Robust Rate Adaptation for 802.11 Wireless Networks

Starsky H.Y. Wong, *et al.*, UCLA

Presented by Alwin Tareen

Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY 14226, USA

January 25, 2011

Key Points of the RRAA Paper

- Rate Adaptation(RA):
  - Allows for each device to adapt the runtime transmission rate based on the dynamic channel condition
- Several existing RA Algorithms are:

ARF	AARF	SampleRate

- 5 popular design guidelines used by these RA algorithms are critiqued
- Proposal of a Robust Rate Adaptation Algorithm(RRAA):
  - Uses a *short-term loss ratio* to guide rate selection
  - Applies an adaptive RTS filter to suppress collision losses
- Experimental analysis confirms RRAA superiority

The IEEE 802.11 WLAN Standard

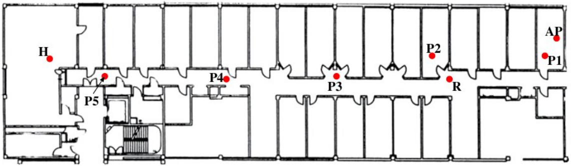
- Access Points and Clients use 802.11 a/b/g devices
- 802.11 DCF mode: a DATA-ACK exchange is performed between Access Point and Client
- Each device may adapt to the following transmission rate options:

802.11b(2.4GHz):	1, 2, 5.5, 11 Mbps
802.11a(5GHz):	6, 9, 12, 18, 24, 36, 48, 54 Mbps
802.11g(2.4GHz):	1,2,5.5,6,9,11,12,18,24,36,48,54 Mbps

- The goal of rate adaptation is to maximize the transmission throughput at the receiver
- Existing RA algorithms make rate decisions based on ACK, indicating successful DATA packet delivery or transmission failure

Experimental Setup

- Access Points:
  - AP uses the Agere 802.11a/b/g chipset, all 3 clients supported
  - It implements the 802.11 MAC in the FPGA firmware
  - H periodically broadcasts packets, acts as a hidden station



- Receiving Clients:
  - P1, P2, P3, P4, P5 and R
  - Linux 2.6 kernel, CISCO Aironet 802.11a/b/g Adapters
  - The wireless device driver is MADWiFi

Experimental Methodology

- Static scenario: All devices stationary
  - Evaluates the stability and robustness of the algorithms
  - Can the RA stabilize around an optimal rate?
  - Explores sensitivity to random frame losses
- Mobile scenario: AP stationary, Client in motion
  - Evaluates algorithm responsiveness in adapting to significant channel variations
- Hidden station scenario
  - Assesses how an algorithm performs under collision losses
- Uncontrolled field trial
  - Performed during regular office hours
  - Evaluates how the different RA algorithms perform in realistic situations

Existing RA Algorithms: ARF, AARF and SampleRate

- AutoRate Fallback(ARF):
  - Uses *probe packets* sent at a higher transmission rate
  - If the packet succeeds, the rate is increased
  - The rate is decreased on 2 consecutive transmission failures
- Adaptive AutoRate Fallback(AARF):
  - Improves the stability of ARF
  - When a probe packet fails, the probing threshold is doubled
- SampleRate:
  - Best algorithm for static settings
  - Transmits at the rate with the smallest transmission time
  - Periodically sends out probe packets to a randomly selected rate

The RA Mechanisms

- Estimation of the best transmission rate
  - What information can be used in the estimation?
  - How should the best transmission rate be estimated?

Physical-layer	direct estimation	SNR or PHY metrics	RBAR, OAR
Link-layer	indirect estimation	frame transmission	ARF, SampleRate
Hybrid	inference	both PHY and link-layer	HRC

The RA Mechanisms

- Collection of Link-layer information
- *Data-frame* approach:
  - Probing: a few data frames are periodically transmitted at a rate different from the current one
- *Signaling-frame* approach
- Estimation strategies
  - *Deterministic pattern:* Consecutive frame successes indicate good channel conditions
  - *Statistical frame metrics*
- Rate adjustment actions
  - *Sequential:* Increase/decrease the current rate by one level
  - *Best rate:* Jump multiple levels to a better rate

