http://www.ietf.org/mail-archive/web/vnfpool/current/msg00224.html

Network functions such as firewalls, load balancers, and WAN optimizers are conventionally deployed as specialized hardware servers in both network operators' networks and data center networks as the building blocks of the network services. There is a trend to implement such network functions as software instances running on commodity servers, via a virtualization layer (i.e., hypervisors). These virtualized functions are called Virtualized Network Functions (VNFs).

We call a group of VNFs a VNF set. A VNF set can include a single or multiple types of VNF (e.g., virtual firewall, virtual load balancer), where each type of VNF corresponds to a number of VNFs which is also referred to as a VNF pool. A VNF set can be used to build network services. For example, a VNF set can be used as a Service Function Chain (SFC), where the VNFs are sequentially connected (i.e., chained) to build a network service. Generally, a VNF set can be used not only as a SFC, but also merely as a collection of multiple VNFs without specific topological constraint.

A VNF set and the virtualized functions can introduce additional points of failure beyond those inherent in a single specialized server, and therefore poses additional challenges for the reliability of the provided services. A single VNF would typically not have built-in reliability mechanisms on its host (i.e., commodity server). Instead, there are more factors of risk such as software failure, server overload, and instance scaling and migration that may lead to VNF failures. Existing pooling and other redundancy mechanisms should be investigated and may be applied to address some reliability issues of a single VNF.

However, the complexity of coordinating a growing number of VNFs including stateful and stateless functions, and extending the redundancy within a VNF set (i.e., multiple pools for multiple VNFs) requires further solution development. For example, when a live VNF pool member goes out of service, how do adjacent entities learn which pool member will replace it? How do VNFs learn the states of adjacent VNFs before the failure of an adjacent VNF happens? How are the service states of a VNF held and accessed for efficient synchronization with backup VNFs?

Ideally, the reliability of a VNF set means that the services provided by such a VNF set will continue throughout an interruption within the VNF set, and the outages of one or more VNFs will not be visible to the users of the VNF set. The VNFPool WG focuses on mechanisms supporting the reliability of a VNF set: redundancy within a VNF set, and stateful failover among VNF pool members. Additional mechanisms for reliable VNF set might be included after further gap analysis between identified requirements and existing IETF technologies. The VNFPool WG currently does not work to resolve the service availability issue, although the reliability of VNF set will benefit service availability.

The overall problem space can be further broken down into the following objectives:
. Signaling between members of a VNF pool, and across different VNF pools for VNF transition (e.g., state change, scaling, moving) notification, and backup advertisement;

. Identification and evaluation of state sharing mechanisms between members of a VNF pool, including distributed shared memory, gossip protocols, pfsync, and state check pointing;
. Exchange of reliability related information between a VNF set and VNF set users, and information between a VNF set and underlying network (e.g., interfacing with ALTO, I2RS);
. Identify and analyze reliable transport characteristics for the aforementioned control plane traffic of VNF pools;
. Analysis of transport security characteristics to provide protection against known threats, and identification of an appropriate trust model;

Initially the VNFPool WG will develop a problem statement, VNF pooling requirements and architecture, use cases, and gap analysis of existing technologies against the architecture and requirements. It is our expectation that we will be able to rely heavily on existing IETF technologies, but that gaps will be found around problems like redundancy mechanisms for a VNF set, state transfer, and trust/security, all of which will need to be considered and addressed. The VNFPool WG will include considerations on the manageability of VNF pools in the requirements and architecture work items. The VNFPool WG will seek re-chartering before adopting any work to develop new, or extend existing, protocols.

Particularly, we will work closely with the SFC WG, as we believe that SFC and VNFPool are independent but complementary. SFC targets on steering packets among VNFs, while VNFPool focuses on the redundancy, e.g., managing active/standby instances, handling failure cases, without caring about how to construct the data path. VNFPool could interact with an SFC control entity to either advertise the status of instance pools, or receive the redundancy requirement from the SFC control entity. VNFPool is not only used in the case of "chained VNFs", but also applicable to other cases where the VNFs are not necessarily sequentially connected.

Goals and Milestones:
December 2014 - Submit VNFPool Problem Statement to IESG for publication as Informational
April 2015 - Submit VNFPool Use Cases to IESG for publication as Informational
August 2015 - Submit VNF Pooling Requirements and Architecture including the manageability of VNF pools to IESG for publication as Proposed Standard
August 2015 - Submit Gap Analysis to IESG for publication as Informational