QoS and Security

→ QoS in TCP / UDP apps

Problem with TCP: performs poorly in ad-hoc (vehicular networks, better for wire-line networks, assumes all losses are due to congestion.

It has loss detection, congestion control mechanisms.

However mobile networks have:

- mobility
- high bit error rate
- unpredictability / variability
- contention
- poor performance for long connections.

Delays cause TCP to send unnecessary Retransmissions, being more inneficient.

→ TCP Cubic

Cubic takes into account the time that has elapsed since the last congestion, allowing from faster recovery and greater throughput.

→ QUIC

It is built on top of UDP to improve performance by reducing latency.

Handshake combines negotiation of cryptographic (TLS) and transport parameters (uni and bidirectional streams, stream flow and connection flow control).

RTT: time for data packet to be sent + receive ACK of that packet.

Uses separate sequence numbers for data and packet delivery.

Lower RTT indicates better network conditions, and it's ability to manage multiple streams independently provides more accurate and responsive RTT estimation.

→ TCP Vegas

Senses congestion before any packet loss, decreasing window size.

Uses the difference between the expected rate and actual rate.

It has a modified slow-start and new Retransmission.

Note: TCP variants try to improve performance by estimating the available bandwidth or exploiting buffering capability.

↳ uses explicit route failure notifications
↳ TCP sender doubles retransmission timout
↳ avoid timeout and unnecessary retransmissions
↳ requires assistance from intermidiate nodes (to find path).

→ QoS in UDP

Trade-offs:

- intserv in multi-hop wireless network:
  - hard to estimate available resources
  - hard to do resource reservation
  - resource reservation is pinned to a route.

- diffserv " ".
  - hard to do admission control
  - hard to maintain assurances

→ QoS Routing

Essential component for QoS.

Can inform source node of bandwidth, QoS availability of destination and path

Add QoS requirements in routing metrics:

- difficult route maintenance
- overhead
- reserved resources not guaranteed
- responsive to nodes mobility

-> QoS for AODV

Adds extensions to route messages.

Node that receives can only forward if it meets QoS requirements.

Routing tables need to be updated (fields for delays, bandwidths)

Delay: maximum delay extension

list of sources requesting delay guarantees, used during route discovery

Bandwidth: similar to delay

REQ can include both

Loosing QoS: if node detects QoS cannot be maintained, generates QoS_LOST to depending nodes

Reasons: increased load of node

→ Transmission Quality (Batman)

Add local link quality in the TQ value

$$TQ = TQ(incoming) - TQ(local)$$

→ QoS for OLSR

Multi-point relays are the nodes with best QoS.

QoS depends on available bandwidth, % of nodes on other street and lane weight.