

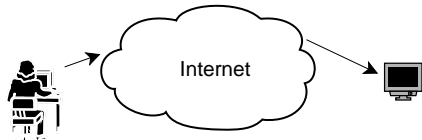
## Amphi 8 : Internet

19 juin

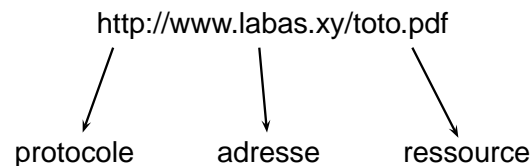
- I. Présentation générale.
- II. Un peu d'histoire.
- III. La communication point à point.
- IV. Quelques outils de communication.
- V. Pale machine.

## I. Internet: présentation générale

**Internet, comment ça marche?** liaison point à point (*end to end*), dirigée par les applications (*application driven*).



**Ex:** quand on tape une URL (*Uniform Resource Locator*) dans son navigateur:



Le navigateur utilise le protocole `http` pour récupérer la ressource `toto.pdf` et la traiter. Typiquement, un fichier pdf est ouvert à l'aide d'un programme de visualisation.

## 0. Où en sommes-nous?

- Amphi 1: introduction.
- Amphi 2: programmer en Java.
- Amphi 3: fonctions/fonctions récursives.
- Amphi 4: tableaux/String.
- Amphi 5: classes.
- Amphi 6: tables.
- Amphi 7: algorithmes et complexité.
- Amphi 8: Internet.
- Amphi 9: listes.
- Amphi 10: système/sécurité.

## IP (Internet Protocol)

Chaque ordinateur a un **numéro IP**.

**Ex.** `sil` a pour adresse IP `129.104.247.3` (quatre entiers entre 0 et 255); en IPv6, adresses sur 128 bits.

**Dans l'ancien temps:** trois classes d'adresses **classe A** =  $2^{24} = 16$  millions de machines; **classe B** =  $2^{16}$  machines; **classe C** =  $2^8$ .

**Désormais:** `|prefix|host_path|` pour récupérer plus d'adresses.

**Nom logique:** `nom.domaine.{pays,type}`. Les noms sont gérés par INTERNIC (AFNIC pour la France).

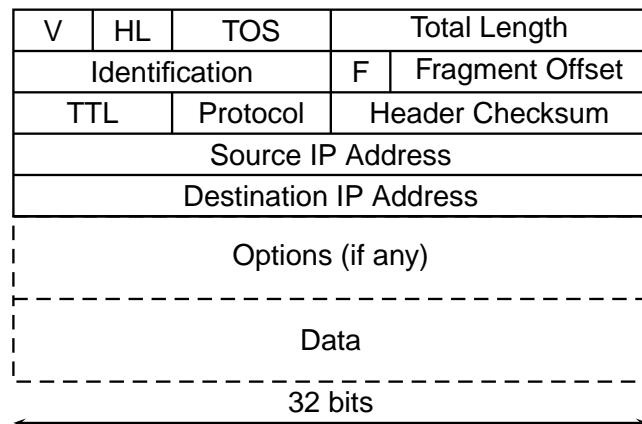
# Le DNS (Domain Name Server)

Service qui répond aux requêtes du type : **quel est le numéro IP de cette machine ?** (`nslookup`), quel le nom de la machine de numéro IP ; etc.

Il connaît toutes les machines de son site ; s'il ne sait pas, il demande au-dessus ; **cache** les requêtes (i.e., garde en mémoire pendant un certain temps les réponses aux questions les plus fréquentes);

# IP

Unité de données échangées sur le réseau.  
Principe de **best effort** (modèle économique!) : pas de garantie de délai, réussite de transmission. ⇒ rajouter une couche, par exemple TCP.



