Decongestion Transport Protocol: a Transport Protocol for Anarchical Networks

P-U Tournoux (Univ. Réunion), E. Lochin (ISAE), J. Lacan (ISAE)

Anarchical Networks

- **Principles given by**
  - A. Snoeren : **Decongestion Control** (ACM HotNets'07)
  - T. Bonald : **Is the Law of Jungle Sustanaible for the Internet** (IEEE Infocom'09)
- **Main hypothesis**
  - A sender can emit as fast/much as possible
  - Use of retransmission is not recommended / No retransmission possible
    - Need a perfect coding scheme (i.e. each received packets must be a useful packet)
    - Repair symbols are genererated arbiritraly
  - **Fair AQM strategies must be deployed inside the network**
    - No E2E congestion control but fair AQM is mandatory

Anarchical Networks (state of the art)

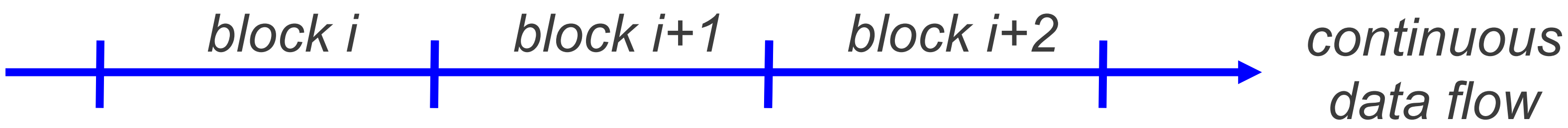
- **A. Snoeren: Decongestion Control**
  - First paper in the area
  - Proposal (called **Achoo**) based on a block code scheme
  - No implementation/measurments
- **T. Bonald: Is the Law of Jungle Sustanaible for the Internet**
  - Fluid model that evaluates the efficiency of the concept
  - No proposal, fully assume the aformentioned hypothesis
  - Claim : overestimation of the congestion collapse
  - Advantage : resistant to DDOS, misconfiguration, ...
  - Efficiency is 80 % while reaches 100 % with fair-dropping mechanisms in the core

Our proposal

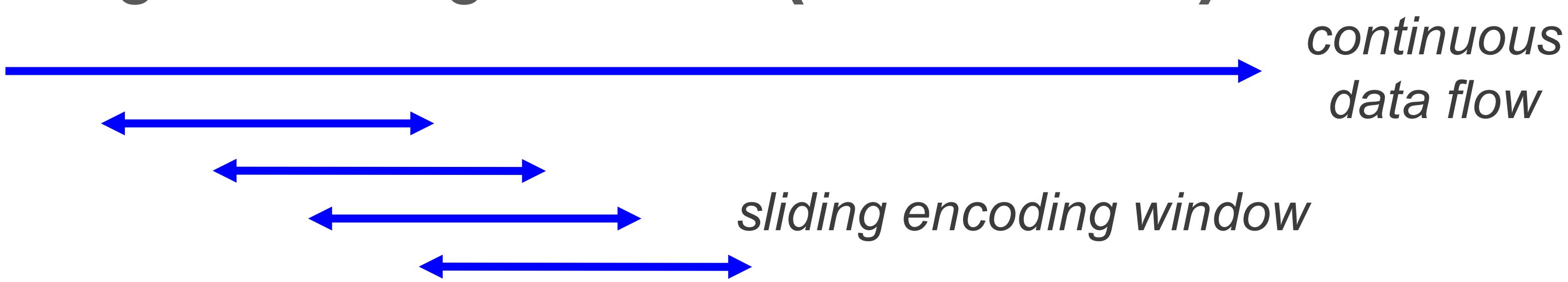
- **To propose an efficient mechanism for Anarchical Networks**
  - Based on an on-the-fly convolutional coding scheme named **Tetrys**
    - http://websites.isae.fr/tetrys
  - Objective: to approach the performance of a perfect coding scheme in order to allow the deployment of a « de-congestion » transport protocol
- **Implementation of such proposal called DCTP (DeCongestion Transport Protocol)**
  - Comparison with our own Achoo [Snoeren] implementation over several topolgies studied in [Bonald]
  - Evaluation of the concept with real-time applications

