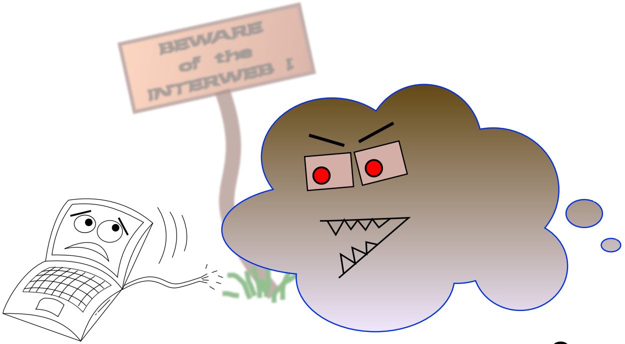
The difficulties of a peer-to-peer VPN on the hostile Internet

Brussel, February 6, 2010



Guus Sliepen
guus@tinc-vpn.org

Tinc development started in September 1997, after introduction of ethertap in Linux 2.1.53.

Current features:

- Connects multiple sites together
- Can act as router (layer 3) or switch (layer 2)
- Full support for IPv6
- No central server
- You configure some endpoints, tinc will do the rest

Modus operandi:

- Metadata exchanges via TCP
- VPN packets directly via UDP
- Fall back to TCP if UDP is not possible

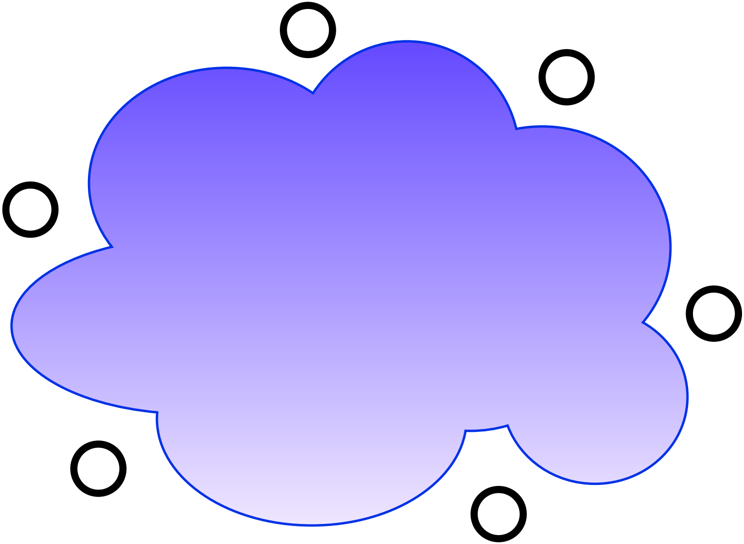
The competition:

- CIPE†
- VTun†
- IPsec
- OpenVPN
- Hamachi

But also:

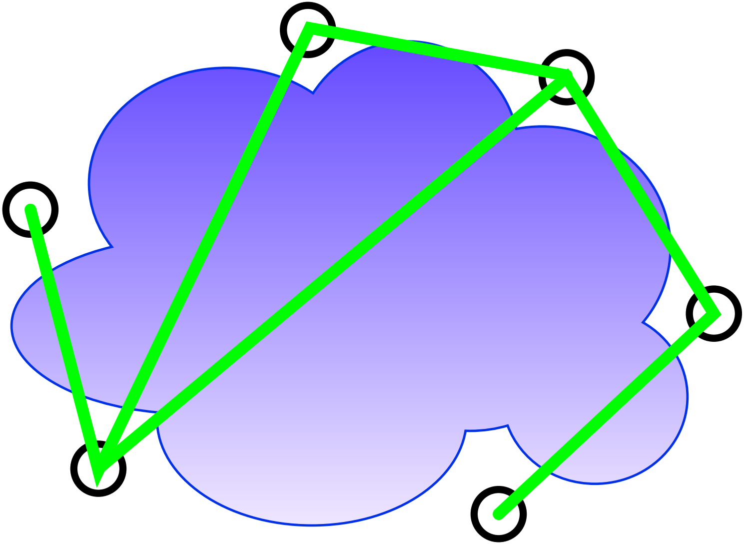
- GVPE
- CloudVPN
- SocialVPN
- n2n
- VDE

