Softwarisation and Virtualisation of Network Coding Function
Link between NWCRG - NFVRG

M. Á. Vázquez-Castro, (Autonomous University of Barcelona, Spain)
L. M. Contreras (Telefonica I+D / Global CTO Unit / Telefónica, Spain)
S. P. Romano (University of Napoli, Italy)

Layout

1. **Motivation**
   - Network coding as a function (NCF) provided as an operator service (on Demand).
2. **Example use-cases**
   - NC for reliable multicast with re-encoding - overlay (hybrid) networks.
   - NC for efficient caching - broadcast (hybrid) networks.
3. **Proposal for Softwarisation and Virtualisation**
   - Softwarisation: NCFs functional architecture mapped to SDN
     - NCF design:
       - Coding domain
       - **Functional domain (and APIs) - software functional architecture**
       - Protocol domain
   - Virtualization: integration with ETSI VNF architectures
4. **Links to other work**
   - WGs: NWCRG-NFVRG
   - Drafts.
5. **Next steps.**

Motivation

Network Coding as a Function (NCF)

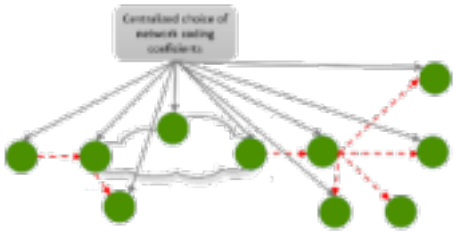
- Under the network softwarization point of view Network Functions (NFs) are pieces of software, which should be efficiently designed, deployed and executed.

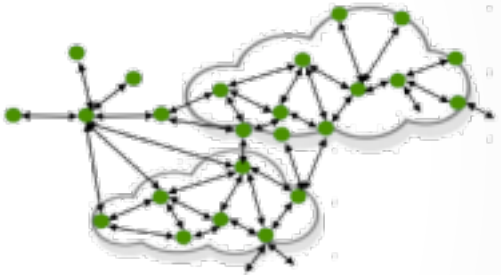
- NFs are currently subject of great development by diverse vendors for a large range of applications beyond traditional well known networking functionalities.

- **Network Coding can be designed as a Network Coding Function (NCF):**
  - **Why?: next slide.**
  - **How?: the rest of slides.**

Why NCFs?

- Our preferred explanation: due to two distinguishing features of network coding.
  - **TRANSVERSAL APPLICATION: NC can be applied over packet flows at different layers in many network scenarios with different degrees of distributed and centralized (coding) operation.**





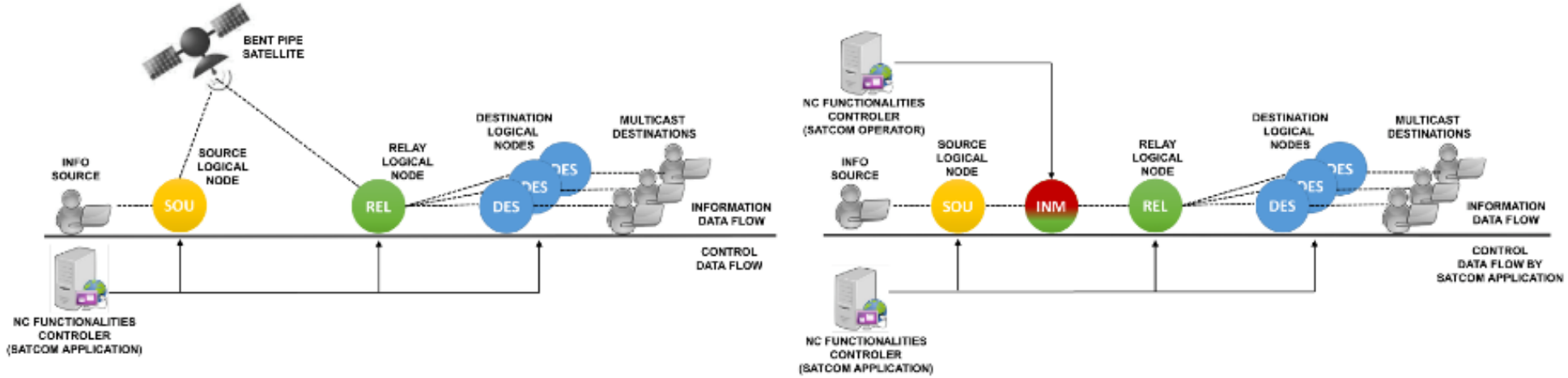
NCFs allows implementing NC operation in all above cases as software pieces that can be re-used: versatility, modularity and scalability.

