NeuRA: Using Neural Networks to Improve WiFi Rate Adaptation

Shervin Khastoo, Tim Brecht and Ali Abedi
Background

Two critical decisions before transmitting each frame

1) Which physical (PHY) rate to use

2) How many subframes (MPDUs) to aggregate in a frame (A-MPDU length)

Both can have a big impact on throughput
Main Contributions

NeuRA: uses a neural network to improve rate adaptation and throughput

Offline Statistically Optimal: rate adaptation and frame aggregation algorithm
  Upper bound on throughput
  Can finally better determine how well algorithms are performing
Rate Adaptation
Rate Adaptation

Time

PHY rate
52 Mbps

52 Mbps
Rate Adaptation

PHY rate

52 Mbps
52 Mbps

Time
Rate Adaptation

PHY rate
- 52 Mbps
- 52 Mbps
- 81 Mbps

Time
Rate Adaptation

PHY rate
- 52 Mbps
- 52 Mbps
- 81 Mbps
- 72.2 Mbps

Time
Rate Adaptation

PHY rate

52 Mbps
52 Mbps
81 Mbps
72.2 Mbps
65 Mbps

Time
Rate Adaptation

PHY rate

52 Mbps
52 Mbps
81 Mbps
72.2 Mbps
65 Mbps
65 Mbps

Time
Rate Adaptation

Challenge: Channel is constantly changing!
Rate Adaptation

PHY rate
- 52 Mbps
- 52 Mbps
- 81 Mbps
- 72.2 Mbps
- 65 Mbps
- 65 Mbps

Challenge:
Channel is constantly changing!
Rate Adaptation

Challenge:
Channel is constantly changing!
Rate Adaptation

Challenge:
Channel is constantly changing!
Rate Adaptation

Challenge:
Channel is constantly changing!

Practical algorithms sample (i.e., test/probe potential rates)
Rate Adaptation

PHY rate
52 Mbps
52 Mbps
81 Mbps
72.2 Mbps
65 Mbps
65 Mbps
65 Mbps

NeuRA:
- Reduce sampling overhead
- Sample smaller subset of rates
- Increase throughput

Neural network to
- Find good set of rates to sample
- Predict tput of other rates

Time
Relationships Exist Between Rates

[Abedi and Brecht, MSWiM, 2016]
Recursive Feature Eliminate (RFE) optimizes Estimation Power Sampling Time

NeuRA Overview

Training Data → Neural Network → Model \_i ∪ Sample Set \_i → Used in NeuRA

Recursive Feature Eliminate (RFE) optimizes Estimation Power Sampling Time
Trace Collection

- Modify WiFi device driver (ath9k)
- Round robin all rates
- Rates see similar channel conditions in round
Training Data

- For 1-second time intervals, throughput of each rate is calculated
- Normalize to maximum: ([0, 1] range) to prepare for neural network training

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>$TPut_1$</th>
<th>$TPut_2$</th>
<th>...</th>
<th>$TPut_{64}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.015</td>
<td>0.039</td>
<td>...</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.016</td>
<td>0.035</td>
<td>...</td>
<td>0.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2399</td>
<td>0.009</td>
<td>0.027</td>
<td>...</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Input: Fixed set of rates and tputs, Output: expected tput of all rates
NeuRA’s Resulting Neural Network Model

Weights on edges determined during training
NeuRA

Sample Set and Throughputs

R_5 \ R_{17} \ R_{23} \ \cdots \ R_{61}

(subset of rates)

Neural Network Model

Estimate Tput for all rates

R_1 \ R_2 \ R_3 \ \cdots \ R_{64}

R_{38}

Rate with best expected throughput
Evaluation Methodology

- Two separate models: 2.4 GHz and 5 GHz
- Two separate sets of traces for each: training and testing (evaluation)

<table>
<thead>
<tr>
<th>Config</th>
<th>Spectrum</th>
<th># Streams</th>
<th>Channel Width</th>
<th># Rates</th>
<th>Channel Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.4 GHz</td>
<td>2</td>
<td>20 MHz</td>
<td>32</td>
<td>Congested</td>
</tr>
<tr>
<td>B</td>
<td>5 GHz</td>
<td>2</td>
<td>40 MHz</td>
<td>64</td>
<td>Unoccupied</td>
</tr>
</tbody>
</table>
Scenarios for Trace Collection