Guideline 1: Decrease transmission rate upon severe packet loss

- Original motivation
  - Link condition between sender and receiver deteriorates
  - Significant losses occur at the current rate
  - Sender adapts by switching to lower rate
- Hidden station scenario
  - A receiver experiences significant packet losses with hidden stations present
  - Decreasing the rate only worsens the collisions
- The RA solution should identify the cause of the packet losses and act accordingly

Guideline 1: Decrease transmission rate upon severe packet loss

- Experimental analysis
  - Setup: A sender at AP, a receiver at R and a hidden terminal at H
  - When H broadcasts its packets at 0.379Mbps, R experiences 60% losses for all algorithms
  - The heavy collision losses cause the RA algorithms to reduce their rates to 1Mbps
  - If RA is turned off and FixedRate is used at 11Mbps, the throughput improves to 1.46Mbps

	ARF	AARF	SampleRate	FixedRate
Throughput (Mbps)	0.65	0.56	0.58	1.46
Loss Ratio	61%	60%	59%	60%

Guideline 2: Use probe packets to assess possible new rates

- Intention
  - *Probe packets:* data frames sent out at a different transmission rate
  - If the probe packets are successful, the algorithm will switch to the better rate
- Downsides
  - A successful probe can be misleading, and trigger an incorrect rate increase
  - An unsuccessful probe can incur overly harsh penalties on any future rate adaptations
- A statistically small number of probe packets can dramatically influence the RA algorithms

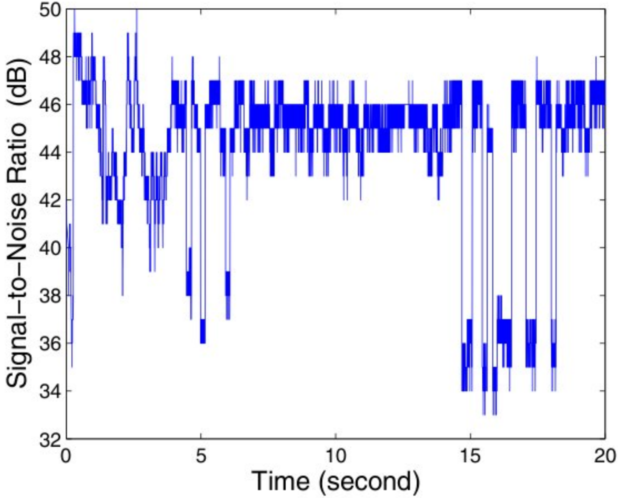
Guideline 3: Use consecutive transmission successes/losses to increase/decrease rate

- Intention
  - The rate should be changed only when the transmission successes or failures are consecutive
- Experimental analysis
  - AP has RA algorithm and frame retry switched off
  - For each run, manually fix the transmission rate that gives the highest throughput
  - The success/failure event for each packet transmission is recorded in the AP
- Transmission probabilities
  - A statistical analysis reveals that consecutive transmission successes are difficult to consistently achieve
  - Packet losses are randomly distributed, and interfere with deterministic pattern schemes

Guideline 4: Use PHY metrics like SNR to infer new transmission rate

- In theory, physical-layer metrics should lead to an accurate rate estimation
- Practical difficulties
  - Experimental studies show there is no strong correlation between SNR and delivery probability
  - SNR variations make rate estimation highly inaccurate
- Experimental analysis
  - Send back-to-back UDP packets from the AP to the client and sample the SNR value
  - The SNR value can commonly have variations of 5dB between consecutive transmissions
  - The large SNR variation can lead to multiple rate level deviations when adjusting the transmission rate

Guideline 4: Use PHY metrics like SNR to infer new transmission rate



Guideline 5: Long-term smoothened operation produces best average performance

- Underlying hypothesis
  - Long-term estimation/action will smoothen out the impact of random errors
  - Lead to best average performance
- Practical assessment
  - Smaller sampling periods perform much better than longer ones
  - Existing RA algorithms are unable to exploit short-term opportunistic gain in the wireless channel
- Long-term, infrequent rate change decisions can lead to performance penalties

