Network Fundamentals

http://c2.touta.in/?page_id=513 for discussion and exercises

Laurent Toutain

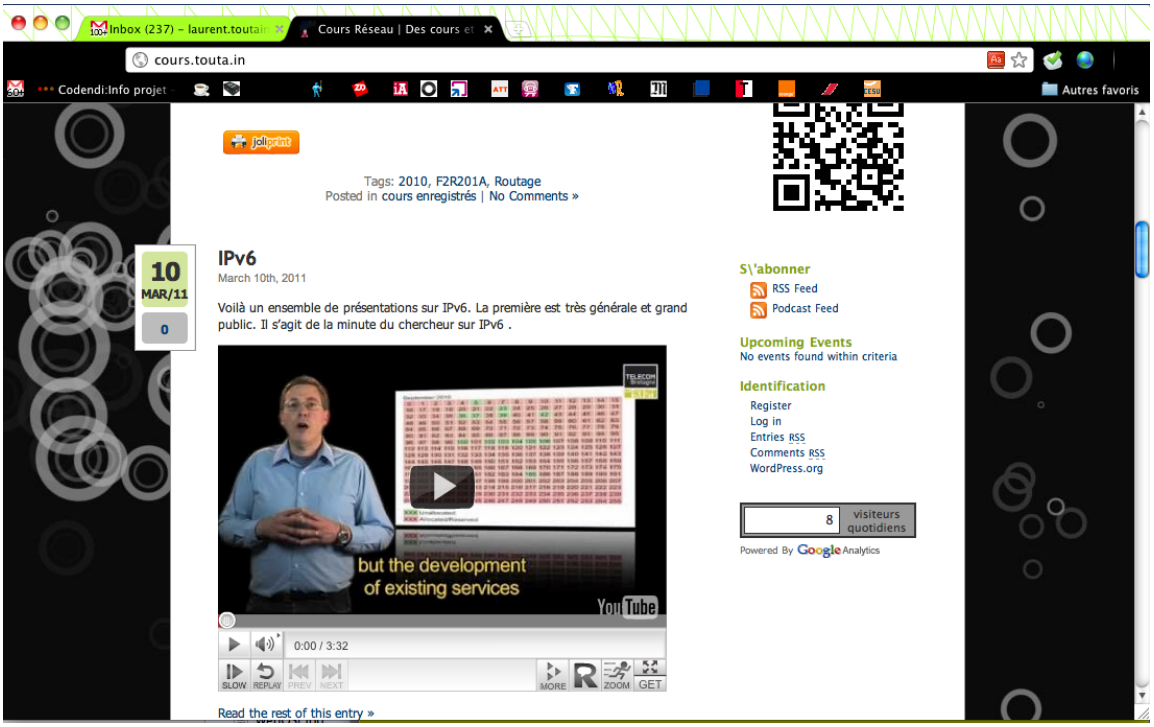
August 26, 2011

Table of Contents

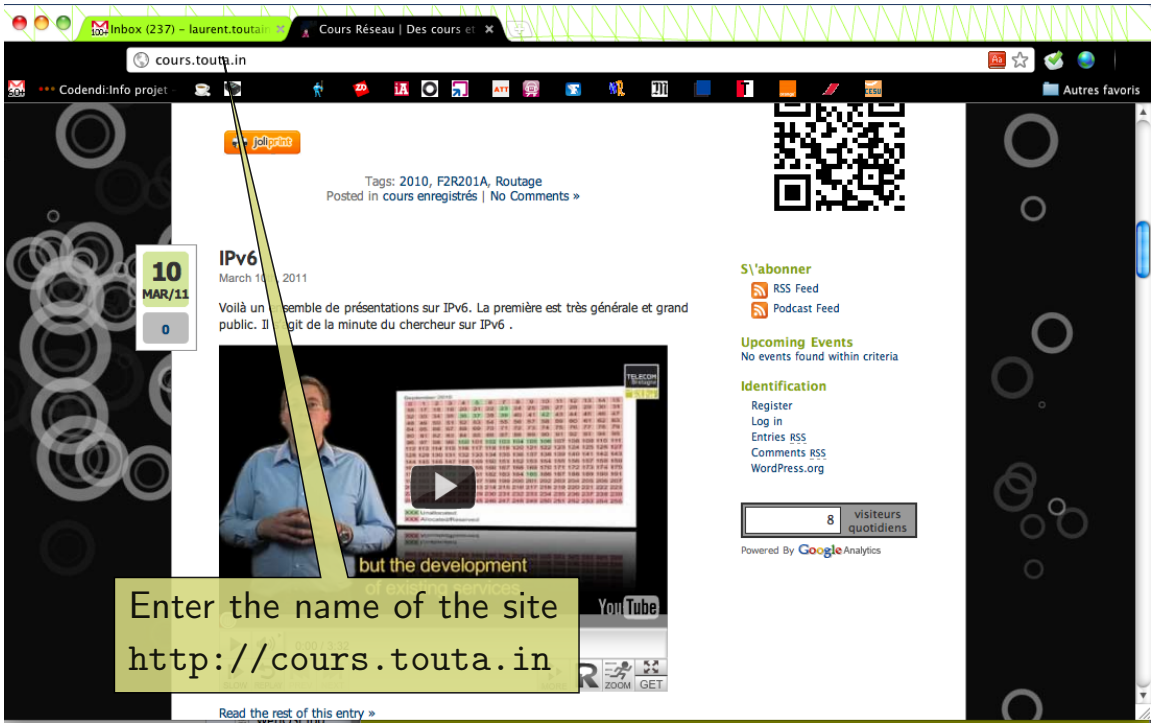
Introduction

Basic architecture

How to get this?



How to get this?

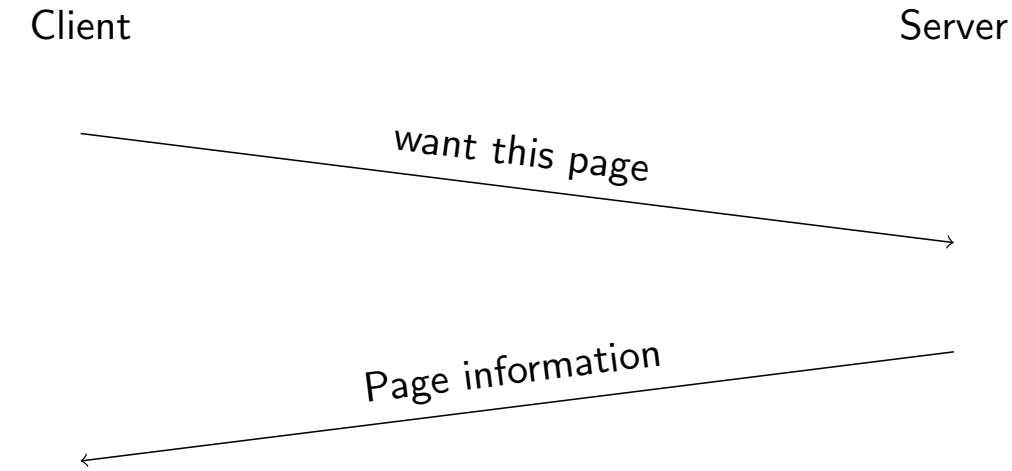


In more details

Client Server

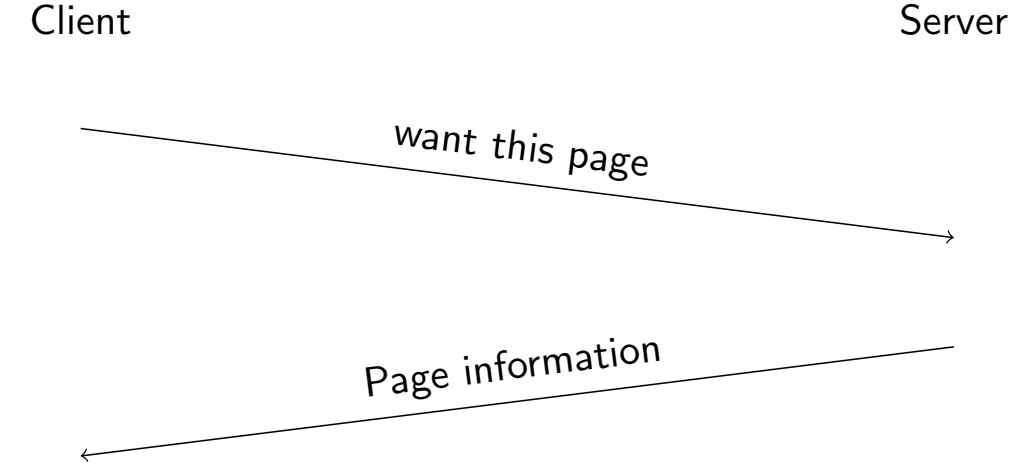
want this page

In more details



In more details

Introduction ▶ Basic architecture



Slide 5 Laurent Toutain RES 1

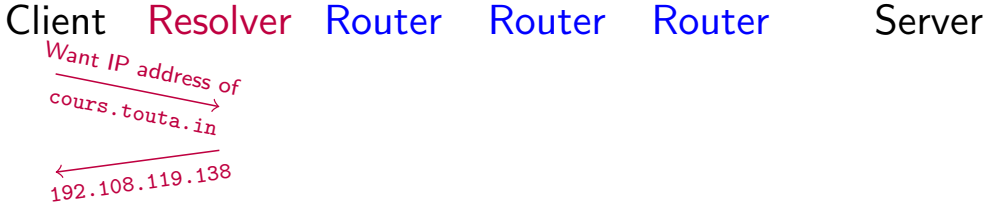
In more, more details

In more, more details

Introduction ▶ Basic architecture

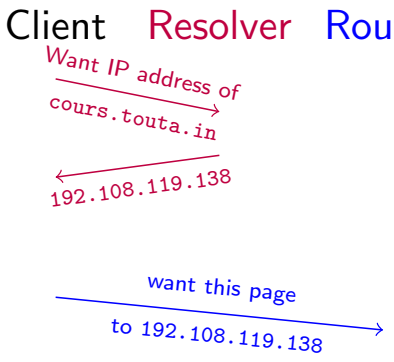
Slide 6 Laurent Toutain RES 1

In more, more details

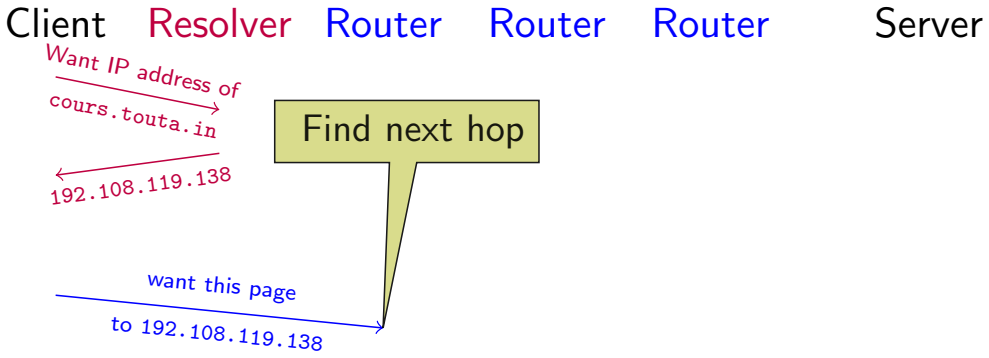


In more, more details

Introduction ▶ Basic architecture

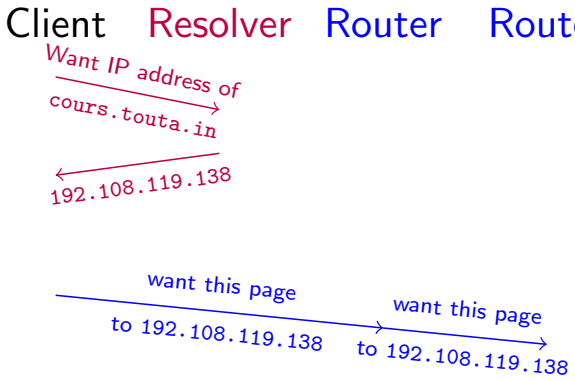


In more, more details

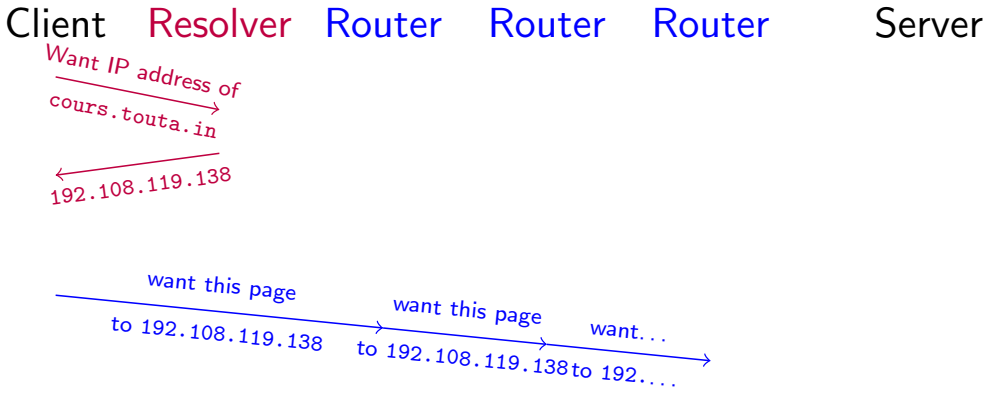


In more, more details

Introduction ▶ Basic architecture

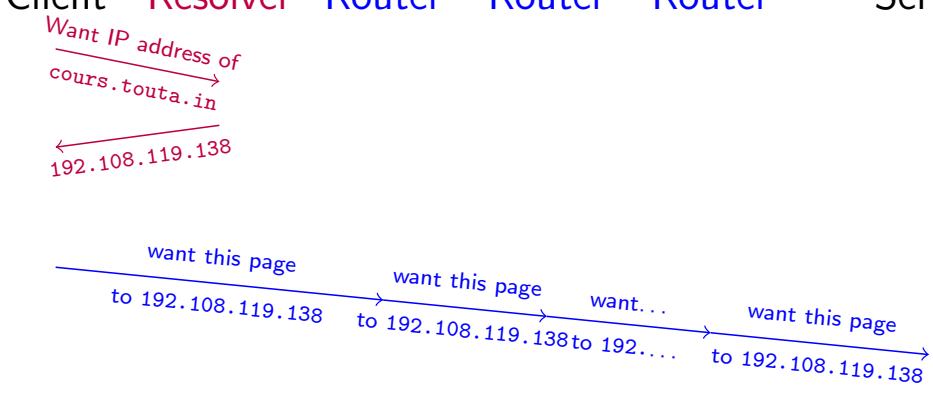


In more, more details

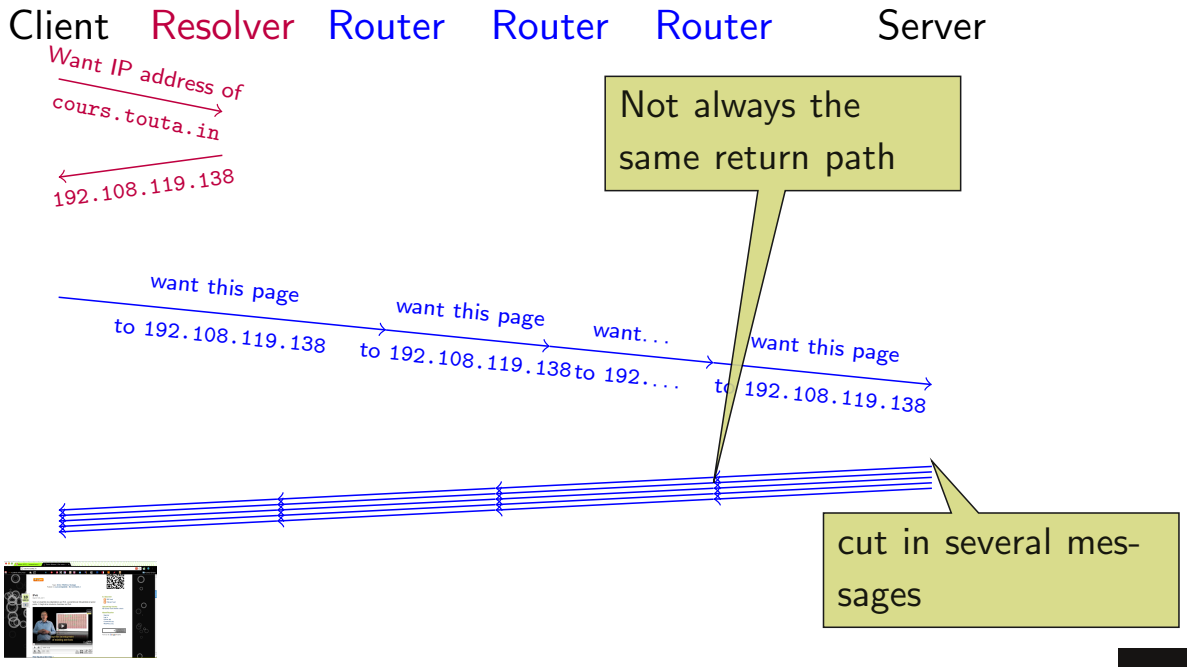


In more, more details

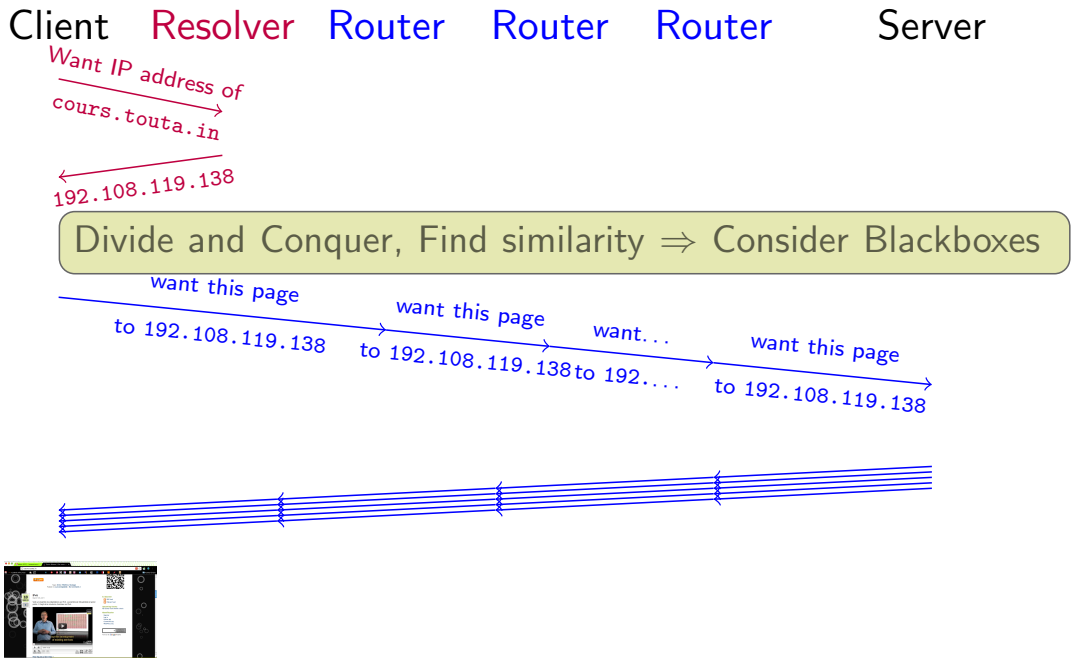
Introduction ▶ Basic architecture



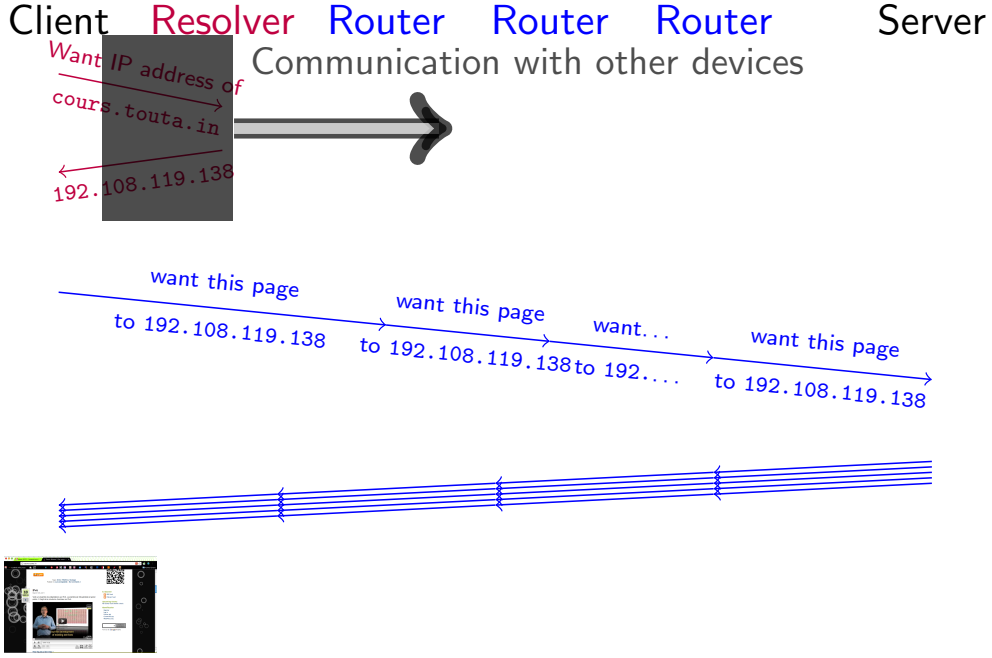
In more, more details



Main principles

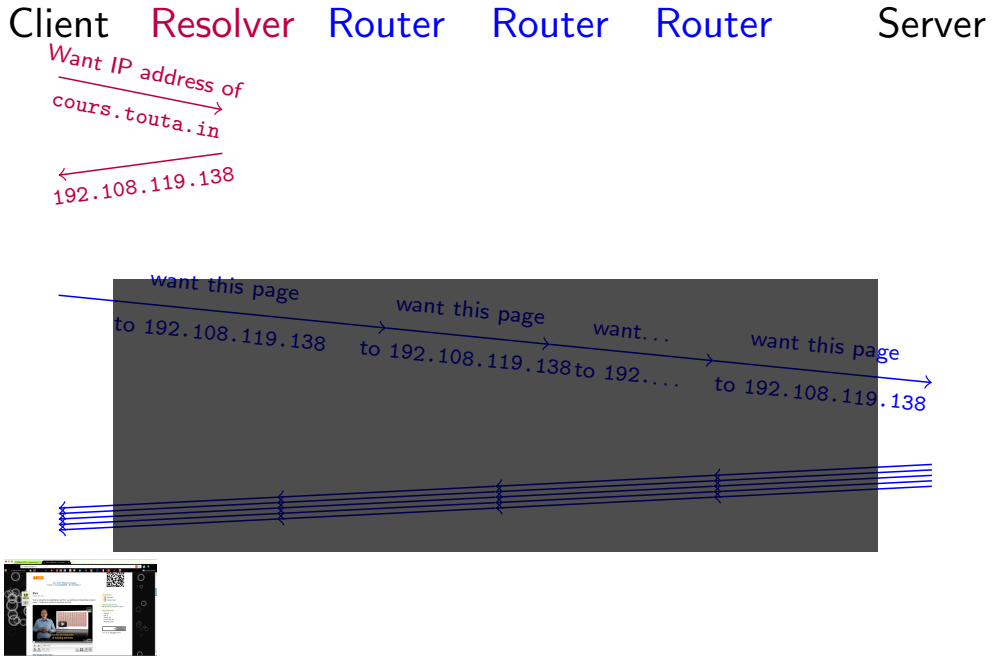


Main principles

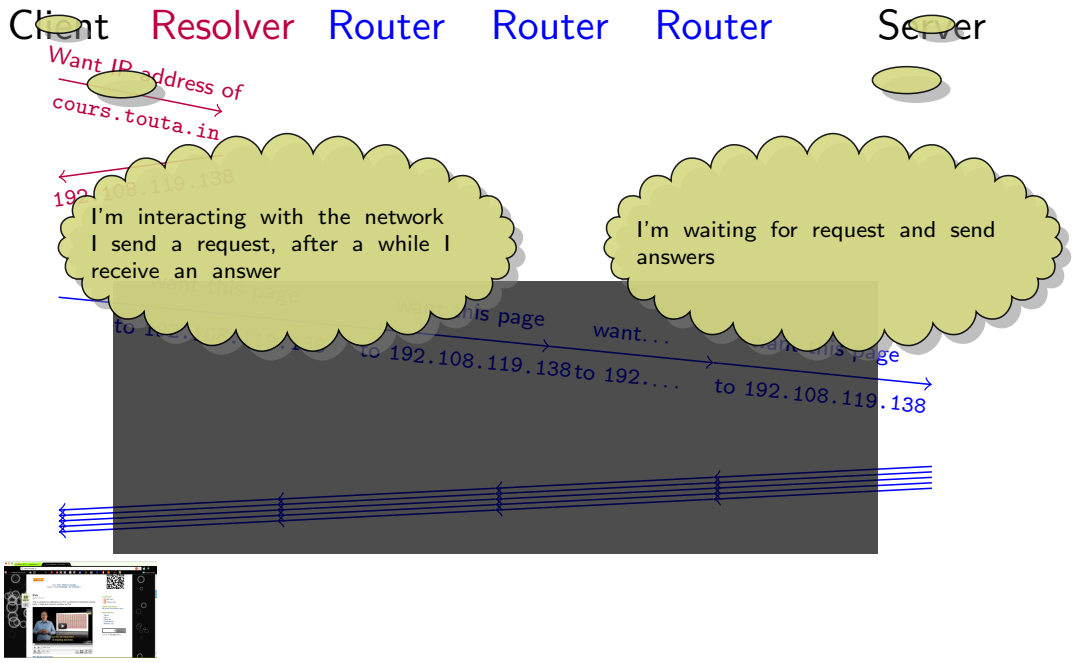


Introduction ▶ Basic architecture

Main principles



Main principles

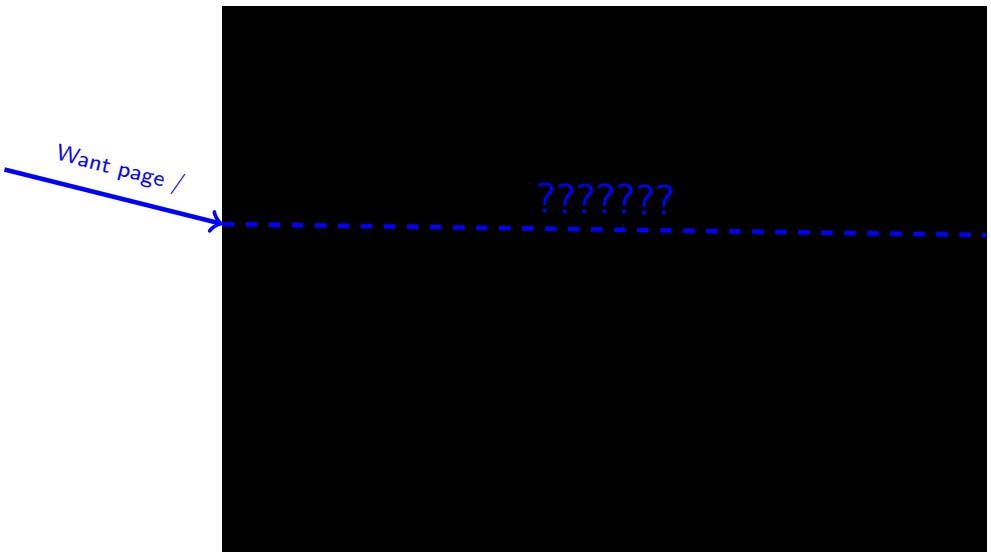


Interactions

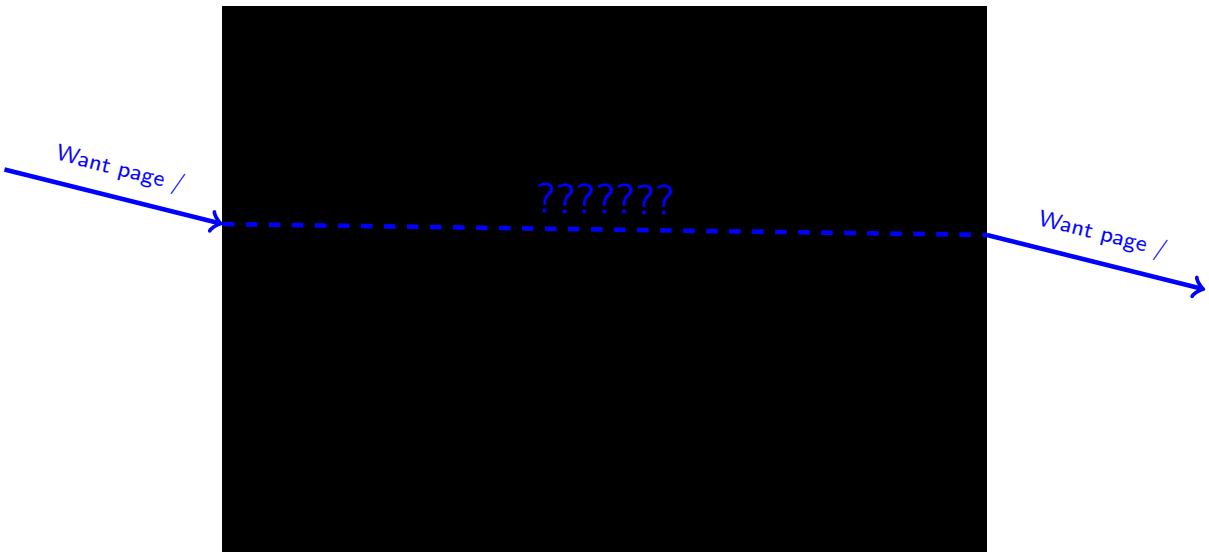
Interactions

Introduction ▶ Basic architecture

Interactions

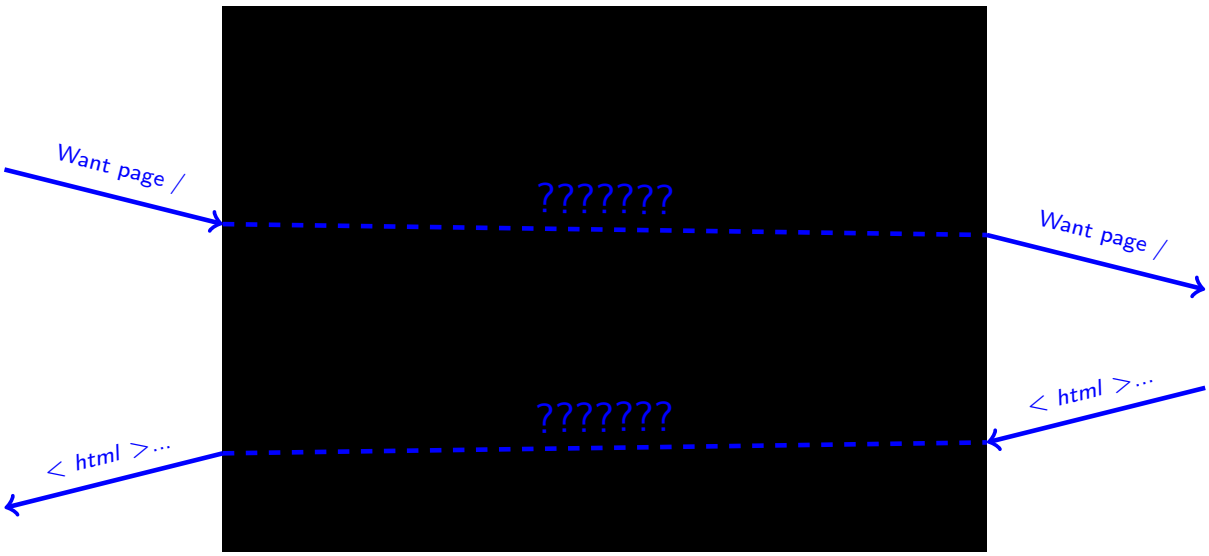


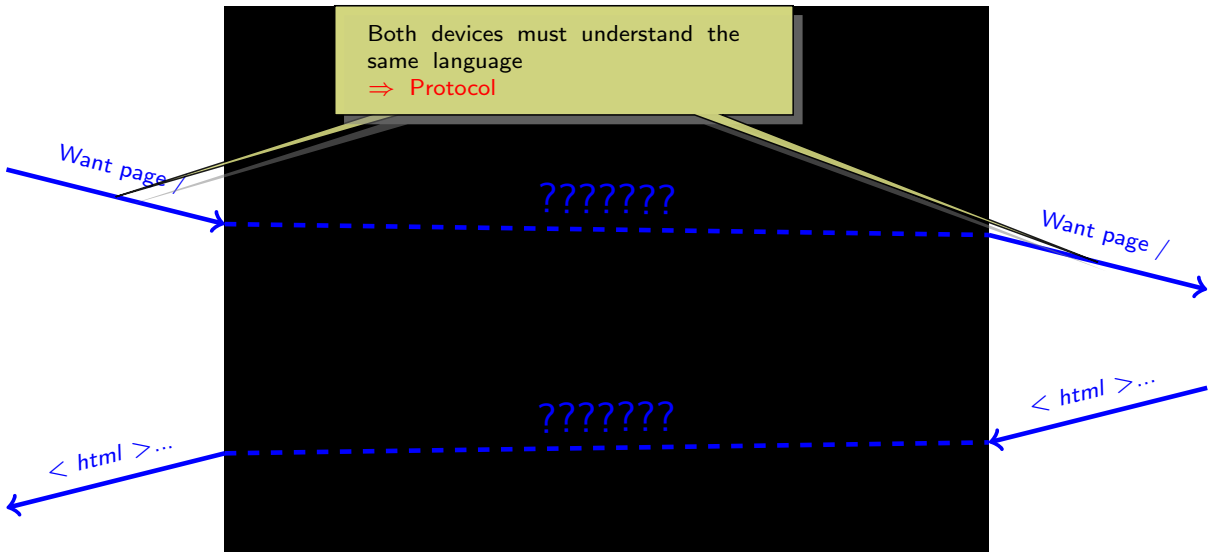
Interactions



Introduction ▶ Basic architecture

Interactions





Introduction

Protocols

Protocols

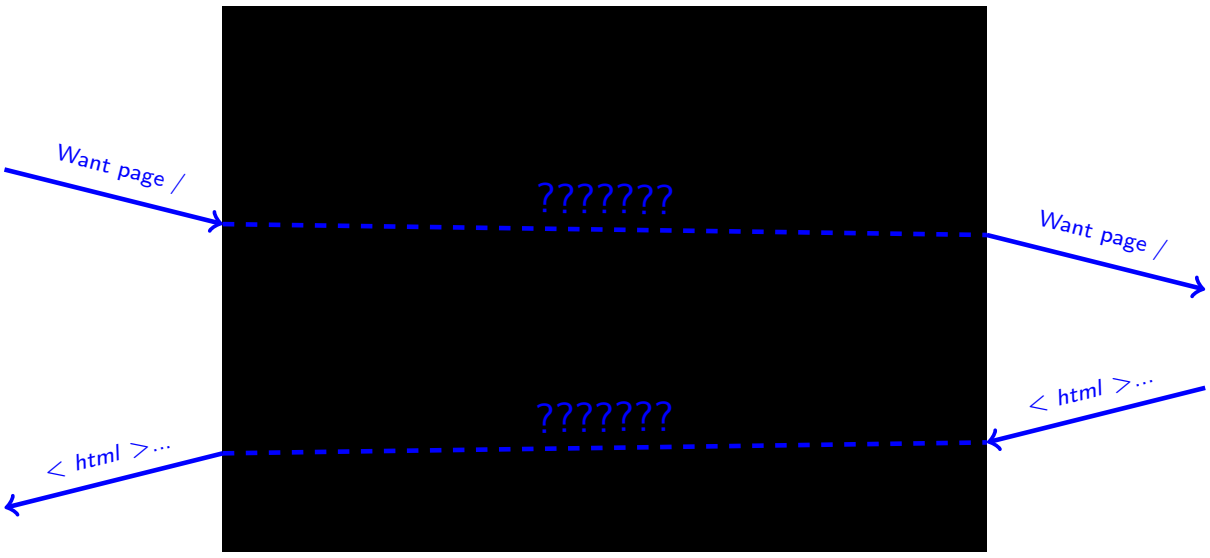
Definition

Protocol: common rules between two or several piece of equipment.

