


# draft-porfiri-tsvwg-sctp- natsupp

A minimal approach for allowing SCTP Hosts to be instantiated behind  
NAT

# Use cases for SCTP in a NATted Network

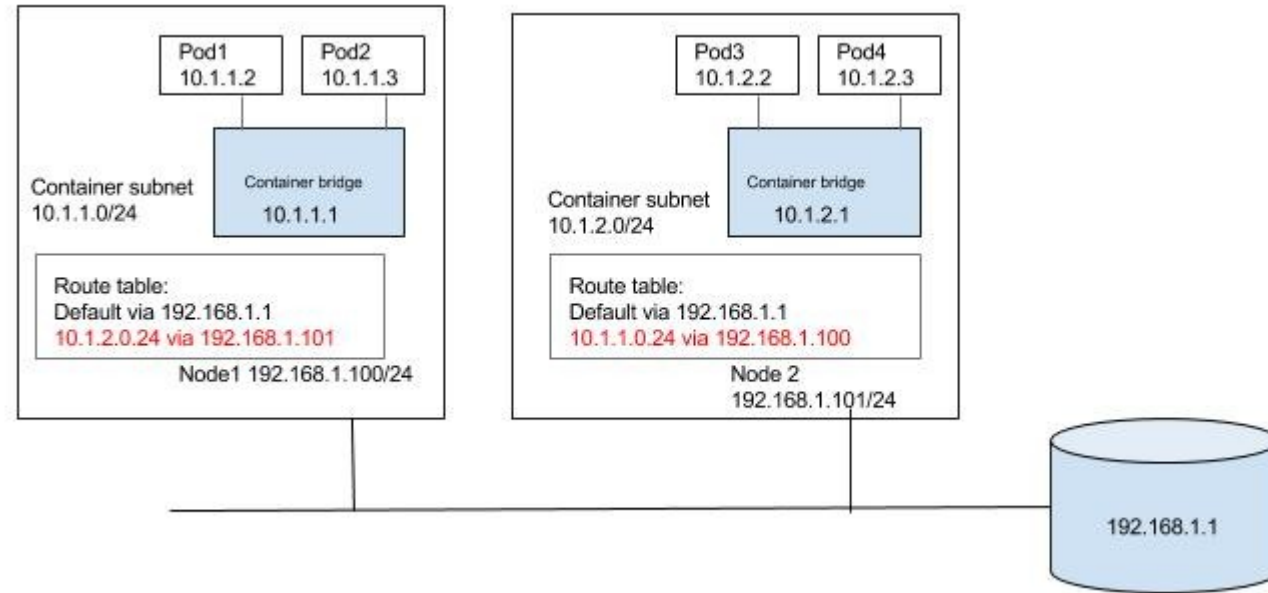
- Single Homed SCTP Clients
- Single Homed SCTP Server
- Multihomed SCTP Clients
- Multihomed SCTP Server
- Single Homed SCTP Client with Distributed Endpoint
- Multihomed SCTP Client with Distributed Endpoint
- Single Homed SCTP Server with Distributed Endpoint
- Multihomed SCTP Server with Distributed Endpoint



Cloud Based  
Deployment  
In a Kubernetes  
Execution  
environment

# Why Distributed SCTP Endpoint

- Scalability
- Reliability



A Kubernetes Cluster is deployed on multiple machines, a Service such as an SCTP Endpoint is instantiated on multiple PODs, those are exposed to the public network as a single instance of an SCTP Host. Networking is implemented by means of a Container Network Interface based on NAT. Traffic is distributed by means of Load Balancer algorithms.

# Approach of this NAT Support proposal

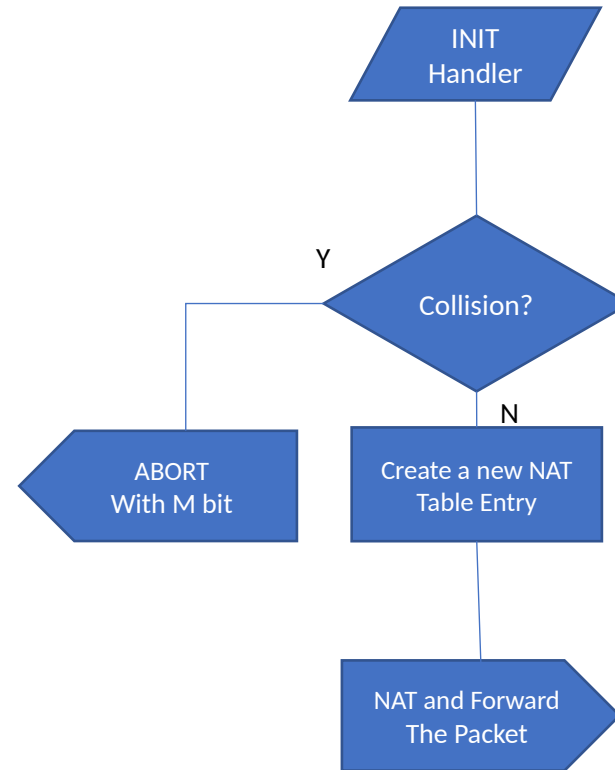
- NAT only parses the SCTP Common Header
- NAT searches for SCTP packet containing INIT chunk and checks for collision
- Collisions are communicated from NAT using an ABORT chunk with M-bit set (same as in draft-ietf-tsvwg-natsupp)
- Change of Source Endpoint in case of collision is up to SCTP User
- Multihoming is implemented as Single Homing being extended with ASCONF (rfc5061 same as in draft-ietf-tsvwg-natsupp)
- NAT setting for Multihoming exploits extra INIT chunks that implement a new option "RJ" (not needed)

