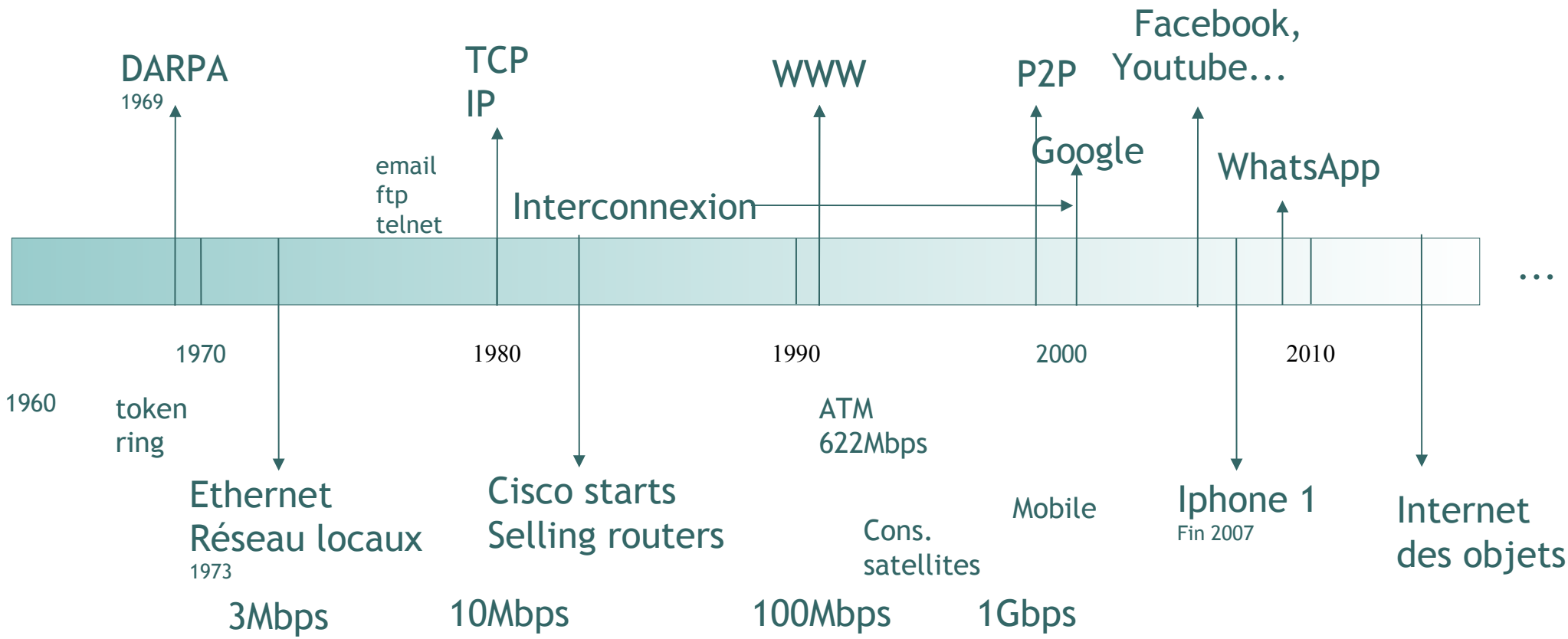
Conception de protocoles réseaux

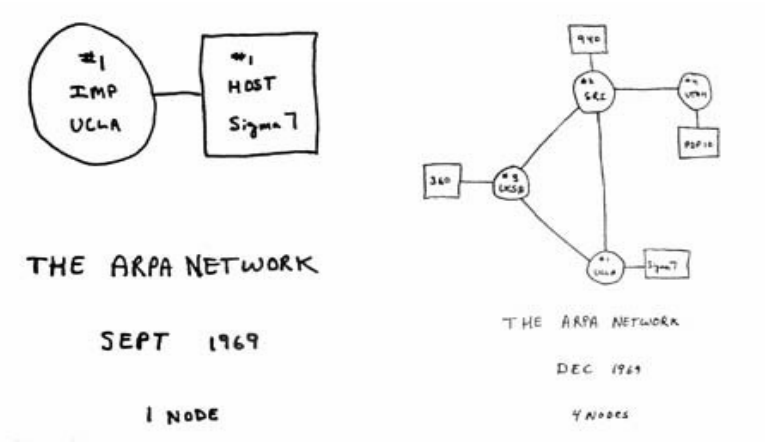
Mickaël Hoerdt
HEPIA

A short history of computer networking

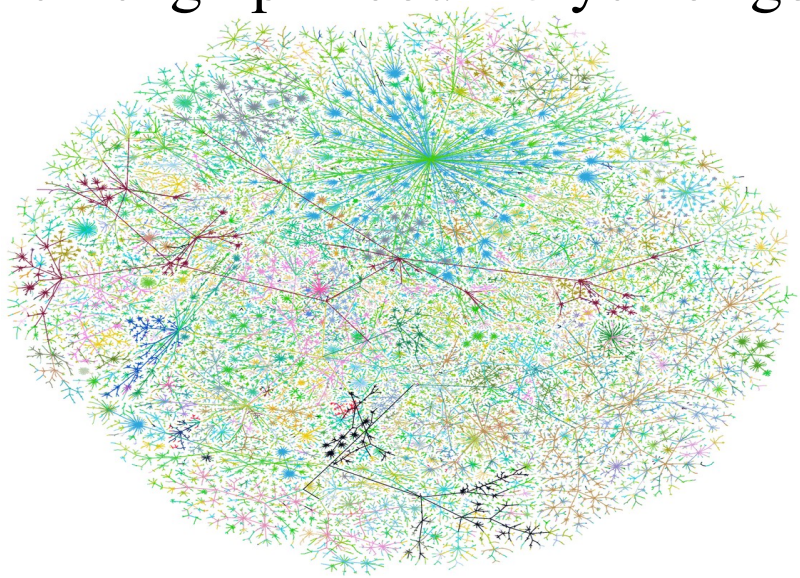


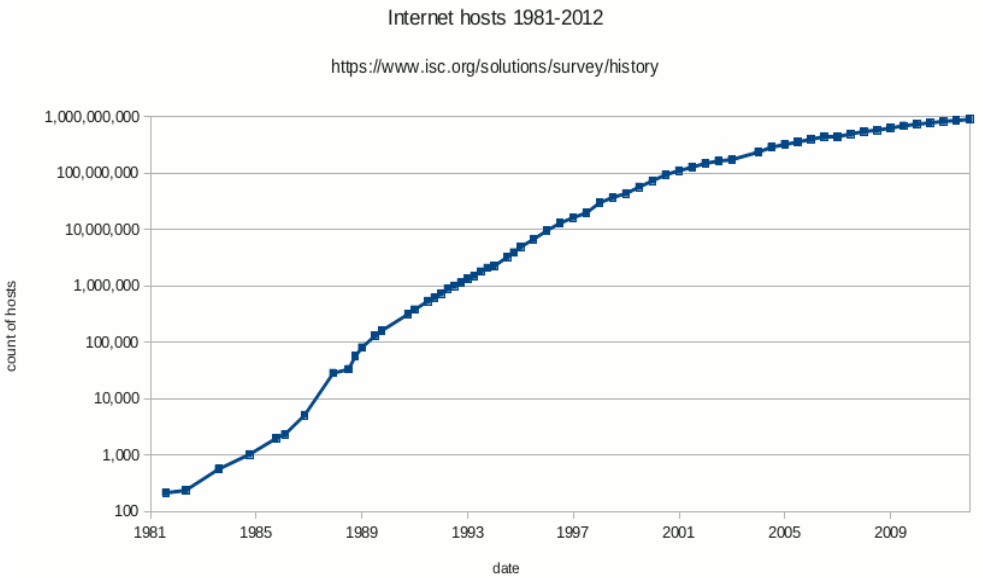
A short history of computer networking

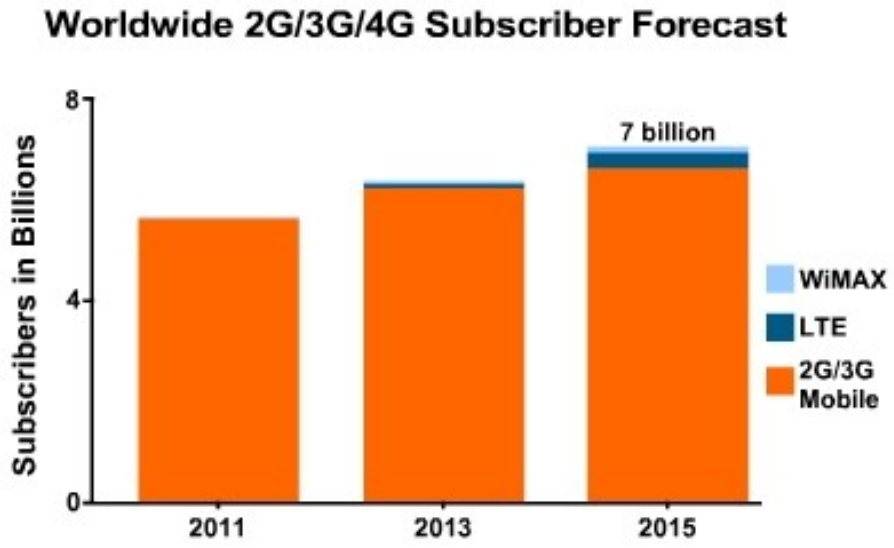
The Internet graph in 1969



The Internet graph about 15 years ago...







© Infonetics Research, *2G/3G/4G (LTE and WiMAX) Infrastructure and Subscribers Quarterly Market Size, Share, and Forecasts*, March 2011

Internet Tomorrow ?

Gartner: Internet of Things will Grow Exponentially to 26 Billion Devices by 2020

The Internet of Things will reach **26 billion** connected devices in 2020, with an exponential growth of 30 times the installed base in 2009, when connected devices in the web were just **900 million**.

According to a new research by Gartner, The Internet of Things (IoT), **which excludes PCs, tablets and smartphones**, will generate incremental revenue exceeding $300 billion in services in 2020. The services include hardware, embedded software, communications services and information services associated with the things.

The growth in IoT will far exceed that of other connected devices. By 2020, the number of smartphones tablets and PCs in use will reach about 7.3 billion units. In contrast, the IoT will have expanded at a much faster rate, resulting in a population of about 26 billion units at that time, the research said.

Source : http://www.gartner.com/newsroom/id/2636073

Objectifs du cours

- Acquérir les bases de la conception de protocole de communications, indépendamment du niveau OSI.

- Mettre en œuvre les méthodes pour spécifier un protocole réseau se basant sur les propriétés du service réseau à offrir et des services déjà disponibles.

- Savoir implémenter et évaluer le bon fonctionnement de ce protocole réseau en utilisant les sockets BSD.

- Maîtriser les concepts avancés de sérialisation de données.

Modalités

- 16 séances (soir)/ 12 séances (jour).
- Documentation en anglais
- 50 % de cours, 50 % de labo/travaux dirigés.
- Évaluation :
  - 50 % d'examen écrit sur la théorie.
  - 50 % sur un **mini-projet d'implémentation de protocole** en binôme (entre 20h et 25h de travail, langage python)

