Université Nice Sophia Antipolis
Membre de UNIVERSITÉ CÔTE D'AZUR

iut réseaux & télécoms Sophia Antipolis

Articulation des disciplines en Réseaux et Télécoms

Module M1101

2017-2018

L. Sassatelli

Plan

- Articulation des enseignements technologiques au sein du monde numérique
  - Cas d'usage : Visionner une vidéo YouTube sur depuis un réseau mobile
- Streaming vidéo sur HTTP : principe
- Des maths pour le réseau :
  - Contrôle de débits de sessions concurrentes
  - Les caches dans le réseau

Sources

- 1ère partie: slides faîtes par les RT2 2016-2017:

Alice Desplanches, Ricardo Janer, Johanna Klose, Keanu Lorfèvre

- Computer Networking : A Top-Down Approach 6th ed. J.F. Kurose and K.W. Ross
- R. Srikant, "The Mathematics of Internet Congestion Control“, Springer Science, 2004
- M. Dehghan, L. Massoulie, D. Towsley, D. Menasche and Y. C. Tay, "A utility optimization approach to network cache design," IEEE International Conference on Computer Communications (INFOCOM), San Francisco, CA, 2016

La société à l'âge de l'internet

Sphère éducation

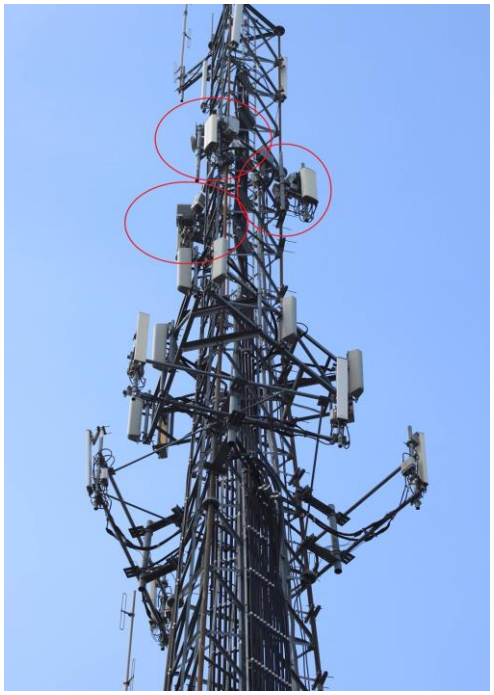
Sphère entreprise

Sphère communication

Une vidéo YouTube sur smartphone

Du téléphone portable

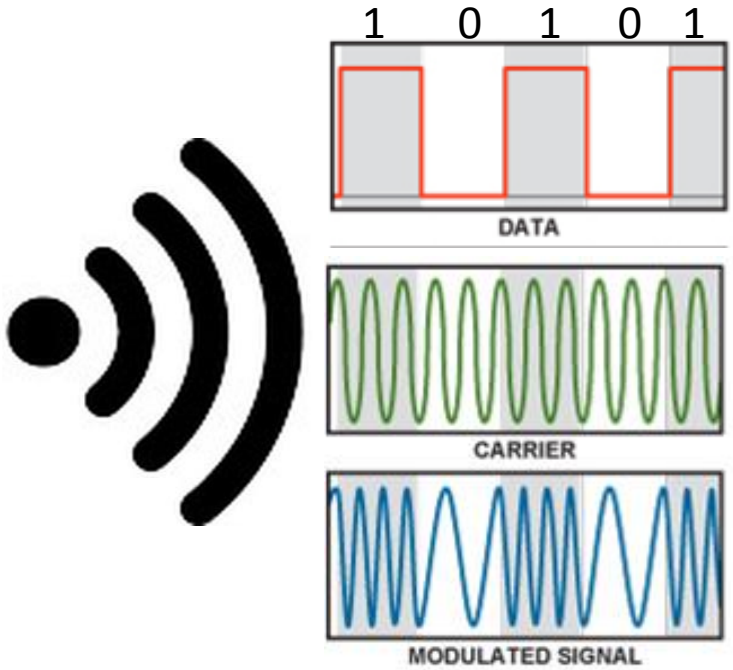
A la station de base



Physique

Premier saut : transmission radio

Information codée en binaire, puis transformée sous la forme d'onde par le contrôle du phénomène électromagnétique



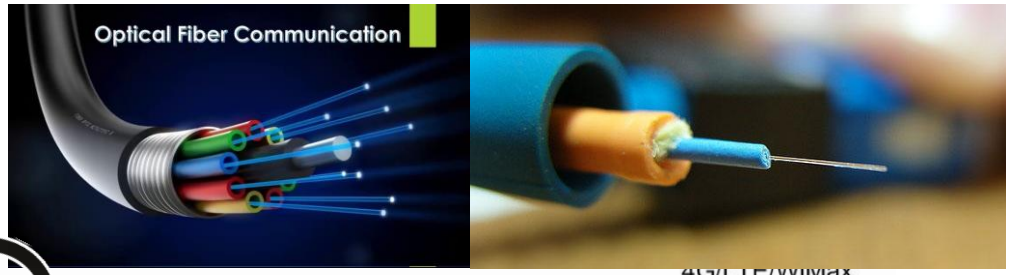
Télécoms

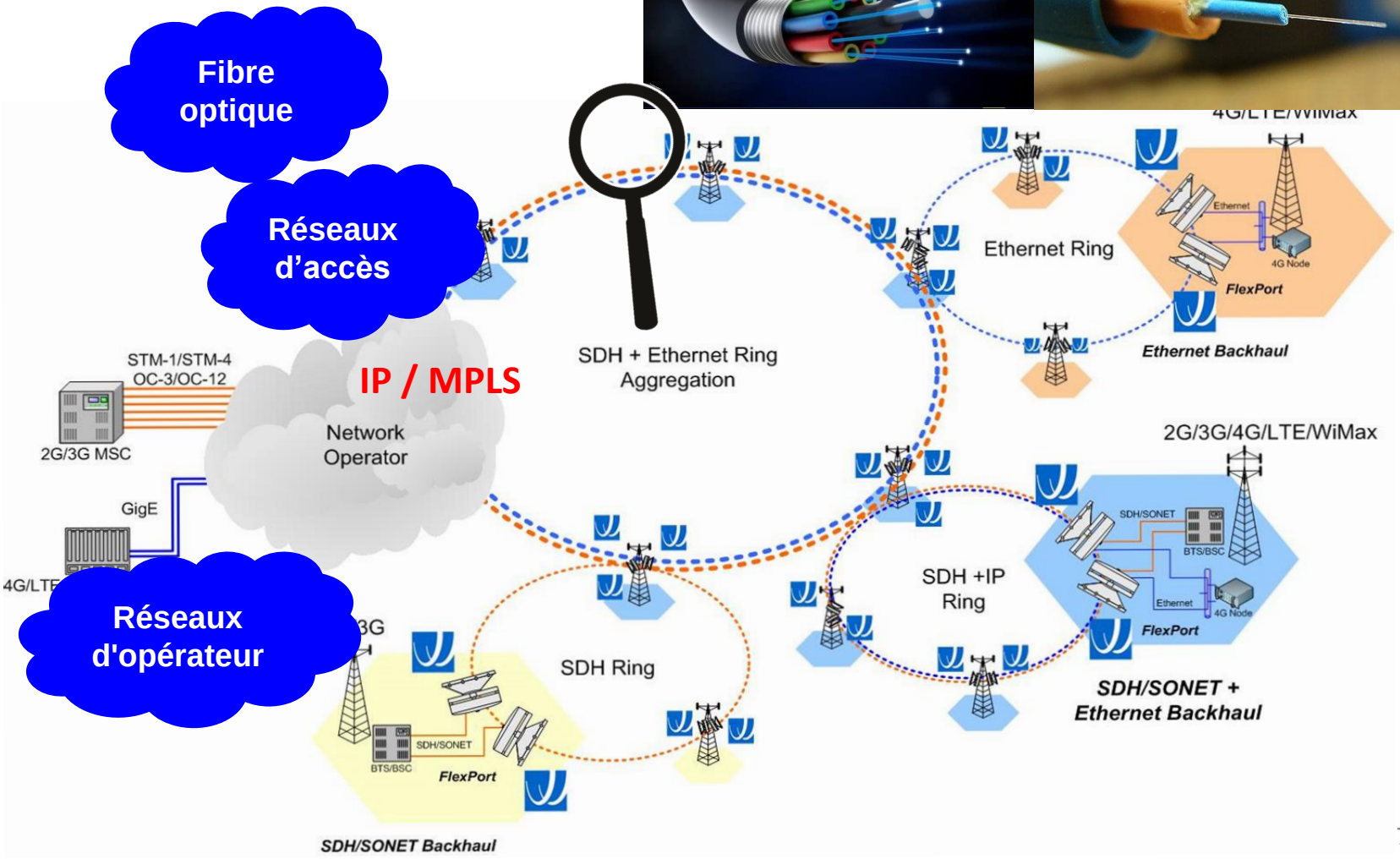
$$s_{FSK}(t) = A_c \cos\left(2\pi \int_0^t [f_c + f_\Delta x_m(\tau)]\, d\tau\right)$$

