Individual Goals - Updated version

EITN30: Internet Inside

Moa Eberhardt - mo5385eb-s
Abdalsalam Muhialdeen - ab5015mu-s

Date: 2024-05-17

LUNDS UNIVERSITET
Lunds Tekniska Högskola

1 Goals

Our main goal consists of exploring three different concept areas of our system and their impact on the system's performance. We would like to use and compare different solutions within each area to hopefully conclude the better option in a systematic and educated way. This can be done through experiments that include implementing different strategies or configuring parameters that are then evaluated using reasonable metrics. More concretely we would like to accomplish a throughput within the range of 300-500Kbps and a latency of 30-100ms by the end of our experiments.

UPDATE:
The evaluation document contains more detail on the resulting parameters of our system. In short, we accomplished a throughput of 400Kbps and a latency of 30ms which are within our wanted intervals.

1.1 Queuing Strategy

To find the most suitable queuing strategy to handle the points of the system that might clog and overload to minimize package loss at the transaction points between layers.

UPDATE:
We chose a strategy that minimizes the package loss on behalf of the latency and focused on improving latency in other areas of the system. The final implementation is a thread-safe queue class that acts as a buffer between tx/rx threads and our virtual interface (TUN).

1.2 Fragmentation

We want to explore the possible ways of breaking down data sent from higher layers with bigger MTU then the current layer while minimizing the impact on the system. This would probably entail a custom header for the radio packages where a number of bytes can be assigned to the control plane purposes.

UPDATE:
We ended up using 1 byte as a control-byte header and used the 31 remaining bytes to fragment the incoming virtual interface packages. Given that we use a certain fixed MTU, a range of values inside our control-byte is used for sequencing the fragments while still leaving enough values for control plane purposes, such as signifying the end of a IP packet (MAXBITS) and canceling packet transmission when a fragment is lost (CANCELBITS).

1.3 Achieve parallel ARQ on both channels

To make the system more efficient, we would like to reason about if the built-in ARQ can be used effectively or if our own implementation of a ARQ is better.

To achieve this goal, several packages/fragments might be sent simultaneously over the channel. Relating the queuing strategy, the packages will be segmented, each package will have a number

given and then be placed into a list by their number. The ARQ needs to handle such cases where fragments are sent/received out of order by re-transmitting and re-ordering fragments.

UPDATE:
We tried implementing our own ACK-threading that uses the queue strategy to send the ACK packages. However, the result at the time being did barely differ so we chose to put our focus on the two other goals to get the system running as smoothly as possible using the radio's built-in Auto ACK.