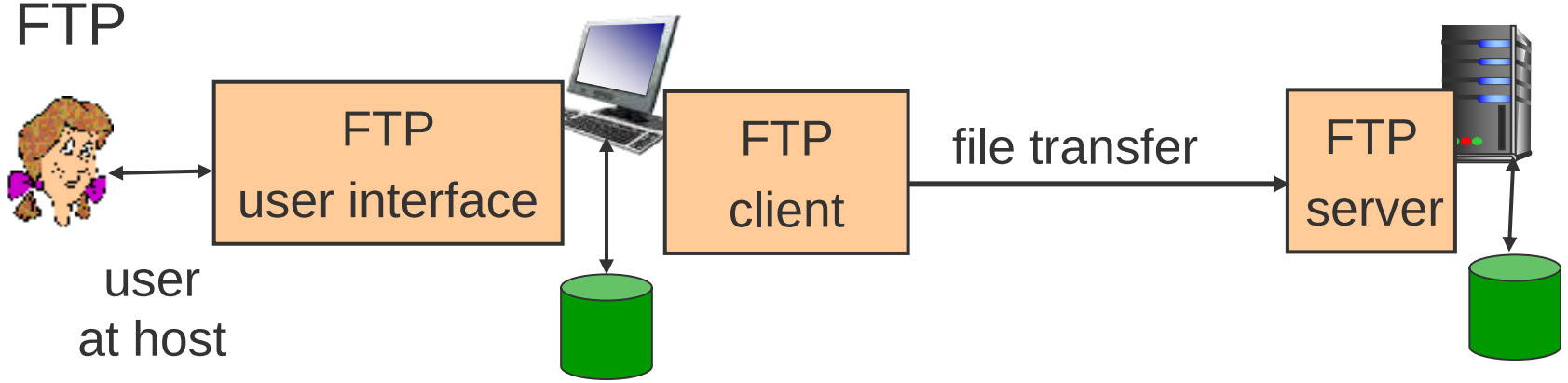


Data: Need for Conversion 1

- Telnet: Remote login Application

  - Issue: End-of-line representation: CR, LF or CR LF?

- FTP



  - Issue: text file End-of-line representation

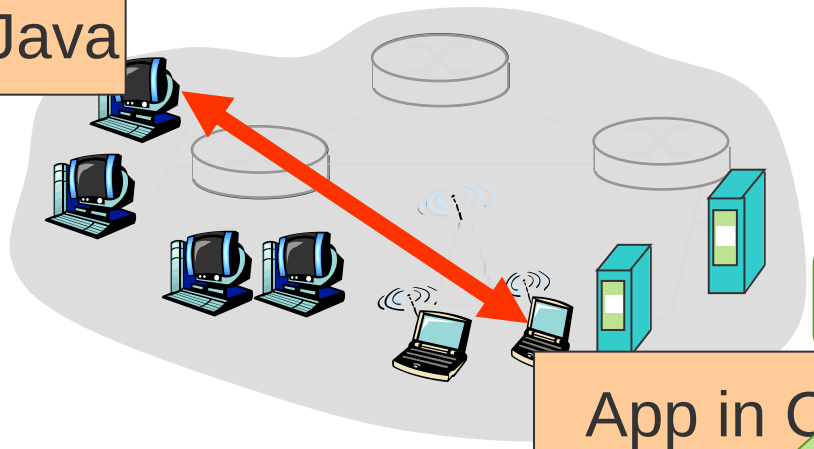
Data: Need for Conversion 2

App in Java

Age: (32-bit int)

Name: (chars)

Addr: (ref)



App in C

Age: (16-bit int)

