



ANNÉE PASSERELLE EN INGÉNIERIE



# Chapitre 10 — TCP et UDP

Introduction à la Téléinformatique

---

Vincent Audergon · HEIA-FR

Année académique 2026-2027

- 01** La couche transport
- 02** UDP : rapidité et simplicité
- 03** TCP : fiabilité et contrôle
- 04** TCP ou UDP ?

À l'issue de ce chapitre, vous saurez :

- **définir** les protocoles TCP et UDP et leur rôle dans le modèle en couches ;
- **expliquer** leurs différences, cas d'utilisation, avantages et inconvénients ;
- **expliquer** la notion de « well-known ports » ;
- **décrire** l'architecture client-serveur ;
- **décrire** l'établissement (three-way) et la terminaison (four-way) d'une connexion TCP.

SECTION

# 01 —

## La couche transport

### 01 La couche transport

02 UDP : rapidité et simplicité

03 TCP : fiabilité et contrôle

04 TCP ou UDP ?

• LA COUCHE TRANSPORT

# TCP vs UDP



# D'un hôte à une application

Les couches 1 à 3 amènent un paquet d'un **hôte** à un autre. Reste à livrer la donnée à la **bonne application** : c'est le rôle de la couche transport (L4 OSI, L3 TCP/IP).

## DÉFINITION • COUCHE TRANSPORT

Fournit un « tunnel » de communication entre deux **applications**, via deux protocoles principaux : **TCP** (fiabilité, contrôle) et **UDP** (rapidité, simplicité).

## DÉFINITION • PORT

Identifiant numérique (16 bits, 0 à 65535) associé à une application sur un hôte. Le couple **IP + port** identifie une application de façon unique (ex. 127.0.0.1:80 ).

## REMARQUE

Les ports sont comme des **boîtes aux lettres** : chaque application a la sienne pour recevoir ses messages, ce qui permet le **multiplexage** des communications sur un même hôte.

# Trois plages de ports

| Plage         | Nom                     | Usage                                                     |
|---------------|-------------------------|-----------------------------------------------------------|
| 0 – 1023      | Well-known ports        | Services standards : HTTP (80), HTTPS (443), SSH (22)...  |
| 1024 – 49151  | Registered ports        | Applications spécifiques, enregistrables auprès de l'IANA |
| 49152 – 65535 | Dynamic / private ports | Ports éphémères, communications temporaires               |

**ATTENTION**

Un port est lié à son **protocole de transport** : le port 53 TCP n'est **pas** le port 53 UDP. On note souvent `PORT/PROTO` (ex. `53/UDP` ).

# Quelques ports bien connus

## ■ TCP:

- Port 22 : SSH (Secure Shell): accès sécurisé à distance
- Port 80 : HTTP (Hypertext Transfer Protocol): navigation web
- Port 25 : SMTP (Simple Mail Transfer Protocol): envoi d'emails

## ■ UDP:

- Port 53 : DNS (Domain Name System): résolution de noms de domaine

### REMARQUE

Les ports enregistrés sont listés sur le site de l'IANA : <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>).

## ? QUESTION 1

À l'aide du registre des ports de l'IANA, indiquez pour chaque service le(s) port(s) et le protocole de transport. Attention : l'IANA réserve souvent TCP **et** UDP par précaution, alors qu'un seul est utilisé en pratique. Indiquez le transport **réellement** employé, et signalez les cas où le registre diffère de l'usage.

## RÉPONSE

- **IMAP** : 143 TCP (UDP réservé par l'IANA, mais non utilisé en pratique)
- **POP3** : 110 TCP (UDP réservé par l'IANA, mais non utilisé en pratique)
- **SMTP** : 25 TCP (UDP réservé par l'IANA, mais non utilisé en pratique)
- **DNS** : 53 TCP et UDP
- **NTP** : 123 UDP (peut utiliser TCP si explicitement configuré)
- **MQTT** : 1883 TCP (UDP réservé par l'IANA, mais non utilisé en pratique)

Remarques : pour la plupart des services, il existe une version sécurisée utilisant TLS, qui utilise généralement un port différent (par exemple, IMAPS sur le port 993 TCP, POP3S sur le port 995 TCP, SMTPS sur le port 465 TCP).