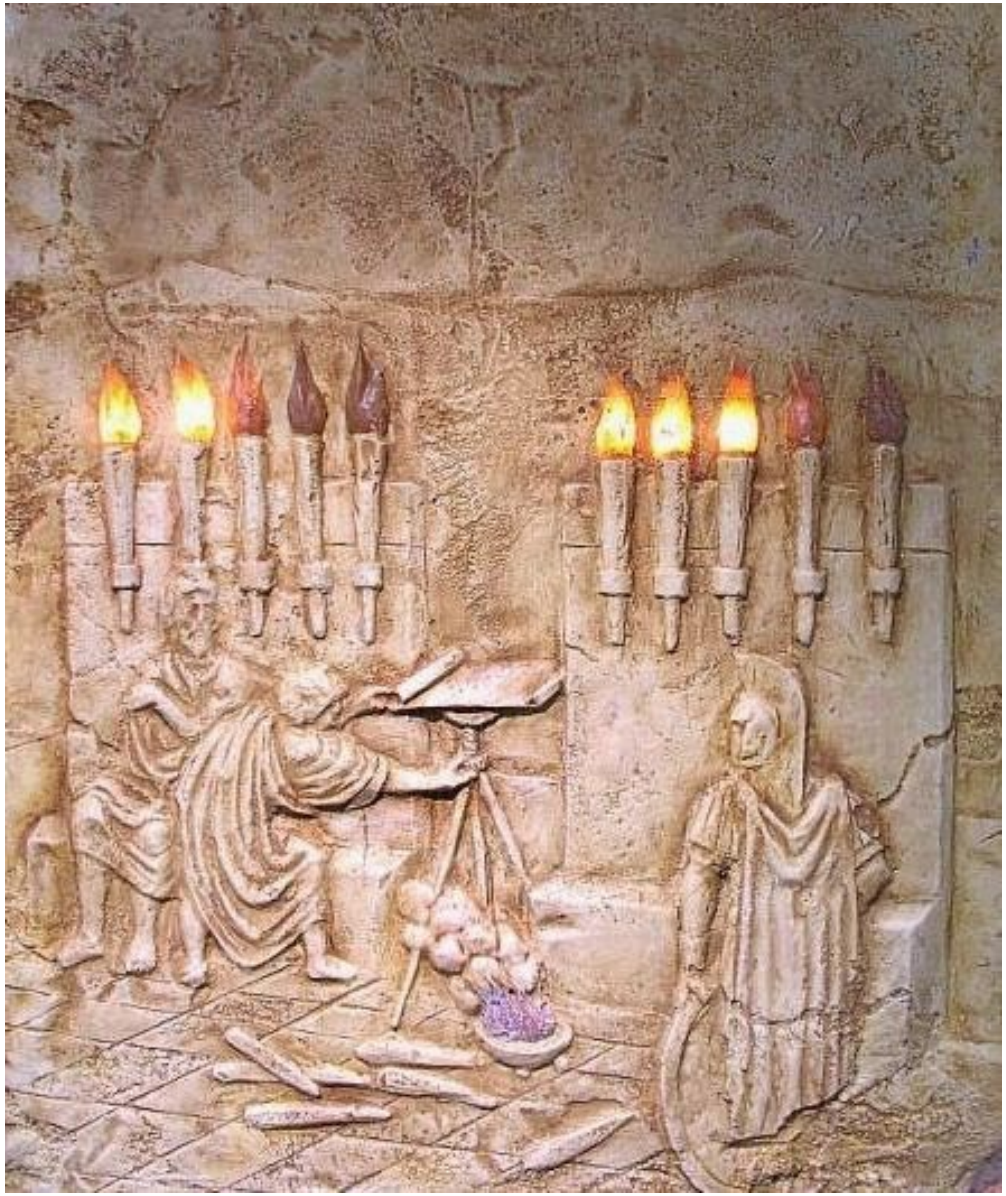
Contact et matériel de cours

- Par mail : mickael.hoerdt@hesge.ch
- Https://hepia.infolibre.ch
- Https://mattermost.hepiapp.ch

Bibliographie indicative

- Computer Networking : Principles, Protocols and Practice, Olivier Bonaventure
- Protocol Engineering, Hartmut Konig, Springer Verlag 2012 ISBN 978-3-642-44093-9
- Design and validation of Computer protocols, Prentice Hall Software Series, Gerard J. Holzman 1997

Polybius Square , -150 BC



- The sender of the message starts by raising two torch on the left side and wait until the received had done the same.

- He then indicates a row number, then a column number on the square to encode a letter

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

What is a protocol ?

A set of rules for the order in which messages of particular types are exchanged between N peers. This includes :

- Format spécification for valid messages (syntax)
- Rules for data exchange (grammar)
- A vocabulary of message and their meaning (semantics)

=> Similar to a language

What is a protocol ?

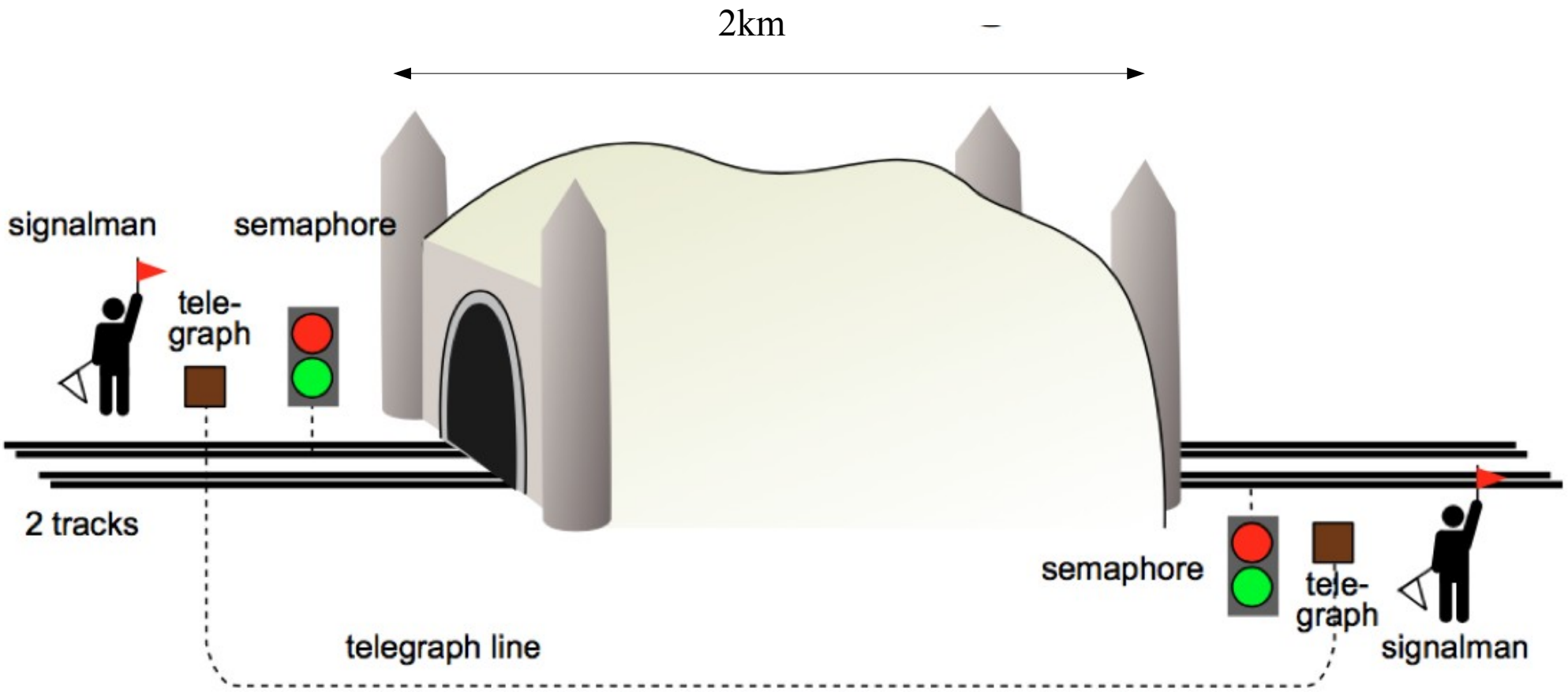
- It’s more than just implementation
- It’s a development process :
  - Requirements
  - Specification and validation
  - Implementation
  - Test and evaluation

