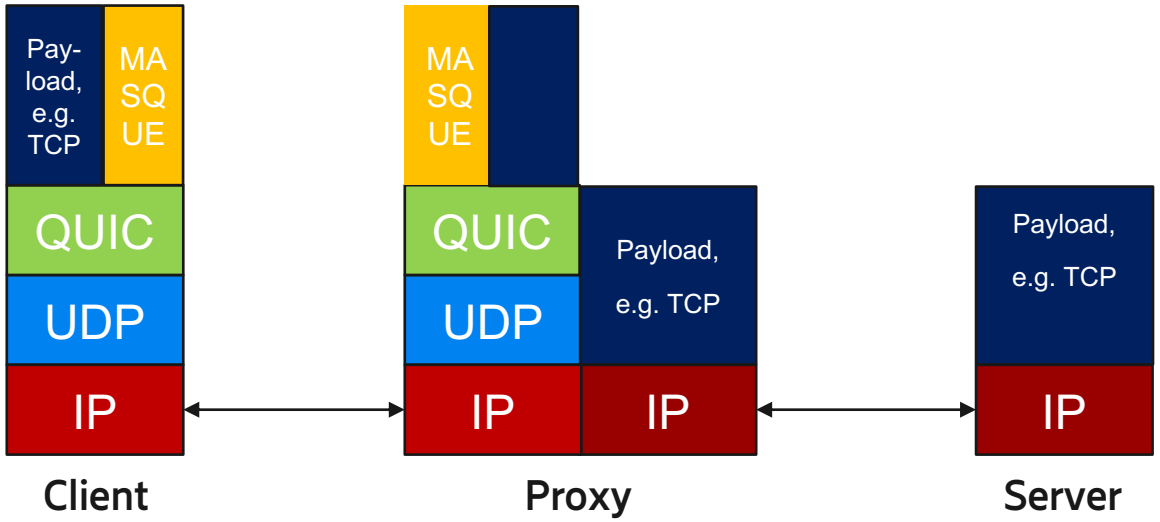
Different approaches for IP proxying

Mirja Kühlewind
Magnus Westerlund
Marcus Ihlar
Zaheduzzaman Sarker

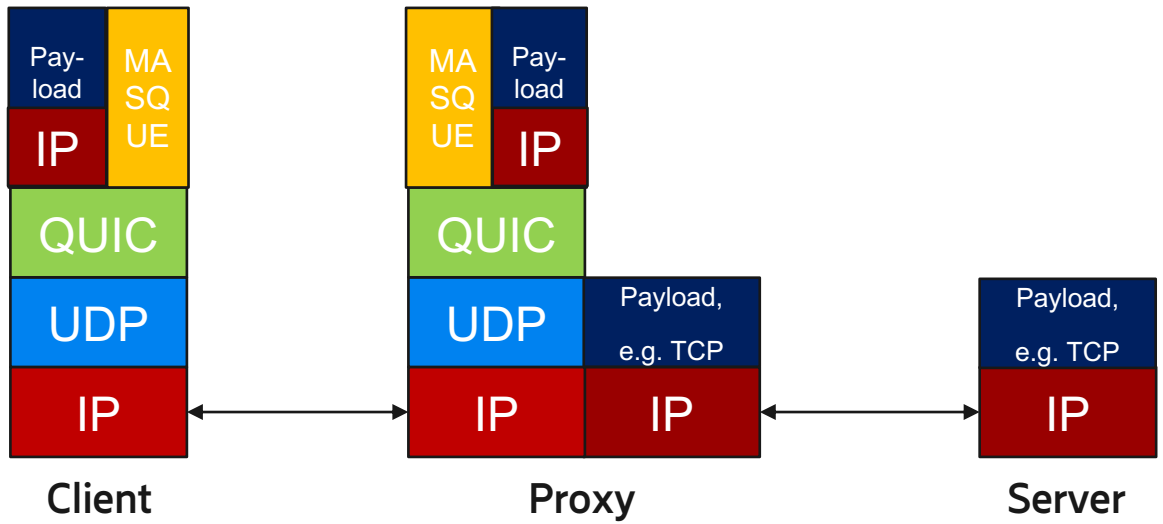
Two possible design approaches

In draft: IP payload forwarding



- Client only provides target IP address (and other relevant information) with CONNECT-IP request
  - Goal: reduce packet overhead
  - Note: Reuse of functions needed for CONNECT-UDP
- Proxy constructs and adds IP header/selects src IP address
- Stateless forwarding of incoming traffic not considered (might be needed for network-to-network use case)

Alternative: IP packet (incl. header) forwarding



- IP header is part of the QUIC tunnel payload
- Easier for Network-to-Network: client provides IP range
- Need for source address validation (or NAT)
- Additional signaling needed for route negotiation for prefixes

Requirements on IP Proxying from draft-ietf-masque-ip-proxy-reqs-01

- **Proxying of IP packets: "**The Data Transports MUST be able to forward packets in their unmodified entirety, although extensions may enable the use of modified packet formats (e.g., compression)."
  - What the reason for making this a MUST? Which function is prohibited if this is not supported? Why should any kind of compression not be part of the core protocol?
- **IP Assignment: "**The client will be able to request to be assigned an IP address range, optionally specifying a preferred range." "For symmetry, the server may request assignment of an IP address range"
  - This covers the network-to-network case. Is this part of the core protocol or an extension? What's about requirements on address validation?
- **Route Negotiation: "**At any point in an IP Session (not limited to its initial negotiation), the protocol will allow both client and server to inform its peer that it can route a set of IP prefixes. Both endpoints can also request a route to a given prefix"
  - What's the use case for this requirement? Does this need to be part of the core protocol?
- **Support HTTP/2 and HTTP/3:** „The protocol SHOULD also support HTTP/2 [H2] as a fallback"
  - Do we have consensus on this? This is noted in the charter as "to consider" but we might need more discussion.