  - **"MATHEMATICAL" FLOW ENGINEERING: with NC, packets flows can be interpreted as mathematical objects that can be mathematically transformed by the NCF, which induces properties on the flow.**

NCFs allows to be interpreted as a flow engineering service given by network operators ON DEMAND.

Examples of Use-Cases

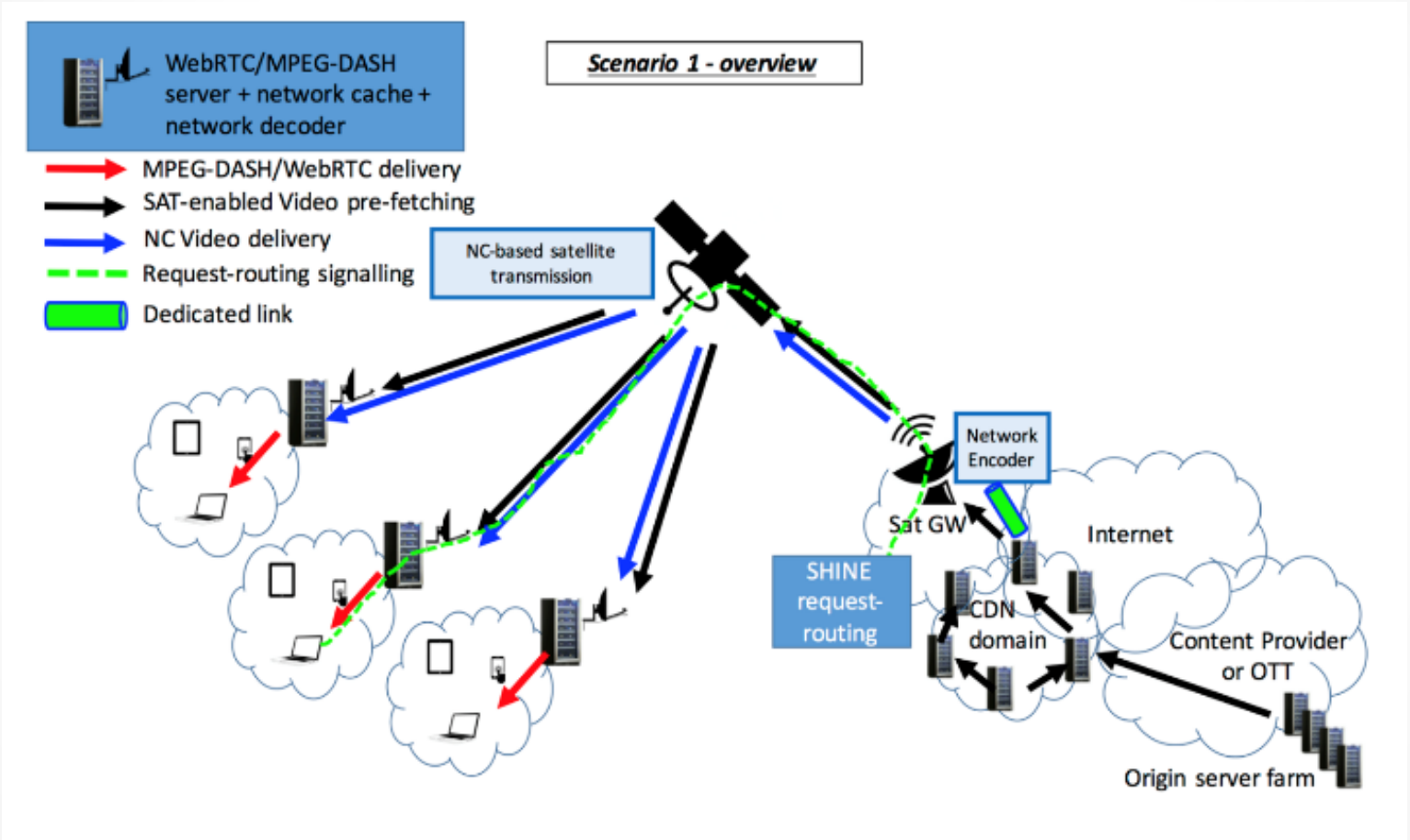
NC for reliable multicast with re-encoding - overlay (hybrid) networks.



Example of simple single relay NC topology (left) and two recoders (right) with NC controllers. Black arrows indicate data flows and dashed arrows control flows.