# Le modèle des 7 couches

Modèle de communication défini dans OSI-RM (*Reference Model for Open Systems Interconnection*, ISO/IEC JTC1 1978, X.200) :

Couche 7	application	application ( <b>mozilla</b> , etc.)
Couche 6	présentation	
Couche 5	session	
Couche 4	transport	transport (TCP, UDP)
Couche 3	réseau	réseau (IP)
Couche 2	data link	couche physique (MAC)
Couche 1	couche physique	
	OSI-RM	INTERNET

**V:** Version (4 ou 6) ;

**HL:** (Header Length) nombre de bytes du header (typ. 20) ;

**TOS:** (Type Of Service) priorité du paquet (obsolète) ;

**Total Length:** longueur totale du paquet, header compris ;

**Identification, F(lags), Fragment Offset:** fragmentation ;

**TTL:** (Time To Live) temps de vie d'un paquet (décrémenté de 1 à chaque traitement, perdu si 0) ;

**Protocol:** 1 pour ICMP, 6 for TCP, 17 for UDP ;

**Header Checksum:** correction d'erreurs ;

**Options:** pour le routage, etc.

*Transmission Control Protocol*: prend un flot de données, le coupe en paquets IP, et les transmet sur le réseau d'une manière fiable (source et destination s'échangent des confirmations).

Le principe **end to end** veut dire que seules sont concernées la source et la destination, pas les nœuds intermédiaires.

Il existe d'autres modes de transport: RTP (*Real Time Protocol*), UDP (*User Datagram Protocol*).

## traceroute

```
kelen% traceroute concorde.inria.fr
traceroute to concorde.inria.fr (192.93.2.39), 30 hops
 1 129.104.247.13 (129.104.247.13) 0.487 ms 0.319 ms
 2 129.104.63.1 (129.104.63.1) 0.590 ms 0.357 ms 0.
 3 129.104.63.5 (129.104.63.5) 1.185 ms 1.201 ms 1.
 4 jussieu.cssi.renater.fr (193.51.12.81) 3.415 ms 2
 5 nio-nl.cssi.renater.fr (194.214.109.5) 2.534 ms 2
 6 rerif-nio-nl.cssi.renater.fr (193.51.206.22) 4.715
 7 stamand1.rerif.ft.net (193.48.53.106) 3.244 ms 4.
 8 inria-rocquencourt.rerif.ft.net (193.48.53.134) 8.
 9 rocq-rerif.inria.fr (192.93.122.2) 6.568 ms 8.068
10 192.93.1.100 (192.93.1.100) 7.556 ms 7.753 ms 9.
11 concorde.inria.fr (192.93.2.39) 9.808 ms * *
```

Chacun sa route, chacun son chemin,  
passe le message à ton voisin.

- Pas de connection directe; information **décentralisée**.
- Pas de chemin inscrit dans le marbre.

### Propriétés:

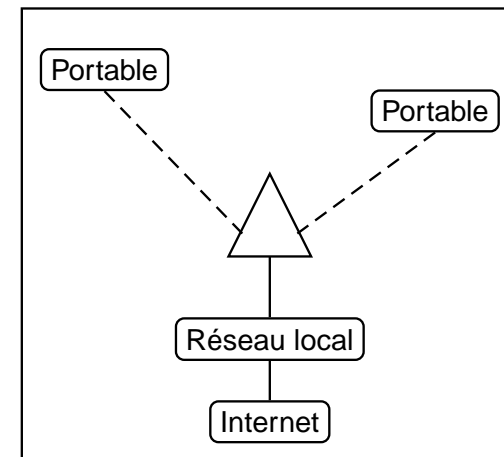
- Chaque nœud du réseau a un numéro.
- Les messages circulent dans le réseau, sur chacun est indiqué le numéro du destinataire.
- Chaque nœud a un certain nombre de voisins.
- Une table en chaque nœud  $i$  indique le voisin  $j$  à qui il faut transmettre un message destiné à  $k$ .
- Algorithme utilisé: variante de Dijkstra, cf. INF-431.

## IP s'adapte à tous les contextes!

WiFi (Wireless Fidelity), **Internet ambient**

INTERNET sans fil, mais à l'aide de la radio.

### Principe :



## II. Un peu d'histoire

- **1957** : le *Department of Defense* (DoD) crée l'agence **ARPA** (*Advanced Research Projects Agency*);
- **1962** : Paul Baran (*Rand Corporation*) décrit le principe d'un réseau **décentralisé, redondant, à commutations de paquets** ( $\neq$  connexions directes);
- **1969** : **première mise en œuvre** à l'UCLA entre quatre nœuds, puis à Stanford, à l'UCSB, etc.;
- **1972** : création de l'**InterNetwork Working Group** (INWG).
- **1972–1974** : **premiers protocoles** (telnet, ftp, TCP, mail);

## Le web

**Début des années 1960** : D. Englebart met au point le **oNLine System** (NLS) – édition hypertexte – et invente la souris.