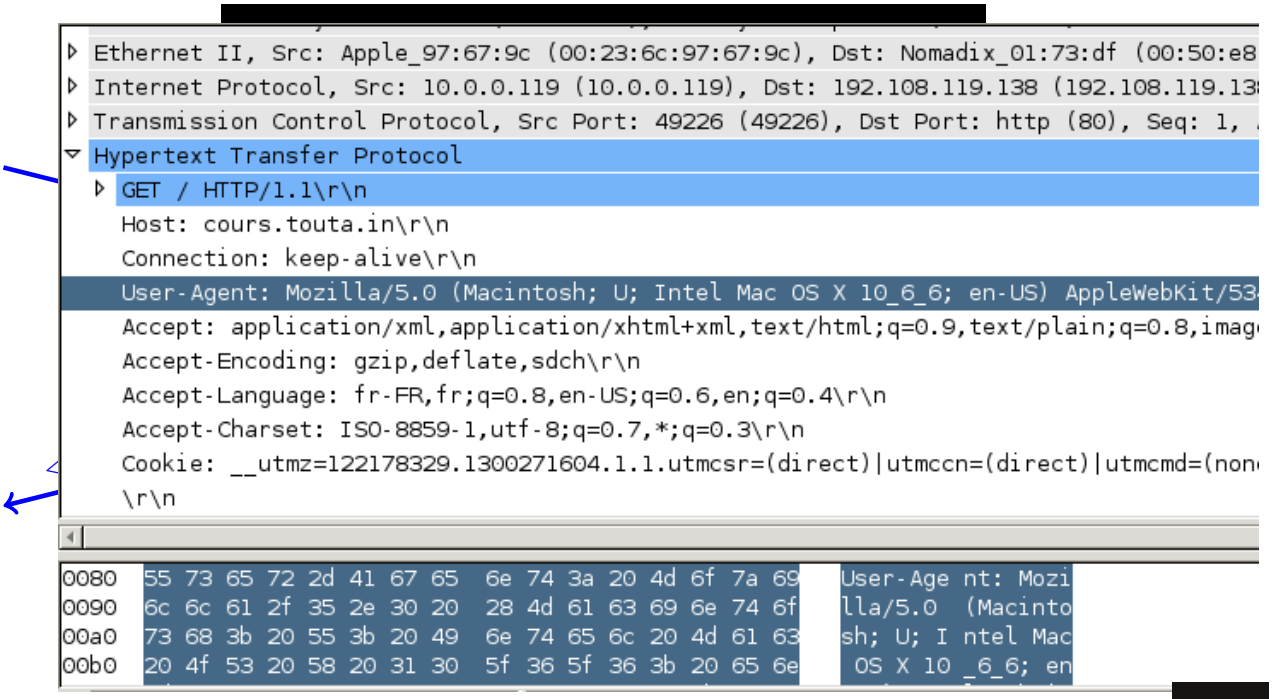
- common data format
- common data representation
- well defined behavior when receiving a message

Protocols are generally defined in Standards

Example

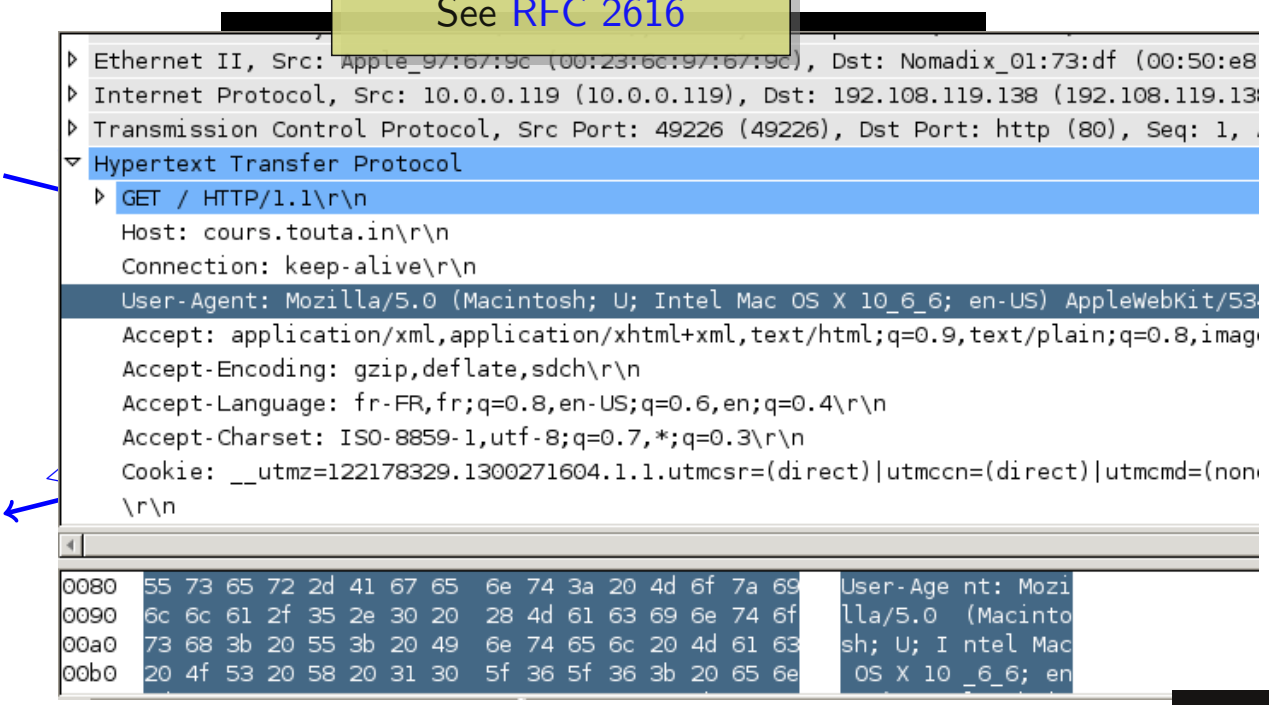


Example



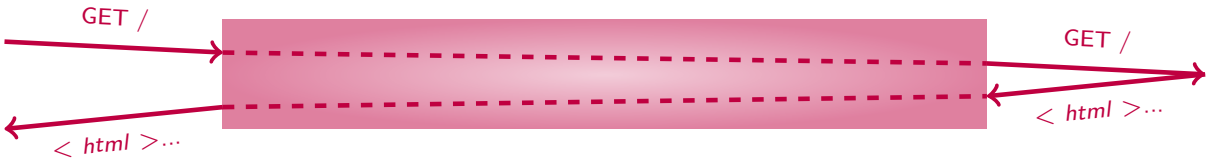
Example

See RFC 2616



Protocol Stack

- Each protocol is used for a single purpose,
- When stacked they can create a communication system.



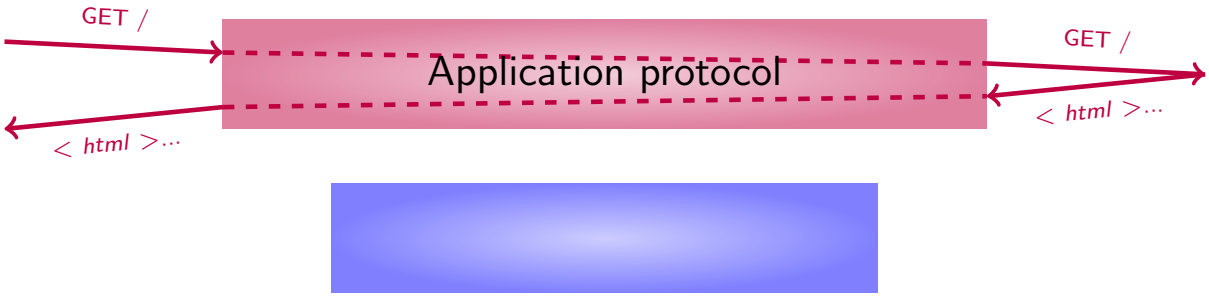
Protocol Stack

Introduction ▶ Protocols

- Each protocol is used for a single purpose,
- When stacked they can create a communication system.

Protocol Stack

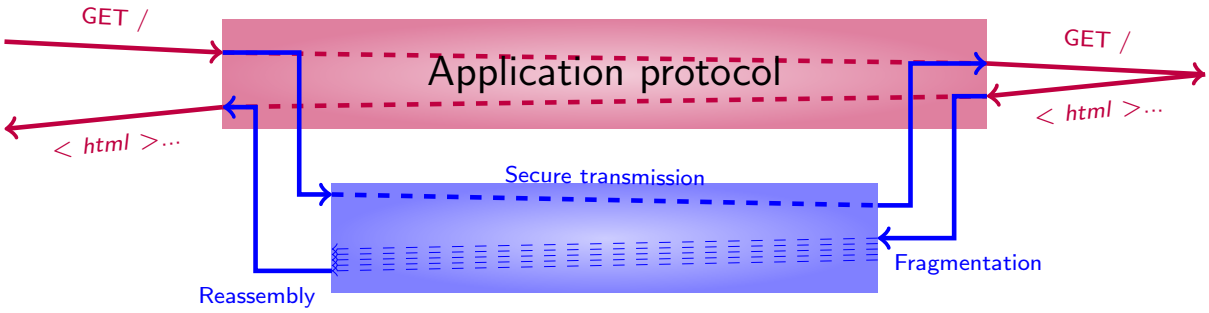
- Each protocol is used for a single purpose,
- When stacked they can create a communication system.



Protocol Stack

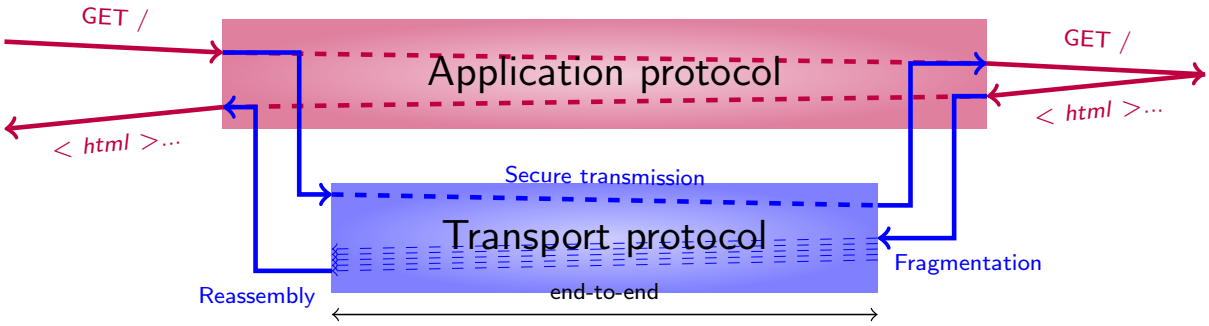
Introduction ▶ Protocols

- Each protocol is used for a single purpose,
- When stacked they can create a communication system.



Protocol Stack

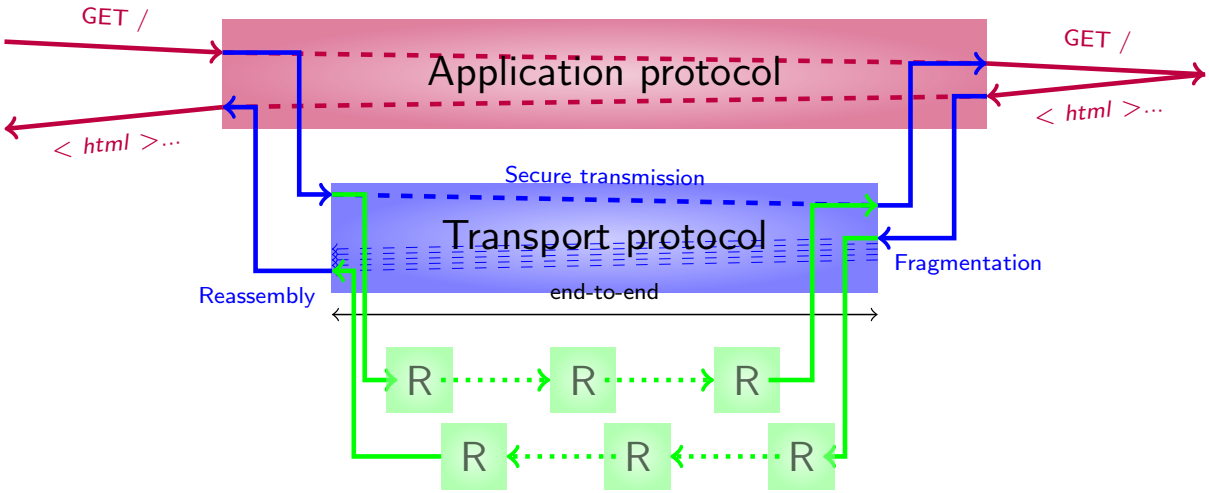
- Each protocol is used for a single purpose,
- When stacked they can create a communication system.



Protocol Stack

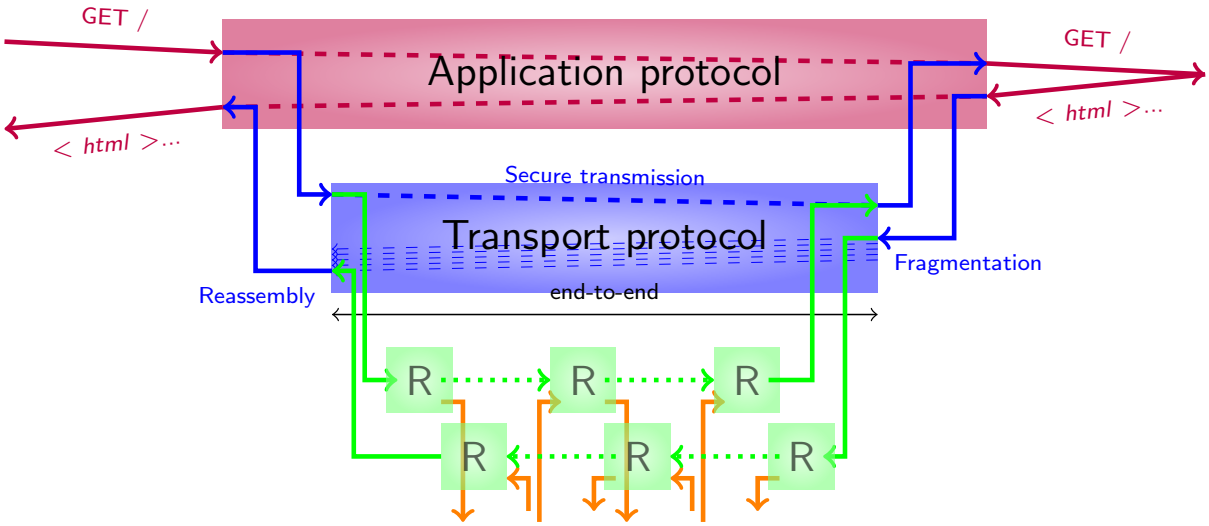
Introduction ▶ Protocols

- Each protocol is used for a single purpose,
- When stacked they can create a communication system.



Protocol Stack

- Each protocol is used for a single purpose,
- When stacked they can create a communication system.



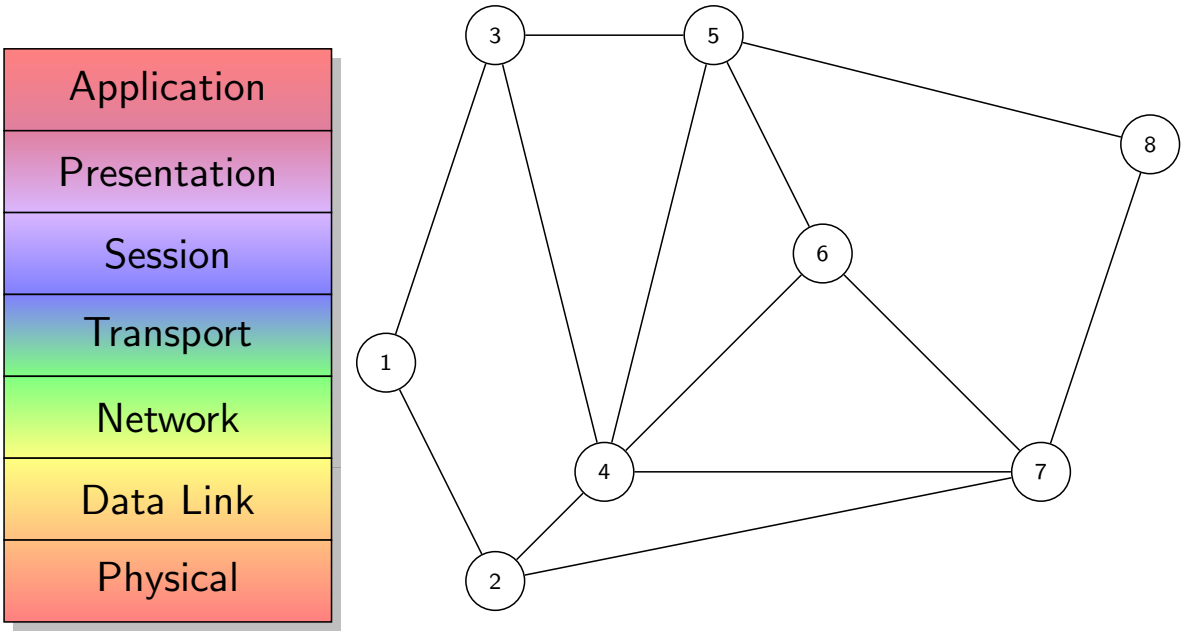
Introduction

OSI Architectural Reference Model

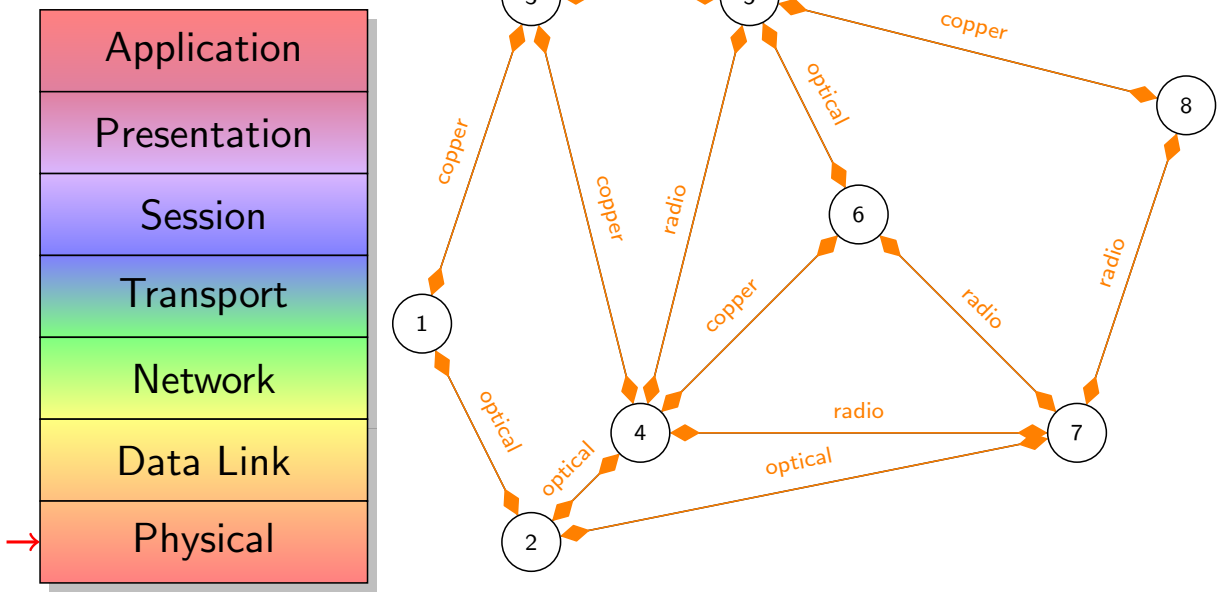
Open System Interconnection Reference Model

- Goal: Divide and conquer, system is divided into layers
- Each layer interact locally with the upper and lower layer.
- Remote interaction is possible only with the same layer
- Each layer regroups one or several entities implementing a protocol.
- Reference model defines 7 layers
  - may be less or more
- This is not an implementation model
  - Layer 2: limited scope, simple addresses
  - Layer 3: no limitation, scalable, uselly hierarchical addresses.

ISO Reference Model

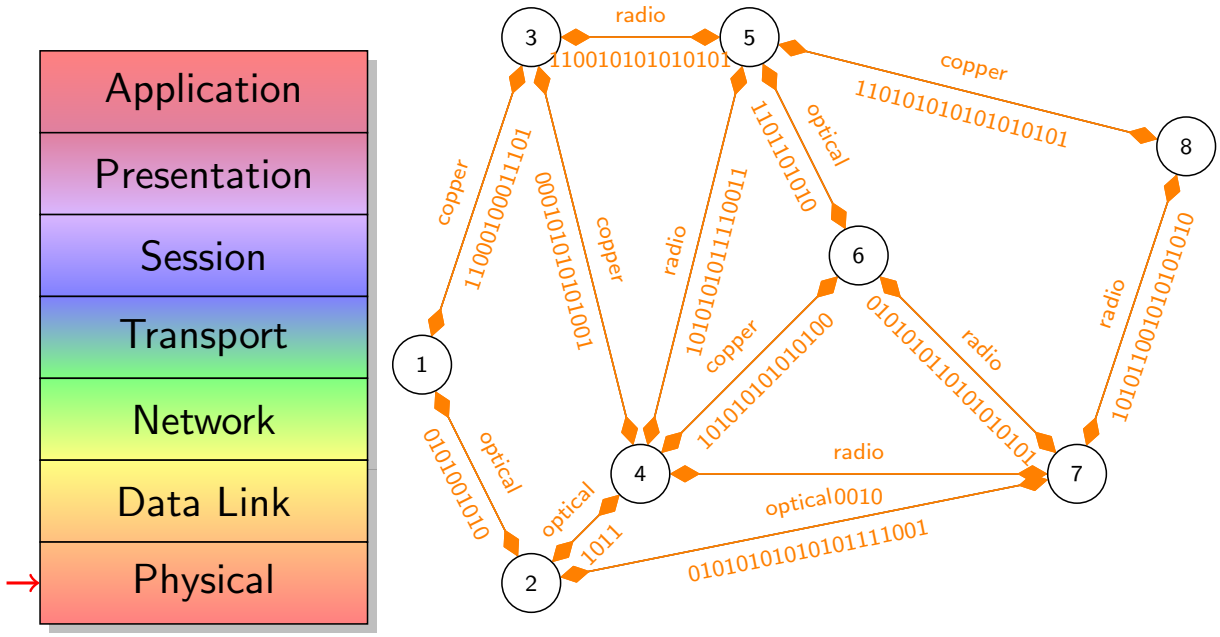


ISO Reference Model

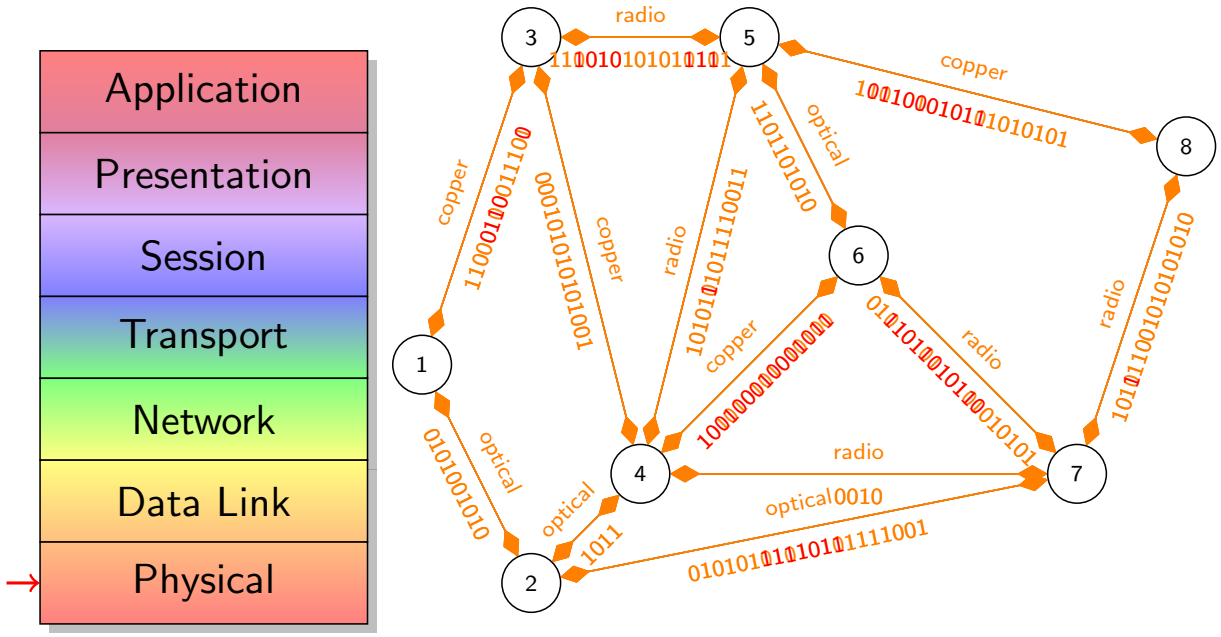


ISO Reference Model

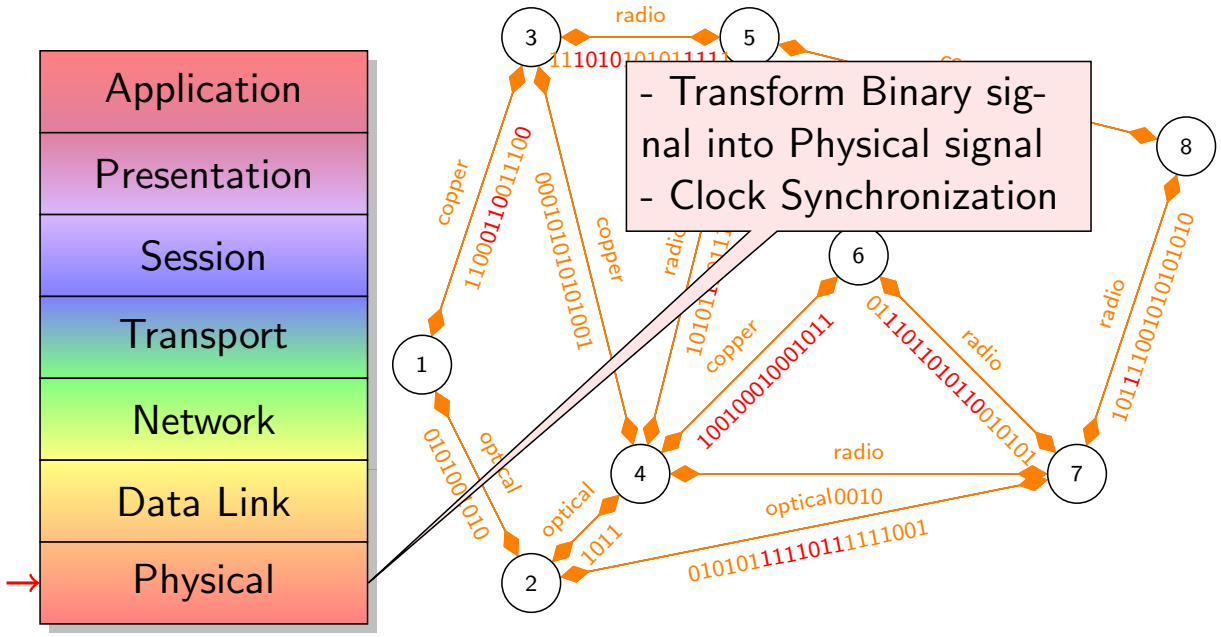
Introduction ▶ OSI Architectural Reference Model



