Rate Adaption Algorithms in 802.11 Networks

Dr. Ramana

I.I.T Rajasthan
Outline of the Lectures

1. Rate Adaption Algorithms
2. Auto Rate Fallback
3. Pitfalls and Design Issues
4. CARA
5. RRAA
6. Soft Rate Algorithm
Why to have Rate Adaptation

- Wireless medium is highly volatile in nature
  - fading, attenuation, interference from other radiation sources
  - mobility of nodes, moving objects
- To achieve a high performance under varying conditions, nodes need to adapt their transmission rate dynamically
- Several research works exist to improve the performance in terms of Throughput and Power Consumption
  - AAF, AARF, RBAR, CARA, RRAA, SoftPhy
## Rate selection (in 802.11a)

<table>
<thead>
<tr>
<th>Tx rate (Mbps)</th>
<th>Modulation</th>
<th>Code Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>BPSK</td>
<td>1/2</td>
</tr>
<tr>
<td>9</td>
<td>BPSK</td>
<td>3/4</td>
</tr>
<tr>
<td>12</td>
<td>QPSK</td>
<td>1/2</td>
</tr>
<tr>
<td>18</td>
<td>QPSK</td>
<td>3/4</td>
</tr>
<tr>
<td>24</td>
<td>16-QAM</td>
<td>1/2</td>
</tr>
<tr>
<td>36</td>
<td>16-QAM</td>
<td>3/4</td>
</tr>
<tr>
<td>48</td>
<td>64-QAM</td>
<td>2/3</td>
</tr>
<tr>
<td>54</td>
<td>64-QAM</td>
<td>3/4</td>
</tr>
</tbody>
</table>
How to know the channel conditions

- Based on packet loss
  - Open loop approach
  - No. of successful transmissions, loss history, statistics
  - ex. ARF, AARF, RRAA

- Based on SNR or RSSI (Received Signal Strength Indicator)
  - Closed or open loop approach
  - Measured signal to noise ratio at Phy Layer
  - ex. RBAR
ARF Algorithm

- Auto Rate Fallback (ARF) is the first commercial implementation that exploits multi-rate capability.
- Basic principle - Sender uses the status of previously transmitted packets to select future transmission rates.
- Achieves performance gain over plain 802.11.
- Algorithm:
  - If two consecutive ACK frames are not received correctly,
    - Use the next available lower rate for the second retry and subsequent transmissions and a timer is started.
  - If past 10 transmissions are successful (received ACKs) or Timer fires,
    - Send a probe frame is sent at the next available higher rate.
    - However, if an ACK is NOT received for this frame, the rate is lowered back and the timer is restarted.
 Adaptive ARF (AARF)

- ARF tries constantly to use a higher rate to be able to channel condition changes
  1. If channel conditions change very quickly $\Rightarrow$ can not adapt rate effectively
  2. If channel conditions never or very slowly changes $\Rightarrow$ too many probe attempts
- AARF tries to address the issue 2
- It updates the threshold (number of successful transmissions to be seen to send a probe packet) dynamically
  - When a probe fails, it switches back to the previous lower rate, and also doubles the threshold (ex. 10 20 40 80..)
- Fewer rate fluctuations than ARF under stable channel conditions
Figure: Mode selection comparison between ARF and AARF.
Closed loop Approaches

- **Receiver Based Auto Rate (RBAR)**
  - Cross layer solution
  - Receiver control senders transmission rate
  - RTS and CTS are modified to contain info on size and rate.
  - Sender sends the last successful transmission rate and the packet size in RTS
  - On receiving RTS, receiver picks rate based on a prior SNR thresholds and sends the same info back to the sender in CTS
  - Not 802.11 compatible

- **Received Signal Strength Link Adaptation (RSSLA)**
  - Each node overhears all on going transmissions in the neighbourhood and maintains the RSSI values as EWMA
  - When a packet needs to be transmit, it maps weighted RSSI to a rate and transmits the packet at this rate
  - Limitations: No corelation of either RSS or SNR with packet delivery probability at a given rate
  - RSSLA - link quality in both directions will not be same
## Pitfalls and Design Issues

### Pitfalls with Current Rate Adaption Algos

1. Decrease transmission rate upon severe packet loss – needed?
2. No - as packet loss can happen not only due to channel errors but also due to collisions
   - For channel errors, transmitter should decrease the transmission rate
   - For the collision error, transmitter should not decrease the rate
   - Note: Decreasing rate upon collision losses, increases the collision probability

Experimental results with/without rate adaption in the presence of a hidden node:

<table>
<thead>
<tr>
<th></th>
<th>ARF</th>
<th>AARF</th>
<th>SampleRate</th>
<th>FixedRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goodput (Mbps)</td>
<td>0.65</td>
<td>0.56</td>
<td>0.58</td>
<td>1.46</td>
</tr>
<tr>
<td>Loss Ratio</td>
<td>61%</td>
<td>60%</td>
<td>59%</td>
<td>60%</td>
</tr>
</tbody>
</table>
1. Relying on single probe packet to increase the rate?
   - Packets could be successfully transmitted at next higher rate with some probability
   - So, if probe packet is successful that doesn’t mean that all (most) transmissions at next rate will be successful

2. An unsuccessful probe can incur severe penalty on future rate adaptation

3. Use consecutive transmission successes/losses to increase/decrease rate

4. Use PHY metrics like SNR to infer new transmission rate

5. Long-term smoothened operation produces best average performance
Frame Retry – Turned Off  
Rate Adaptation – Turned Off  
Fixed Transmission Rate – For highest throughput  

Realistic scenarios = Randomly distributed loss behaviors
Phy metrics cannot be directly used to estimate transmission rates.
Collision Aware Rate Adaption (CARA)

- Employs two methods for identifying collisions:
  1. RTS Probing
  2. Clear Channel Assessment (CCA)- At SIFS time after wireless transmission finishes, assess channel with CCA. Since ACK expected to start at SIFS, if channel assessed as busy (i.e. not an ACK) then assume it is a collision. Not very much effective in all cases.

- Focuses on when to decrease the transmission rate.
- Same threshold of 10 (to increase the rate to next available transmission rate) as in AAF maintained.
- Assumes all RTS transmission failures are due to collisions.
- Transmission failure after RTS/CTS must be due to channel errors.
Algorithm

1. Data frame transmitted without RTS/CTS
2. If the transmission fails, RTS/CTS exchange is activated for the next retransmission
3. If this retransmission fails, then the rate is lowered
4. If retransmission is successful, stay at same rate and send next frame without RTS/CTS
5. (Optionally) Collision detection via CCA (clear channel assessment)
Initial State
m=n=0

Wait for MPDU

Success

m++; reset n;
if (m==Mth) {
  if (rdt < max rdt) {
    rdt ++;
  }
  reset m;
}

TxPend &
((size(MPDU) < RTSThr)
or (n < Pth))

Success

n++; reset m;
if (n>=Nth) {
  if (rdt > min rdt) {
    rdt --;
  }
  reset n;
}

Failure

Failure

DATA Tx

RTS Tx

Dr. Ramana (I.I.T Rajasthan) Rate Adaption Algorithms in 802.11 Networks 16 / 29
<table>
<thead>
<tr>
<th>Notations</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>consecutive success count</td>
</tr>
<tr>
<td>$n$</td>
<td>consecutive failure count</td>
</tr>
<tr>
<td>$M^{th}$</td>
<td>consecutive success threshold</td>
</tr>
<tr>
<td>$N^{th}$</td>
<td>consecutive failure threshold</td>
</tr>
<tr>
<td>TxPend</td>
<td>status: a data frame is pending</td>
</tr>
<tr>
<td>$R_{dt}$</td>
<td>array of transmission rates</td>
</tr>
<tr>
<td></td>
<td>$802.11a = 6, 12, 18, 24, 36, 48, 54$ Mbps</td>
</tr>
<tr>
<td></td>
<td>$802.11b = 1, 2, 5.5, 11$ Mbps</td>
</tr>
<tr>
<td>$r_{dt}$</td>
<td>transmission rate: an element of $R_{dt}$</td>
</tr>
<tr>
<td>++</td>
<td>increase transmission rate to the next higher one</td>
</tr>
<tr>
<td></td>
<td>decrease transmission rate to the next lower one</td>
</tr>
<tr>
<td>$P_{th}$</td>
<td>probe activation threshold</td>
</tr>
<tr>
<td>RTSThr</td>
<td>RTS Threshold</td>
</tr>
</tbody>
</table>
ARF OPERATION

S1

1 2 3 4 5

S2

1 2 3 4 5

RTS PROBING OPERATION

S1

1 2 3 4 5

S2

1 2 3 4 5

11 Mbps Data Tx
5.5 Mbps Data Tx
1 Mbps RTS Tx
Collision
Robust Rate Adaption Algorithm (RRAA)

- Improve the throughput performance
- Robust against various dynamics
- Three components:
  - Loss estimation
  - Rate change
  - Adaptive RTS filter
- Instead of single probe frame, RRAA uses a loss estimation window and computes the estimated loss ratio over the window.
- Uses upper and lower loss threshold for each rate and estimated loss ratio to decide when to switch rates.
- Selective use of RTS/CTS - through adaptive RTS filtering.
- **RTSwindow (RTSwnd)**
  - Window is increased by one when last frame lost without RTS (potentially due to a collision)
  - When the last frame was lost with RTS or succeeded without RTS, RTSwnd is halved (assume no collision involved)
  - When the last frame succeeded with RTS on, window is kept unchanged.
RRAA Design - Modules

