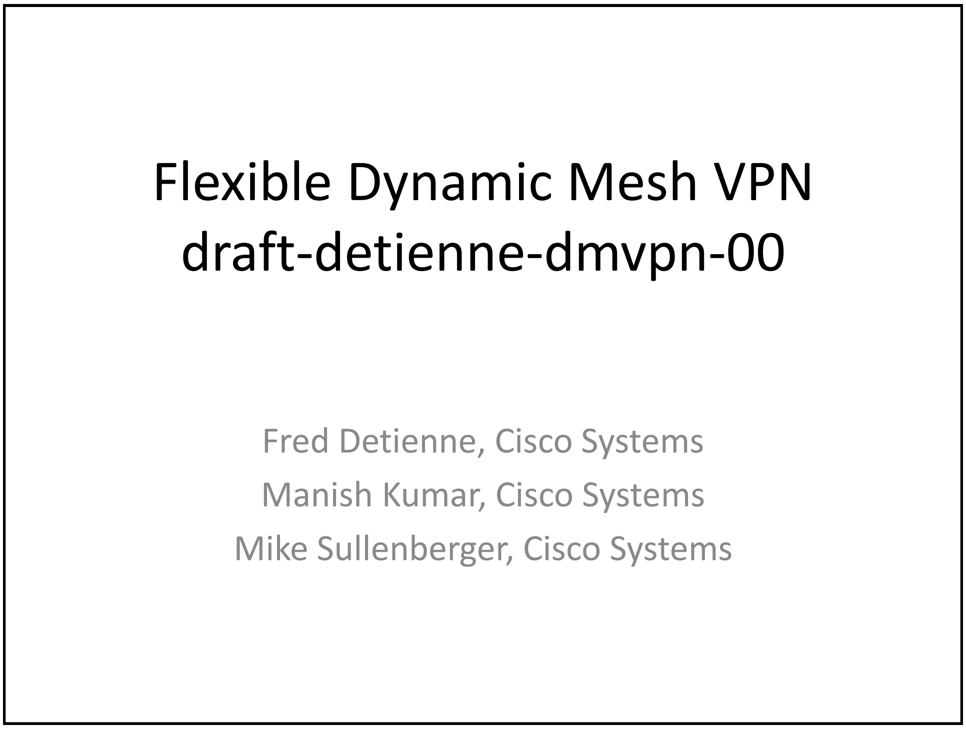
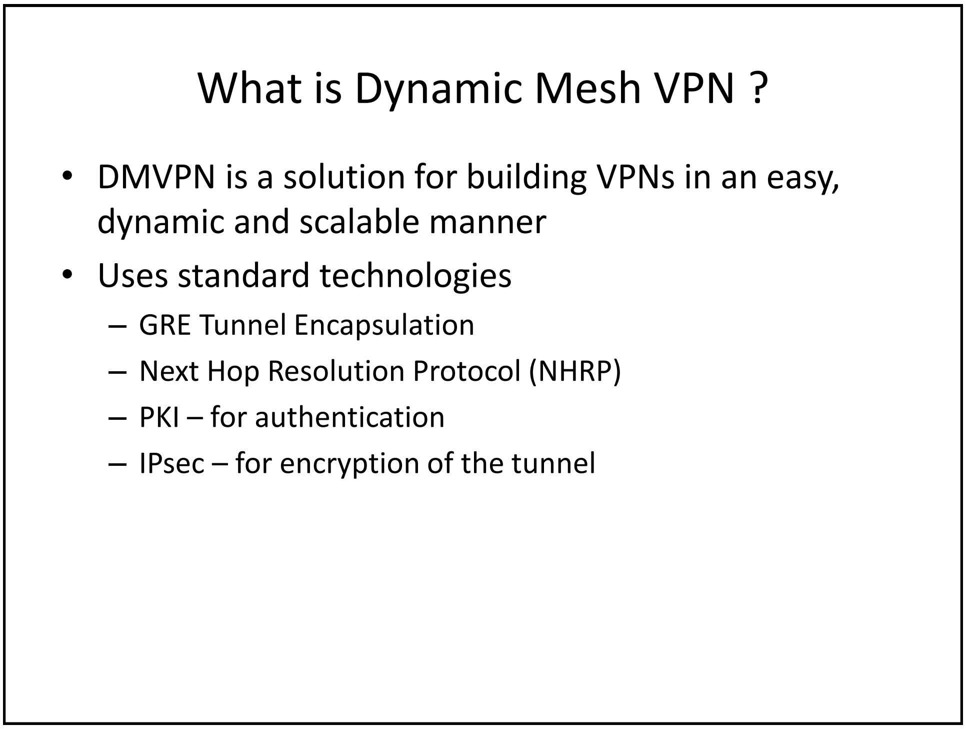


Flexible Dynamic Mesh VPN draft-detienne-dmvpn-00

Fred Detienne, Cisco Systems
Manish Kumar, Cisco Systems
Mike Sullenberger, Cisco Systems



What is Dynamic Mesh VPN ?