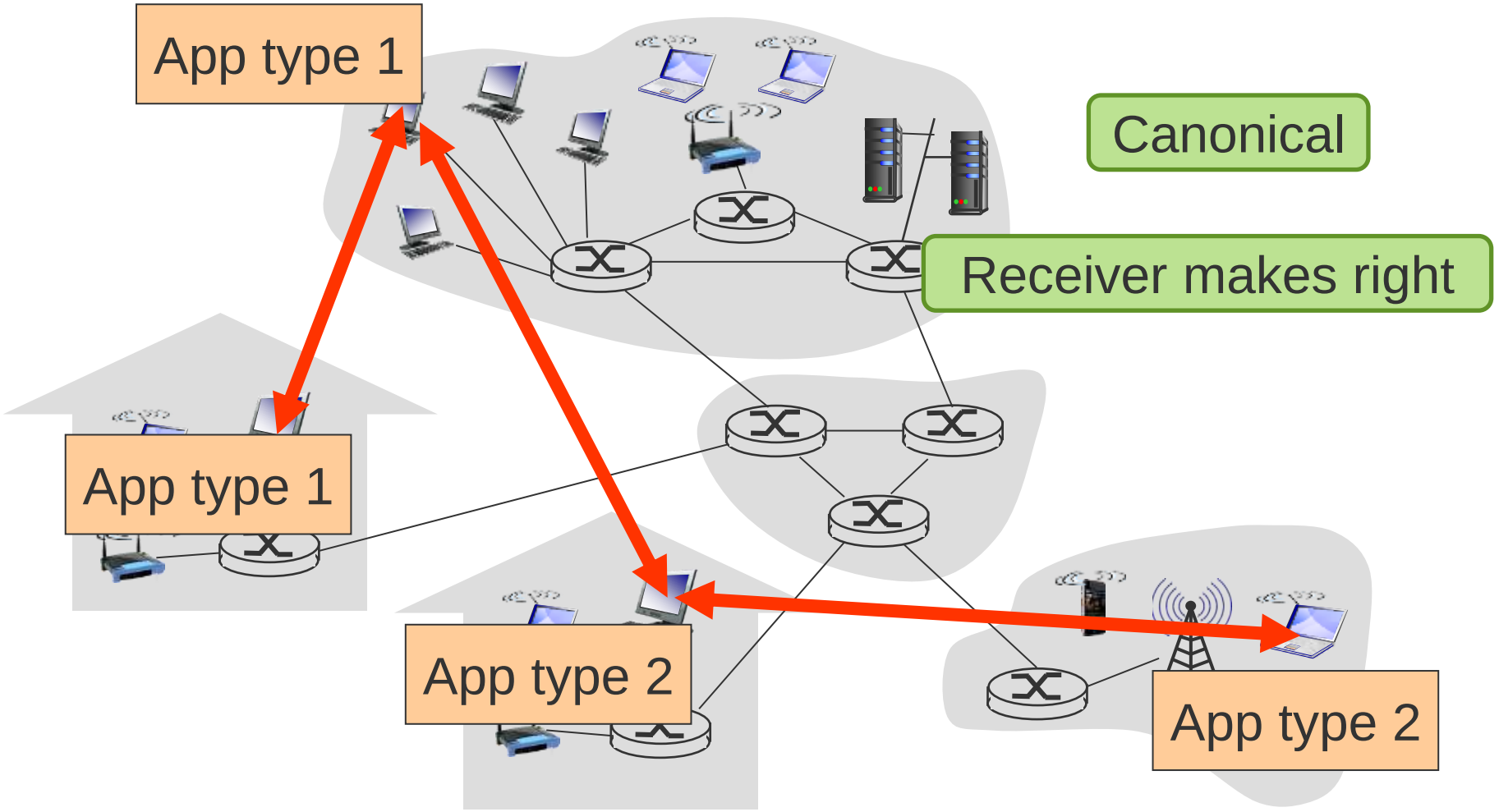
Name: (chars)