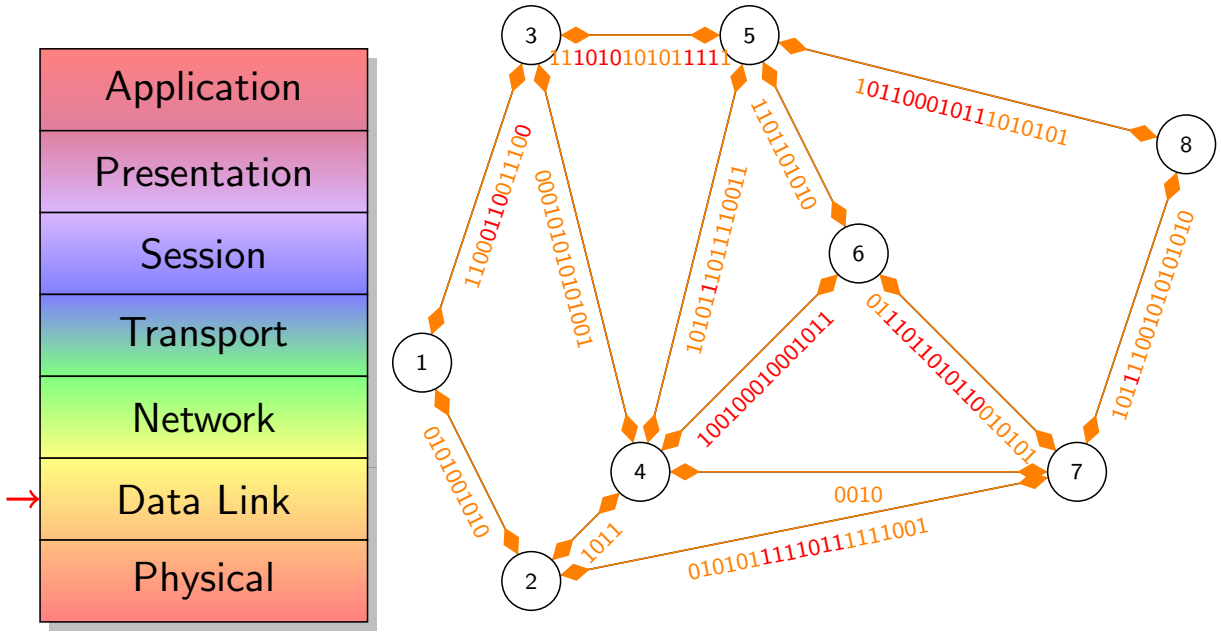
ISO Reference Model



ISO Reference Model

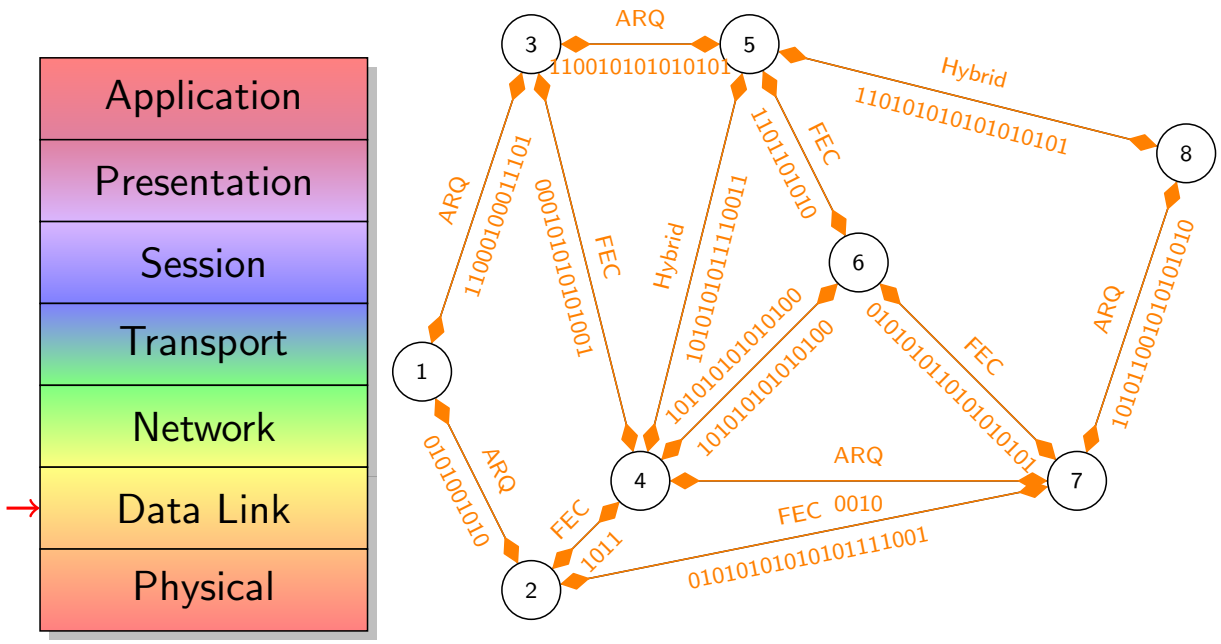


ISO Reference Model

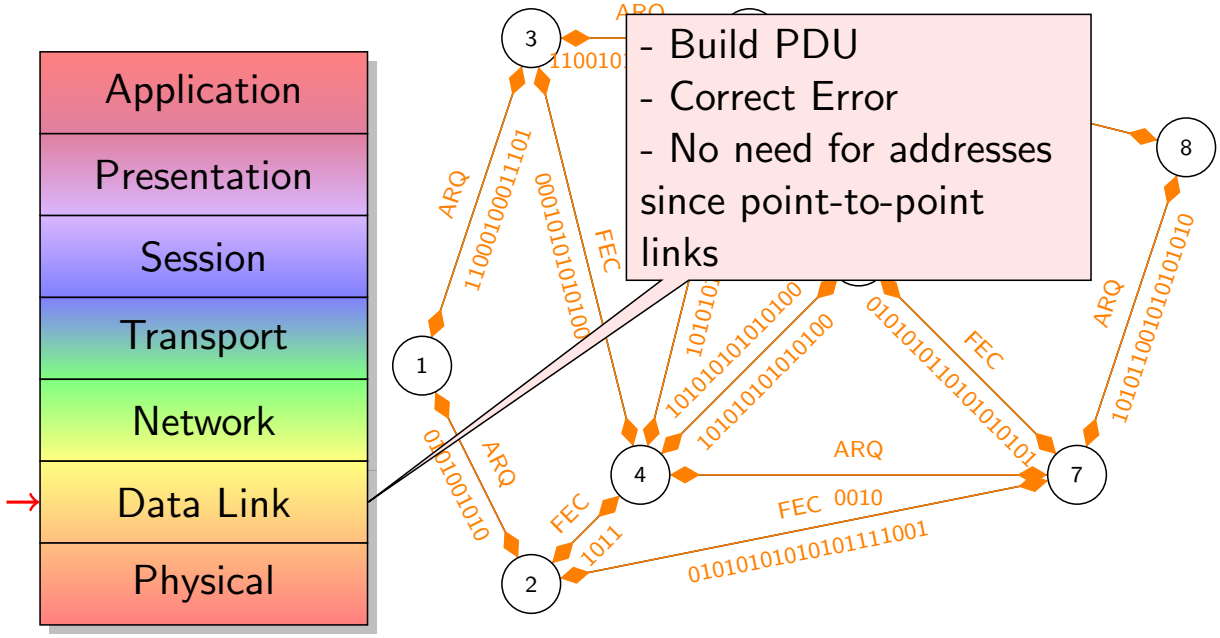


ISO Reference Model

Introduction ▶ OSI Architectural Reference Model

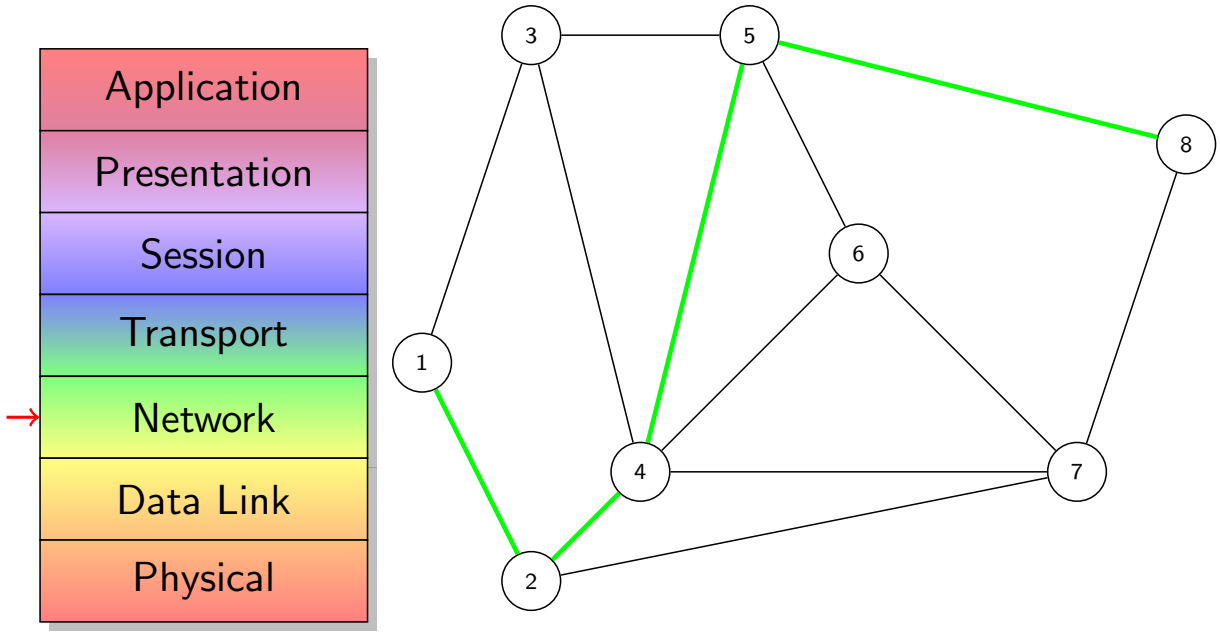


ISO Reference Model

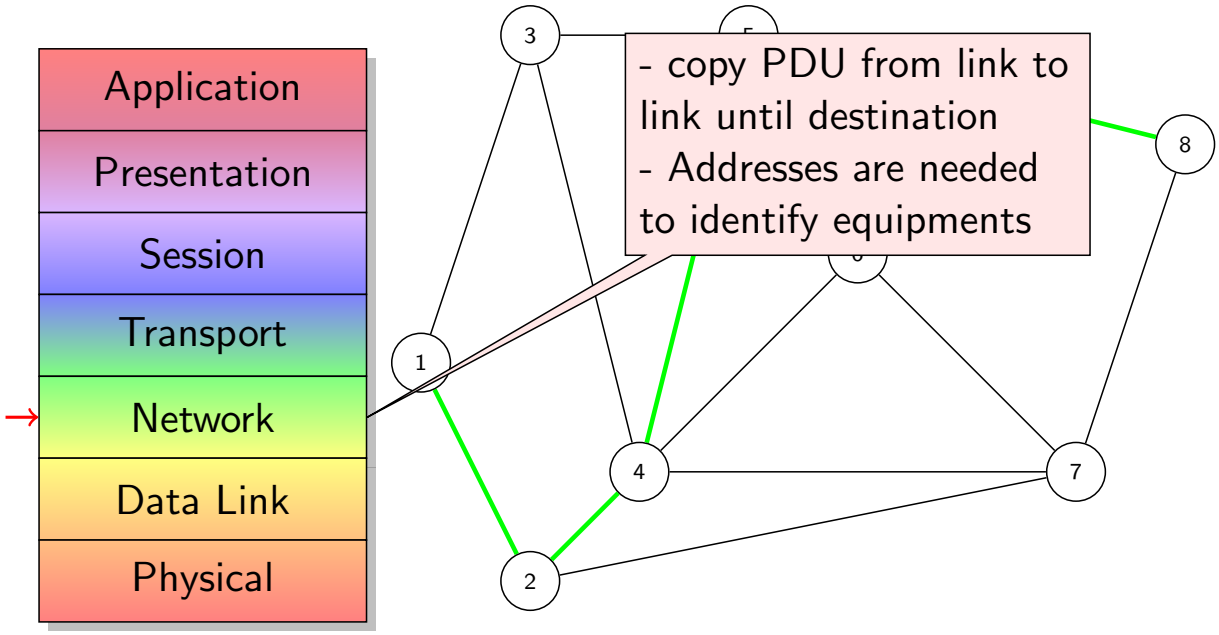


ISO Reference Model

Introduction ▶ OSI Architectural Reference Model

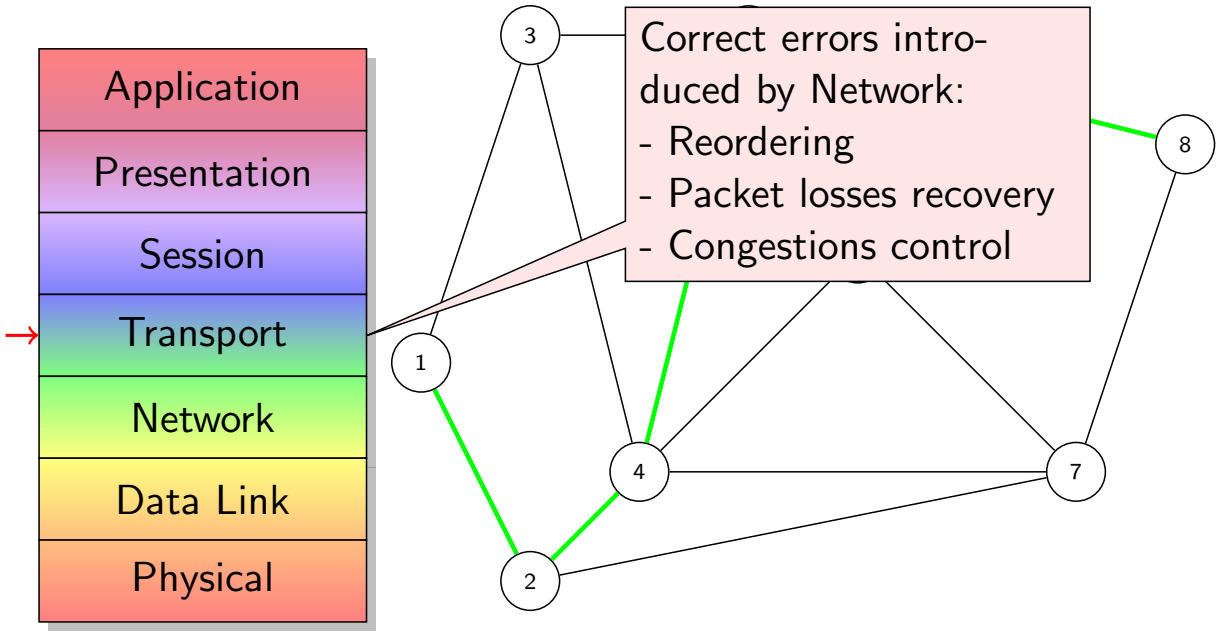


ISO Reference Model

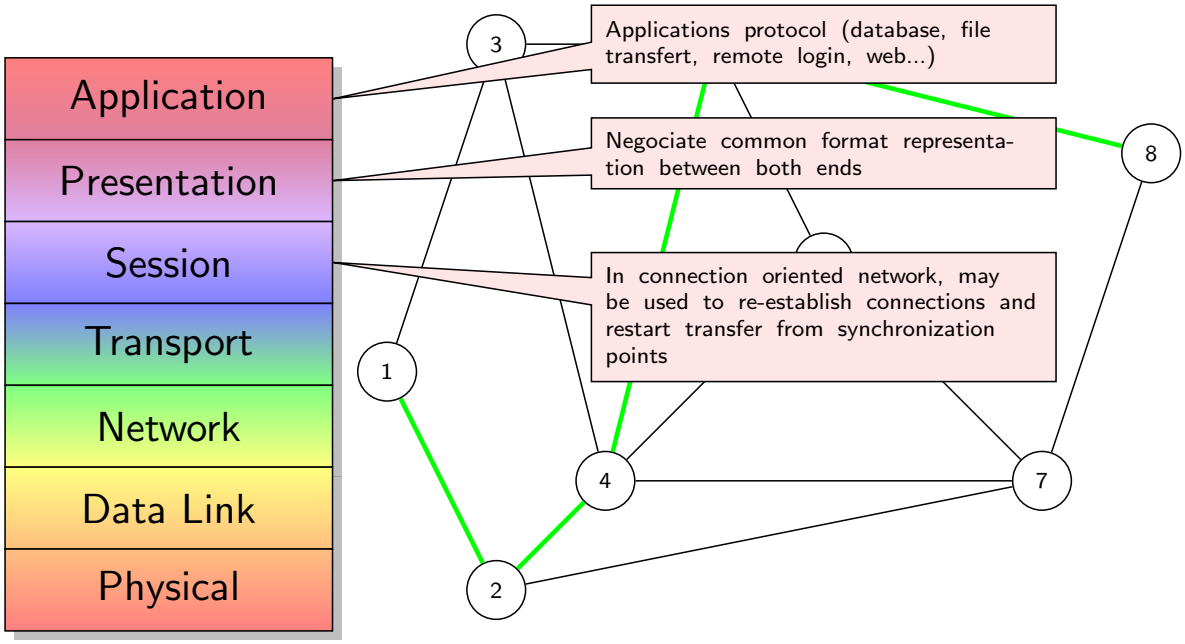


ISO Reference Model

Introduction ▶ OSI Architectural Reference Model



ISO Reference Model



Standardization bodies

Why standards ?

- Allow computers/application from different vendors/manufacturers to work together
- Guaranty that a system will continue to work even if one vendor disappears
- Lower the cost of devices by mass production
- Be sure to get connectivity
- Put patents into standards to earn money from competitors

Who publishes standards:

- Inter/national Organizations:
  - UN with ISO, France with AFNOR
- Vendors
  - IEEE, IETF, Foræ, ...

Telecommunication

- Cross road of different sectors.
- Transmission- International Telecommunication Union:
  - -- --- .-. ... . -.-. --- -.. . http://www.itu.int/rec/R-REC-M.1677/e
  - POTS (Plain Old Telephony Service),
  - More and more concerned by Internet and VoIP
- Local Computer Communications - IEEE
  - Local Area Network to Metropolitan Area Network
- Regular standardization: International Standard Organization
  - Input from ITU, IEEE
  - Well known for 7 layer Reference Model
- Internet - IETF
  - Develop Internet protocols
  - Independent from ITU, ISO
  - Managed by vendors and industrials
- Telephony - ETSI / 3GPP
  - DECT, GSM, 3G, LTE

How to access standards ?

- ITU: Letter . Number
  - Letter: group
    - H: Audiovisual and multimedia systems
    - Q: Switching and signalling
    - X: Data networks, open system communications and security
    - ...
  - Example X.25 (see W http://www.itu.int/rec/T-REC-X.25/en)
  - Standards are regularly revised (see publication date).
- IEEE: P series (see W http://grouper.ieee.org/groups/) and P802 for LAN/MAN
  - IEEE 802.3 (see W http://standards.ieee.org/about/get/802/802.3.html)
  - P1095 Convergent Digital Home Network Working Group (see W http://grouper.ieee.org/groups/1905/1/)
  - Standards are regularly revised (see publication date).

How to access standards ?

- IETF: RFC *number*
  - For instance RFC 2460
  - See W http://www.ietf.org
  - Freely available as well as working drafts
  - When published, RFCs never change, look at W http://www.rfc-editor.org/rfc-index2.html for updates.
- ETSI: ETSI Technical Specification (TS) or Technical Report (TR) *wg.number*

OSI Reference Model

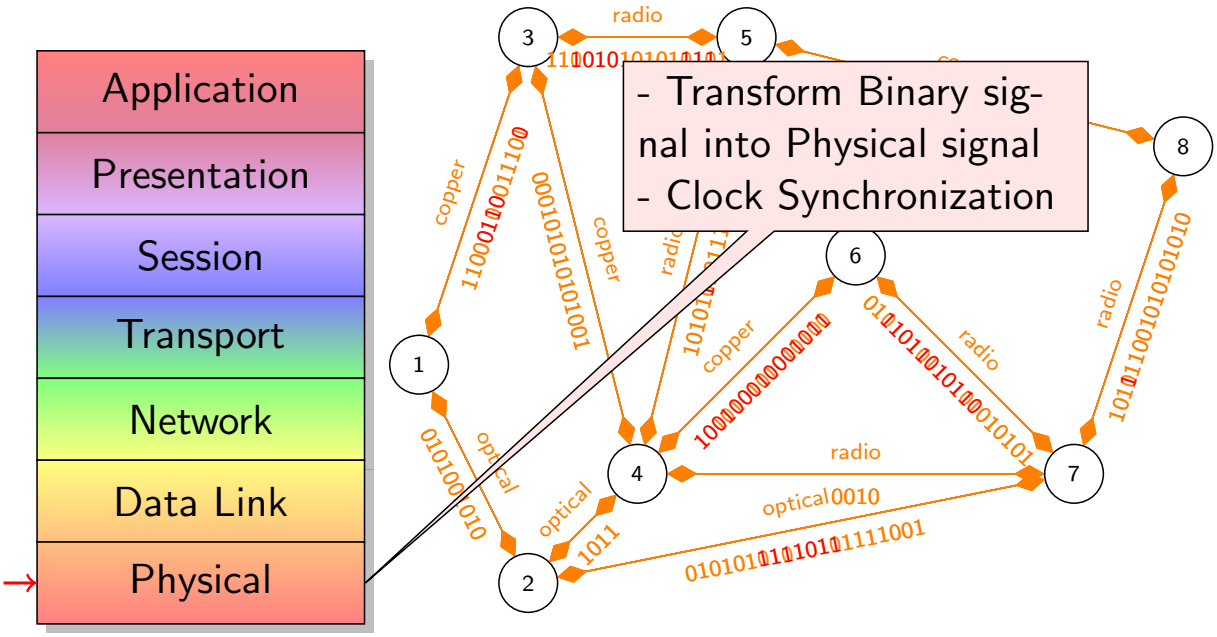
Open System Interconnection Reference Model

- Goal: Divide and conquer, system is divided into layers
- Each layer interact locally with the upper and lower layer.
- Remote interaction is possible only with the same layer
- Each layer regroups one or several entities implementing a protocol.
- Reference model defines 7 layers
  - may be less or more
- This is not an implementation model
  - Layer 2: limited scope, simple addresses
  - Layer 3: no limitation, scalable, uselly hierarchical addresses.

OSI Reference Model

Layer 1: Physical

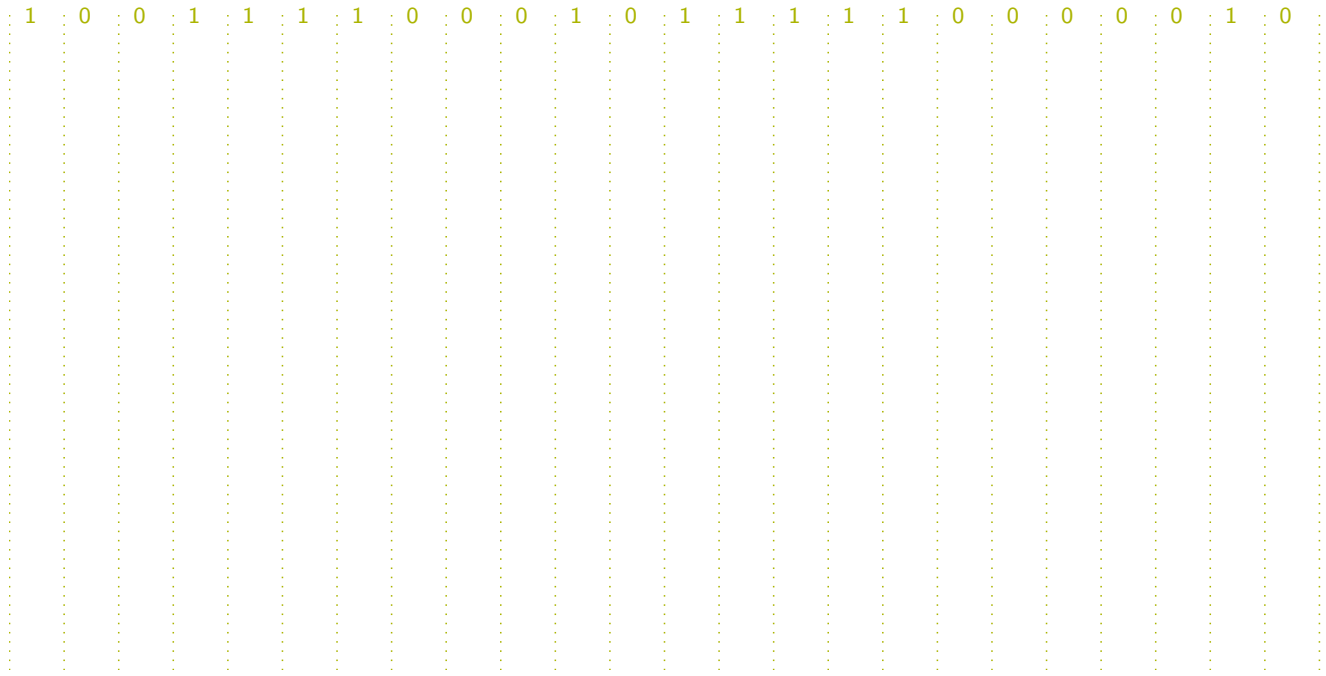
ISO Reference Model



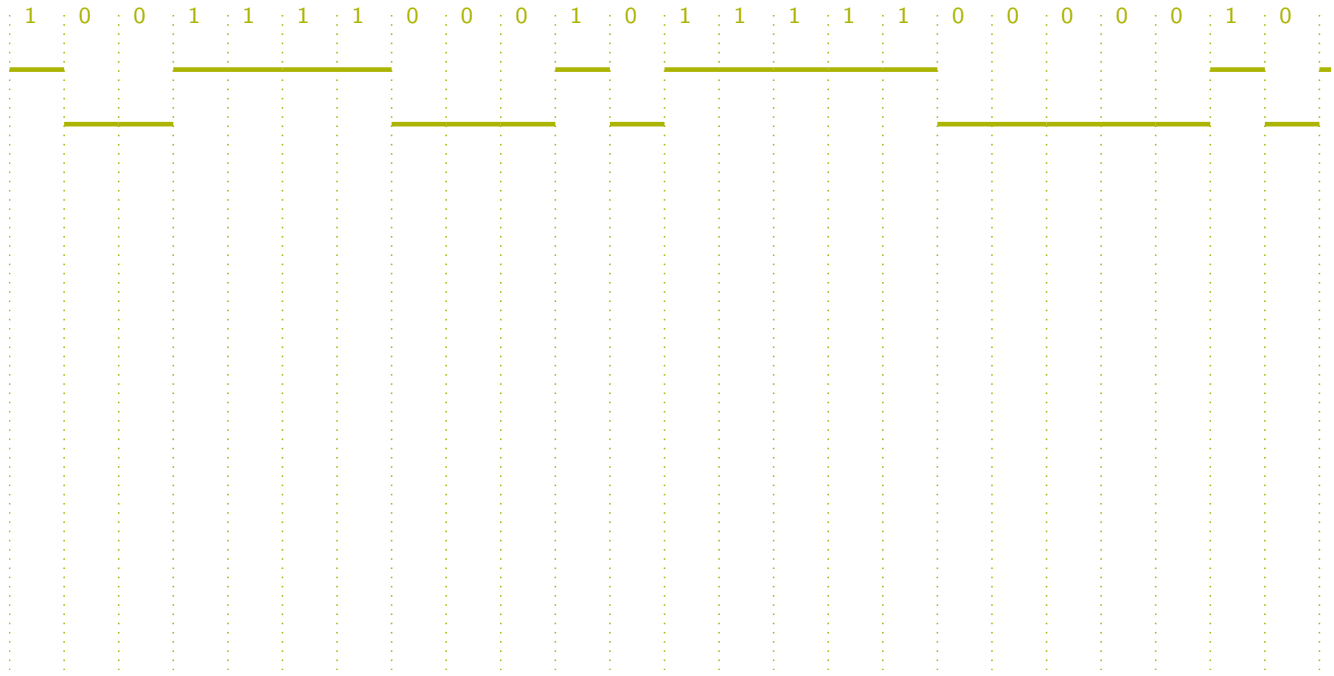
Layer 1 goals

- officially: Allow the interconnection between the device producing/consuming the data (DTE: Data Terminal Equipment) and the device that provide the signal modulation (DCE: Data Communications Equipment)
  - X.21 interface
  - AT commands
- more generally: includes the way the information is coded.
  - Limits error rate
  - Avoid interferences
  - Allow clock synchronisation
  - Enhance channel usage
  - Can be Broadband (some frequencies) or Baseband (all available frequencies)

Baseband Coding

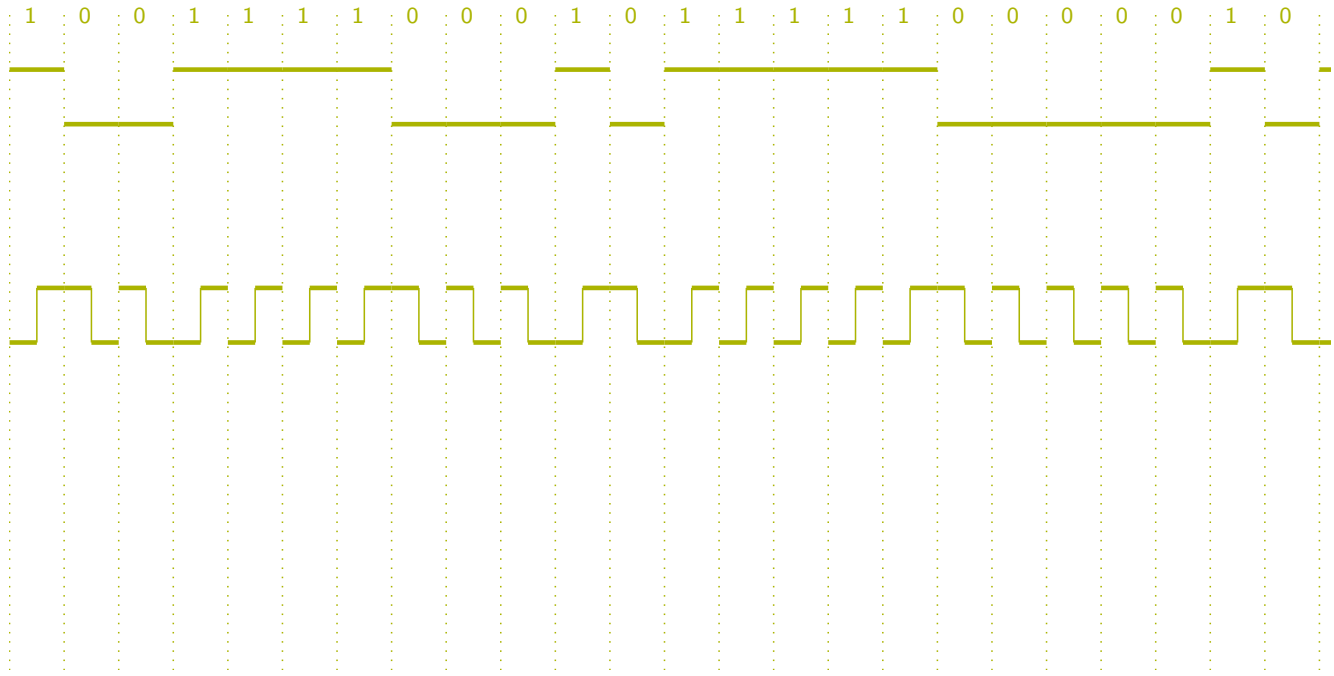


Baseband Coding



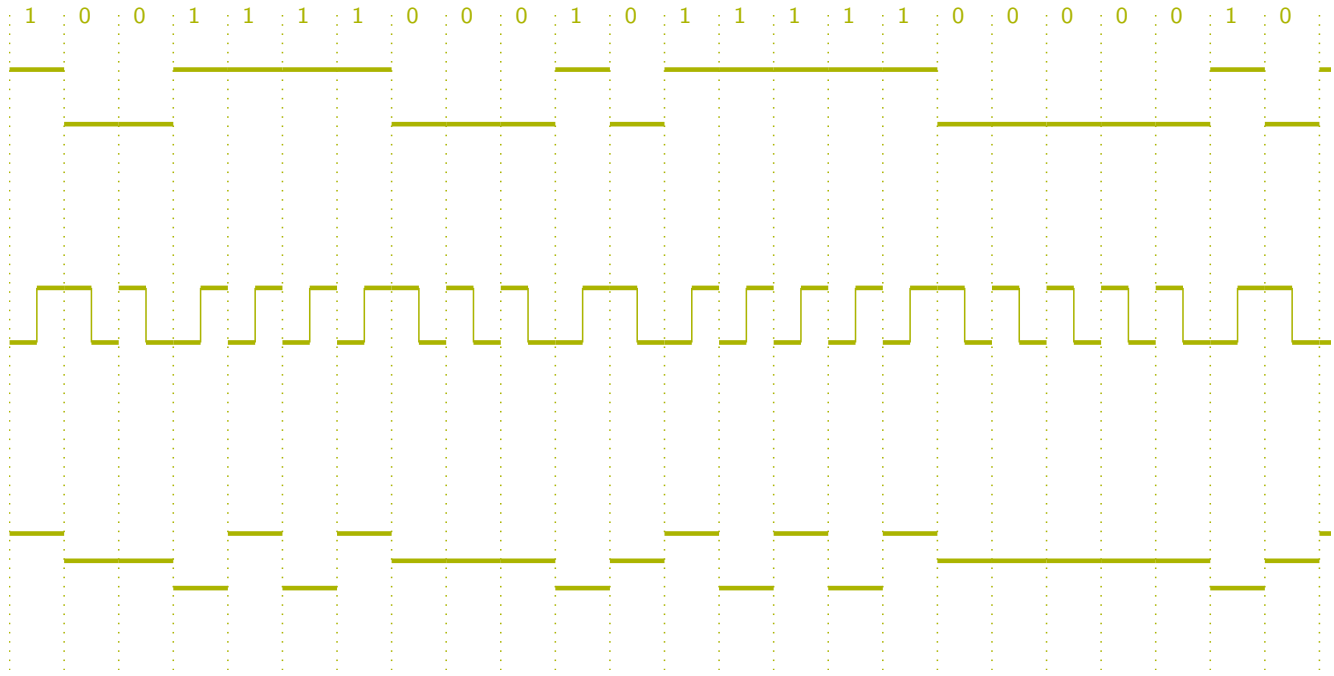
Baseband Coding

OSI Reference Model ▶ Layer 1: Physical



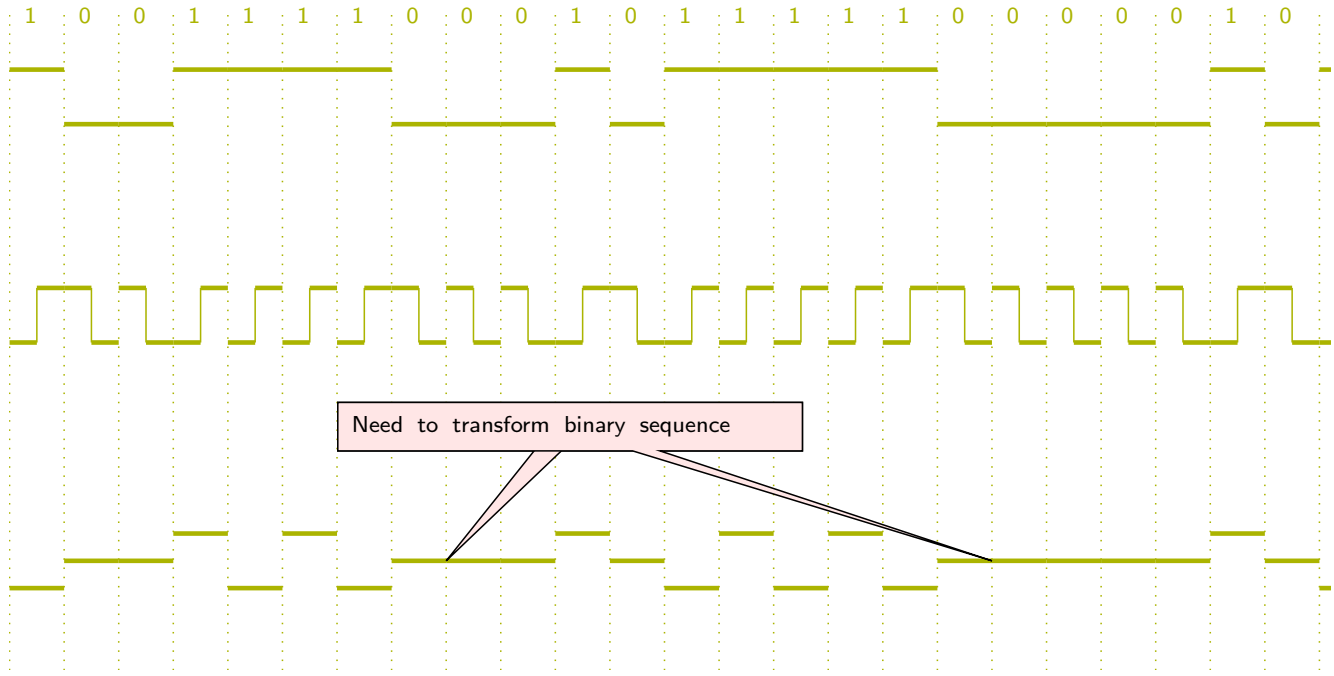
Slide 26 Laurent Toutain RES 1

Baseband Coding



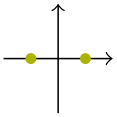
Baseband Coding

OSI Reference Model ▶ Layer 1: Physical

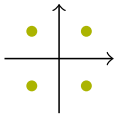


Slide 26 Laurent Toutain RES 1

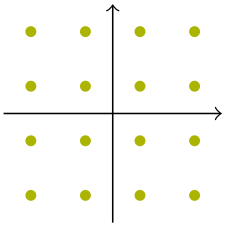
Modulations



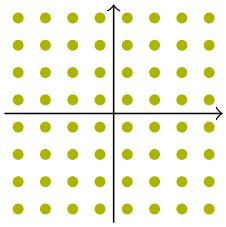
BPSK: Binary Phase Shift Keying (20 MHz)
Coding rate = 1/2 ⇒ 6 Mbit/s
Coding rate = 3/4 ⇒ 9 Mbit/s



QPSK: Quadrature Phase Shift Keying (20 MHz)
Coding rate = 1/2 ⇒ 12 Mbit/s
Coding rate = 3/4 ⇒ 18 Mbit/s

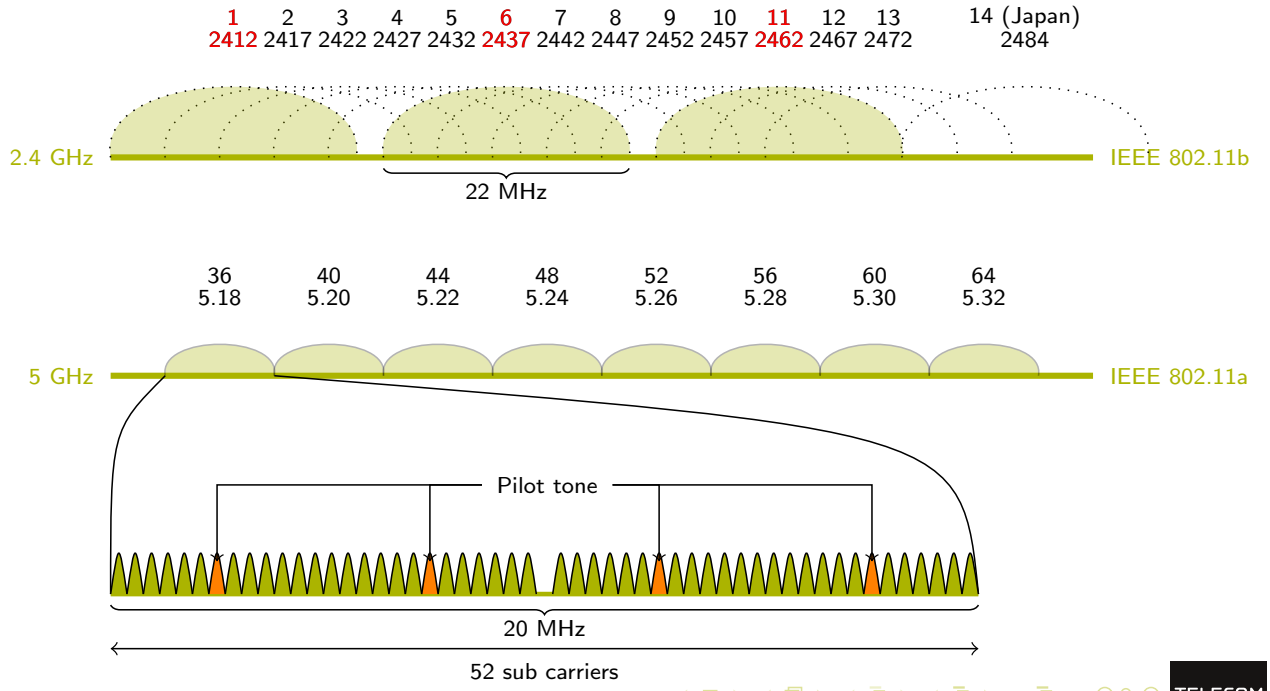


16-QAM: Quadrature Amplitude Modulation (20 MHz)
Coding rate = 1/2 ⇒ 24 Mbit/s
Coding rate = 3/4 ⇒ 36 Mbit/s



64-QAM: Quadrature Amplitude Modulation
Coding rate = 2/3 ⇒ 48 Mbit/s
Coding rate = 3/4 ⇒ 54 Mbit/s

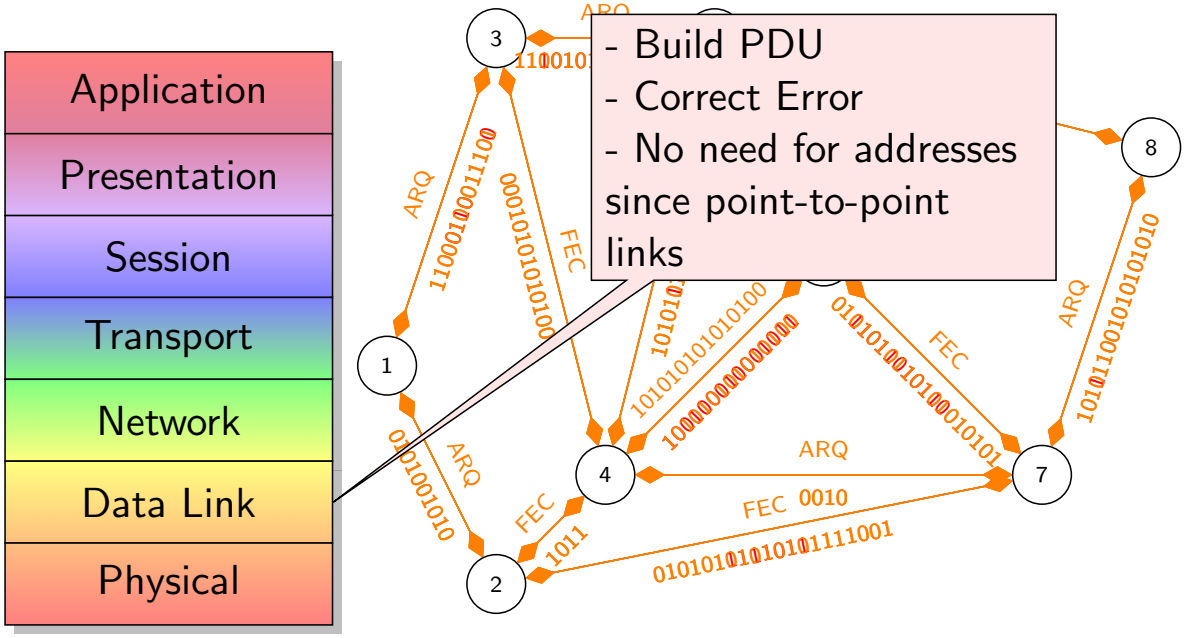
Baseband: Example Wi-Fi



OSI Reference Model

Layer 2: Link Layer

ISO Reference Model



ARQ: Automatic Repeat-reQuest

- Send acknowledgment (positive or negative) to ask for retransmission of lost frames
- Stupid example: Omelet recipe[1]
  1. Buy eggs
  2. Crack the eggs into a glass mixing bowl and beat them until they turn a pale yellow color.
  3. With a heat-resistant rubber spatula, gently push one edge of the egg into the center of the pan, while tilting the pan to allow the still liquid egg to flow in underneath. Repeat with the other edges, until there's no liquid left.
- A cook send the recipe through the postal network to a student.
  - What are the properties of Postal Network ?