- DMVPN is a solution for building VPNs in an easy, dynamic and scalable manner
- Uses standard technologies
  - GRE Tunnel Encapsulation
  - Next Hop Resolution Protocol (NHRP)
  - PKI – for authentication
  - IPsec – for encryption of the tunnel

Terminology

- **Protected Network, Private Network:** a network hosted by one of the nodes. The protected network IP addresses are those that are resolved by NHRP into an NBMA address.
- **Overlay Network:** the entire network composed with the Protected Networks and the IP addresses installed on the Tunnel interfaces instantiating the DMVPN.
- **Transport Network, Public Network:** the network transporting the GRE/IPsec packets.

- **Nodes:** the devices connected by the DMVPN that implement NHRP, GRE, IPsec and IKE.
- **Ingress Node:** The NHRP node that takes data packets from off of the DMVPN and injects them into the DMVPN on either a multi-hop tunnel path or single hop shortcut tunnel. Also the node that will send an NHRP Resolution Request and receive an NHRP Resolution Reply to build a short-cut tunnel.
- **Egress Node:** The NHRP node that extracts data packets from the DMVPN and forwards them off of the DMVPN. Also the node that answers an NHRP Resolution Request and sends an NHRP Resolution Reply.
- **Intermediate Node:** An NHRP node that is in the middle of multi-hop tunnel path between an Ingress and Egress Node. For the particular data traffic in question the Intermediate node will receive packets from the DMVPN and resend them (hair-pin) them back onto the DMVPN.

Base Principles

- Layering model to support separation of Forwarding, Tunneling and Encryption
- Distributed system for scalability
- Redundancy and Load balancing

GRE and NHRP - [RFC2748, RFC2332] used for tunneling

- Both IPv4/IPv6 passenger data over either IPv4 or IPv6 transport.
  - Can easily be extended for other passenger protocols
  - No change needed to IPsec.
- Single tunnel between peers supports any number of data flows
  - Reduces the number of IPsec SAs.
  - Reduces the effectiveness of outsiders doing traffic analysis.
- Easy setup and use of multiple tunnels between peers
  - Redundancy and/or load-balancing of data flows
  - No issue with IPsec SA selectors trying to encrypt same flows to different peers
- NHRP mapping database is naturally distributed across network.
  - No single point of failure or congestion
  - Provides for easy scaling

Standard IP Routing over DMVPN – for forwarding traffic

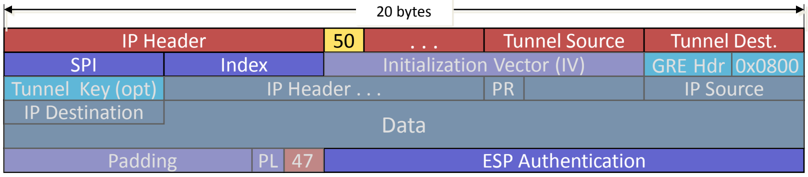
- Dynamically learn destinations automatically
- Changes or movement of hosts/subnets/nodes automatically propagated throughout network
- Already known to handle up to millions of routes
- Standard tools to control routing advertisements
- Optional tools can be layered on top for more sophisticated forwarding control (by application)
- No changes needed in IPsec selectors or policies

IPsec used for Encryption

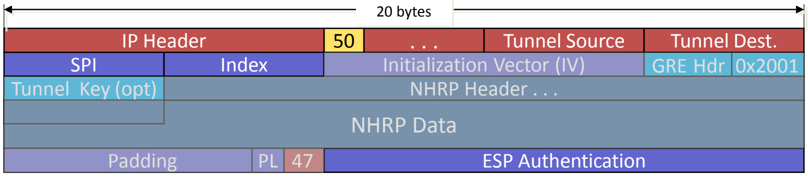
- Encrypt the tunnel not the data flows
  - Can add/remove data flows
    - No change to IPsec selectors
  - Can add new passenger protocols
    - No change to IPsec
- Layering also makes it easier to add in new encryption protocols (Ex: adding support for IKEv2)