Convolutional Network-Coding

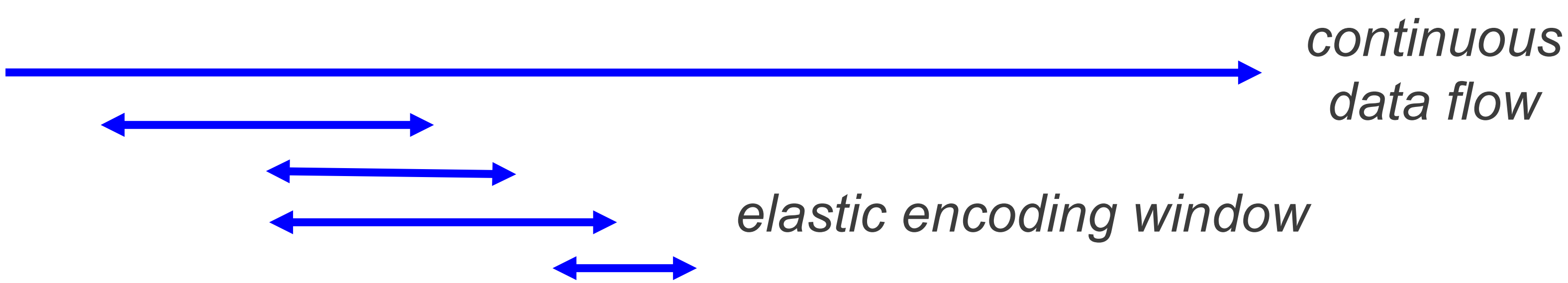
- **Blocks FEC Coding**



- **Sliding encoding window (E. Martinian)**



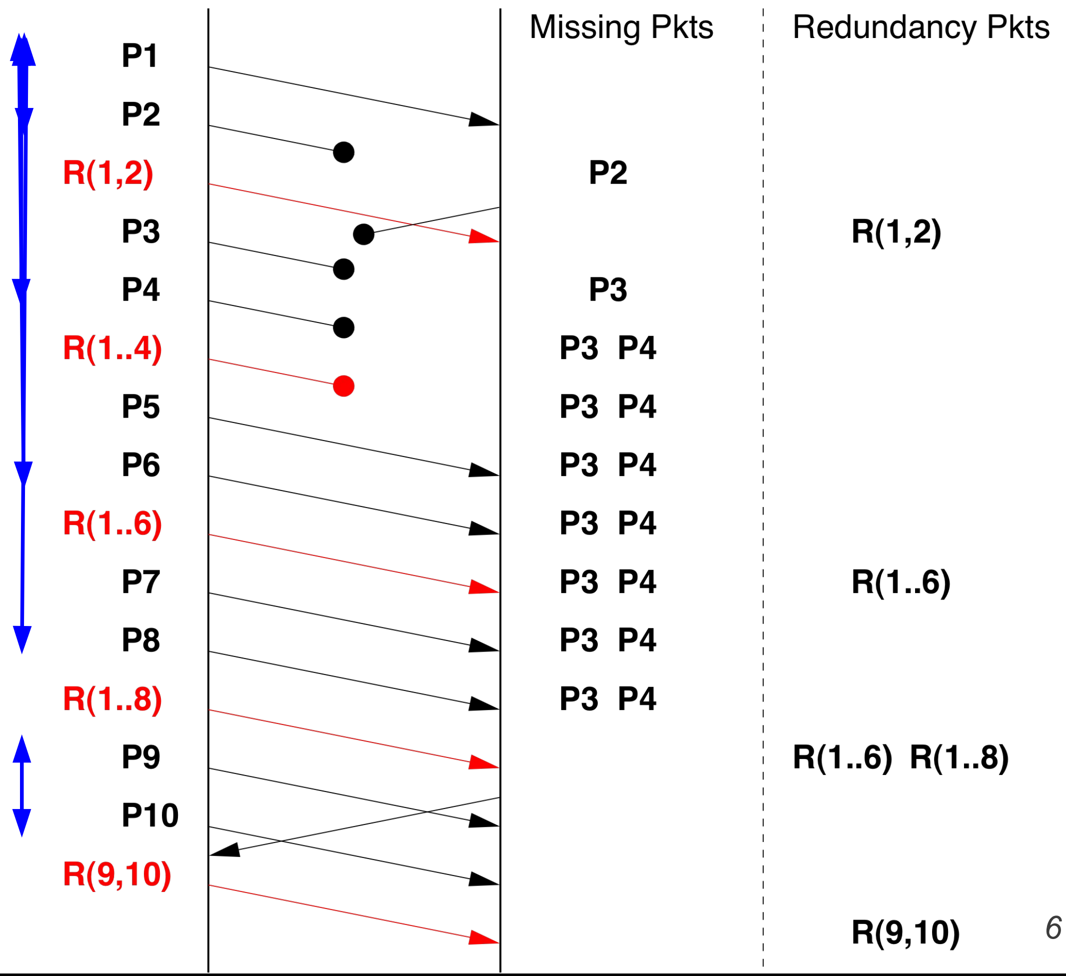
- **Tetrys : elastic encoding window**



Convolutional Network-Coding

- **The encoding window can be adjusted depending on the feedback**

In this example, R(1,2) is a redundant packet calculated with the two packets of the encoding window, P1 and P2. Since the acknowledgment of P1 has been lost, the encoding window grows and encompasses packets P1 to P4 the next time a redundant packet is calculated, R(1,4). In the absence of acknowledgment, the encoding window keeps on growing… until an acknowledgment of packets P1 to P8 arrives at the sender. Then the encoding window is updated to contain only P9 and P10, and the R(9,10) redundant packet is then calculated.



