

Contents

Configuration Model of StoneWork	8
Table of Contents	8
stonework-root.proto	14
Root	14
Root.LinuxConfig	15
Root.NetallocConfig	15
Root.VppConfig	15
abx/abx.proto	16
ABX	16
ABX.AttachedInterface	16
ligato/generic/manager.proto	16
ConfigItem	16
ConfigItem.LabelsEntry	16
Data	17
DumpStateRequest	17
DumpStateResponse	17
GetConfigRequest	17
GetConfigResponse	17
Item	17
Item.ID	17
ItemStatus	18
Notification	18
SetConfigRequest	18
SetConfigResponse	18
StateItem	18
StateItem.MetadataEntry	18
SubscribeRequest	18
SubscribeResponse	18
Subscription	19
UpdateItem	19
UpdateItem.LabelsEntry	19
UpdateResult	19
UpdateResult.Operation	19
ManagerService	19
ligato/generic/model.proto	20
ModelDetail	20
ModelDetail.Option	20
ModelSpec	20

ligato/generic/options.proto	20
File-level Extensions	20
ligato/generic/meta.proto	20
KnownModelsRequest	20
KnownModelsResponse	21
ProtoFileDescriptorRequest	21
ProtoFileDescriptorResponse	21
MetaService	21
ligato/kvscheduler/value_status.proto	22
BaseValueStatus	22
ValueStatus	22
TxnOperation	22
ValueState	22
ligato/govppmux/metrics.proto	23
Metrics	23
ligato/netalloc/netalloc.proto	23
ConfigData	23
IPAllocation	23
IPAddressForm	24
IPAddressSource	24
ligato/vpp/vpp.proto	24
ConfigData	24
Notification	25
Stats	25
ligato/vpp/srv6/srv6.proto	25
LocalSID	25
LocalSID.End	26
LocalSID.EndAD	26
LocalSID.EndDT4	26
LocalSID.EndDT6	27
LocalSID.EndDX2	27
LocalSID.EndDX4	27
LocalSID.EndDX6	27
LocalSID.EndT	28
LocalSID.EndX	28
Policy	28
Policy.SegmentList	28
SRv6Global	29
Steering	29
Steering.L2Traffic	29

Steering.L3Traffic	29
ligato/vpp/abf/abf.proto	29
ABF	29
ABF.AttachedInterface	30
ABF.ForwardingPath	30
ligato/vpp/dns/dns.proto	30
DNSCache	30
ligato/vpp/punt/punt.proto	30
Exception	30
IPRedirect	31
Reason	31
ToHost	31
L3Protocol	31
L4Protocol	32
ligato/vpp/l3/l3.proto	32
DHCPPProxy	32
DHCPPProxy.DHCPsServer	32
IPScanNeighbor	32
ProxyARP	32
ProxyARP.Interface	33
ProxyARP.Range	33
IPScanNeighbor.Mode	33
ligato/vpp/l3/arp.proto	33
ARPEntree	33
ligato/vpp/l3/vrrp.proto	33
VRRPEntree	33
ligato/vpp/l3/l3xc.proto	34
L3XConnect	34
L3XConnect.Path	34
L3XConnect.Protocol	34
ligato/vpp/l3/route.proto	34
Route	34
Route.RouteType	35
ligato/vpp/l3/vrf.proto	35
VrfTable	35
VrfTable.FlowHashSettings	35
VrfTable.Protocol	36
ligato/vpp/l3/teib.proto	36
TeibEntry	36
ligato/vpp/l2/bridge__domain.proto	36

BridgeDomain	36
BridgeDomain.ArpTerminationEntry	36
BridgeDomain.Interface	36
ligato/vpp/l2/xconnect.proto	37
XConnectPair	37
ligato/vpp/l2/fib.proto	37
FIBEntry	37
FIBEntry.Action	37
ligato/vpp/ipfix/ipfix.proto	37
IPFIX	37
IPFIX.Collector	37
ligato/vpp/ipfix/flowprobe.proto	38
FlowProbeFeature	38
FlowProbeParams	38
ligato/vpp/ipsec/ipsec.proto	38
SecurityAssociation	38
SecurityPolicy	38
SecurityPolicyDatabase	39
SecurityPolicyDatabase.Interface	39
SecurityPolicyDatabase.PolicyEntry	39
TunnelProtection	40
CryptoAlg	40
IntegAlg	40
SecurityAssociation.IPSecProtocol	40
SecurityPolicy.Action	41
SecurityPolicyDatabase.PolicyEntry.Action	41
ligato/vpp/acl/acl.proto	41
ACL	41
ACL.Interfaces	41
ACL.Rule	41
ACL.Rule.IpRule	41
ACL.Rule.IpRule.Icmp	42
ACL.Rule.IpRule.Icmp.Range	42
ACL.Rule.IpRule.Ip	42
ACL.Rule.IpRule.PortRange	42
ACL.Rule.IpRule.Tcp	42
ACL.Rule.IpRule.Udp	43
ACL.Rule.MacIpRule	43
ACL.Rule.Action	43
ligato/vpp/stn/stn.proto	43

Rule	43
ligato/vpp/interfaces/interface.proto	43
AfpacketLink	44
BondLink	44
BondLink.BondedInterface	44
GreLink	44
GtpuLink	45
IPIPLink	45
IPSecLink	45
Interface	45
Interface.IP6ND	46
Interface.RxMode	46
Interface.RxPlacement	47
Interface.Unnumbered	47
MemifLink	47
RDMALink	47
SubInterface	47
TapLink	48
VmxNet3Link	48
VxlanLink	48
VxlanLink.Gpe	49
WireguardLink	49
BondLink.LoadBalance	49
BondLink.Mode	49
GreLink.Type	49
GtpuLink.NextNode	50
IPIPLink.Mode	50
Interface.RxMode.Type	50
Interface.Type	50
MemifLink.MemifMode	50
RDMALink.Mode	51
SubInterface.TagRewriteOptions	51
VxlanLink.Gpe.Protocol	51
ligato/vpp/interfaces/dhcp.proto	51
DHCPLease	51
ligato/vpp/interfaces/state.proto	52
InterfaceNotification	52
InterfaceState	52
InterfaceState.Statistics	52
InterfaceStats	52

InterfaceStats.CombinedCounter	53
InterfaceNotification.NotifType	53
InterfaceState.Duplex	53
InterfaceState.Status	53
ligato/vpp/interfaces/span.proto	53
Span	53
Span.Direction	54
ligato/vpp/wireguard/wireguard.proto	54
Peer	54
ligato/vpp/nat/nat.proto	54
DNat44	54
DNat44.IdentityMapping	54
DNat44.StaticMapping	54
DNat44.StaticMapping.LocalIP	55
Nat44AddressPool	55
Nat44Global	55
Nat44Global.Address	56
Nat44Global.Interface	56
Nat44Interface	56
VirtualReassembly	57
DNat44.Protocol	57
DNat44.StaticMapping.TwiceNatMode	57
ligato/linux/punt/punt.proto	57
PortBased	57
Proxy	57
SocketBased	58
PortBased.L3Protocol	58
PortBased.L4Protocol	58
ligato/linux/linux.proto	58
ConfigData	58
Notification	58
ligato/linux/l3/arp.proto	58
ARPEntary	59
ligato/linux/l3/route.proto	59
Route	59
Route.Scope	59
ligato/linux/iptables/iptables.proto	59
RuleChain	59
RuleChain.ChainType	59
RuleChain.Policy	60

RuleChain.Protocol	60
RuleChain.Table	60
ligato/linux/interfaces/interface.proto	60
Interface	60
TapLink	61
VethLink	61
VrfDevLink	61
Interface.Type	61
VethLink.ChecksumOffloading	62
ligato/linux/interfaces/state.proto	62
InterfaceNotification	62
InterfaceState	62
InterfaceState.Statistics	63
InterfaceNotification.NotifType	63
InterfaceState.Status	63
ligato/linux/namespace/namespace.proto	63
NetNamespace	63
NetNamespace.ReferenceType	63
ligato/configurator/statspoller.proto	64
PollStatsRequest	64
PollStatsResponse	64
Stats	64
StatsPollerService	64
ligato/configurator/configurator.proto	64
Config	64
DeleteRequest	65
DeleteResponse	65
DumpRequest	65
DumpResponse	65
GetRequest	65
GetResponse	65
Notification	65
NotifyRequest	65
NotifyResponse	65
UpdateRequest	66
UpdateResponse	66
ConfiguratorService	66
bfd/bfd.proto	66
BFD	66
BFDEvent	66

WatchBFDEventsRequest	67
BFDEvent.SessionState	67
BFDWatcher	67
nat64/nat64.proto	67
Nat64AddressPool	67
Nat64IPv6Prefix	67
Nat64Interface	68
Nat64StaticBIB	68
Nat64Interface.Type	68
Nat64StaticBIB.Protocol	68
Scalar Value Types	68

Configuration Model of StoneWork

Table of Contents

- *stonework-root.proto*
 - Root
 - Root.LinuxConfig
 - Root.NetallocConfig
 - Root.VppConfig
- *abx/abx.proto*
 - ABX
 - ABX.AttachedInterface
- *ligato/generic/manager.proto*
 - ConfigItem
 - ConfigItem.LabelsEntry
 - Data
 - DumpStateRequest
 - DumpStateResponse
 - GetConfigRequest
 - GetConfigResponse
 - Item
 - Item.ID
 - ItemStatus
 - Notification
 - SetConfigRequest
 - SetConfigResponse
 - StateItem
 - StateItem.MetadataEntry
 - SubscribeRequest
 - SubscribeResponse
 - Subscription
 - UpdateItem

- UpdateItem.LabelsEntry
 - UpdateResult
 - UpdateResult.Operation
 - ManagerService
- *ligato/generic/model.proto*
 - ModelDetail
 - ModelDetail.Option
 - ModelSpec
- *ligato/generic/options.proto*
 - File-level Extensions
- *ligato/generic/meta.proto*
 - KnownModelsRequest
 - KnownModelsResponse
 - ProtoFileDescriptorRequest
 - ProtoFileDescriptorResponse
 - MetaService
- *ligato/kvscheduler/value__status.proto*
 - BaseValueStatus
 - ValueStatus
 - TxnOperation
 - ValueState
- *ligato/govppmux/metrics.proto*
 - Metrics
- *ligato/netalloc/netalloc.proto*
 - ConfigData
 - IPAllocation
 - IPAddressForm
 - IPAddressSource
- *ligato/vpp/vpp.proto*
 - ConfigData
 - Notification
 - Stats
- *ligato/vpp/srv6/srv6.proto*
 - LocalSID
 - LocalSID.End
 - LocalSID.EndAD
 - LocalSID.EndDT4
 - LocalSID.EndDT6
 - LocalSID.EndDX2
 - LocalSID.EndDX4
 - LocalSID.EndDX6
 - LocalSID.EndT
 - LocalSID.EndX
 - Policy
 - Policy.SegmentList
 - SRv6Global
 - Steering

- Steering.L2Traffic
 - Steering.L3Traffic
- *ligato/vpp/abf/abf.proto*
 - ABF
 - ABF.AttachedInterface
 - ABF.ForwardingPath
- *ligato/vpp/dns/dns.proto*
 - DNSCache
- *ligato/vpp/punt/punt.proto*
 - Exception
 - IPRedirect
 - Reason
 - ToHost
 - L3Protocol
 - L4Protocol
- *ligato/vpp/l3/l3.proto*
 - DHCPProxy
 - DHCPProxy.DHCPServer
 - IPScanNeighbor
 - ProxyARP
 - ProxyARP.Interface
 - ProxyARP.Range
 - IPScanNeighbor.Mode
- *ligato/vpp/l3/arp.proto*
 - ARPEntry
- *ligato/vpp/l3/vrrp.proto*
 - VRRPEntry
- *ligato/vpp/l3/l3xc.proto*
 - L3XConnect
 - L3XConnect.Path
 - L3XConnect.Protocol
- *ligato/vpp/l3/route.proto*
 - Route
 - Route.RouteType
- *ligato/vpp/l3/vrf.proto*
 - VrfTable
 - VrfTable.FlowHashSettings
 - VrfTable.Protocol
- *ligato/vpp/l3/teib.proto*
 - TeibEntry
- *ligato/vpp/l2/bridge_domain.proto*
 - BridgeDomain
 - BridgeDomain.ArpTerminationEntry