IPsec/GRE Packets

IPsec Transport Mode + GRE + IP Data



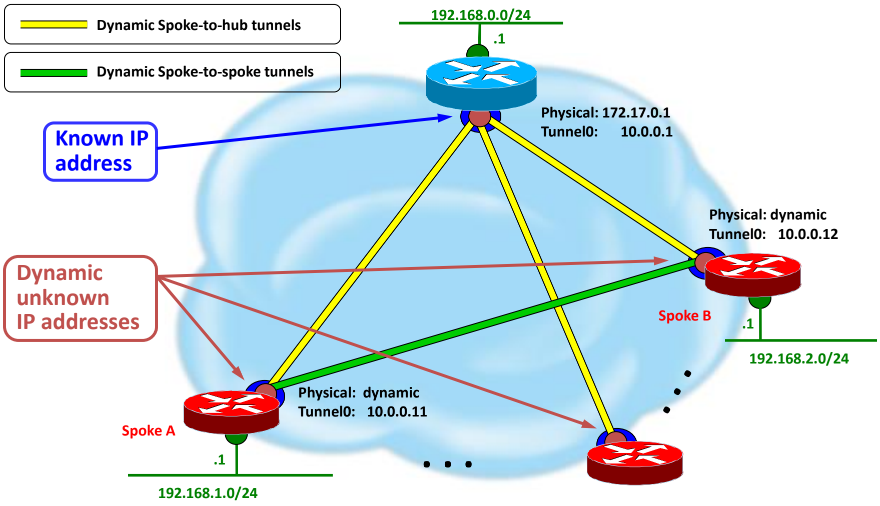
IPsec Transport Mode + GRE + NHRP



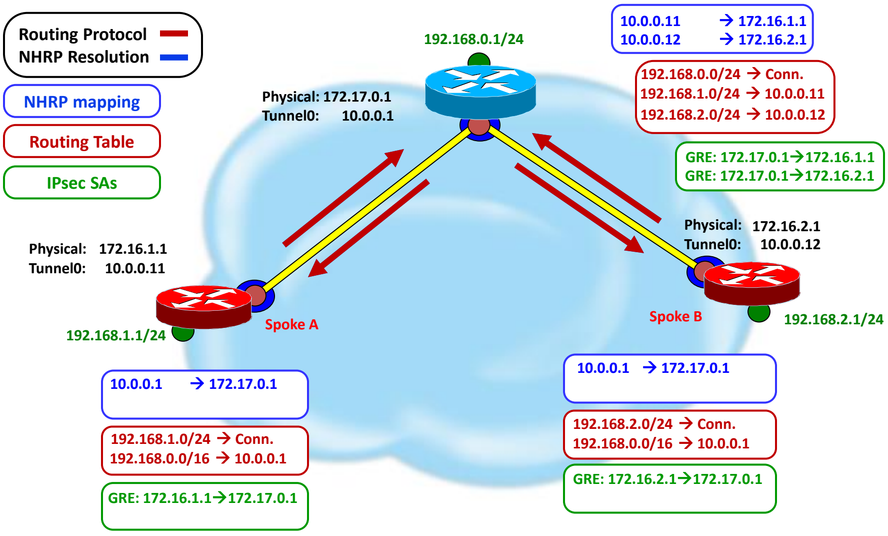
DMVPN – How it works

- Spokes build a dynamic permanent GRE/IPsec tunnel to the hub, but not to other spokes. Spokes register as NHRP clients of the NHRP server, hub.
- When a spoke needs to send packets to a destination (private) subnet behind other spokes, they send packets via the hub(s) path.
- Hubs sends NHRP redirect message to spokes to trigger spokes to build direct path.
- Spokes send NHRP query via the hub(s) path to get the real (outside) address of the remote spoke.
- Spokes can then initiate a dynamic GRE/IPsec tunnel to the remote spoke (because it knows the peer address).

Dynamic Mesh VPN—Example



Registration



Routing Protocol 🟥
NHRP Resolution 🟦

NHRP mapping

Routing Table

IPsec SAs

Hub:

10.0.0.11 → 172.16.1.1
10.0.0.12 → 172.16.2.1

192.168.0.0/24 → Conn.
192.168.1.0/24 → 10.0.0.11
192.168.2.0/24 → 10.0.0.12

GRE: 172.17.0.1→172.16.1.1
GRE: 172.17.0.1→172.16.2.1

Spoke A:

10.0.0.1 → 172.17.0.1

192.168.1.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

GRE: 172.16.1.1→172.17.0.1

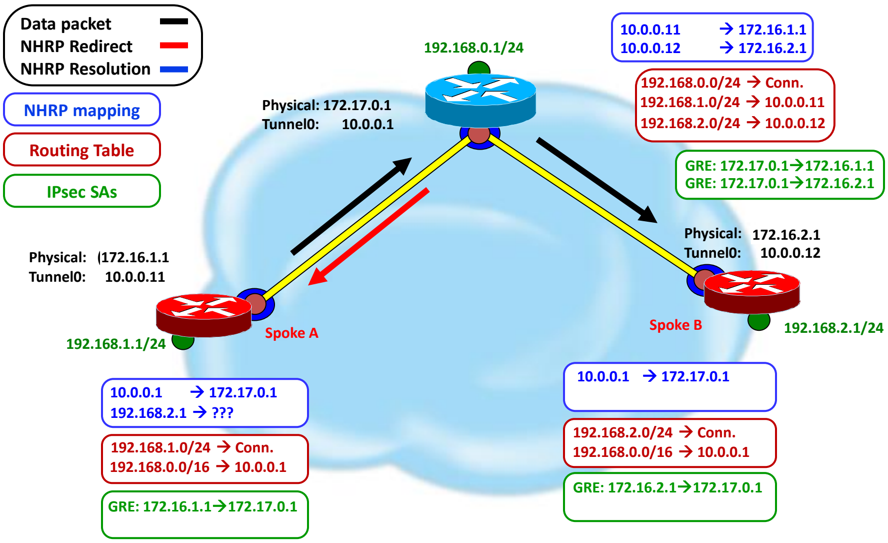
Spoke B:

10.0.0.1 → 172.17.0.1

192.168.2.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

GRE: 172.16.2.1→172.17.0.1

Redirect



Data packet ⬛
NHRP Redirect 🟥
NHRP Resolution 🟦

NHRP mapping

Routing Table

IPsec SAs

Hub:

10.0.0.11 → 172.16.1.1
10.0.0.12 → 172.16.2.1

192.168.0.0/24 → Conn.
192.168.1.0/24 → 10.0.0.11
192.168.2.0/24 → 10.0.0.12

GRE: 172.17.0.1→172.16.1.1
GRE: 172.17.0.1→172.16.2.1

Spoke A:

10.0.0.1 → 172.17.0.1
192.168.2.1 → ???

192.168.1.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

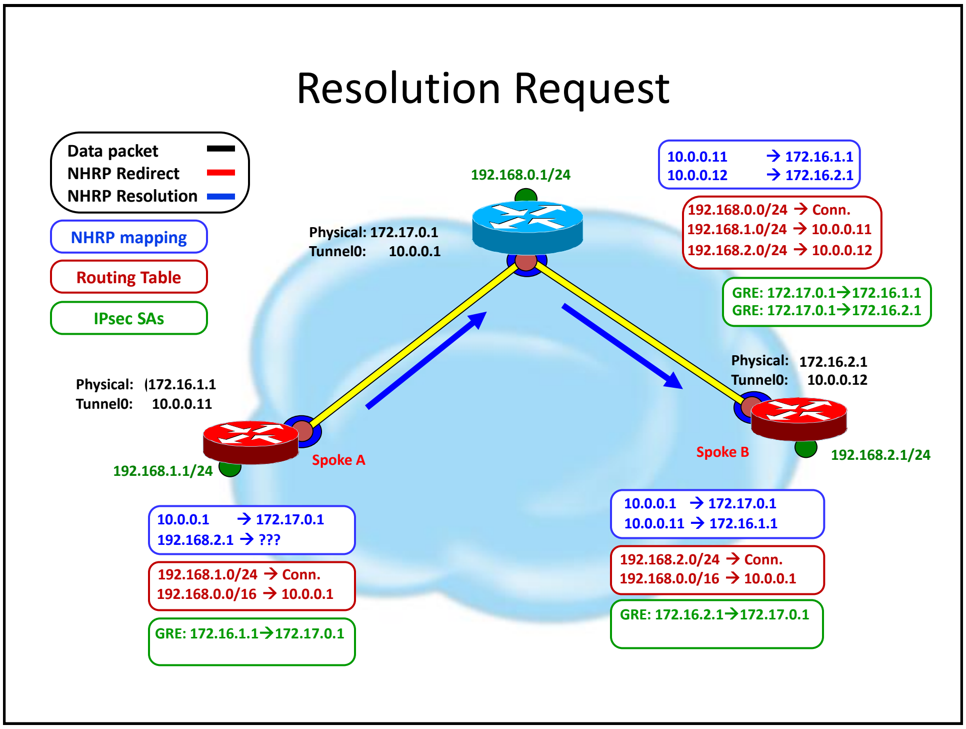
GRE: 172.16.1.1→172.17.0.1

Spoke B:

10.0.0.1 → 172.17.0.1

192.168.2.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

GRE: 172.16.2.1→172.17.0.1



Resolution Request

Data packet ▬
NHRP Redirect ▬
NHRP Resolution ▬

NHRP mapping

Routing Table

IPsec SAs

10.0.0.11 → 172.16.1.1
10.0.0.12 → 172.16.2.1

192.168.0.0/24 → Conn.
192.168.1.0/24 → 10.0.0.11
192.168.2.0/24 → 10.0.0.12

GRE: 172.17.0.1→172.16.1.1
GRE: 172.17.0.1→172.16.2.1

10.0.0.1 → 172.17.0.1
192.168.2.1 → ???