[1] Source: http://culinaryarts.about.com/od/eggsdairy/r/omelet.htm

ARQ: Automatic Repeat-reQuest

OSI Reference Model ▶ Layer 2: Link Layer

- Send acknowledgment (positive or negative) to ask for retransmission of lost frames
- Stupid example: Omelet recipe[1]
  1. Buy eggs
  2. Crack the eggs into a glass mixing bowl and beat them until they turn a pale yellow color.
  3. With a heat-resistant rubber spatula, gently push one edge of the egg into the center of the pan, while tilting the pan to allow the still liquid egg to flow in underneath. Repeat with the other edges, until there's no liquid left.
- A cook send the recipe through the postal network to a student.
  - What are the properties of Postal Network ?
  - Allow to send limited set of information far away
  - May disorder letters (or loose letters)

[1] Source: http://culinaryarts.about.com/od/eggsdairy/r/omelet.htm

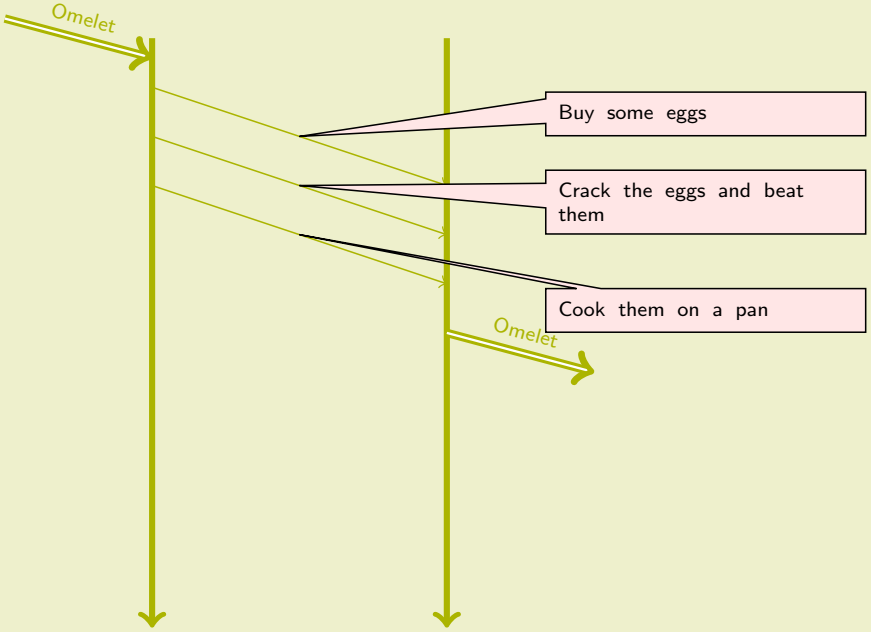
Protocol definition

Define a protocol to allow the cook to ask the student to cook an omelet

Protocol definition

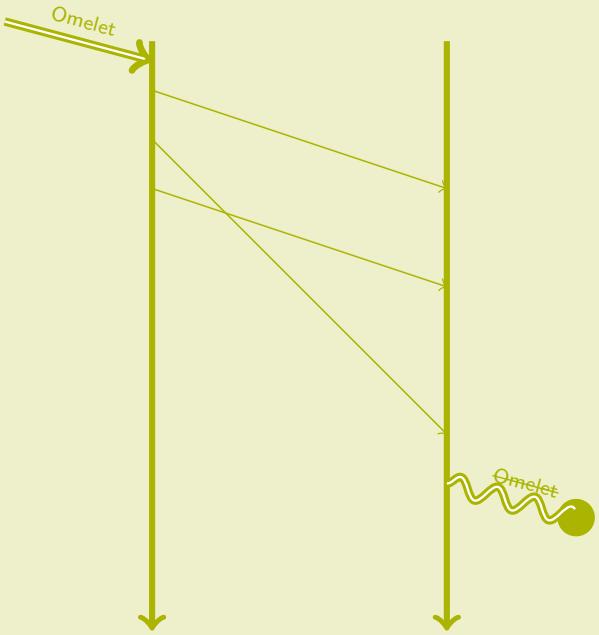
OSI Reference Model ▶ Layer 2: Link Layer

Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

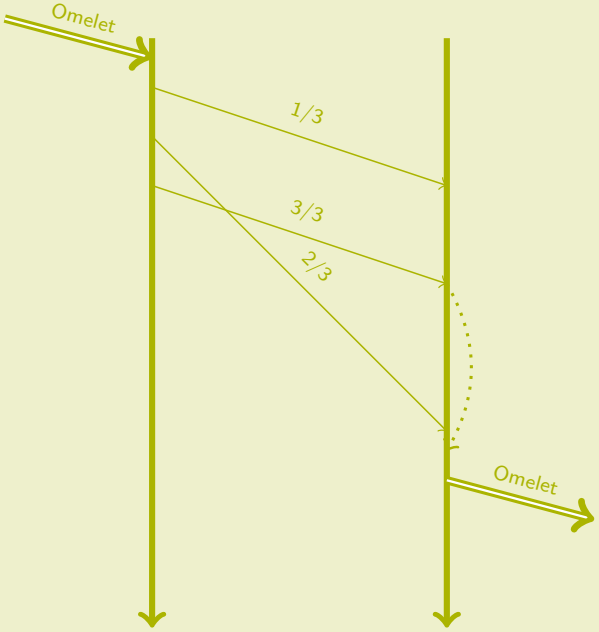
Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

OSI Reference Model ▶ Layer 2: Link Layer

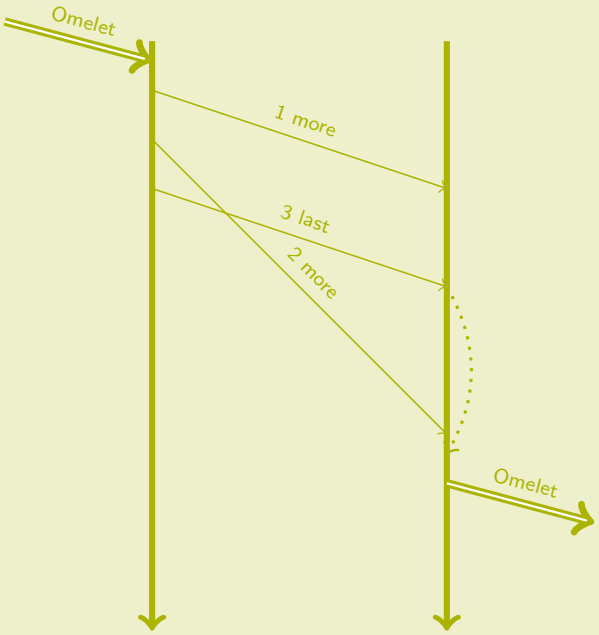
Define a protocol to allow the cook to ask the student to cook an omelet



Slide 34 Laurent Toutain RES 1

Protocol definition

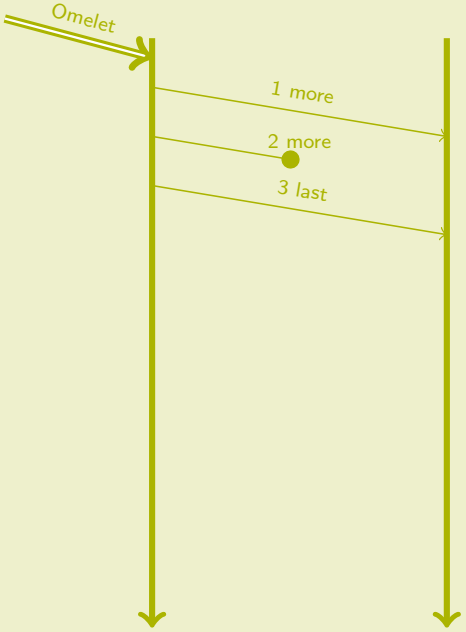
Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

OSI Reference Model ▶ Layer 2: Link Layer

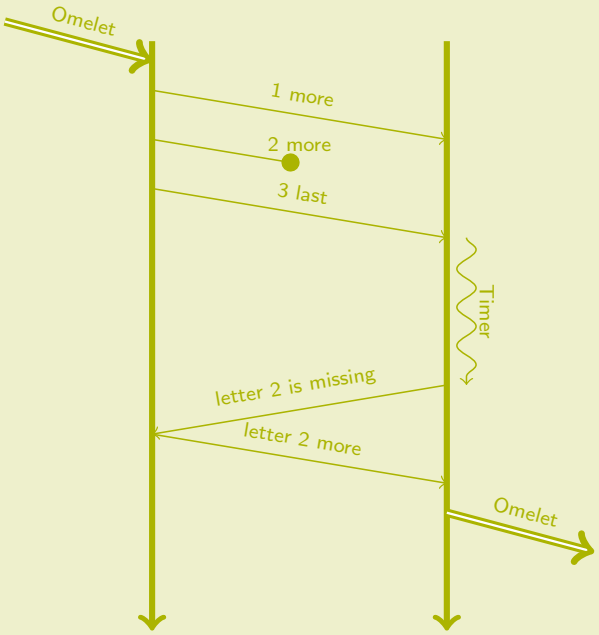
Define a protocol to allow the cook to ask the student to cook an omelet



Slide 34 Laurent Toutain RES 1

Protocol definition

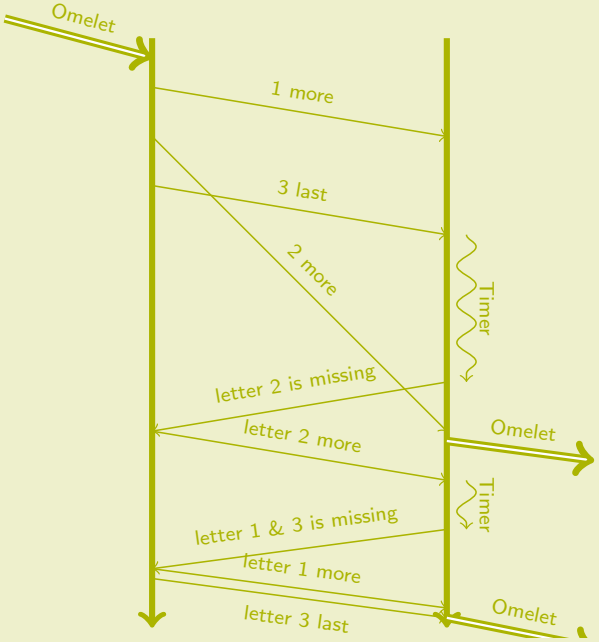
Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

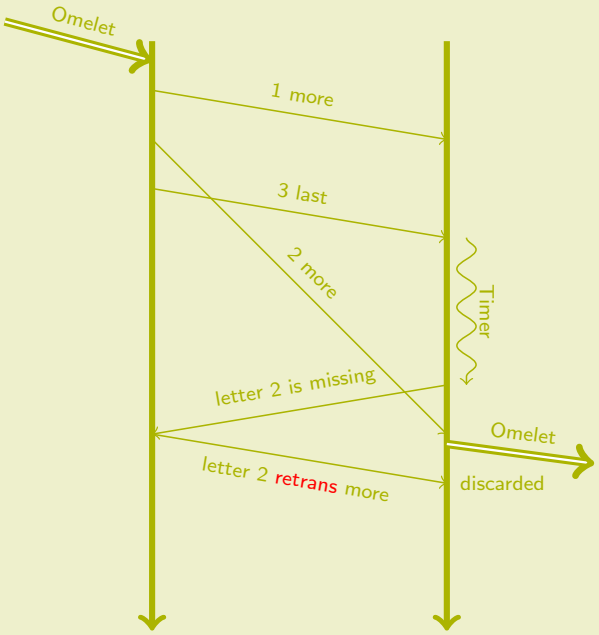
OSI Reference Model ▶ Layer 2: Link Layer

Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

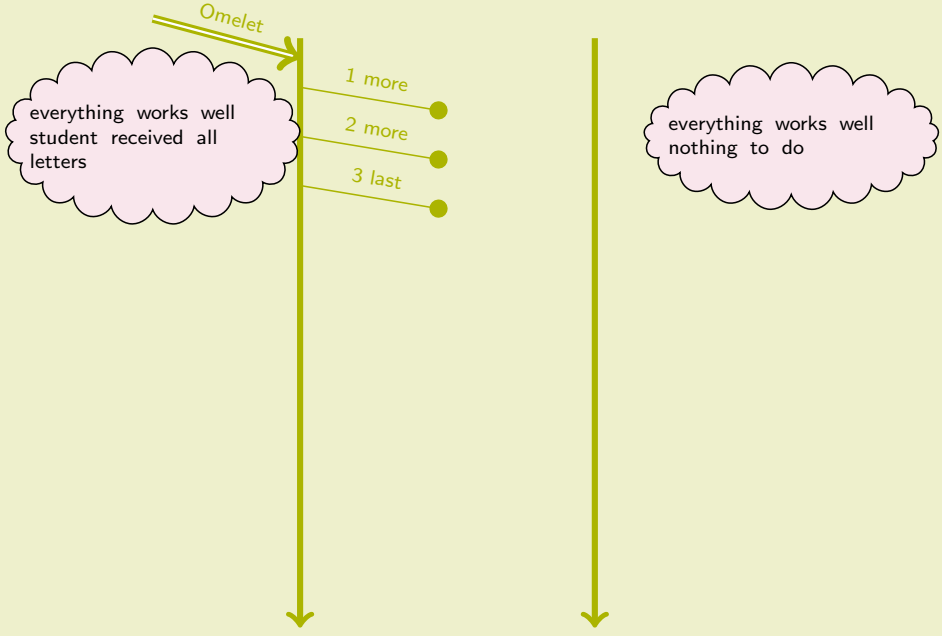
Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

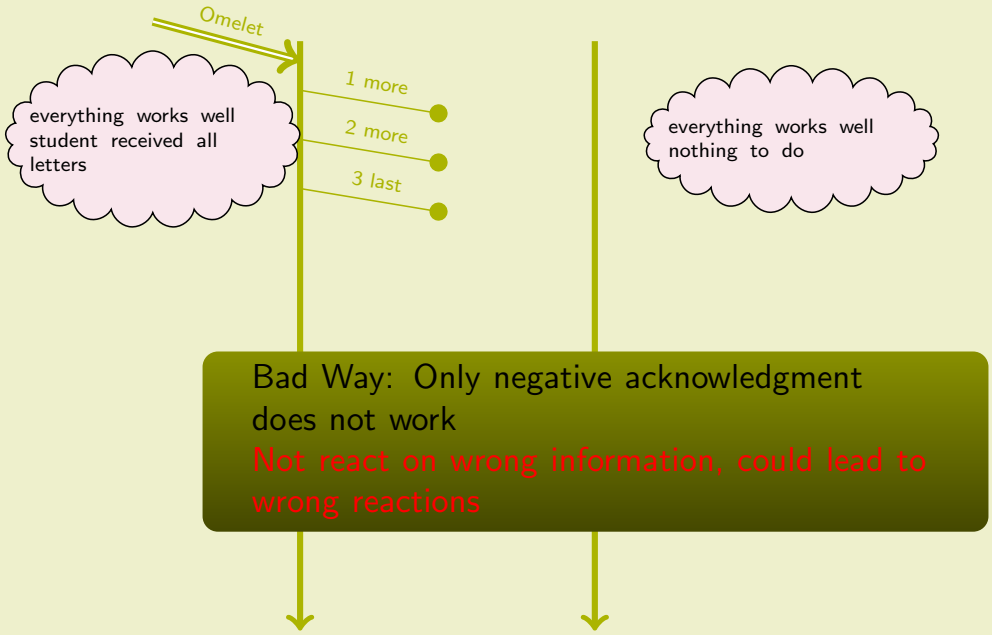
OSI Reference Model ▶ Layer 2: Link Layer

Define a protocol to allow the cook to ask the student to cook an omelet



Protocol definition

Define a protocol to allow the cook to ask the student to cook an omelet



Bad Way: Only negative acknowledgment does not work

Not react on wrong information, could lead to wrong reactions

TELECOM Bretagne

OSI Reference Model

CRC

Parity bits

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1	0
1	0	0	1	1	0	1	0	0	1
1	1	1	0	0	0	1	0	1	0
1	0	0	0	0	0	0	0	1	0
1	1	0	0	1	0	0	0	0	1
1	0	0	1	0	1	0	1	0	1
1	1	1	0	1	0	0	0	1	0
1	0	1	1	0	1	0	1	0	1
1	0	1	0	0	0	1	1	1	0

Parity bits

OSI Reference Model ▶ CRC

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1	6
0	1	0	0	0	0	1	0	1	0	3
1	0	0	1	1	0	1	0	0	1	5
1	1	1	0	0	0	1	0	1	0	5
1	0	0	0	0	0	0	0	1	0	2
1	1	0	0	1	0	0	0	0	1	4
1	0	0	1	0	1	0	1	0	1	5
1	1	1	0	1	0	0	0	1	0	5
1	0	1	1	0	1	0	1	0	1	6
1	0	1	0	0	0	1	1	1	0	5
9	4	5	4	3	3	4	3	6	5	

Slide 36 Laurent Toutain RES 1

Parity bits

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1	0	6
0	1	0	0	0	0	1	0	1	0	1	3
1	0	0	1	1	0	1	0	0	1	1	5
1	1	1	0	0	0	1	0	1	0	1	5
1	0	0	0	0	0	0	0	1	0	0	2
1	1	0	0	1	0	0	0	0	1	0	4
1	0	0	1	0	1	0	1	0	1	1	5
1	1	1	0	1	0	0	0	1	0	1	5
1	0	1	1	0	1	0	1	0	1	0	6
1	0	1	0	0	0	1	1	1	0	1	5
1	0	1	0	1	1	0	1	0	1		
9	4	5	4	3	3	4	3	6	5		

Parity bits

OSI Reference Model ▶ CRC

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1	0
0	1	0	0	0	0	1	0	1	0	1
1	0	0	1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0	1	0	1
1	0	0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0	1	0
1	0	0	1	0	1	0	1	0	1	1
1	1	1	0	1	0	0	0	1	0	1
1	0	1	1	0	1	0	1	0	1	0
1	0	1	0	0	0	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1	

Parity bits

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1	0
0	1	0	0	0	0	1	0	1	0	1
1	0	0	1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	1	1	0	1
1	0	0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0	1	0
1	0	0	1	0	1	0	1	0	1	1
1	1	1	0	1	0	0	0	1	0	1
1	0	1	1	0	1	0	1	0	1	0
1	0	1	0	0	0	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1	

Parity bits

OSI Reference Model ▶ CRC

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1	0
0	1	0	0	0	0	1	0	1	0	1
1	0	0	1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	1	1	0	1
1	0	0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0	1	0
1	0	0	1	0	1	0	1	0	1	1
1	1	1	0	1	0	0	0	1	0	1
1	0	1	1	0	1	0	1	0	1	0
1	0	1	0	0	0	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1	

Slide 36 Laurent Toutain RES 1 TELECOM Bretagne

Parity bits

- Use redundancy information to correct or to detect inconsistencies
- Simplest solution is parity bit

1	0	1	1	0	1	0	0	1	1	0
0	1	0	0	0	0	1	0	1	0	1
1	0	0	1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0	1	0	1
1	0	0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0	1	0
1	0	0	1	0	1	0	1	0	1	1
1	1	1	0	1	0	0	0	1	0	1
1	0	1	1	0	1	0	1	0	1	0
1	0	1	0	0	0	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1	

Cyclic Redondancy Check

- use modulo 2 arithmetic:

+	0	1	-	0	1
0	0	1	0	0	1
1	1	0	1	1	0

- Based on polynomial division
- A binary value can be represented as a polynom
  - 01110100100111001
- All devices uses a well-known generating polynom usually call $G(x)$

Cyclic Redondancy Check

- use modulo 2 arithmetic:

+	0	1	-	0	1
0	0	1	0	0	1
1	1	0	1	1	0

- Based on polynomial division
- A binary value can be represented as a polynom
  - 01110100100111001
  - $0.x^{16} + 1.x^{15} + 1.x^{14} + 1.x^{13} + 0.x^{12} + 1.x^{11} + 0.x^{10} + 0.x^9 + 1.x^8 + 0.x^7 + 0.x^6 + 1.x^5 + 1.x^4 + 1.x^3 + 0.x^2 + 0.x^1 + 1.x^0$
- All devices uses a well-known generating polynom usually call $G(x)$

Cyclic Redondancy Check

OSI Reference Model ▶ CRC

- use modulo 2 arithmetic:

+	0	1	-	0	1
0	0	1	0	0	1
1	1	0	1	1	0

- Based on polynomial division
- A binary value can be represented as a polynom
  - 01110100100111001
  - $0.x^{16} + 1.x^{15} + 1.x^{14} + 1.x^{13} + 0.x^{12} + 1.x^{11} + 0.x^{10} + 0.x^9 + 1.x^8 + 0.x^7 + 0.x^6 + 1.x^5 + 1.x^4 + 1.x^3 + 0.x^2 + 0.x^1 + 1.x^0$
  - $x^{15} + x^{14} + x^{13} + x^{11} + x^8 + x^5 + x^4 + x^3 + 1$
- All devices uses a well-known generating polynom usually call $G(x)$

Slide 37 Laurent Toutain RES 1 TELECOM Bretagne

CRC: Algorithm

1. Initial sequence $I(x)$

CRC: Algorithm

OSI Reference Model ▶ CRC

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$

CRC: Algorithm

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits

CRC: Algorithm

OSI Reference Model ▶ CRC

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$

CRC: Algorithm

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$
   - $I(x) = N(x).G(x) + R(x)$

CRC: Algorithm

OSI Reference Model ▶ CRC

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$
   - $I(x) = N(x).G(x) + R(x)$
   - note that highest degree of $R(x)$ is less than r

CRC: Algorithm

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$
   - $I(x) = N(x).G(x) + R(x)$
   - note that highest degree of $R(x)$ is less than r
4. add $R(x)$

CRC: Algorithm

OSI Reference Model ▶ CRC

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$
   - $I(x) = N(x).G(x) + R(x)$
   - note that highest degree of $R(x)$ is less than r
4. add $R(x)$
   - $I(x) + R(x) = N(x).G(x) + R(x) + R(x) = N(x).G(x)$

CRC: Algorithm

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$
   - $I(x) = N(x).G(x) + R(x)$
   - note that highest degree of $R(x)$ is less than r
4. add $R(x)$
   - $I(x) + R(x) = N(x).G(x) + R(x) + R(x) = N(x).G(x)$
   - $I(x)$ is intact

CRC: Algorithm

OSI Reference Model ▶ CRC

1. Initial sequence $I(x)$
2. Multiply by x^r where r is the highest degree in $G(x)$
   - Equivalent to a shift of r bits
3. divide $I(x)$ by $G(x)$
   - $I(x) = N(x).G(x) + R(x)$
   - note that highest degree of $R(x)$ is less than r
4. add $R(x)$
   - $I(x) + R(x) = N(x).G(x) + R(x) + R(x) = N(x).G(x)$
   - $I(x)$ is intact
   - $R(x)$ is the CRC

CRC: Example

$x^2 + 1$

OSI Reference Model ▶ CRC

CRC: Example

$x^{15} + x^{14} + x^{13} + x^{11} + x^8 + x^5 + x^4 + x^3 + 1$ | $x^2 + 1$

Slide 39 Laurent Toutain RES 1

CRC: Example

$$x^{17} + x^{16} + x^{15} + x^{13} + x^{10} + x^7 + x^6 + x^5 + x^2 \quad \Big| \quad x^2 + 1$$

CRC: Example

OSI Reference Model ▶ CRC

$x^{17} + x^{16} + x^{15} + x^{13} + x^{10} + x^7 + x^6 + x^5 + x^2 \quad | \quad x^2 + 1$

x^{15}

Slide 39 Laurent Toutain RES 1

CRC: Example

$$\begin{array}{l} x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} \\ x^{17}+x^{15} \end{array} \quad \Big| \; x^2+1$$

x^{15}

CRC: Example

OSI Reference Model ▶ CRC

$$\begin{array}{l} x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} \\ x^{17}+x^{15} \\ \hline x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} \end{array} \quad \Big| \; x^2+1$$

x^{15}

CRC: Example

$$
\begin{array}{r|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\underline{x^{17}+x^{15}\phantom{+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}\phantom{+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{15}+x^{14}
\end{array}
$$

CRC: Example

OSI Reference Model ▶ CRC

$$
\begin{array}{r|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\underline{x^{17}+x^{15}\phantom{+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}\phantom{+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{15}+x^{14}+x^{12} \\
\underline{x^{14}+x^{12}\phantom{+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} &
\end{array}
$$

Slide 39 Laurent Toutain RES 1

CRC: Example

$$
\begin{array}{rl|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & x^{2}+1 \\
\cline{3-3}
x^{17}+x^{15} & & \\
\cline{1-1}
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & \\
x^{16}+x^{14} & & \\
\cline{1-1}
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & x^{15}+x^{14}+x^{12}+x^{11} \\
x^{14}+x^{12} & & \\
\cline{1-1}
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & \\
x^{13}+x^{11} & & \\
\cline{1-1}
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & &
\end{array}
$$

CRC: Example

OSI Reference Model ▶ CRC

$$
\begin{array}{rl|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & x^{2}+1 \\
\cline{3-3}
x^{17}+x^{15} & & \\
\cline{1-1}
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & \\
x^{16}+x^{14} & & \\
\cline{1-1}
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & x^{15}+x^{14}+x^{12}+x^{11}+x^{10} \\
x^{14}+x^{12} & & \\
\cline{1-1}
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & \\
x^{13}+x^{11} & & \\
\cline{1-1}
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & & \\
x^{12}+x^{10} & & \\
\cline{1-1}
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & &
\end{array}
$$

CRC: Example

$$
\begin{array}{r|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\cline{2-2}
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9} \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} &
\end{array}
$$

CRC: Example

$$
\begin{array}{r|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\cline{2-2}
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9}+x^{7} \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2} &
\end{array}
$$

CRC: Example

$$
\begin{array}{l|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9}+x^{7}+x^{4} \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2} & \\
\underline{x^{6}+x^{4}} & \\
x^{5}+x^{4}+x^{2} &
\end{array}
$$

CRC: Example

OSI Reference Model ▶ CRC

$$
\begin{array}{l|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9}+x^{7}+x^{4}+x^{3} \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2} & \\
\underline{x^{6}+x^{4}} & \\
x^{5}+x^{4}+x^{2} & \\
\underline{x^{5}+x^{3}} & \\
x^{4}+x^{3}+x^{2} &
\end{array}
$$

CRC: Example

$$
\begin{array}{r|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2} & \\
\underline{x^{6}+x^{4}} & \\
x^{5}+x^{4}+x^{2} & \\
\underline{x^{5}+x^{3}} & \\
x^{4}+x^{3}+x^{2} & \\
\underline{x^{4}+x^{2}} & \\
x^{3} &
\end{array}
\qquad x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9}+x^{7}+x^{4}+x^{3}+x^{2}
$$

CRC: Example

OSI Reference Model ▶ CRC

$$
\begin{array}{r|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^{2}+1 \\
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2} & \\
\underline{x^{6}+x^{4}} & \\
x^{5}+x^{4}+x^{2} & \\
\underline{x^{5}+x^{3}} & \\
x^{4}+x^{3}+x^{2} & \\
\underline{x^{4}+x^{2}} & \\
x^{3}+x & \\
\underline{x^{3}} & \\
x &
\end{array}
\qquad x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9}+x^{7}+x^{4}+x^{3}+x^{2}+x
$$

CRC: Example

$$
\begin{array}{l|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & x^2+1 \\
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{16}+x^{14}} & \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2} & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2} & \\
\underline{x^{6}+x^{4}} & \\
x^{5}+x^{4}+x^{2} & \\
\underline{x^{5}+x^{3}} & \\
x^{4}+x^{3}+x^{2} & \\
\underline{x^{4}+x^{2}} & \\
x^{3} & \\
\underline{x^{3}+x} & \\
x &
\end{array}
$$

send: $x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}+x$
$\Rightarrow$01110100100111001**10**

CRC: Example (reception)

$$
\begin{array}{l|l}
x^{17}+x^{16}+x^{15}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}+x & x^2+1 \\
\underline{x^{17}+x^{15}} & \\
x^{16}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{16}+x^{14}} & x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{9}+x^{7}+x^{4}+x^{3}+x^{2}+x \\
x^{14}+x^{13}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{14}+x^{12}} & \\
x^{13}+x^{12}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{13}+x^{11}} & \\
x^{12}+x^{11}+x^{10}+x^{7}+x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{12}+x^{10}} & \\
x^{11}+x^{7}+x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{11}+x^{9}} & \\
x^{9}+x^{7}+x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{9}+x^{7}} & \\
x^{6}+x^{5}+x^{2}+x & \\
\underline{x^{6}+x^{4}} & \\
x^{5}+x^{4}+x^{2}+x & \\
\underline{x^{5}+x^{3}} & \\
x^{4}+x^{3}+x^{2}+x & \\
\underline{x^{4}+x^{2}} & \\
x^{3}+x & \\
\underline{x^{3}+x} & \\
0 &
\end{array}
$$

CRC: Example (reception)

0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1

CRC: Example (reception)

OSI Reference Model ▶ CRC

0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0 Room for CRC

CRC: Example (reception)


```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
```

step 1: align to the left generating polynom

CRC: Example (reception)

OSI Reference Model ▶ CRC

```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
```

step 1: align to the left generating polynom

step 2 : send bits on the left of GP

Slide 41 Laurent Toutain RES 1

CRC: Example (reception)


```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
```

step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

CRC: Example (reception)

OSI Reference Model ▶ CRC

```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
```

step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

CRC: Example (reception)


```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
```

step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

CRC: Example (reception)

OSI Reference Model ▶ CRC

```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
```

step 1: align to the left generating polynom

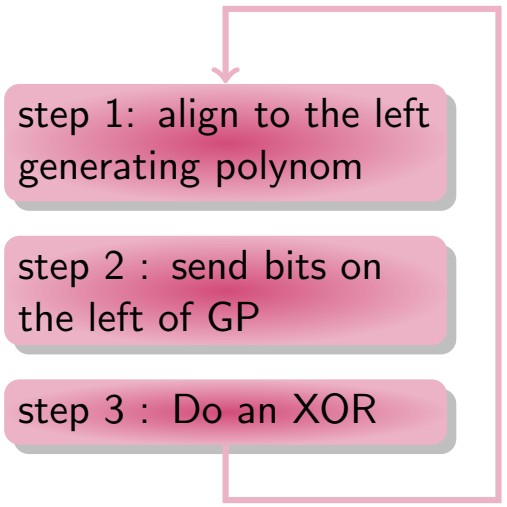
step 2 : send bits on the left of GP

step 3 : Do an XOR

Slide 41 Laurent Toutain RES 1

CRC: Example (reception)


```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
          1 0 1
            1 1 1 0 0 1 1 1 0 0 1 0 0
```



CRC: Example (reception)

OSI Reference Model ▶ CRC

```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
          1 0 1
            1 1 1 0 0 1 1 1 0 0 1 0 0
              1 0 1
                1 0 0 0 1 1 1 0 0 1 0 0
```

step 1: align to the left generating polynom
step 2 : send bits on the left of GP
step 3 : Do an XOR

Slide 41 Laurent Toutain RES 1

CRC: Example (reception)


```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
          1 0 1
            1 1 1 0 0 1 1 1 0 0 1 0 0
            1 0 1
              1 0 0 0 1 1 1 0 0 1 0 0
              1 0 1
                1 0 1 1 1 0 0 1 0 0
```

step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

CRC: Example (reception)

OSI Reference Model ▶ CRC

```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
          1 0 1
            1 1 1 0 0 1 1 1 0 0 1 0 0
            1 0 1
              1 0 0 0 1 1 1 0 0 1 0 0
              1 0 1
                1 0 1 1 1 0 0 1 0 0
                1 0 1
                    1 1 0 0 1 0 0
```

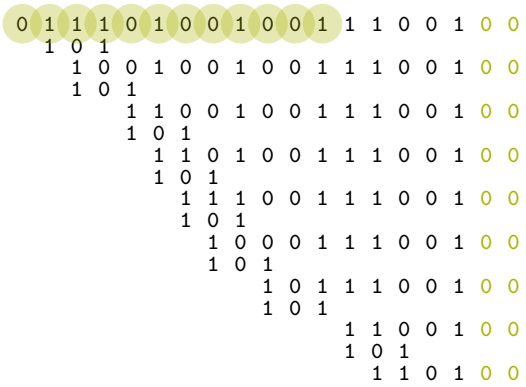
step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

Slide 41 Laurent Toutain RES 1

CRC: Example (reception)



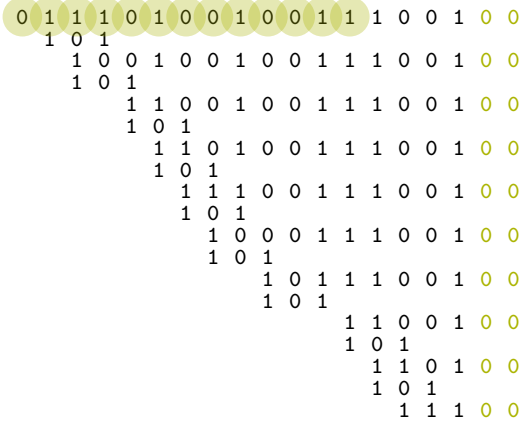
step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

CRC: Example (reception)

OSI Reference Model ▶ CRC



step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

CRC: Example (reception)


```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
          1 0 1
            1 1 1 0 0 1 1 1 0 0 1 0 0
            1 0 1
              1 0 0 0 1 1 1 0 0 1 0 0
              1 0 1
                1 0 1 1 1 0 0 1 0 0
                1 0 1
                      1 1 0 0 1 0 0
                      1 0 1
                        1 1 0 1 0 0
                        1 0 1
                          1 1 1 0 0
                          1 0 1
                            1 0 0 0
```

- step 1: align to the left generating polynom
- step 2 : send bits on the left of GP
- step 3 : Do an XOR

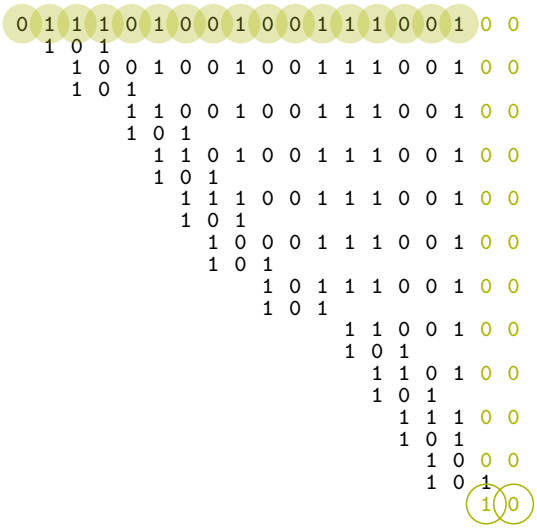
CRC: Example (reception)

OSI Reference Model ▶ CRC

```
0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
  1 0 1
    1 0 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0
    1 0 1
        1 1 0 0 1 0 0 1 1 1 0 0 1 0 0
        1 0 1
          1 1 0 1 0 0 1 1 1 0 0 1 0 0
          1 0 1
            1 1 1 0 0 1 1 1 0 0 1 0 0
            1 0 1
              1 0 0 0 1 1 1 0 0 1 0 0
              1 0 1
                1 0 1 1 1 0 0 1 0 0
                1 0 1
                      1 1 0 0 1 0 0
                      1 0 1
                        1 1 0 1 0 0
                        1 0 1
                          1 1 1 0 0
                          1 0 1
                            1 0 0 0
                            1 0 1 0
                                  1 0
```

- step 1: align to the left generating polynom
- step 2 : send bits on the left of GP
- step 3 : Do an XOR

CRC: Example (reception)



step 1: align to the left generating polynom

step 2 : send bits on the left of GP

step 3 : Do an XOR

OSI Reference Model

Send and Wait Protocol

Send & Wait

- Sender:
  1. Wait for data from upper layers
  2. Take the sequence to transmit
  3. Compute CRC
  4. Send data and trigger timer
  5. if received Ack remove data from memory, goto to step 1
  6. if timer expires, go to step 4
- Receiver:
  1. Wait frames from physical layer
  2. Validate CRC
  3. if CRC is correct send data to upper layer and go to step 1
  4. if CRC is wrong, discard information and go to step 1

Send & Wait

OSI Reference Model ▶ Send and Wait Protocol

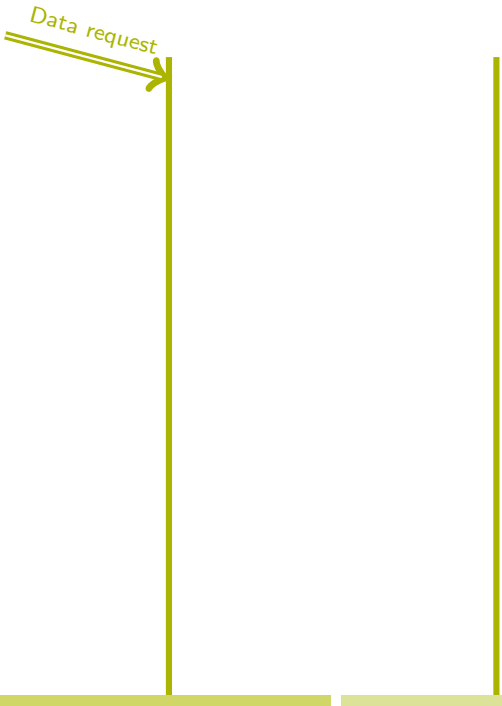
- Sender:
  1. Wait for data from upper layers
  2. Take the sequence to transmit
  3. Compute CRC
  4. Send data and trigger timer
  5. if received Ack remove data from memory, goto to step 1
  6. if timer expires, go to step 4
- Receiver:
  1. Wait frames from physical layer
  2. Validate CRC
  3. if CRC is correct send data to upper layer and go to step 1
  4. if CRC is wrong, discard information and go to step 1
- No frame losses, frames are discarded if CRC is wrong: Do not react on wrong information

