LOAD AWARE HAND-OFFS IN SOFTWARE DEFINED WLANS

ANIL KUMAR RANGISETTI, B HARDIK, B PRADEEP KUMAR, and T BHEEMARJUNA REDDY, IIT Hyderabad, India
Outline

- Introduction
- Existing Work
- Proposed Work
  - Load Aware Hand-off Algorithm for Software Defined WLANs
- Testbed Setup
- Experimental Results
- Conclusions and Future Work
Introduction

- **Software Defined WLANs**
  - APs are integrated with *Open vSwitch*
  - APs are integrated with handling User/kernel space (e.g. Click) software for handling Lower and Upper MAC functions

- **WLAN control algorithms runs on top of OpenFlow controller**
  - Example: Mobility, Hand-Off, Interference etc. algorithms

- **Central Information Base**
  - Maintains AP related information like AP load, operating channel, clients connected to each AP etc.
  - Neighbors of each AP
  - Client Specific attributes MAC, IP, traffic details

---

**Architecture of Software Defined WLAN System**
Existing Work

- ODIN is a programmable WLAN Arch.
  - It provided open source framework for developing SD WLAN algorithms
  - It simplified client association mechanism with Virtual Access Point (VAP) concept
    - Each connected client associated with LVAP (unique BSSID) on particular AP
    - It reduced Hand-off delays with ADD LVAP and REM LVAP messages
- But, it did not consider issues related to RSSI based clients association or load balance in WLANs with static clients
- A major limitation: all neighboring APs of WLAN need to operate on same channel
ODIN LVAP mechanism for Hand-Offs

Initially
Client
Connected to AP1 with PROBE, AUTH, ASSO Messg Exchange

Major issues:
Lot of Interference
Each AP depends on same Subscription for Hand-Off decision
No Network awareness for Hand-off decision
Load imbalance can result into network due to RSSI based decisions

Client Connection removed from old AP (REM_LVAP) and Moved to new AP with (ADD-LVAP)
Client need not send any RE-ASSOC MSGs
So it reduce Hand-Off delay
But major limitation: both AP should operate on Same Channel
Proposed Work

- ODIN architecture extended to support load-aware hand-offs in multi-channel WLANs
  - Hand-off decisions are defined based on RSSI and AP traffic loads
  - Fewer messages between APs and the Controller to gather required network information
  - IEEE Channel Switch Announcement (CSA) messages are integrated to the extended architecture to support APs operating on different channels
Load of an AP

- Load of an each AP is defined as Traffic Intensity (TI) [1].
- TI measures utilization of channel resources in a range from 0 to 1, where 0 indicates channel is idle and 1 indicates channel is fully occupied.
  - It's a lower bound on actual busyness as it does not consider failed packet transmissions and IFS times in between packet transmissions
- TI is defined as the amount of time the AP is busy (AP-BUSYTIME) with transmission or reception of N frames in a given time period T (in seconds).
- Its calculation includes all Data, Management, and Control MAC frames
- For each frame $d_i = \frac{\text{frame length}}{\text{PHY data rate}}$
- $TI(t) = \frac{\sum_{i=1}^{N} d_i}{T}$
- TI(t) is sent to the controller in PUBLISH_AP_LOAD messages
- Controller updates for each AP the load as CTI(t)
  \[ CTI(t) = 0.9 \times TI(t) + 0.1 \times CTI(t-1) \]
  - We have given 90% weightage to Current load to reflect sudden load changes at that AP

Network Awareness Messages

- PUBLISH-AP-LOAD message is for periodically sending AP load to the SDN controller
- PUBLISH-CLIENT-LOAD message is for reporting each client individual load
- PER-AP-HANDOFF-SUBSCRIPTION message is for changing AP initial subscription according to its new load
Load Aware Hand-off Algorithm

