UNIVERSITY OF ABERDEEN

IETF-75

The UDP Tunnel Transport mode

draft-fairhurst-6man-tsvwg-udptt-01 (21-Jun-09)
(Individual Submission)

Gorry Fairhurst

SCHOOL OF ENGINEERING
UNIVERSITY OF ABERDEEN

Part 1: Transport Checksum for IPv6

In IPv4 checksum is not required

- Endpoint association protected by IP checksum
- Still is recommended [RFC 5405]

IPv6 [RFC 2460] mandates transport checksum

- No IPv6 header checksum

Why do people some want to change?

- **AMT, Automatic IP Multicast Without Explicit Tunnels**
  - draft-ietf-mboned-auto-multicast-09
  - AMT outer checksum protects only outer IP & UDP header, type, and Nonce
  - Issue is mis-delivery to standard UDP stacks

- Desirable:
  - No Checksum computation at sender/receiver
  - UDP-like traversal of middleboxes (header value 136)

UNIVERSITY OF ABERDEEN

Effect of corruption

- Errors could...
- Cause packet to go in wrong direction, or to wrong port
  Such packets should be discarded
- Cause the inner packet to become corrupted
  Such packets should be discarded

Nice to make ***wrong*** endpoint do the checksum,
but avoid processing on ***actual*** tunnel endpoints...

UNIVERSITY OF ABERDEEN

What are the options?

- Simplest solution is to **require use of UDP**
  - Safe and no standards action needed
  - Needs to calculate checksums
- Change to allow **UDP with zero checksum**
  - Would need to update IPv6 base standard
  - Tunnel hosts need to update nodes (?) and middleboxes (?)
  - Can we *really* assume this will only be used by “router boxes”?
  - Need to ensure this is only used for tunnels
- Change to allow **UDPTT mode (“fixed” checksum)**
  - Would need to update IPv6 base standard
  - Tunnel hosts need to update nodes (?) and middleboxes (?)
  - Need to ensure this is only used for tunnels

Part 2: UDPTT Header Format

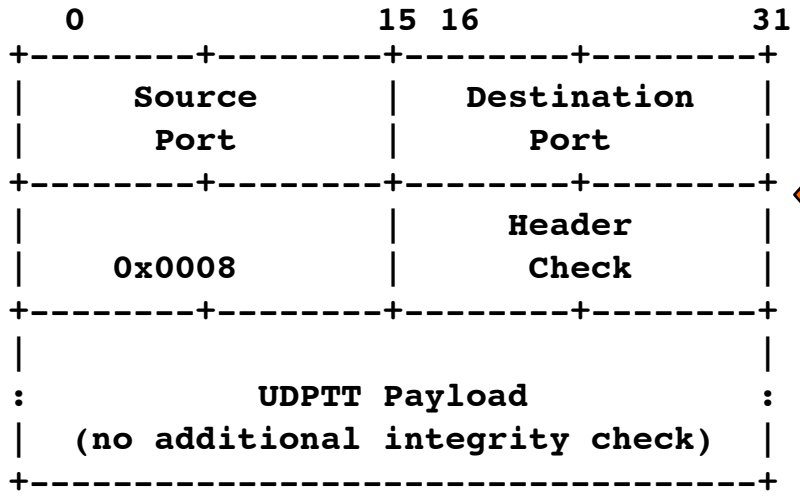

Figure 1: UDPTT Header Format

Fixed Header per Flow

UDPTT pseudo-header for IPv6:

- 128b Source Address
- 128b Destination Address
- 32b 0x0000000008
- 24b zero
- 8b Next H value

Two length fields: in IP and Transport

- **Three receiver behaviours:**

- 1 Standard checksum calculation
  - If “UDP” length used, **could truncate with no payload**

- 2 Standard checksum calculation
  - If “corrected IP” length used, **would be OK**

- 3 Checksum used IP length rather than UDP length
  - Fails, **discarded** (Non-compliant to RFC 2460)

Conclusion: What Next?

I'll make another rev. of the UDPTT spec.

The author thinks this is ready for WG consideration!

I'd love to receive feedback on the spec.

Will start a thread on ***6man*** to discuss IPv6 base spec change

I'd love to receive feedback on the issues.

Spare Slides (for 6man)

New in Revision -01

- **Updated text:**

  3.2. Requirements for Tunnelled Protocols

  3.3. Backwards compatibility with RFC 2460

  3.1 UDPTT Usage Guidelines

  6. Security Considerations

  Appendix B. Applicability for AMT

- **Currently known remaining issues:**

  Middleboxes /SHOULD/MUST/ NOT truncate IPv6 datagrams

  Specify simple API (sockopt)

  ??? v4-v6 protocol translation (PT)

UNIVERSITY OF ABERDEEN

What may middleboxes do?

Most NATs adjust transport checksums and don't (re) compute.

Some middleboxes automatically drop zero checksums.

Some middleboxes may correctly forward UDPTT

But ... there are many variants!