Slide 43 Laurent Toutain RES 1 TELECOM Bretagne

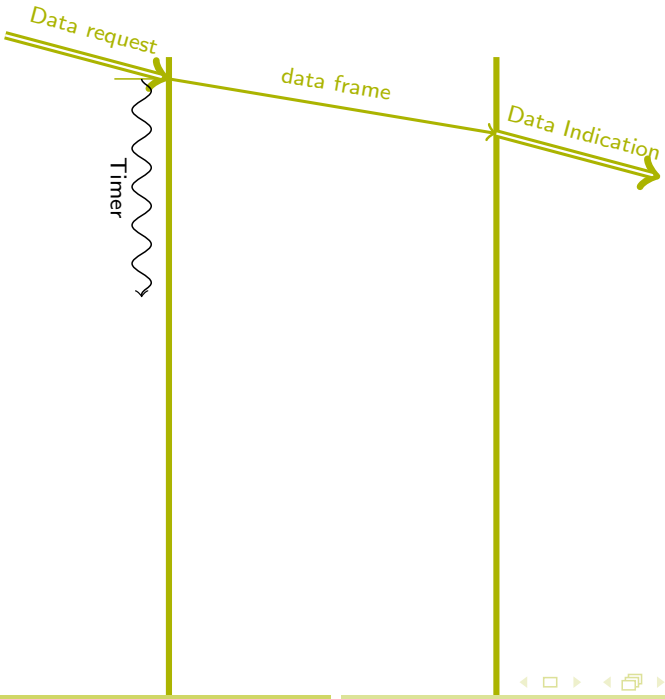
Send & Wait

- Sender:
  1. Wait for data from upper layers
  2. Take the sequence to transmit
  3. Compute CRC
  4. Send data and trigger timer
  5. if received Ack remove data from memory, goto to step 1
  6. if timer expires, go to step 4
- Receiver:
  1. Wait frames from physical layer
  2. Validate CRC
  3. if CRC is correct send data to upper layer and go to step 1
  4. if CRC is wrong, discard information and go to step 1
- No frame losses, frames are discarded if CRC is wrong: Do not react on wrong information
- Use positive Ack when something correct is received

Send & Wait

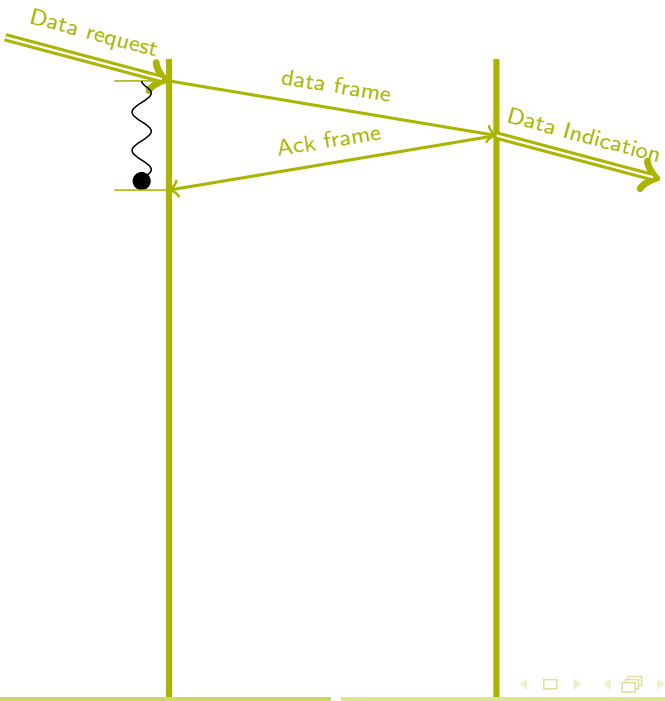


Send & Wait

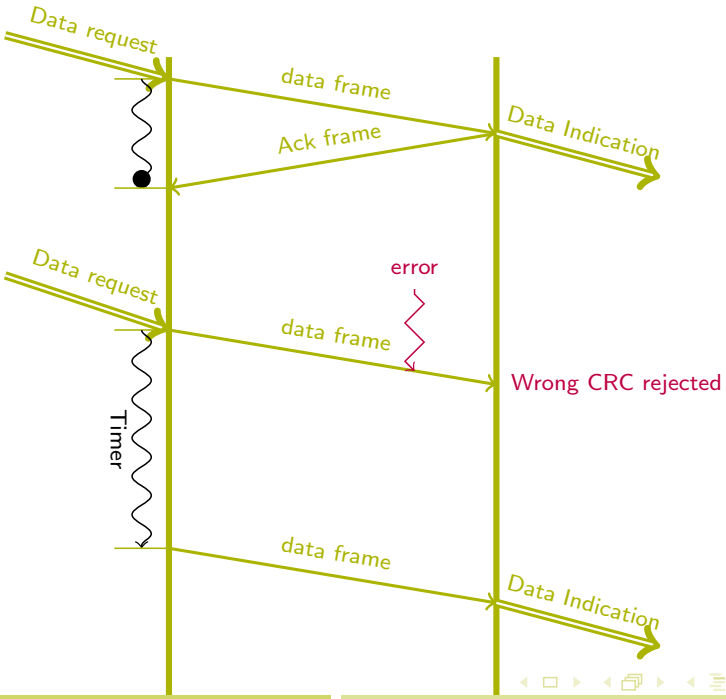


Send & Wait

OSI Reference Model ▶ Send and Wait Protocol

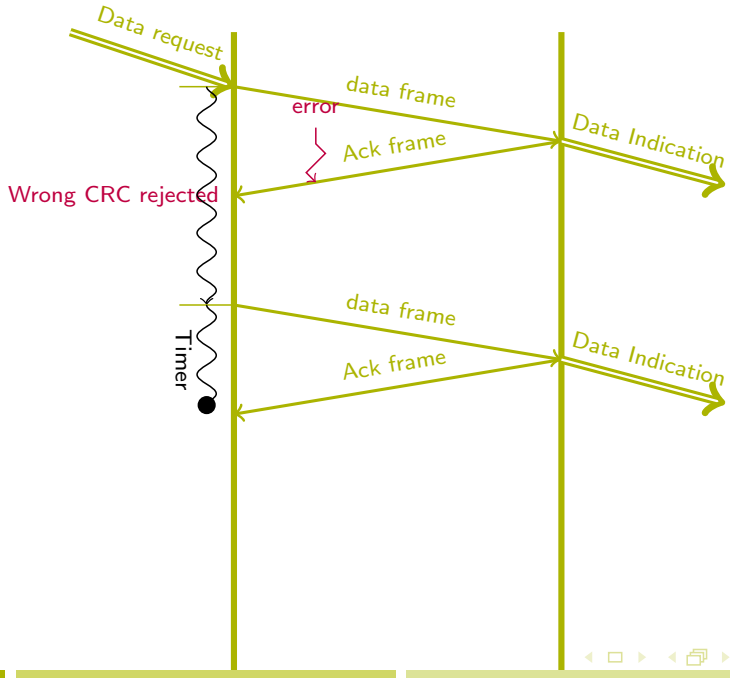


Send & Wait

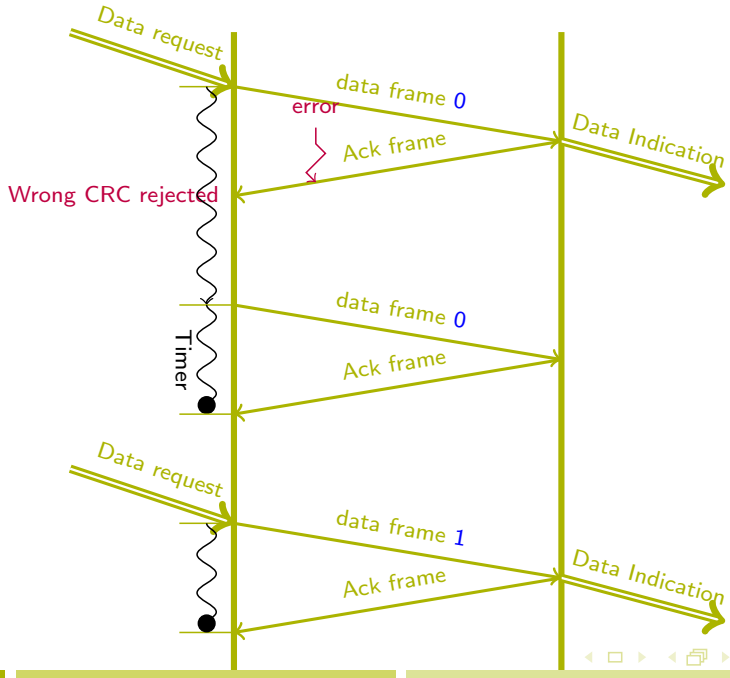


Send & Wait

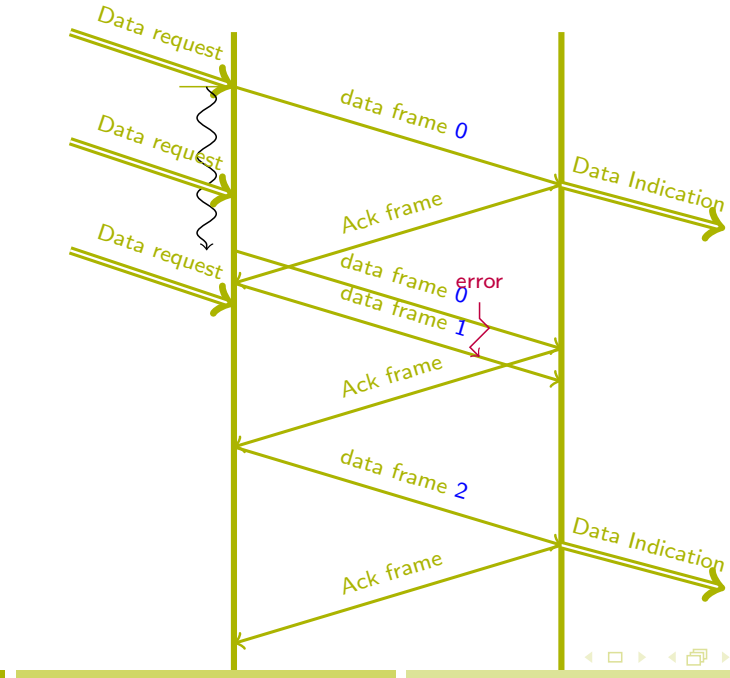
OSI Reference Model ▶ Send and Wait Protocol



Send & Wait



Send & Wait (Timer underestimated)



Formalization

- What will be the size of the data counter ?
- Give a detailed description of the enhanced version of the protocol to allow implementers to write interoperating applications

Formalization

- What will be the size of the data counter ? 1 bit
- Give a detailed description of the enhanced version of the protocol to allow implementers to write interoperating applications

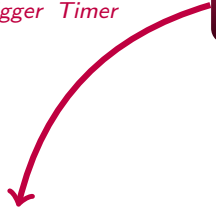
Very complex to describe with words, a more formal language will be better.

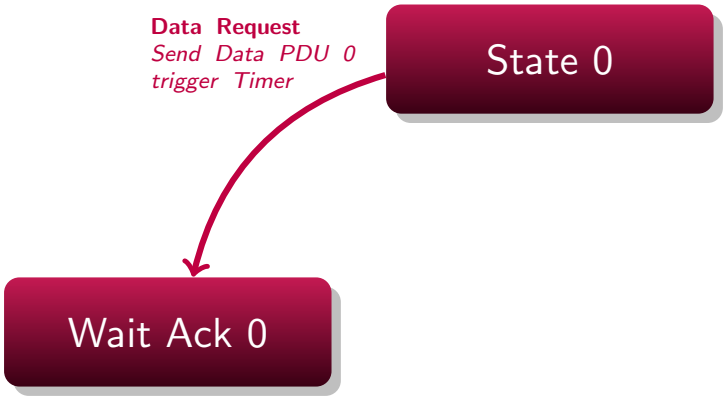
State 0

Formalization
OSI Reference Model ▶ Send and Wait Protocol

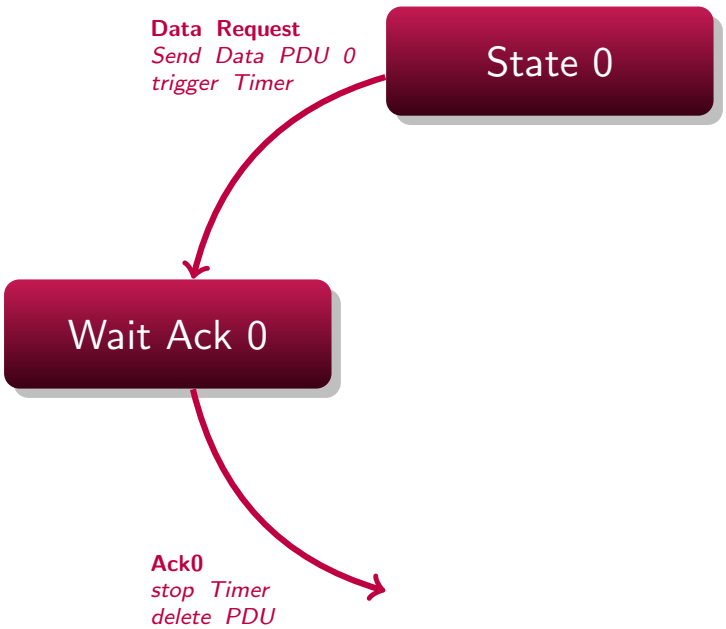
Data Request
Send Data PDU 0
trigger Timer

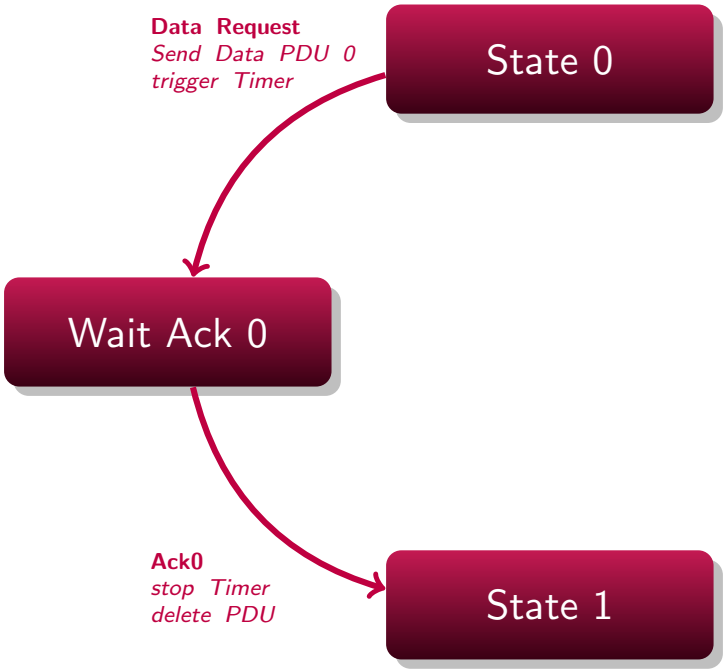
State 0



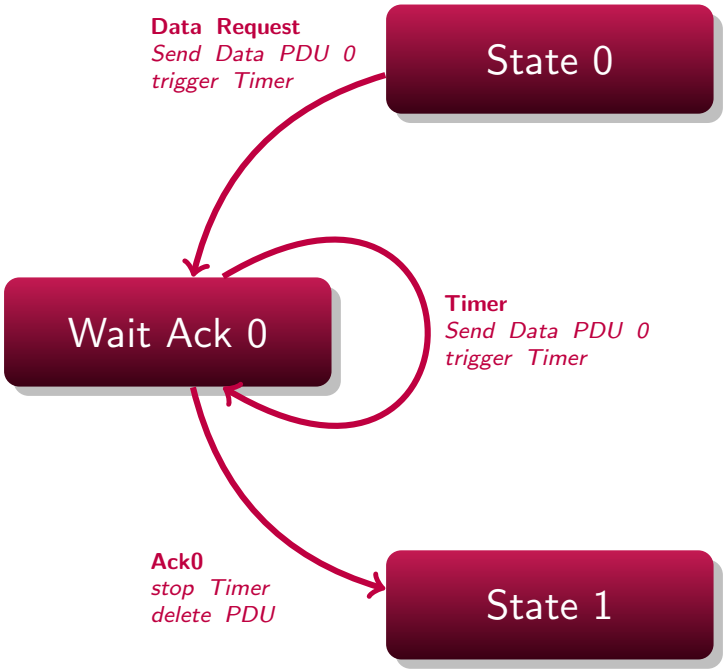


Formalization
OSI Reference Model ▶ Send and Wait Protocol

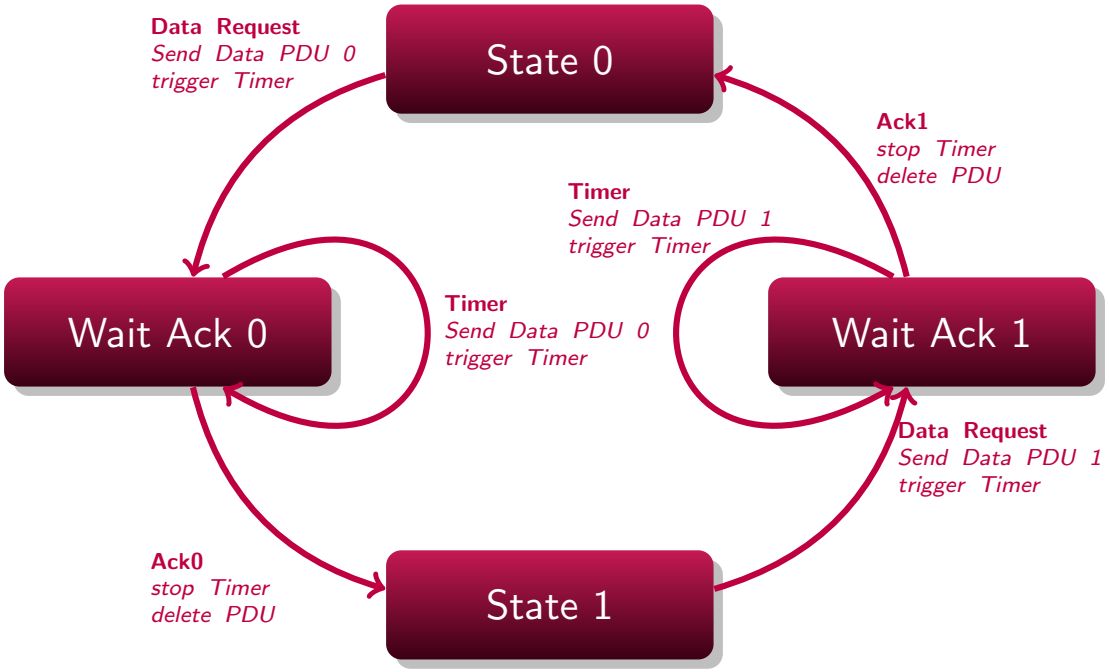




Formalization
OSI Reference Model ▶ Send and Wait Protocol



Formalization



Formalization

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request				
Ack 0				
Ack 1				
Timer				

Formalization

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 0**			
Ack 0				
Ack 1				
Timer				

Formalization

OSI Reference Model ▶ Send and Wait Protocol

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 0**			
Ack 0		*stop Timer and delete PDU.* **Goto State 1**		
Ack 1				
Timer				

Formalization

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 0**			
Ack 0		*stop Timer and delete PDU.* **Goto State 1**		
Ack 1				
Timer		*Send Data PDU 0 and trigger Timer* **Goto Wait Ack 0**		

Formalization

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 0**		*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 1**	
Ack 0		*stop Timer and delete PDU.* **Goto State 1**		
Ack 1				*stop Timer and delete PDU.* **Goto State 0**
Timer		*Send Data PDU 0 and trigger Timer* **Goto Wait Ack 0**		*Send Data PDU 1 and trigger Timer* **Goto Wait Ack 1**

Formalization

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 0**	*Memorize* **Goto Wait Ack 0**	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 1**	*Memorize* **Goto Wait Ack 1**
Ack 0		*stop Timer and delete PDU.* **Goto State 1**		
Ack 1				*stop Timer and delete PDU.* **Goto State 0**
Timer		*Send Data PDU 0 and trigger Timer* **Goto Wait Ack 0**		*Send Data PDU 1 and trigger Timer* **Goto Wait Ack 1**

Formalization

OSI Reference Model ▶ Send and Wait Protocol

Event/State	State 0	Wait Ack0	State 1	Wait Ack1
Data Request	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 0**	*Memorize* **Goto Wait Ack 0**	*Send Data PDU 0 and trigger Timer.* **Goto Wait Ack 1**	*Memorize* **Goto Wait Ack 1**
Ack 0	*Discard* **Goto State 0**	*stop Timer and delete PDU.* **Goto State 1**	*Discard* **Goto State 1**	*Discard* **Goto Wait Ack 1**
Ack 1	*Discard* **Goto State 0**	*Discard* **Goto Wait Ack 0**	*Discard* **Goto State 1**	*stop Timer and delete PDU.* **Goto State 0**
Timer	*Ignore* **Goto State 0**	*Send Data PDU 0 and trigger Timer* **Goto Wait Ack 0**	*Ignore* **Goto State 1**	*Send Data PDU 1 and trigger Timer* **Goto Wait Ack 1**

Protocol performances

Define a PDU format for Send and Wait Protocol

How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.

Protocol performances

OSI Reference Model ▶ Send and Wait Protocol

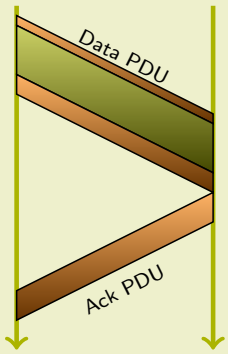
Define a PDU format for Send and Wait Protocol

How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.

Protocol performances

Define a PDU format for Send and Wait Protocol

How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.

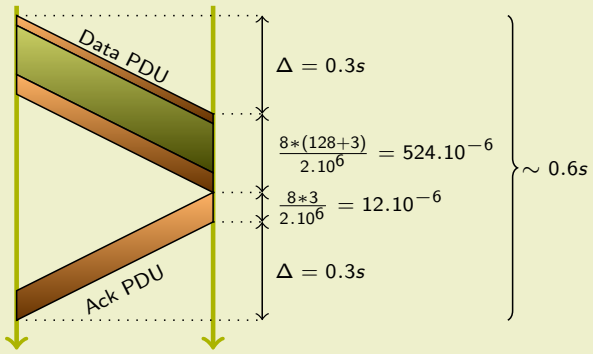


Protocol performances

OSI Reference Model ▶ Send and Wait Protocol

Define a PDU format for Send and Wait Protocol

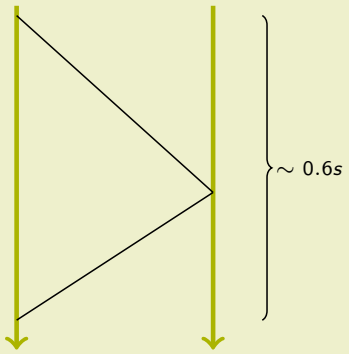
How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.



Protocol performances

Define a PDU format for Send and Wait Protocol

How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.

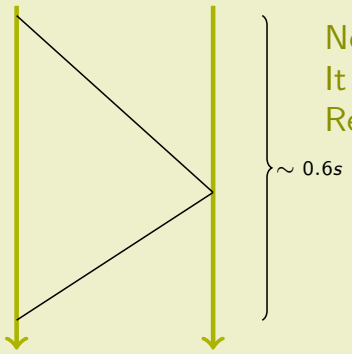


Protocol performances

OSI Reference Model ▶ Send and Wait Protocol

Define a PDU format for Send and Wait Protocol

How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.

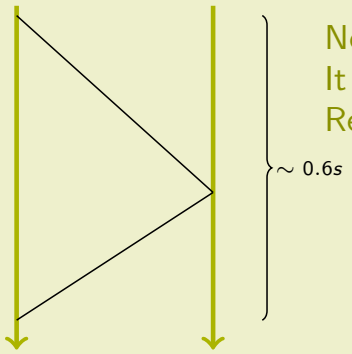


Need $\frac{2.1024}{128} = 16$ exchanges to send the complete file.
It will takes $16 * 0.6 = 9.6s$ to send the file.
Real throughput is $\frac{2.1024}{9.6} = 213bit/s$

Protocol performances

Define a PDU format for Send and Wait Protocol

How long does it takes to send a 2 Kbit file using Send and Wait protocol? Propagation delay $\Delta = 0.3s$, Throughput $T = 2MBit/s$, SDU size $S = 128Bytes$. There is NO transmission error.



Need $\frac{2.1024}{128} = 16$ exchanges to send the complete file.
It will takes $16 * 0.6 = 9.6s$ to send the file.
Real throughput is $\frac{2.1024}{9.6} = 213bit/s$

Protocol is inefficient since it spends time waiting for frames

Protocol performances

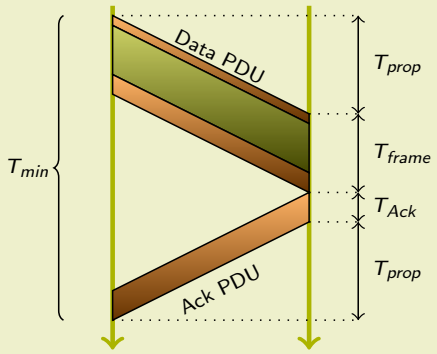
Let p the probability to have a error on a Data PDU, p_r the probability of an error on a Ack PDU, T_{frame} the time needed to send a Data PDU and T_{Ack}, the time needed to send an Ack PDU, T_1 is the timer delay used to retransmit lost PDU and T_{prop} is the propagation delay between the two entities using Send and Wait protocol to exchange data.
1) What is the T_{min} time to send correctly a PDU
2) Define P_{succ} the probability to send correctly a PDU
3) Give the total time T_t it will take to send correctly a PDU

$\sum_{i=0}^{\infty}(1-x)^i = \frac{1}{x}$ $\sum_{i=0}^{\infty} i.(1-x)^i = \frac{x}{(1-x)^2}$

Protocol performances

Let p the probability to have a error on a Data PDU, p_r the probability of an error on a Ack PDU, T_{frame} the time needed to send a Data PDU and T_{Ack}, the time needed to send an Ack PDU, T_1 is the timer delay used to retransmit lost PDU and T_{prop} is the propagation delay between the two entities using Send and Wait protocol to exchange data.

1) What is the T_{min} time to send correctly a PDU
2) Define P_{succ} the probability to send correctly a PDU
3) Give the total time T_t it will take to send correctly a PDU



$$T_{min} = T_{frame} + T_{Ack} + 2.T_{prop}$$

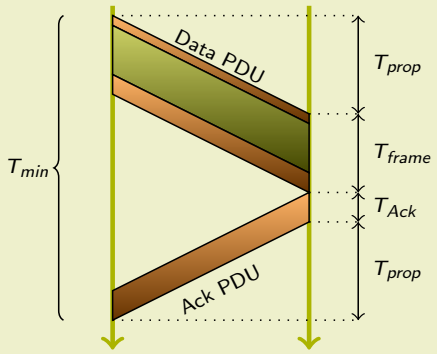
$\sum_{i=0}^{\infty}(1-x)^i = \frac{1}{x}$ $\sum_{i=0}^{\infty} i.(1-x)^i = \frac{x}{(1-x)^2}$

Protocol performances

OSI Reference Model ▶ Send and Wait Protocol

Let p the probability to have a error on a Data PDU, p_r the probability of an error on a Ack PDU, T_{frame} the time needed to send a Data PDU and T_{Ack}, the time needed to send an Ack PDU, T_1 is the timer delay used to retransmit lost PDU and T_{prop} is the propagation delay between the two entities using Send and Wait protocol to exchange data.

1) What is the T_{min} time to send correctly a PDU
2) Define P_{succ} the probability to send correctly a PDU
3) Give the total time T_t it will take to send correctly a PDU



$$T_{min} = T_{frame} + T_{Ack} + 2.T_{prop}$$

$$P_{succ} = (1-p)(1-p_r)$$

$\sum_{i=0}^{\infty}(1-x)^i = \frac{1}{x}$ $\sum_{i=0}^{\infty} i.(1-x)^i = \frac{x}{(1-x)^2}$

Protocol performances

Let p the probability to have a error on a Data PDU, p_r the probability of an error on a Ack PDU, T_{frame} the time needed to send a Data PDU and T_{Ack}, the time needed to send an Ack PDU, T_1 is the timer delay used to retransmit lost PDU and T_{prop} is the propagation delay between the two entities using Send and Wait protocol to exchange data.

1) What is the T_{min} time to send correctly a PDU
2) Define P_{succ} the probability to send correctly a PDU
3) Give the total time T_t it will take to send correctly a PDU

$T_t = P_{succ}.T_{min} + (1-P_{succ})P_{succ}.(T1+T_{min}) + \ldots + (1-P_{succ})^i P_{succ}.(i.T1+T_{min}) + \ldots$

$T_t = \sum_{i=0}^{\infty}(1-P_{succ})^i P_{succ}.(i.T1+T_{min})$

$T_t = T1.P_{succ}\sum_{i=0}^{\infty} i.(1-P_{succ})^i + T_{min}.P_{succ}\sum_{i=0}^{\infty}(1-P_{succ})^i$

$T_t = T1.P_{succ}\frac{1-P_{succ}}{{P_{succ}}^2} + T_{min}.P_{succ}.\frac{1}{P_{succ}}$

$T_t = T1.\frac{1-P_{succ}}{P_{succ}} + T_{min}$

$\sum_{i=0}^{\infty}(1-x)^i = \frac{1}{x}$ $\sum_{i=0}^{\infty} i.(1-x)^i = \frac{x}{(1-x)^2}$

OSI Reference Model

HDLC

Anticipation Protocol:HDLC

- Allow do send several frames anticipating acknowledgments
- HDLC: High Level Data Link Control
- Historic protocol, adapted in lots of networks (LAP-DM, PPP,...)
- Two behaviors: Connection oriented and Datagram
- Historically X.25 is using a subset of HDLC, called LAB-B.
- LAP-B is connection oriented

Anticipation Protocol:HDLC

- Physical layer gives a bit-stream.
- first step: signal PDUs in the bit stream
- HDLC use a flag (`0111 1110`) that separate PDUs
- flags must not appears inside PDUs
  - if sent sequence contains sequence `011111` insert a 0
  - if received sequence contains `0111110` remove last 0

Flag in frame

Find frames in :

01111110011100100100110111011011011111001010100100110110110111000101010101111111001001001001110011001100111100111000001000101001011111110

What will be sent to physical layer, if we have the following sequence:

0100101110111110000111111111110111111000101011111

What will be sent to upper layers, if we receive the following sequence :

01111110011111010100111101110111111011110110001111100111011111101

What could be sent, if we have nothing to send?

Flag in frame

Find frames in:

01111110011100100100110111011011011111001010100100110110110111000101010101111111001001001001110011001100111100111000001000101001011111110

What will be sent to physical layer, if we have the following sequence:

0100101110111110000011111011111010111110100010101111101

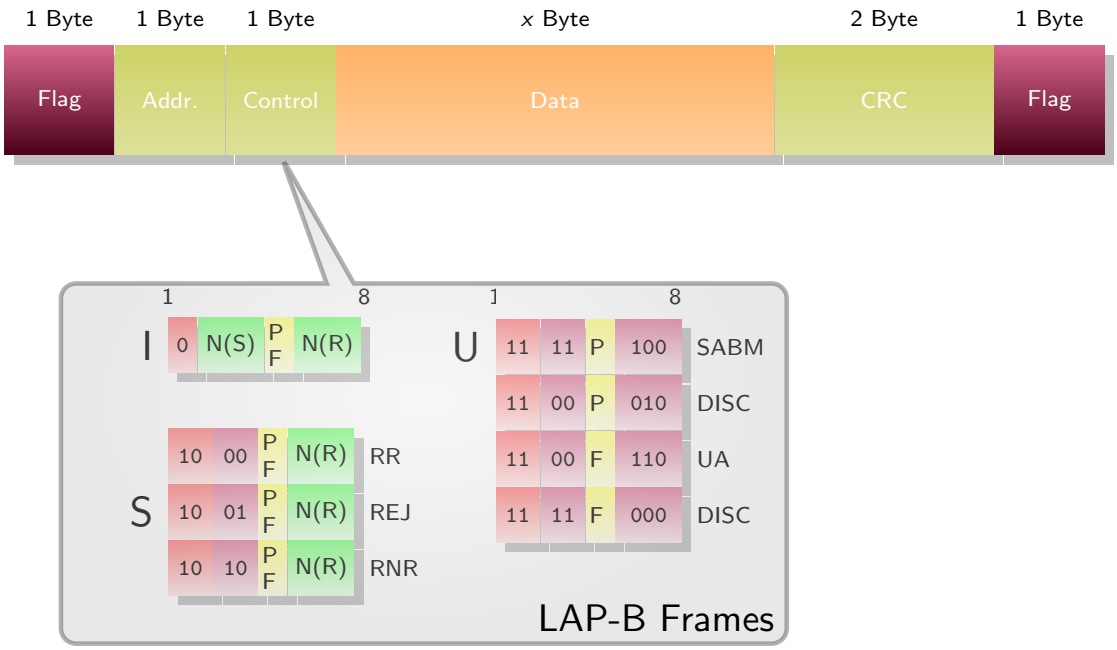
What will be sent to upper layers, if we receive the following sequence :

01111110011111~~0~~10100111101110111111~~0~~11110110001111~~0~~0111011111110

What could be sent, if we have nothing to send?

01111110011111001111110

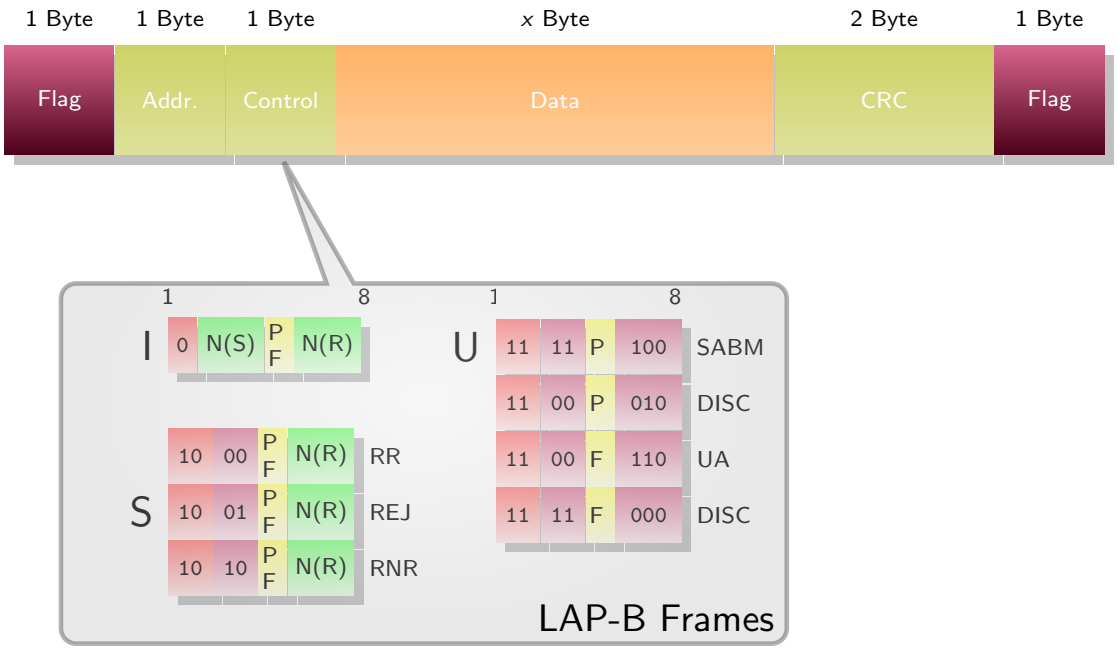
HDLC Frame Format



What field is missing in the HDLC frame ?