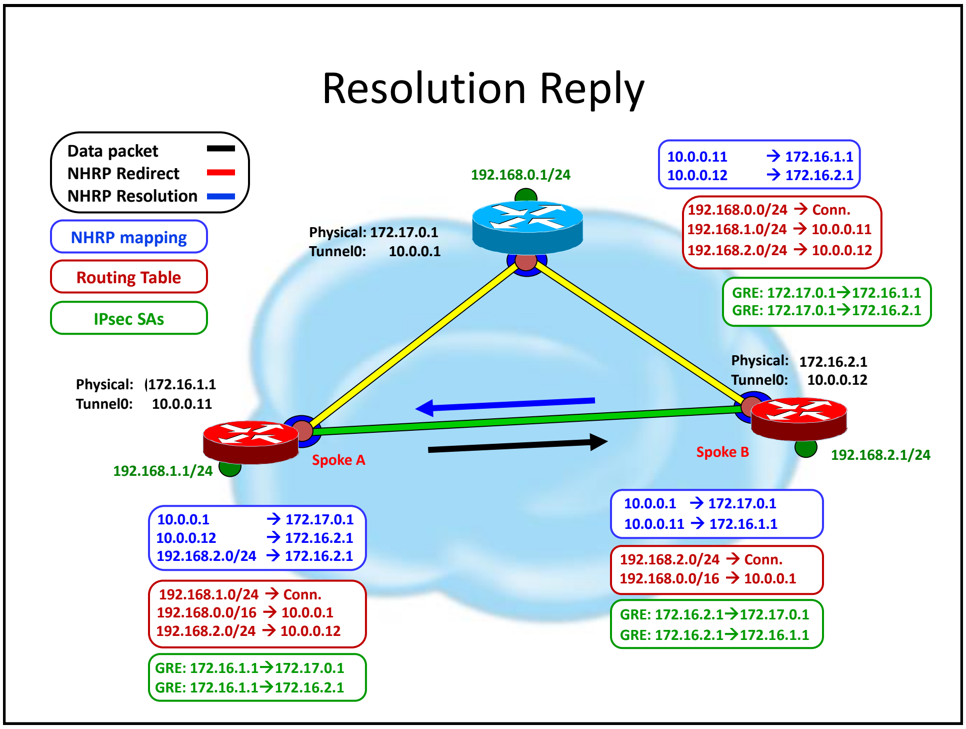
192.168.1.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

GRE: 172.16.1.1→172.17.0.1

10.0.0.1 → 172.17.0.1
10.0.0.11 → 172.16.1.1

192.168.2.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

GRE: 172.16.2.1→172.17.0.1



Resolution Reply

Data packet ▬
NHRP Redirect ▬
NHRP Resolution ▬

NHRP mapping

Routing Table

IPsec SAs

10.0.0.11 → 172.16.1.1
10.0.0.12 → 172.16.2.1

192.168.0.0/24 → Conn.
192.168.1.0/24 → 10.0.0.11
192.168.2.0/24 → 10.0.0.12

GRE: 172.17.0.1→172.16.1.1
GRE: 172.17.0.1→172.16.2.1

10.0.0.1 → 172.17.0.1
10.0.0.12 → 172.16.2.1
192.168.2.0/24 → 172.16.2.1

192.168.1.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1
192.168.2.0/24 → 10.0.0.12