Office Environment
Graduate student offices / lab
Scenarios for Trace Collection

Office Environment
Graduate student offices / lab

Access Point
PC with ath9k WiFi (802.11n)

TP-Link WDN4800
Scenarios for Trace Collection: Training Data

Stationary:
- Close to AP ~1 m
- Far from AP ~10 m

- Samsung Galaxy Note 5
- Laptop with TL-WDN4200 USB device

Hallway
Scenarios for Trace Collection: Training Data

Mobile: Walking
Scenarios for Trace Collection: Training Data

Mobile: Toy Train
Fast and slow
Relative Rate Adaptation Error

- Rate adaptation using model (avg. error on testing dataset)
• Rate adaptation using model (avg. error on testing dataset)
Relative Rate Adaptation Error

- Rate adaptation using model (avg. error on testing dataset)
Evaluation: Algorithms

Rate Adaptation Algorithms
- Minstrel HT
- NeuRA
- Intel iw1-mvm-rs
- Minstrel HT w/o LGI Sampling

Frame Aggregation Algorithms
- Minstrel HT + PNOFA
- Minstrel HT + OSOFA

Both
- STRALE
- Offline Statistically Optimal
Evaluation: Algorithms

Rate Adaptation Algorithms

- Minstrel HT
- NeuRA
- Intel iwl-mvm-rs
- Minstrel HT w/o LGI Sampling
Evaluation: Algorithms

Rate Adaptation Algorithms

- Minstrel HT
- NeuRA
- Intel iwl-mvm-rs
- Minstrel HT w/o LGI Sampling

- Most widely used algorithm
- 100’s of millions of devices
- In Linux
- Use as a basis for comparison
Evaluation: Algorithms

Rate Adaptation Algorithms

• Minstrel HT
• NeuRA
• Intel iwl-mvm-rs
• Minstrel HT w/o LGI Sampling

• From this work
Evaluation: Algorithms

Rate Adaptation Algorithms

- Minstrel HT
- NeuRA
- Intel iwl-mvm-rs
- Minstrel HT w/o LGI Sampling

- Another practical widely used alg
- Used in recent Intel chipsets
- Described in and code ported from
Evaluation: Algorithms

Rate Adaptation Algorithms

- Minstrel HT
- NeuRA
- Intel iwl-mvm-rs
- Minstrel HT w/o LGI Sampling

- From “relationships” paper [Abedi and Brecht, MSWiM, 2016]
- Proof of concept for relationships
- Samples SGI rates, estimates LGI
Evaluation: Algorithms

Rate Adaptation Algorithms

- Minstrel HT
- NeuRA
- Intel iwl-mvm-rs
- Minstrel HT w/o LGI Sampling

Frame Aggregation: all maximize number of frames
Except: NeuRA in 5 GHz (PNOFA)
Evaluation: Algorithms

- **Practical Near Optimal Frame Aggregation**
- **Offline Statistically Optimal Frame Aggregation**

PNOFA paper

[Abedi et al, MSWiM, 2020]
Evaluation: Algorithms

• Adjusts Frame Length and Rate
  [Byeon et al. INFOCOM 2017]

Both
• STRALE
• Offline Statistically Optimal
Evaluation: Algorithms

Both

- STRALE
- Offline Statistically Optimal
Offline Statistically Optimal: FA and RA Algorithm

**Key contribution**

- Statistically optimal frame length and rate
- Upper bound on throughput of practical RA and FA algorithms

Previously weak understanding of how well algorithms were doing

- Only relative to each other
- No idea of how much room there is for improvement
- When do we stop creating new algorithms?
Offline Statistically Optimal: FA and RA Algorithm

1. Subframe error rates from oracle
2. Calculate best aggregation length for each rate
3. Return the rate and length with highest throughput
Trace-Based Evaluation

- **T-SIMn**: trace-driven simulator [Abedi et al. MSWiM, 2016]

- Trace-based: all algorithms see the same channel conditions. Differences are due to algorithms, not changes in the channel.