**1965** : T. Nelson invente le terme **Hypertext** lors de la conférence nationale de l'ACM à New York.

**1980** : T. Berners-Lee (CERN) écrit **Enquire-Within-Upon-Everything**.

**Mars 1989** : TBL invente le **WEB**; 1ère page en novembre 1990 : <http://nxoc01.cern.ch/hypertext/WWW/TheProject.html>

**1993** : Le CERN décide que le WEB est utilisable **librement**.

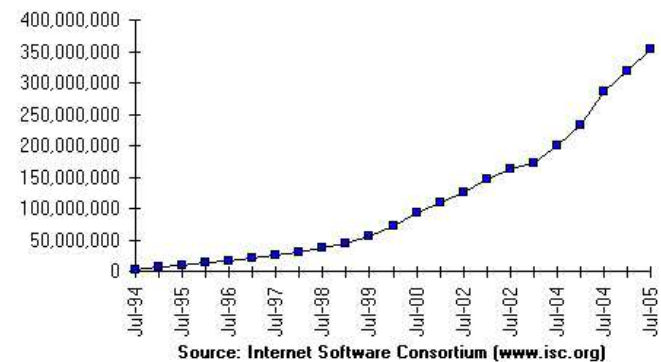
**1994** : création du Web Consortium (W3C).

- **1970ff** : réseau cyclade (CNET, INRIA ; **Louis Pouzin (X50)** – inventeur des datagrammes, repris dans ARPANET, **Hubert Zimmermann (X61)**), . . . , TRANSPAC.
- **1986** : la NSF encourage le développement de **NFSNET** qui sert d'épine dorsale au réseau ;
- **à partir de 1989** : **INTERNET s'impose** peu à peu dans le monde entier, en incorporant de multiples architectures réseau (interopérabilité) ;
- **fin 1991** : **5000 sous-réseaux** dans **40 pays**, reliant **700,000 ordinateurs** et **4,000,000 d'utilisateurs** ;
- **1992** : création de l'**Internet Society** (ISOC).

## Cartes du réseau

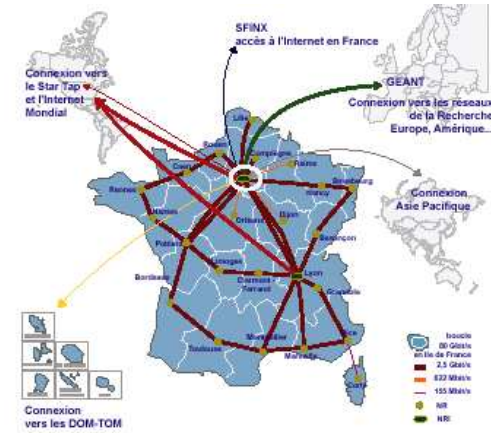
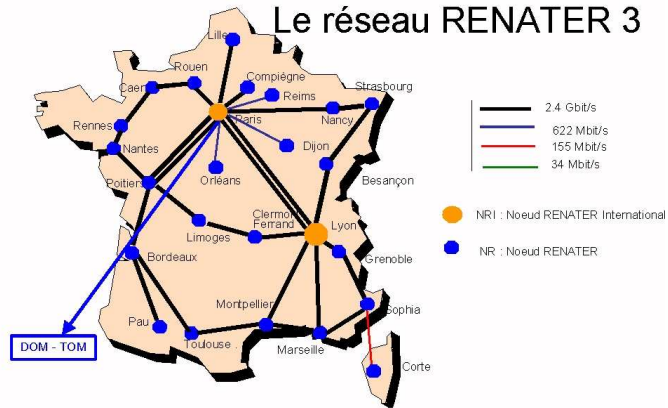
Nombre de machines routables :

Internet Domain Survey Host Count

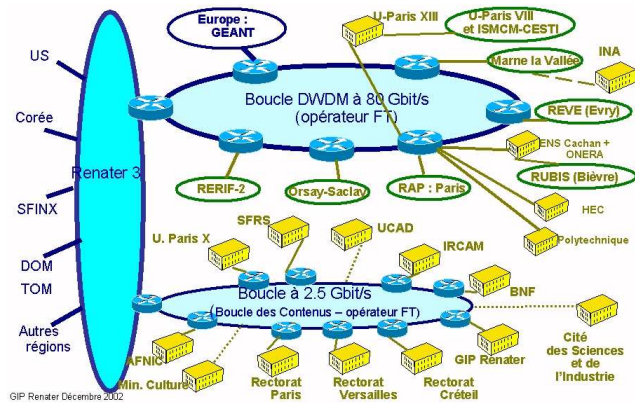


Domaine **.fr** : au 23 mars 2006, 456918 sous-domaines répertoriés par l'AFNIC (resp. 346126 au 25 mars 2005).