- loop
- Updates RSSI Threshold value to be matched against APs’ load level
- When it receives HAND-OFF request from an Competing AP
  - if $(\text{Load}_{\text{connected AP}} - \text{Load}_{\text{competing AP}}) > \text{HT}_{\text{LOAD}}$ and $(\text{RSSI}_{\text{connected AP}} < (\text{RSSI}_{\text{competing AP}} + \text{HT}_{\text{RSSI}}))$ – Condition 1
    - then Hand-off the client to the competing AP
  - else if $(\text{RSSI}_{\text{competing AP}} > (\text{RSSI}_{\text{connected AP}} + \text{HT}_{\text{RSSI}}))$ and $(\text{Load}_{\text{competing AP}} < (\text{Load}_{\text{connected AP}} + \text{HT}_{\text{LOAD}}))$ – Condition 2
    - then Hand-off the client to the competing AP
  - else
    - No Hand-Off
- end if
- end loop

<table>
<thead>
<tr>
<th>Load Level</th>
<th>RSSI Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>215</td>
</tr>
<tr>
<td>medium</td>
<td>200</td>
</tr>
<tr>
<td>low</td>
<td>185</td>
</tr>
</tbody>
</table>
How to ensure Load-Aware Seamless Hand-off in multi-channel WLANS

- Identify max RSSI variations in environment and map each AP’s RSSI threshold according to its load levels
- Before removing LVAP from Connected AP, it sends IEEE CSA Channel Switch Announcement message to target AP

<table>
<thead>
<tr>
<th>ElementID</th>
<th>Length</th>
<th>Channel Switch Mode = 1</th>
<th>New Channel Number = Target AP Channel</th>
<th>Channel Switch Count =3</th>
</tr>
</thead>
</table>

Connected AP CH=3

Connected Client

Target AP CH=9

- TX
- Beacon with CSA count =3, CH=9
- Beacon with CSA count =2, CH=9
- Beacon with CSA count =1, CH=9
- Switched to CH = 9
- TX
Sequence diagram of seamless load-aware hand-offs

Client

Client is Associated with AP1 initially

AP1 CH=3

ADD-SUBSCRIPTION

PUBLISH_AGENT

LOAD

Monitor Transmission

Transmission

Client Transmits for CH = 9

AP2 CH=9

ADD-SUBSCRIPTION

PUBLISH_AGENT

LOAD

Match against subscription
If clients experiencing RSSI > STHR of subscription

Hand-off event

Controller

Controller Updates Each AP Hand-off Subscriptions

ADD-SUBSCRIPTION

Controller Updates Each AP Hand-off Subscriptions

If load difference > load threshold And Client is getting better RSSI

Hand-off event

Hand-off to AP2

Controller

Load Aware Algorithm

ADD-LVAP

REM-LVAP

Beacon with CSA indicating target AP channel
Testbed

- A server runs TCP `iperf` server and floodlight controller based SDWLAN
- AP runs click based Agents
- Clients are configured with static IP addresses
- On Client, `iperf TCP` client is running
# Testbed Setup Details

<table>
<thead>
<tr>
<th>Testbed Components</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of APs</td>
<td>2 (Alix3d3 boards)</td>
</tr>
<tr>
<td>AP Wi-fi Cards</td>
<td>2 (Ath9k 802.11bgn)</td>
</tr>
<tr>
<td>APs’ operating mode</td>
<td>802.11g</td>
</tr>
<tr>
<td>Operating System on AP</td>
<td>Openwrt</td>
</tr>
<tr>
<td>Other tools on AP</td>
<td>Click2.0.1 and Open vSwitch1.9.0</td>
</tr>
<tr>
<td>Number of Clients</td>
<td>3</td>
</tr>
<tr>
<td>Linux Kernel version of Clients</td>
<td>3.5</td>
</tr>
<tr>
<td>Controller Software</td>
<td>Floodlight</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configurable Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial STHR</td>
<td>185</td>
</tr>
<tr>
<td>$H_{LOAD}$</td>
<td>30%</td>
</tr>
<tr>
<td>$H_{R S S I 1}$, $H_{R S S I 2}$</td>
<td>10, 15</td>
</tr>
<tr>
<td>Channels tested</td>
<td>3, 9 of 2.4GHz</td>
</tr>
<tr>
<td>$\alpha$ (Weightage to current load)</td>
<td>0.9</td>
</tr>
<tr>
<td>$T$ (Periodicity of Load Reports)</td>
<td>1 sec</td>
</tr>
</tbody>
</table>
Experiment 1