A difficult task, even for simple protocols

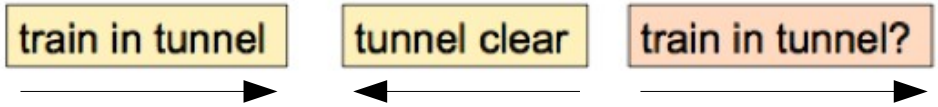
Bad protocol design or human failure ?

- The Clayton Tunnel accident: rail crash in Clayton, West Sussex, UK in 1861 due to a false signal
- Example for a failure of a simple communication protocol
- 21 dead, 176 injured

The setting

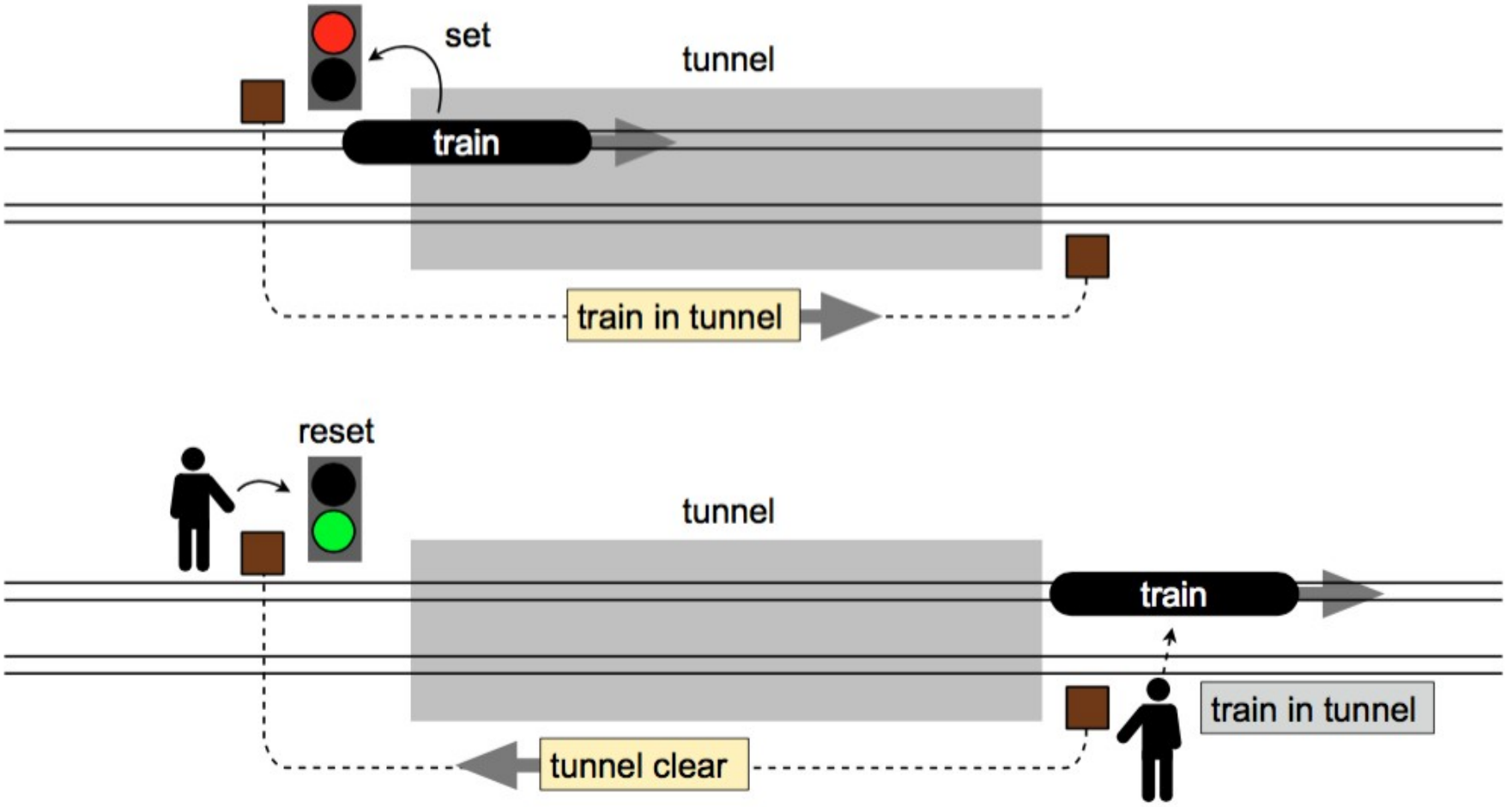


3 codes/messages: Whith train direction



- A semaphore blocks automatically once a train passes.
- It is reset by a signalman, if the other signalman reports that the train left the tunnel

