Wi-Fi Mobility without Fast Handover

Andrei Croitoru (student), Costin Raiciu and Dragos Niculescu University Politehnica of Bucharest

1. INTRODUCTION AND MOTIVATION

As mobile traffic keeps growing, offloading to Wifi has been an obvious solution for sometime. Ideally, in dense areas, mobiles should just use Wifi deployments with little help from the cellular network. This is easier said than done: while cellular data works in true mobility scenarios, WiFi is currently mostly a static connectivity solution. Although it offers mobility mechanisms, they are not efficient and the mobile experience is poor despite dense WiFi deployments in urban areas.

The current mobility mechanisms for WiFi rely on fast handover. Research has been done in this area that allows for very short handover times when exiting the range of an access point (AP), improving the user experience by reducing delays. However, we think this is the wrong approach, for many reasons: a) To start the handover mechanism, a client has to lose connectivity to the AP. b) There is no good way to decide which of the many APs to associate with for best performance. c) Once the decision has been made, there is no way to dynamically adjust to changes in AP signal strength.

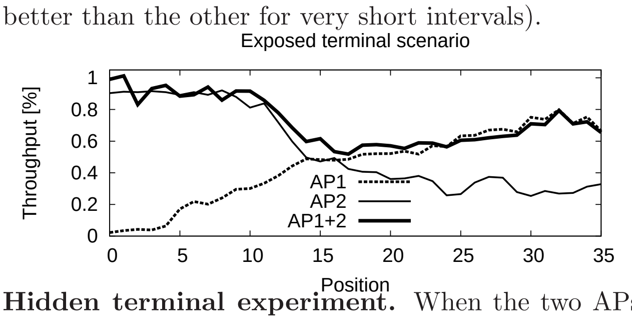
2. APPROACH

Consider a radically different approach: instead of using only one AP at a time, **mobile clients should connect to all APs at any given time**. We rely on the recently adopted MPTCP protocol to spread data across all the APs, with one subflow per AP.

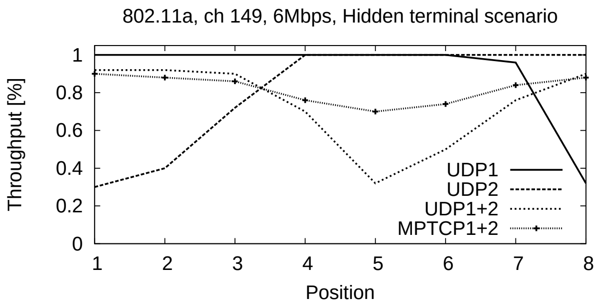
Consider what happens if the wireless NIC is restricted to use a single channel (e.g. 1,6 or 11 in the 2.4Ghz range). If we disregard WiFi interference between APs, the theoretically optimal solution would be to always connect to every visible AP, as MPTCP will handle load balancing at the transport layer: if an AP has poor signal strength, the bytes will simply migrate to the APs with better connectivity to the client. This way, handover delays are eliminated and we offer a continuous connectivity solution to the mobile client.

Interference, of course, can be a major issue. In the context of using a single channel, we ran two experiments to understand the practicality of our approach.

Exposed terminal experiment. This case appears when there are two APs on the same channel within Carrier Sense range, and a client connects to both APs. We have tested this use case by moving our laptop from one AP to the other in discrete steps and measuring the throughput of an MPTCP connection through both APs; results are given in the figure below. When the client is at halfway distance between the two APs, the throughput obtained through MPTCP is higher than the throughput obtained through either AP individually. After careful investigation of TCP state variables and WiFi retransmissions, we believe this effect is due MPTCP spreading data over two APs which allows the latter to exploit channel diversity (one channel being better than the other for very short intervals).



Hidden terminal experiment. When the two APs are outside of CS range and the client is connected to both, the frames coming from the two APs will collide at the client, greatly harming throughput. We have tested this use case in a similar fashion to the previous one, and the results can be seen below. As we expected, the combined UDP throughput of two simultaneous iperf sessions is greatly diminished by the hidden terminal situation. However, by running two simultaneous MPTCP subflows, the combined throughput is surprisingly good. This is due to interactions between WiFi and the congestion control mechanisms of TCP which leads to a capture effect, whereby one subflow get zero throughput for long periods of time while the other is sending at full rate. We have confirmed this effect in a number of scenarios, with and without RTS/CTS enabled at the APs, and also in ns2 simulation.



3. DISCUSSION AND FUTURE WORK

Connecting to many APs simultaneously seems a viable mobility solution within a single channel. A complete solution must support multiple channels. We are considering two approaches: a) use multiple Wifi NICs, one per channel: offers best performance at high energy costs, or b) use channel switching to emulate multiple NICs, e.g. as in FatVAP: should offer reasonable performance at smaller energy consumption.