0 <= age <= 150

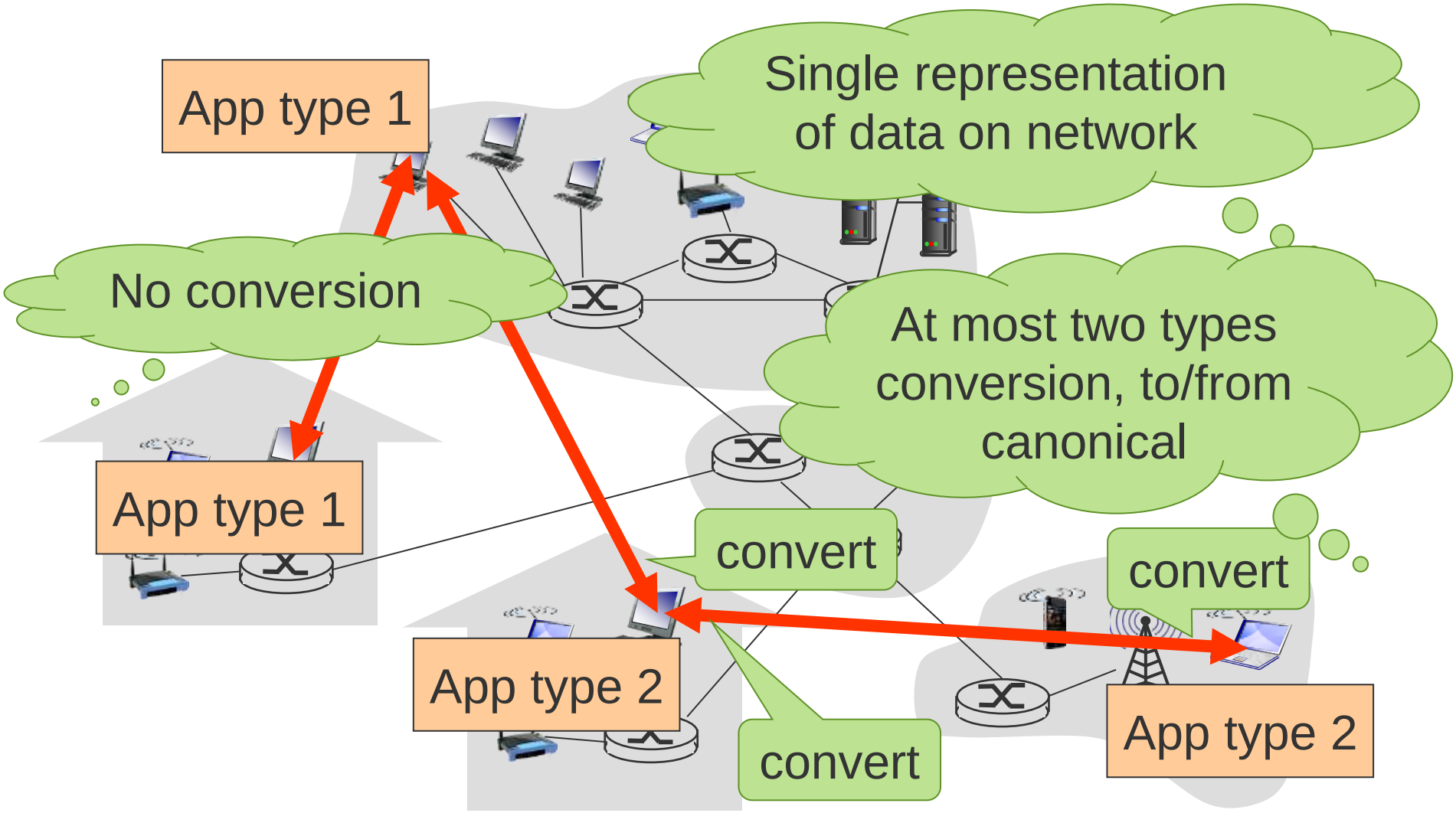
What representation is transmitted?

How many characters are sent?

