Supporting End-to-End Connectivity in Federated Networks using SDN

Scenario & Goal
- Federated networks: independent but collaborating partners that share resources; promising approach but limited technological progress.
- Software-Defined Networking (SDN): established approach, mostly applied within data centers; most adopted specification: OpenFlow.
- Leverage SDN on Customer Premises Equipment (CPE) to support connectivity between the private networks of federated partners.
- Use simple Network Address and Port Translation (NAPT) to comply with standard IP-based Internet routing.

Approach (described only for IPs: TCP/UDP; port handling is analogous)
Each ISP runs an IP backbone and has an SDN controller that a) knows customers' private address spaces, b) has a pool of public IPs, and c) manages CPEs. Controllers can speak with each other.

1) (Optional) Controllers exchange customers’ locations and private IP address spaces (IP_Map).
2) A (TCP/UDP) data packet is received by CPE1, which sends it to controller C1. The latter picks a public IP used to map the packet's source IP, and locates the recipient's controller C2 (Optional: using Find_Ctrl to query a directory service).
3) C1 notifies C2 of the binding between the source private IP and the public IP. C1 asks C2 (Map_Request) for another mapping to be used for the destination IP.
4) C2 installs IP address translation rules on CPE3 (FlowModification), waits for confirmation (Barrier Request), and acknowledges C1 (Map_Reply or ACK). C1 then installs IP translation rules on CPE1.
5) Data packets can travel from Customer 1 to Customer 3 without further involving the controllers. Translations for packets traveling in the opposite direction can be set up similarly.

Unlike standard NAPT...
...we alter source AND destination IP/port ...
...and restore them at the remote end (like tunnels, without encapsulation)
No need for "NAT traversal/punching"

Experiments
Queue-based simulation to assess end-to-end connection establishment latency, using data from [4] and Pareto service times. Functional tests in Mininet + Open vSwitch.

3 address mapping strategies:
- Client Port Preservation (CPP): map source IP only, unless port collisions occur
- Lazy Address and Port Selection (LAPS): map also port
- Client Announces Port Selection (CAPS): IP/port mappings decided by the sender's controller (fewer messages needed)

Future Work
Overlapping customer address spaces, inter-controller signaling protocol + cache, other port mapping strategies, PoC

Connection establishment latency (ms)

<table>
<thead>
<tr>
<th>CPP Strategy</th>
<th>CAPS Strategy</th>
<th>LAPS Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Machines in Customer Site 1 (with a private IP addressing plan), attached to ISP1, may want to communicate with machines in Customer Site 3 (with another private IP addressing plan), attached to ISP2.