SatNetCode ESA-funded project.

NC for efficient caching - broadcast (hybrid) networks.



SHINE (SECURE HYBRID IN NETWORK CACHING ENVIRONMENT) ESA-funded project

Proposal for Softwarisation and Virtualisation

IETF #101 – London

Proposal for softwarisation and virtualisation

- Our **proposal for softwarization of NCFs:** visible internal logic (as opposite to legacy black box NFs).
  - Given as functional software architecture that maps to SDN architectures, hence, centralised control.
    - Note: still enables interoperable NCFs with proprietary blocks. **The approach prevents stagnation of networking ecosystems due to proprietary software in softwarized networks.**

- Our **proposal for virtualization of NCFs:** integration with ETSI VNF architectures
  - It takes advantage of unified computation and network resources.

Preliminary ideas to be updated in

draft-vazquez-nfvrg-netcod-function-virtualization-02

Design domains of NCFs

Coding Domain. This is the domain for the design of codebooks, encoding/decoding schemes, identification of performance theoretical limits and maps whereby mathematical objects map to information entities and vice versa.

Functional Domain. This is the domain to design functionalities needed to make network coding to achieve service-intent based design objectives.

Protocol Domain. At some point, functions need to instantiate processes in time possibly within and across different logical layers and/or planes, and this is done designing protocols. This is also the domain for the design of the physical transportation of the network coded information flow.

Preliminary ideas to be updated in

draft-vazquez-nfvrg-netcod-function-virtualization-02

Network coding functions with (visible) internal logic defined as a software architecture that can be mapped to SDN and VFN architectures.

Design domains of N

Coding Domain. Th
codebooks, encodi
performance theoretical limits and
mathematical objects map to info
versa.

n of
s whereby
n entities and vice

Functional Domain. This is the domain to design functionalities needed to make network coding to achieve service-intent based design objectives.

Protocol Domain. At some point, functions need to instantiate processes in time possibly within and across different logical layers and/or planes, and this is done designing protocols. This is also the domain for the design of the physical transportation of the network coded information flow.

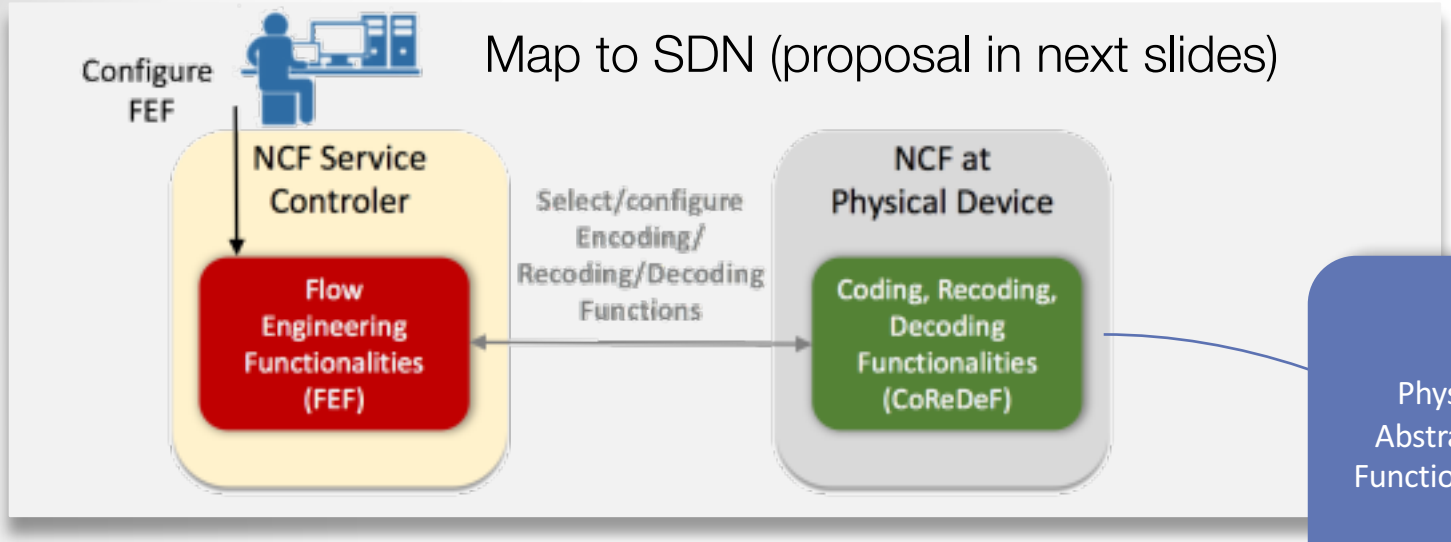
Functional domain: software architecture