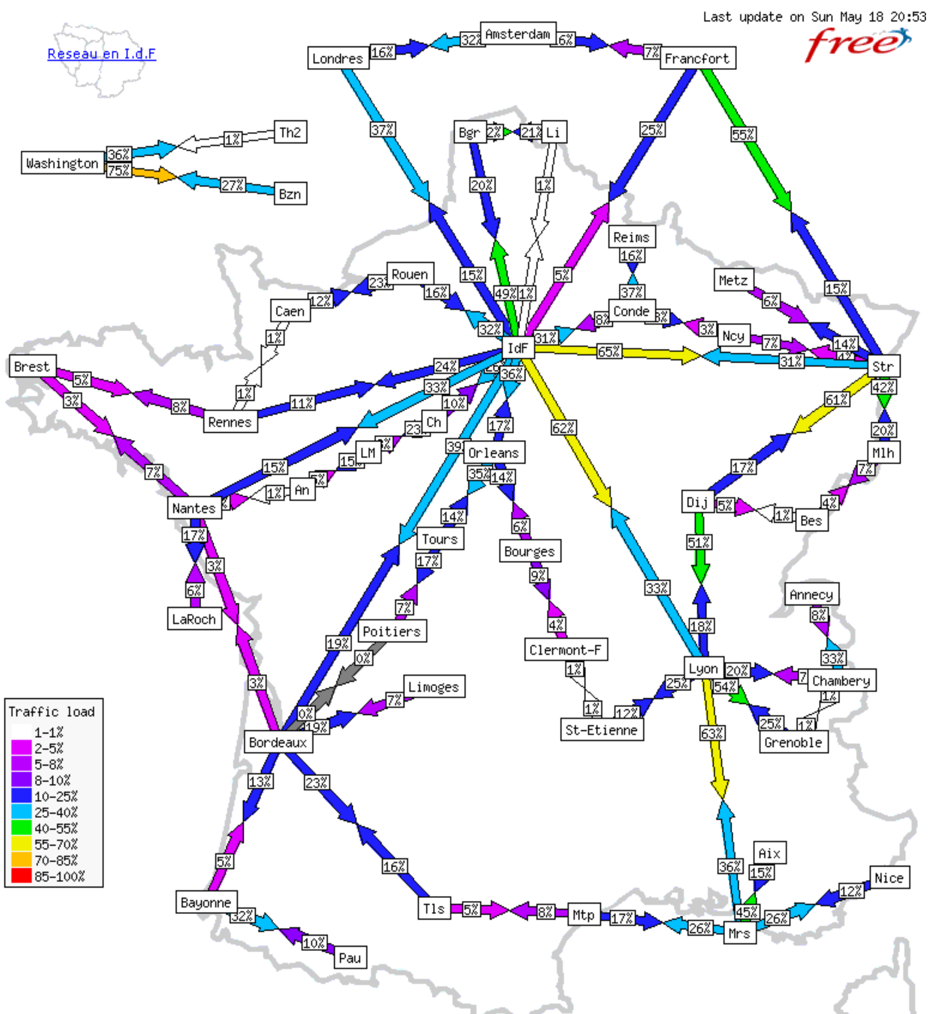
Convolutional Network-Coding

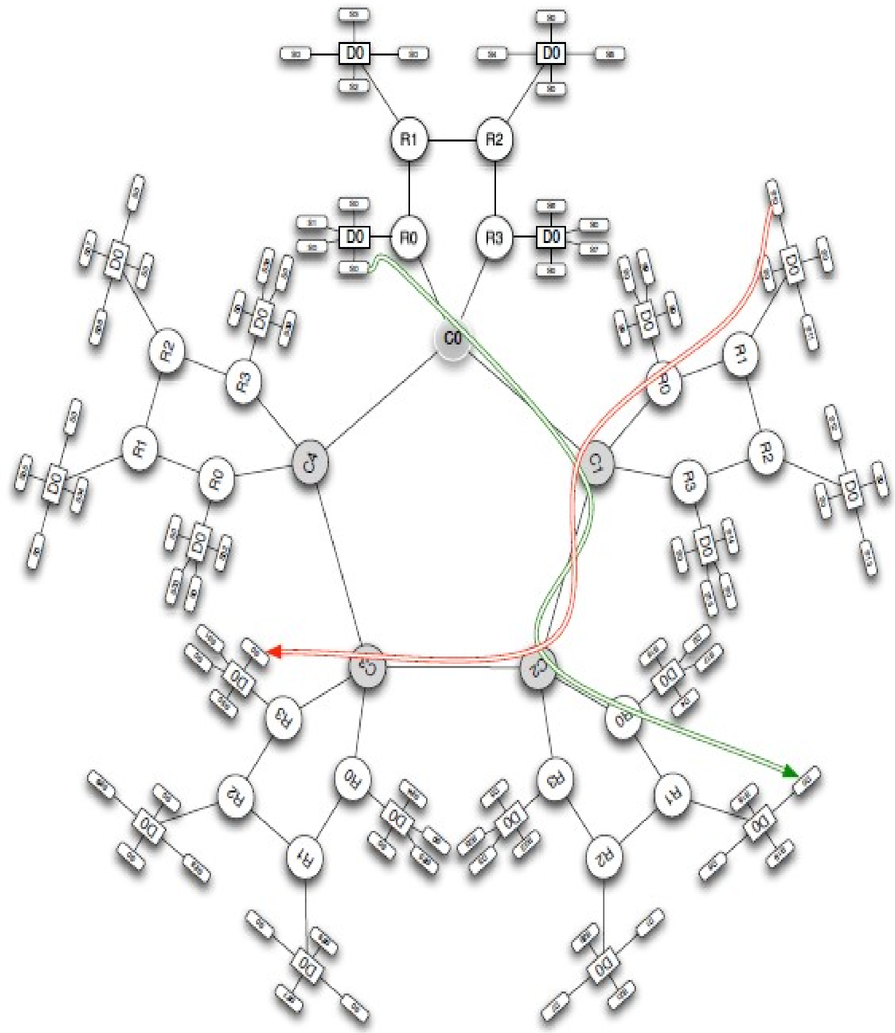
- **Blocks FEC Coding**
  - burst erasure protection increases with the block size
  - as well as the encoding/decoding complexity
  - there are limits on the block size with real time flows
  - useless data are sent between two blocks
- **Tetrys, elastic encoding window**
  - full reliability is achievable
  - encoding/decoding complexity are only function of the RTT
  - the recovery delay is **independent of the RTT**
  - recovery delay easily adjustable with the code-rate
  - Compliant with elastic and non-elastic traffic requirements

DCTP

- **Two versions of DTCP are tested**
  - **DTCP-max**
    - The sender always emits at his maximum throughput
  - **DTCP-min**
    - Always sends at a higher rate than the theoretical e2e fair-share but tends toward this value in order to reduce «dead packets » (packets not useful)
- **Tests with ns-2 over an equivalent ISP topology**