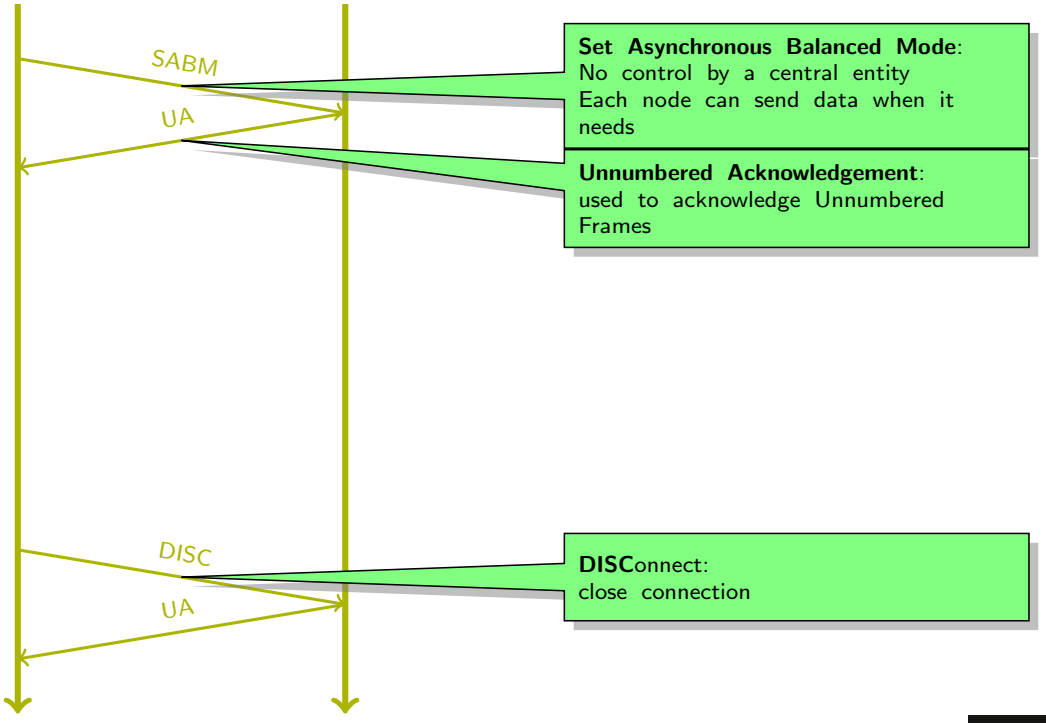
HDLC Frame Format

OSI Reference Model ▶ HDLC



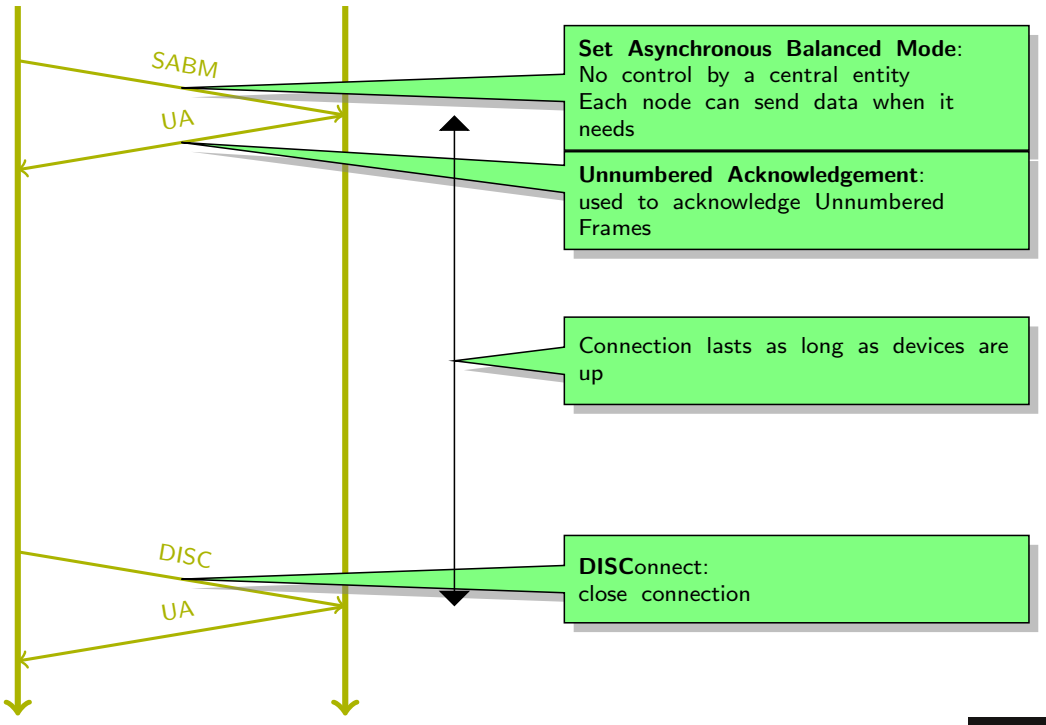
What field is missing in the HDLC frame ? Upper Layer Protocol

Unnumbered Frames

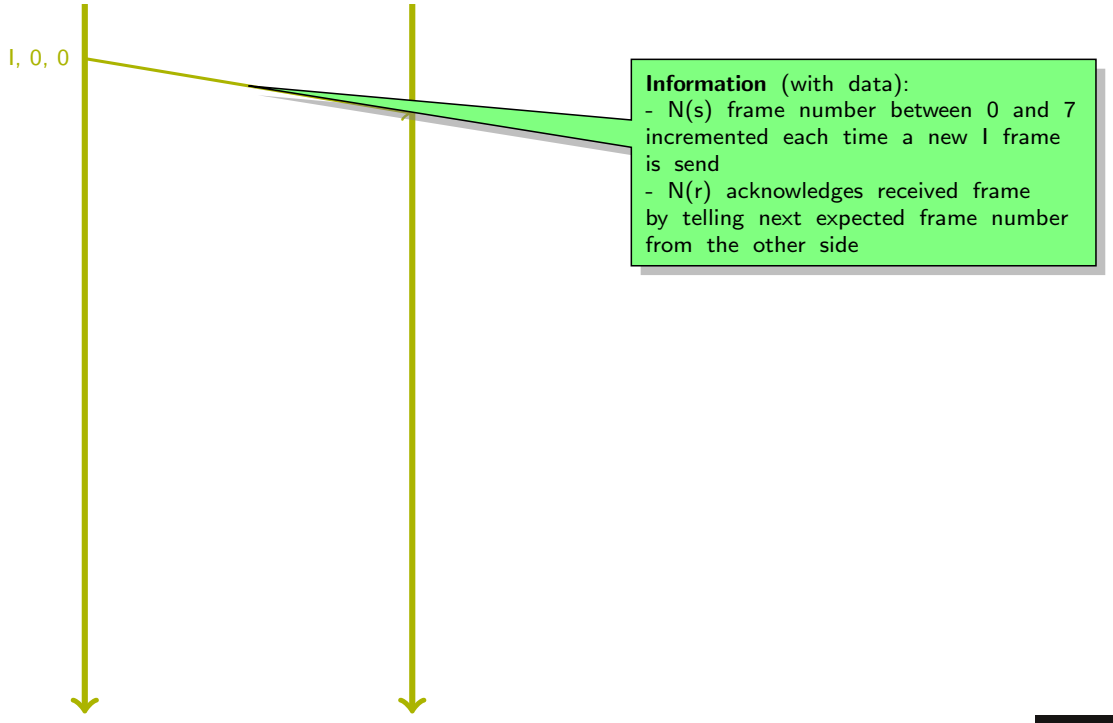


Unnumbered Frames

OSI Reference Model ▶ HDLC

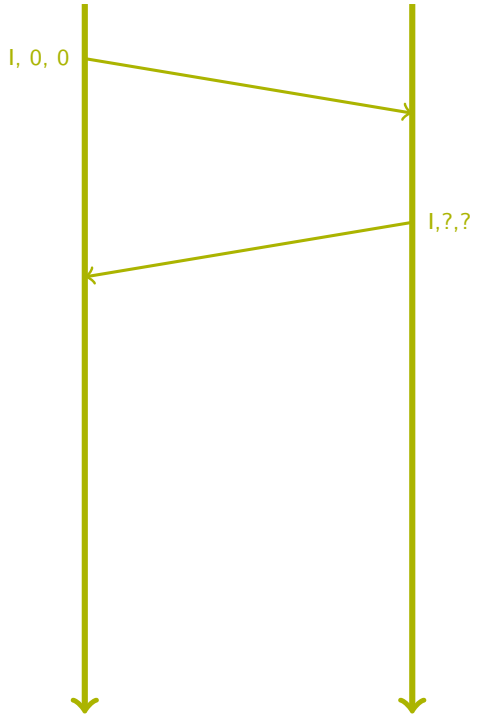


Unnumbered Frames



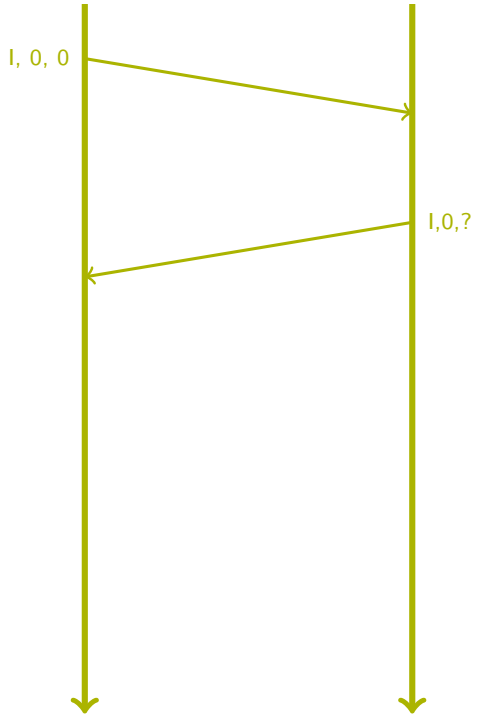
Unnumbered Frames

OSI Reference Model ▶ HDLC



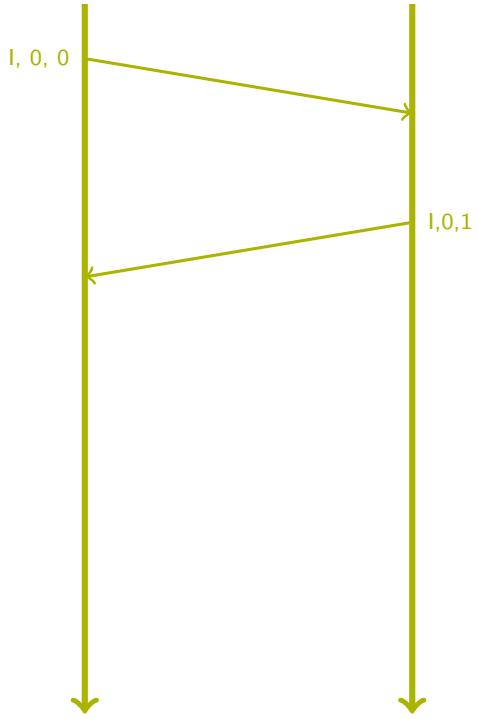
Slide 62 Laurent Toutain RES 1

Unnumbered Frames

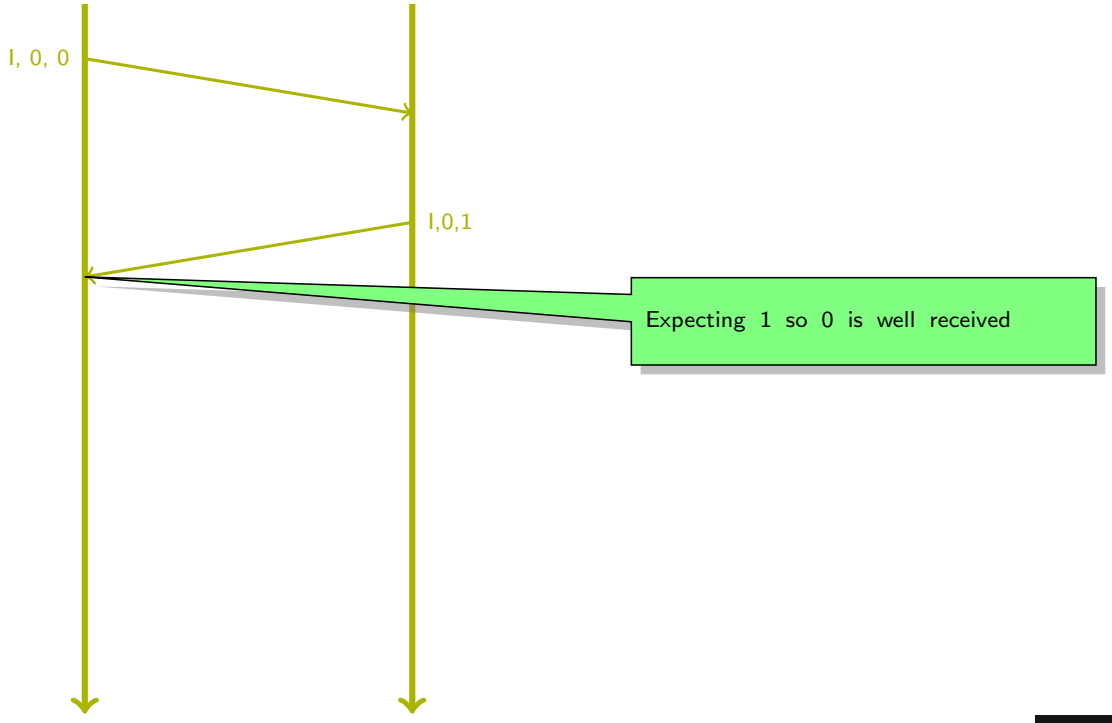


Unnumbered Frames

OSI Reference Model ▶ HDLC

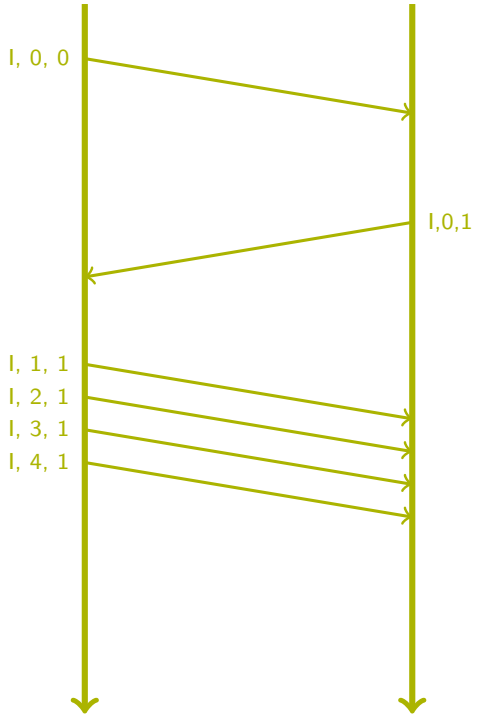


Unnumbered Frames



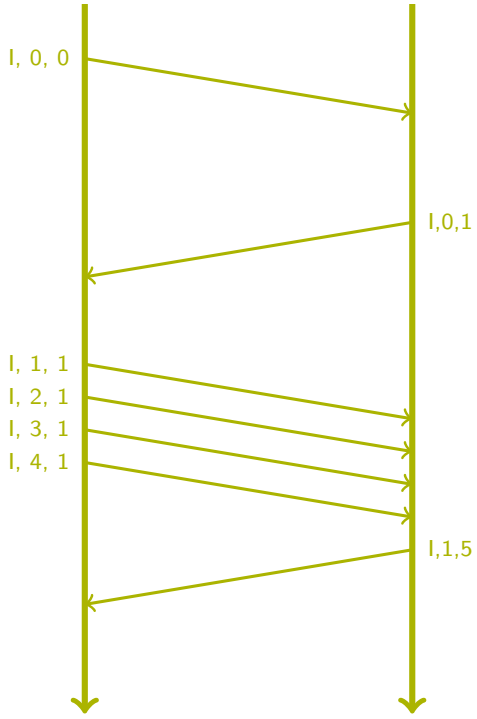
Unnumbered Frames

OSI Reference Model ▶ HDLC



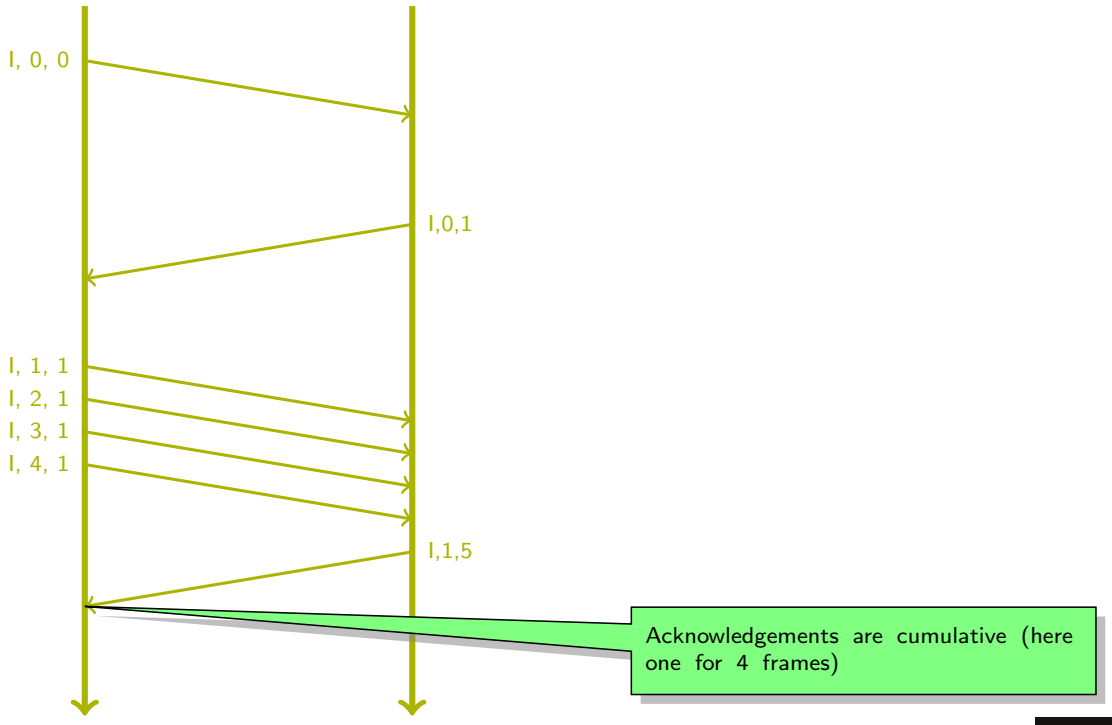
Slide 62 Laurent Toutain RES 1

Unnumbered Frames



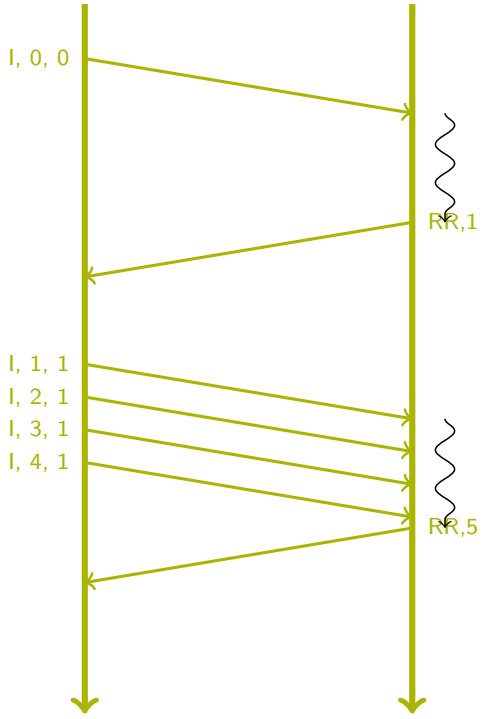
Unnumbered Frames

OSI Reference Model ▶ HDLC



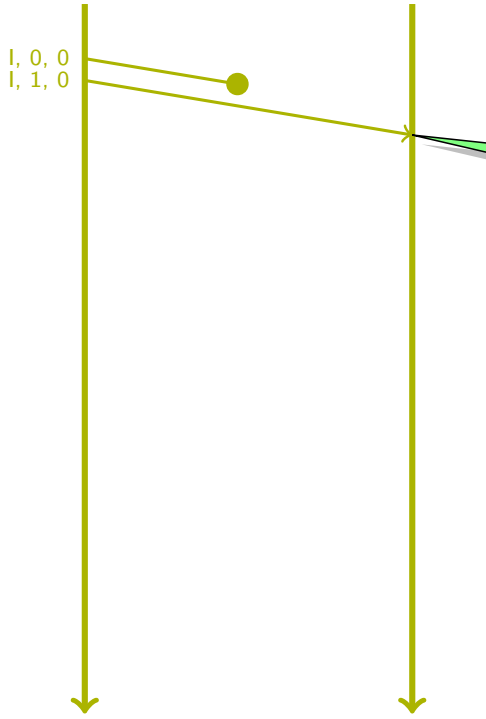
Slide 62 Laurent Toutain RES 1

Supervision Frame : RR (Receiver Ready)



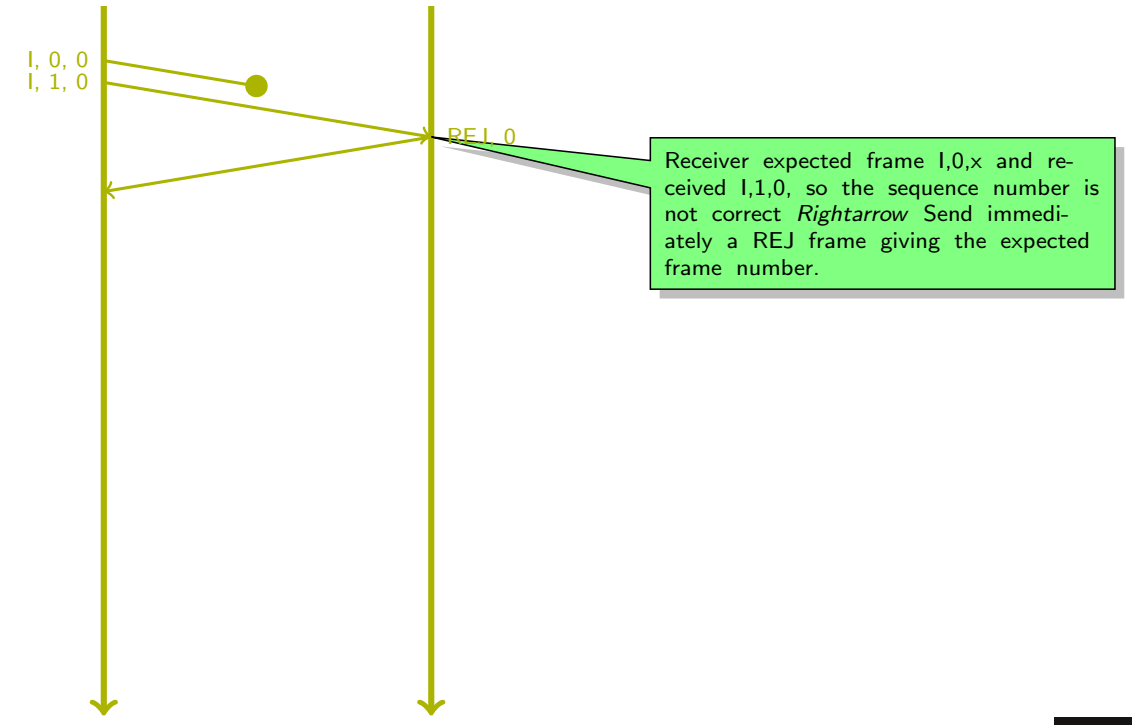
If receiver has no information to send, it can use RR frames instead.

Supervision Frame : REJ (Reject)



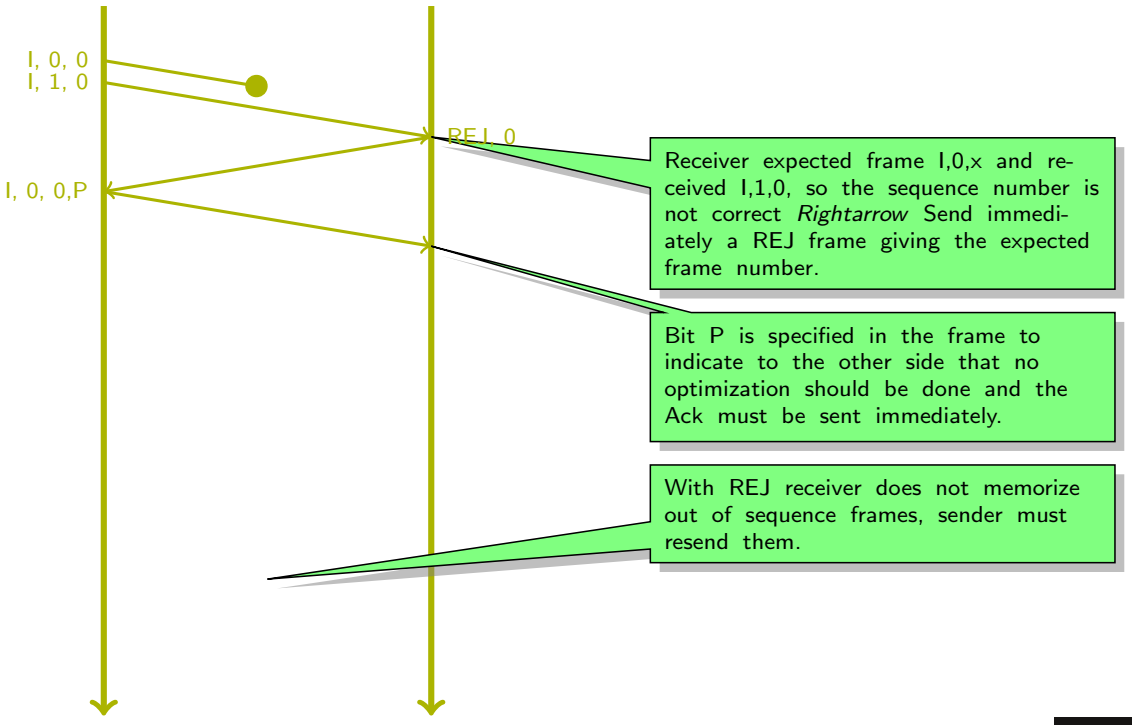
Receiver expected frame I,0,x and received I,1,0, so the sequence number is not correct *Rightarrow* Send immediately a REJ frame giving the expected frame number.

Supervision Frame : REJ (Reject)

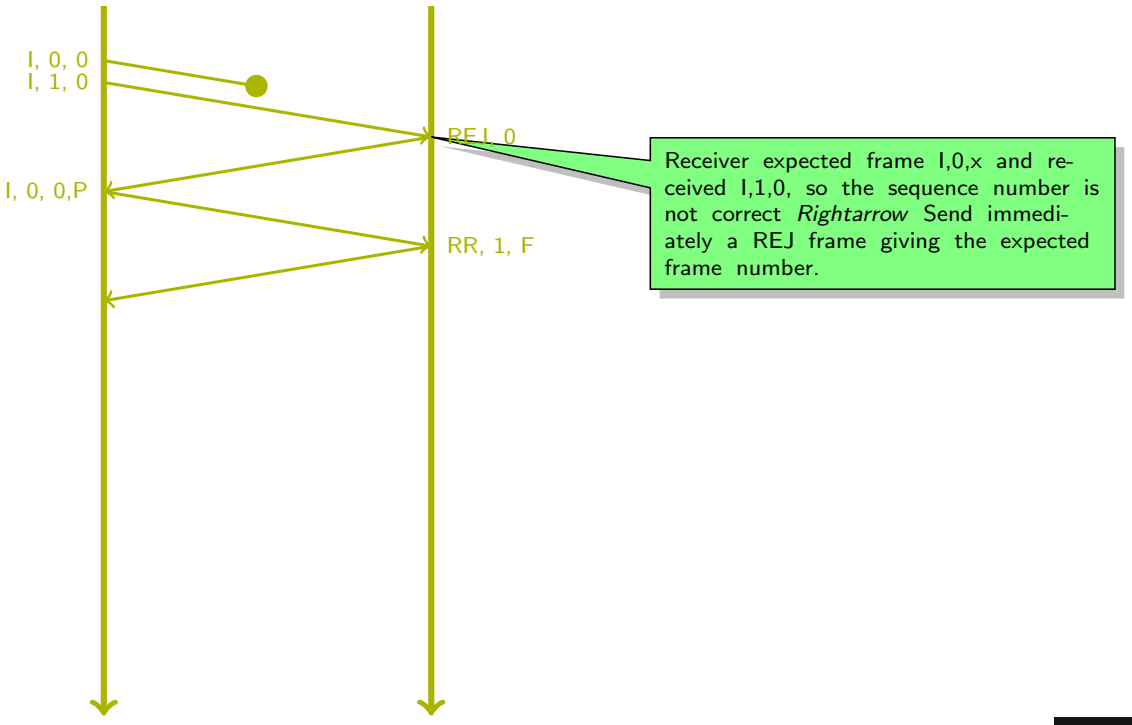


Supervision Frame : REJ (Reject)

OSI Reference Model ▶ HDLC

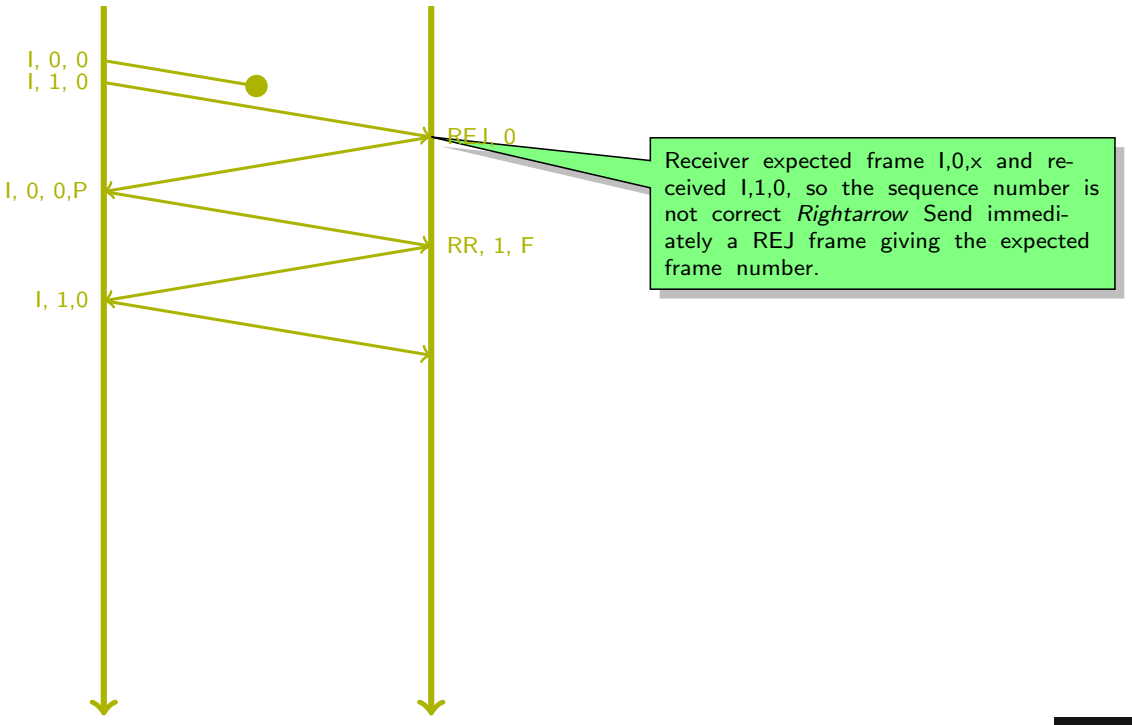


Supervision Frame : REJ (Reject)

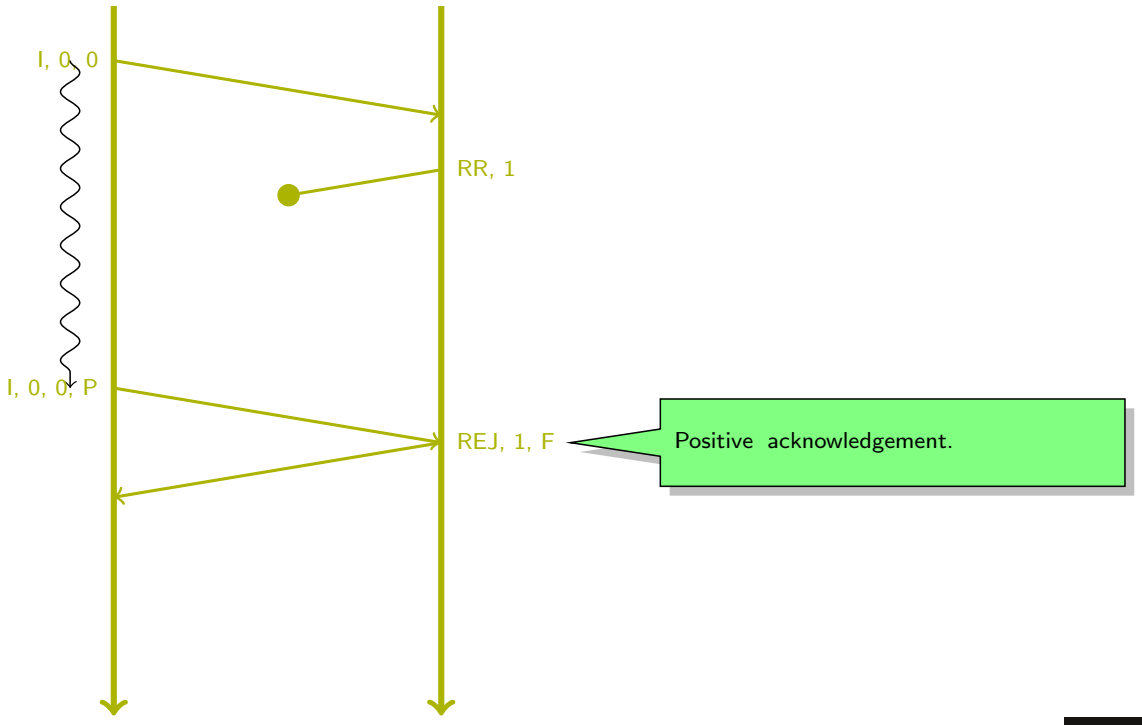


Supervision Frame : REJ (Reject)