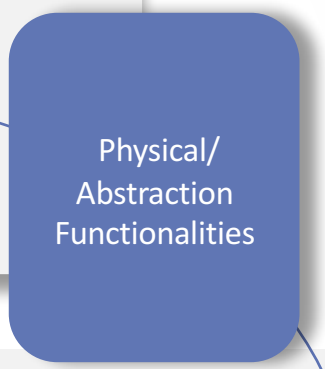
draft-vazquez-nfvrg-netcod-function-virtualization-02

We identify a **software architecture with three functional components**, which assume **centralised control.**

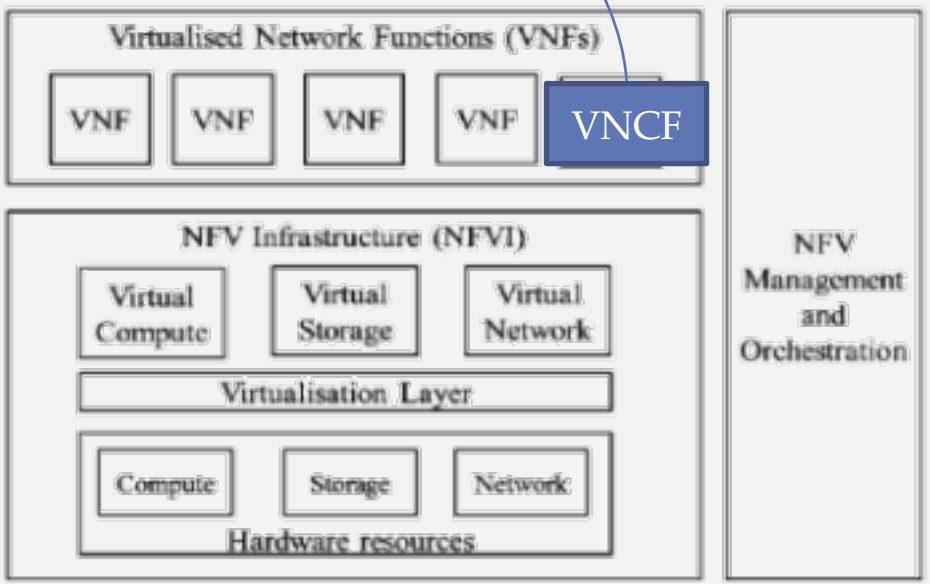
- **Coding, Recoding, Decoding Functionalities (CoReDeF)**
  - These functionalities should be able to instantiate different network coding options, e.g.
    - Different coefficients: random, structured (e.g. regenerative), hybrid.
    - Different finite-length behaviour: admisible NC schemes.
- **Flow Engineering Functionalities (FEF)**
  - FEF refers to the per-flow or per-aggregated flow rules that shape the flow-level ("microscopic") behaviour that will induce the network-wide ("macroscopic") behaviour of the traffic flows, which are also affected by the action of other NFs.
- **Physical/Abstraction Functionalities (PAF)**
  - These functionalities realise the interaction with available storage, computation and other physical devices and resources that are abstracted by the other functionalities. These functionalities also realise virtualization.