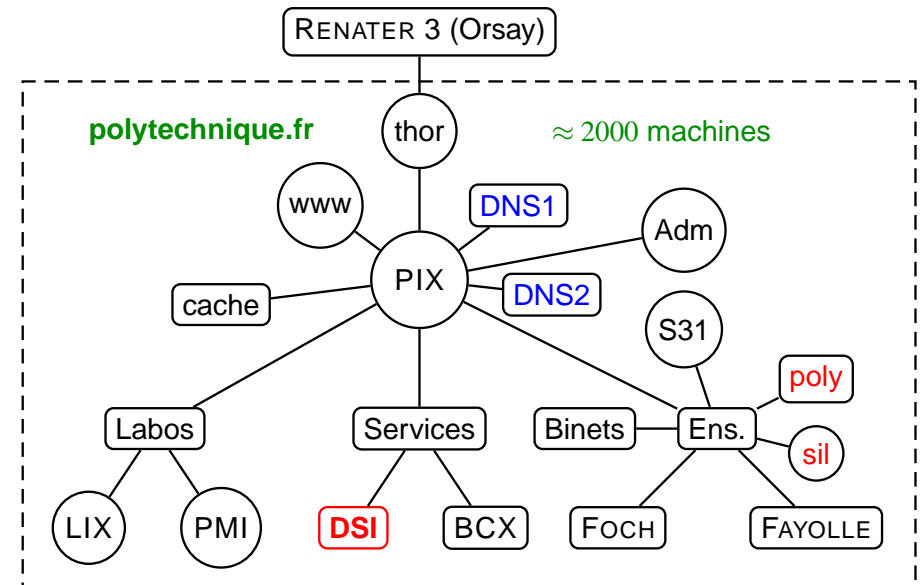
# Le réseau RENATER



## Connexion à Jussieu



## Le réseau de l'École



# INTERNET : un monde sans lois ?

We reject kings, presidents, and voting; we believe in rough consensus and running code (Dave Clark, 1992).

L'Internet Society (ISOC, [www.isoc.org](http://www.isoc.org)) :

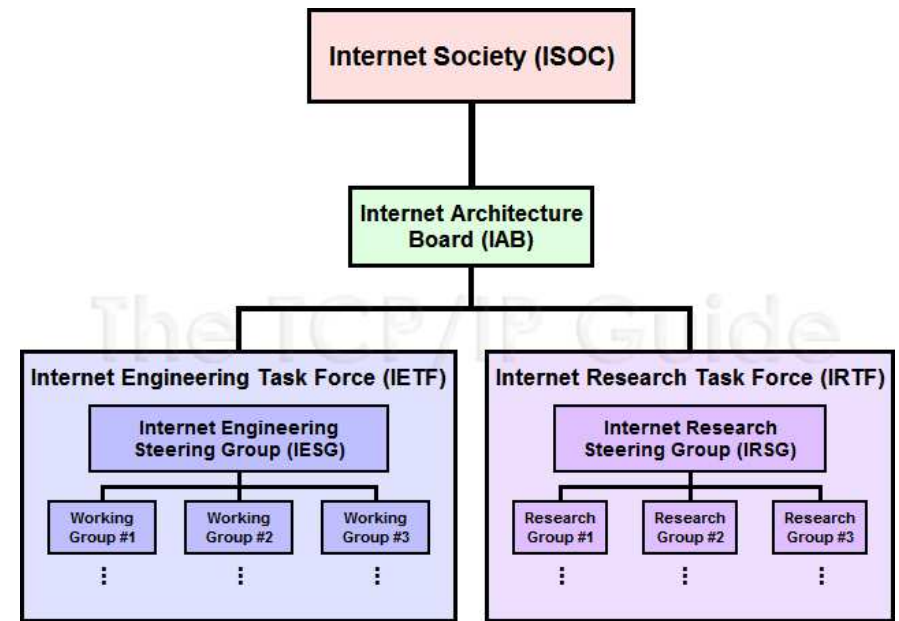
- créée en 1992 ;
- "organisation globale et internationale destinée à promouvoir l'interconnexion ouverte des systèmes et l'INTERNET" ;
- Conseil de Gestion (**Board of Trustees**) : élu par les membres de l'association dirige des comités dont le plus important est l'IAB (**Internet Architecture Board**), dont le but est de gérer l'évolution des protocoles TCP/IP.