GRE: 172.16.1.1→172.17.0.1
GRE: 172.16.1.1→172.16.2.1

10.0.0.1 → 172.17.0.1
10.0.0.11 → 172.16.1.1

192.168.2.0/24 → Conn.
192.168.0.0/16 → 10.0.0.1

GRE: 172.16.2.1→172.17.0.1
GRE: 172.16.2.1→172.16.1.1

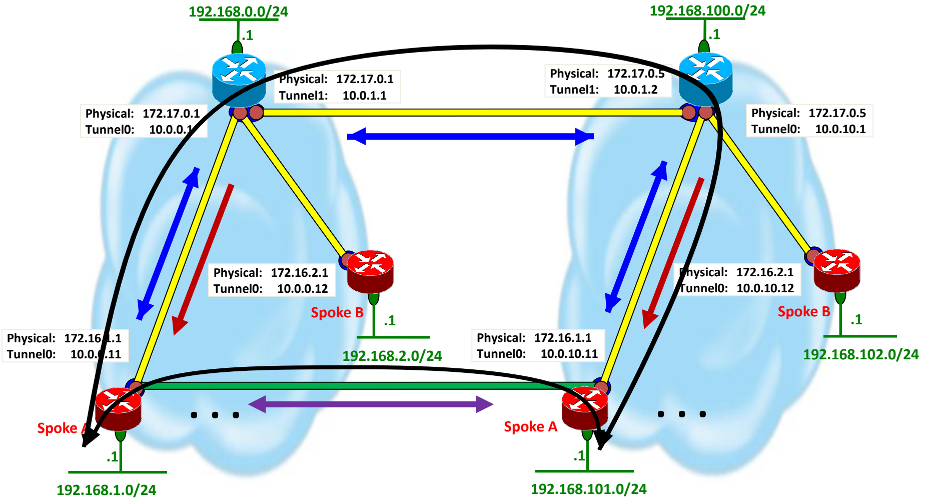
Dynamic Mappings

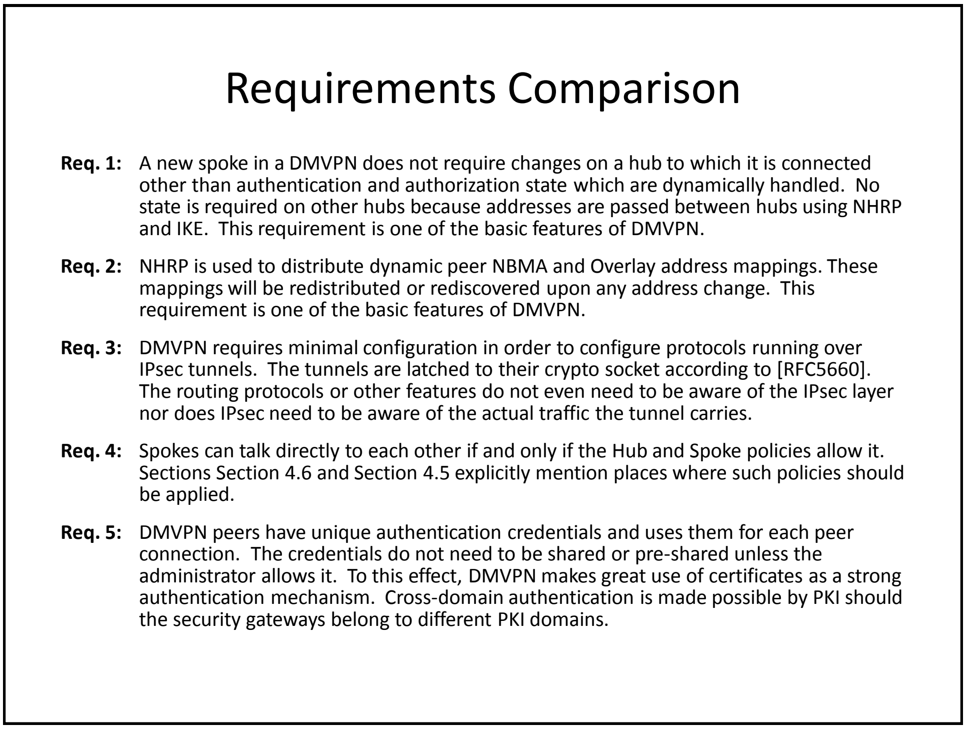
Refresh or Remove

- Dynamic NHRP mapping entries have finite lifetime
  - Controlled by NHRP on source of mapping (spoke)
- Refresh entries prior to expiration
  - Send another Resolution request and reply
    - Resolution request/reply sent over direct tunnel
- If NHRP mapping entry expires it is removed
  - If using IPsec and last entry using NBMA address
    - Trigger IPsec to remove IPsec and IKE SAs
- If IPsec discovers problem with tunnel
  - Example: Crypto DPD keepalives
    - IPsec clears IPsec and IKE SAs
    - Trigger NHRP to clear mappings

Cross Domain Support

Data packet ▬
NHRP Redirect 🟥
NHRP Res. Req. 🟦
NHRP Res. Reply 🟪





Requirements Comparison

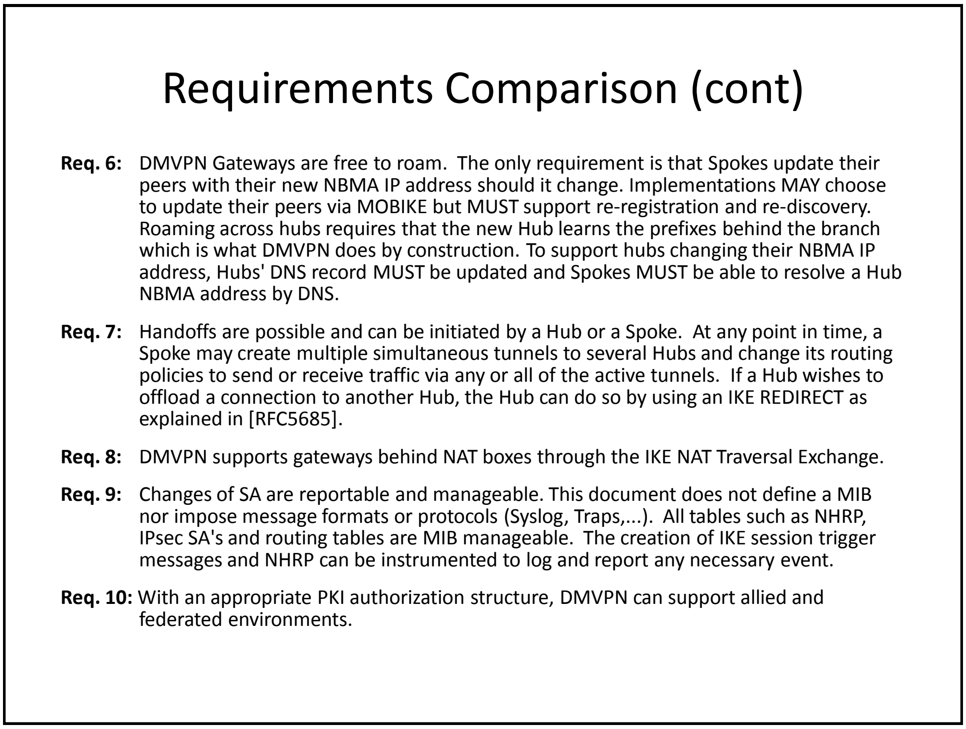
Req. 1: A new spoke in a DMVPN does not require changes on a hub to which it is connected other than authentication and authorization state which are dynamically handled. No state is required on other hubs because addresses are passed between hubs using NHRP and IKE. This requirement is one of the basic features of DMVPN.