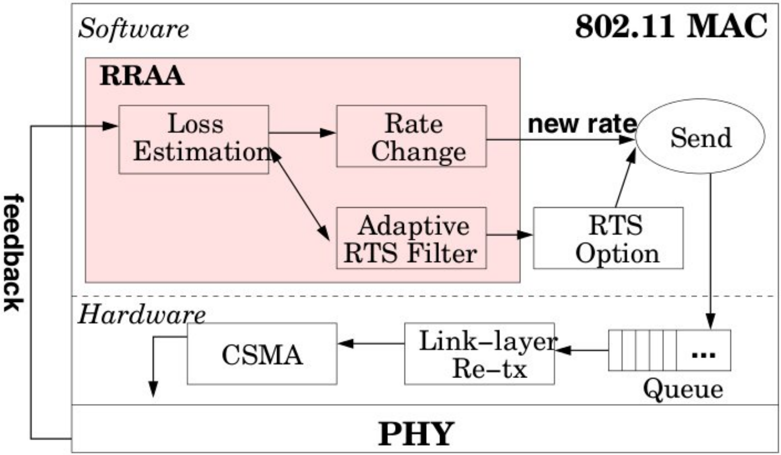
Guideline 5: Long-term smoothened operation produces best average performance

- Experimental analysis
  - Sender at P2 uses the ONOE algorithm in MADWiFi to send packets to the AP
  - The sampling period is varied and the results are tabulated
  - A small sampling period of 100ms produces the best average performance in the long term

Sampling intervals (ms)	5000	1000	500	100
UDP Throughput (Mbps)	14.9	15.3	16.5	17.1

- Mobile experiment
  - A person carries the 802.11b receiver in a route around the building
  - Compare the ARF and SampleRate algorithms
  - SampleRate has a smaller average UDP throughput, because it is penalized by delayed rate-change decisions

The Robust Rate Adaptation Algorithm (RRAA)



The Robust Rate Adaptation Algorithm (RRAA)

- Maximize the aggregate throughput in the presence of various channel dynamics.
- Design goals:
- Robust against random loss
  - Mild, random channel variations should not affect the stability of the rate and throughput
- Responsive to drastic channel changes
  - Quickly track the rate increase/decrease associated with the channel change
  - Responds properly to severe channel degradation e.g. hidden terminals, microwave ovens, etc.

The Robust Rate Adaptation Algorithm (RRAA)

- Design concepts:
- Short-term loss ratio
  - Assess the channel with a *frame loss ratio* and adapt the transmission rate accordingly
- Adaptive application of the RTS handshake
  - Selectively turns on the RTS/CTS exchange to suppress collision losses

- RRAA Modules:

Loss Estimation	Rate Change	Adaptive RTS Filter

- RRAA adjusts the transmission rate based on:
  - The frame loss ratio
  - Calculated over the previous *short-term* time window

RRAA: Loss Estimation and Rate Change

- The RRAA Algorithm
  - Starts with the highest rate
  - Selection of a new rate initializes an estimation window of *ewnd* frames
  - The loss ratio is based on the number of frames lost in *ewnd*
- Runtime loss ratio:

$$P = \frac{\#LostFrames}{\#TransmittedFrames}$$

RRAA: Loss Estimation and The RRAA-BASIC Algorithm

```
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_high_rate();
        counter = ewnd(R);
    send(next_frame,R);
    counter--;
```

RRAA: Loss Estimation and Rate Change

- A new transmission rate is selected, based on the loss ratio P
- The rate is decreased if:
  - The loss ratio P is larger than the *Maximum Tolerable Loss* threshold, P_{MTL}
- The rate is increased if:
  - The loss ratio P is smaller than the *Opportunistic Rate Increase* threshold, P_{ORI}
- If the loss ratio P lies between P_{MTL} and P_{ORI}
  - The estimation window keeps *sliding forward*

RRAA: Loss Estimation and Rate Change

- Calculation of the P_{MTL} threshold
  - R_{-} is the next lowest rate to R
  - With a loss ratio of P^*, the throughput at R is the same as the loss-free throughput at R_{-}

$$P^*(R) = 1 - \frac{Throughput(R_{-})}{Throughput(R)} = 1 - \frac{txTime(R)}{txTime(R_{-})}$$