Maths

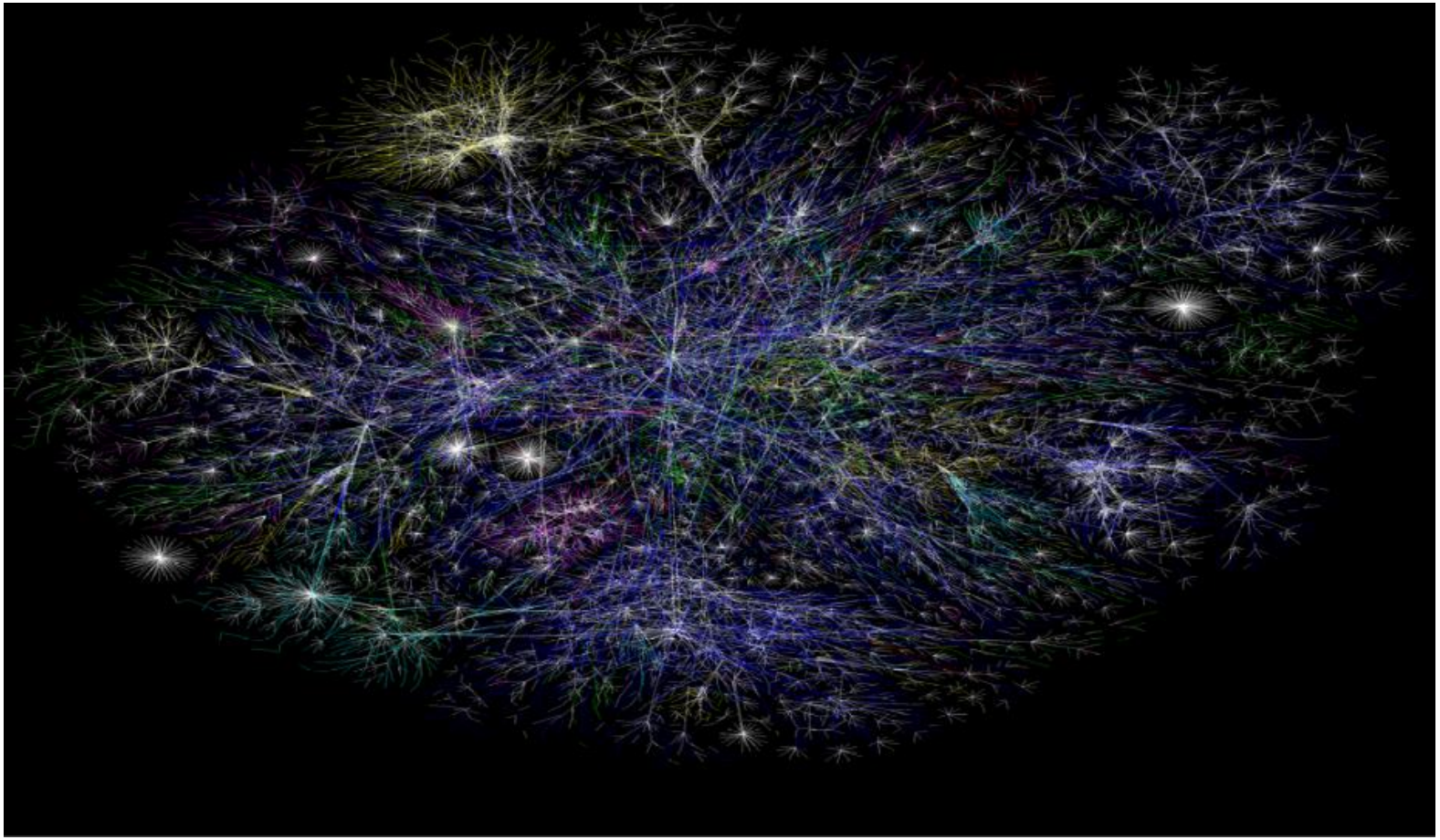
Analyse des signaux

Derrière l'antenne relai : le réseau

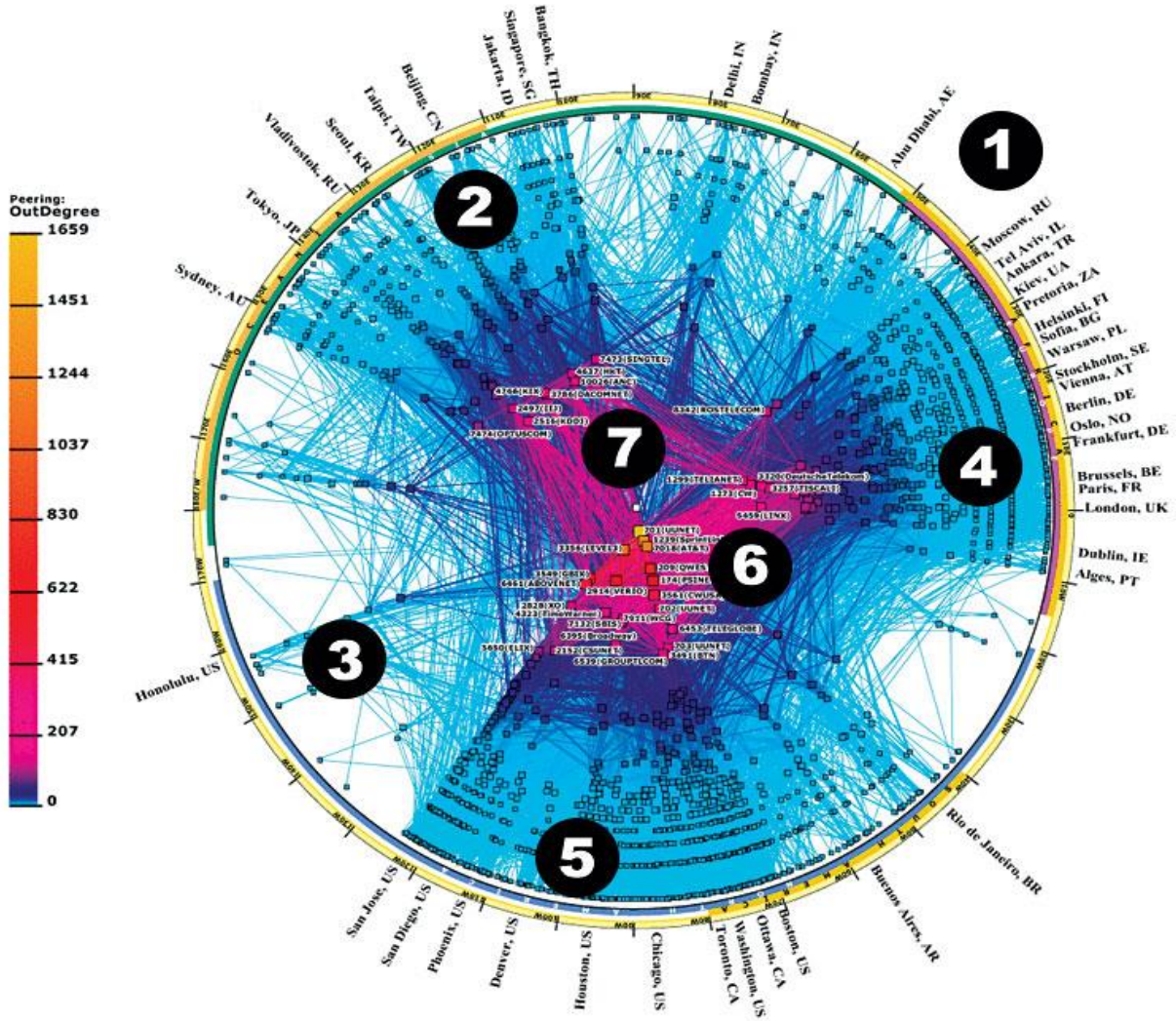




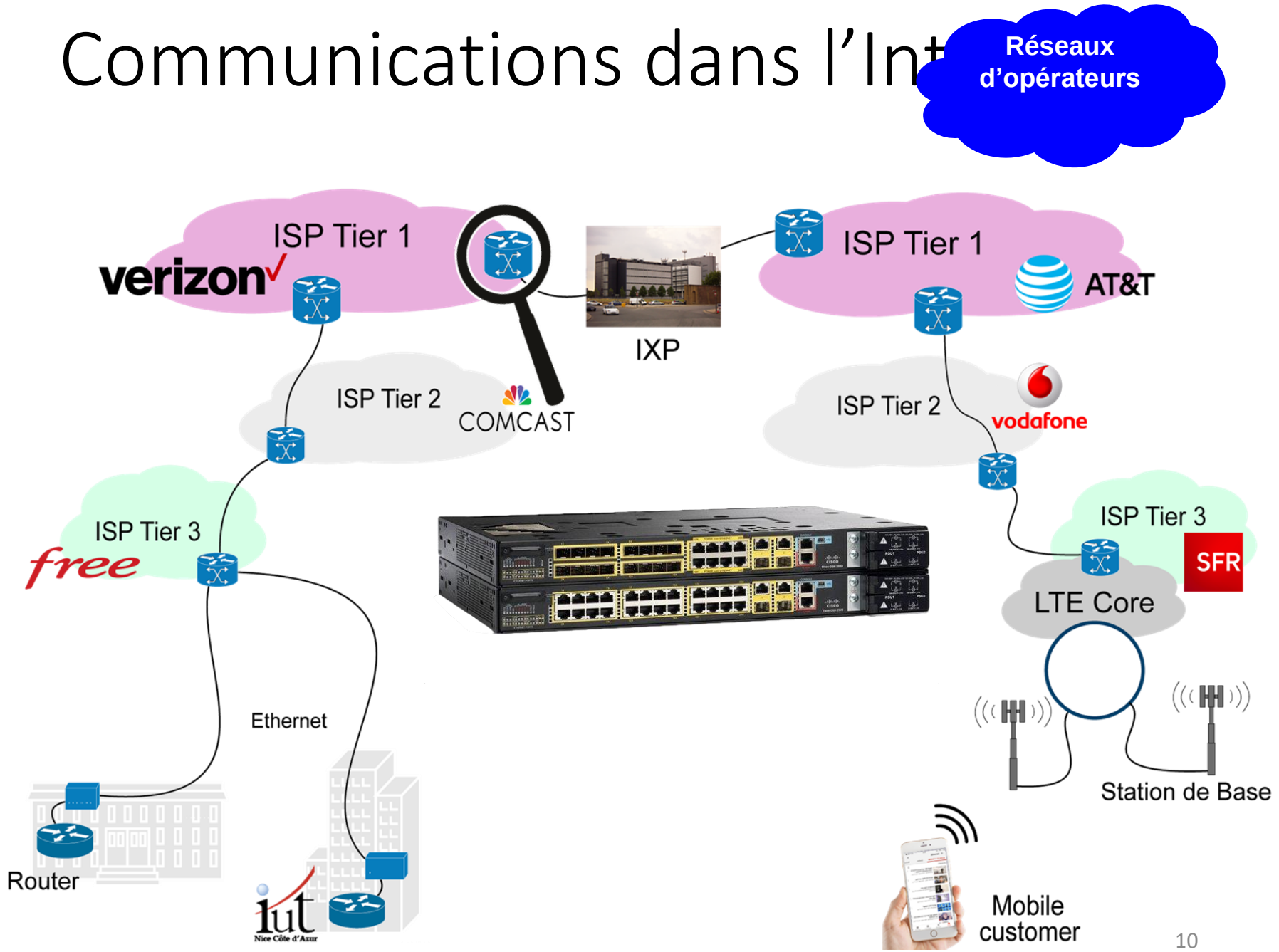
L’Internet à l’échelle globale



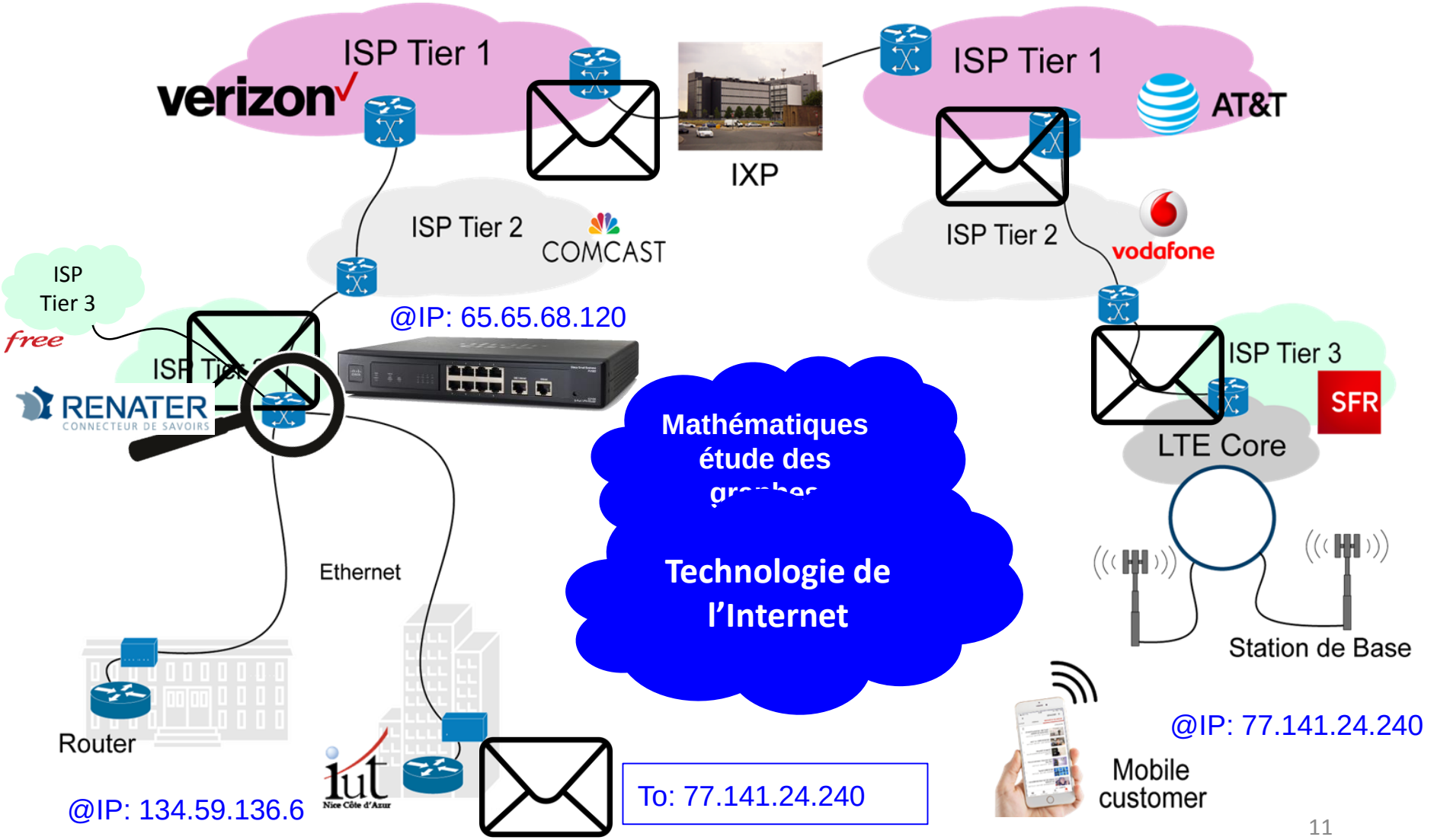
L’Internet à l’échelle globale