# NAT implementation

- Implementation at NAT is kept as simple as possible
  - NAT doesn't parse SCTP chunks, thus it doesn't keep the Association State, it only supervises the Association 4-uple and removes the NAT Table entry when a timer supervision expires.
  - When receiving an INIT chunk, it check if the related 4-uple already exists in the NAT Table, if so NAT answers with ABORT, otherwise it sets the new entry in the NAT table and forwards the packet.
  - When receiving any other **outgoing** SCTP packet, if the related 4-uple exists it forwards it, otherwise it sets the new entry in the NAT table and forwards the packet.
  - When receiving any other **incoming** SCTP packet, if the related 4-uple exists it forwards it, otherwise it silently discards the packet.

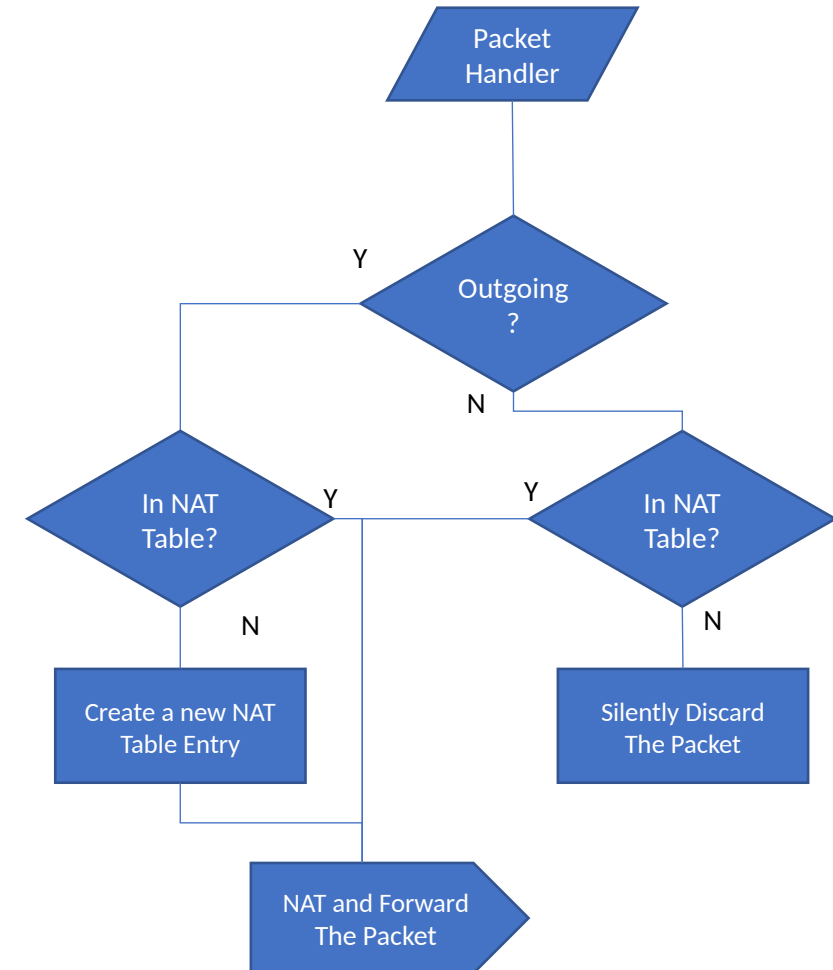
# Handling of INIT in NAT

- INIT handling in NAT devices is the same for outgoing or incoming packets.
- NAT doesn't need to parse INIT, it doesn't even need to know whether INIT has RJ option.



# Handling of other SCTP packets in NAT

- A packet that doesn't contain INIT chunk is handled depending on the direction
- Outgoing SCTP packets are always forwarded, if no NAT entries exist, those are created.
- Incoming SCTP packets are forwarded only if NAT entries exist.



# Handling of INIT for Multihoming

- Approach to Multihoming is the same as in the draft-ietf-tsvwg-nat supp, that is creating a single-homed association first and then adding the other IP addresses.
- The proposal needs the NAT functions to be set before extra addresses are added, this is accomplished by sending INIT chunks to the peer from the extra IP addresses. Once the peer has answered with INIT-ACK, the procedure from rfc5061 can be used.
- The proposal adds an extra parameter "Repetita Juvant" to INIT, but this is not actually needed.



# Role of the Load Balancer

- When an Endpoint is distributed among SCTP Hosts, NAT cannot decide how to distribute Associations by itself.
- NAT is generally not responsible for Load Balancing, there's the need of an extra LB function for supporting those cases.
- LB is not part of the proposal, a number of different strategies can be implemented as rules for Load Balancing.
- There are cases described in the examples that need NAT to be supported by a LB such as in section 7.3

# Conclusions

- The current proposal is a minimal approach to solve some problems of integrating NAT and SCTP.
- The key of the proposal is simplicity both at the SCTP Hosts and at the NAT functions.
- The proposal doesn't need vTAG handling at NAT
- Since it keeps most of the concepts from draft-ietf-tsvwg-natsupp, it may be seen as a simplified version of that.
- I's wish to thank all the authors of draft-ietf-tsvwg-natsupp, especially Michael Tüxen, and Magnus Westerlund for the comments and suggestions.