The protocol

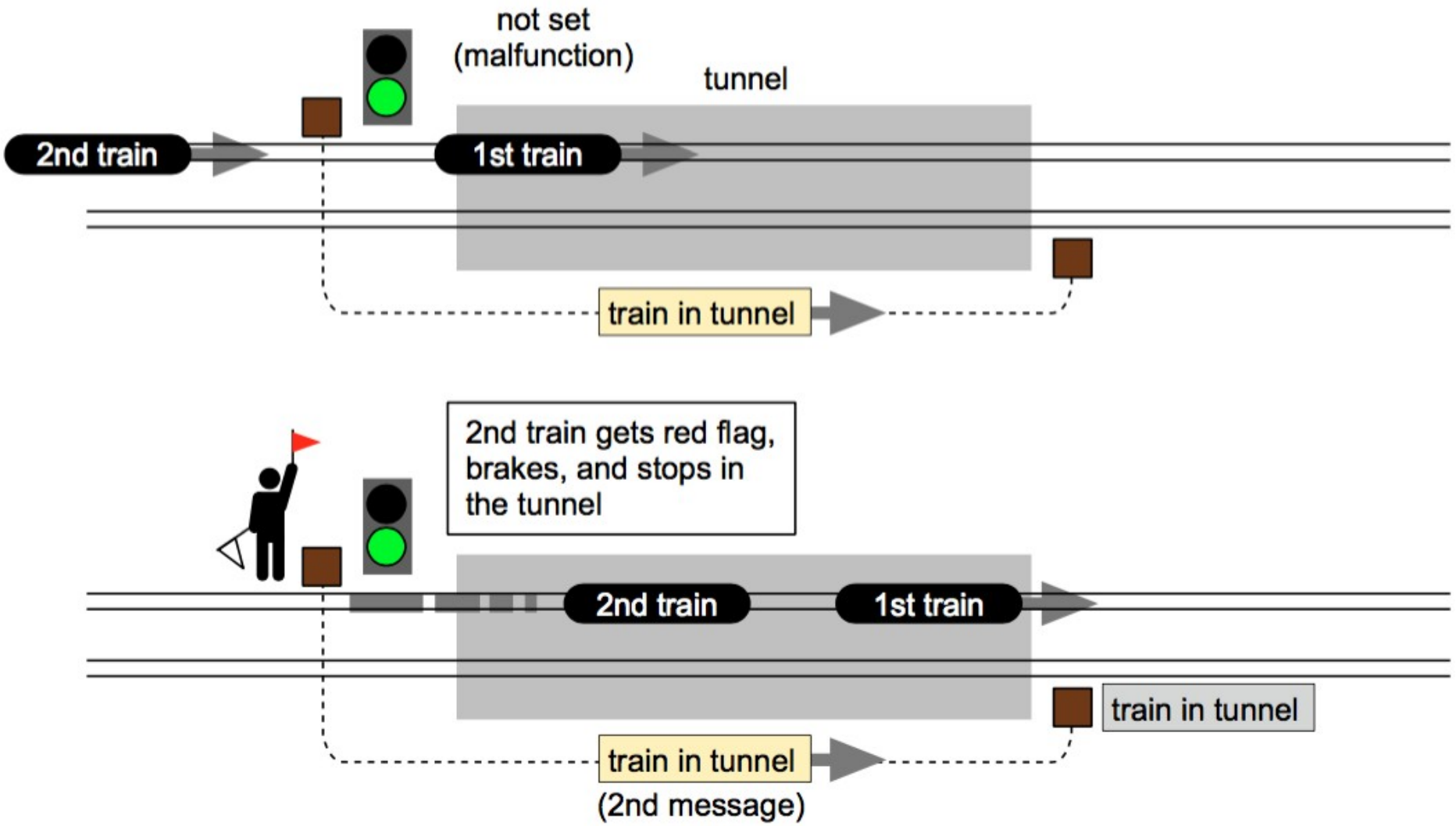


The protocol(2)

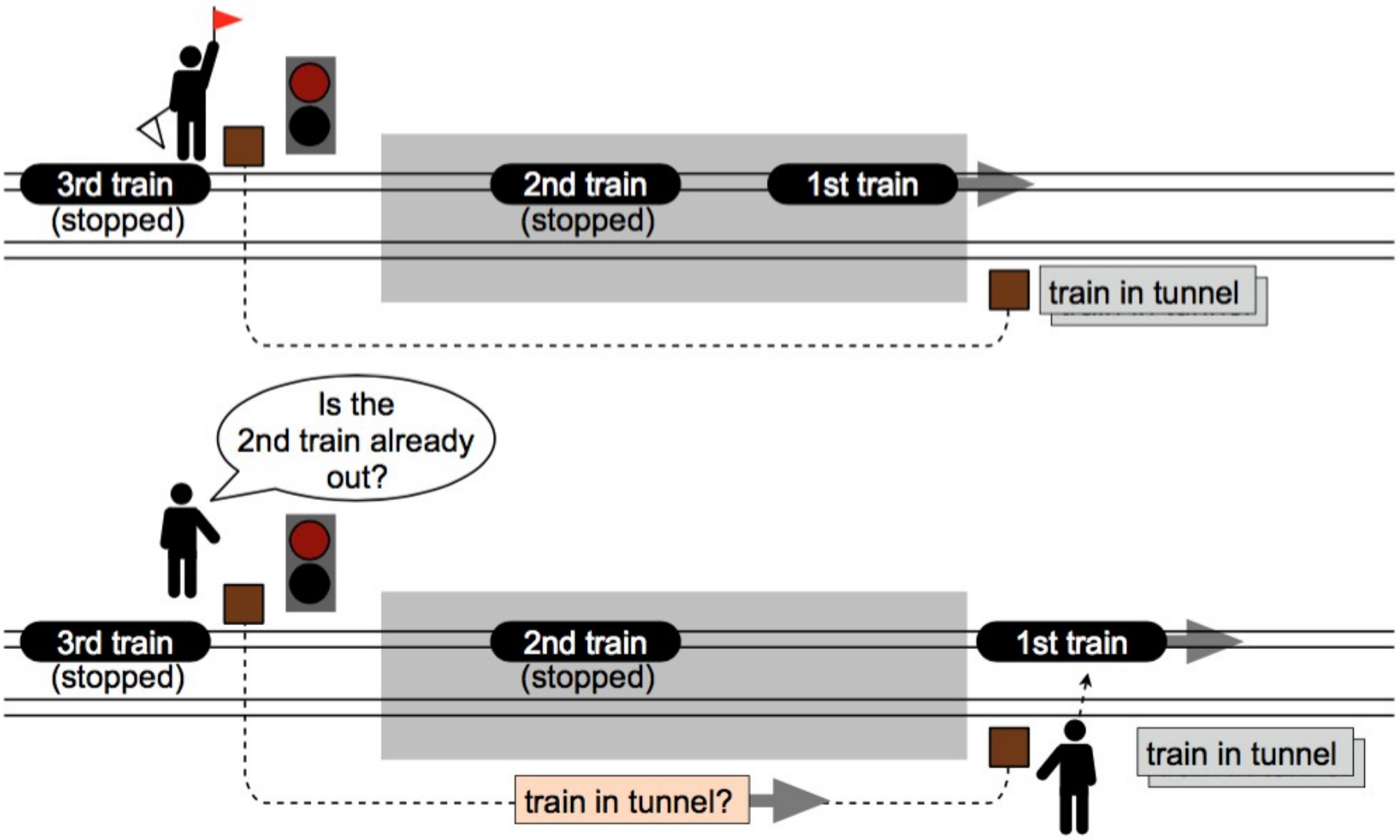
- The protocol should ensure that only one train per track is in the tunnel
- In case of a semaphore malfunction the signalmen are notified and use their flags
- The third message (“train in tunnel?”) is optional and can be used to request a message again