- BridgeDomain.Interface
- *ligato/vpp/l2/xconnect.proto*
 - XConnectPair
- *ligato/vpp/l2/fib.proto*
 - FIBEntry
 - FIBEntry.Action
- *ligato/vpp/ipfix/ipfix.proto*
 - IPFIX
 - IPFIX.Collector
- *ligato/vpp/ipfix/flowprobe.proto*
 - FlowProbeFeature
 - FlowProbeParams
- *ligato/vpp/ipsec/ipsec.proto*
 - SecurityAssociation
 - SecurityPolicy
 - SecurityPolicyDatabase
 - SecurityPolicyDatabase.Interface
 - SecurityPolicyDatabase.PolicyEntry
 - TunnelProtection
 - CryptoAlg
 - IntegAlg
 - SecurityAssociation.IPSecProtocol
 - SecurityPolicy.Action
 - SecurityPolicyDatabase.PolicyEntry.Action
- *ligato/vpp/acl/acl.proto*
 - ACL
 - ACL.Interfaces
 - ACL.Rule
 - ACL.Rule.IpRule
 - ACL.Rule.IpRule.Icmp
 - ACL.Rule.IpRule.Icmp.Range
 - ACL.Rule.IpRule.Ip
 - ACL.Rule.IpRule.PortRange
 - ACL.Rule.IpRule.Tcp
 - ACL.Rule.IpRule.Udp
 - ACL.Rule.MacIpRule
 - ACL.Rule.Action
- *ligato/vpp/stn/stn.proto*
 - Rule
- *ligato/vpp/interfaces/interface.proto*
 - AfpacketLink
 - BondLink
 - BondLink.BondedInterface
 - GreLink

- GtpuLink
- IPIPLink
- IPSecLink
- Interface
- Interface.IP6ND
- Interface.RxMode
- Interface.RxPlacement
- Interface.Unnumbered
- MemifLink
- RDMALink
- SubInterface
- TapLink
- VmxNet3Link
- VxlanLink
- VxlanLink.Gpe
- WireguardLink
- BondLink.LoadBalance
- BondLink.Mode
- GreLink.Type
- GtpuLink.NextNode
- IPIPLink.Mode
- Interface.RxMode.Type
- Interface.Type
- MemifLink.MemifMode
- RDMALink.Mode
- SubInterface.TagRewriteOptions
- VxlanLink.Gpe.Protocol
- *ligato/vpp/interfaces/dhcp.proto*
 - DHCPLease
- *ligato/vpp/interfaces/state.proto*
 - InterfaceNotification
 - InterfaceState
 - InterfaceState.Statistics
 - InterfaceStats
 - InterfaceStats.CombinedCounter
 - InterfaceNotification.NotifType
 - InterfaceState.Duplex
 - InterfaceState.Status
- *ligato/vpp/interfaces/span.proto*
 - Span
 - Span.Direction
- *ligato/vpp/wireguard/wireguard.proto*
 - Peer
- *ligato/vpp/nat/nat.proto*

- DNat44
- DNat44.IdentityMapping
- DNat44.StaticMapping
- DNat44.StaticMapping.LocalIP
- Nat44AddressPool
- Nat44Global
- Nat44Global.Address
- Nat44Global.Interface
- Nat44Interface
- VirtualReassembly
- DNat44.Protocol
- DNat44.StaticMapping.TwiceNatMode
- *ligato/linux/punt/punt.proto*
 - PortBased
 - Proxy
 - SocketBased
 - PortBased.L3Protocol
 - PortBased.L4Protocol
- *ligato/linux/linux.proto*
 - ConfigData
 - Notification
- *ligato/linux/l3/arp.proto*
 - ARPEntry
- *ligato/linux/l3/route.proto*
 - Route
 - Route.Scope
- *ligato/linux/iptables/iptables.proto*
 - RuleChain
 - RuleChain.ChainType
 - RuleChain.Policy
 - RuleChain.Protocol
 - RuleChain.Table
- *ligato/linux/interfaces/interface.proto*
 - Interface
 - TapLink
 - VethLink
 - VrfDevLink
 - Interface.Type
 - VethLink.ChecksumOffloading
- *ligato/linux/interfaces/state.proto*
 - InterfaceNotification
 - InterfaceState
 - InterfaceState.Statistics
 - InterfaceNotification.NotifType

- InterfaceState.Status
- *ligato/linux/namespace/namespace.proto*
 - NetNamespace
 - NetNamespace.ReferenceType
- *ligato/configurator/statspoller.proto*
 - PollStatsRequest
 - PollStatsResponse
 - Stats
 - StatsPollerService
- *ligato/configurator/configurator.proto*
 - Config
 - DeleteRequest
 - DeleteResponse
 - DumpRequest
 - DumpResponse
 - GetRequest
 - GetResponse
 - Notification
 - NotifyRequest
 - NotifyResponse
 - UpdateRequest
 - UpdateResponse
 - ConfiguratorService
- *bfd/bfd.proto*
 - BFD
 - BFDEvent
 - WatchBFDEventsRequest
 - BFDEvent.SessionState
 - BFDWatcher
- *nat64/nat64.proto*
 - Nat64AddressPool
 - Nat64IPv6Prefix
 - Nat64Interface
 - Nat64StaticBIB
 - Nat64Interface.Type
 - Nat64StaticBIB.Protocol

stonework-root.proto

Proto file with the configuration model of StoneWork.

Root

Configuration root wrapping all models supported by StoneWork.

Field	Type	Label	Description
linuxConfig	Root.LinuxConfig		
netallocConfig	Root.NetallocConfig		
vppConfig	Root.VppConfig		

Root.LinuxConfig

Field	Type	Label	Description
RuleChain_list	ligato.linux.iptables.RuleChain	repeated	
arp_entries	ligato.linux.l3.ARPEntry	repeated	
interfaces	ligato.linux.interfaces.Interface	repeated	
routes	ligato.linux.l3.Route	repeated	

Root.NetallocConfig

Field	Type	Label	Description
ip_addresses	ligato.netalloc.IPAllocation	repeated	

Root.VppConfig

Field	Type	Label	Description
ABX_list	vpp.abx.ABX	repeated	
BFD_list	bfd.BFD	repeated	
DNSCache	ligato.vpp.dns.DNSCache		
Nat64AddressPool_list	nat64.Nat64AddressPool	repeated	
Nat64IPv6Prefix_list	nat64.Nat64IPv6Prefix	repeated	
Nat64Interface_list	nat64.Nat64Interface	repeated	
Nat64StaticBIB_list	nat64.Nat64StaticBIB	repeated	
Rule_list	ligato.vpp.stn.Rule	repeated	
VRRPEntry_list	ligato.vpp.l3.VRRPEntry	repeated	
abfs	ligato.vpp.abf.ABF	repeated	
acls	ligato.vpp.acl.ACL	repeated	
arps	ligato.vpp.l3.ARPEntry	repeated	
bridge_domains	ligato.vpp.l2.BridgeDomain	repeated	
dhcp_proxies	ligato.vpp.l3.DHCPProxy	repeated	
dnat44s	ligato.vpp.nat.DNat44	repeated	
fibs	ligato.vpp.l2.FIBEntry	repeated	
interfaces	ligato.vpp.interfaces.Interface	repeated	
ipfix_flowprobe_params	ligato.vpp.ipfix.FlowProbeParams		
ipfix_flowprobes	ligato.vpp.ipfix.FlowProbeFeature	repeated	
ipfix_global	ligato.vpp.ipfix.IPFIX		
ipscan_neighbor	ligato.vpp.l3.IPScanNeighbor		
ipsec_sas	ligato.vpp.ipsec.SecurityAssociation	repeated	
ipsec_spds	ligato.vpp.ipsec.SecurityPolicyDatabase	repeated	
ipsec_sps	ligato.vpp.ipsec.SecurityPolicy	repeated	
ipsec_tunnel_protections	ligato.vpp.ipsec.TunnelProtection	repeated	
l3xconnects	ligato.vpp.l3.L3XConnect	repeated	
nat44_global	ligato.vpp.nat.Nat44Global		
nat44_interfaces	ligato.vpp.nat.Nat44Interface	repeated	
nat44_pools	ligato.vpp.nat.Nat44AddressPool	repeated	
proxy_arp	ligato.vpp.l3.ProxyARP		
punt_exceptions	ligato.vpp.punt.Exception	repeated	

Field	Type	Label	Description
punt_ipredirects	ligato.vpp.punt.IPRedirect	repeated	
punt_tohosts	ligato.vpp.punt.ToHost	repeated	
routes	ligato.vpp.l3.Route	repeated	
spans	ligato.vpp.interfaces.Span	repeated	
srv6_global	ligato.vpp.srv6.SRv6Global		
srv6_localsids	ligato.vpp.srv6.LocalSID	repeated	
srv6_policies	ligato.vpp.srv6.Policy	repeated	
srv6_steerings	ligato.vpp.srv6.Steering	repeated	
teib_entries	ligato.vpp.l3.TeibEntry	repeated	
vrf	ligato.vpp.l3.VrfTable	repeated	
wg_peers	ligato.vpp.wireguard.Peer	repeated	
xconnect_pairs	ligato.vpp.l2.XConnectPair	repeated	

abx/abx.proto

ABX

ACL based xconnect

Field	Type	Label	Description
index	uint32		ABX index (unique identifier)
acl_name	string		Name of the associated access list
output_interface	string		Name of outgoing interface
dst_mac	string		Rewrite destination mac address
attached_interfaces	ABX.AttachedInterface	repeated	

ABX.AttachedInterface

List of interfaces attached to the ABX

Field	Type	Label	Description
input_interface	string		
priority	uint32		

ligato/generic/manager.proto

ConfigItem

Field	Type	Label	Description
item	Item		
status	ItemStatus		
labels	ConfigItem.LabelsEntry	repeated	

ConfigItem.LabelsEntry

Field	Type	Label	Description
key	string		
value	string		

Data

Data represents encoded data for an item.

Field	Type	Label	Description
any	<code>google.protobuf.Any</code>		

DumpStateRequest

Field	Type	Label	Description
ids	<code>Item.ID</code>	repeated	

DumpStateResponse

Field	Type	Label	Description
items	<code>StateItem</code>	repeated	

GetConfigRequest

Field	Type	Label	Description
ids	<code>Item.ID</code>	repeated	

GetConfigResponse

Field	Type	Label	Description
items	<code>ConfigItem</code>	repeated	

Item

Item represents single instance described by the Model.

Field	Type	Label	Description
id	<code>Item.ID</code>		
data	<code>Data</code>		

Item.ID

ID represents identifier for distinguishing items.

Field	Type	Label	Description
model	<code>string</code>		
name	<code>string</code>		

ItemStatus

Item status describes status of an item.

Field	Type	Label	Description
status	string		
message	string		

Notification

Field	Type	Label	Description
item	Item		
status	ItemStatus		

SetConfigRequest

Field	Type	Label	Description
updates	UpdateItem	repeated	
overwrite_all	bool		The overwrite_all can be set to true to overwrite all other configuration (this is also known as Full Resync)

SetConfigResponse

Field	Type	Label	Description
results	UpdateResult	repeated	

StateItem

Field	Type	Label	Description
item	Item		
metadata	StateItem.MetadataEntry	repeated	

StateItem.MetadataEntry

Field	Type	Label	Description
key	string		
value	string		

SubscribeRequest

Field	Type	Label	Description
subscriptions	Subscription	repeated	

SubscribeResponse

Field	Type	Label	Description
notifications	Notification	repeated	

Subscription

Field	Type	Label	Description
id	Item.ID		

UpdateItem

Field	Type	Label	Description
item	Item		The item describes item to be updated. For a delete operation set fields item.Data to nil.
labels	UpdateItem.LabelsEntry	repeated	The labels can be used to define user-defined labels for item.

UpdateItem.LabelsEntry

Field	Type	Label	Description
key	string		
value	string		

UpdateResult

Field	Type	Label	Description
id	Item.ID		
key	string		
op	UpdateResult.Operation		
status	ItemStatus		

UpdateResult.Operation

Name	Number	Description
UNSPECIFIED	0	
CREATE	1	
UPDATE	2	
DELETE	3	

ManagerService

ManagerService defines the RPC methods for managing config using generic model, allowing extending with custom models.

Method Name	Request Type	Response Type	Description
SetConfig	SetConfigRequest	SetConfigResponse	SetConfig is used to update desired configuration.
GetConfig	GetConfigRequest	GetConfigResponse	GetConfig is used to read the desired configuration.

Method Name	Request Type	Response Type	Description
DumpState	DumpStateRequest	DumpStateResponse	DumpState is used to retrieve the actual running state.
Subscribe	SubscribeRequest	SubscribeResponse stream	Subscribe is used for subscribing to events. Notifications are returned by streaming updates.

ligato/generic/model.proto

ModelDetail

ModelDetail represents info about model details.