- Loss Estimation
- Rate Change
- Adaptive RTS Filter
- RTS Option
- Send
- feedback
- CSMA
- Link-layer Re-tx
- Queue

802.11 MAC
Adaptive RTS Filter - Scheme

<table>
<thead>
<tr>
<th>RTSwnd</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTScounter</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Loss</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Frame transmitted

DATA

RTS
RRAA

PMTL: Threshold on maximum tolerable loss which is a function of \( P*(R) \)

PORI: Threshold on opportunistic rate increase.

PORI of rate R is a function of \( \text{PMTL}(R+)/2 \)

ewnd : estimation window for rate R

\[
P = \frac{\#\text{Lost_frames}}{\#\text{Transmitted_frames}}
\]

```python
R = highest_rate;
counter = ewnd(R);
while true do
    rcv_tx_status(last_frame);
P = update_loss_ratio();
    if (counter == 0)
        if (P > PMTL) then R = next_lower_rate();
        elseif (P < PORI) then R = next_higher_rate();
        counter = ewnd(R);
    send(next_frame,R);
    counter--;
```

\[
P*(R) = 1 - \frac{\text{Throughput}(R_\text{mid})}{\text{Throughput}(R)} = 1 - \frac{\text{tx_time}(R_\text{mid})}{\text{tx_time}(R)}
\]

PMTL: Threshold on maximum tolerable loss which is a function of \( P*(R) \)

PORI: Threshold on opportunistic rate increase.

PORI of rate R is a function of \( \text{PMTL}(R+)/2 \)

ewnd: estimation window for rate R
PORI, PMTL, ewnd, Critical Ratio for Various Rates in 11a

<table>
<thead>
<tr>
<th>Rate (Mbps)</th>
<th>Critical Loss Ratio (%)</th>
<th>$P_{ORI}$</th>
<th>$P_{MTL}$</th>
<th>ewnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>N/A</td>
<td>50.00</td>
<td>N/A</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>31.45</td>
<td>14.34</td>
<td>39.32</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>22.94</td>
<td>18.61</td>
<td>28.68</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>29.78</td>
<td>13.25</td>
<td>37.22</td>
<td>20</td>
</tr>
<tr>
<td>24</td>
<td>21.20</td>
<td>16.81</td>
<td>26.50</td>
<td>40</td>
</tr>
<tr>
<td>36</td>
<td>26.90</td>
<td>11.50</td>
<td>33.63</td>
<td>40</td>
</tr>
<tr>
<td>48</td>
<td>18.40</td>
<td>4.70</td>
<td>23.00</td>
<td>40</td>
</tr>
<tr>
<td>54</td>
<td>7.52</td>
<td>N/A</td>
<td>9.40</td>
<td>40</td>
</tr>
</tbody>
</table>
Cross-Layer Wireless Bit Rate Adaptation - Soft Rate Algorithm

- Bit rate adaptation on basis of BER rather more common SNR
- Physical layer hints at receiver are collected by sender to adjust bit rate
- Cross layer solution
- Observations
  1. At any SNR, the BER is a monotonically increasing function of the bit rate
  2. Within the BER range that a bit rate is usable (i.e., BER below $10^{-2}$), its BER at a given SNR is at least a factor of 10 higher than that of the next-lower bit rate.
- The design of SoftRate centers on three main mechanisms.
  1. Heuristic to predict channel BER.
  2. Given the interference-free BER estimate from the receiver and optimal thresholds at each bit rate, the SoftRate sender adjusts its bit rate in the direction of the optimal rate.
Observations

For any SNR adjacent rates have an order of magnitude difference in BER
When is the current rate optimal?

![Diagram showing the optimality range for 18 Mbps, 24 Mbps, and 12 Mbps with BER and Throughput axes.]
BER from SoftPHY Hints

Soft Rate Algorithm

Soft Output Viterbi (or) BCJR decoder

decoded bits

Pr(correctly decoded)

Pr(incorrectly decoded)

log

s

1 - p

p

1 + e^s

For linear block or convolutional code

Log Likelihood Ratio

SoftPHY hint of a bit

s = log \( \frac{1-p}{p} \)

Probability of bit error

p = \( \frac{1}{1 + e^s} \)

BER = Average p over all bits in the packet
The SoftRate Protocol

Sender

- Precompute optimality ranges
  - If BER below optimality range, increase rate.
  - If above range, decrease rate.
  - Otherwise, continue at current rate.

Receiver

- Interference Detection
- Interference-free BER
- SoftPHY Hints
References