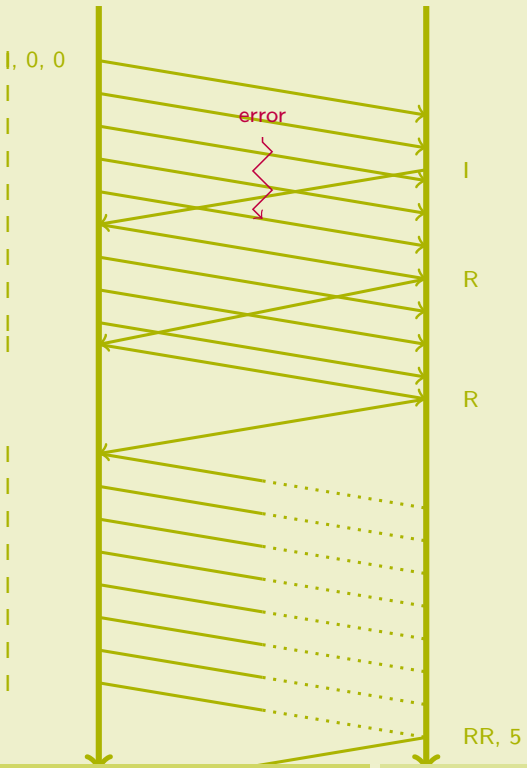
OSI Reference Model ▶ HDLC



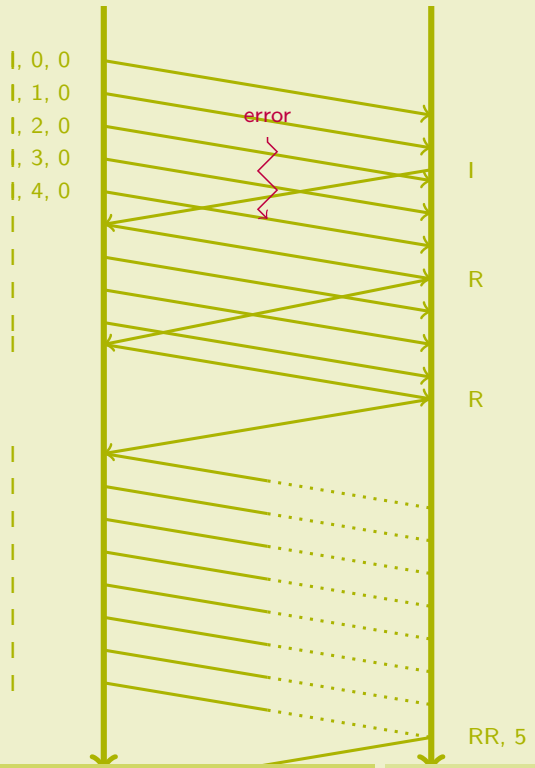
REJ ≠ NAK



Complete the exchange



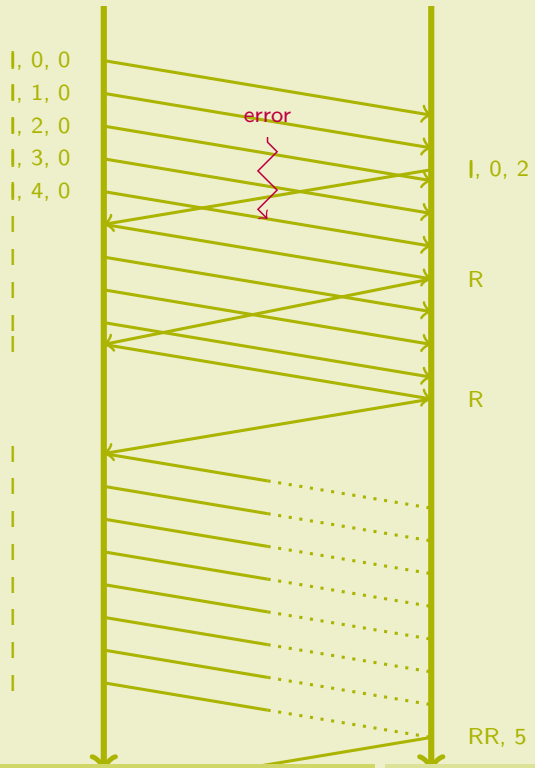
Complete the exchange



TELECOM Bretagne

Complete the exchange

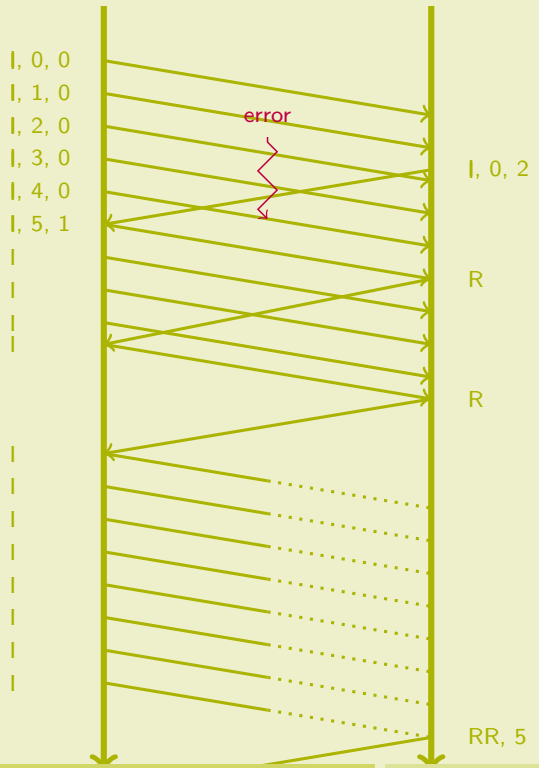
OSI Reference Model ▶ HDLC



Slide 67 Laurent Toutain RES 1

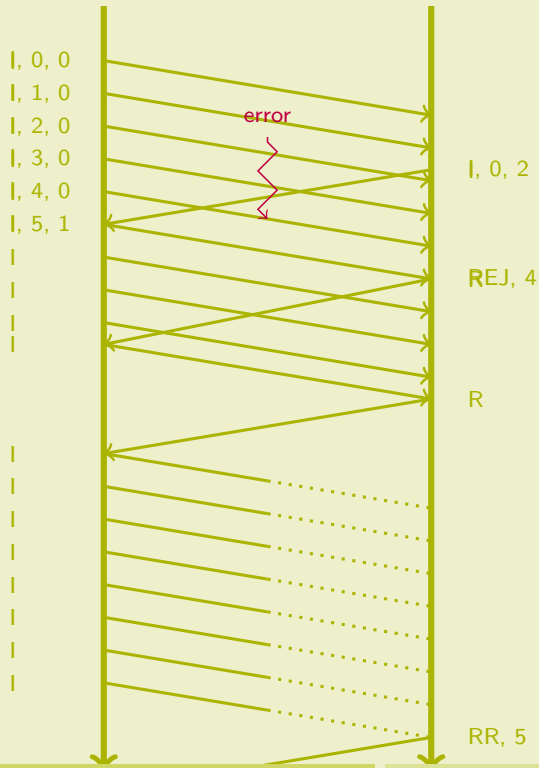
TELECOM Bretagne

Complete the exchange



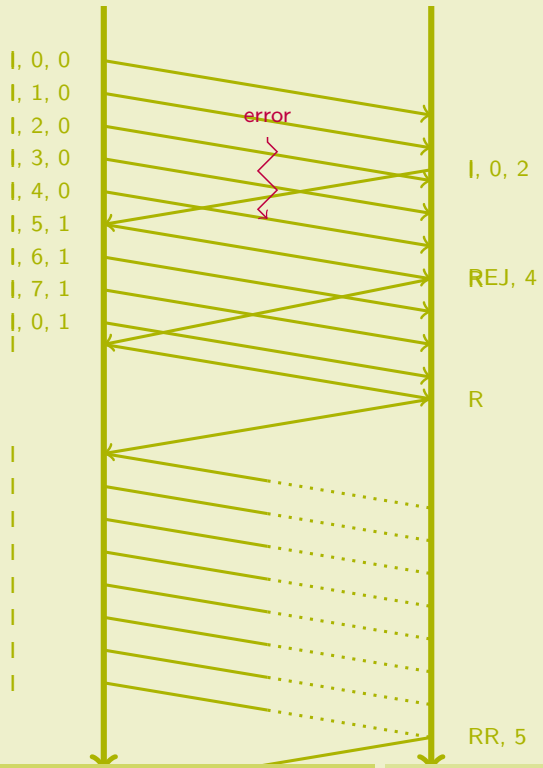
Complete the exchange

OSI Reference Model ▶ HDLC



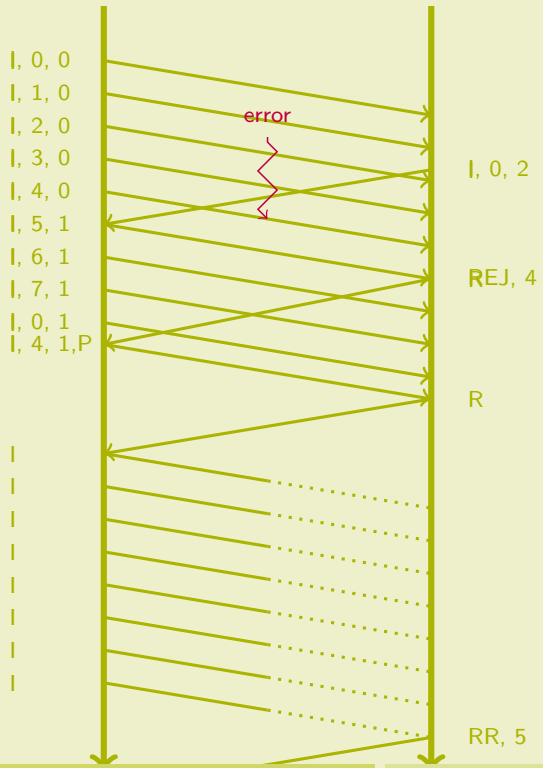
Slide 67 Laurent Toutain RES 1

Complete the exchange



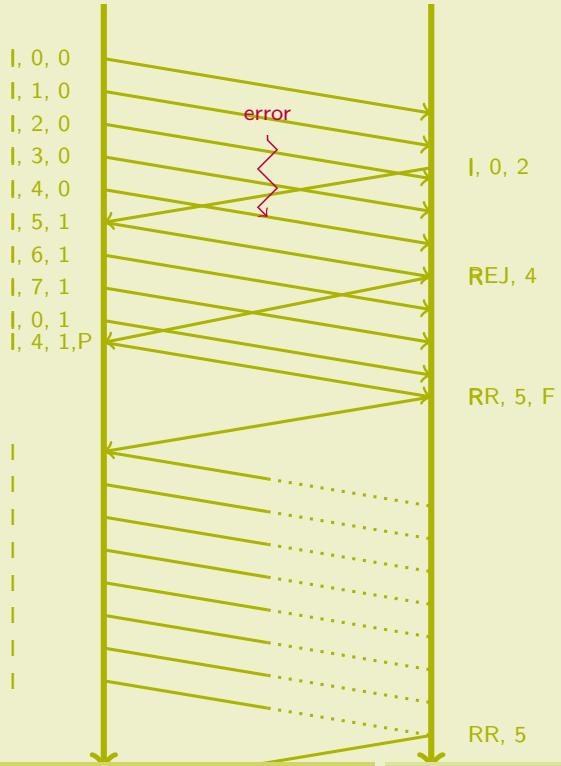
Complete the exchange

OSI Reference Model ▶ HDLC



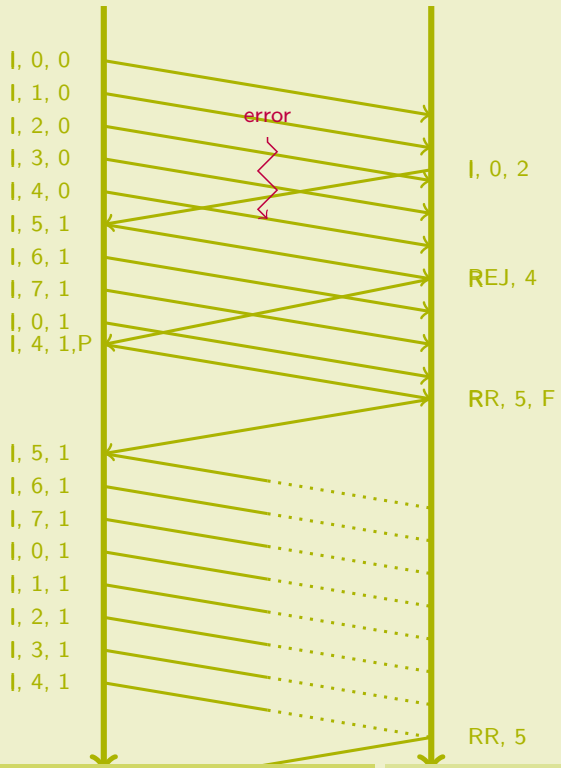
Slide 67 Laurent Toutain RES 1

Complete the exchange



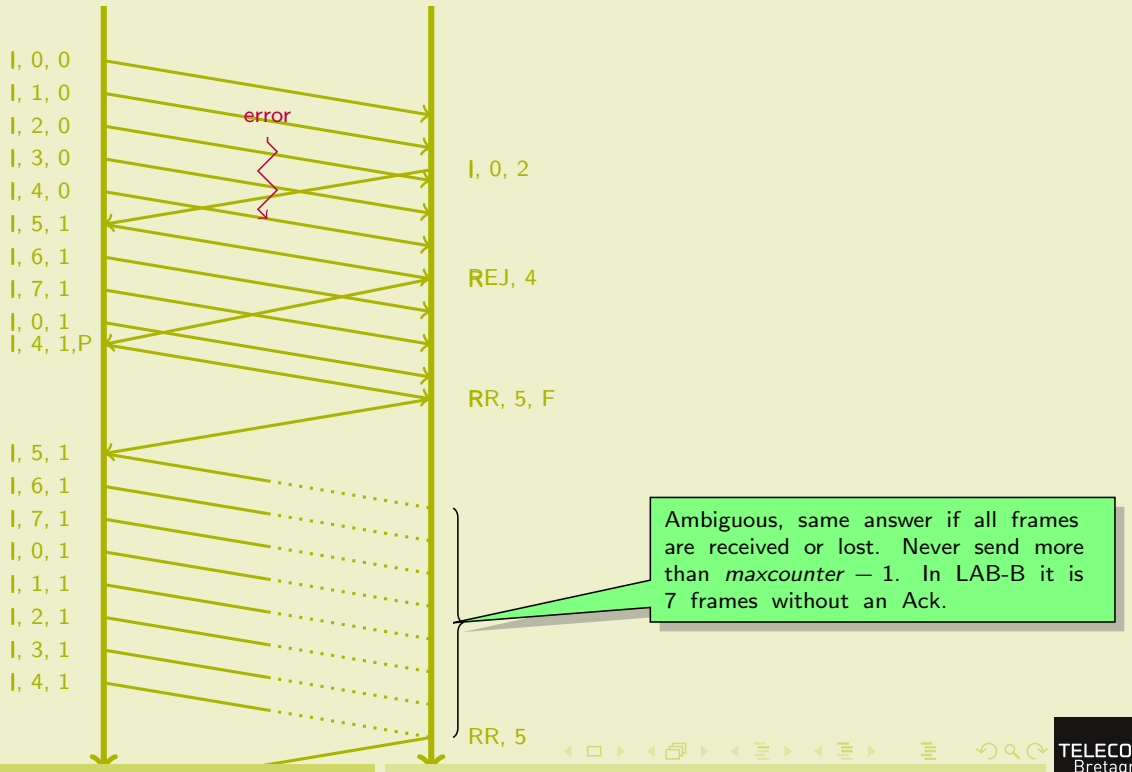
Complete the exchange

OSI Reference Model ▶ HDLC



Slide 67 Laurent Toutain RES 1

Complete the exchange



Complete the exchange

Dir	Len.	add	ctrl	data	desa
T	2	01	3F		SABM
R	2	01	73		UA
R	7	03	00	10 00 FB 07 00	I 0 0
T	7	01	00	10 00 FB 00 00	I 0 0
R	2	03	31		RR 1pf
R	2	01	21		RR 1
T	2	03	21		RR 1
...	...	...	...	...	...
T	2	03	31		RR 1pf
T	13	01	22	10 08 0B 04 29 09 00 01 00 00 00	I 1 1

Which fields are added/missing by the analyzer ?

Verify output given by Analyzer

Which frames contain data ?

Complete the exchange

Dir	Len.	add	ctrl	data	desa
T	2	01	3F		SABM
R	2	01	73		UA
R	7	03	00	10 00 FB 07 00	I 0 0
T	7	01	00	10 00 FB 00 00	I 0 0
R	2	03	31		RR 1pf
R	2	01	21		RR 1
T	2	03	21		RR 1
...	...	...	...	...	...
T	2	03	31		RR 1pf
T	13	01	22	10 08 0B 04 29 09 00 01 00 00 00	I 1 1

Which fields are added/missing by the analyzer ? added: Dir, Len. & Desa missing: CRC

Verify output given by Analyzer

Which frames contain data ?

Complete the exchange

OSI Reference Model ▶ HDLC

Dir	Len.	add	ctrl	data	desa
T	2	01	3F		SABM
R	2	01	73		UA
R	7	03	00	10 00 FB 07 00	I 0 0
T	7	01	00	10 00 FB 00 00	I 0 0
R	2	03	31		RR 1pf
R	2	01	21		RR 1
T	2	03	21		RR 1
...	...	...	...	...	...
T	2	03	31		RR 1pf
T	13	01	22	10 08 0B 04 29 09 00 01 00 00 00	I 1 1

Which fields are added/missing by the analyzer ? added: Dir, Len. & Desa missing: CRC

Verify output given by Analyzer Beware bit order

`0x3F = 0011 1111 → 1111 1100 = SABM`

Which frames contain data ?

Complete the exchange

Dir	Len.	add	ctrl	data	desa
T	2	01	3F		SABM
R	2	01	73		UA
R	7	03	00	10 00 FB 07 00	I 0 0
T	7	01	00	10 00 FB 00 00	I 0 0
R	2	03	31		RR 1pf
R	2	01	21		RR 1
T	2	03	21		RR 1
...	...	...	...	...	...
T	2	03	31		RR 1pf
T	13	01	22	10 08 0B 04 29 09 00 01 00 00 00	I 1 1

Which fields are added/missing by the analyzer ? added: Dir, Len. & Desa missing: CRC

Verify output given by Analyzer Beware bit order

`0x3F = 0011 1111 → 1111 1100 = SABM`

Which frames contain data ? I Frames

OSI Reference Model

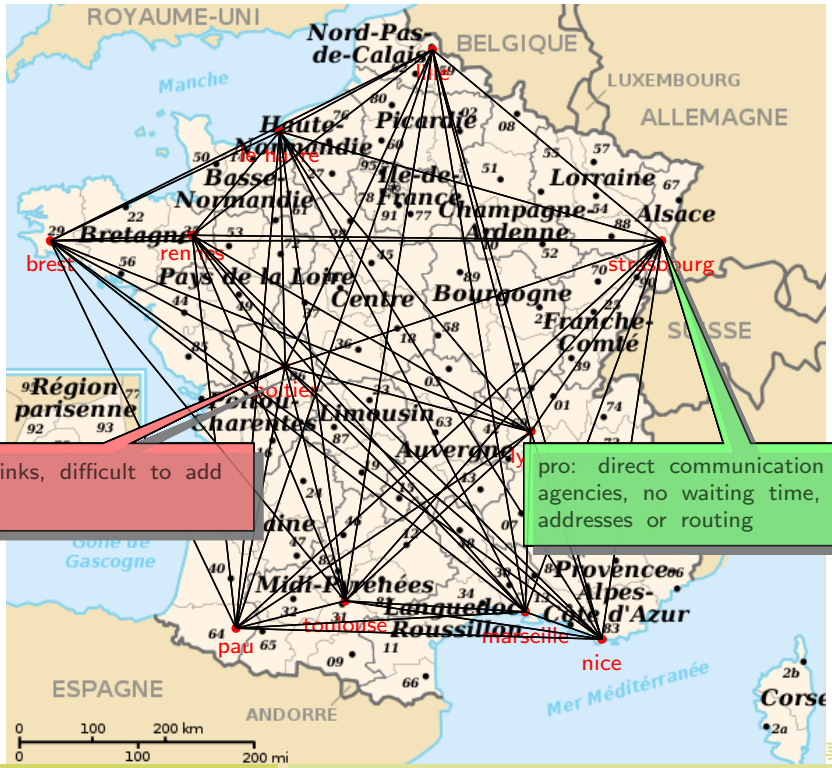
Layer 3: Network

Packet Justification

TELECOM Bretagne

Packet Justification

OSI Reference Model ▶ Layer 3: Network



cons: too much links, difficult to add a new agency

pro: direct communication between agencies, no waiting time, no need for addresses or routing

Slide 71 Laurent Toutain RES 1

TELECOM Bretagne

Packet Justification

Packet Justification

OSI Reference Model ▶ Layer 3: Network

Packet Justification

OSI Reference Model ▶ Layer 3: Network

pro: less expensive resources, easy to add an agency

Packet Justification

OSI Reference Model ▶ Layer 3: Network

cons: need to designate agencies, infrastructure need to route messages toward destination, link failure may stop several communications

pro: less expensive resources, easy to add an agency

Packet Justification

How to establish communication

- preset some period of time for communication, for instance for 0200 to 0300 communication from Lille to Nice
  - not very flexible, loose bandwidth if no data to send.
- define a communication with the network and tell it to establish a path between Lille and Nice
  - define a way to identify agencies
  - define a "signalization" protocol
  - configure intermediary nodes for this path
  - More complex to implement, but allow a better use of resources
  - may not allow other communication when one is established (for instance Brest - Pau)
- Store and Forward
  - Store message on intermediary node before transmit it on the next link
  - All simultaneous communications and better multiplexing
  - Introduce some memorization delays

Packet Justification

Packet Justification

OSI Reference Model ▶ Layer 3: Network

Packet Justification

Packet Justification

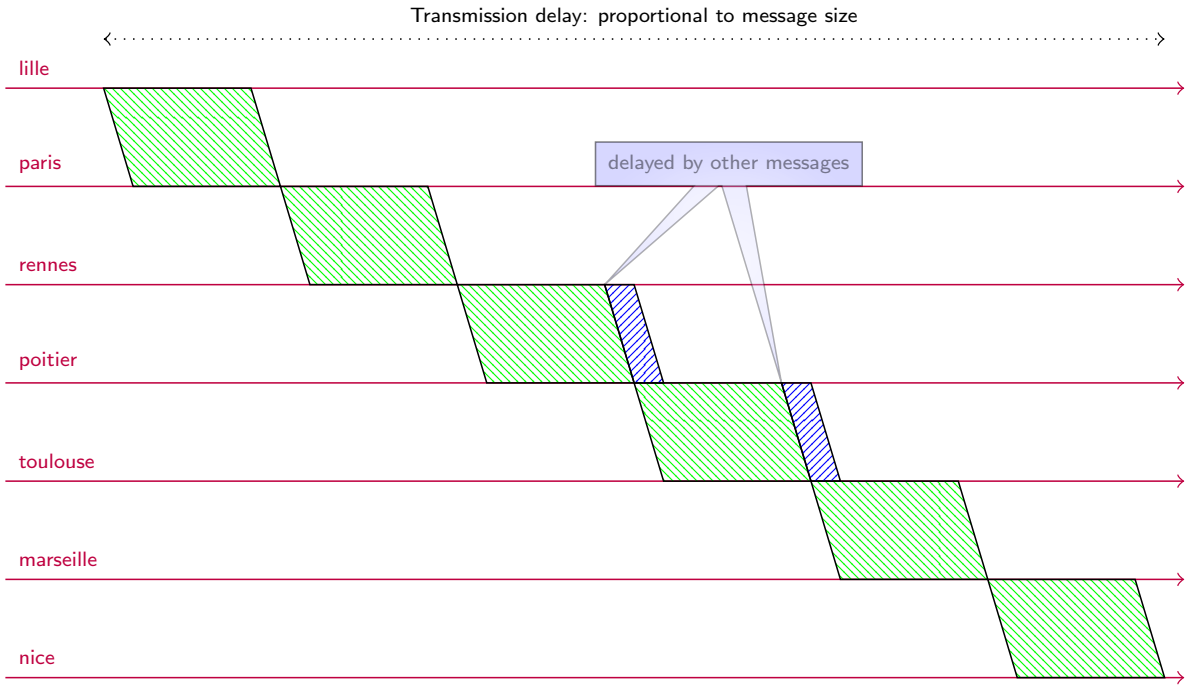
OSI Reference Model ▶ Layer 3: Network

Packet Justification

Packet Justification

OSI Reference Model ▶ Layer 3: Network

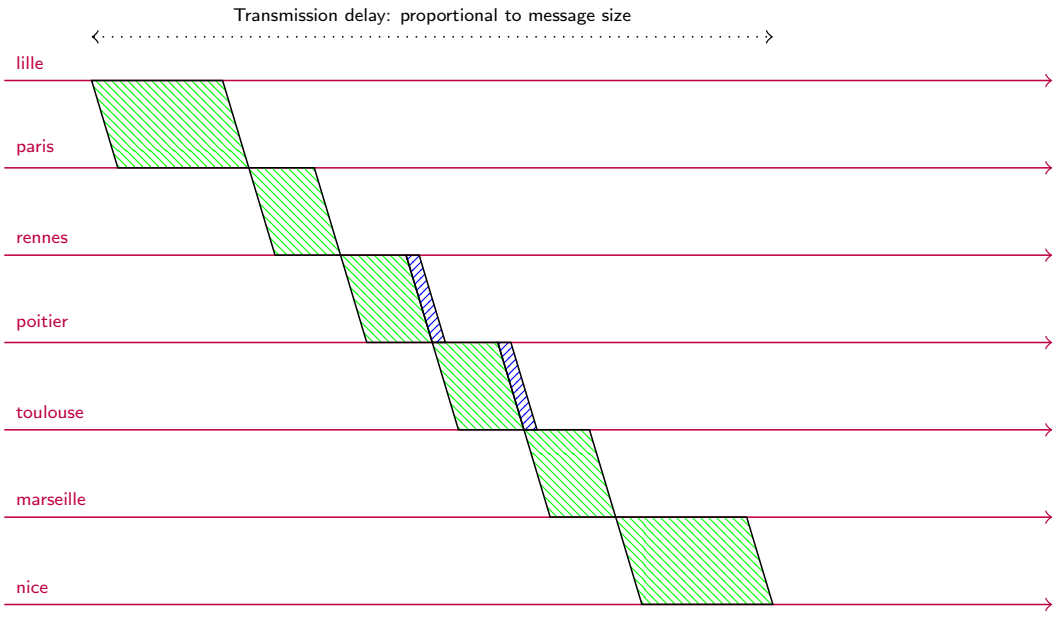
Store and Forward: other view



Store and Forward: other view

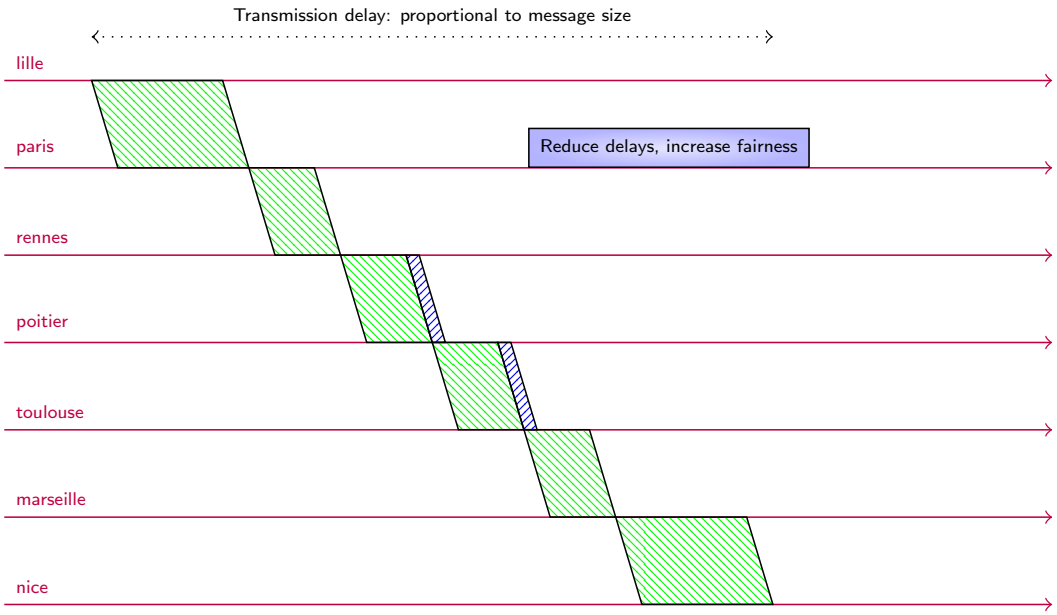
OSI Reference Model ▶ Layer 3: Network

What happens if infrastructure speed is doubled ?



Store and Forward: other view

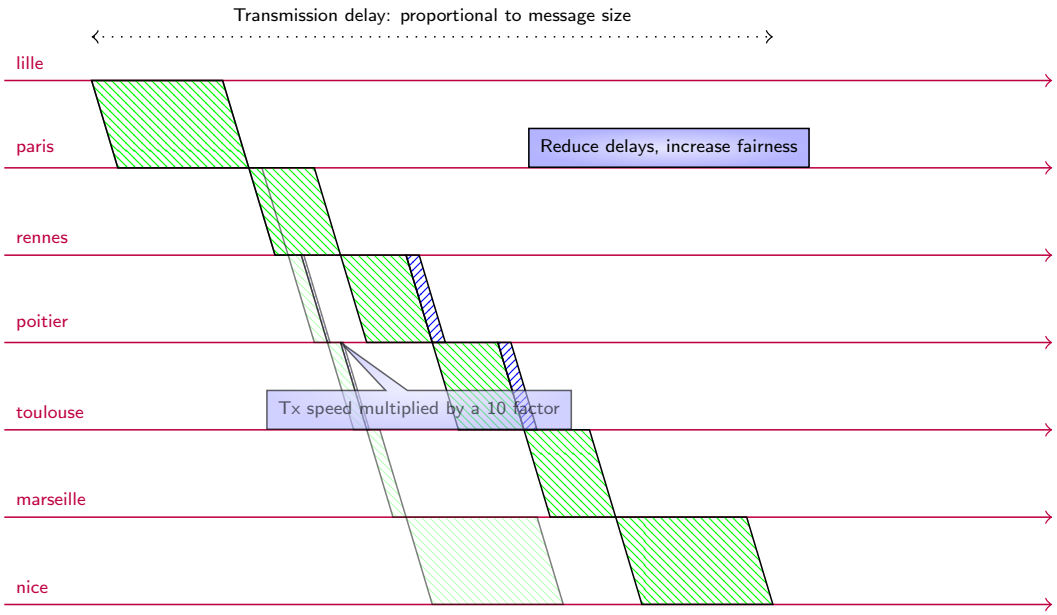
What happens if infrastructure speed is doubled ?



OSI Reference Model ▶ Layer 3: Network

Store and Forward: other view

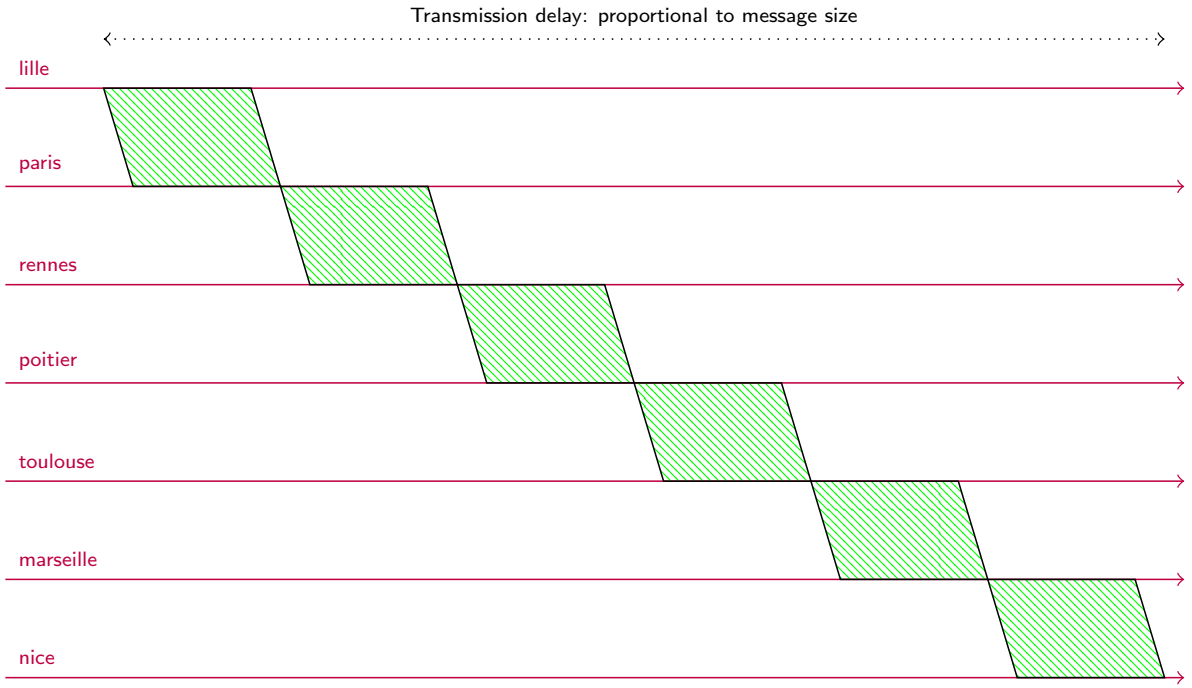
What happens if infrastructure speed is doubled ?



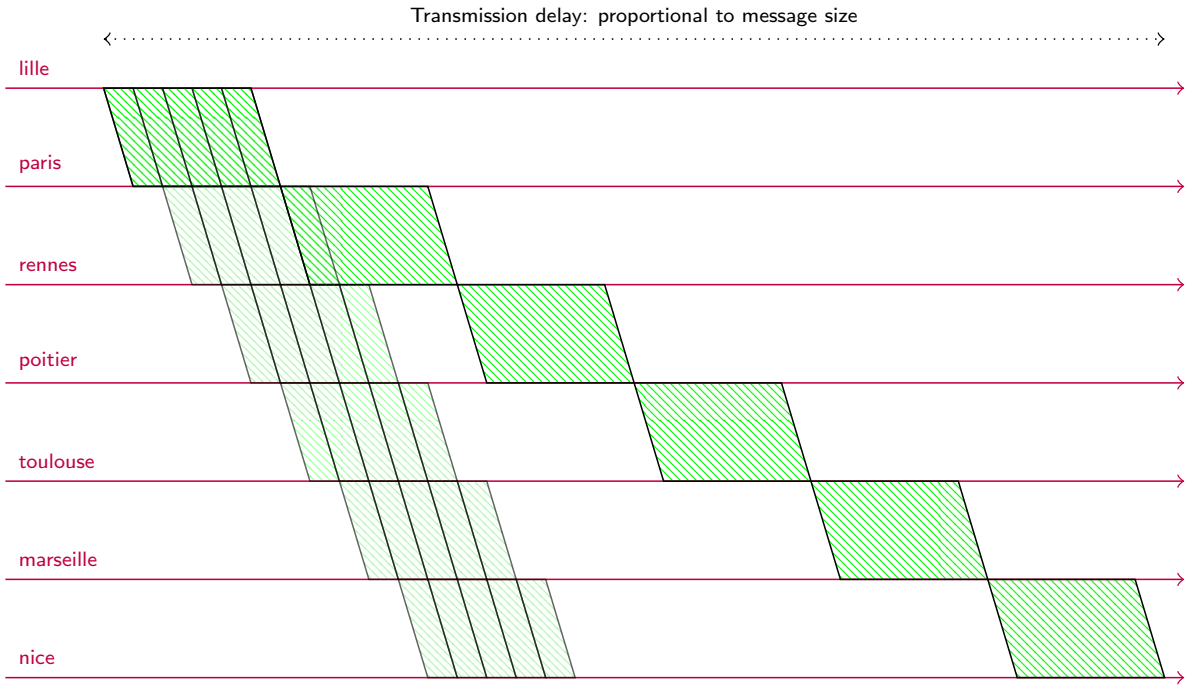
Store and Forward: other view

- messages are not bounded
  - size introduces delays
  - intermediary node memory cannot be defined
  - long messages create unfairness between flows
- Solution: limit the message size
  - Packet: PDU with a maximum length well known by all the nodes
  - larger messages are fragmented into several packets

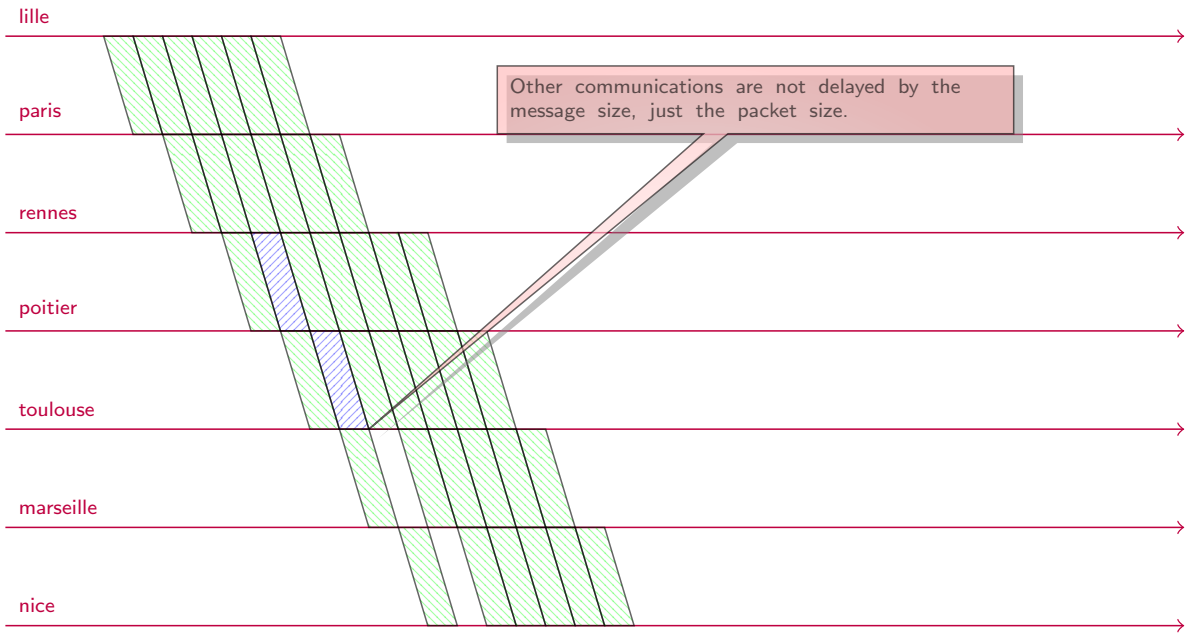
Store and Forward: Increase propagation speed



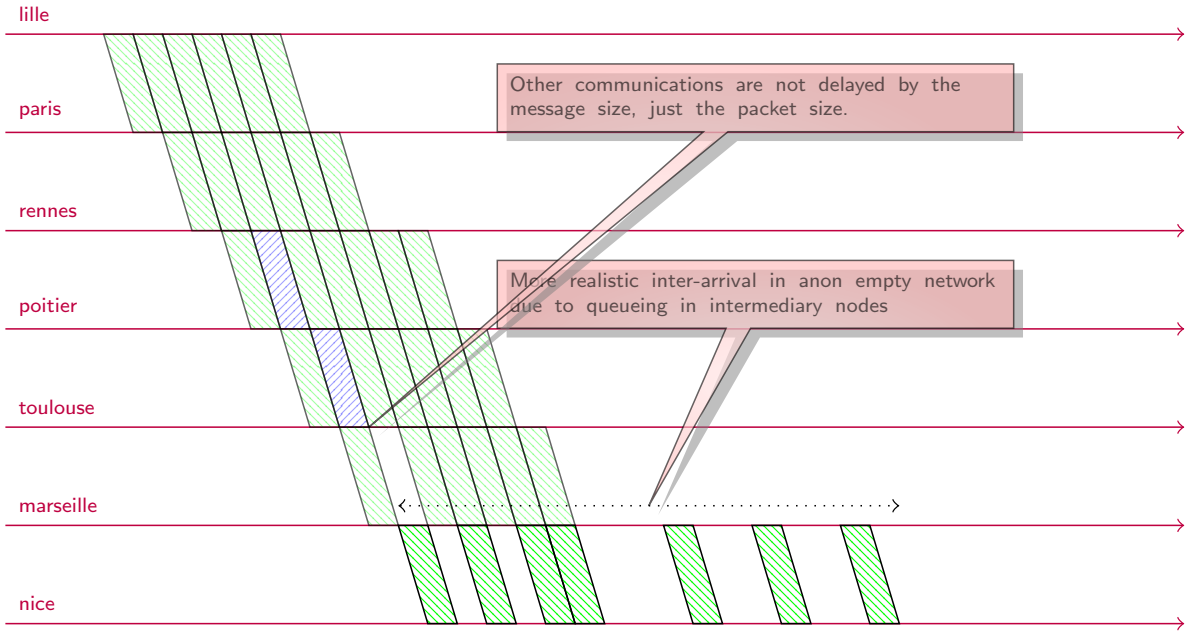
Store and Forward: Increase propagation speed



Store and Forward: Increase fairness

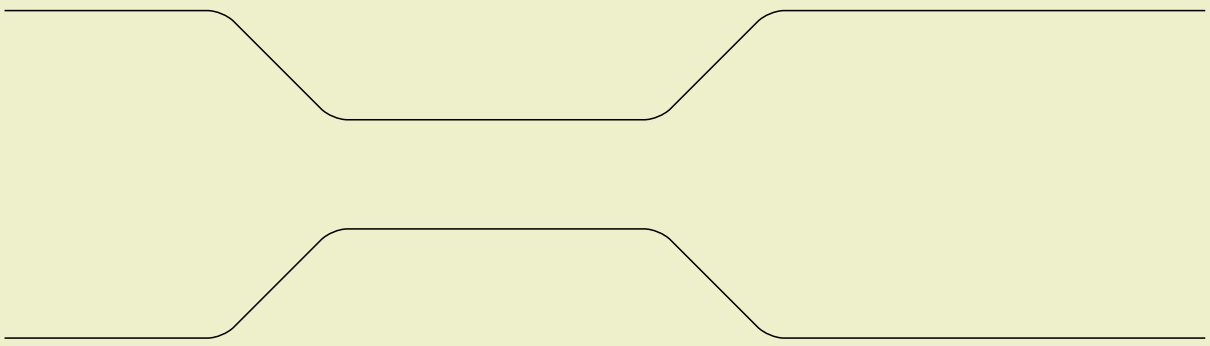


Store and Forward: Increase fairness



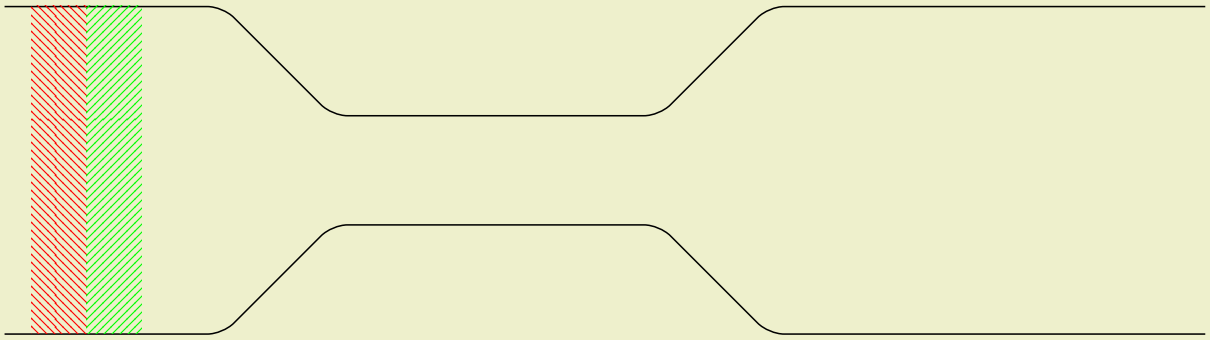
Packet Pairs

A method called packet pair is used to measure bottleneck bandwidth. Two packets are send consecutively and the inter-arrival time is measured after the bottleneck. Give the formula to get bandwidth.