Is this protocol reliable?

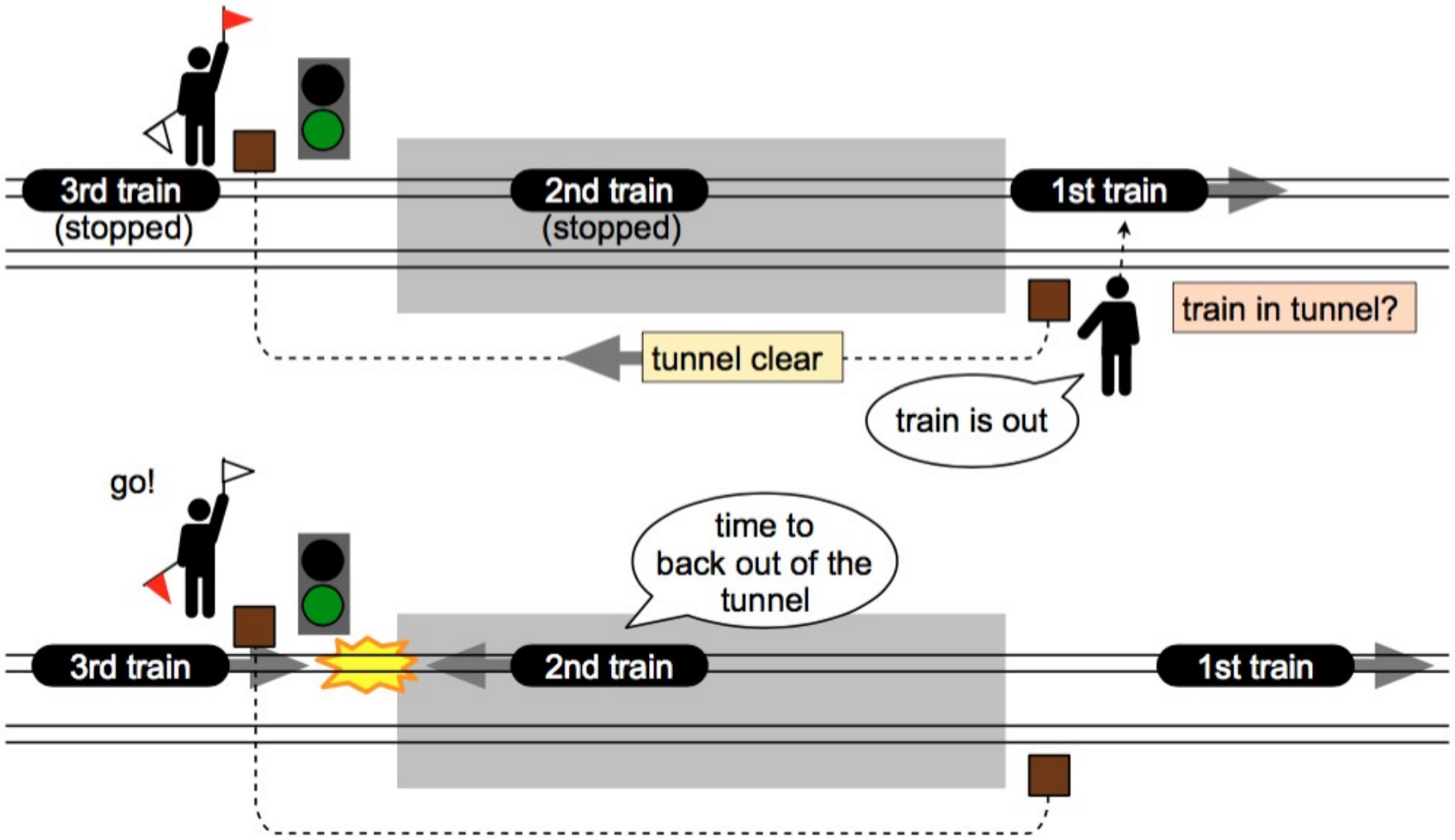
The accident(1)