- Can implement Offline Statistically Optimal (look ahead in trace)
Different Traces and Scenarios for Testing

- All new traces
- Some similar setting as training
- Previously unseen scenarios
  - 2 new devices
  - New mobility patterns (extreme movement)
- 7 scenarios for each model
- 5 - 20 minutes each
- Stationary and mobile

Traces from WiFi experiments collected using real-world conditions
Trace-Based Evaluation (Model A, 2.4 GHz)

Tput relative to Minstrel HT

-40% -30% -20% -10% 0% +10% +20% +30% +40% +50% +60%

Scenario

Similar

A1

Intel iwl-mvm-rs

Minstrel HT w/o LGI

STRALE

Similar

A2

NeuRA

Similar

A3

Minstrel HT + PNOFA

Minstrel HT + OSOFA

Similar

A4

Online Stat.

Unseen

A5

Optimal

Unseen

A6

Unseen

A7

48
Trace-Based Evaluation (Model A, 2.4 GHz)

Scenario: NeuRA, Intel iwl-mvm-rs, Minstrel HT w/o LGI, Minstrel HT + PNOFA, Minstrel HT + OSOFA, Offline Stat. Optimal

Tput relative to Minstrel HT

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Similar</th>
<th>Unseen</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Trace-Based Evaluation (Model A, 2.4 GHz)

Scenario

Similar A1
Similar A2
Similar A3
Similar A4
Unseen A5
Unseen A6
Unseen A7

Tput relative to Minstrel HT

NeuRA
Intel iwl-mvm-rs
Minstrel HT w/o LGI
STRALE
Minstrel HT + PNOFA
Minstrel HT + OSOFA
Offline Stat. Optimal
Trace-Based Evaluation (Model A, 2.4 GHz)

![Graph showing Tput relative to Minstrel HT for different scenarios and scenarios.]

- **Similar Scenarios:**
  - A1: NeuRA
  - A2: Intel iwlmvm-rs
  - A3: STRALE

- **Unseen Scenarios:**
  - A4: NeuRA
  - A5: Minstrel HT w/o LGI
  - A6: Minstrel HT + PNOFA
  - A7: Minstrel HT + OSOFA

- **Performance Metrics:**
  - Tput relative to Minstrel HT
Trace-Based Evaluation (Model A, 2.4 GHz)

- Similar
- Unseen

A1 A2 A3 A4 A5 A6 A7

Tput relative to Minstrel HT

- +60%
- +50%
- +40%
- +30%
- +20%
- +10%
- 0%
- -10%
- -20%
- -30%
- -40%

Scenario

NeuRA
Intel iwl-mvm-rs
Minstrel HT w/o LGI
STRALE
Minstrel HT + PNOFA
Minstrel HT + OSOFA
Offline Stat. Optimal
Trace-Based Evaluation (Model B, 5 GHz)

Tput relative to Minstrel HT

-10% 0% +10% +20% +30% +40%

Similar Similar Similar Unseen Unseen Unseen Unseen

B1 B2 B3 B4 B5 B6 B7

Scenario

NeuRA Minstrel HT w/o LGI Minstrel HT + PNOFA Offline Stat. Optimal
Intel iwl-mvm-rs STRALE Minstrel HT + OSOFA
Trace-Based Evaluation (Model B, 5 GHz)

-10%  0%  +10%  +20%  +30%  +40%

<table>
<thead>
<tr>
<th>Scenario</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar</td>
<td>NeuRA</td>
<td>Intel iwl-mvm-rs</td>
<td>Minstrel HT w/o LGI</td>
<td>Minstrel HT + PNOFA</td>
<td>Minstrel HT + OSOFA</td>
<td>Offline Stat. Optimal</td>
<td></td>
</tr>
<tr>
<td>Unseen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary of Trace-Driven Evaluation

• NeuRA
  • Up to 24% higher tput than Minstrel HT (16% on average)
  • Up to 32% higher tput than Intel iwl-mvm-rs (13% on average)
  • Reduces gap between Minstrel HT and upper bound by half
  • Remaining gap not overly large
Real-World Prototype (in Linux)

- **CPU: 20% of a 800 MHz core**
Conclusions

NeuRA

• Use predictions from neural network model, reduce sampling overhead
• Generalized model improves throughput on unseen scenarios
• Low processing overhead to improve throughput in real world
• Potentially greater impact with more rates (802.11ax: up to 768!)

Offline Statistically Optimal Algorithm

• Obtain upper bound on throughput (NeuRA is not that far from opt)

Simulator, Traces, Algorithms to be made available
https://cs.uwaterloo.ca/~brecht/neura