## III. La communication point à point

Un numéro IP sert à identifier une machine sur le réseau. Le numéro de port sert à identifier un processus spécifique sur une machine donnée. Donc un couple (IP, port) sert à identifier un processus donné sur une machine donnée sur Internet. Se fait par le canal physique entre deux interfaces (cartes) réseaux. Le système avertit les programmes qui **écoutent sur ce port**.

Quelques exemples de ports conventionnels:

Protocole	nom complet	Port(s)
FTP	File Transfer Protocol	21/20
SMTP	Simple Mail Transfer Protocol	25
DNS	Domain Name System	53
HTTP	Hypertext Transfer Protocol	80
NNTP	Network News Transfer Protocol	119



## Exemple: le routage du mail

From: madeleine@bruxelles  
To: jacques@amsterdam  
Subject: Re: frites chez Eugène ?

SMTP

CONNECT <129.104.1.1.25,133.12.104.12.25>

TCP

133.12.104.12, port 25



## Quelques problèmes

- **IP et la sécurité:** dans IPv4, la partie donnée n'est pas protégée *a priori*. Tout se passe comme si on envoyait une **carte postale** avec des données visibles par tout le monde (sniffage des paquets).

**Solution :** norme **IPSec**, qui décrit comment établir un **tunnel sécurisé** entre deux routeurs, grâce auquel les paquets sont chiffrés. Implantée dans IPv6.

- **Inertie:** il est très difficile de changer quelque chose alors que tout le système marche et ne peut s'arrêter. Le consensus sert plus à bloquer qu'à autre chose.

## À propos du spam



**Thm.** Détecter un spam automatiquement est indécidable.

**Outils:** SpamOracle, spamassassin, etc.

**Comment opèrent-ils?** Repérage de mauvais headers (règles non respectées, nom d'envoyeur bizarre); listes noires; difficile de repérer des numéros IP bidons (un jour: DNS sécurisés); contenu étrange avec mots clefs publicitaires.

## IV. Quelques outils de communication

**Problème central:** où se trouve l'information ? Comment la gérer ?  
Comment la partager ?

- Google; wikipedia.
- Forums; news; blogs; flux RSS.
- Courrier électronique.

## La réponse automatique de Dan Bernstein

```
From: "The qsecretary program" <djb-notbulkmail-46c063d  
To: morain@lix.polytechnique.fr  
Subject: qsecretary notice
```

```
Hi. This is D. J. Bernstein's automated  
mail-handling program. I've received a  
message from you. [...]
```

```
Professor Bernstein receives many interesting  
messages. Unfortunately, he also receives a  
torrent of unsolicited commercial mail, [...]
```

```
Professor Bernstein has asked me to reject all  
bulk mail messages. But I'm a rather primitive  
computer program, and I'm not sure whether your  
message is bulk mail.
```

## V. Préparons-nous à la pale machine

**But:** écrire un programme qui marche et résolvant le problème posé.

**Règles du jeu :**

- le programme à écrire tient en quelques dizaines de lignes;
- énoncé simple, entrées et sorties complètement spécifiées; les fichiers fournis en entrée respectent les règles et ne sont pas erronés;
- l'énoncé est accompagné de fichiers d'entrée et des fichiers de sortie correspondant.
- le sujet est découpé en trois exercices qui s'enchaînent, et qui doivent être tous faits dans l'ordre.

If you reply to this notice, you are  
(1) acknowledging that Professor Bernstein does not want to receive bulk mail;  
(2) confirming that your message is not part of a bulk mailing;

(3) agreeing to pay Professor Bernstein \$250 if your message is part of a bulk mailing.

[...]