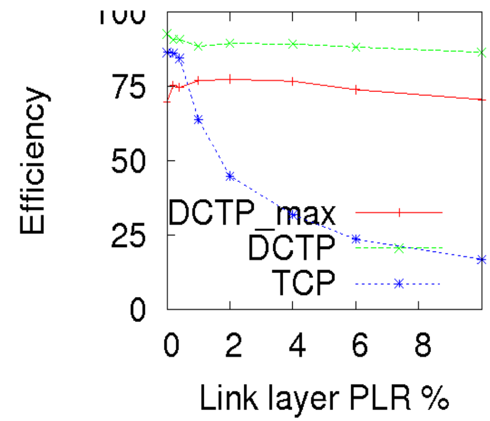
Topology used in ns-2



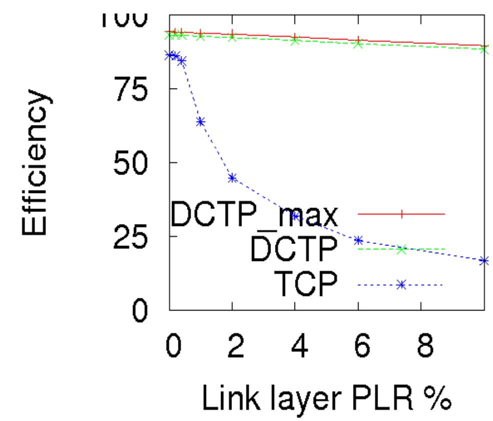


Comparison DCTP / TCP : - Loss rate tolerance

- **Queue size on router :**
  - 20 for DCTP
  - BDP for TCP
- **BW between core routers**
  - 16mb/s
- **Efficiency is defined as**
  - $\frac{\textit{achieved total goodput}}{\textit{maximum achievable total goodput}}$
- **Losses occures on access links**



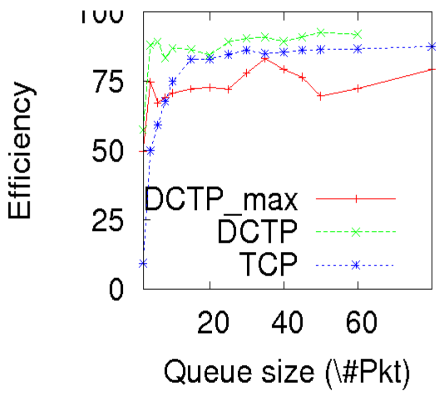
- **Without AQM (Drop Tail)**



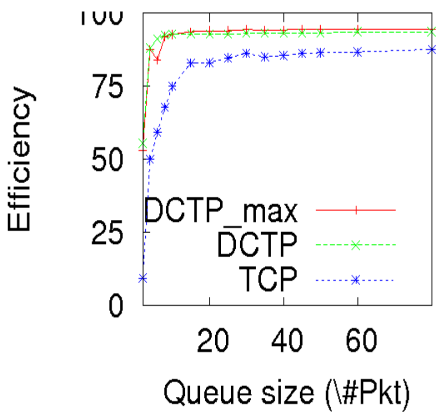
- **With AQM (Fair Drop)**

Comparison DCTP / TCP :
- Impact of router's queue size

- **Queue size on router :**
  - As shown on the x-axis
- **BW between core routers**
  - 16mb/s
- **Efficiency is defined as**
  - $\frac{\text{achieved total goodput}}{\text{maximum achievable total goodput}}$



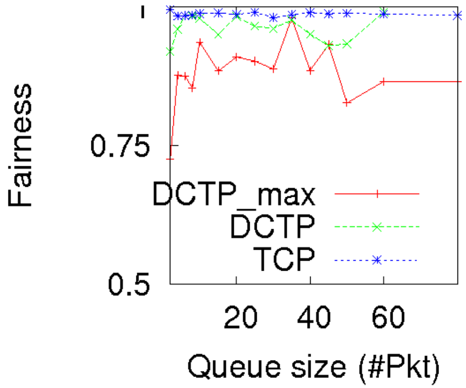
- **Without AQM (Drop Tail)**



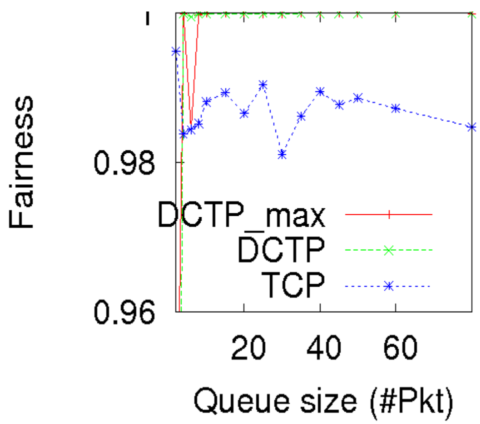
- **With AQM (Fair Drop)**

Comparison DCTP / TCP :
- Note on fairness

- **Queue size on router :**
  - As shown on the x-axis
- **BW between core routers**
  - 16mb/s
- **Fairness is defined as**
  - Jain's index



- **Without AQM (Drop Tail)**



- **With AQM (Fair Drop)**

Comparison DCTP / TCP :
- Promising remarks

- DTCP allows to reach max capacity despite link layer losses
- DCTP-min saves a lot of capacity compared to DCTP-max, the fully anarchical version
- DCTP allows to use small queue size (< 10) compared to TCP
- AQM (Fair-drop) isn't mandatory but increases the capacity by (10~20 %) and allows DTCP to achieve better fairness than TCP