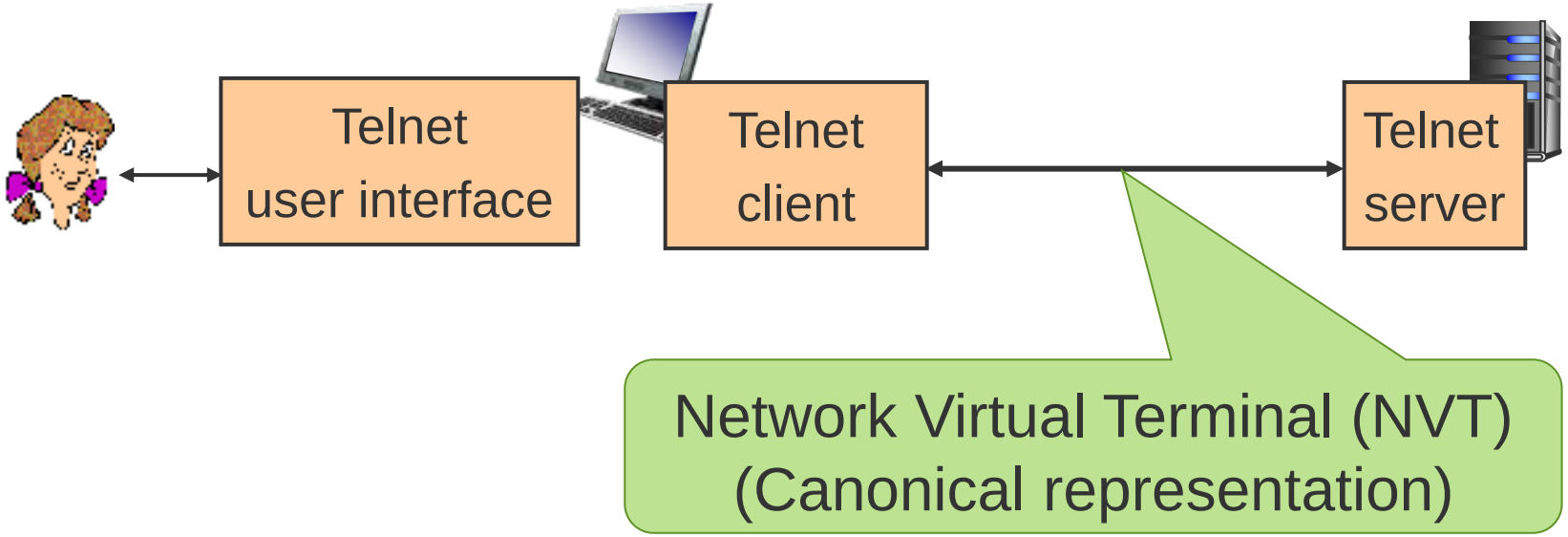
Data: Conversion Strategies



Data: Conversion – Canonical



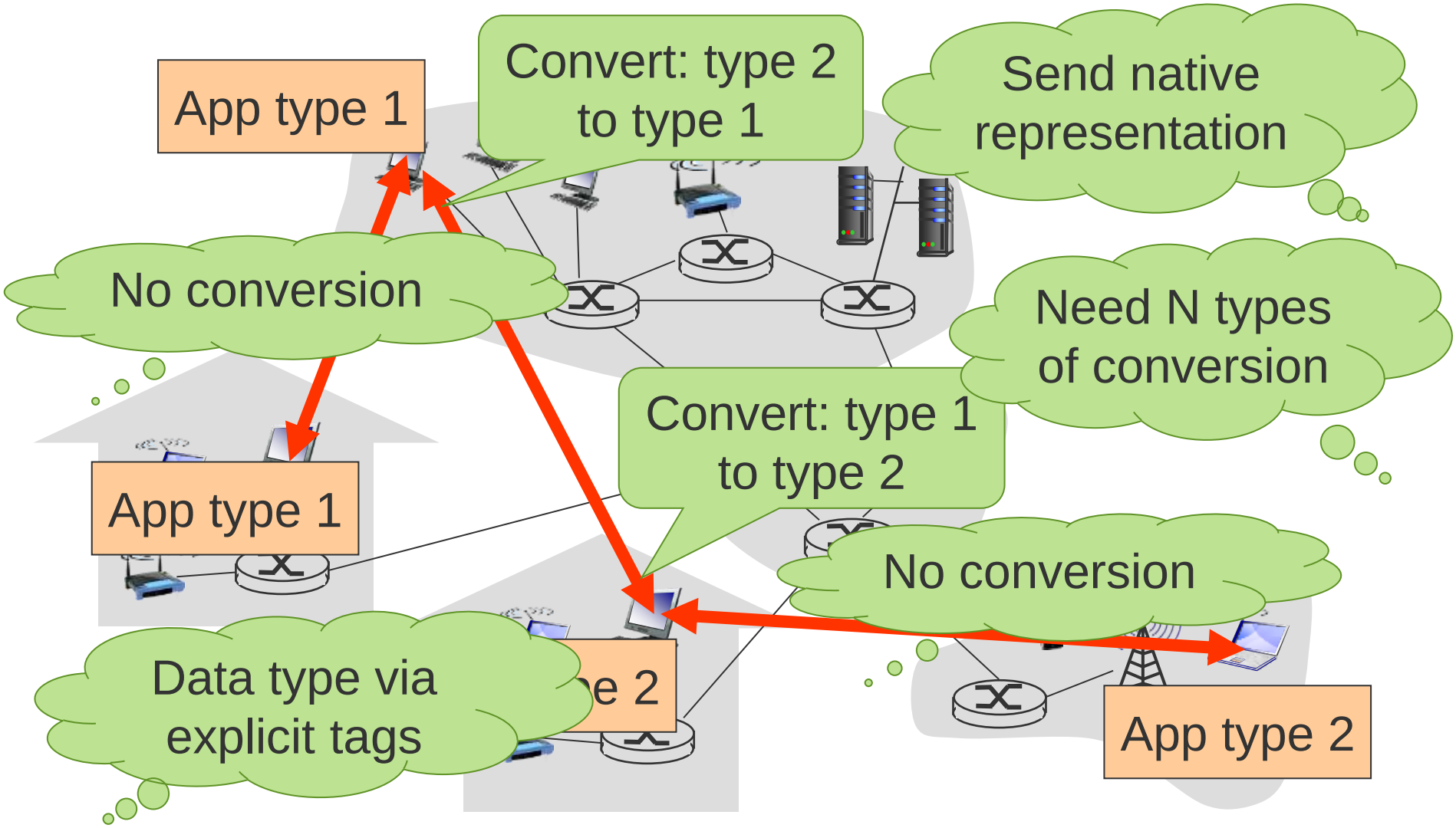
Data: Canonical - Example