- Utilisée par la plupart des applications réseau : un **serveur** écoute sur un port bien connu, et un **client** se connecte à ce port pour demander un service.
- C'est le client qui initie la communication, le serveur se contente d'attendre les requêtes.
- Le serveur peut gérer plusieurs clients simultanément, chacun identifié par sa paire **IP + port** unique.
- D'autres architectures existent, comme **peer-to-peer** (P2P) ou **publish-subscribe**, mais elles sont moins courantes pour les applications grand public.

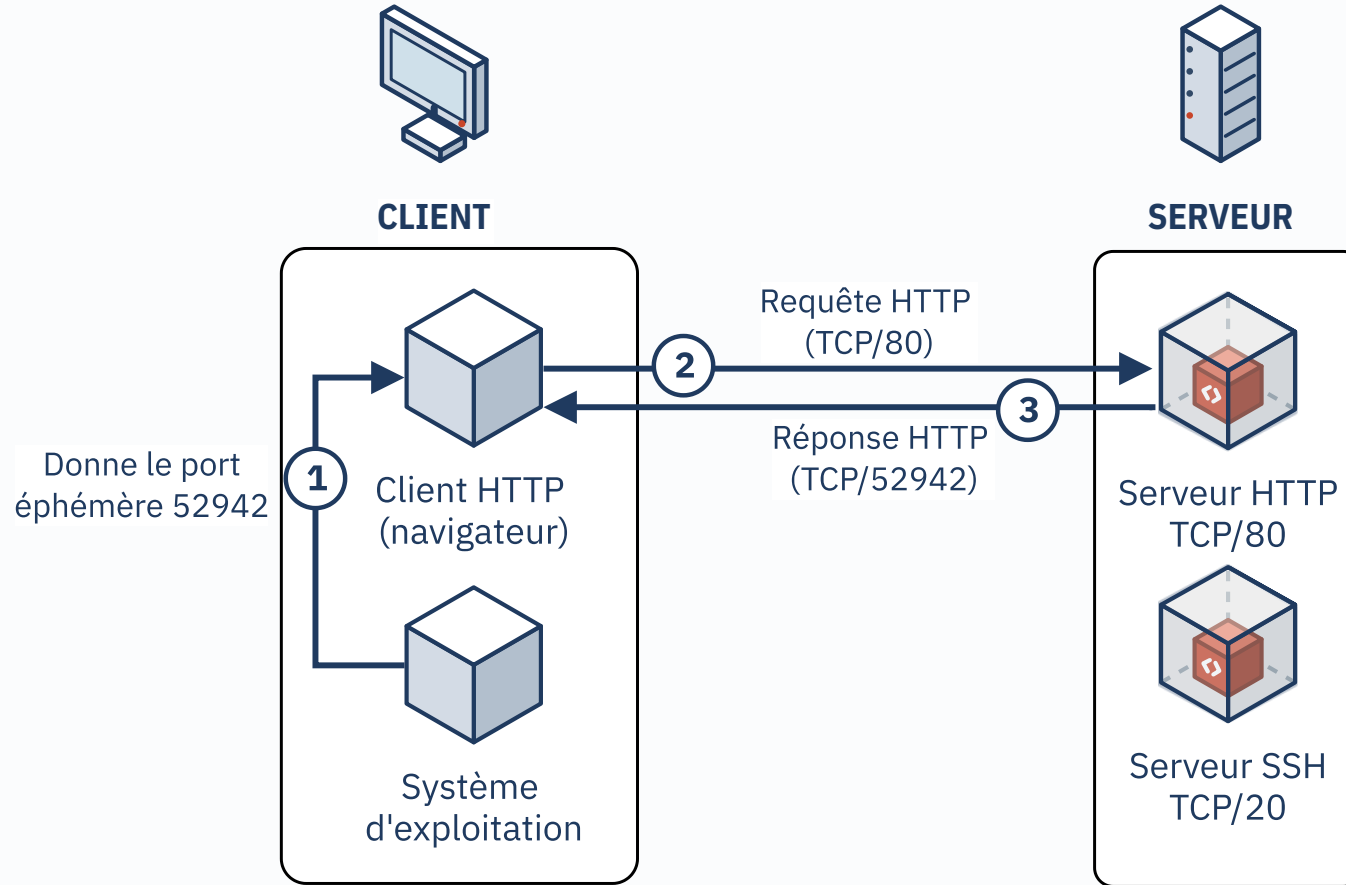
## ? QUESTION 2

Un navigateur ouvre une connexion vers un serveur web. Quel type de port le système lui attribue-t-il, et dans quelle plage se situe-t-il ?

