Robust Network Coding in Transport Networks

Bence Ladóczki, Carolina Fernandez, Oscar Moya, Péter Babarczi, János Tapolcai, Daniel Guija

Budapest University of Technology and Economics, MTA-BME Future Internet Research Group, Hungary
i2CAT Foundation, Distributed Applications Networks Area, Spain

Motivation

The **MINERVA Open Call** project proposes a robust network coding scheme in transport networks and implements it in the OpenFlow facility of the pan-European research and education network (GÉANT). This novel scheme brings together the

- *capacity efficiency* (of network coding),
- *simplicity* (by XOR coding), and
- *low recovery time* (avoiding flow rerouting or packet retransmission)

of known protection schemes, by dividing the user data into two parts and ensuring resiliency for the practical case of a single link failure.

A network coding implementation based on the proposed scheme and UDP makes this a real alternative to the reliable service of TCP, by trading redundancy for recovery time.

Definitions

Definition 1: A routing is ***survivable*** if sufficient information reaches the destination, even if a single link fails (e.g., a link-disjoint path-pair is survivable). Otherwise, it is ***vulnerable***.

Definition 2: A survivable routing is ***critical*** if we cannot further decrease the flow value along an arbitrary edge without making the routing vulnerable. (**Corollary:** a minimum cost survivable routing is critical).

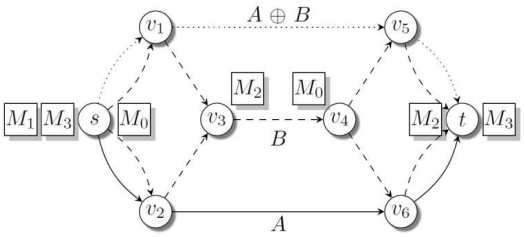
Theoretical results (Resilient Flow Decomposition)

Theorem 1: The minimum cost survivable routing (i.e., the optimal coding subgraph for network coding) can be computed in polynomial time.

Alija Pašić, János Tapolcai, Péter Babarczi, E. Bérczi-Kovács, Zoltán Király, and Lajos Rónyai, Survivable Routing Meets Diversity Coding, in Proceedings of the IFIP Networking (Networking), pp. 1-9, Toulouse, France, 2015.

Theorem 2: Every critical survivable routing can be decomposed into three end-to-end directed acyclic graphs (resulting simple network codes).

Péter Babarczi, János Tapolcai, Lajos Rónyai, and Muriel Médard, Resilient Flow Decomposition of Unicast Connections with Network Coding, in Proceedings of the IEEE International Symposium on Information Theory (ISIT), pp. 116-120, Honolulu, HI, USA, 2014.



NF modules for Resilient Flow Decomposition (RFD)

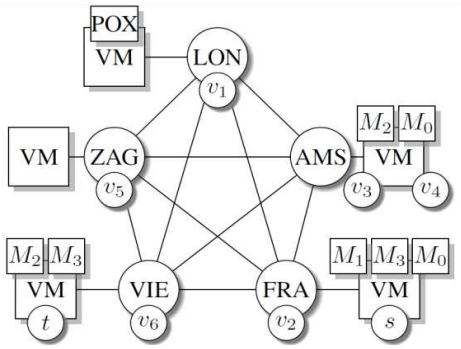
Splitter (M_0): duplicates incoming packets and forwards them through two different links (e.g., at nodes s and v_4).

Sequencer (M_1): divides the input stream at the source node s into flows A and B (e.g., based on parity) and marks each with its own MPLS label.

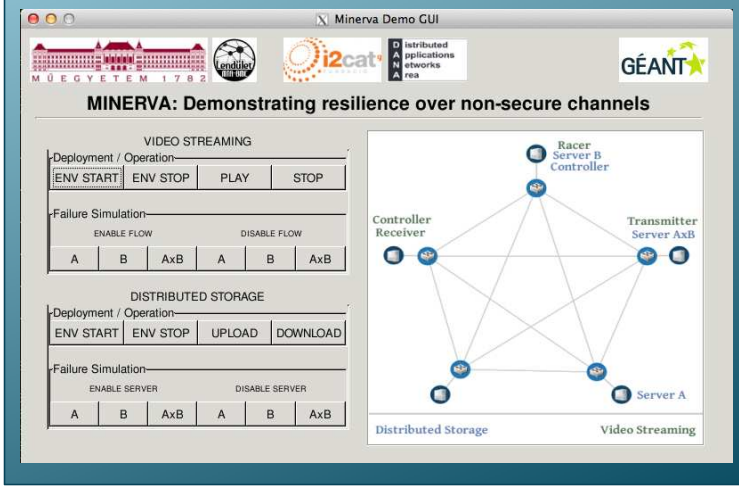
Merger (M_2): receives the same flow (i.e., with the same MPLS label) on two incoming links and forwards one of them (or the intact one upon link failure) through its single outgoing link. See nodes v_3 and t.

Coding/Decoding (M_3): closely related NFs that perform fast packet processing using XOR operation and queues to handle the incoming packets. These functions are always placed at s and t.

Péter Babarczi, Alija Pašić, János Tapolcai, Felicián Németh, and Bence Ladóczki, Instantaneous Recovery of Unicast Connections in Transport Networks: Routing versus Coding, accepted to Elsevier Computer Networks (ComNet), impact factor 1.282 (in 2013), 2015



Demonstration of use cases



Video Streaming

A video stream is split in three flows and sent from specific nodes (s to t) through three disjoint paths – one of them traversing the racer machine. The different flows are consecutively enabled and disabled to demonstrate that two flows are enough to recover the stream in the transmitter.

Distributed Storage

A given file is encoded and split in three chunks, then each uploaded via the client, distributed across disjoint paths computed on-the-fly, and stored in a server from the data center. Upon download, the received chunks are properly transformed for recovery. The failures are simulated by disabling the reception of the different flows in the servers.