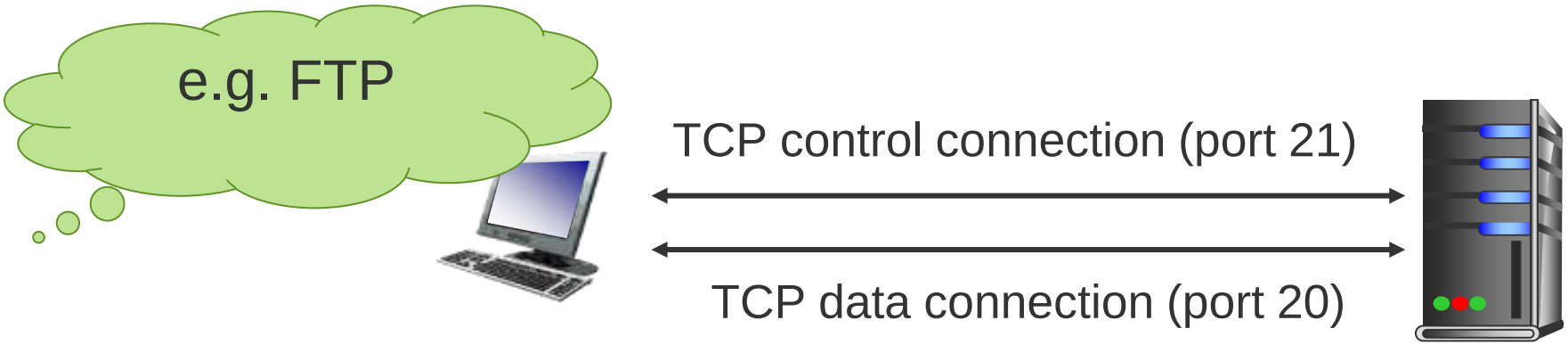
End-of-line
always CR LF

Change functionality
using options

Data: Conversion – Make Right

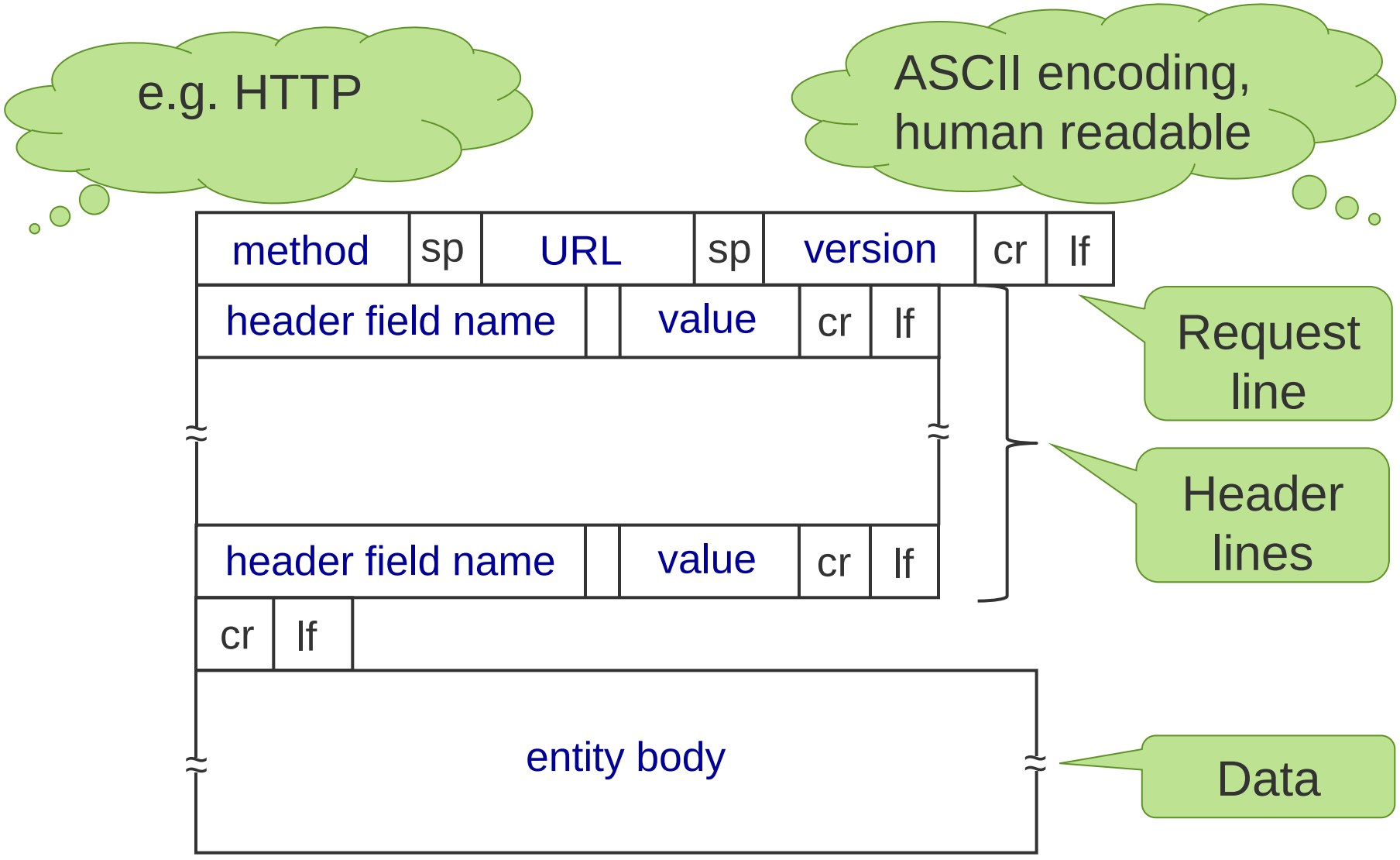