Field	Type	Label	Description
spec proto_name	ModelSpec string		Spec is a specificaiton the model was registered with. ProtoName is a name of protobuf message representing the model.
options	ModelDetail.Option	repeated	

ModelDetail.Option

Field	Type	Label	Description
key	string		
values	string	repeated	

ModelSpec

ModelSpec defines a model specification to identify a model.

Field	Type	Label	Description
module	string		Module describes grouping for the model.
version	string		Version describes version of the model schema.
type	string		Type describes name of type described by this model.
class	string		Class describes purpose for the model.

ligato/generic/options.proto

File-level Extensions

Extension	Type	Base	Number	Description
model	ModelSpec	.google.protobuf.MessageOptions	50222	

ligato/generic/meta.proto

KnownModelsRequest

Field	Type	Label	Description
class	string		

KnownModelsResponse

Field	Type	Label	Description
known_models	ModelDetail	repeated	
active_modules	string	repeated	

ProtoFileDescriptorRequest

Field	Type	Label	Description
full_proto_file_name	string		full_proto_file_name is full name of proto file that is needed to identify it. It has the form “<proto package name (‘:’ replaced with ‘/’)>/” (i.e. for this proto model it is “ligato/generic/meta.proto”). If you are using rpc ProtoFileDescriptor for additional information retrieve for known models from rpc KnownModels call, you can use usually present ModelDetail’s generic.ModelDetail_Option for key “protoFile” that is containing full proto file name in correct format.

ProtoFileDescriptorResponse

Field	Type	Label	Description
file_descriptor	google.protobuf.FileDescriptorProto		file_descriptor is proto message representing proto file descriptor
file_import_descriptors	google.protobuf.FileDescriptorSet		file_import_descriptors is set of file descriptors that the file_descriptor is using as import. This is needed when converting file descriptor proto to protorelect.FileDescriptor (using “google.golang.org/protobuf/reflect/protodesc”.NewFile(...))

MetaService

MetaService defines the RPC methods for managing generic models.

Method Name	Request Type	Response Type	Description
KnownModels	KnownModelsRequest	KnownModelsResponse	KnownModels returns information about service capabilities including list of models supported by the server.
ProtoFileDescriptor	ProtoFileDescriptorRequest	ProtoFileDescriptorResponse	ProtoFileDescriptor returns proto file descriptor for proto file identified by full name. The proto file descriptor is in form of proto messages (file descriptor proto and proto of its imports) so there are needed additional steps to join them into protorelect.FileDescriptor (“google.golang.org/protobuf/reflect/protodesc”.NewFile(...))

This rpc can be used together with knownModels rpc to retrieve additional model information. Message descriptor can be retrieved from file descriptor corresponding to knownModel message and used with proto reflecting to get all kinds of information about the known model.

Due to nature of data retrieval, it is expected that at least one message from that proto file is registered as known model.

ligato/kvscheduler/value_status.proto

BaseValueStatus

Field	Type	Label	Description
value	ValueStatus		
derived_values	ValueStatus	repeated	

ValueStatus

Field	Type	Label	Description
key	string		
state	ValueState		
error	string		error returned by the last operation (none if empty string)
last_operation	TxnOperation		
details	string	repeated	- for invalid value, details is a list of invalid fields - for pending value, details is a list of missing dependencies (labels)

TxnOperation

Name	Number	Description
UNDEFINED	0	
VALIDATE	1	
CREATE	2	
UPDATE	3	
DELETE	4	

ValueState

Name	Number	Description
NONEXISTENT	0	ValueState_NONEXISTENT is assigned to value that was deleted or has never existed.
MISSING	1	ValueState_MISSING is assigned to NB value that was configured but refresh found it to be missing.
UNIMPLEMENTED	2	ValueState_UNIMPLEMENTED marks value received from NB that cannot be configured because there is no registered descriptor associated with it.
REMOVED	3	ValueState_REMOVED is assigned to NB value after it was removed or when it is being re-created. The state is only temporary: for re-create, the value transits to whatever state the following Create operation produces, and delete values are removed from the graph (go to the NONEXISTENT state) immediately after the notification about the state change is sent.
CONFIGURED	4	ValueState_CONFIGURED marks value defined by NB and successfully configured.
OBTAINED	5	ValueState_OBTAINED marks value not managed by NB, instead created automatically or externally in SB. The KVScheduler learns about the value either using Retrieve() or through a SB notification.

Name	Number	Description
DISCOVERED	6	ValueState_DISCOVERED marks NB value that was found (=retrieved) by refresh but not actually configured by the agent in this run.
PENDING	7	ValueState_PENDING represents (NB) value that cannot be configured yet due to missing dependencies.
INVALID	8	ValueState_INVALID represents (NB) value that will not be configured because it has a logically invalid content as declared by the Validate method of the associated descriptor. The corresponding error and the list of affected fields are stored in the structure available via
FAILED	9	ValueState_FAILED marks (NB) value for which the last executed operation returned an error. The error and the type of the operation which caused the error are stored in the structure available via
RETRYING	10	ValueState_RETRYING marks unsuccessfully applied (NB) value, for which, however, one or more attempts to fix the error by repeating the last operation are planned, and only if all the retries fail, the value will then transit to the FAILED state.

ligato/govppmux/metrics.proto

Metrics

Field	Type	Label	Description
channels_created	uint64		
channels_open	uint64		
requests_sent	uint64		
requests_done	uint64		
requests_fail	uint64		
request_replies	uint64		

ligato/netalloc/netalloc.proto

ConfigData

ConfigData wraps all configuration items exported by netalloc. TBD: MACs, VXLAN VNIs, memif IDs, etc.

Field	Type	Label	Description
ip_addresses	IPAllocation	repeated	

IPAllocation

IPAllocation represents a single allocated IP address.

To reference allocated address, instead of entering specific IP address for interface/route/ARP/..., use one of the following string templates prefixed with netalloc keyword “alloc” followed by colon: a) reference IP address allocated for an interface: “alloc:” b) when interface is given (e.g. when asked for IP from interface model), interface_name can be omitted: “alloc:” c) reference default gateway IP address assigned to an interface: “alloc://GW” d) when asking for GW IP for interface which is given, interface_name can be omitted: “alloc://GW”

Field	Type	Label	Description
network_name	string		NetworkName is some label assigned to the network where the IP address was assigned to the given interface. In theory, interface can have multiple IP addresses or there can be multiple address allocators and the network name allows to separate them. The network name is not allowed to contain forward slashes.
interface_name	string		InterfaceName is the logical VPP or Linux interface name for which the address is allocated.
address	string		Address is an IP address allocated to the interface inside the given network. If the address is specified without a mask, the all-ones mask (/32 for IPv4, /128 for IPv6) will be assumed.
gw	string		Gw is the address of the default gateway assigned to the interface in the given network. If the address is specified without a mask, then either: a) the mask of the

IPAddressForm

IPAddressForm can be used in descriptors whose models reference allocated IP addresses, to ask for a specific form in which the address should be applied.

Name	Number	Description
UNDEFINED_FORM	0	
ADDR_ONLY	1	ADDR_ONLY = apply address without mask, e.g. 192.168.2.5
ADDR_WITH_MASK	2	ADDR_WITH_MASK = apply address including the mask of the network, e.g. 192.168.2.5/24
ADDR_NET	3	ADDR_NET = apply network implied by the address, e.g. for 192.168.2.10/24 apply 192.168.2.0/24
SINGLE_ADDR_NET	4	SINGLE_ADDR_NET = apply address with an all-ones mask (i.e. /32 for IPv4, /128 for IPv6)

IPAddressSource

IPAddressSource can be used to remember the source of an IP address. (e.g. to distinguish allocated IP addresses from statically defined ones)

Name	Number	Description
UNDEFINED_SOURCE	0	
STATIC	1	STATIC is IP address statically assigned in the NB configuration.
FROM_DHCP	2	FROM_DHCP is set when IP address is obtained from DHCP.
ALLOC_REF	3	ALLOC_REF is a reference inside NB configuration to an allocated IP address.
EXISTING	4	EXISTING is set when IP address is assigned to (EXISTING) interface externally (i.e. by a different agent or manually by an administrator).

ligato/vpp/vpp.proto

ConfigData

ConfigData holds the entire VPP configuration.

Field	Type	Label	Description
interfaces	interfaces.Interface	repeated	
spans	interfaces.Span	repeated	
acls	acl.ACL	repeated	
abfs	abf.ABF	repeated	
bridge_domains	l2.BridgeDomain	repeated	
fibs	l2.FIBEntry	repeated	
xconnect_pairs	l2.XConnectPair	repeated	
routes	l3.Route	repeated	
arps	l3.ARPEntary	repeated	
proxy_arp	l3.ProxyARP		
ipscan_neighbor	l3.IPScanNeighbor		
vrf	l3.VrfTable	repeated	
l3xconnects	l3.L3XConnect	repeated	
dhcp_proxies	l3.DHCPProxy	repeated	
teib_entries	l3.TeibEntry	repeated	
nat44_global	nat.Nat44Global		
dnat44s	nat.DNat44	repeated	
nat44_interfaces	nat.Nat44Interface	repeated	
nat44_pools	nat.Nat44AddressPool	repeated	
ipsec_spds	ipsec.SecurityPolicyDatabase	repeated	
ipsec_sas	ipsec.SecurityAssociation	repeated	
ipsec_tunnel_protections	ipsec.TunnelProtection	repeated	
ipsec_sps	ipsec.SecurityPolicy	repeated	
punt_ipredirects	punt.IPRedirect	repeated	
punt_tohosts	punt.ToHost	repeated	
punt_exceptions	punt.Exception	repeated	
srv6_global	srv6.SRv6Global		
srv6_localsids	srv6.LocalSID	repeated	
srv6_policies	srv6.Policy	repeated	
srv6_steerings	srv6.Steering	repeated	
ipfix_global	ipfix.IPFIX		
ipfix_flowprobe_params	ipfix.FlowProbeParams		
ipfix_flowprobes	ipfix.FlowProbeFeature	repeated	
wg_peers	wireguard.Peer	repeated	
dns_cache	dns.DNSCache		

Notification

Field	Type	Label	Description
interface	interfaces.InterfaceNotification		

Stats

Field	Type	Label	Description
interface	interfaces.InterfaceStats		

ligato/vpp/srv6/srv6.proto

LocalSID

Field	Type	Label	Description
sid	string		segment id (IPv6 Address)
installation_vrf_id	uint32		ID of IPv6 VRF table where to install LocalSID routing components (LocalSids with End.AD function ignore this setting due to missing setting in the API. The End.AD functionality is separated from the SRv6 functionality and have no binary API. It has only the CLI API and that doesn't have the installation vrf id (in VPP API called FIB table) setting configurable.) Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)
base_end_function	LocalSID.End		
end_function_x	LocalSID.EndX		
end_function_t	LocalSID.EndT		
end_function_dx2	LocalSID.EndDX2		
end_function_dx4	LocalSID.EndDX4		
end_function_dx6	LocalSID.EndDX6		
end_function_dt4	LocalSID.EndDT4		
end_function_dt6	LocalSID.EndDT6		
end_function_ad	LocalSID.EndAD		

LocalSID.End

End function behavior of simple endpoint

Field	Type	Label	Description
psp	bool		use PSP (penultimate segment POP of the SRH) or by default use USP (Ultimate Segment Pop of the SRH)

LocalSID.EndAD

End function behavior of dynamic segment routing proxy endpoint

Field	Type	Label	Description
outgoing_interface	string		name of interface on segment routing proxy side sending data to segment routing unaware service
incoming_interface	string		name of interface on segment routing proxy side receiving data from segment routing unaware service
l3_service_address	string		IPv6/IPv4 address of L3 SR-unaware service (address type depends whether service is IPv4 or IPv6 service), in case of L2 service it must be empty

LocalSID.EndDT4

End function behavior of endpoint with decapsulation and specific IPv4 table lookup

Field	Type	Label	Description
vrf_id	uint32		vrf index of IPv4 table that should be used for lookup. vrf_index and fib_table_id should refer to the same routing table. VRF index refer to it from client side and FIB table id from VPP-internal side (index of memory allocated structure from pool)(source: https://wiki.fd.io/view/VPP/Per-feature_Notes). Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)

LocalSID.EndDT6

End function behavior of endpoint with decapsulation and specific IPv6 table lookup

Field	Type	Label	Description
vrf_id	uint32		vrf index of IPv6 table that should be used for lookup. vrf_index and fib_table_id should refer to the same routing table. VRF index refer to it from client side and FIB table id from VPP-internal side (index of memory allocated structure from pool)(source: https://wiki.fd.io/view/VPP/Per-feature_Notes). Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)