Map/integration to SDN and VFN architectures



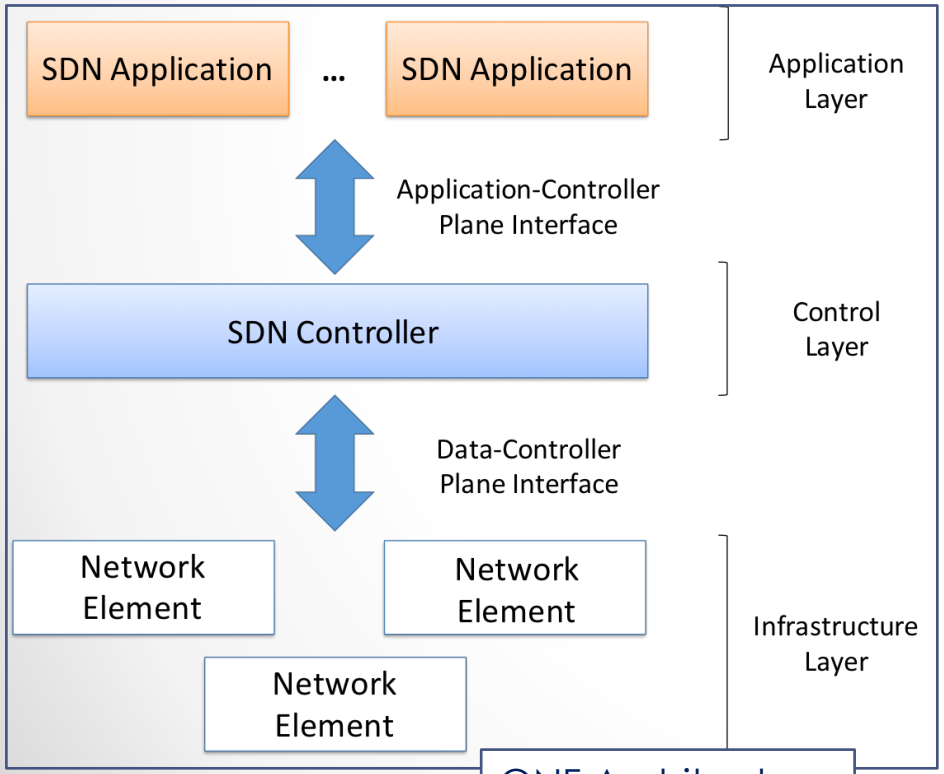


Integration as VNCF in the ETSI NFV architecture.

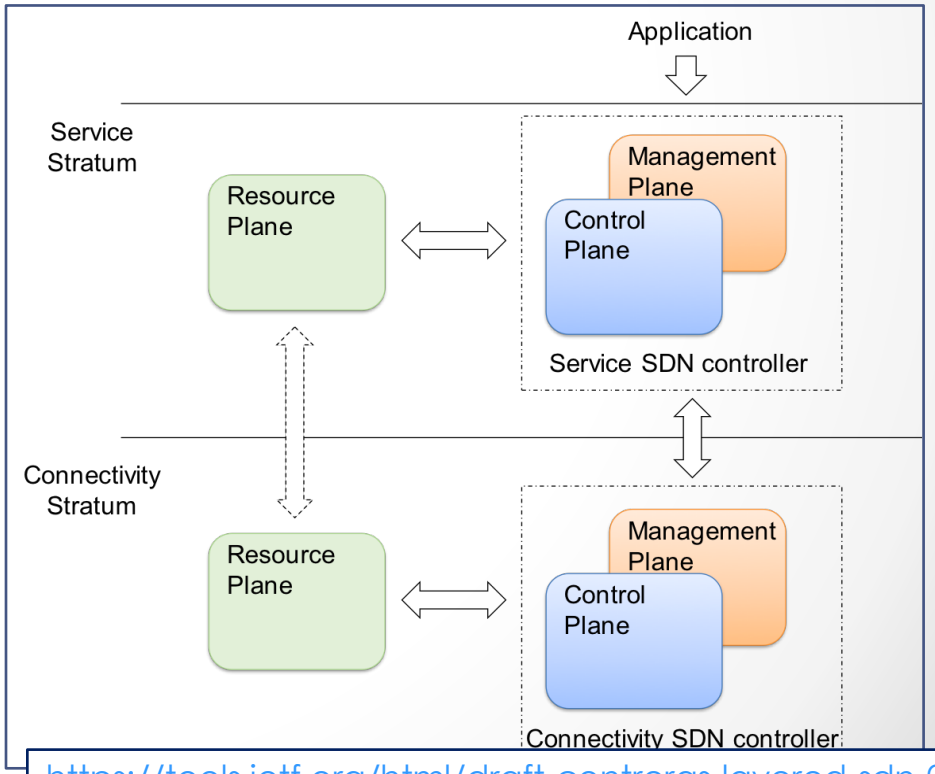


Logical map to SDN

- Our functional software architecture maps to a software network architecture.
- We take as reference the **basic SDN architecture defined by the Open Networking Foundation (OPN):** application layer, control layer and an infrastructure layer.