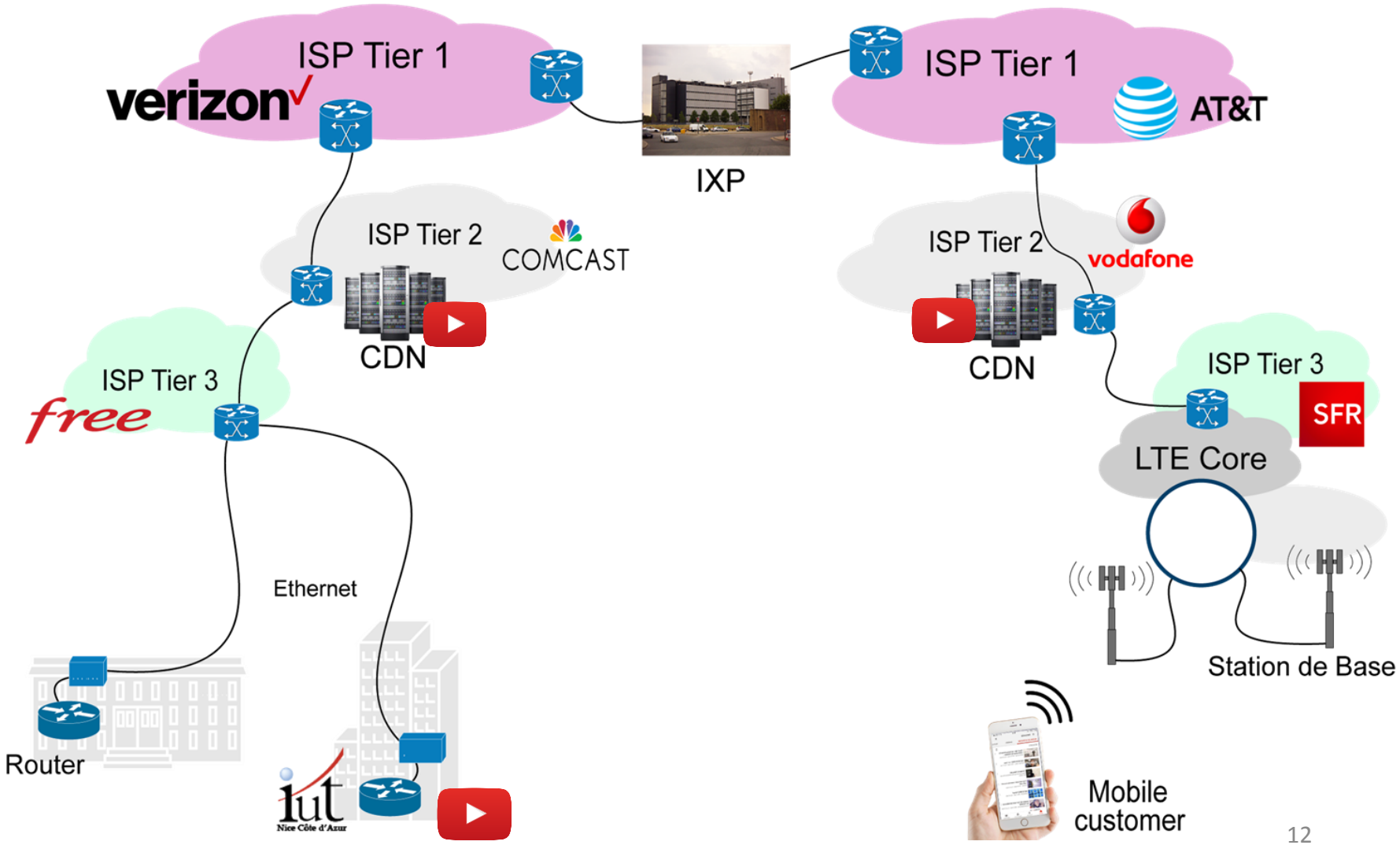
Communications dans l'Int



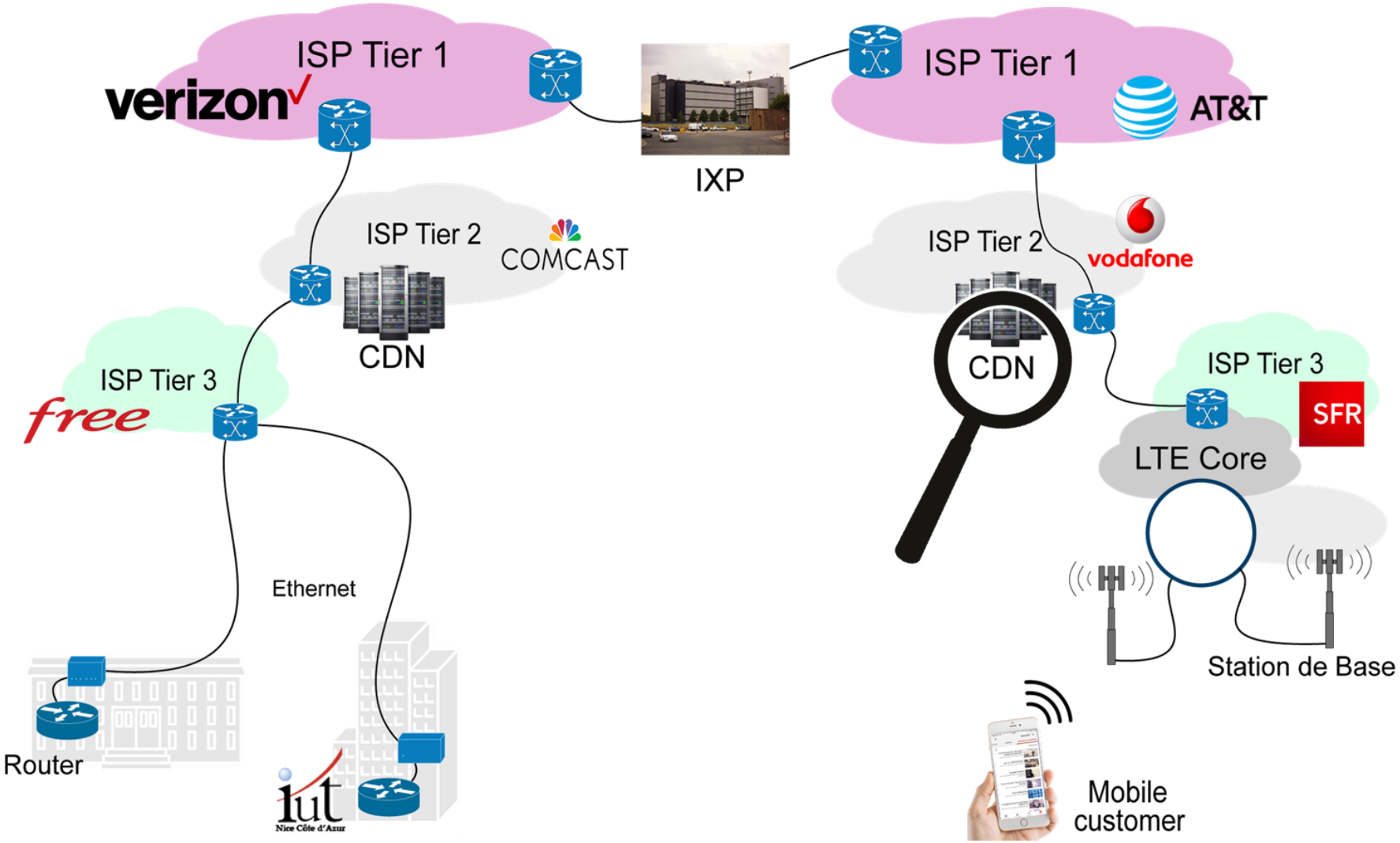
Communications dans l'Internet



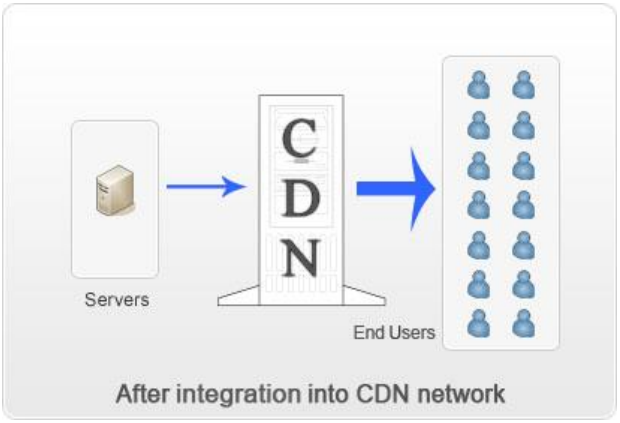
Distribution de contenu vidéo dans l'Internet



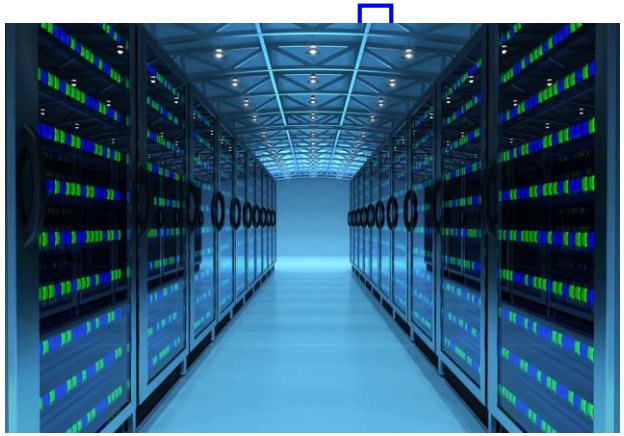
Où est stocké le contenu ?