LocalSID.EndDX2

End function behavior of endpoint with decapsulation and Layer-2 cross-connect (or DX2 with egress VLAN rewrite when VLAN notzero - not supported this variant yet)

Field	Type	Label	Description
vlan_tag	uint32		Outgoing VLAN tag
outgoing_interface	string		name of cross-connected outgoing interface

LocalSID.EndDX4

End function behavior of endpoint with decapsulation and IPv4 cross-connect

Field	Type	Label	Description
outgoing_interface	string		name of cross-connected outgoing interface
next_hop	string		next hop address for cross-connected link

LocalSID.EndDX6

End function behavior of endpoint with decapsulation and IPv6 cross-connect

Field	Type	Label	Description
outgoing_interface	string		name of cross-connected outgoing interface
next_hop	string		next hop address for cross-connected link

LocalSID.EndT

End function behavior of endpoint with specific IPv6 table lookup

Field	Type	Label	Description
psp	bool		use PSP (penultimate segment POP of the SRH) or by default use USP (Ultimate Segment Pop of the SRH)
vrf_id	uint32		vrf index of IPv6 table that should be used for lookup. vrf_index and fib_table_id should refer to the same routing table. VRF index refer to it from client side and FIB table id from VPP-internal side (index of memory allocated structure from pool)(source: https://wiki.fd.io/view/VPP/Per-feature_Notes). Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)

LocalSID.EndX

End function behavior of endpoint with Layer-3 cross-connect (IPv6)

Field	Type	Label	Description
psp	bool		use PSP (penultimate segment POP of the SRH) or by default use USP (Ultimate Segment Pop of the SRH)
outgoing_interface	string		name of cross-connected outgoing interface
next_hop	string		IPv6 next hop address for cross-connected link

Policy

Model for SRv6 policy (policy without at least one policy segment is only cached in ligato and not written to VPP)

Field	Type	Label	Description
bsid	string		binding SID (IPv6 Address)
installation_vrf_id	uint32		ID of IPv6 VRF table where to install Policy routing components (for loadbalancing/spray are used VPP features that are using VRF table) Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)
srh_encapsulation	bool		are SR headers handled by encapsulation? (no means insertion of SR headers)
spray_behaviour	bool		spray(multicast) to all policy segments? (no means to use PolicySegment.weight to loadbalance traffic)
segment_lists	Policy.SegmentList	repeated	

Policy.SegmentList

Model for SRv6 Segment List

Field	Type	Label	Description
weight	uint32		used for loadbalancing in case of multiple policy segments in routing process (ignored in case of spray policies)
segments	string	repeated	list of sids creating one segmented road

SRv6Global

Global SRv6 config

Field	Type	Label	Description
encap_source_address	string		IPv6 source address for sr encapsulated packets

Steering

Model for steering traffic to SRv6 policy

Field	Type	Label	Description
name	string		globally unique steering identification (used in keys when is steering stored in key-value stores(i.e. ETCD))
policy_bsid	string		BSID identifier for policy to which we want to steer routing into (policyBSID and policyIndex are mutual exclusive)
policy_index	uint32		(vpp-internal)Index identifier for policy to which we want to steer routing into (policyBSID and policyIndex are mutual exclusive)
l2_traffic	Steering.L2Traffic		
l3_traffic	Steering.L3Traffic		

Steering.L2Traffic

L2 traffic that should be steered into SR policy

Field	Type	Label	Description
interface_name	string		name of interface with incoming traffic that should be steered to SR policy

Steering.L3Traffic

L3 traffic that should be steered into SR policy

Field	Type	Label	Description
installation_vrf_id	uint32		ID of IPv4/IPv6 VRF table where to install L3 Steering routing components (VRF table type (IPv4/IPv6) is decided by prefix_address value) Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)
prefix_address	string		IPv4/IPv6 prefix address(CIRD format) of traffic destination. All traffic with given destination will be steered to given SR policy

ligato/vpp/abf/abf.proto

ABF

ABF defines ACL based forwarding.

Field	Type	Label	Description
index	uint32		ABF index (unique identifier)
acl_name	string		Name of the associated access list
attached_interfaces	ABF.AttachedInterface	repeated	
forwarding_paths	ABF.ForwardingPath	repeated	

ABF.AttachedInterface

List of interfaces attached to the ABF

Field	Type	Label	Description
input_interface	string		
priority	uint32		
is_ipv6	bool		

ABF.ForwardingPath

List of forwarding paths added to the ABF policy (via)

Field	Type	Label	Description
next_hop_ip	string		
interface_name	string		
weight	uint32		
preference	uint32		
dvr	bool		

ligato/vpp/dns/dns.proto

DNSCache

DNSCache configuration models VPP's DNS cache server functionality. The main goal of this functionality is to cache DNS records and minimize external DNS traffic. The presence of this configuration enables the VPP DNS functionality and VPP start to acts as DNS cache Server. It responds on standard DNS port(53) to DNS requests. Removing of this configuration disables the VPP DNS functionality.

Field	Type	Label	Description
upstream_dns_servers	string	repeated	List of upstream DNS servers that are contacted by VPP when unknown domain name needs to be resolved. The results are cached and there should be no further upstream DNS server request for the same domain name until cached DNS record expiration.

ligato/vpp/punt/punt.proto

Exception

Exception allows specifying punt exceptions used for punting packets. The type of exception is defined by reason name.

Field	Type	Label	Description
reason	string		Name should contain reason name, e.g. ipsec4-spi-0 .

Field	Type	Label	Description
socket_path	string		SocketPath defines path to unix domain socket used for punt packets to the host. In dumps, it will actually contain the socket defined in VPP config under punt section.

IPRedirect

IPRedirect allows otherwise dropped packet which destination IP address matching some of the VPP addresses to redirect to the defined next hop address via the TX interface.

Field	Type	Label	Description
l3_protocol	L3Protocol		L3 protocol to be redirected
rx_interface	string		Receive interface name. Optional, only redirect traffic incoming from this interface
tx_interface	string		Transmit interface name
next_hop	string		Next hop IP where the traffic is redirected

Reason

Reason represents punt reason used in exceptions. List of known exceptions can be retrieved in VPP CLI with following command:

```
vpp# show punt reasons [0] ipsec4-spi-0 from:[ipsec ] [1] ipsec6-spi-0 from:[ipsec ] [2] ipsec4-spi-o-udp-0 from:[ipsec ] [3] ipsec4-no-such-tunnel from:[ipsec ] [4] ipsec6-no-such-tunnel from:[ipsec ] [5] VXLAN-GBP-no-such-v4-tunnel from:[vxlan-gbp ] [6] VXLAN-GBP-no-such-v6-tunnel from:[vxlan-gbp ]
```

Field	Type	Label	Description
name	string		Name contains reason name.

ToHost

ToHost allows otherwise dropped packet which destination IP address matching some of the VPP interface IP addresses to be punted to the host. L3 and L4 protocols can be used for filtering */

Field	Type	Label	Description
l3_protocol	L3Protocol		L3 destination protocol a packet has to match in order to be punted.
l4_protocol	L4Protocol		L4 destination protocol a packet has to match. Currently VPP only supports UDP.
port	uint32		Destination port
socket_path	string		SocketPath defines path to unix domain socket used for punt packets to the host. In dumps, it will actually contain the socket defined in VPP config under punt section.

L3Protocol

L3Protocol defines Layer 3 protocols.

Name	Number	Description
UNDEFINED_L3	0	
IPV4	4	
IPV6	6	
ALL	10	

L4Protocol

L4Protocol defines Layer 4 protocols.

Name	Number	Description
UNDEFINED_L4	0	
TCP	6	
UDP	17	

ligato/vpp/l3/l3.proto

DHCPProxy

DHCP Proxy

Field	Type	Label	Description
source_ip_address	string		
rx_vrf_id	uint32		
servers	DHCPProxy.DHCPServer	repeated	

DHCPProxy.DHCPServer

Field	Type	Label	Description
vrf_id	uint32		
ip_address	string		

IPScanNeighbor

IP Neighbour Config

Field	Type	Label	Description
mode	IPScanNeighbor.Mode		
scan_interval	uint32		
max_proc_time	uint32		
max_update	uint32		
scan_int_delay	uint32		
stale_threshold	uint32		

ProxyARP

ARP Proxy

Field	Type	Label	Description
interfaces	ProxyARP.Interface	repeated	List of interfaces proxy ARP is enabled for.
ranges	ProxyARP.Range	repeated	

ProxyARP.Interface

Field	Type	Label	Description
name	string		

ProxyARP.Range

Field	Type	Label	Description
first_ip_addr	string		
last_ip_addr	string		
vrf_id	uint32		

IPScanNeighbor.Mode

Name	Number	Description
DISABLED	0	
IPV4	1	
IPV6	2	
BOTH	3	

ligato/vpp/l3/arp.proto

ARPEntry

Field	Type	Label	Description
interface	string		
ip_address	string		
phys_address	string		
static	bool		

ligato/vpp/l3/vrrp.proto

VRRPEntry

VRRPEntry represents Virtual Router desired state.

Field	Type	Label	Description
interface	string		This field refers to logical interface name
vr_id	uint32		Should be > 0 and <= 255
priority	uint32		Priority defines which router becomes master. Should be > 0 and <= 255.
interval	uint32		VR advertisement interval in milliseconds, should be => 10 and <= 655350. (Later, in implemetation it is converted into centisecons, so precision may be lost).

Field	Type	Label	Description
preempt	bool		Controls whether a (starting or restarting) higher-priority Backup router preempts a lower-priority Master router.
accept	bool		Controls whether a virtual router in Master state will accept packets addressed to the address owner's IPvX address as its own if it is not the IPvX address owner.
unicast	bool		Unicast mode may be used to take advantage of newer token ring adapter implementations that support non-promiscuous reception for multiple unicast MAC addresses and to avoid both the multicast traffic and usage conflicts associated with the use of token ring functional addresses.
ip_addresses enabled	string bool	repeated	Ip address quantity should be > 0 and <= 255.

ligato/vpp/l3/l3xc.proto

L3XConnect

Field	Type	Label	Description
interface	string		
protocol	L3XConnect.Protocol		
paths	L3XConnect.Path	repeated	

L3XConnect.Path

Field	Type	Label	Description
outgoing_interface	string		
next_hop_addr	string		
weight	uint32		
preference	uint32		

L3XConnect.Protocol

Name	Number	Description
IPV4	0	
IPV6	1	

ligato/vpp/l3/route.proto

Route

Field	Type	Label	Description
type	Route.RouteType		
vrf_id	uint32		VRF identifier, field required for remote client. This value should be consistent with VRF ID in static route key. If it is not, value from key will be preferred and this field will be overridden. Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto)

Field	Type	Label	Description
dst_network	string		Destination network defined by IP address and prefix (format:
next_hop_addr	string		Next hop address.
outgoing_interface	string		Interface name of the outgoing interface.
weight	uint32		Weight is used for unequal cost load balancing.
preference	uint32		Preference defines path preference. Lower preference is preferred. Only paths with the best preference contribute to forwarding (a poor man's primary and backup).
via_vrf_id	uint32		Specifies VRF ID for the next hop lookup / recursive lookup

Route.RouteType

Name	Number	Description
INTRA_VRF	0	Forwarding is being done in the specified vrf_id only, or according to the specified outgoing interface.
INTER_VRF	1	Forwarding is being done by lookup into a different VRF, specified as via_vrf_id field. In case of these routes, the outgoing interface should not be specified. The next hop IP address does not have to be specified either, in that case VPP does full recursive lookup in the via_vrf_id VRF.
DROP	2	Drops the network communication designated for specific IP address.

ligato/vpp/l3/vrf.proto

VrfTable

Field	Type	Label	Description
id	uint32		ID is mandatory identification for VRF table. NOTE: do not confuse with fib index (shown by some VPP CLIs), which is VPP's internal offset in the vector of allocated tables.
protocol	VrfTable.Protocol		
label	string		Label is an optional description for the table. - maximum allowed length is 63 characters - included in the output from the VPP CLI command "show ip fib" - if undefined, then VPP will generate label using the template "-VRF:"
flow_hash_settings	VrfTable.FlowHashSettings		

VrfTable.FlowHashSettings

FlowHashSettings allows tuning of hash calculation of IP flows in the VRF table. This affects hash table size as well as the stickiness of flows by load-balancing. If not defined, default settings that are implicitly enabled are: - use_src_ip, use_dst_ip, use_src_port, use_dst_port, use_protocol

Field	Type	Label	Description
use_src_ip	bool		
use_dst_ip	bool		
use_src_port	bool		
use_dst_port	bool		
use_protocol	bool		

Field	Type	Label	Description
reverse	bool		
symmetric	bool		

VrfTable.Protocol

Protocol define IP protocol of VRF table.