Packet Pairs

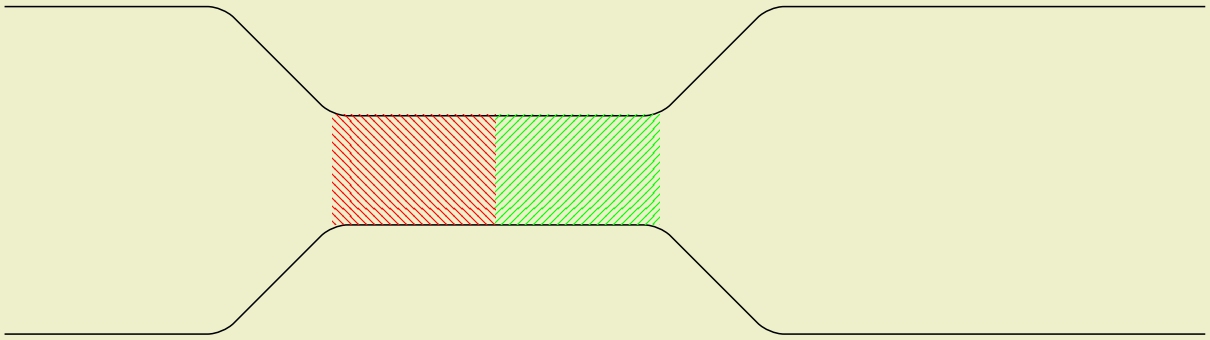
A method called packet pair is used to measure bottleneck bandwidth. Two packets are send consecutively and the inter-arrival time is measured after the bottleneck. Give the formula to get bandwidth.



Packet Pairs

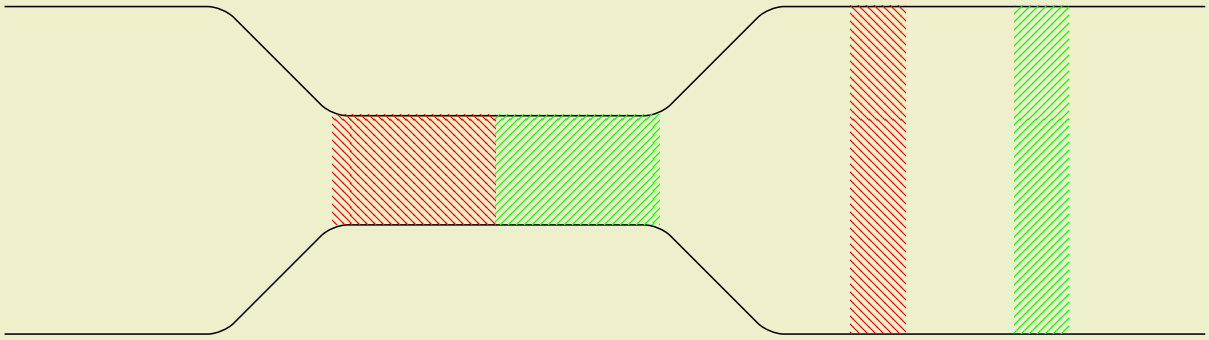
OSI Reference Model ▶ Layer 3: Network

A method called packet pair is used to measure bottleneck bandwidth. Two packets are send consecutively and the inter-arrival time is measured after the bottleneck. Give the formula to get bandwidth.



Packet Pairs

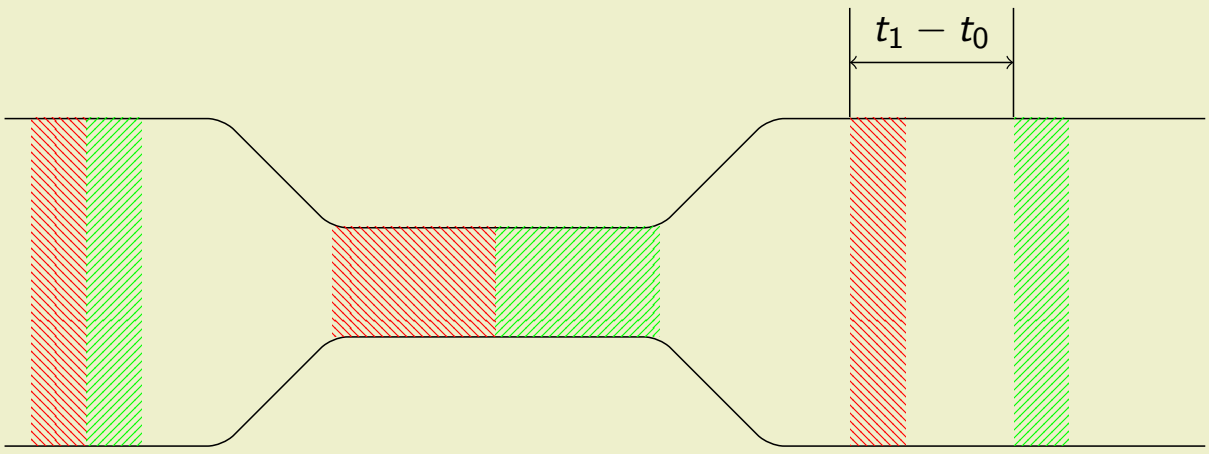
A method called packet pair is used to measure bottleneck bandwidth. Two packets are send consecutively and the inter-arrival time is measured after the bottleneck. Give the formula to get bandwidth.



Packet Pairs

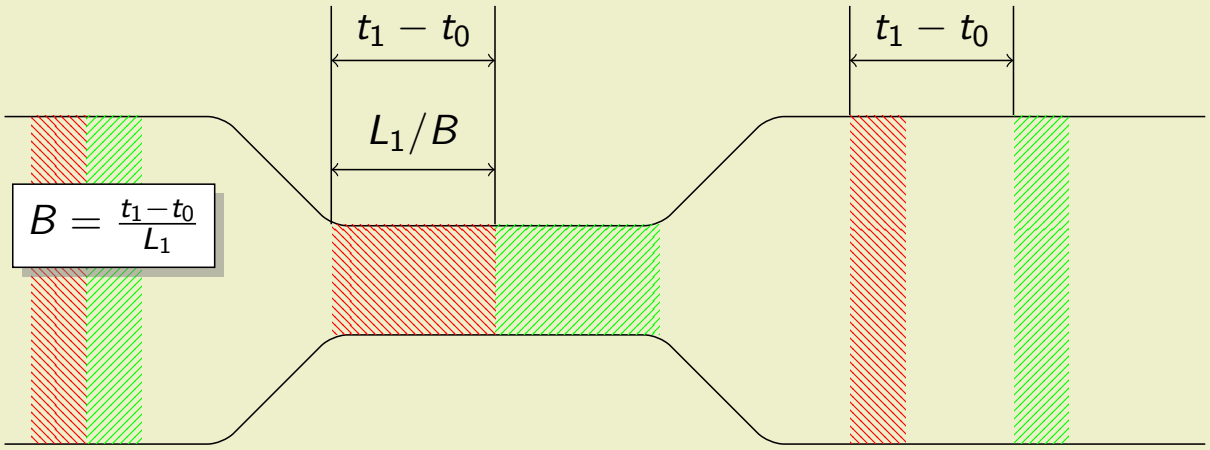
OSI Reference Model ▶ Layer 3: Network

A method called packet pair is used to measure bottleneck bandwidth. Two packets are send consecutively and the inter-arrival time is measured after the bottleneck. Give the formula to get bandwidth.



Packet Pairs

A method called packet pair is used to measure bottleneck bandwidth. Two packets are send consecutively and the inter-arrival time is measured after the bottleneck. Give the formula to get bandwidth.



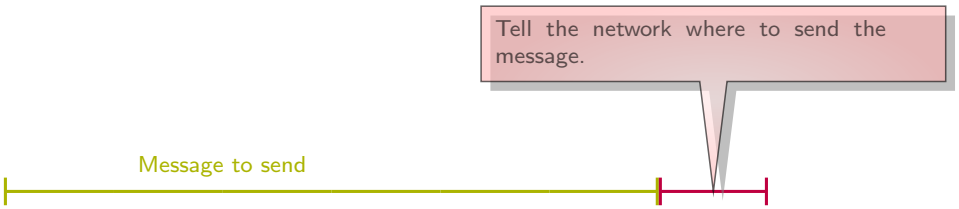
OSI Reference Model

Connection oriented vs Datagram

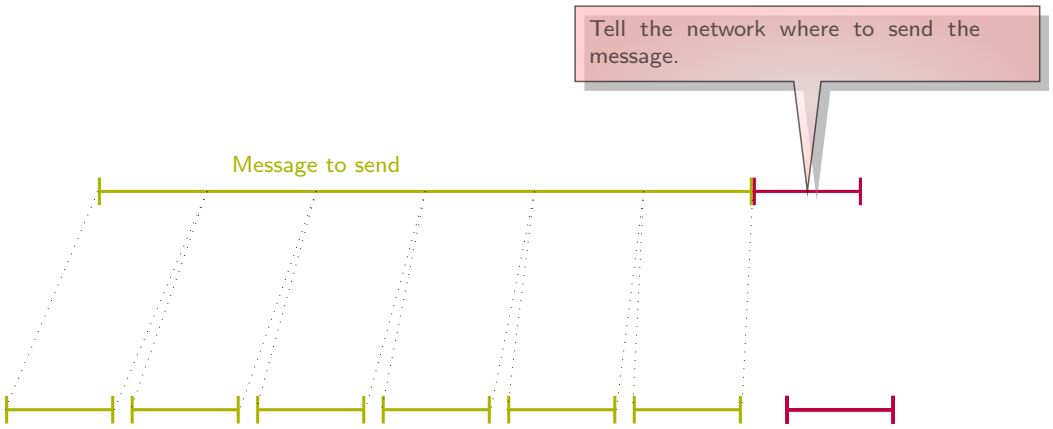
C.O. vs Datagram

C.O. vs Datagram

OSI Reference Model ▶ Connection oriented vs Datagram

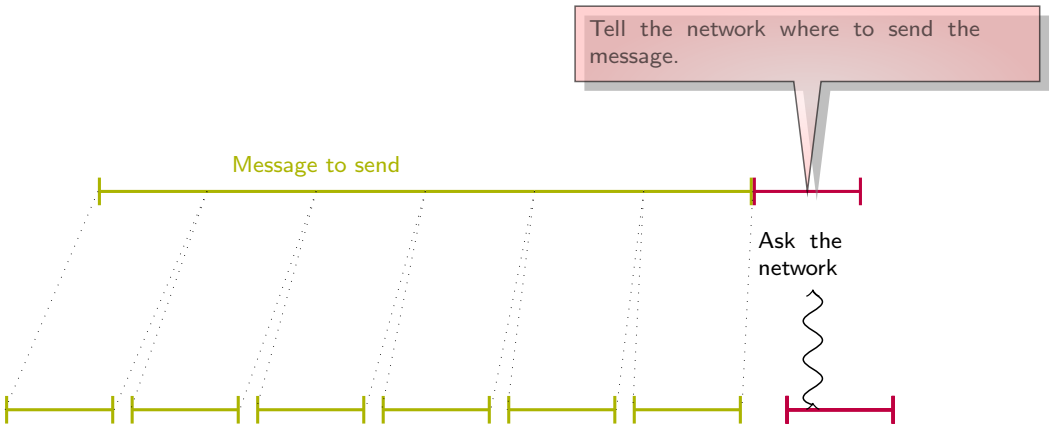


C.O. vs Datagram



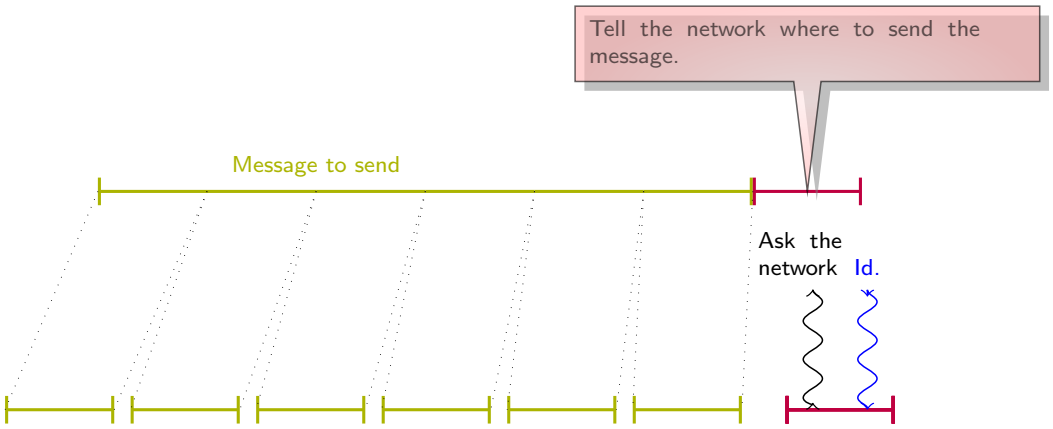
C.O. vs Datagram

OSI Reference Model ▶ Connection oriented vs Datagram



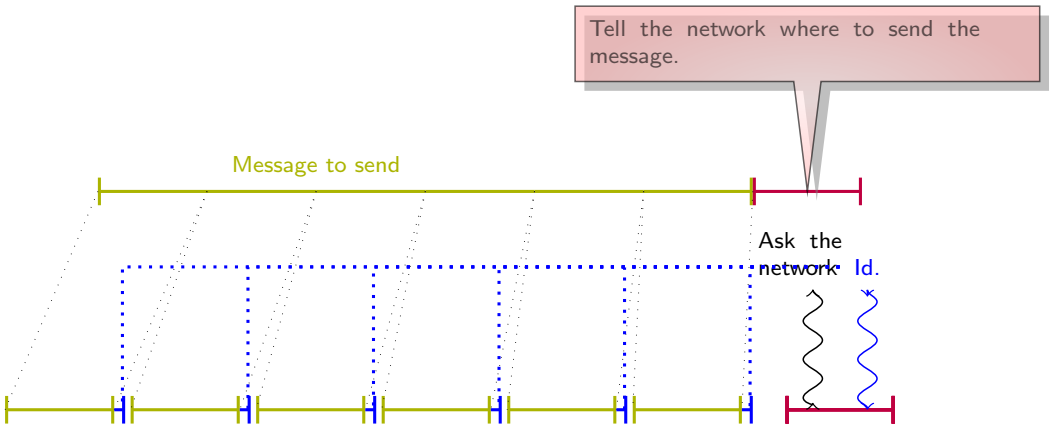
Slide 82 Laurent Toutain RES 1

C.O. vs Datagram

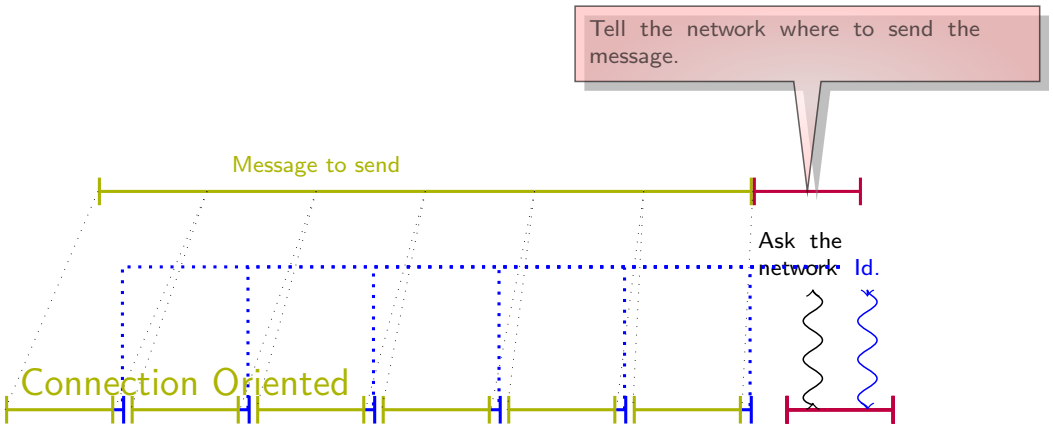


C.O. vs Datagram

OSI Reference Model ▶ Connection oriented vs Datagram

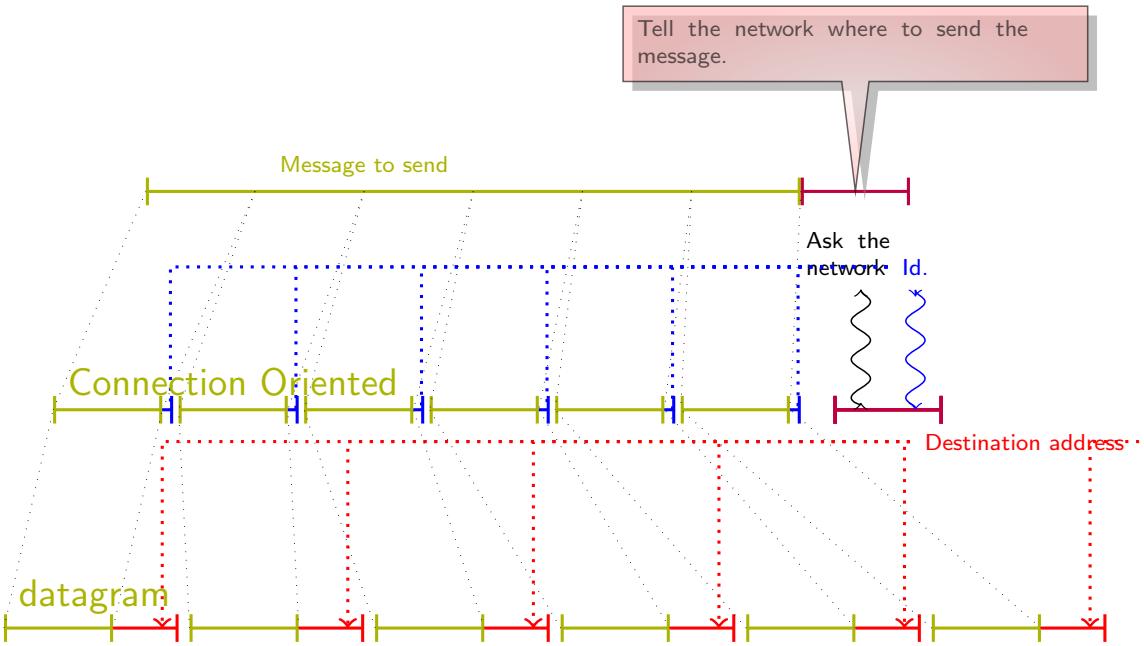


C.O. vs Datagram

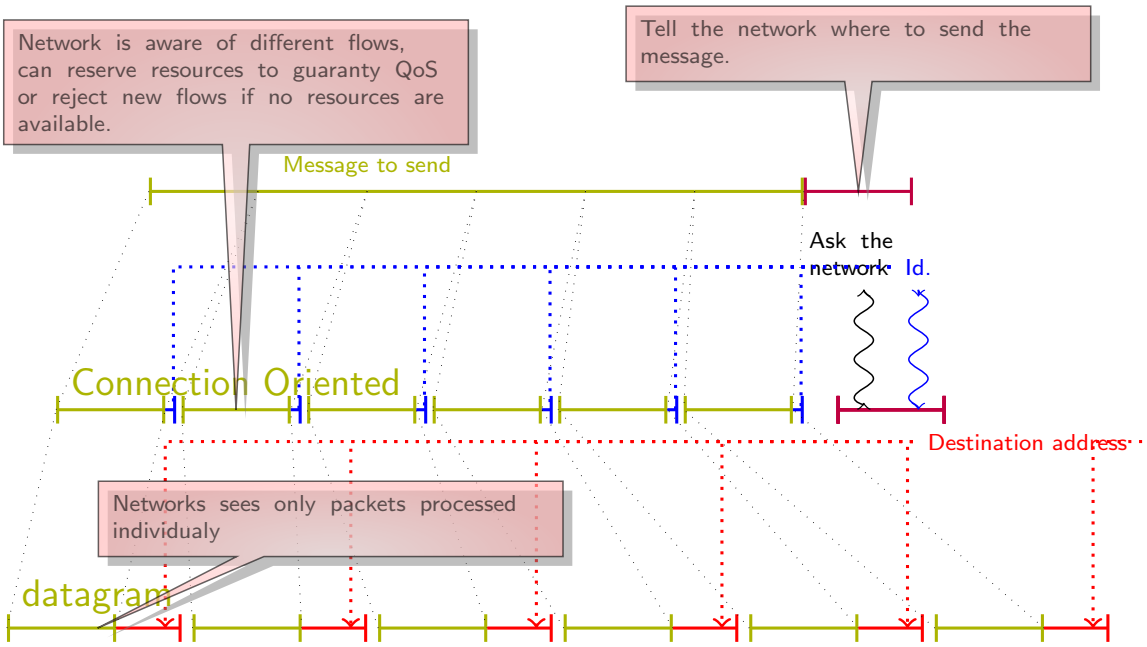


C.O. vs Datagram

OSI Reference Model ▶ Connection oriented vs Datagram



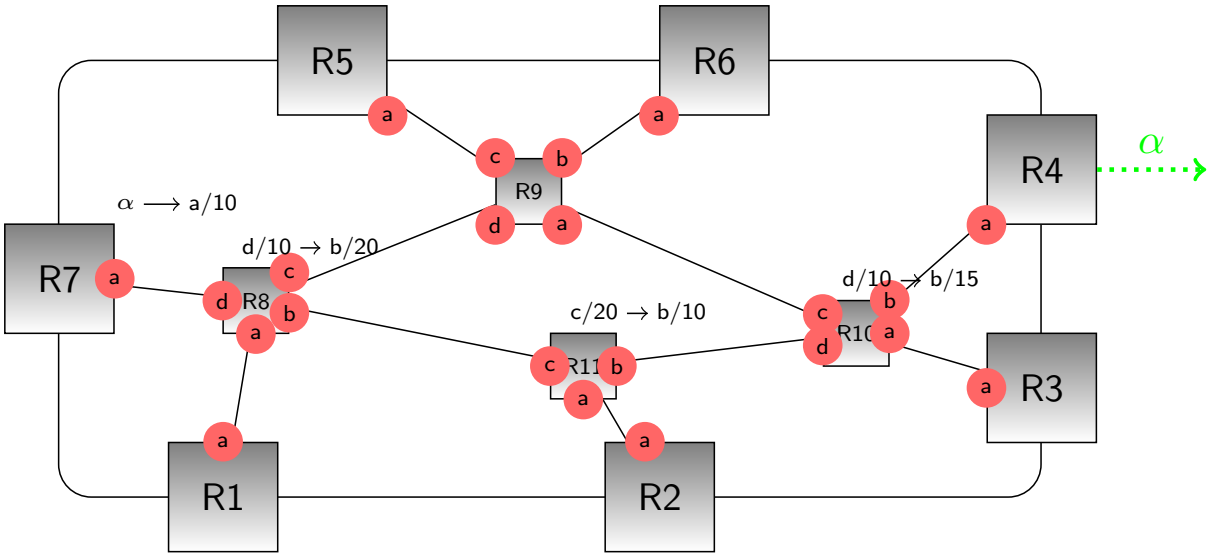
C.O. vs Datagram



Connection Oriented

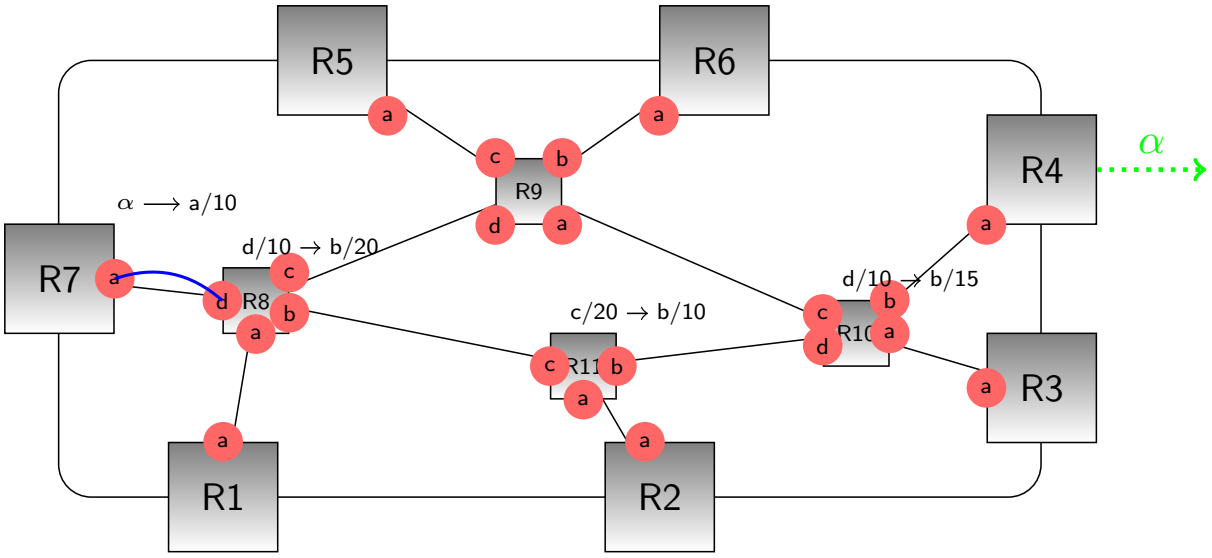
- The node send a well-known message to the network requesting to open a connection to the destination
  - The message can contain some Quality of Service parameters such as the needed bandwidth, the propagation time, the reliability, ...
  - The network can reject call if not enough resources are available.
- Destination receives an indication from network (with source address) and an id.
- id view by source and destination can be different
  - id is just unique on a link
  - and changed from link to link
  - a switch matrix is on each intermediary node telling based on each couple (entry interface and id) the exiting interface and new id.
  - the matrix is built during the opening phase.
- this is called a Virtual Circuit

Example: MPLS

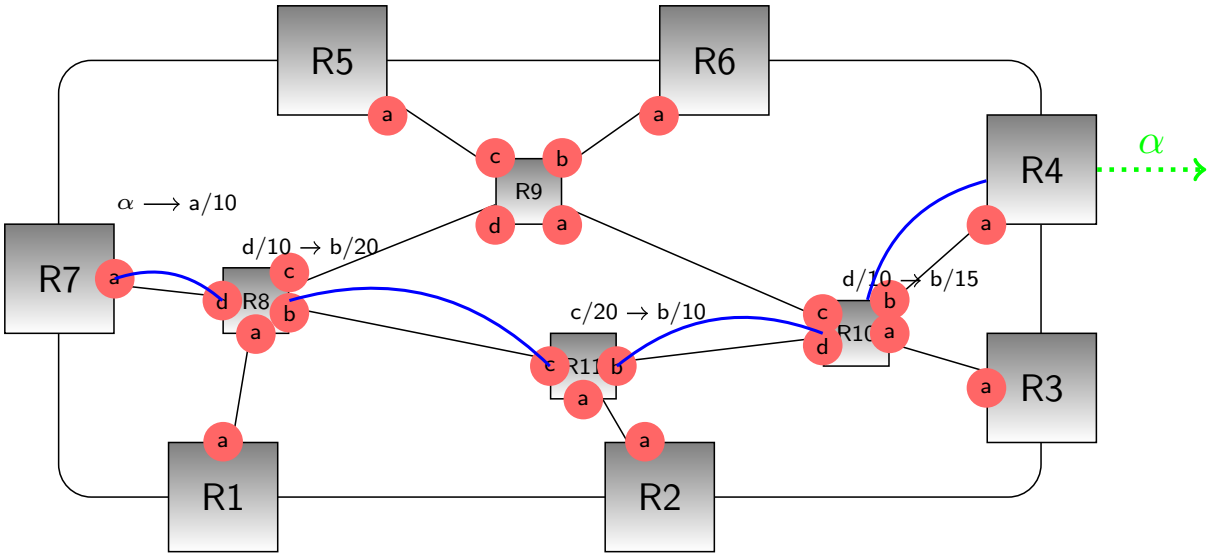


Example: MPLS

OSI Reference Model ▶ Connection oriented vs Datagram



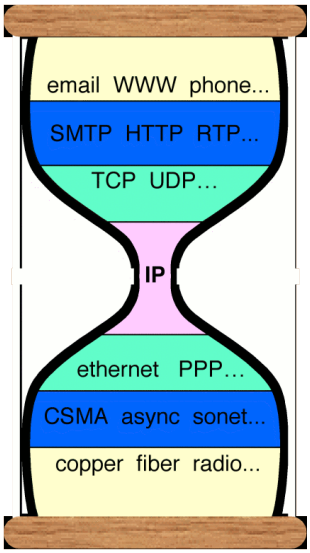
Example: MPLS



Concepts

Datagram

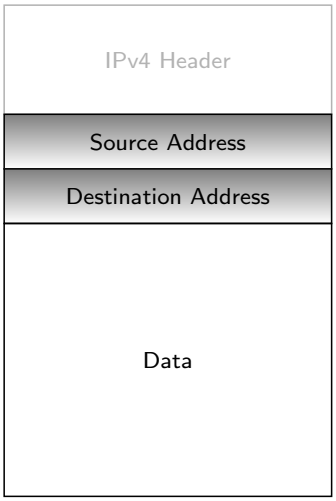
IP Layer



- IP is kept simple
  - Forwards packet towards destination
- IP on everything
  - Adapt IP protocol on every layer 2
- Everything on IP
  - Write applications to use IP layer (through L4: TCP, UDP)
- IP must facilitate network interconnection
  - Avoid ambiguities on addresses

W http://www.ietf.org/proceedings/01aug/slides/plenary-1/index.html Steve deering, Watching the Waist of the Protocol Hourglass, IETF 51, London

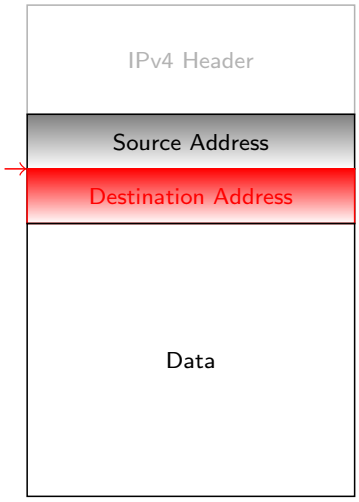
Destination Address Processing



The destination address must be easily accessible:

- Fixed location
- Fixed size
- Aligment in memory

Destination Address Processing

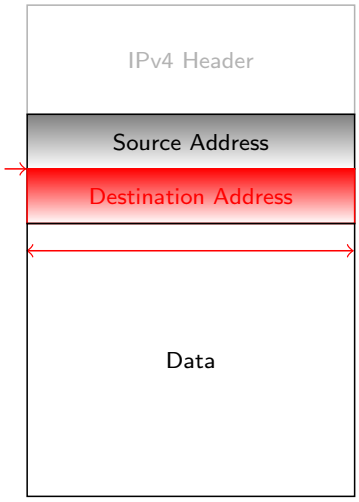


The destination address must be easily accessible:

- Fixed location
- Fixed size
- Aligment in memory

Destination Address Processing

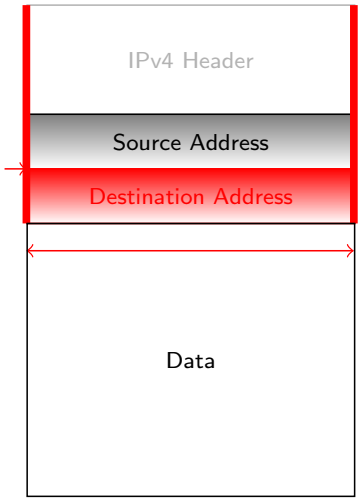
Concepts ▶ Datagram



The destination address must be easily accessible:

- Fixed location
- Fixed size
- Aligment in memory

Destination Address Processing

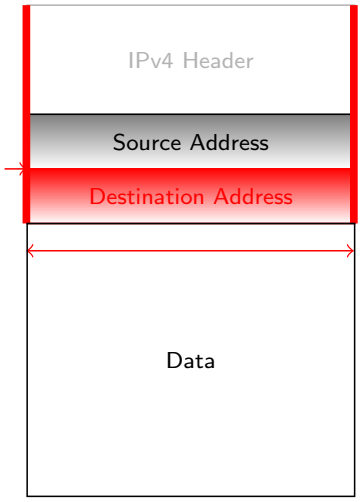


The destination address must be easily accessible:

- Fixed location
- Fixed size
- Aligment in memory

Destination Address Processing

Concepts ▶ Datagram



The destination address must be easily accessible:

- Fixed location
- Fixed size
- Aligment in memory

RFC 791 (Sept 1981)

`Addresses are fixed length of four octets (32 bits)`