Req. 2: NHRP is used to distribute dynamic peer NBMA and Overlay address mappings. These mappings will be redistributed or rediscovered upon any address change. This requirement is one of the basic features of DMVPN.

Req. 3: DMVPN requires minimal configuration in order to configure protocols running over IPsec tunnels. The tunnels are latched to their crypto socket according to [RFC5660]. The routing protocols or other features do not even need to be aware of the IPsec layer nor does IPsec need to be aware of the actual traffic the tunnel carries.

Req. 4: Spokes can talk directly to each other if and only if the Hub and Spoke policies allow it. Sections Section 4.6 and Section 4.5 explicitly mention places where such policies should be applied.

Req. 5: DMVPN peers have unique authentication credentials and uses them for each peer connection. The credentials do not need to be shared or pre-shared unless the administrator allows it. To this effect, DMVPN makes great use of certificates as a strong authentication mechanism. Cross-domain authentication is made possible by PKI should the security gateways belong to different PKI domains.



Requirements Comparison (cont)

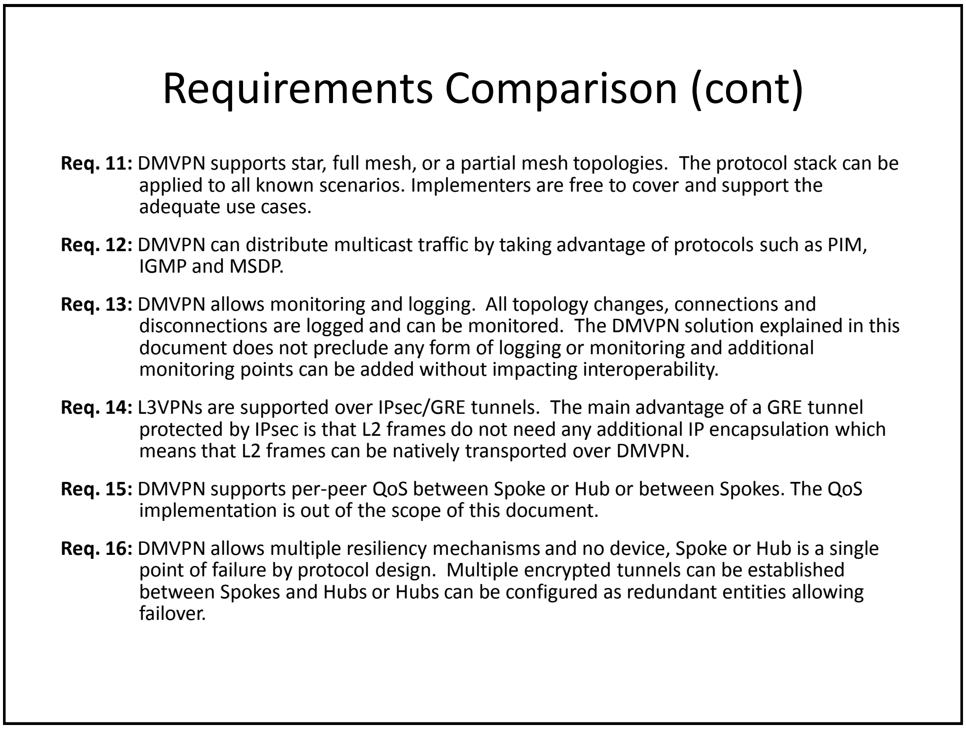
Req. 6: DMVPN Gateways are free to roam. The only requirement is that Spokes update their peers with their new NBMA IP address should it change. Implementations MAY choose to update their peers via MOBIKE but MUST support re-registration and re-discovery. Roaming across hubs requires that the new Hub learns the prefixes behind the branch which is what DMVPN does by construction. To support hubs changing their NBMA IP address, Hubs' DNS record MUST be updated and Spokes MUST be able to resolve a Hub NBMA address by DNS.