Name	Number	Description
IPV4	0	
IPV6	1	

ligato/vpp/l3/teib.proto

TeibEntry

TeibEntry represents an tunnel endpoint information base entry.

Field	Type	Label	Description
interface	string		Interface references a tunnel interface this TEIB entry is linked to.
peer_addr	string		IP address of the peer.
next_hop_addr	string		Next hop IP address.
vrf_id	uint32		VRF ID used to reach the next hop.

ligato/vpp/l2/bridge_domain.proto

BridgeDomain

Field	Type	Label	Description
name	string		bridge domain name (can be any string)
flood	bool		enable/disable broadcast/multicast flooding in the BD
unknown_unicast_flood	bool		enable/disable unknown unicast flood in the BD
forward	bool		enable/disable forwarding on all interfaces in the BD
learn	bool		enable/disable learning on all interfaces in the BD
arp_termination	bool		enable/disable ARP termination in the BD
mac_age	uint32		MAC aging time in min, 0 for disabled aging
interfaces	BridgeDomain.Interface	repeated	list of interfaces
arp_termination_table	BridgeDomain.ArpTerminationEntry	repeated	list of ARP termination entries

BridgeDomain.ArpTerminationEntry

Field	Type	Label	Description
ip_address	string		IP address
phys_address	string		MAC address matching to the IP

BridgeDomain.Interface

Field	Type	Label	Description
name	string		interface name belonging to this bridge domain
bridged_virtual_interface	bool		true if this is a BVI interface
split_horizon_group	uint32		VXLANs in the same BD need the same non-zero SHG

ligato/vpp/l2/xconnect.proto

XConnectPair

Field	Type	Label	Description
receive_interface	string		
transmit_interface	string		

ligato/vpp/l2/fib.proto

FIBEntry

Field	Type	Label	Description
phys_address	string		unique destination MAC address
bridge_domain	string		name of bridge domain this FIB table entry belongs to
action	FIBEntry.Action		action to tke on matching frames
outgoing_interface	string		outgoing interface for matching frames
static_config	bool		true if this is a statically configured FIB entry
bridged_virtual_interface	bool		the MAC address is a bridge virtual interface MAC

FIBEntry.Action

Name	Number	Description
FORWARD	0	forward the matching frame
DROP	1	drop the matching frame

ligato/vpp/ipfix/ipfix.proto

IPFIX

IPFIX defines the IP Flow Information eXport (IPFIX) configuration.

Field	Type	Label	Description
collector	IPFIX.Collector		
source_address	string		
vrf_id	uint32		
path_mtu	uint32		
template_interval	uint32		

IPFIX.Collector

Field	Type	Label	Description
address	string		
port	uint32		

ligato/vpp/ipfix/flowprobe.proto

FlowProbeFeature

Field	Type	Label	Description
interface	string		
l2	bool		
ip4	bool		
ip6	bool		

FlowProbeParams

Field	Type	Label	Description
active_timer	uint32		
passive_timer	uint32		
record_l2	bool		
record_l3	bool		
record_l4	bool		

ligato/vpp/ipsec/ipsec.proto

SecurityAssociation

Security Association (SA)

Field	Type	Label	Description
index	uint32		Numerical security association index, serves as a unique identifier
spi	uint32		Security parameter index
protocol	SecurityAssociation.IPSecProtocol		
crypto_alg	CryptoAlg		Cryptographic algorithm for encryption
crypto_key	string		
crypto_salt	uint32		
integ_alg	IntegAlg		Cryptographic algorithm for authentication
integ_key	string		
use_esn	bool		Use extended sequence number
use_anti_replay	bool		Use anti replay
tunnel_src_addr	string		
tunnel_dst_addr	string		
enable_udp_encap	bool		Enable UDP encapsulation for NAT traversal
tunnel_src_port	uint32		
tunnel_dst_port	uint32		

SecurityPolicy

Field	Type	Label	Description
spd_index	uint32		Security policy database index
sa_index	uint32		Security association index
priority	int32		
is_outbound	bool		
remote_addr_start	string		
remote_addr_stop	string		
local_addr_start	string		
local_addr_stop	string		
protocol	uint32		
remote_port_start	uint32		
remote_port_stop	uint32		
local_port_start	uint32		
local_port_stop	uint32		
action	SecurityPolicy.Action		

SecurityPolicyDatabase

Security Policy Database (SPD)

Field	Type	Label	Description
index	uint32		Numerical security policy database index, serves as a unique identifier
interfaces	SecurityPolicyDatabase.Interfaces	Interfaces	List of interfaces belonging to this SPD
policy_entries	SecurityPolicyDatabase.PolicyEntry	PolicyEntry	Deprecated. List of policy entries belonging to this SPD. Deprecated and actually trying to use this will return an error. Use separate model for Security Policy (SP) defined below.

SecurityPolicyDatabase.Interface

Field	Type	Label	Description
name	string		Name of the related interface

SecurityPolicyDatabase.PolicyEntry

Field	Type	Label	Description
sa_index	uint32		Security association index
priority	int32		
is_outbound	bool		
remote_addr_start	string		
remote_addr_stop	string		
local_addr_start	string		
local_addr_stop	string		
protocol	uint32		
remote_port_start	uint32		
remote_port_stop	uint32		
local_port_start	uint32		
local_port_stop	uint32		
action	SecurityPolicyDatabase.PolicyEntry.Action		

TunnelProtection

TunnelProtection allows enabling IPSec tunnel protection on an existing interface (only IPIP tunnel interfaces are currently supported)

Field	Type	Label	Description
interface	string		Name of the interface to be protected with IPSec.
sa_out	uint32	repeated	Outbound security associations identified by SA index.
sa_in	uint32	repeated	Inbound security associations identified by SA index.
next_hop_addr	string		(Optional) Next hop IP address, used for multipoint tunnels.

CryptoAlg

Cryptographic algorithm for encryption

vpp/src/vnet/ipsec/ipsec_sa.h:22

Name	Number	Description
NONE_CRYPTO	0	
AES_CBC_128	1	
AES_CBC_192	2	
AES_CBC_256	3	
AES_CTR_128	4	
AES_CTR_192	5	
AES_CTR_256	6	
AES_GCM_128	7	
AES_GCM_192	8	
AES_GCM_256	9	
DES_CBC	10	
DES3_CBC	11	3DES_CBC

IntegAlg

Cryptographic algorithm for authentication

vpp/src/vnet/ipsec/ipsec_sa.h:44

Name	Number	Description
NONE_INTEG	0	
MD5_96	1	RFC2403
SHA1_96	2	RFC2404
SHA_256_96	3	draft-ietf-ipsec-ciph-sha-256-00
SHA_256_128	4	RFC4868
SHA_384_192	5	RFC4868
SHA_512_256	6	RFC4868

SecurityAssociation.IPSecProtocol

Name	Number	Description
AH	0	Authentication Header, provides a mechanism for authentication only
ESP	1	Encapsulating Security Payload is for data confidentiality and authentication

SecurityPolicy.Action

Name	Number	Description
BYPASS	0	Note: this particular action is unused in VPP
DISCARD	1	
RESOLVE	2	
PROTECT	3	

SecurityPolicyDatabase.PolicyEntry.Action

Name	Number	Description
BYPASS	0	Note: this particular action is unused in VPP
DISCARD	1	
RESOLVE	2	
PROTECT	3	

ligato/vpp/acl/acl.proto

ACL

ACL defines Access Control List.

Field	Type	Label	Description
name	string		The name of an access list. A device MAY restrict the length and value of this name, possibly spaces and special characters are not allowed.
rules	ACL.Rule	repeated	
interfaces	ACL.Interfaces		

ACL.Interfaces

The set of interfaces that has assigned this ACL on ingres or egress.

Field	Type	Label	Description
egress	string	repeated	
ingress	string	repeated	

ACL.Rule

List of access list entries (Rules). Each Access Control Rule has a list of match criteria and a list of actions. Access List entry that can define: - IPv4/IPv6 src ip prefix - src MAC address mask - src MAC address value - can be used only for static ACLs.

Field	Type	Label	Description
action	ACL.Rule.Action		
ip_rule	ACL.Rule.IpRule		
macip_rule	ACL.Rule.MacIpRule		

ACL.Rule.IpRule

Field	Type	Label	Description
ip	ACL.Rule.IpRule.Ip		
icmp	ACL.Rule.IpRule.Icmp		
tcp	ACL.Rule.IpRule.Tcp		
udp	ACL.Rule.IpRule.Udp		

ACL.Rule.IpRule.Icmp

Field	Type	Label	Description
icmpv6	bool		ICMPv6 flag, if false ICMPv4 will be used
icmp_code_range	ACL.Rule.IpRule.Icmp.Range		Inclusive range representing icmp codes to be used.
icmp_type_range	ACL.Rule.IpRule.Icmp.Range		

ACL.Rule.IpRule.Icmp.Range

Field	Type	Label	Description
first	uint32		
last	uint32		

ACL.Rule.IpRule.Ip

IP used in this Access List Entry.

Field	Type	Label	Description
destination_network	string		Destination IPv4/IPv6 network address (/)
source_network	string		Destination IPv4/IPv6 network address (/)
protocol	uint32		IP protocol number (http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml) Zero value (i.e. undefined protocol) means that the protocol to match will be automatically selected from one of the ICMP/ICMP6/TCP/UDP based on the rule definition. For example, if “icmp” is defined and src/dst addresses are IPv6 then packets of the ICMP6 protocol will be matched, etc.

ACL.Rule.IpRule.PortRange

Inclusive range representing destination ports to be used. When only lower-port is present, it represents a single port.

Field	Type	Label	Description
lower_port	uint32		
upper_port	uint32		If upper port is set, it must be greater or equal to lower port

ACL.Rule.IpRule.Tcp

Field	Type	Label	Description
destination_port_range	ACL.Rule.IpRule.PortRange		
source_port_range	ACL.Rule.IpRule.PortRange		
tcp_flags_mask	uint32		Binary mask for tcp flags to match. MSB order (FIN at position 0). Applied as logical AND to tcp flags field of the packet being matched, before it is compared with tcp-flags-value.
tcp_flags_value	uint32		Binary value for tcp flags to match. MSB order (FIN at position 0). Before tcp-flags-value is compared with tcp flags field of the packet being matched, tcp-flags-mask is applied to packet field value.

ACL.Rule.IpRule.Udp

Field	Type	Label	Description
destination_port_range	ACL.Rule.IpRule.PortRange		
source_port_range	ACL.Rule.IpRule.PortRange		

ACL.Rule.MacIpRule

Field	Type	Label	Description
source_address	string		
source_address_prefix	uint32		
source_mac_address	string		Before source-mac-address is compared with source mac address field of the packet being matched, source-mac-address-mask is applied to packet field value.
source_mac_address_mask	string		Source MAC address mask. Applied as logical AND with source mac address field of the packet being matched, before it is compared with source-mac-address.

ACL.Rule.Action

Name	Number	Description
DENY	0	
PERMIT	1	
REFLECT	2	

ligato/vpp/stn/stn.proto

Rule

Field	Type	Label	Description
ip_address	string		
interface	string		

ligato/vpp/interfaces/interface.proto

AfpacketLink

AfpacketLink defines configuration for interface type: AF_PACKET

Field	Type	Label	Description
host_if_name	string		Name of the host (Linux) interface to bind to. This type of reference is suitable for scenarios when the target interface is not managed (and should not be touched) by the agent. In such cases the interface does not have logical name in the agent's namespace and can only be referenced by the host interface name (i.e. the name used in the Linux network stack). Please note that agent learns about externally created interfaces through netlink notifications. If, however, the target interface is managed by the agent, then it is recommended to use the alternative reference (see below), pointing to the interface by its logical name. One advantage of such approach is, that if AF-PACKET and the target Linux interface are requested to be created at the same time, then it can be done inside the same transaction because the agent does not rely on any notification from the Linux. It is mandatory to define either or .
linux_interface	string		Logical name of the Linux interface to bind to. This is an alternative interface reference to and preferred if the target interface is managed by the agent and not created externally (see comments for for explanation). It is mandatory to define either or .