Control/Data: Separate

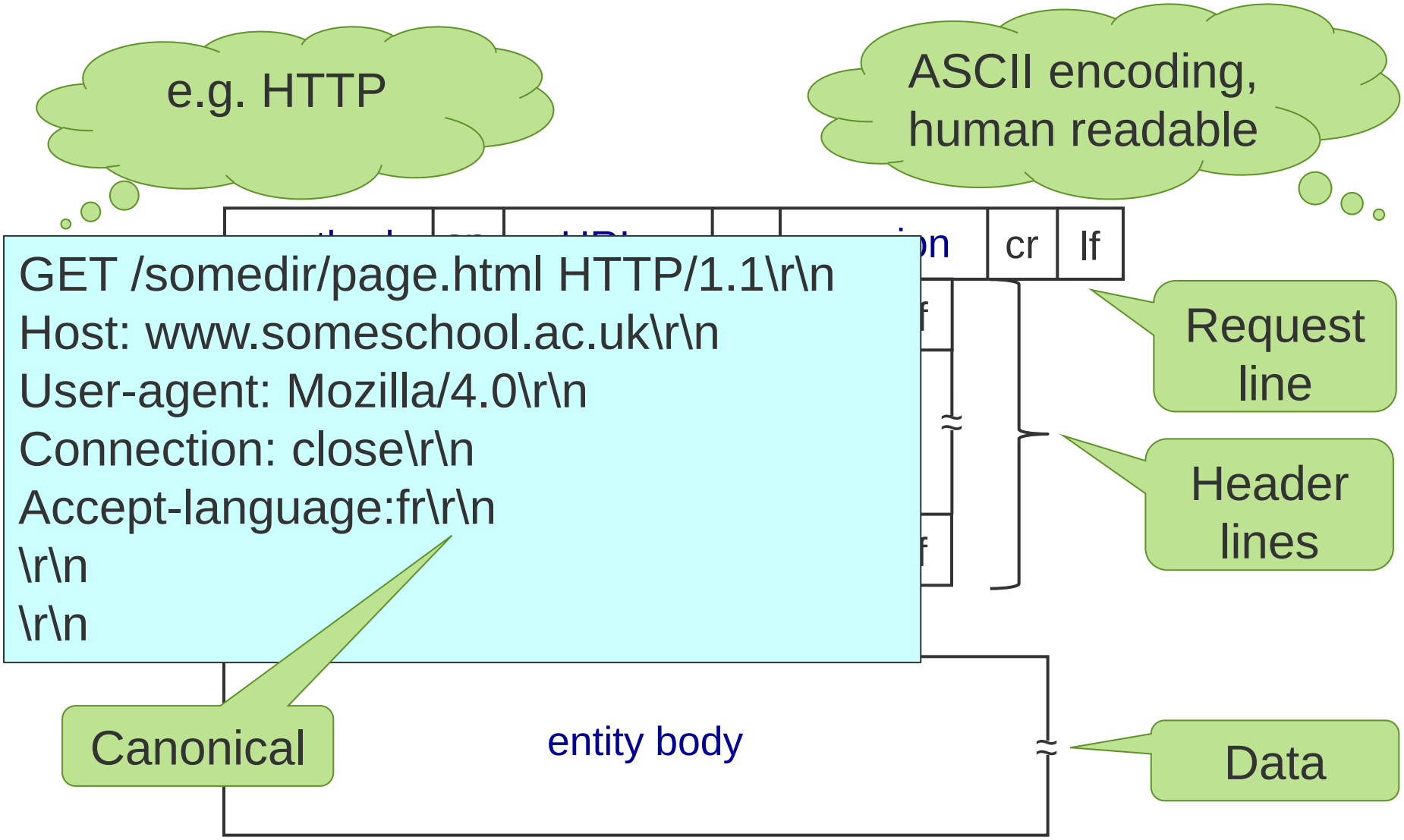


- FTP client contacts FTP server at port TCP 21
- Client authorized over control connection
- When server receives file transfer command
  - server opens 2nd TCP connection (for file) to client
- After one transfer, server closes data connection.