### RÉPONSE

Un **port éphémère** (dynamique), choisi automatiquement dans la plage **49152 – 65535**. Le serveur, lui, écoute sur un **well-known port** (80 ou 443).

# Exemple : serveur web



SECTION

# 02 —

## UDP : rapidité et simplicité

01 La couche transport

**02 UDP : rapidité et simplicité**

03 TCP : fiabilité et contrôle

04 TCP ou UDP ?

## DÉFINITION • UDP

*User Datagram Protocol* — protocole **sans connexion** (*connectionless*) qui envoie des **datagrammes** sans garantie de livraison ni d'ordre.

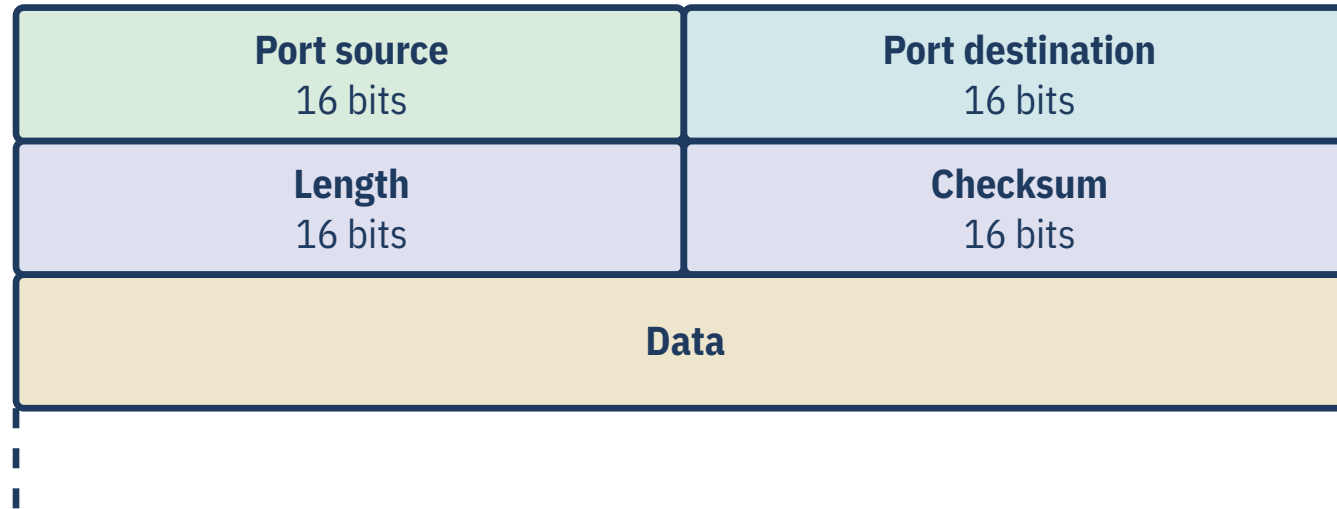
- Pas d'établissement de connexion, pas de retransmission. (principe du « fire-and-forget »)
- Supporte **unicast**, **multicast** et **broadcast**.
- Idéal quand la **vitesse** prime sur la fiabilité : streaming vidéo/audio, jeux en temps réel...

Seulement **quatre champs**, tous simples :

- **Port source** (16 bits)
- **Port destination** (16 bits)
- **Longueur** (16 bits) : taille totale du datagramme
- **Checksum** (16 bits) : intégrité des données

# Structure d'un datagramme UDP

Datagramme UDP



SECTION

03

# TCP : fiabilité et contrôle

- 01 La couche transport
- 02 UDP : rapidité et simplicité
- 03 TCP : fiabilité et contrôle**
- 04 TCP ou UDP ?

## DÉFINITION • TCP

*Transmission Control Protocol* : protocole **orienté connexion** qui garantit une transmission fiable, ordonnée et sans perte de **segments**.

TCP gère :

- **Transfert ordonné** : segments numérotés, réassemblés à la réception.
- **Retransmission** : les segments non accusés de réception sont renvoyés.
- **Contrôle de flux** : adapte le débit à la capacité du **récepteur**.
- **Contrôle de congestion** : adapte le débit à l'état du **réseau**.
- **Détection d'erreurs** : via un checksum par segment.