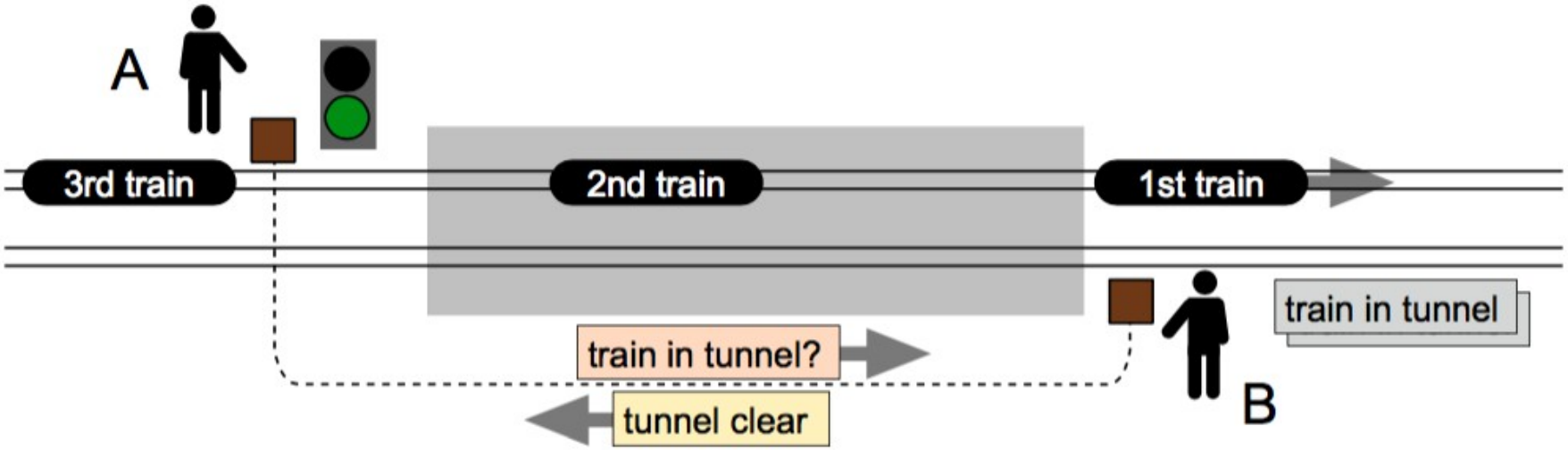
The accident(2)



The accident(3)



Who is to blame ?



- The driver of the second train? ...but he followed the signal.
- Signalman A, who misinterpreted the “tunnel clear” message?
  ... but how could he know which train was meant?
- Signalman B, who did not react on two “train in tunnel” msgs.?
  ... but this was **not specified**!

Conclusion

- The signalmen did not have the appropriate set of messages
- An unexpected case occurred that could not be handled by the protocol.
- The protocol could not recover from an error, it was *incomplete* in this sense

Lessons learned

- Consider that errors might occur (expect the unexpected)
- Allow to handle unexpected errors
- Check whether unnecessary or invalid assumptions are made

The problem of protocol design

- **How to find the appropriate rules** for communication that are minimal, logically consistent, complete, and efficient?

  ... difficult.

- **First,** we begin with a thorough specification of the rules, the messages and assumptions about the environment, ... all **elements of the protocol**

Protocol structure

It's about formalizating the use of a communication channel, for instance
(Holzman 1997) :

- Initialization and termination of data Exchange.

- Synchronisation of senders and receivers.

- Detection and correction of transmission Errors.

- Formatting and encoding of data.

Various level of abstraction can describe each items, how to to know when a description is well described enough ?

=> First step is to structure the description

Five elements of a protocol

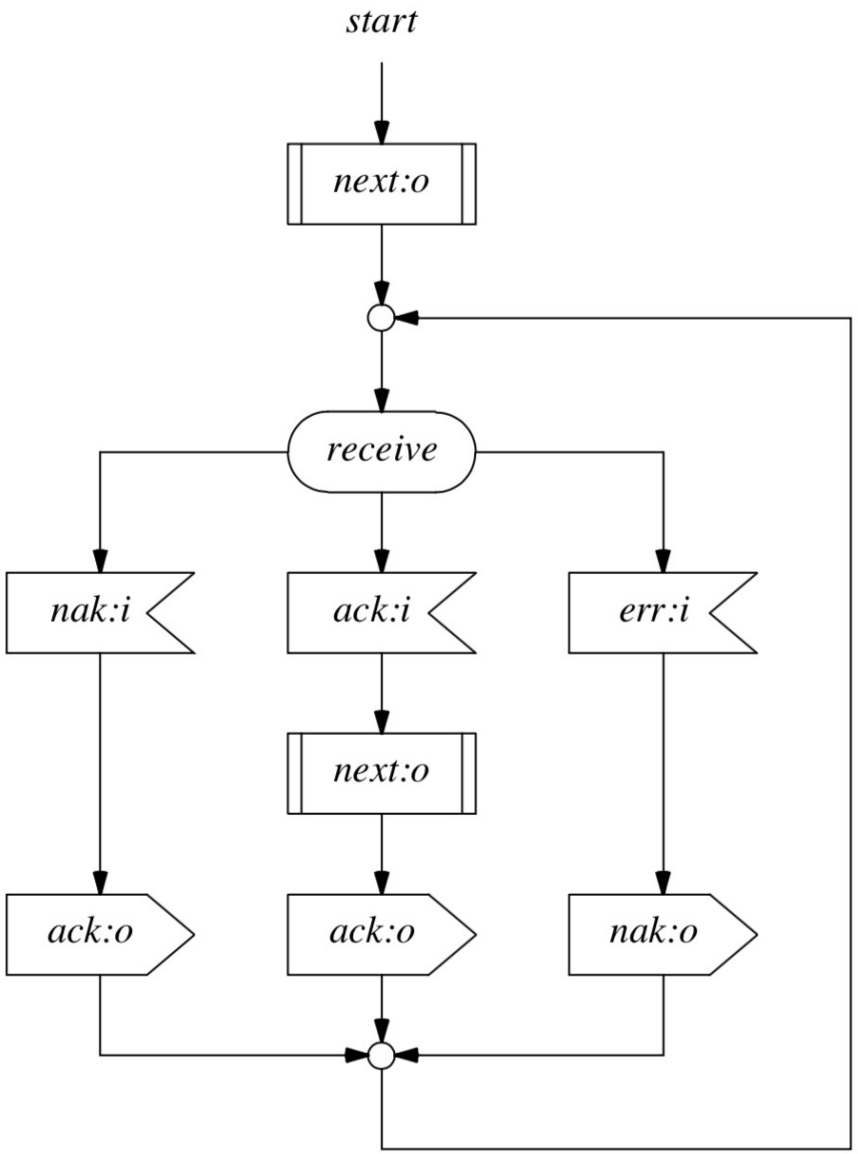
Elements of a protocol specification (Holzmann 1997):