Où est stocké le contenu ?



Réseau de distribution de contenu



Data center = Ferme d'ordinateurs

```php
<?php

class foo {
    public $file = "test.txt";
    public $data = "text";
    function __destruct()
    {
        file_put_contents($this->file, $this->data);
    }
}

$file_name = $_GET['session_filename'];

print "Readfile " . $file_name ."<br>";
if(!file_exists($file_name))
{
    print "no file\n";
}
else
{
    unserialize(file_get_contents($file_name));
}
```

PhP

```java
package org.nuxeo.upcoming.test;

import java.io.Serializable;

public class EventTest extends RepositoryOSGITestCase {

    public void testEventIsHandled() throws Exception {
        //check for this name, if you want, using the debugger on the event l
        String eventName = "my event: there are many like it but this one is
        //get the event production service, so we can send an event
        EventProducer producer=Framework.getService(EventProducer.class);
        assertNotNull("Found the event producer ok?",producer);
        //create a fake event object with some bogus values
        Map<String, Serializable> props = new HashMap<String, Serializable>()
        props.put("test-name", "testEventBasic");
        //the event context is a really an "EventGenerator" in some sense
        EventContext context=new InlineEventContext(null,props);
        Event syntheticEvent = context.newEvent(eventName);
        //the event service is the center of all event processing...
        EventService service=Framework.getService(EventService.class);
        assertNotNull("Found event service ok?",service);
        //EventListenerHelper is the wrapper around our DocumentCreationListe
        service.addEventListener(new EventListenerHelper());
```

Java

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <arpa/inet.h>

void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
    }

    bzero(&monAddr, sizeof(monAddr));
    monAddr.sin_family = AF_INET;
    monAddr.sin_port = htons(ports.port1);
    monAddr.sin_addr.s_addr = INADDR_ANY;
    bzero(&addrServ2, sizeof(addrServ2));
```

C

```python
# a scheduler class, to schedule and run events after a delay
class Scheduler:
    def __init__(self):
        # begin with no events
        self.events = []

    # after the delay, run the function
    def schedule(self, delay, function):
        if delay <= 0:
            # if no delay, run function straight away
            function()
        else:
            # queue the function
            self.events.append([delay, function])
            # sort the queue, smallest delay first
            self.events.sort(key = lambda event: event[0])

    def run(self):
        # loop while there are events to process
        while len(self.events) > 0:
            # get the first event
            delay, function = self.events.pop(0)
            # subtract time from remaining events
            for event in self.events:
                event[0] -= delay
            # wait for required delay
            pyb.delay(delay)
            # run function
            function()
```

Python

Programmation

Informatique

```php
<?php

class foo {
    public $file = "test.txt";
    public $data = "text";
    function __destruct()
    {
        file_put_contents($this->file, $this->data);
    }
}

$file_name = $_GET['session_filename'];

print "Readfile " . $file_name ."<br>";
if(!file_exists($file_name))
{
    print "no file\n";
}
else
{
    unserialize(file_get_contents($file_name));
}
```

PhP

```java
package org.nuxeo.upcoming.test;

import java.io.Serializable;

public class EventTest extends RepositoryOSGITestCase {

    public void testEventIsHandled() throws Exception {
        //check for this name, if you want, using the debugger on the event li
        String eventName = "my event: there are many like it but this one is

        //get the event production service, so we can send an event
        EventProducer producer=Framework.getService(EventProducer.class);
        assertNotNull("Found the event producer ok?",producer);

        //create a fake event object with some bogus values
        Map<String, Serializable> props = new HashMap<String, Serializable>();
        props.put("test-name", "testEventBasic");
        //the event context is a really an "EventGenerator" in some sense
        EventContext context=new InlineEventContext(null,props);
        Event syntheticEvent = context.newEvent(eventName);

        //the event service is the center of all event processing...
        EventService service=Framework.getService(EventService.class);
        assertNotNull("Found event service ok?",service);

        //EventListenerHelper is the wrapper around our DocumentCreationListe
        service.addEventListener(new EventListenerHelper());
```

Java

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <arpa/inet.h>

void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
    }

    bzero(&monAddr, sizeof(monAddr));
    monAddr.sin_family = AF_INET;
    monAddr.sin_port = htons(ports.port1);
    monAddr.sin_addr.s_addr = INADDR_ANY;
    bzero(&addrServ2, sizeof(addrServ2));
```

C

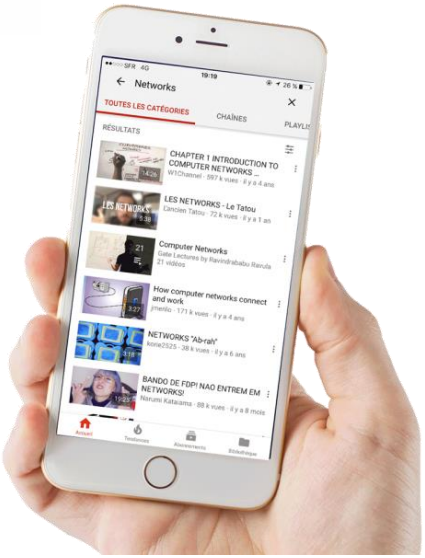
```python
# a scheduler class, to schedule and run events after a delay
class Scheduler:
    def __init__(self):
        # begin with no events
        self.events = []

    # after the delay, run the function
    def schedule(self, delay, function):
        if delay <= 0:
            # if no delay, run function straight away
            function()
        else:
            # queue the function
            self.events.append([delay, function])
            # sort the queue, smallest delay first
            self.events.sort(key = lambda event: event[0])

    def run(self):
        # loop while there are events to process
        while len(self.events) > 0:
            # get the first event
            delay, function = self.events.pop(0)
            # subtract time from remaining events
            for event in self.events:
                event[0] -= delay
            # wait for required delay
            pyb.delay(delay)
            # run function
            function()
```

Python

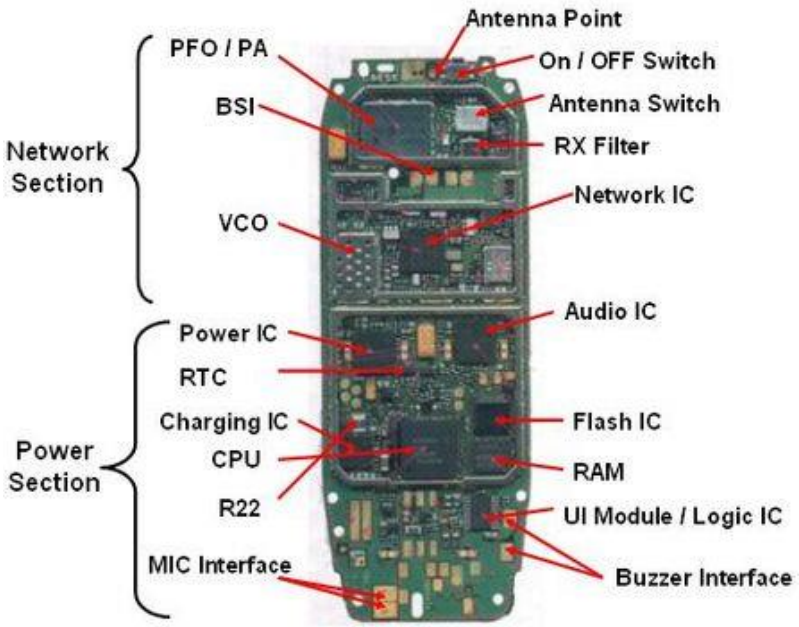
Programmation Android

Composition interne d’un téléphone portable

Electronique



Composition interne d'un téléphone portable



NOTES:

1. **UEM** = Logic IC + Charging IC + Audio IC + Power IC
2. **PFO** = Antenna Switch + PFO
3. **Flash IC** = RAM + Flash IC

Démodulateur FSK

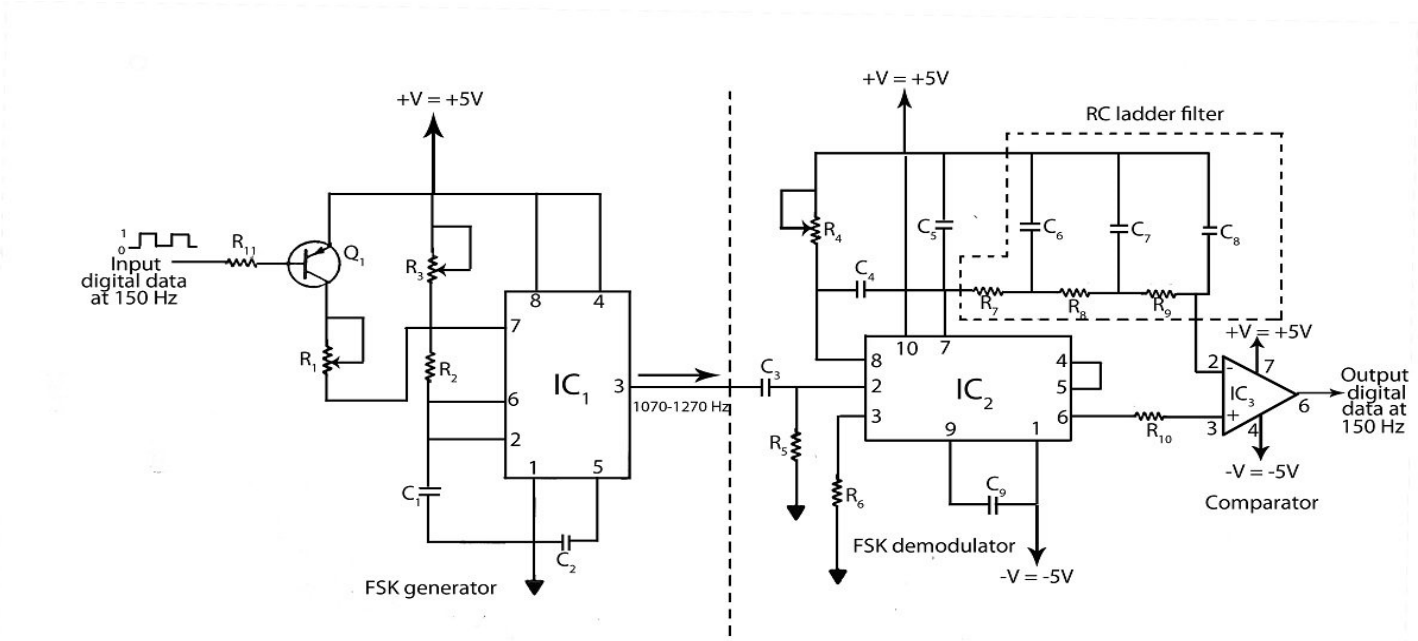
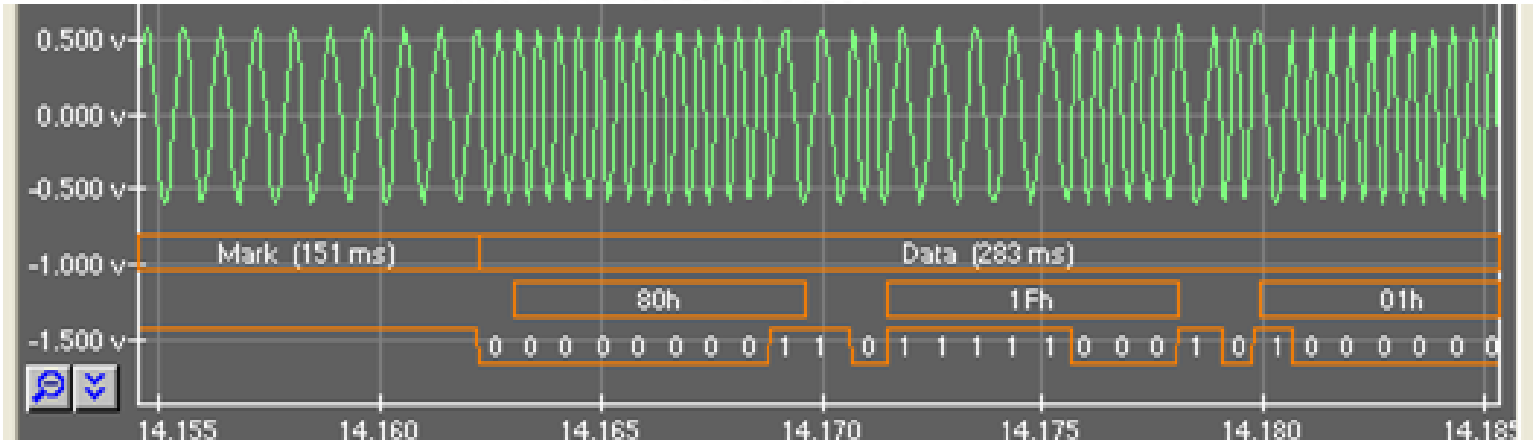