## Contrôle de flux

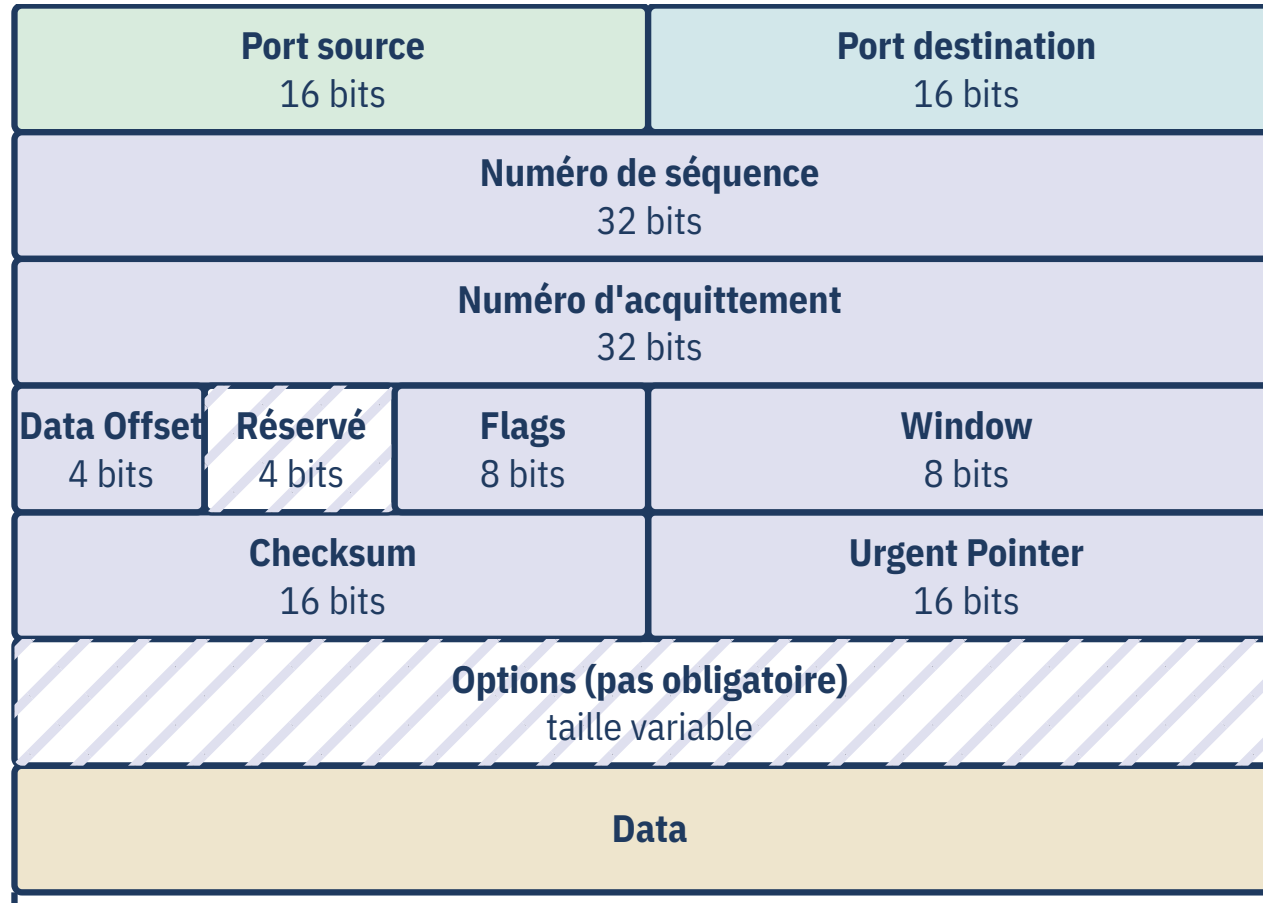
- Négociation **émetteur** ↔ **récepteur**.
- S'appuie sur le champ **Window** de l'en-tête.
- Protège le récepteur d'une **surcharge**.

## Contrôle de congestion

- Réaction à l'état du **réseau**.
- Déclenché par pertes ou délais accrus.
- Protège les équipements **intermédiaires**.

