Passing Errored-Packets to Applications

IETF-57 IAB Plenary

Aaron Falk
Allison Mankin

The Problem: Real-time apps are hungry for bits

- **Some links have "variable" error rates**
  - E.g., cellular, wireless
- **Some applications can tolerate bit errors in data**
  - E.g., voice codecs
- **But, transport protocols traditionally checksum the entire packet**
  - Errors anywhere result in packet discard
- **So, some folks would like the ability to pass packets with errors to the application**

Traditionally, IETF protocols don't pass data with known errors.

Thus, an architectural discussion has ensued...

Can links pass errored-packets?

- **One view:**
  - Today's link technologies are so good that either all or none of the bits in a packet get through
- **Another view:**
  - Some links use FEC that protects packet headers differentially
    - E.g., 3GPP (deployed)

IPv6 interactions

- **One view:**
  - IPv6's lack of checksum should make the use of a transport checksum mandatory
- **Another view:**
  - Transport protocols may be crafted to provide partial/modular checksum coverage

Encryption & authentication fail

- **One view:**
  - One man's errored-packet is another man's spoofed data; errored-packets will fail authentication & decryption
- **Another view:**
  - Some use cases don't require security; there are encryption schemes which *are* bit-error tolerant

Congestion vs. Corruption

- **One view:**
  - CC algorithms will be able to better distinguish between congestion and corruption.
- **Another view:**
  - We don't have a clue on how to respond to corruption. In particular, we don't know when corruption is, in fact, an indication of congestion.

Differential protection of headers has some challenges…

- **IP options, encapsulated headers result in protection of variable regions**
  - Links would need to become “transport-aware”
- **Therefore, all IP packets are not treated equally**
  - Process cycles at link interface may be prohibitive

However, lacking a solution…

- **Today, some users are running UDP with checksum disabled**
  - Port #, header info may be corrupted
  - Receiver doesn't need to agree for this to happen
  - Potential BIG problems with IPv6 – no IP checksum
  - VoIP packets of this nature have been observed in the wild

So, is this a good idea?

Discussion is happening in the Transport Area

Two proposals:

- **draft-ietf-tsvwg-udp-lite-01.txt**
  - In IESG review
- **draft-ietf-dccp-spec-01.txt**
  - Near WG last call

Opportunity for discussion tonight...

Thank you.