Is the Internet doomed to collapse?

Mathieu Feuillet
Based on a joint work with T. Bonald and A. Proutière

Rocquencourt Junior seminar

15/05/2012
- **Topic:** Designing and analyzing network algorithms
- **Tools:** Mathematical modeling and probabilistic methods
Why mathematical modeling?

- Networks can be huge
Why mathematical modeling?

- Networks can be **huge**

- **Experiments:** < 100 nodes
Why mathematical modeling?

- Networks can be huge
- Experiments: < 100 nodes
- Simulations: < 1000 nodes
Why mathematical modeling?

- Networks can be **huge**
- **Experiments:** < 100 nodes
- **Simulations:** < 1000 nodes
- **Both:** finite number of parameters
Why mathematical modeling?

- Networks can be huge
- Experiments: < 100 nodes
- Simulations: < 1000 nodes
- Both: finite number of parameters

Mathematical analysis allows to tackle the huge size and the different parameters... with some simplifications.
Why probabilistic modeling?

- Traffic is inherently random.
- User behaviors
- Non-deterministic failures (WiFi, ...)
- Stochastic algorithms
Why probabilistic modeling?

Traffic is inherently random.
Why probabilistic modeling?

Traffic is inherently random.

- Users behaviors
Why probabilistic modeling?

Traffic is inherently random.

- Users behaviors
- Non-deterministic failures (WiFi...
Why probabilistic modeling?

Traffic is inherently random.

- Users behaviors
- Non-deterministic failures (WiFi...) 
- Stochastic algorithms
Why probabilistic modeling?

Traffic is inherently random.

- Users behaviors
- Non-deterministic failures (WiFi...) 
- Stochastic algorithms 
- ...
What is the Internet?
Original goals

- Transmission of data from one computer to another
Original goals

- Transmission of data from one computer to another
- Large network
Original goals

- Transmission of data from one computer to another
- Large network
- Distributed System
Original goals

- Transmission of data from one computer to another
- Large network
- Distributed System
- Resist to the partial destruction
Packet switching

Data flow is cut into packets
Packet switching

Data flow is cut into packets
Packet switching

Data flow is cut into packets

Each packet is sent independently
Packet switching

Data flow is cut into packets

Each packet is sent independently
Packet switching

Data flow is cut into packets

Each packet is sent independently
Packet switching

Data flow is cut into packets

Each packet is sent independently
Packet switching

Data flow is cut into packets

Each packet is sent independently
Packet switching

Data flow is cut into packets

Each packet is sent independently
Packet switching

Data flow is cut into packets

Each packet is sent independently

Internet: routing and resource allocation
A router

Input rate > Output rate: congestion and packet drop
Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate:
congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
A router

Input

Output

Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
Input rate > Output rate:
congestion and packet drop
A router

Input rate > Output rate: congestion and packet drop
Packet switching and QoS

Drawbacks:

- No **reliability**: packets can be dropped on the route
Packet switching and QoS

Drawbacks:
- No reliability: packets can be dropped on the route
- Buffering/Queuing induces delays
Packet switching and QoS

Drawbacks:
- No reliability: packets can be dropped on the route
- Buffering/Queuing induces delays
- No bandwidth allocation algorithm
Packet switching and QoS

Drawbacks:
- No reliability: packets can be dropped on the route
- Buffering/Queuing induces delays
- No bandwidth allocation algorithm

<table>
<thead>
<tr>
<th>Application</th>
<th>Reliability</th>
<th>Delay</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>File transfer</td>
<td>++</td>
<td>- -</td>
<td>+ -</td>
</tr>
<tr>
<td>Web access</td>
<td>++</td>
<td>+ -</td>
<td>+ -</td>
</tr>
<tr>
<td>Video on Demand</td>
<td>- -</td>
<td>- -</td>
<td>+ +</td>
</tr>
<tr>
<td>Telephony</td>
<td>- -</td>
<td>+ +</td>
<td>- -</td>
</tr>
</tbody>
</table>
Packet switching and QoS

Drawbacks:
- **No reliability**: packets can be dropped on the route
- **Buffering/Queuing induces delays**
- **No bandwidth allocation algorithm**

<table>
<thead>
<tr>
<th>Application</th>
<th>Reliability</th>
<th>Delay</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>File transfer</td>
<td>++</td>
<td>- -</td>
<td>+ -</td>
</tr>
<tr>
<td>Web access</td>
<td>++</td>
<td>+ -</td>
<td>+ -</td>
</tr>
<tr>
<td>Video on Demand</td>
<td>- -</td>
<td>- -</td>
<td>+ +</td>
</tr>
<tr>
<td>Telephony</td>
<td>- -</td>
<td>+ +</td>
<td>- -</td>
</tr>
</tbody>
</table>
Transport Control Protocol (TCP)
Principles

Reliability:
- An acknowledgement for each packet.
- Each lost packet is retransmitted.
Principles

Reliability:
- An acknowledgement for each packet.
- Each lost packet is retransmitted.