Aim: Load balance even for static clients with hand-offs

At $t=0$
- SC1 with AP1 starts TCP iperf of duration = 240 sec
- SC2 with AP2 starts TCP iperf of duration = 120 sec
  - Load of AP1 and AP2 are equal
  - SC1 and SC2 are utilizing entire Bandwidth (9 Mbps each)

At $t=60$
- Test Client (TC) with AP1 starts TCP iperf of duration = 180 sec
  - Now with AP1 network bandwidth sharing equally between SC1 and TC (Each is getting 4.5 Mbps)
  - Still load of AP1 and AP2 are equal

At $t=120$
- SC2 with AP2 completed TCP iperf session.
  - Now Load of AP1 and AP2 are become High and Low respectively
  - By detecting this load imbalance the load aware hand-off algorithm handed-off TC to AP2

From $t=120$ to $t=240$
- SC1 with AP1 is utilizing entire available network bandwidth of 9Mbps
- TC with AP2 also is utilizing entire available network bandwidth of 9Mbps
Experiment 2

- **Aim:** Seamless Hand-off even in multi channel operating WLAN
- **From 0 to 60 Sec** Client is moving from AP1 to AP2
  - At \( t = 40 \) Sec Approximately Test Client Handed-off
    - No Drop in throughput
- **From 60 to 120 Sec** Client is moving from AP2 to AP1
  - At \( t = 90 \) Sec Approximately Test Client Handed-off
    - No Drop in throughput

![TCP Network throughput](image)
Experiment 3

- **Aim:** According to load changes do Hand-off
  - At $t=0$ SC, TC are connected with AP1 started iperf TCP for 180 sec
    - AP1 is Highly loaded compared to AP2
    - AP2 coverage is set to High by load aware hand-off algorithm
  - At $t>0$ TC started moving towards AP2
    - At $t=20$ TC Handed-off by load aware hand-off algorithm
    - Early Hand-off results improved throughput to SC(4.5 Mbps) and TC(3.8 Mbps) between 20 sec to 50 sec interval
    - Load-aware algorithm adjusts again STHR value to be matched against AP1, AP2 according to their load
  - After $t=90$ sec TC moving backwards to AP1
    - As AP1, AP2 are now equally loaded
    - At $t=133$ (approximately) TC again handed-off to AP1
    - Now both are equally loaded so hand-off occurs based on better RSSI

Due to Handing-off TC to less loaded AP2 by load-aware algorithm, SC gets average throughput increase of 4.5 Mbps compared to Fixed RSSI algorithm during this 30 sec interval.
Conclusions and Future Work

- From Experiment 1: we can observe how static clients also can utilized available network bandwidth and get benefited from the load aware hand-off algorithm.

- In Experiment 2 we showed how SD WLANs can deployed in multi-channel environment for controlling interference and still achieving seamless hand-offs.

- In Experiment 3: we showed that, during mobility of clients, how to ensure good RSSI to clients in addition to load aware decisions without any ping-pong effects.

- We tested these results in a small scale testbed, as part of future work we plan to test it on an enterprise WLAN testbed.

- We are implementing SDN LTE network framework in NS-3 and testing network aware load balance application.
References


- “Click Modular Router,” www.read.cs.ucla.edu/click/click.


Thank You

- Feedback and Queries
- Contact details:
  - R Anil Kumar: cs12p1001@iith.ac.in
  - B Hardik: cs10b007@iith.ac.in
  - B Pradeep Kumar: cs10b006@iith.ac.in
  - Dr T Bheemarjuna Reddy: tbr@iith.ac.in