Figure 1-1 The 565 as an FSK demodulator

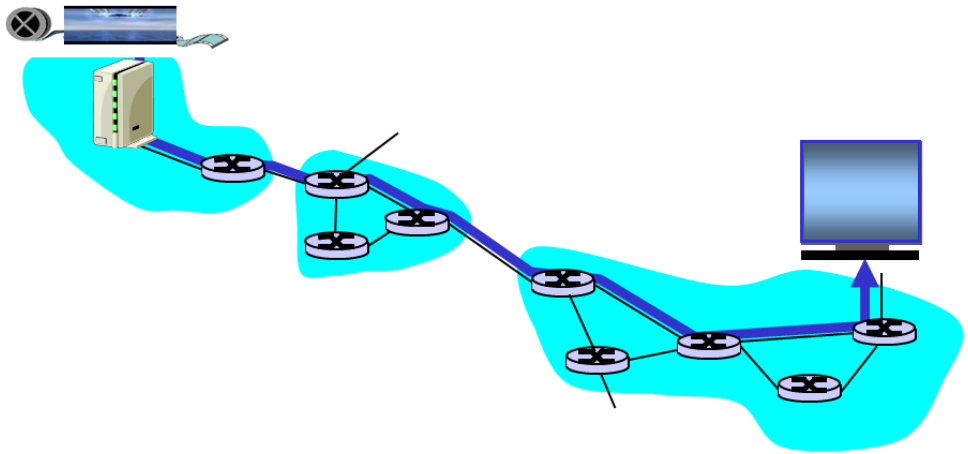


Plan

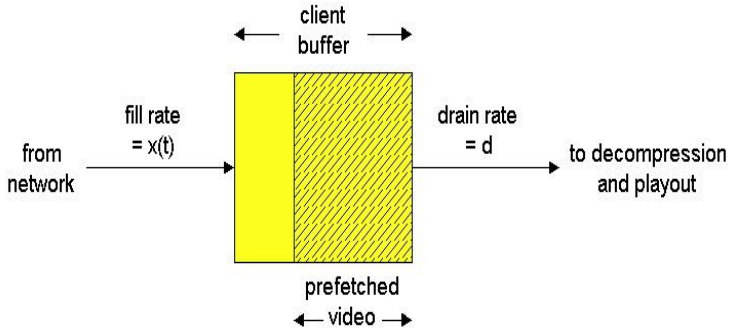
- Articulation des enseignements technologiques au sein du monde numérique
  - Cas d'usage : Visionner une vidéo YouTube sur depuis un réseau mobile
- Streaming vidéo sur HTTP : principe
- Des maths pour le réseau :
  - Contrôle de débits de sessions concurrentes
  - Les caches dans le réseau

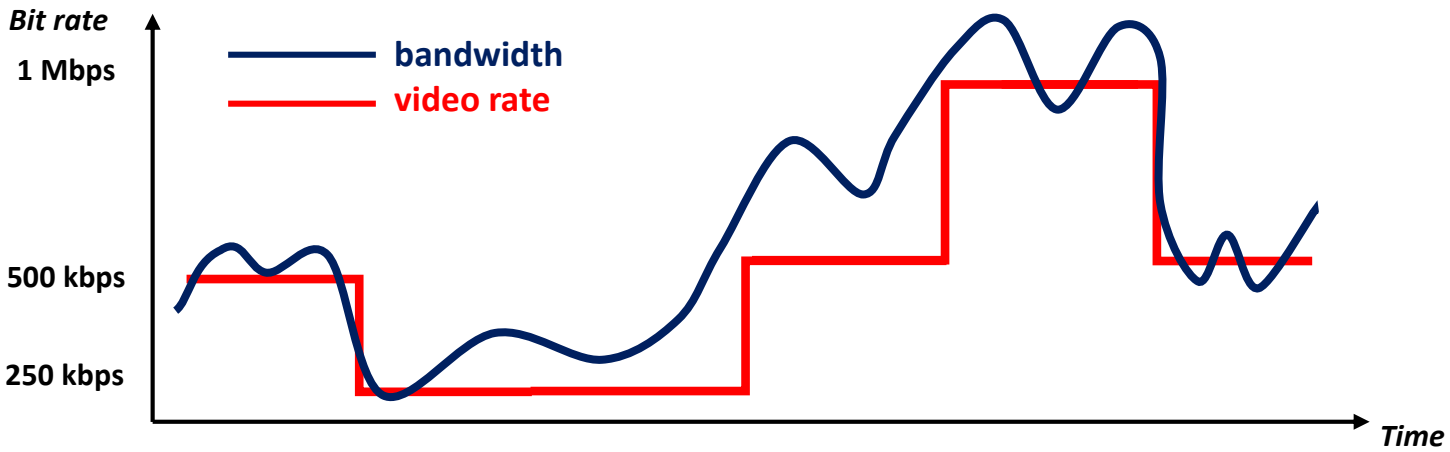
Streaming: définition

- Streaming: The content is sent from the source to the destination, which starts using it before its complete download.
- Terminology:
  - *video rate*: rate (in bps) at which the video has been encoded
  - *bandwidth*: rate available on the path from src to dst