1. The **service** to be provided by the protocol
2. The **assumptions about the environment** in which the protocol is executed
3. The **vocabulary** of messages used to implement the protocol
4. The **encoding** (format) of each message in the vocabulary
5. The **procedure rules** guarding the consistency of message exchanges

One example, still incomplete

See Lynch Protocol from Holzman's book chapter 2. p. 22 for an example of each 5 elements

Let's try to formalize Lynch's protocol rules

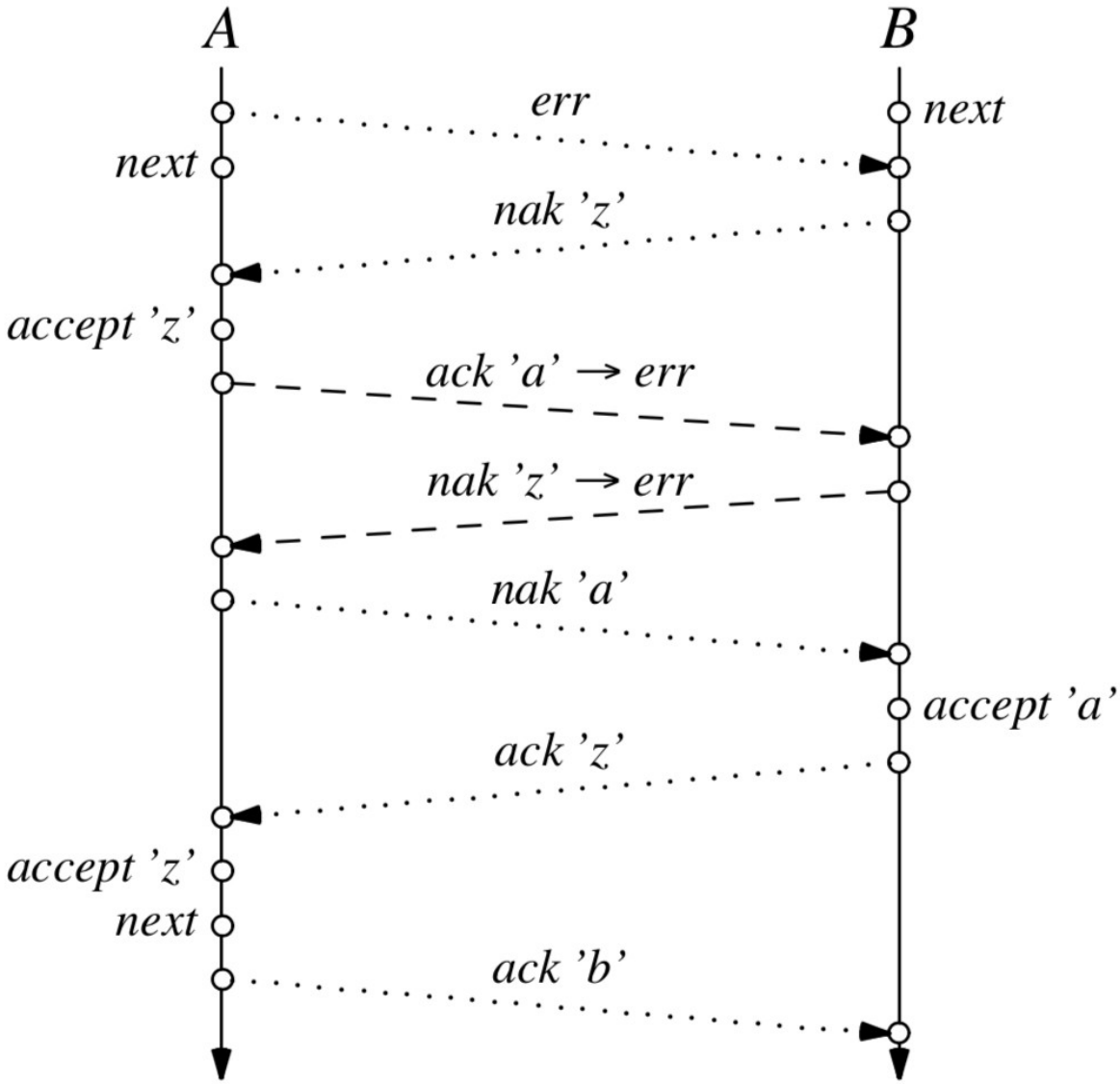


i: variable containing next input byte

o: variable containing next ouput byte

Holzmann 1997

And another unexpected case appears :



Holzmann 1997

10 rules for protocol design

1- Make sure that the problem is well-defined. All design criteria, requirements and constraints, should be enumerated before a design is started

2- Define the service to be performed at every level of abstraction before deciding which structures should be used to realize these services (what comes before how).

3- Design external functionality before internal functionality. First consider the solution as a black-box and decide how it should interact with its environment. Then decide how the black-box can internally be organized. Likely it consists of smaller black-boxes that can be refined in a similar fashion

4- Keep it simple. Fancy protocols are buggier than simple ones; they are harder to implement, harder to verify, and often less efficient. There are few truly complex problems in protocol design. Problems that appear complex are often just simple problems huddled together. Our job as designers is to identify the simpler problems, separate them, and then solve them individually.

5- Do not connect what is independent. Separate orthogonal concerns.

10 rules for protocol design

6- Do not introduce what is immaterial. Do not restrict what is irrelevant. A good design is “open-ended,” i.e., easily extensible. A good design solves a class of problems rather than a single instance.

7- Before implementing a design, build a high-level prototype and verify that the design criteria are met

8- Implement the design, measure its performance, and if necessary, optimize it

9- Check that the final optimized implementation is equivalent to the high-level design that was verified

10 - Don’t skip Rules 1 to 7.

The most violated rule is rule 10.