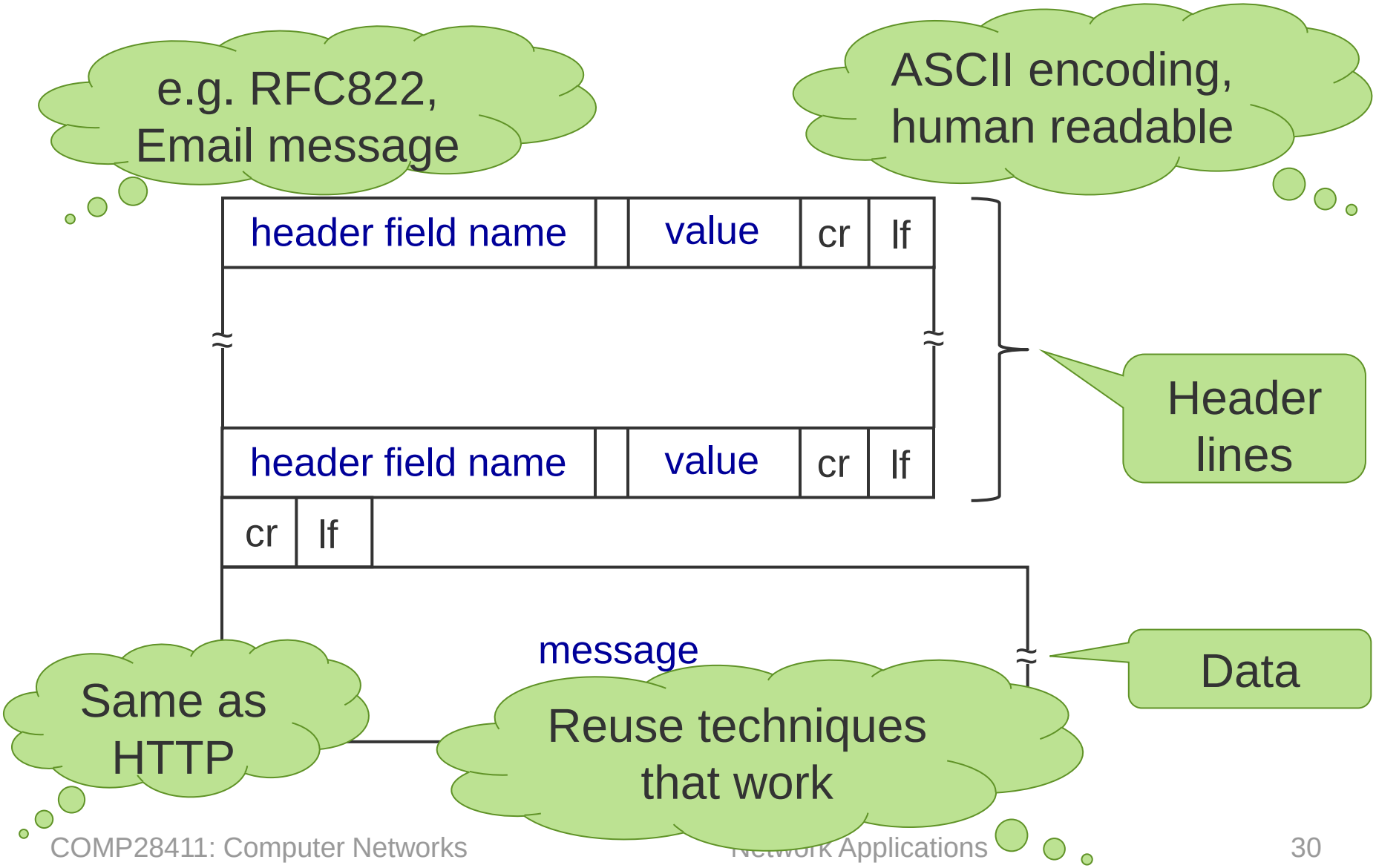
Control/Data: Data in Control 1



Control/Data: Data in Control 1



Control/Data: Data in Control 2



Control/Data: Control in Data

e.g. Telnet, Remove login

Source

Data	Data

Flag command coming

Send: Source + Command

Data	IAC	Cmd	Data

Command

Interpret as command

What if IAC in data?

Source

Data	IAC	Data

Duplicate

Send

Data	IAC	IAC	Data

Same as \ in Java strings

Summary

- Begun to look at applications
- What is important is underlying principles
  - not how a particular application works
- Demands placed on underlying network infrastructure
- Architecture: p2p or client-server
- Understanding data
- Relationship of data and control
- More next time