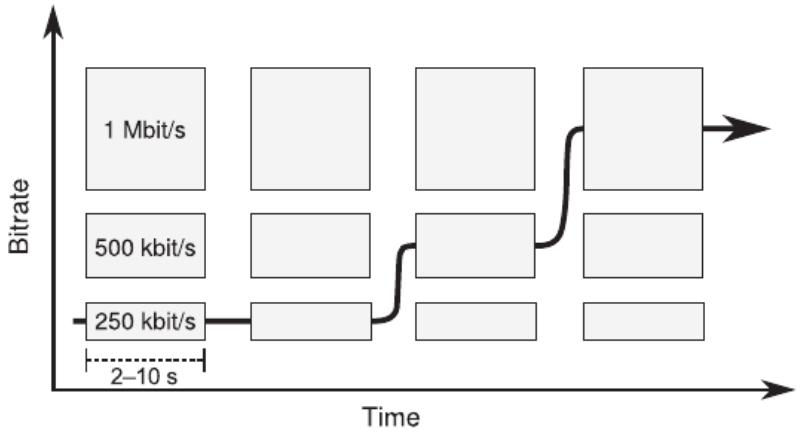


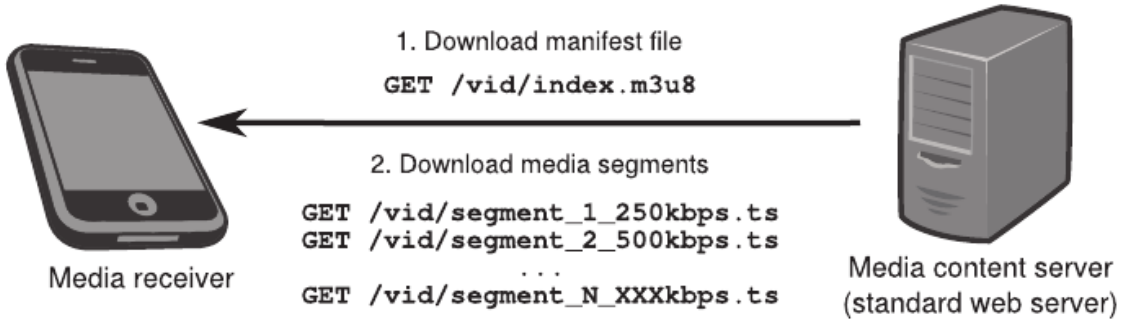
Streaming adaptatif: principe





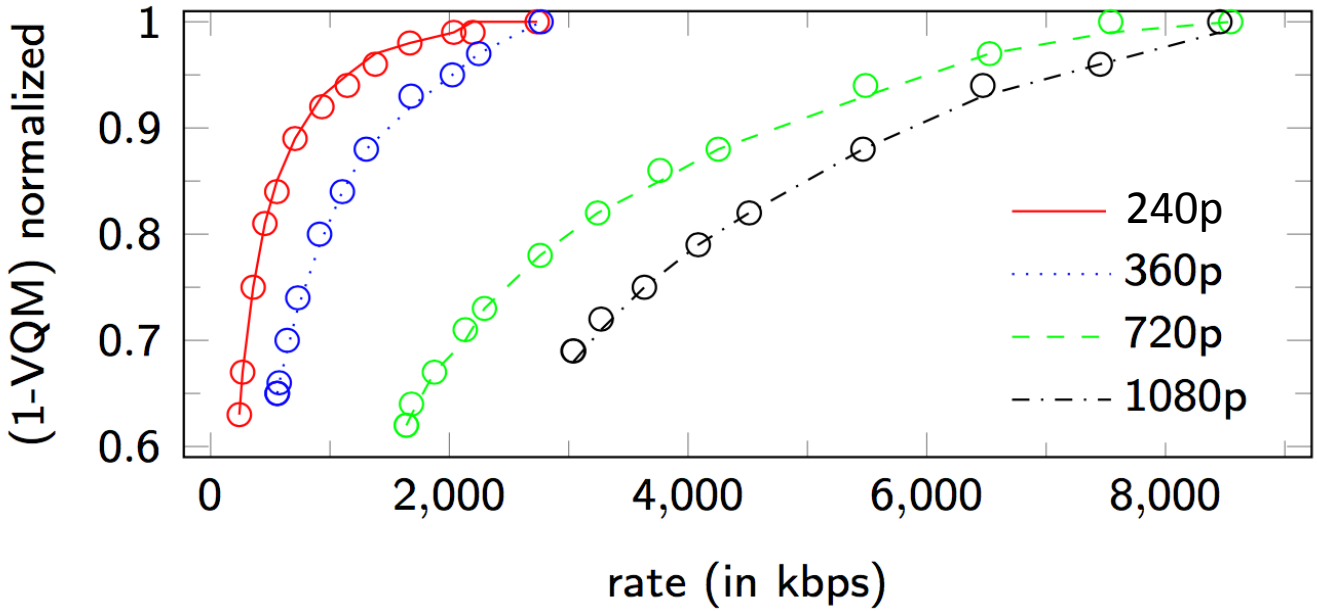
Streaming adaptatif: principe





La qualité perçue n'est pas le débit du réseau

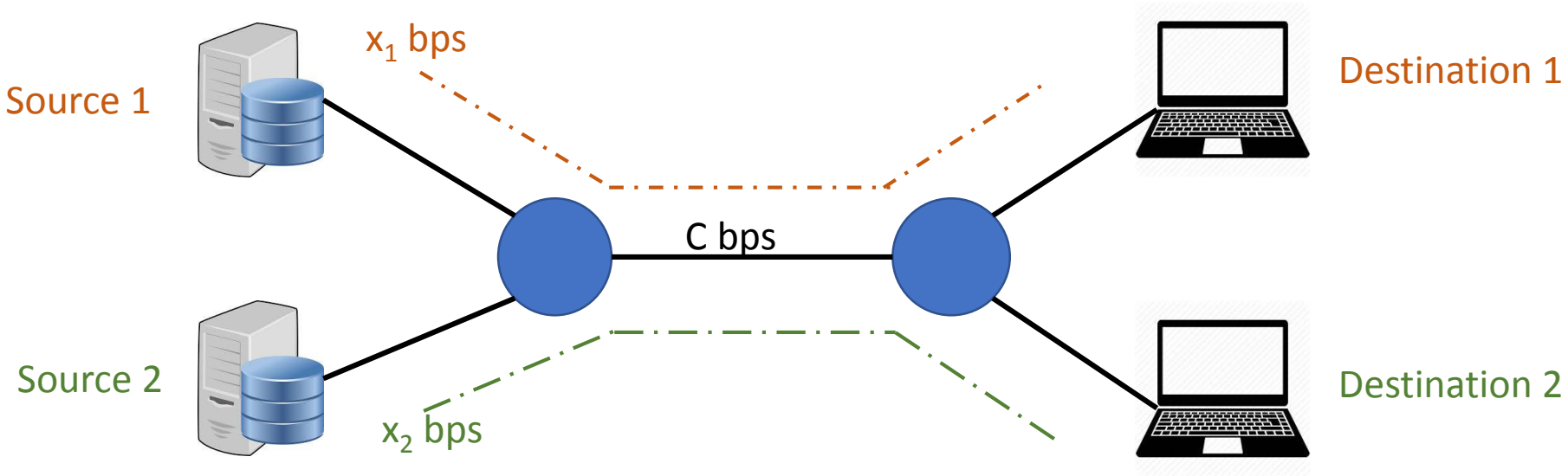
- On peut relier le débit du réseau à la qualité visuelle perçue pour chaque résolution :



Plan

- Articulation des enseignements technologiques au sein du monde numérique
  - Cas d'usage : Visionner une vidéo YouTube sur depuis un réseau mobile
- Streaming vidéo sur HTTP : principe
- Des maths pour le réseau :
  - Contrôle de débits de sessions concurrentes
  - Les caches dans le réseau

Comment partager la capacité ? Le contrôle de débit



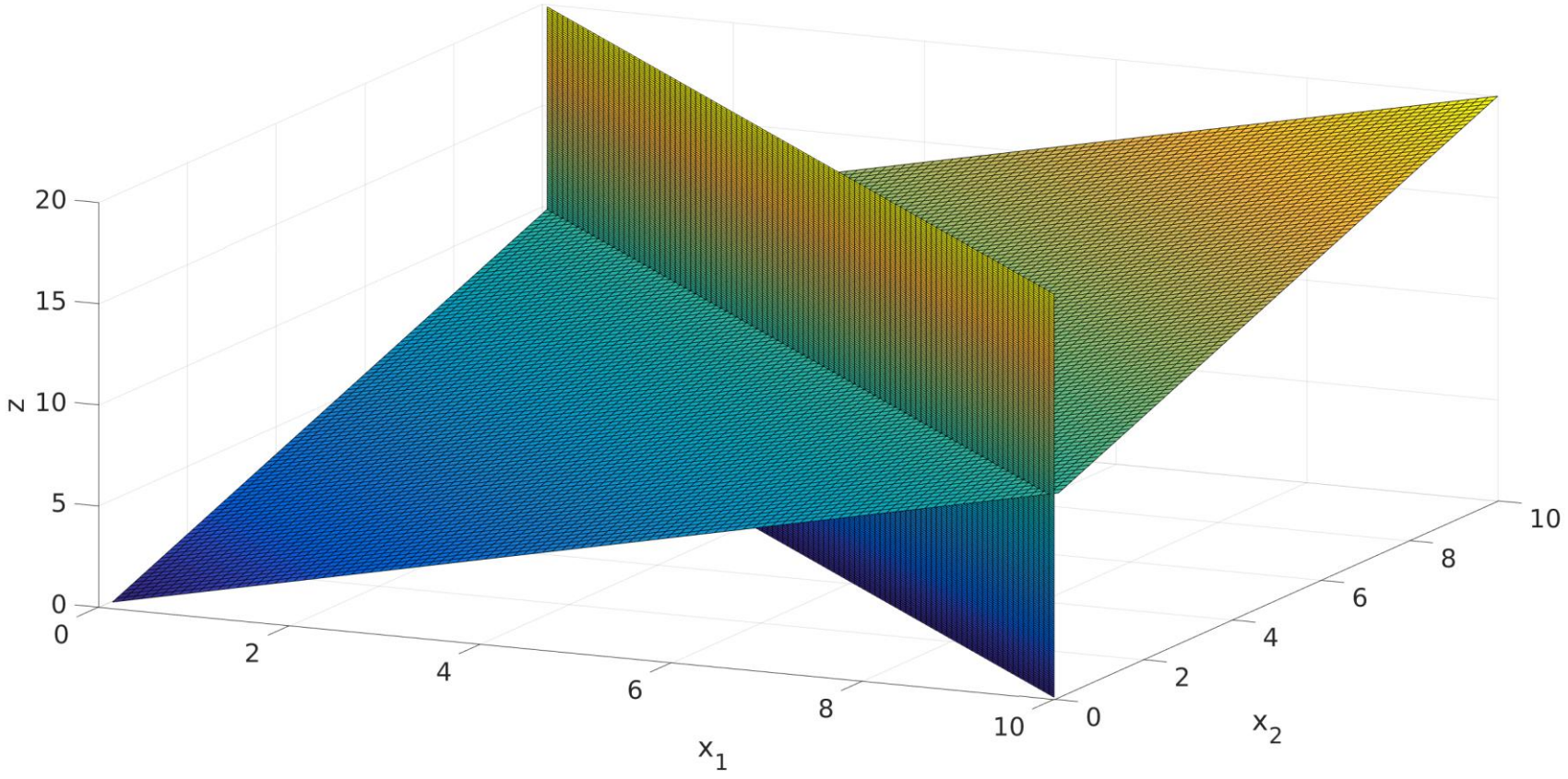
- Trouver x_1 et x_2 solutions au problème