BondLink

BondLink defines configuration for interface type: BOND_INTERFACE

Field	Type	Label	Description
id	uint32		
mode	BondLink.Mode		
lb	BondLink.LoadBalance		Load balance is optional and valid only for XOR and LACP modes
bonded_interfaces	BondLink.BondedInterface	repeated	

BondLink.BondedInterface

Field	Type	Label	Description
name	string		
is_passive	bool		
is_long_timeout	bool		

GreLink

Field	Type	Label	Description
tunnel_type	GreLink.Type		
src_addr	string		
dst_addr	string		
outer_fib_id	uint32		

Field	Type	Label	Description
session_id	uint32		

GtpuLink

Field	Type	Label	Description
src_addr	string		Source VTEP address
dst_addr	string		Destination VTEP address
multicast	string		Name of multicast interface
teid	uint32		Tunnel endpoint identifier - local
remote_teid	uint32		Tunnel endpoint identifier - remote
encap_vrf_id	uint32		VRF id for the encapsulated packets
decap_next	GtpuLink.NextNode		Deprecated. DEPRECATED - use decap_next_node
decap_next_node	uint32		Next VPP node after decapsulation

IPIPLink

Field	Type	Label	Description
tunnel_mode	IPIPLink.Mode		Mode of the IPIP tunnel
src_addr	string		Source VTEP IP address
dst_addr	string		Destination VTEP IP address

IPSecLink

IPSecLink defines configuration for interface type: IPSEC_TUNNEL

Field	Type	Label	Description
esn	bool		Extended sequence number
anti_replay	bool		Anti replay option
local_ip	string		Local IP address
remote_ip	string		Remote IP address
local_spi	uint32		Local security parameter index
remote_spi	uint32		Remote security parameter index
crypto_alg	ligato.vpp.ipsec.CryptoAlg		Cryptographic algorithm for encryption
local_crypto_key	string		
remote_crypto_key	string		
integ_alg	ligato.vpp.ipsec.IntegAlg		Cryptographic algorithm for authentication
local_integ_key	string		
remote_integ_key	string		
enable_udp_encap	bool		

Interface

Interface defines a VPP interface.

Field	Type	Label	Description
name	string		Name is mandatory field representing logical name for the interface. It must be unique across all configured VPP interfaces.
type	Interface.Type		Type represents the type of VPP interface and it must match the actual Link.

Field	Type	Label	Description
enabled	bool		Enabled controls if the interface should be UP.
phys_address	string		PhysAddress represents physical address (MAC) of the interface. Random address will be assigned if left empty.
ip_addresses	string	repeated	IPAddresses define list of IP addresses for the interface and must be defined in the following format: /. Interface IP address can be also allocated via netalloc plugin and referenced here, see: api/models/netalloc/netalloc.proto
vrf	uint32		Vrf defines the ID of VRF table that the interface is assigned to. The VRF table must be explicitly configured (see api/models/vpp/l3/vrf.proto). When using unnumbered interface the actual vrf is inherited from the interface referenced by the numbered interface and this field is ignored.
set_dhcp_client	bool		SetDhcpClient enables DHCP client on interface.
ip6_nd	Interface.IP6ND		
mtu	uint32		Mtu sets MTU (Maximum Transmission Unit) for this interface. If set to zero, default MTU (usually 9216) will be used.
unnumbered	Interface.Unnumbered		
rx_modes	Interface.RxMode	repeated	
rx_placements	Interface.RxPlacement	repeated	
sub	SubInterface		
memif	MemifLink		
afpacket	AfpacketLink		
tap	TapLink		
vxlan	VxlanLink		
ipsec	IPSecLink		Deprecated. Deprecated in VPP 20.01+. Use IPIP_TUNNEL + ipsec.TunnelProtection instead.
vmx_net3	VmxNet3Link		
bond	BondLink		
gre	GreLink		
gtpu	GtpuLink		
ipip	IPIPLink		
wireguard	WireguardLink		
rdma	RDMALink		

Interface.IP6ND

Ip6Nd is used to enable/disable IPv6 ND address autoconfiguration and setting up default routes

Field	Type	Label	Description
address_autoconfig	bool		Enable IPv6 ND address autoconfiguration.
install_default_routes	bool		Enable installing default routes.

Interface.RxMode

Field	Type	Label	Description
queue	uint32		
mode	Interface.RxMode.Type		
default_mode	bool		DefaultMode, if set to true, the field will be ignored and the will be used as a default for all the queues.

Interface.RxPlacement

Field	Type	Label	Description
queue	uint32		Select from interval <0, number-of-queues)
worker	uint32		Select from interval <0, number-of-workers)
main_thread	bool		Let the main thread to process the given queue - if enabled, value of is ignored

Interface.Unnumbered

Unnumbered is used for inheriting IP address from another interface.

Field	Type	Label	Description
interface_with_ip	string		InterfaceWithIp is the name of interface to inherit IP address from.

MemifLink

MemifLink defines configuration for interface type: MEMIF

Field	Type	Label	Description
mode	MemifLink.MemifMode		
master	bool		
id	uint32		Id is a 32bit integer used to authenticate and match opposite sides of the connection
socket_filename	string		Filename of the socket used for connection establishment
secret	string		
ring_size	uint32		The number of entries of RX/TX rings
buffer_size	uint32		Size of the buffer allocated for each ring entry
rx_queues	uint32		Number of rx queues (only valid for slave)
tx_queues	uint32		Number of tx queues (only valid for slave)

RDMALink

https://github.com/FDio/vpp/blob/master/src/plugins/rdma/rdma_doc.rst

Field	Type	Label	Description
host_if_name	string		Linux interface name representing the RDMA-enabled network device to attach into.
mode	RDMALink.Mode		Mode at which the RDMA driver operates.
rxq_num	uint32		Number of receive queues. By default only one RX queue is used.
rxq_size	uint32		The size of each RX queue. Default is 1024 bytes.
txq_size	uint32		The size of each TX queue. Default is 1024 bytes.

SubInterface

SubInterface defines configuration for interface type: SUB_INTERFACE

Field	Type	Label	Description
parent_name	string		Name of the parent (super) interface
sub_id	uint32		SubInterface ID, used as VLAN
tag_rw_option	SubInterface.TagRewriteOptions		VLAN tag rewrite rule applied for given tag for sub-interface
push_dot1q	bool		Set ether-type of the first tag to dot1q if true, dot1ad otherwise
tag1	uint32		First tag (required for PUSH1 and any TRANSLATE)
tag2	uint32		Second tag (required for PUSH2 and any TRANSLATE)

TapLink

TapLink defines configuration for interface type: TAP

Field	Type	Label	Description
version	uint32		1 / unset = use the original TAP interface; 2 = use a fast virtio-based TAP
host_if_name	string		Name of the TAP interface in the host OS; if empty, it will be auto-generated (suitable for combination with TAP_TO_VPP interface from Linux ifplugin, because then this name is only temporary anyway)
to_microservice	string		If TAP connects VPP with microservice, fill this parameter with the target microservice name - should match with the namespace reference of the associated TAP_TO_VPP interface (it is still moved to the namespace by Linux-ifplugin but VPP-ifplugin needs to be aware of this dependency)
rx_ring_size	uint32		Rx ring buffer size; must be power of 2; default is 256; only for TAP v.2
tx_ring_size	uint32		Tx ring buffer size; must be power of 2; default is 256; only for TAP v.2
enable_gso	bool		EnableGso enables GSO mode for TAP interface.
enable_tunnel	bool		EnableTunnel enables tunnel mode for TAP interface.

VmxNet3Link

VmxNet3Link defines configuration for interface type: VMXNET3_INTERFACE PCI address (unsigned 32bit int) is derived from vmxnet3 interface name. It is expected that the interface name is in format “vmxnet3-///”, where ‘d’ stands for domain (max ffff), ‘b’ is bus (max ff), ‘s’ is slot (max 1f) and ‘f’ is function’ (max 7). All values are base 16

Field	Type	Label	Description
enable_elog	bool		Turn on elog
rxq_size	uint32		Receive queue size (default is 1024)
txq_size	uint32		Transmit queue size (default is 1024)

VxlanLink

VxlanLink defines configuration for interface type: VXLAN_TUNNEL

Field	Type	Label	Description
src_address	string		SrcAddress is source VTEP address
dst_address	string		DstAddress is destination VTEP address

Field	Type	Label	Description
vni	uint32		Vni stands for VXLAN Network Identifier
multicast	string		Multicast defines name of multicast interface
gpe	VxlanLink.Gpe		

VxlanLink.Gpe

Gpe (Generic Protocol Extension) allows encapsulating not only Ethernet frame payload.

Field	Type	Label	Description
decap_vrf_id	uint32		
protocol	VxlanLink.Gpe.Protocol		Protocol defines encapsulated protocol

WireguardLink

Field	Type	Label	Description
private_key	string		Private-key base64
port	uint32		Listen UDP port
src_addr	string		Source IP address

BondLink.LoadBalance

Name	Number	Description
L2	0	
L34	1	
L23	2	
RR	3	Round robin
BC	4	Broadcast
AB	5	Active backup

BondLink.Mode

Name	Number	Description
UNKNOWN	0	
ROUND_ROBIN	1	
ACTIVE_BACKUP	2	
XOR	3	
BROADCAST	4	
LACP	5	

GreLink.Type

Name	Number	Description
UNKNOWN	0	
L3	1	L3 GRE (i.e. this tunnel is in L3 mode)
TEB	2	TEB - Transparent Ethernet Bridging - the tunnel is in L2 mode
ERSPAN	3	ERSPAN - the tunnel is for port mirror SPAN output

GtpuLink.NextNode

Name	Number	Description
DEFAULT	0	The default next node is l2-input
L2	1	l2-input
IP4	2	ip4-input
IP6	3	ip6-input

IPIPLink.Mode

Name	Number	Description
POINT_TO_POINT	0	point-to-point tunnel
POINT_TO_MULTIPPOINT	1	point-to multipoint tunnel (supported starting from VPP 20.05)

Interface.RxMode.Type

Type definition is from: vpp/include/vnet/interface.h

Name	Number	Description
UNKNOWN	0	
POLLING	1	
INTERRUPT	2	
ADAPTIVE	3	
DEFAULT	4	

Interface.Type

Type defines VPP interface types.

Name	Number	Description
UNDEFINED_TYPE	0	
SUB_INTERFACE	1	
SOFTWARE_LOOPBACK	2	
DPDK	3	
MEMIF	4	
TAP	5	
AF_PACKET	6	
VXLAN_TUNNEL	7	
IPSEC_TUNNEL	8	Deprecated in VPP 20.01+. Use IPIP_TUNNEL + ipsec.TunnelProtection instead.
VMXNET3_INTERFACE	9	
BOND_INTERFACE	10	
GRE_TUNNEL	11	
GTPU_TUNNEL	12	
IPIP_TUNNEL	13	
WIREGUARD_TUNNEL	14	
RDMA	15	

MemifLink.MemifMode

Name	Number	Description
ETHERNET	0	
IP	1	
PUNT_INJECT	2	

RDMALink.Mode

Name	Number	Description
AUTO	0	
IBV	1	InfiniBand Verb (using libibverb).
DV	2	Direct Verb allows the driver to access the NIC HW RX/TX rings directly instead of having to go through libibverb and suffering associated overhead. It will be automatically selected if the adapter supports it.