- TCP : FIABILITÉ ET CONTRÔLE

# L'en-tête TCP

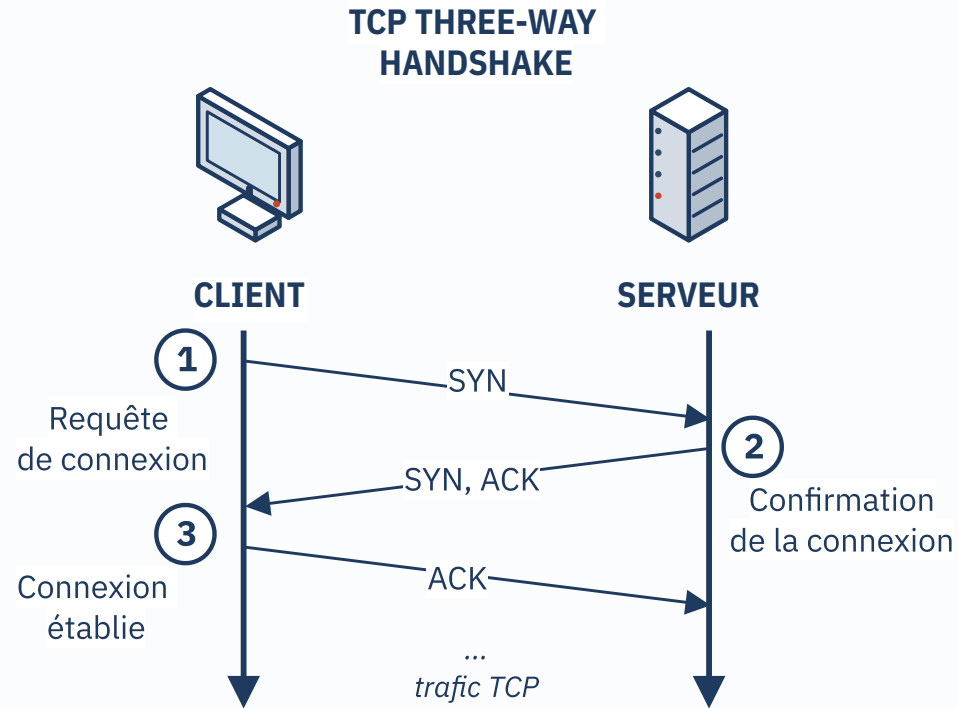


Bien plus riche que l'en-tête UDP.

- Port source et destination (16 bits chacun)
- Numéro de séquence et numéro d'acquittement (32 bits chacun), pour **l'ordonnancement** et le **renvoi** des segments non reçus.
- Flags : SYN, ACK, FIN, RST, PSH, URG, pour **l'établissement**, la **termination** et le **contrôle** de la connexion.
- Window size (16 bits) : pour le **contrôle de flux**.
- Checksum (16 bits) : pour la **détection d'erreurs**.
- Options (variable) : pour des fonctionnalités avancées (ex. taille de fenêtre étendue, timestamp).