$$\max_{x_1,x_2} x_1+x_2$$

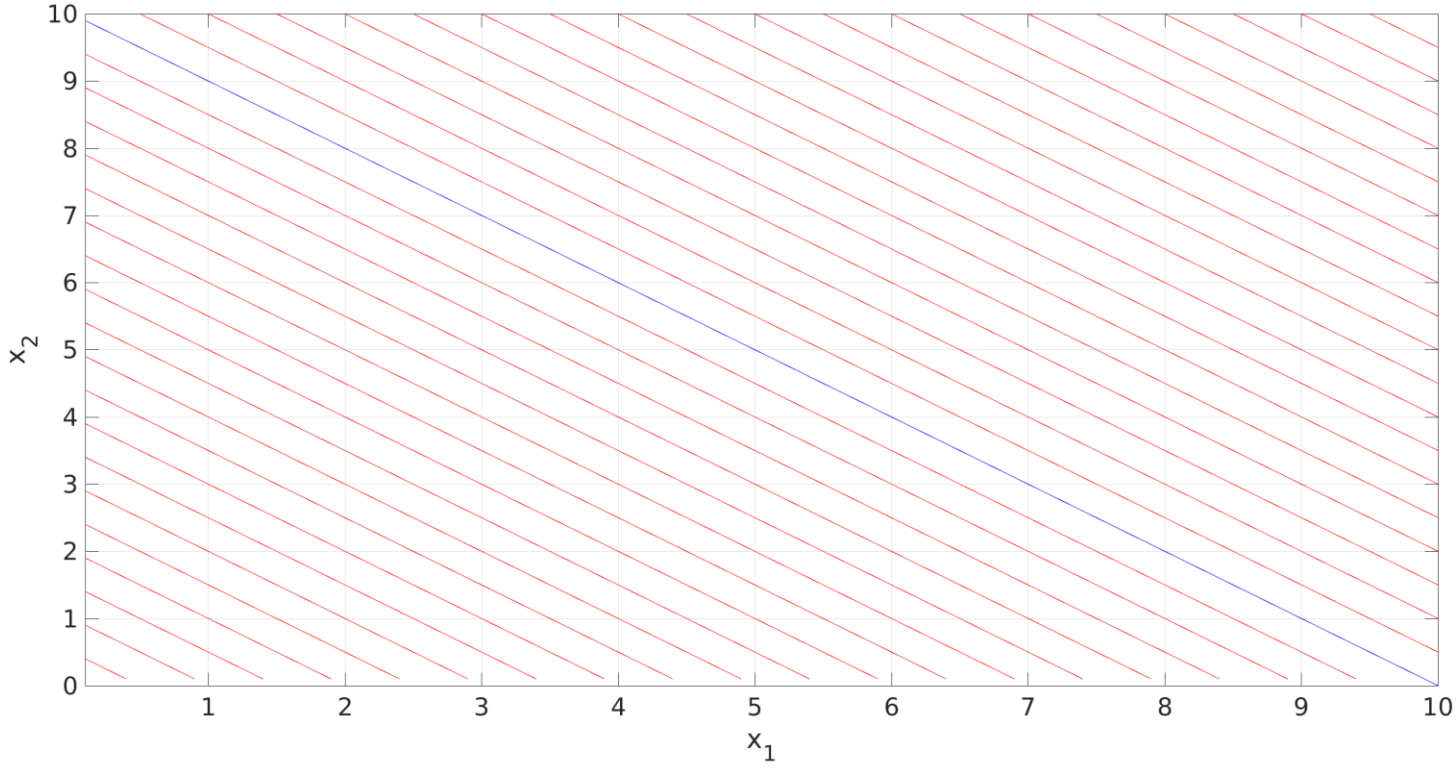
Tel que

$$x_1+x_2 = C$$

Comment partager la capacité ? Le contrôle de débit

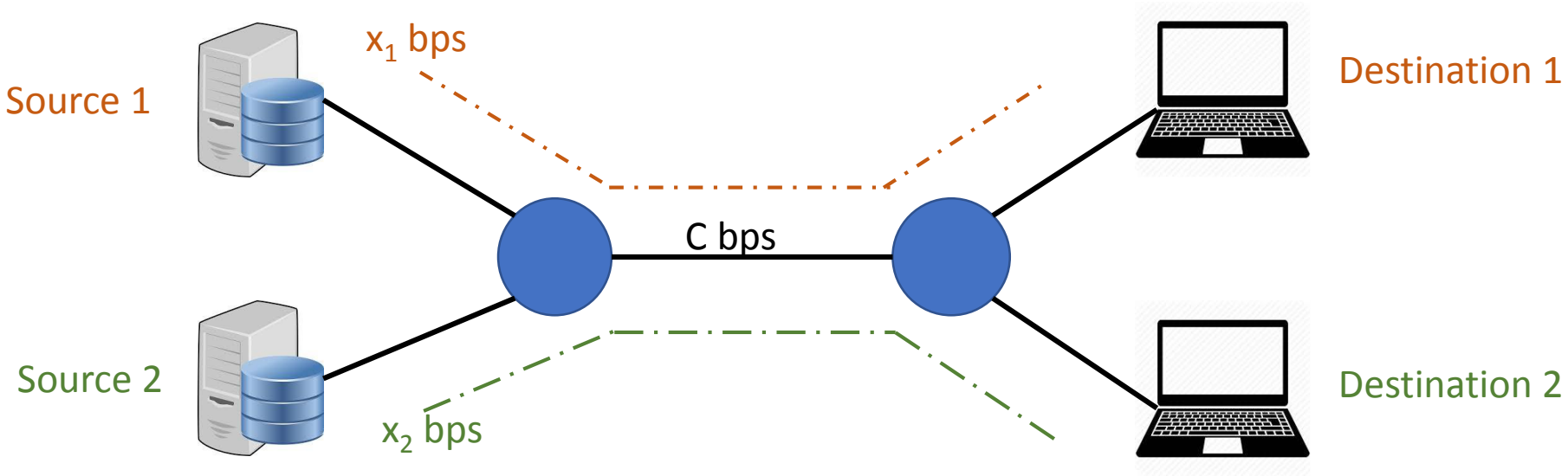


Comment partager la capacité ? Le contrôle de débit



-> pas de contrainte d'équité entre les sessions

Comment partager la capacité ? Le contrôle de débit



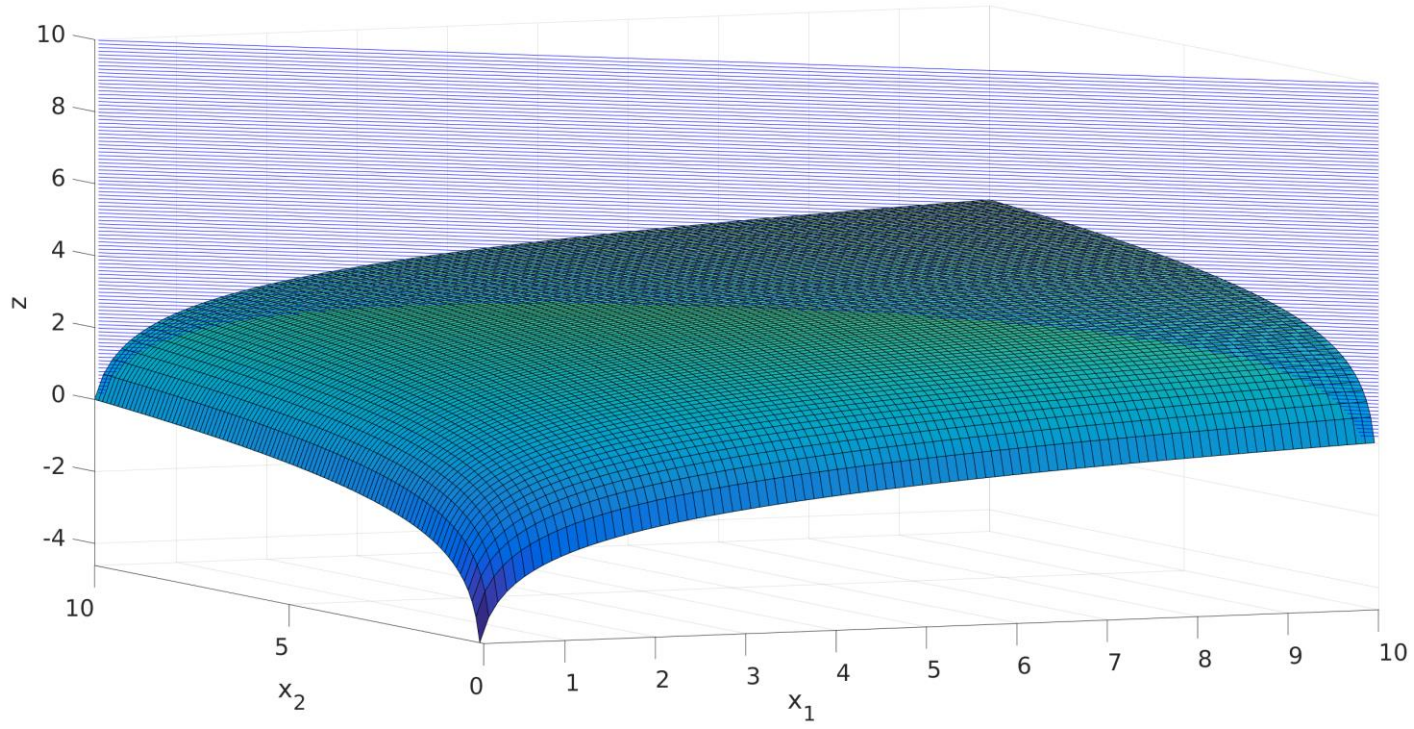
- Equité:

$$\max_{x_1,x_2} \log(x_1)+\log(x_2)$$

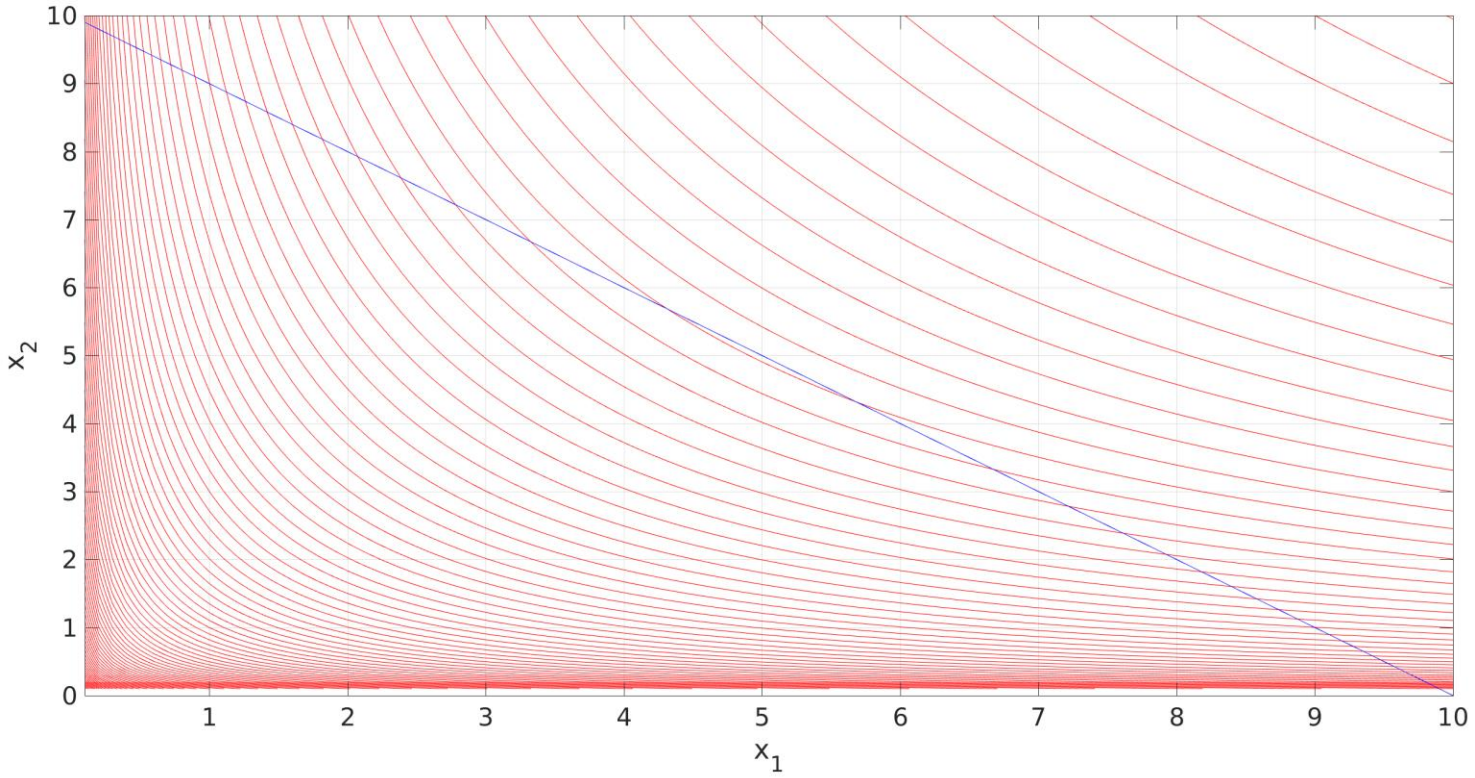
Tel que

$$x_1+x_2 = C$$

Comment partager la capacité ? Le contrôle de débit



Comment partager la capacité ? Le contrôle de débit

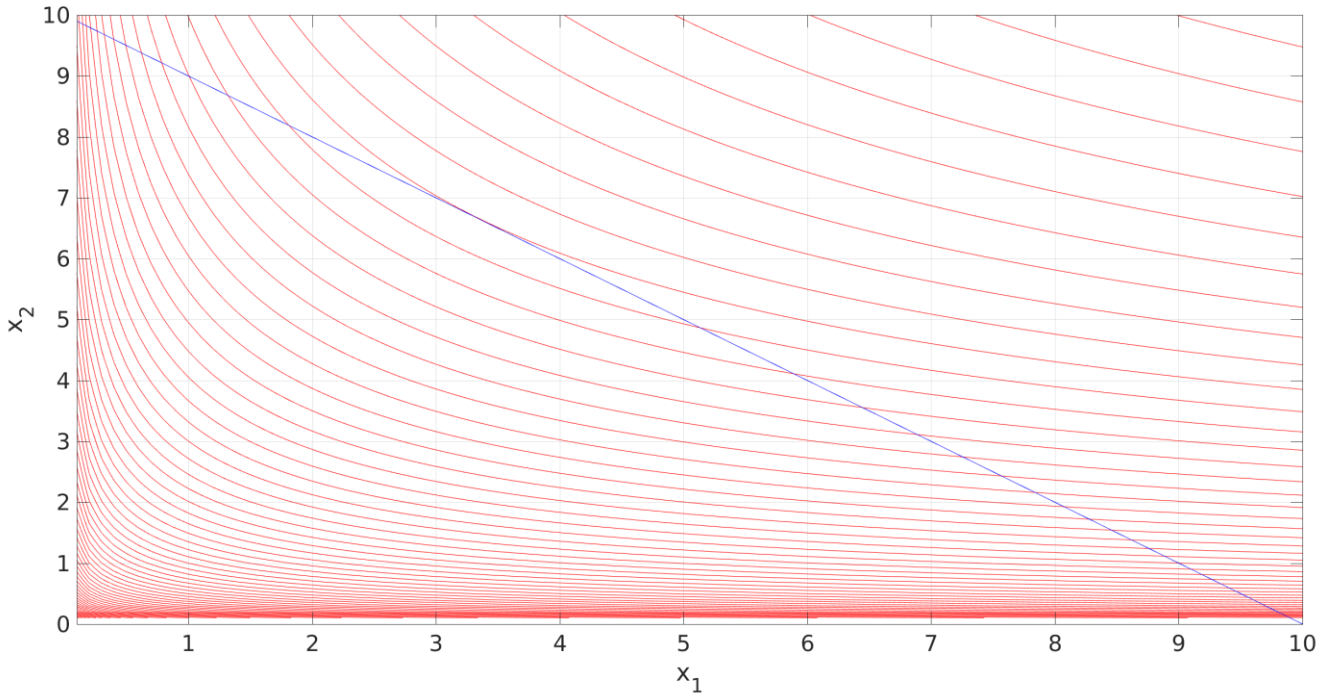


Comment partager la capacité ? Le contrôle de débit

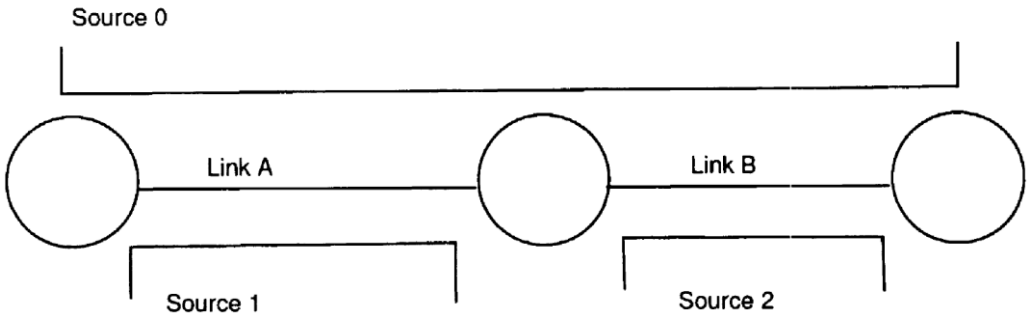
- Equité proportionnelle:

$$\max 0.5*\log(x_1)+\log(x_2)$$

Tel que $x_1+x_2 = C$



Cas général: TCP !



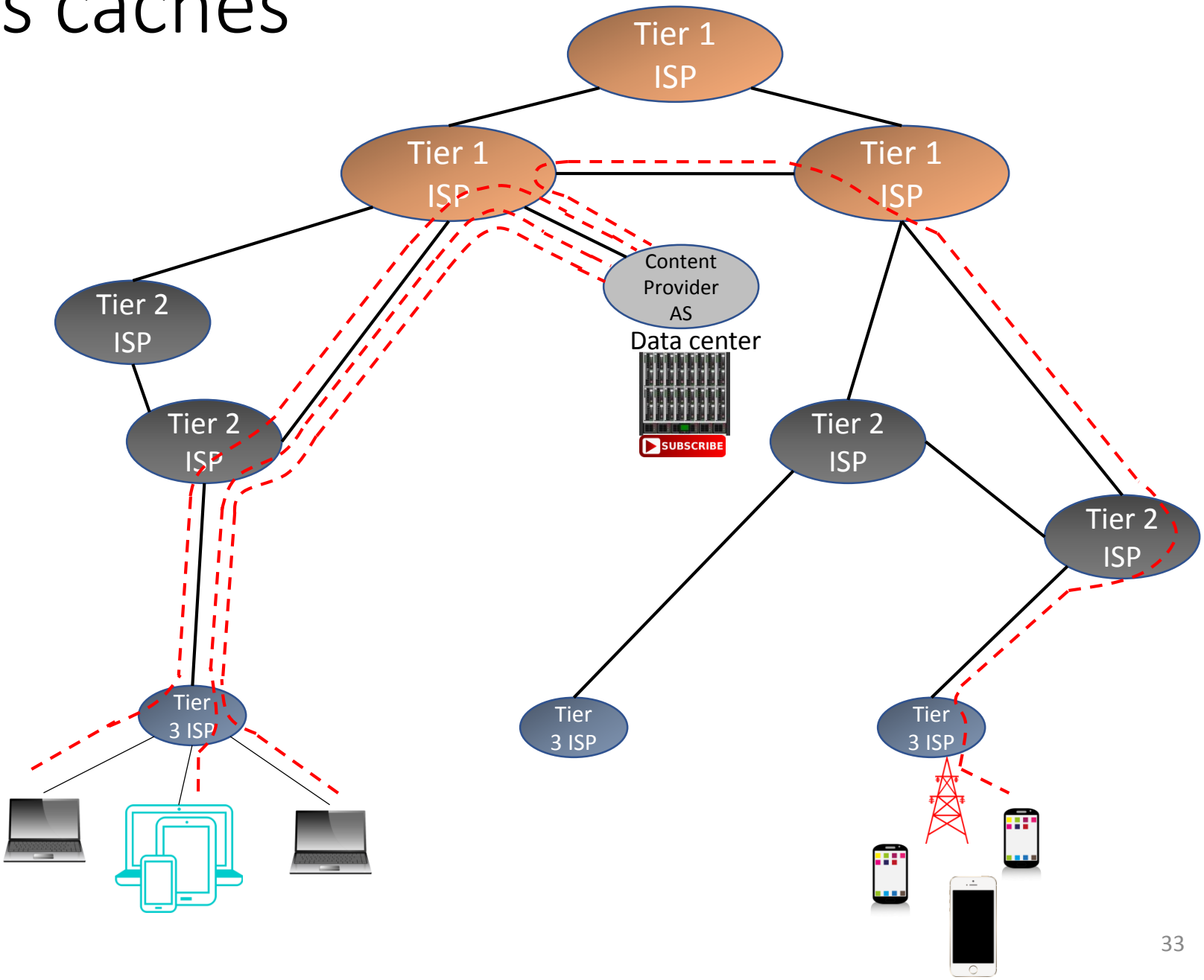
$$\max_{\{x_r\} \in \mathcal{S}} \sum_r U_r(x_r)$$

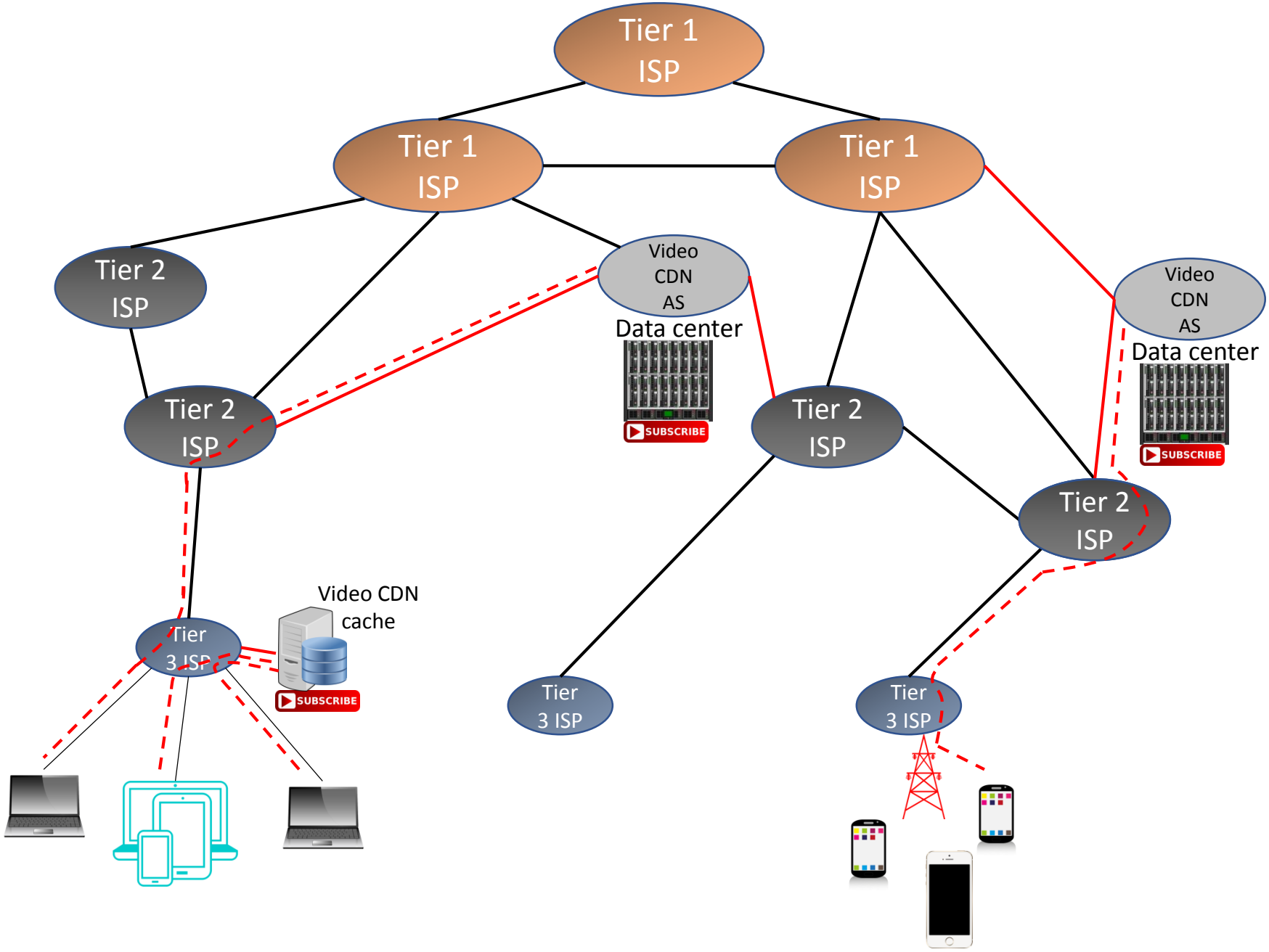
Tel que :

$$\sum_{r:l \in r} x_r \leq c_l, \qquad l \in \mathcal{L},$$

$$x_r \geq 0, \qquad r \in \mathcal{S},$$

Les caches


Tier 1
ISP
Tier 1
ISP
Tier 1
ISP
Content
Provider
AS
Data center
Tier 2
ISP
Tier 2
ISP
Tier 2
ISP
Tier 2
ISP
Tier
3 ISP
Tier
3 ISP
Tier
3 ISP


Tier 1 ISP
Tier 1 ISP
Tier 1 ISP
Tier 2 ISP
Video CDN AS
Data center
Video CDN AS
Data center
Tier 2 ISP
Tier 2 ISP
Tier 2 ISP
Video CDN cache
Tier 3 ISP
Tier 3 ISP
Tier 3 ISP
SUBSCRIBE
SUBSCRIBE
SUBSCRIBE

Quoi garder en cache ?

- In a TTL cache, each file *i* is associated with a timer t_i
- Hit probability of file *i*: $h_i = 1 - e^{-\lambda_i t_i}$
- Design a cache management such that:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N} U_i(h_i) \\ \text{such that} \quad & \sum_{i=1}^{N} h_i = B \\ & 0 \leq h_i \leq 1, \quad i = 1, 2, \ldots, N. \end{aligned}$$

- Solution:

$$h_i = U_i'^{-1}(\alpha) \qquad \sum_i U_i'^{-1}(\alpha) = B$$