Preliminary experiments with Multipath-TCP VPNs

Claudio Pisa
clauz@ninux.org
@cl4u2

Wireless BattleMesh v8
Maribor 2015
Multipath TCP

- RFC 6824 - TCP Extensions for Multipath Operation with Multiple Addresses
  - Use multiple paths to:
    - Improve resource usage
    - Improve resilience to network failure

- Needs to be supported on the connection endpoints
<table>
<thead>
<tr>
<th>IPv6 Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001:4c00:893b:b055::/64</td>
<td>Subnet Emix (Libera non specificata)</td>
</tr>
<tr>
<td>2001:4c00:893b:ba3::/64</td>
<td>Subnet Nodo BAM</td>
</tr>
<tr>
<td>2001:4c00:893b:baba::/64</td>
<td>Subnet Pierluigi S.Nemesio</td>
</tr>
<tr>
<td>2001:4c00:893b:bb::/64</td>
<td>Subnet LuX Acacie</td>
</tr>
<tr>
<td>2001:4c00:893b:bee::/64</td>
<td>Matteo zip zap Grottaferrata test x MOBILE IPV6</td>
</tr>
<tr>
<td>2001:4c00:893b:beef::/64</td>
<td>Subnet Niccolo - GilgaMesh</td>
</tr>
<tr>
<td>2001:4c00:893b:c0ca::/64</td>
<td>Subnet Emix emixDelta Gibilmanna</td>
</tr>
<tr>
<td>2001:4c00:893b:c1c0::/64</td>
<td>Subnet SansPapiers</td>
</tr>
<tr>
<td>2001:4c00:893b:caca::/64</td>
<td>Subnet Stefano Consoli</td>
</tr>
<tr>
<td>2001:4c00:893b:cacb::/64</td>
<td>Hotspot Stefano</td>
</tr>
<tr>
<td>2001:4c00:893b:caf3::/64</td>
<td>Subnet Clauz</td>
</tr>
<tr>
<td>2001:4c00:893b:cafe::/64</td>
<td>Subnet Clauz tetto-balcone+hotspot</td>
</tr>
<tr>
<td>2001:4c00:893b:ca5a::/64</td>
<td>Subnet HispanicoHome</td>
</tr>
<tr>
<td>2001:4c00:893b:ccc2::/64</td>
<td>Subnet SigNodeHome</td>
</tr>
<tr>
<td>2001:4c00:893b:ccc3::/64</td>
<td>Subnet SigNodeGarden</td>
</tr>
<tr>
<td>2001:4c00:893b:cec0::/64</td>
<td>Subnet Cowabunga</td>
</tr>
<tr>
<td>2001:4c00:893b:d10::/64</td>
<td>Subnet Crucis</td>
</tr>
<tr>
<td>2001:4c00:893b:d102::/64</td>
<td>Subnet Domini</td>
</tr>
<tr>
<td>2001:4c00:893b:dad0::/64</td>
<td>Subnet Longinus</td>
</tr>
<tr>
<td>2001:4c00:893b:dead::/64</td>
<td>Subnet Mara Cairoli</td>
</tr>
<tr>
<td>2001:4c00:893b:deadf::/64</td>
<td>Hotspot Mara</td>
</tr>
</tbody>
</table>
Ninux Rome

• IPv4 + IPv6 networks
  – Uplinks:
    • BGP peerings (both IPv6 and IPv4)
    • DSLs (IPv4 only)
  – Multipath routing already in place!

• By using MultiPath-TCP VPNs perhaps we could:
  – aggregate uplink bandwidth
  – have more resiliency
Binder: A System to Aggregate Multiple Internet Gateways in Community Networks

- Aim: use different community network internet gateways, for:
  - Bandwidth aggregation
  - Load balancing
  - Fault tolerance
- Relies on:
  - OpenVPN over MultiPath-TCP
  - Loose source routing
Figure 1: (a) Binder components in a community network context; (b) Binder software architecture.
Why TCP Over TCP Is A Bad Idea

A frequently occurring idea for IP tunneling applications is to run a protocol like PPP, which encapsulates IP packets in a format suited for a stream transport (like a modem line), over a TCP-based connection. This would be an easy solution for encrypting tunnels by running PPP over SSH, for which several recommendations already exist (one in the Linux HOWTO base, one on my own website, and surely several others). It would also be an easy way to compress arbitrary IP traffic, while datagram based compression has hard to overcome efficiency limits.

Unfortunately, it doesn't work well. Long delays and frequent connection aborts are to be expected. Here is why.

- **TCP over TCP** is a bad idea
  - The *upper TCP* assumes that packet loss will not be handled by the layers below
- **TCP over wireless** is also a bad idea
  - TCP assumes that packet loss is caused by congestion
  - Packets lost due to interference trigger congestion control mechanisms
- **TCP over TCP over wireless** looks like a *very bad* idea

- Loose source routing is deprecated
- Need for an “external” aggregating server
- Binder has been tested in an emulated environment only (AFAIK)
MP-TCP-based VPN

• But: just a few tools are needed and they are easily available:
  – MP-TCP kernel
  – OpenVPN (or other VPNs that can use TCP)

• So why not give it a try?
MP-TCP-based VPN
MP-TCP-based VPN

• Testbeds:
  – GNS-3 emulation
    • Different link speeds and latencies (~5-10Mbps, ~10-100ms delay)
  – Real World testbed
    • (slow ~8Mbps) Wi-Fi connection, 3G connection, ethernet

• Tried:
  – OpenVPN (over TCP)
  – tinc-VPN (over TCP)
  – vtund (TCP mode)
  – ssh -w
  – socat
MP-TCP-based VPN

• Disclaimer:
  – Only some very preliminary tests were performed!

• Tested using UDP (iperf)
  – To work around TCP-in-TCP effects

• Observations:
  – Packets are going through different links at the beginning, but then they use one link only
    • The one with the smallest latency
  – Instability
    • In the emulated network packets sometimes used both links (with vtund!)
      – But: both links used at the speed of the link with the smallest latency
        • BDP does not count!?

• Is userspace processing increasing latency too much?

• Is it really worth investigating further?
So...

- VPNs over MP-TCP seem wrong anyway
  - Worth trying just because all tools are easily available
- Other ways to go:
  - Migrating operating systems to multipath TCP?
  - Stop using TCP sockets and have application-level multipath support?
- An UDP-based multipath VPN sounds great!
  - It could get rid of TCP-specific issues
    - TCP in TCP
    - TCP over wireless
  - No need to change end host operating systems or applications
  - Perhaps it could be based on existing source code:
    - MP-MOSH ?
    - QUIC ?
Thank you
References

- MultiPath TCP: http://multipath-tcp.org/
- http://wiki.ninux.org
- Why TCP over TCP is a Bad Idea http://sites.inka.de/bigred/devel/tcp-tcp.html