SubInterface.TagRewriteOptions

Name	Number	Description
DISABLED	0	
PUSH1	1	
PUSH2	2	
POP1	3	
POP2	4	
TRANSLATE11	5	
TRANSLATE12	6	
TRANSLATE21	7	
TRANSLATE22	8	

VxlanLink.Gpe.Protocol

Name	Number	Description
UNKNOWN	0	
IP4	1	
IP6	2	
ETHERNET	3	
NSH	4	

ligato/vpp/interfaces/dhcp.proto

DHCPLease

DHCPLease is a notification, i.e. flows from SB upwards

Field	Type	Label	Description
interface_name	string		
host_name	string		
is_ipv6	bool		
host_phys_address	string		
host_ip_address	string		IP addresses in the format /
router_ip_address	string		IP addresses in the format /

ligato/vpp/interfaces/state.proto

InterfaceNotification

Field	Type	Label	Description
type	InterfaceNotification.NotifType		
state	InterfaceState		

InterfaceState

Field	Type	Label	Description
name	string		
internal_name	string		
type	Interface.Type		
if_index	uint32		
admin_status	InterfaceState.Status		
oper_status	InterfaceState.Status		
last_change	int64		
phys_address	string		
speed	uint64		
mtu	uint32		
duplex	InterfaceState.Duplex		
statistics	InterfaceState.Statistics		

InterfaceState.Statistics

Field	Type	Label	Description
in_packets	uint64		
in_bytes	uint64		
out_packets	uint64		
out_bytes	uint64		
drop_packets	uint64		
punt_packets	uint64		
ipv4_packets	uint64		
ipv6_packets	uint64		
in_nobuf_packets	uint64		
in_miss_packets	uint64		
in_error_packets	uint64		
out_error_packets	uint64		

InterfaceStats

Field	Type	Label	Description
name	string		
rx	InterfaceStats.CombinedCounter		
tx	InterfaceStats.CombinedCounter		
rx_unicast	InterfaceStats.CombinedCounter		
rx_multicast	InterfaceStats.CombinedCounter		
rx_broadcast	InterfaceStats.CombinedCounter		
tx_unicast	InterfaceStats.CombinedCounter		
tx_multicast	InterfaceStats.CombinedCounter		
tx_broadcast	InterfaceStats.CombinedCounter		

Field	Type	Label	Description
rx_error	uint64		
tx_error	uint64		
rx_no_buf	uint64		
rx_miss	uint64		
drops	uint64		
punts	uint64		
ip4	uint64		
ip6	uint64		
mpls	uint64		

InterfaceStats.CombinedCounter

Field	Type	Label	Description
packets	uint64		
bytes	uint64		

InterfaceNotification.NotifType

Name	Number	Description
UNKNOWN	0	
UPDOWN	1	
COUNTERS	2	

InterfaceState.Duplex

Name	Number	Description
UNKNOWN_DUPLEX	0	
HALF	1	
FULL	2	

InterfaceState.Status

Name	Number	Description
UNKNOWN_STATUS	0	
UP	1	
DOWN	2	
DELETED	3	

ligato/vpp/interfaces/span.proto

Span

Field	Type	Label	Description
interface_from	string		
interface_to	string		
direction	Span.Direction		
is_l2	bool		

Span.Direction

Name	Number	Description
UNKNOWN	0	
RX	1	
TX	2	
BOTH	3	

ligato/vpp/wireguard/wireguard.proto

Peer

Field	Type	Label	Description
public_key	string		Public-key base64
port	uint32		Peer UDP port
persistent_keepalive	uint32		Keepalive interval (sec)
endpoint	string		Endpoint IP
wg_if_name	string		The name of the wireguard interface to which this peer belongs
flags	uint32		Flags WIREGUARD_PEER_STATUS_DEAD = 0x1
allowed_ips	string	repeated	Allowed IPs

ligato/vpp/nat/nat.proto

DNat44

DNat44 defines destination NAT44 configuration.

Field	Type	Label	Description
label	string		Unique identifier for the DNAT configuration.
st_mappings	DNat44.StaticMapping	repeated	A list of static mappings in DNAT.
id_mappings	DNat44.IdentityMapping	repeated	A list of identity mappings in DNAT.

DNat44.IdentityMapping

IdentityMapping defines an identity mapping in DNAT.

Field	Type	Label	Description
vrf_id	uint32		VRF (table) ID. Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto).
interface	string		Name of the interface to use address from; preferred over ip_address.
ip_address	string		IP address.
port	uint32		Port (do not set for address mapping).
protocol	DNat44.Protocol		Protocol used for identity mapping.

DNat44.StaticMapping

StaticMapping defines a list of static mappings in DNAT.

Field	Type	Label	Description
external_interface	string		Interface to use external IP from; preferred over external_ip.
external_ip	string		External address.
external_port	uint32		Port (do not set for address mapping).
local_ips	DNat44.StaticMapping.LocalIP	LocalIP	List of local IP addresses. If there is more than one entry, load-balancing is enabled.
protocol	DNat44.Protocol		Protocol used for static mapping.
twice_nat	DNat44.StaticMapping.TwiceNatMode		Enable/disable (self-)twice NAT.
twice_nat_pool_ip	string		IP address from Twice-NAT address pool that should be used as source IP in twice-NAT processing. This is override for default behaviour of choosing the first IP address from twice-NAT pool that has available at least one free port (NAT is tracking translation sessions and exhausts free ports for given IP address). This is needed for example in use cases when multiple twice-NAT translations need to use different IP Addresses as source IP addresses. This functionality works with VPP 20.09 and newer. It also needs to have twice_nat set to ENABLED. It doesn't work for load-balanced static mappings (=local_ips has multiple values).
session_affinity	uint32		Session affinity. 0 means disabled, otherwise client IP affinity sticky time in seconds.

DNat44.StaticMapping.LocalIP

LocalIP defines a local IP addresses.

Field	Type	Label	Description
vrf_id	uint32		VRF (table) ID. Non-zero VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto).
local_ip	string		Local IP address).
local_port	uint32		Port (do not set for address mapping).
probability	uint32		Probability level for load-balancing mode.

Nat44AddressPool

Nat44AddressPool defines an address pool used for NAT44.

Field	Type	Label	Description
name	string		Unique name for address pool
vrf_id	uint32		VRF id of tenant, 0xFFFFFFFF means independent of VRF. Non-zero (and not all-ones) VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto).
first_ip	string		First IP address of the pool.
last_ip	string		Last IP address of the pool. Should be higher than first_ip or empty.
twice_nat	bool		Enable/disable twice NAT.

Nat44Global

Nat44Global defines global NAT44 configuration. In VPP version 21.01 and newer the NAT44 plugin has to be explicitly enabled (by default it is disabled so that it doesn't consume any computational resources). With ligato control-plane the NAT44 plugin is enabled by submitting the NAT44Global configuration (even default values will make the plugin

enabled). Without Nat44Global, all other NAT44 configuration items (DNat44, Nat44Interface and Nat44AddressPool) will be in the PENDING state.

Field	Type	Label	Description
forwarding	bool		Enable/disable forwarding. By default it is disabled.
endpoint_independent	bool		Enable/disable endpoint-independent mode. In endpoint-independent (also known as “simple”) mode the VPP NAT plugin holds less information for each session, but only works with outbound NAT and static mappings. In endpoint-dependent mode, which ligato selects as the default, the VPP NAT plugin uses more information to track each session, which in turn enables additional features such as out-to-in-only and twice-nat. In older versions of VPP (<= 20.09) this field is ignored because mode at which the NAT44 plugin operates is given by the VPP startup configuration file (i.e. config created before VPP even starts, therefore not managed by ligato). The endpoint-independent mode is the default and the dependent mode is turned on with this config stanza (included in vpp.conf used by ligato for older VPPs): nat { endpoint-dependent }
nat_interfaces	Nat44Global.Interface	repeated	Deprecated. List of NAT-enabled interfaces. Deprecated - use separate Nat44Interface entries instead.
address_pool	Nat44Global.Address	repeated	Deprecated. Address pool used for source IP NAT. Deprecated - use separate Nat44AddressPool entries instead.
virtual_reassembly	VirtualReassembly		Virtual reassembly for IPv4.

Nat44Global.Address

Address defines an address to be used for source IP NAT.

Field	Type	Label	Description
address	string		IPv4 address.
vrf_id	uint32		VRF id of tenant, 0xFFFFFFFF means independent of VRF. Non-zero (and not all-ones) VRF has to be explicitly created (see api/models/vpp/l3/vrf.proto).
twice_nat	bool		Enable/disable twice NAT.

Nat44Global.Interface

Interface defines a network interface enabled for NAT.

Field	Type	Label	Description
name	string		Interface name (logical).
is_inside	bool		Distinguish between inside/outside interface.
output_feature	bool		Enable/disable output feature.

Nat44Interface

Nat44Interface defines a local network interfaces enabled for NAT44.

Field	Type	Label	Description
name	string		Interface name (logical).
nat_inside	bool		Enable/disable NAT on inside.
nat_outside	bool		Enable/disable NAT on outside.
output_feature	bool		Enable/disable output feature.

VirtualReassembly

VirtualReassembly defines NAT virtual reassembly settings.

Field	Type	Label	Description
timeout	uint32		Reassembly timeout.
max_reassemblies	uint32		Maximum number of concurrent reassemblies.
max_fragments	uint32		Maximum number of fragments per reassembly.
drop_fragments	bool		If set to true fragments are dropped, translated otherwise.

DNat44.Protocol

Available protocols.

Name	Number	Description
TCP	0	
UDP	1	
ICMP	2	ICMP is not permitted for load balanced entries.

DNat44.StaticMapping.TwiceNatMode

Available twice-NAT modes.

Name	Number	Description
DISABLED	0	
ENABLED	1	
SELF	2	

ligato/linux/punt/punt.proto

PortBased

Define network socket type

Field	Type	Label	Description
l4_protocol	PortBased.L4Protocol		
l3_protocol	PortBased.L3Protocol		
port	uint32		

Proxy

Proxy allows to listen on network socket or unix domain socket, and resend to another network/unix domain socket

Field	Type	Label	Description
rx_port	PortBased		
rx_socket	SocketBased		
tx_port	PortBased		
tx_socket	SocketBased		

SocketBased

Define unix domain socket type for IPC

Field	Type	Label	Description
path	string		

PortBased.L3Protocol

L3 protocol

Name	Number	Description
UNDEFINED_L3	0	
IPV4	1	
IPV6	2	
ALL	3	

PortBased.L4Protocol

L4 protocol

Name	Number	Description
UNDEFINED_L4	0	
TCP	6	
UDP	17	

ligato/linux/linux.proto

ConfigData

Field	Type	Label	Description
interfaces	interfaces.Interface	repeated	
arp_entries	l3.ARPEntry	repeated	
routes	l3.Route	repeated	

Notification

Field	Type	Label	Description
interface	interfaces.InterfaceNotification		

ligato/linux/l3/arp.proto

ARPEntry

Field	Type	Label	Description
interface	string		
ip_address	string		
hw_address	string		

ligato/linux/l3/route.proto

Route

Field	Type	Label	Description
outgoing_interface	string		Outgoing interface logical name (mandatory).
scope	Route.Scope		The scope of the area where the link is valid.
dst_network	string		Destination network address in the format
gw_addr	string		Gateway IP address (without mask, optional). Address can be also allocated via netalloc plugin and referenced here, see: api/models/netalloc/netalloc.proto
metric	uint32		routing metric (weight)

Route.Scope

Name	Number	Description
UNDEFINED	0	
GLOBAL	1	
SITE	2	
LINK	3	
HOST	4	

ligato/linux/iptables/iptables.proto

RuleChain

Field	Type	Label	Description
name	string		logical name of the rule chain across all configured rule chains (mandatory)
namespace	ligato.linux.namespace.NetNamespace		network namespace in which this rule chain is applied
interfaces	string	repeated	list of interfaces referred by the rules (optional)
protocol	RuleChain.Protocol		protocol (address family) of the rule chain
table	RuleChain.Table		table the rule chain belongs to
chain_type	RuleChain.ChainType		type of the chain
chain_name	string		name of the chain, used only for chains with CUSTOM chain_type
default_policy	RuleChain.Policy		default policy of the chain. Used for FILTER tables only.
rules	string	repeated	ordered list of strings containing the match and action part of the rules, e.g. “-i eth0 -s 192.168.0.1 -j ACCEPT”

RuleChain.ChainType

Name	Number	Description
CUSTOM	0	
INPUT	1	
OUTPUT	2	
FORWARD	3	
PREROUTING	4	
POSTROUTING	5	

RuleChain.Policy

Name	Number	Description
NONE	0	
ACCEPT	1	
DROP	2	
QUEUE	3	
RETURN	4	

RuleChain.Protocol

Name	Number	Description
IPV4	0	
IPV6	1	

RuleChain.Table

Name	Number	Description
FILTER	0	
NAT	1	
MANGLE	2	
RAW	3	
SECURITY	4	

ligato/linux/interfaces/interface.proto

Interface

Field	Type	Label	Description
name	string		Name is mandatory field representing logical name for the interface. It must be unique across all configured interfaces.
type	Interface.Type		Type represents the type of interface and It must match with actual Link.
namespace	ligato.linux.namespace.NetNamespace		Namespace is a reference to a Linux network namespace where the interface should be put into.
host_if_name	string		Name of the interface in the host OS. If not set, the host name will be the same as the interface logical name.
enabled	bool		Enabled controls if the interface should be UP.

Field	Type	Label	Description
ip_addresses	string	repeated	IPAddresses define list of IP addresses for the interface and must be defined in the following format: /. Interface IP address can be also allocated via netalloc plugin and referenced here, see: api/models/netalloc/netalloc.proto
phys_address	string		PhysAddress represents physical address (MAC) of the interface. Random address will be assigned if left empty. Not used (and not supported) by VRF devices. MTU is the maximum transmission unit value.
mtu	uint32		
veth	VethLink		VETH-specific configuration
tap	TapLink		TAP_TO_VPP-specific configuration
vrf_dev	VrfDevLink		VRF_DEVICE-specific configuration
link_only	bool		Configure/Resync link only. IP/MAC addresses are expected to be configured externally - i.e. by a different agent or manually via CLI.
vrf_master_interface	string		Reference to the logical name of a VRF_DEVICE interface. If defined, this interface will be enslaved to the VRF device and will thus become part of the VRF (L3-level separation) that the device represents. Interfaces enslaved to the same VRF_DEVICE master interface therefore comprise single VRF with a separate routing table.