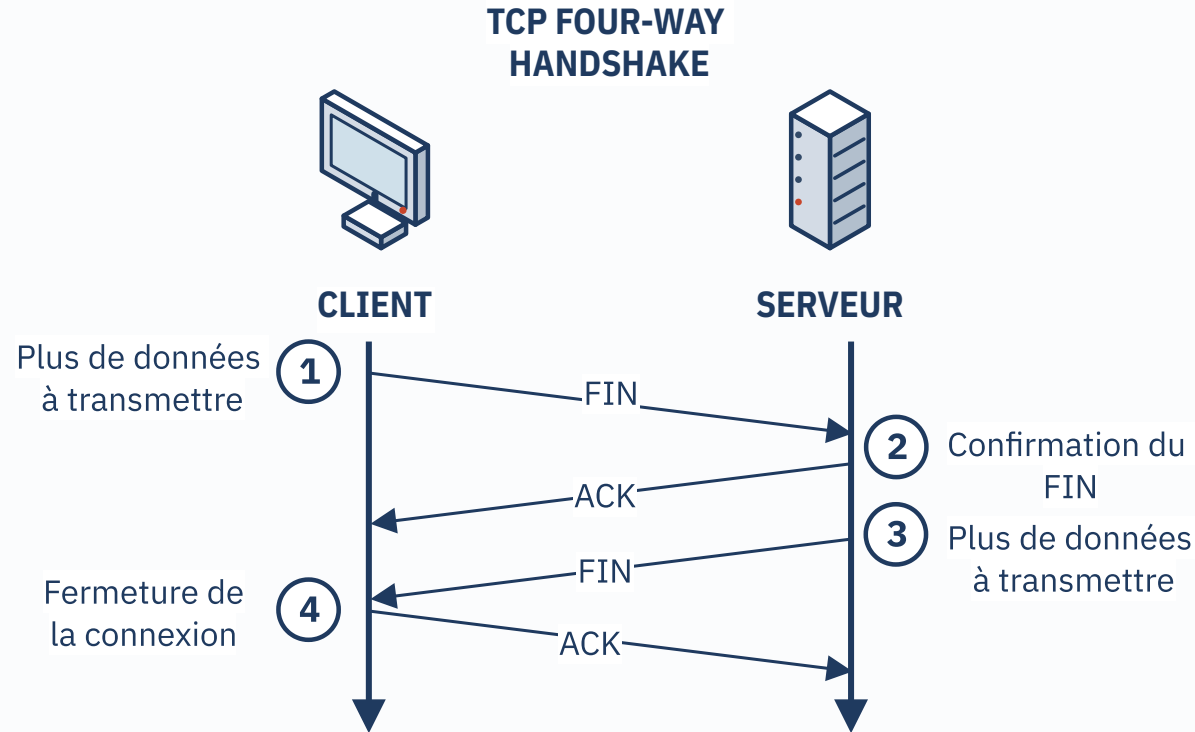
- Le flag URG et le champ Urgent Pointer permettent de signaler des données prioritaires, mais sont rarement utilisés dans les applications modernes.
- Le « data offset » indique la taille de l'en-tête TCP, permettant de localiser le début des données.

# Établir la connexion : three-way handshake



SYN (1) PUIS SYN-ACK (2) PUIS ACK (3) : LA CONNEXION EST ÉTABLIE

# Terminer la connexion : four-way handshake



FIN (1) PUIS ACK (2) PUIS FIN (3) PUIS ACK (4) : CHAQUE PARTIE FERME SON PROPRE SENS

SECTION

04

# TCP ou UDP ?

- 01 La couche transport
- 02 UDP : rapidité et simplicité
- 03 TCP : fiabilité et contrôle
- 04 TCP ou UDP ?**

# Choisir le bon protocole

|                    | TCP                            | UDP                   |
|--------------------|--------------------------------|-----------------------|
| <b>Connexion</b>   | Orientée connexion             | Sans connexion        |
| <b>Fiabilité</b>   | Garantie (ACK, retransmission) | Aucune garantie       |
| <b>Ordre</b>       | Toujours respecté              | Non garanti           |
| <b>Vitesse</b>     | Plus lent (overhead)           | Rapide, léger         |
| <b>Cas d'usage</b> | Web, email, fichiers           | Streaming, jeux, VoIP |

## ASTUCE

Question à se poser : « **la perte d'un peu de données est-elle acceptable ?** » Si oui → UDP. Si l'intégrité prime → TCP.

### ? QUESTION 3

TCP ou UDP pour : (A) le transfert d'une facture, (B) un appel vidéo en direct, (C) la synchronisation de personnages dans un jeu multijoueur ?

#### RÉPONSE

A. **TCP**, l'intégrité des données est cruciale. B et C. **UDP**, la faible latence prime, la perte de quelques paquets est tolérable.

### ■ CE QU'IL FAUT RETENIR

- La couche transport livre les données à la bonne **application**, grâce aux **ports** (16 bits, well-known / registered / dynamic).
- **UDP** : sans connexion, rapide, sans garantie. Idéal pour le temps réel.
- **TCP** : orienté connexion, fiable, ordonné. Three-way handshake pour ouvrir, four-way handshake pour fermer.
- **Contrôle de flux** (protège le récepteur) vs **contrôle de congestion** (protège le réseau).

**Des questions ?**

Les sources utilisées pour ce cours (normes, documentation officielle, articles, ouvrages de référence) sont citées et listées intégralement dans la **bibliographie du support de cours écrit**, disponible sur le site du cours :

<https://api-teleinfo.isc.heia-fr.ch>