Bandwidth allocation and congestion control:
- No loss detected: Sending rate increased linearly.
- Losses detected: Sending rate divided by a factor.
Problem solved?

1980  1990  2000  2010

TCP

Why so many versions?
Problem solved?

1980  1990  2000  2010

TCP

Tahoe
Reno

Why so many versions?
Problem solved?

1980  1990  2000  2010

TCP

Tahoe  Reno  Vegas  WTCP
Problem solved?

1980  1990  2000  2010

TCP

Tahoe
Reno

Vegas

Westwood

Fast

Illinois

Compound

Cubic

Libra
Problem solved?

1980  1990  2000  2010

TCP
Taheoe  Reno
Vegas  Westwood
WTCP  Fast
Illinois
Compound
Cubic
Libra

Why so many versions?
Problem solved?

1980  1990  2000  2010

TCP

Tahoe  Reno  Vegas  Westwood  Fast
WTCP  Illinois  Compound
Cubic  Libra

Why so many versions?
TCP issues

- TCP over wireless

Users are required to cooperate and be well-behaved. What if computers stop being polite?
TCP issues

- TCP over wireless
- Losses in burst
TCP issues

- TCP over wireless
- Losses in burst
- High-speed networks
TCP issues

- TCP over wireless
- Losses in burst
- High-speed networks
- but mainly

Users are required to cooperate and be well-behaved. What if computers stop being polite?
TCP issues

- TCP over wireless
- Losses in burst
- High-speed networks
- but mainly

TCP-friendliness

Users are required to cooperate and be well-behaved
TCP issues

- TCP over wireless
- Losses in burst
- High-speed networks
- but mainly

TCP-friendliness

Users are required to cooperate and be well-behaved

What if computers stop being polite?
No congestion control: the Law of the Jungle
Principles

- Use source coding (digital fountains) to protect against losses

- Send data in the network at maximum rate (worst case)

- Bandwidth allocation is determined by dropping policies in routers
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Tail Dropping

When the queue is full, drop any new packet
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Dropping policies

Fair Dropping

When the queue is full, drop from the biggest flow
Fluid model

Mathematical model:

- Packets replaced by fluid of data
- Flows arrive in the network according to Poisson processes
- Each flow has an exponentially distributed size

With the dropping policy, it defines a Markov process.
Fluid model

**Stability**: The network is stable if each flow is served in finite time.

**Maximal stability condition**: At each link, the offered traffic is less than the capacity of the link.
Fluid model

**Stability:** The network is stable if each flow is served in finite time.

**Maximal stability condition:** At each link, the offered traffic is less than the capacity of the link.

When is the network stable?
Tail Dropping

Maximal stability condition!
Tail Dropping

Maximal stability condition!

Suboptimal stability condition!
Tail Dropping

Maximal stability condition!

Suboptimal stability condition!

Never stable! Congestion collapse!
Fair Dropping

**Theorem:** Any network is stable under the maximal stability condition.

**Source coding** and **Fair dropping** are sufficient to ensure an efficient use of the network resources independently of users!
Main simplification: Remove all randomness except arrivals and departure
Technical corner

**Main simplification:** Remove all randomness except arrivals and departure

Stability of network $\iff$ Ergodicity of Markov process
Technical corner

**Main simplification:** Remove all randomness except arrivals and departure

Stability of network $\iff$ Ergodicity of Markov process

Proofs based on:

- Stochastic domination
- Lyapunov functions
- Scaling methods (fluid limits)
Technical corner

Main simplification: Remove all randomness except arrivals and departure

Stability of network $\iff$ Ergodicity of Markov process

Proofs based on:
- Stochastic domination
Main simplification: Remove all randomness except arrivals and departure

Stability of network ⇔ Ergodicity of Markov process

Proofs based on:
- Stochastic domination
- Lyapunov functions
Main simplification: Remove all randomness except arrivals and departure

Stability of network ⇔ Ergodicity of Markov process

Proofs based on:
- Stochastic domination
- Lyapunov functions
- Scaling methods (fluid limits)
Conclusion

- An example of mathematical analysis of network performance
Conclusion

- An example of mathematical analysis of network performance

- Will the Internet collapse?
Conclusion

- An example of mathematical analysis of network performance

- Will the Internet collapse?
  - With Fair Dropping, no
  - With Tail Dropping, possibly...
Thank you!