TapLink

Field	Type	Label	Description
vpp_tap_if_name	string		Logical name of the VPP TAP interface (mandatory for TAP_TO_VPP)

VethLink

Field	Type	Label	Description
peer_if_name	string		Name of the VETH peer, i.e. other end of the linux veth (mandatory for VETH)
rx_checksum_offloading	VethLink.ChecksumOffloading		Checksum offloading - Rx side (enabled by default)
tx_checksum_offloading	VethLink.ChecksumOffloading		Checksum offloading - Tx side (enabled by default)

VrfDevLink

Field	Type	Label	Description
routing_table	uint32		Routing table associated with the VRF. Table ID is an 8-bit unsigned integer value. Please note that 253, 254 and 255 are reserved values for special routing tables (main, default, local). Multiple VRFs inside the same network namespace should each use a different routing table. For more information, visit: http://linux-ip.net/html/routing-tables.html

Interface.Type

Name	Number	Description
UNDEFINED	0	
VETH	1	
TAP_TO_VPP	2	TAP created by VPP to have the Linux-side further configured
LOOPBACK	3	LOOPBACK is used to attach configuration to an existing “lo” interface, but unlike EXISTING type it is not limited to the default network namespace (i.e. loopbacks in other containers can be referenced also). To create an additional interface which effectively acts as a loopback, use DUMMY interface (see below).
EXISTING	4	Wait for and potentially attach additional network configuration to an interface created externally (i.e. not by this agent) in the default network namespace (i.e. same as used by the agent). Behaviour of the EXISTING interface depends on the values of ip_addresses and link_only attributes as follows: 1. link_only=false and ip_addresses are empty: agent waits for interface to be created externally and then configures it in the L2-only mode (resync will remove any IP addresses configured from outside of the agent) 2. link_only=false and ip_addresses are non-empty: agent waits for interface to be created externally and then attaches the selected IP addresses to it (resync removes any other IPs added externally) 3. link_only=true and ip_addresses are empty: agent only waits for the interface to exist (it doesn't wait for or change any IP addresses attached to it) 4. link_only=true and ip_addresses are non empty: agent waits for the interface to exist and the selected IP addresses to be assigned (i.e. there will be derived value for each expected IP address in the PENDING state until the address is assigned to the interface externally)
VRF_DEVICE	5	In Linux, VRF is implemented as yet another type of netdevice (i.e. listed with <code>ip link show</code>). Network interfaces are then assigned to VRF simply by enslaving them to the VRF device. For more information, visit: https://www.kernel.org/doc/Documentation/networking/vrf.txt
DUMMY	6	Create a dummy Linux interface which effectively behaves just like the loopback.

VethLink.ChecksumOffloading

Name	Number	Description
CHKSM_OFFLOAD_DEFAULT	0	
CHKSM_OFFLOAD_ENABLED	1	
CHKSM_OFFLOAD_DISABLED	2	

ligato/linux/interfaces/state.proto

InterfaceNotification

Field	Type	Label	Description
type	<code>InterfaceNotification.NotifType</code>		
state	<code>InterfaceState</code>		

InterfaceState

Field	Type	Label	Description
name	<code>string</code>		
internal_name	<code>string</code>		

Field	Type	Label	Description
type	Interface.Type		
if_index	int32		
admin_status	InterfaceState.Status		
oper_status	InterfaceState.Status		
last_change	int64		
phys_address	string		
speed	uint64		
mtu	uint32		
statistics	InterfaceState.Statistics		

InterfaceState.Statistics

Field	Type	Label	Description
in_packets	uint64		
in_bytes	uint64		
out_packets	uint64		
out_bytes	uint64		
drop_packets	uint64		
in_error_packets	uint64		
out_error_packets	uint64		

InterfaceNotification.NotifType

Name	Number	Description
UNKNOWN	0	
UPDOWN	1	

InterfaceState.Status

Name	Number	Description
UNKNOWN_STATUS	0	
UP	1	
DOWN	2	
DELETED	3	

ligato/linux/namespace/namespace.proto

NetNamespace

Field	Type	Label	Description
type	NetNamespace.ReferenceType		
reference	string		Reference defines reference specific to the namespace type: * namespace ID (NSID) * PID number (PID) * file path (FD) * microservice label (MICROSERVICE)

NetNamespace.ReferenceType

Name	Number	Description
UNDEFINED	0	
NSID	1	named namespace
PID	2	namespace of a given process
FD	3	namespace referenced by a file handle
MICROSERVICE	4	namespace of a docker container running given microservice

ligato/configurator/statspoller.proto

PollStatsRequest

Field	Type	Label	Description
period_sec	uint32		PeriodSec defines polling period (in seconds). Set to zero to return just single polling.
num_polls	uint32		NumPolls defines number of pollings. Set to non-zero number to stop the polling after specified number of pollings is reached.

PollStatsResponse

Field	Type	Label	Description
poll_seq	uint32		PollSeq defines the sequence number of this polling response.
stats	Stats		Stats contains polled stats data.

Stats

Stats defines stats data returned by StatsPollerService.

Field	Type	Label	Description
vpp_stats	ligato.vpp.Stats		

StatsPollerService

StatsPollerService provides operations for collecting statistics.

Method Name	Request Type	Response Type	Description
PollStats	PollStatsRequest	PollStatsResponse stream	PollStats is used for polling stats with specific period and number of pollings.

ligato/configurator/configurator.proto

Config

Config describes all supported configs into a single config message.

Field	Type	Label	Description
vpp_config	ligato.vpp.ConfigData		
linux_config	ligato.linux.ConfigData		

Field	Type	Label	Description
netalloc_config	<code>ligato.netalloc.ConfigData</code>		

DeleteRequest

Field	Type	Label	Description
delete	<code>Config</code>		Delete is a config data to be deleted.
wait_done	<code>bool</code>		WaitDone option can be used to block until either config delete is done (non-pending) or request times out.

NOTE: WaitDone is intended to be used for config updates that depend on some event from dataplane to fully configure. Using this with incomplete config updates will require another update request to unblock. |

DeleteResponse

DumpRequest

DumpResponse

Field	Type	Label	Description
dump	<code>Config</code>		Dump is a running config.

GetRequest

GetResponse

Field	Type	Label	Description
config	<code>Config</code>		Config describes desired config retrieved from agent.

Notification

Notification describes all known notifications into a single message.

Field	Type	Label	Description
vpp_notification	<code>ligato.vpp.Notification</code>		
linux_notification	<code>ligato.linux.Notification</code>		

NotifyRequest

Field	Type	Label	Description
idx	<code>uint32</code>		
filters	<code>Notification</code>	repeated	

NotifyResponse

Field	Type	Label	Description
next_idx	uint32		Index of next notification
notification	Notification		Notification contains notification data.

UpdateRequest

Field	Type	Label	Description
update	Config		Update is a config data to be updated.
full_resync	bool		FullResync option can be used to overwrite all existing config with config update.

NOTE: Using FullResync with empty config update will remove all existing config. | | wait_done | **bool** | | WaitDone option can be used to block until either config update is done (non-pending) or request times out.

NOTE: WaitDone is intended to be used for config updates that depend on some event from dataplane to fully configure. Using this with incomplete config updates will require another update request to unblock. |

UpdateResponse

ConfiguratorService

ConfiguratorService provides basic operations for managing configuration and monitoring actual state.

Method Name	Request Type	Response Type	Description
Get	GetRequest	GetResponse	Get is used for listing desired config.
Update	UpdateRequest	UpdateResponse	Update is used for updating desired config.
Delete	DeleteRequest	DeleteResponse	Delete is used for deleting desired config.
Dump	DumpRequest	DumpResponse	Dump is used for dumping running config.
Notify	NotifyRequest	NotifyResponse stream	Notify is used for subscribing to notifications.

bfd/bfd.proto

BFD

Single-hop UDP-based bidirectional forwarding detection session

Field	Type	Label	Description
interface	string		Name of the interface the BFD session is attached to.
local_ip	string		Local IP address. The interface must have the same address configured.
peer_ip	string		IP address of the peer, must be the same IP version as the local address.
min_tx_interval	uint32		Desired minimum TX interval in milliseconds.
min_rx_interval	uint32		Required minimum RX interval in milliseconds.
detect_multiplier	uint32		Detect multiplier, must be non-zero value.

BFDEvent

BFDEvent is generated whenever a BFD state changes.

Field	Type	Label	Description
interface	string		
local_ip	string		
peer_ip	string		
session_state	BFDEvent.SessionState		

WatchBFDEventsRequest

Request message for the WatchBFDEvents method.

Field	Type	Label	Description
subscription_label	string		

BFDEvent.SessionState

Name	Number	Description
Unknown	0	
Down	1	
Init	2	
Up	3	

BFDWatcher

BFDWatcher provides API to watch for BFD events.

Method Name	Request Type	Response Type	Description
WatchBFDEvents	WatchBFDEventsRequest	BFDEvent stream	WatchBFDEvents allows to subscribe for BFD events.

nat64/nat64.proto

Nat64AddressPool

Nat44AddressPool defines an address pool used for NAT64.

Field	Type	Label	Description
vrf_id	uint32		VRF id of tenant, 0xFFFFFFFF means independent of VRF. Non-zero (and not all-ones) VRF has to be explicitly created (see proto/ligato/vpp/l3/vrf.proto).
first_ip	string		First IP address of the pool.
last_ip	string		Last IP address of the pool. Should be higher than first_ip or empty.

Nat64IPv6Prefix

IPv4-Embedded IPv6 Address Prefix used for NAT64. If no prefix is configured (at all or for a given VRF), then the well-known prefix (64:ff9b::/96) is used.

Field	Type	Label	Description
vrf_id	uint32		VRF id of tenant. At most one IPv6 prefix can be configured for a given VRF (that's why VRF is part of the key but prefix is not). Non-zero (and not all-ones) VRF has to be explicitly created (see proto/ligato/vpp/l3/vrf.proto).
prefix	string		NAT64 prefix in the / format.

Nat64Interface

Nat64Interface defines a local network interfaces enabled for NAT64.

Field	Type	Label	Description
name	string		Interface name (logical).
type	Nat64Interface.Type		

Nat64StaticBIB

Static NAT64 binding allowing IPv4 host from the outside to access IPv6 host from the inside.

Field	Type	Label	Description
vrf_id	uint32		VRF (table) ID. Non-zero VRF has to be explicitly created (see proto/ligato/vpp/l3/vrf.proto).
inside_ipv6_address	string		IPv6 host from the inside/local network.
inside_port	uint32		Inside port number (of the IPv6 host).
outside_ipv4_address	string		IPv4 host from the outside/external network.
outside_port	uint32		Outside port number (of the IPv4 host).
protocol	Nat64StaticBIB.Protocol		

Nat64Interface.Type

Name	Number	Description
IPV6_INSIDE	0	Interface connecting inside/local network with IPv6 endpoints.
IPV4_OUTSIDE	1	Interface connecting outside/external network with IPv4 endpoints.

Nat64StaticBIB.Protocol

Protocol to which the binding applies.

Name	Number	Description
TCP	0	
UDP	1	
ICMP	2	

Scalar Value Types

.proto Type	Notes	C++	Java	Python	Go
double		double	double	float	float64
float		float	float	float	float32
int32	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int	int32
int64	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.	int64	long	int/long	int64
uint32	Uses variable-length encoding.	uint32	int	int/long	uint32
uint64	Uses variable-length encoding.	uint64	long	int/long	uint64
sint32	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.	int32	int	int	int32
sint64	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.	int64	long	int/long	int64
fixed32	Always four bytes. More efficient than uint32 if values are often greater than 2 ²⁸ .	uint32	int	int	uint32
fixed64	Always eight bytes. More efficient than uint64 if values are often greater than 2 ⁵⁶ .	uint64	long	int/long	uint64
sfixed32	Always four bytes.	int32	int	int	int32
sfixed64	Always eight bytes.	int64	long	int/long	int64
bool		bool	boolean	boolean	bool
string	A string must always contain UTF-8 encoded or 7-bit ASCII text.	string	String	str/unicode	string
bytes	May contain any arbitrary sequence of bytes.	string	ByteString	str	[]byte