Extensions for IP Proxying from draft-ietf-masque-ip-proxy-reqs-01

- **Reliable Transmission of IP Packets**
  - As datagram support is optional and TCP fallback would only provided an reliable service, client should be able to indicate use of reliable streaming mode as part of the core protocol.
- **Data Transport Compression**
  - Why is this required to be an extension? Is there is an easy way to reduce overhead, that should be considered as part of the core protocol.

Non-Requirements on IP Proxying from draft-ietf-masque-ip-proxy-reqs-01

- **Non-requirement – Address Architecture: "**Similarly, "ownership" of an IP range is out of scope. […] Whether or not to trust this information is left to individual implementations and deployments."
  - Basic address validation should be required for traffic that is routed on the public Internet, e.g. check on address spoofing and return routeability.
- **Non-requirement - Translation: "**Some servers may wish to perform Network Address Translation (NAT) or any other modification to packets they forward. Doing so is out of scope for the proxying protocol."
  - MASQUE should support a way to expose the outfacing IP to the client (if NAT is done by proxy); further client should be able to require NAT for address obfuscation use case. This should be part of the core protocol and added as explicit requirements.

Requirements and open issues that also apply for CONNECT-UDP

- **Maximum Transmission Unit:** "The protocol will allow endpoints to inform each other of the Maximum Transmission Unit (MTU) they are willing to forward." (also issue #7 CONNECT-UDP draft)
  - E.g. use of GET/POST-based signalling to exchange configuration files
- **Extensibility: "**Once the session is established, the protocol will provide a mechanism that allows reliably exchanging vendor-specific messages in both directions at any point in the lifetime of the IP Session."
  - Per-packet information: Extension to HTTP datagram frames.
  - Per-flow information: Use of GET/POST scheme to exchange configuration files
    - Alternatively: use new HTTP control frames to be interleaved with data on forwarding stream