Network before VPN is configured:



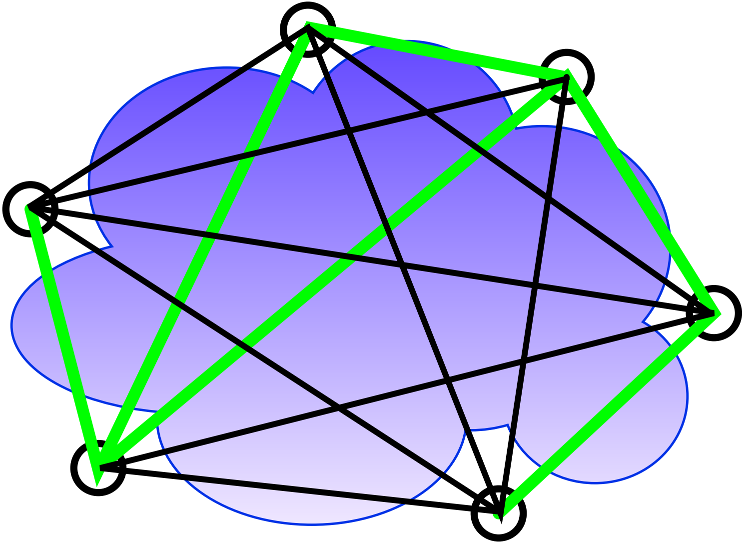
Blue cloud: the Internet
Black circles: VPN nodes

Initial connections configured by user:



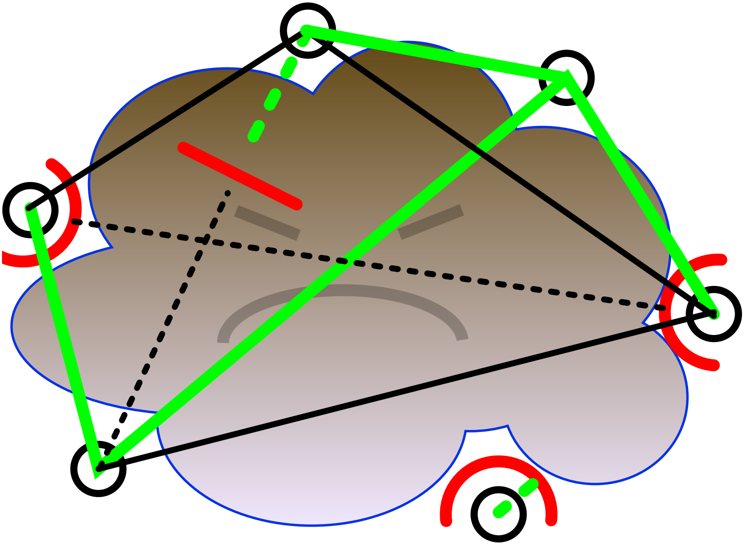
Green lines: initial connections

Full mesh created by tinc:



Black lines: UDP tunnels

Reality is not so nice:



Red arcs: NAT
Red line: ISP blocking traffic
Dotted lines: failed connections

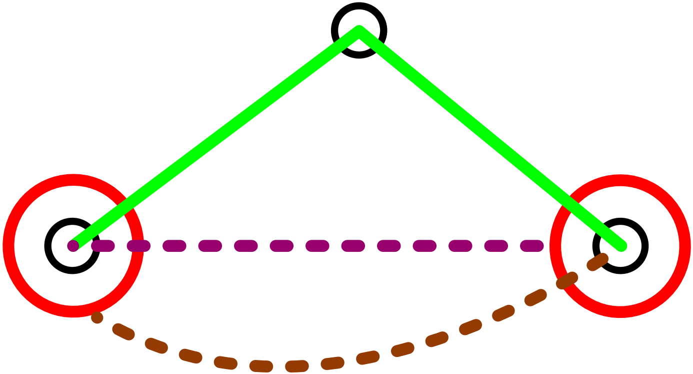
The problem of NAT:

- Source address and port change
- Incoming connections blocked

Solutions:

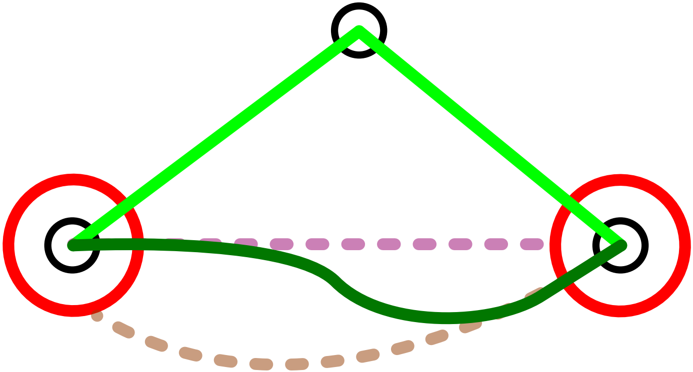
- Routing via non-NAT node (not efficient)
- Port forwarding (not always possible, manual work)
- UPnP (needs router support, complex)
- STUN/ICE (not always possible, complex)

Two nodes behind NAT:



Both nodes can talk to a third node.
NAT changes ports, nodes cannot talk to each other.

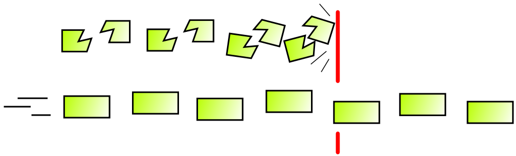
STUN in action:



Third node tells other nodes about their addresses and ports.
Nodes connect to each other using this information.

The problem of packet fragmentation:

- MTU inside tunnel smaller than outside
- Outer layer fragments bad for performance
- Some firewalls/ISPs block fragments

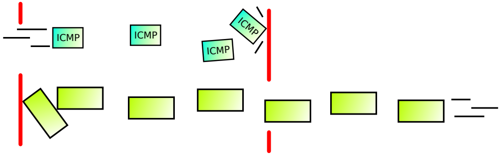


Solution:

- Determine path MTU between nodes
- Generate ICMP Fragmentation Needed packets
- Should work for all IPv4/IPv6 traffic
- Fall back to TCP for other traffic

The problem of firewalls/ISPs blocking ICMP:

- ICMP Fragmentation Needed does not work!
- Happens when network traffic leaves the VPN (for example, when having default gateway on VPN)



Solution:

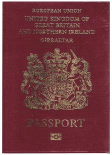
- Clamp MSS field in TCP packets to path MTU
- Works only for TCP

Other problems:

- Frequently changing IP addresses
  - use dyndns
  - cache & forward known addresses between nodes
- Only allowing certain ports, like HTTP
  - tunnel over ICMP/DNS/HTTPS
- ISPs dropping/delaying small UDP packets
  - because they think it's VoIP!
  - severely slows down TCP streams inside tunnel

Authentication and authorization

- Authentication = proving who you are
- Authorization = proving you are allowed to do something



Two well known (mostly authentication) methods:

- X.509 certificates
  - centralized approach
  - focused on identities (LDAP like), and URLs
- OpenPGP keys
  - decentralized approach
  - focused on email addresses

We need OpenPGP-like features:

- Completely decentralized
- Web of trust

We need more than OpenPGP can offer:

- Authorise anything, not just email
- Everyone can add/remove authorizations
- Negative authorization: forbid things
- Group decisions

libfides: lightweight p2p authorization framework

- Create and maintain repository of many certificates
  "X said at time T that Y is allowed to do Z"
- Newer certificates overrule older ones
  "X said at time T+1 that Y is not allowed to do Z"
- Make it simple to query the repository
  "Is Y allowed to do Z?"

- Fast & easy synchronization of repositories
- Application does not need to know about crypto
- Libfides itself uses only ECCDSA primitives

- Still in alpha stage.

Conclusions:

- The Internet eats your packets.
- Lots of techniques necessary to work around it.
- Distributed authorization is a challenge.

That's it.

- Questions?

Visit the website:

`http://tinc-vpn.org/`