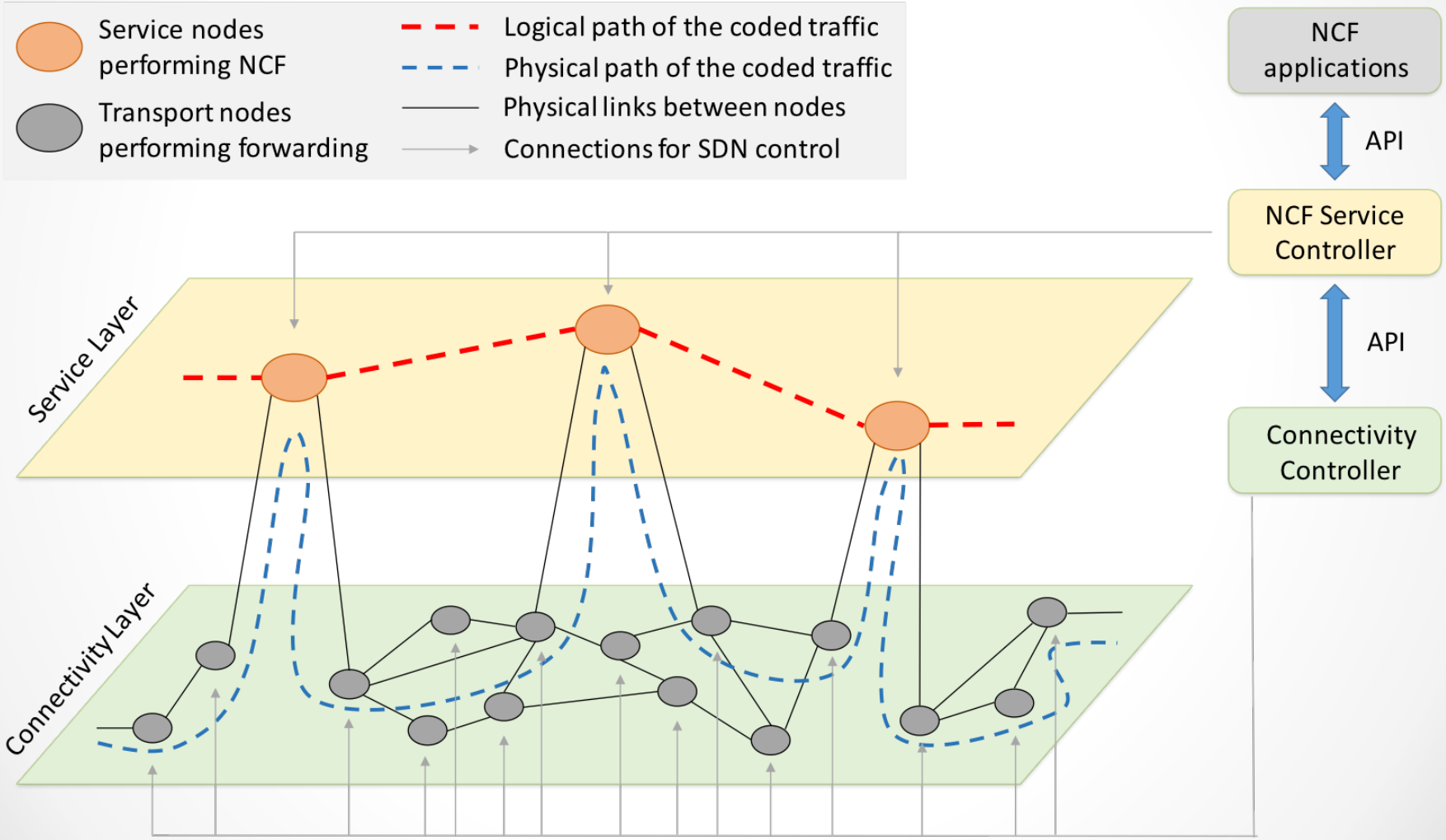
ONF Architecture



https://tools.ietf.org/html/draft-contreras-layered-sdn-01

Logical map to SDN

- Proposal for NCFs.



Links to related ongoing works

Identified links

- RGs
  - **NFVRG - Virtualization Research Group (NFVRG)**
- Drafts
  - https://tools.ietf.org/html/draft-kuhn-nwcrg-network-coding-satellites-03
  - https://datatracker.ietf.org/doc/draft-roca-nwcrg-generic-fec-api/
  - https://tools.ietf.org/html/draft-quic-coding-00
  - https://tools.ietf.org/html/draft-contreras-layered-sdn-01
  - draft-vazquez-nfvrg-netcod-function-virtualization-02

Next steps

Next steps

- **Short term - Next meeting:**
  - Coordinate efforts between **NWCRG – NFVRG** so that work is complementary.
  - Work out example use-cases identifying relevant details of operation.
  - Update drafts accordingly.
- **Medium/long term:**
  - Identify milestones.

- In addition to our work in the **Functional Domain**, we are also working on the **Coding Domain** (codes with algebraic structure).

Thank you