Path Selection in Storage-Aware Ad-hoc Routing

This article describes a generalized storage-aware metric for ad-hoc networks that captures the end-to-end latency associated with transferring a large block through the network. This metric can be used during path selection in an ad-hoc routing protocol.

1 Generalized Storage-Aware Metric

The following are assumptions made:

- Blocks are transferred in a hop-by-hop fashion
- Routers have storage that they allow other nodes in the network to utilize
- The network layer has access to some type of link quality information for the links adjacent to it
- The network is generally connected; e.g., most of the time, instantaneous end-to-end paths exist

As a block traverses a path, each node along the path will perform the following steps, each of which increases the total end-to-end latency of the block:

1. Store the block until ready to send
2. Attempt to gain access to the channel
3. Transmit the block to the next node along the path

Assume nodes along an arbitrary path are labelled $1, 2, ..., k$ where 1 is the source and k is the destination. Let $BACK_i$ be the expected amount time a block will spend in storage at node i due to backpressure from the hop-by-hop transport. Let $HOLD_i$ be the expected amount time a block will spend in storage at node i due to a routing decision to store not related to forced backpressure. Let P_i be the probability that a routing decision to store (not due to backpressure) is made by node i. Let $CHAN_i$ be the expected amount of time a block will wait while node i gains access to the channel to send the block. Let T_i be the expected amount of time a block will be in transit from node i to node $i+1$.

The total end-to-end latency for a block over a path is:

$$delay_{1,2,...,k} = \sum_{i=1}^{k} (P_i \cdot HOLD_i + BACK_i + CHAN_i + T_i)$$

Storage time due to backpressure at node i will directly depend on the amount of data at node i with higher priority that the block in question that is also going to node $i+1$, in addition to how long it takes for space to open at node $i+1$. Making the simplifying assumption that blocks are served on a first-come-first-serve basis:

$$BACK_i = \frac{D_i}{tx_{i,i+1}} + TWS_{i+1}(D_i)$$

where D_i is the amount of data being held at node i due to backpressure, $tx_{i,i+1}$ is the transmission rate between node i and node $i+1$, and $TWS_{i+1}(D_i)$ is the time node i waits on node $i+1$ to clear out D_i space.

The transmission time, T_i, is:

$$T_i = \frac{blockSize}{tx_{i,i+1}}$$

2 Practical Instantiation for CNF

One approach taken by CNF is to hold data close to the source until the SETT path metric is reasonable in relation to the LETT path metric. The following simplifying assumptions to the model are for a particular path p.

- Blocks are held at the source, and after they are released they will not be held (except for queuing due to backpressure) at any intermediate node. Therefore, $P_i = 0$ for $2 \leq i \leq k$.
- $TWS_i = 0$ for $1 \leq i \leq k$, which is a result of holding the block at the source until the path becomes of normal quality.
- Channel access time is negligible, since the block sizes are quite large.
- All storage buffers in the network are large enough to handle any valid block size when empty

The current values of D_i and $tx_{i,i+1}$ can be estimated by node i and transmitted to all other nodes via LSA flooding. Furthermore, the source node can estimate P_1 in the following way. $P_1 = 0$ if the current SETT path metric is normal or low relative to the LETT path metric. $P_1 = 1$ if the current SETT path metric is high relative to the LETT path metric. Furthermore, $HOLD_1$ is the average amount of time for that path that the SETT is relatively high. This can be estimated by proactively keeping track of the following: given the SETT is high, on average how long does it take to become normal or low?

With this information, a source node has everything needed to compute the delay metric for any given path. This metric can be used to select the best path from numerous ones. This particular instantiation of the metric is therefore:

$$delay_{1,2,\ldots,k} = P_1 \cdot HOLD_1 + \sum_{i=1}^{k} (BACK_i + T_i)$$

where D_i and $tx_{i,i+1}$ are estimated from received LSA message and $TWS_i = 0 \ \forall i$.