Req. 7: Handoffs are possible and can be initiated by a Hub or a Spoke. At any point in time, a Spoke may create multiple simultaneous tunnels to several Hubs and change its routing policies to send or receive traffic via any or all of the active tunnels. If a Hub wishes to offload a connection to another Hub, the Hub can do so by using an IKE REDIRECT as explained in [RFC5685].

Req. 8: DMVPN supports gateways behind NAT boxes through the IKE NAT Traversal Exchange.

Req. 9: Changes of SA are reportable and manageable. This document does not define a MIB nor impose message formats or protocols (Syslog, Traps,...). All tables such as NHRP, IPsec SA's and routing tables are MIB manageable. The creation of IKE session trigger messages and NHRP can be instrumented to log and report any necessary event.

Req. 10: With an appropriate PKI authorization structure, DMVPN can support allied and federated environments.



Requirements Comparison (cont)

Req. 11: DMVPN supports star, full mesh, or a partial mesh topologies. The protocol stack can be applied to all known scenarios. Implementers are free to cover and support the adequate use cases.

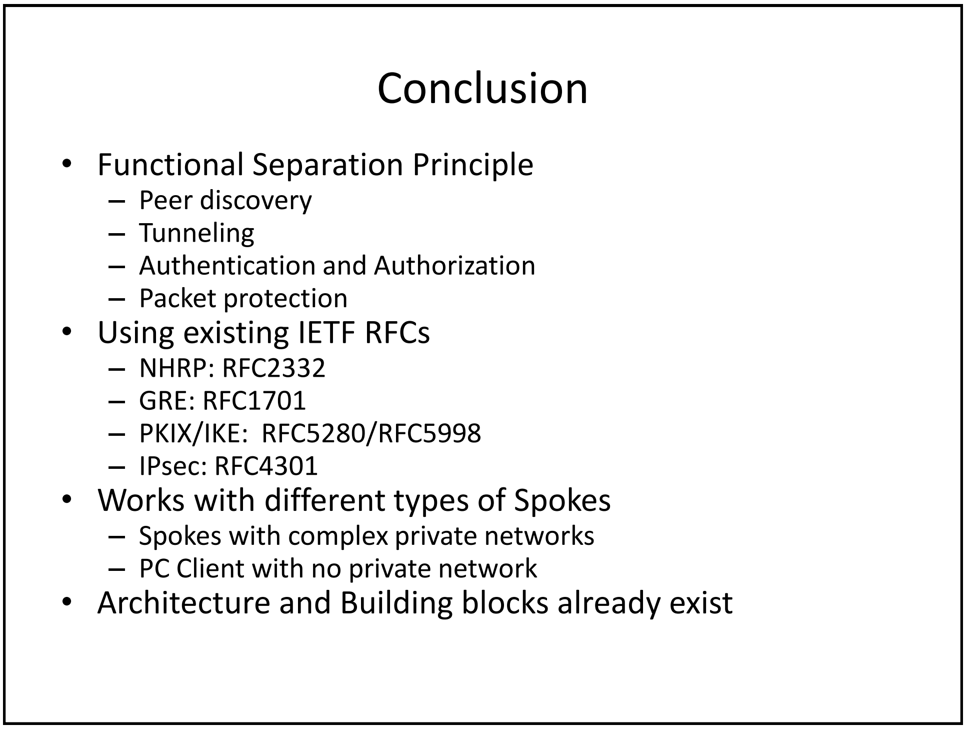
Req. 12: DMVPN can distribute multicast traffic by taking advantage of protocols such as PIM, IGMP and MSDP.

Req. 13: DMVPN allows monitoring and logging. All topology changes, connections and disconnections are logged and can be monitored. The DMVPN solution explained in this document does not preclude any form of logging or monitoring and additional monitoring points can be added without impacting interoperability.

Req. 14: L3VPNs are supported over IPsec/GRE tunnels. The main advantage of a GRE tunnel protected by IPsec is that L2 frames do not need any additional IP encapsulation which means that L2 frames can be natively transported over DMVPN.

Req. 15: DMVPN supports per-peer QoS between Spoke or Hub or between Spokes. The QoS implementation is out of the scope of this document.

Req. 16: DMVPN allows multiple resiliency mechanisms and no device, Spoke or Hub is a single point of failure by protocol design. Multiple encrypted tunnels can be established between Spokes and Hubs or Hubs can be configured as redundant entities allowing failover.



Conclusion

- Functional Separation Principle
  - Peer discovery
  - Tunneling
  - Authentication and Authorization
  - Packet protection
- Using existing IETF RFCs
  - NHRP: RFC2332
  - GRE: RFC1701
  - PKIX/IKE: RFC5280/RFC5998
  - IPsec: RFC4301
- Works with different types of Spokes
  - Spokes with complex private networks
  - PC Client with no private network
- Architecture and Building blocks already exist

End