Sincerely,  
The qsecretary program

P.S. Professor Bernstein has asked me to convey his own apologies to you if you're someone he knows. I'm sure he'll tell me to accept subsequent messages from you without confirmation.

[...]

**Vérification automatique:** réponse donnée dès que possible (sauf < 15 minutes avant la fin). Plusieurs soumissions possibles.

**Barème:**

**Déf. fait** = le programme compile, et produit les résultats annoncés sur les fichiers d'entrée.

**Mieux vaut un programme qui marche que trois qui ne compilent pas!**

- exercices 1+2+3 faits: 20/20.
- exercices 1+2 faits: 14/20 (16/20 si faits en 1h30).
- exercice 1 fait: 08/20.
- aucun exercice, mais fichier déposé: correction manuelle, 1 point en moins par intervention à partir de 08/20. Plancher à 04/20.
- aucun fichier déposé: 0/20.

## Notation

**Notation du module:** HC; note pale machine (/20) + note pale papier (/20) + note participation (/2). Avec la formule:

$$\frac{19}{20} \frac{M+P}{2} + p - 1,$$

suivi d'un transcodage vers A, B, C, D, E (F).

**Calendrier :** pale cet après-midi.

Bien sûr, ... la présence aux examens est obligatoire.



# Les consignes pour la pale

N'arrivez pas au dernier moment.

## Consignes générales :

- lisez bien le sujet, faites un exemple à la main ;
- ne cherchez pas midi à quatorze heures ;
- évitez de déposer plusieurs fois, svp.

Essayez avec des fichiers d'entrées à vous.

**Important:** vous allez être livrés à vous-mêmes, les enseignants ne vous expliquent que les messages d'erreur.

**Que faire si mon programme plante ?** affichez des messages et des valeurs des variables à des endroits bien choisis du programme pour voir où et pourquoi.

# Exemple: les décimales de $\pi$

Il est parfois utile de connaître les premières décimales du nombre  $\pi = 3.14159\dots$ . Un poème bien connu remplit cette tâche. Il commence comme suit:

Que j aime a faire apprendre ce nombre utile aux sages !  
Immortel Archimede, artiste ingenieur,

doit être transformé en

31415926535  
8979

Il n'y a pas de mots de longueur > 10, mais 10 == 0.

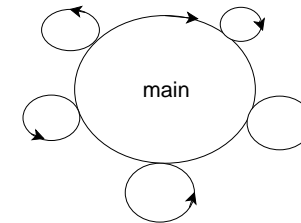
**Principe fondamental:** on fait pousser le programme!

**Première phase:** vérifions qu'on sait lire. On écrit le programme qui lit et affiche ce qu'il a lu.

```
public class Pi{
    public static void main(String[] args){
        int nl = TC.lireInt();

        for(int i = 0; i < nl; i++){
            String ligne = TC.lireLigne();
            System.out.println(ligne);
        }
    }
}
```

**Deuxième phase:** le squelette.



```
public class Pi{
    static void traiter(String mot){
        System.out.println(mot);
    }
    public static void main(String[] args){
        int nl = TC.lireInt();
        for(int i = 0; i < nl; i++){
            String ligne = TC.lireLigne();
            String[] mots = TC.motsDeChaine(ligne);
            for(int j = 0; j < mots.length; j++){
                traiter(mots[j]);
            }
            System.out.println();
        }
    }
}
```

Et finalement: le vrai programme.

```
public class Pi{
    static void traiter(String mot){
        int nc = 0;
        for(int i = 0; i < mot.length(); i++)
            if(Character.isLetter(mot.charAt(i)))
                nc++;
        if(nc != 0){
            if(nc == 10)
                nc = 0;
            System.out.print(nc);
        }
    }
    public static void main(String[] args){
        int nl = TC.lireInt();
        for(int i = 0; i < nl; i++){
            String ligne = TC.lireLigne();
            String[] mots = TC.motsDeChaine(ligne);
            for(int j = 0; j < mots.length; j++)
                traiter(mots[j]);
            System.out.println();
        }
    }
}
```

## Derniers mots

Prochains rendez-vous : **PALE MACHINE!!!!**

On commence à l'heure pile, arrivez-donc un peu avant...

Groupes	TD 8
1-6	13h30-15h30
7-12	15h45-17h45

Prochain amphi: lundi 26 juin à 10h30.

**11 juillet:** pale papier.