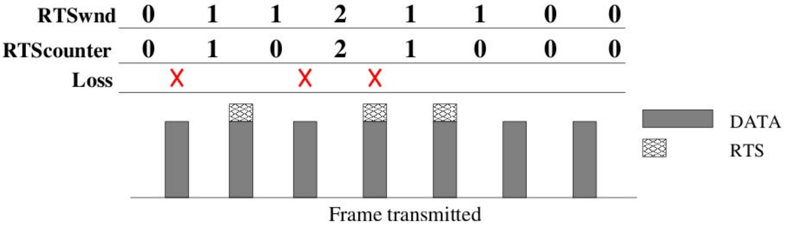
- Calculation of the P_{ORI} threshold
  - $P_{MTL}(R^+)$ is the threshold of the next higher rate
  - The loss ratio at R must be small enough for $R+$ to stabilize

$$P_{ORI} = \frac{P_{MTL}(R^+)}{\beta}$$

RRAA Implementation Parameters

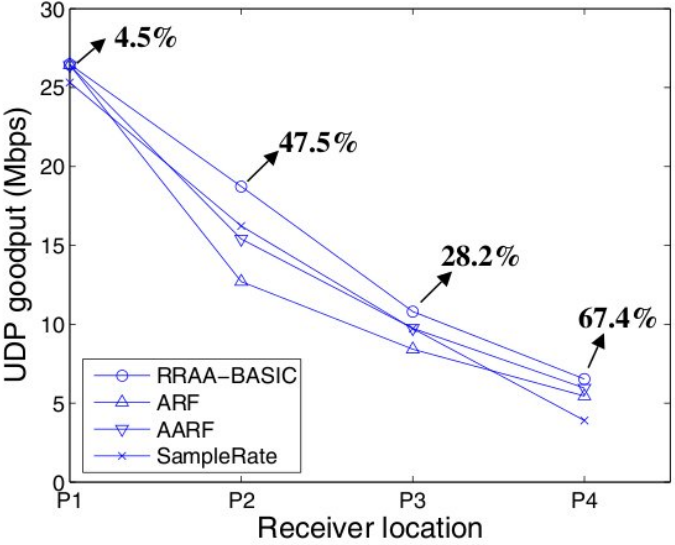
Rate (Mbps)	Critical Loss Ratio (%)	P_{ORI}	P_{MTL}	$ewnd$
6	N/A	50.00	N/A	6
9	31.45	14.34	39.32	10
12	22.94	18.61	28.68	20
18	29.78	13.25	37.22	20
24	21.20	16.81	26.50	40
36	26.90	11.50	33.63	40
48	18.40	4.70	23.00	40
54	7.52	N/A	9.40	40

RRAA: The Adaptive RTS Filter

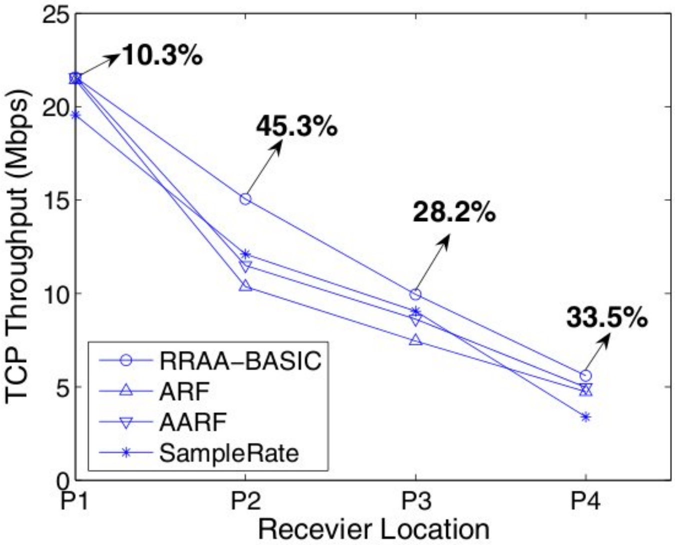


- Selectively turn on the RTS/CTS exchange to suppress collison losses
- When collision losses are severe, more frames are sent with RTS on
- During periods of mild or absent collision losses, use of the RTS/CTS exchange is minimized

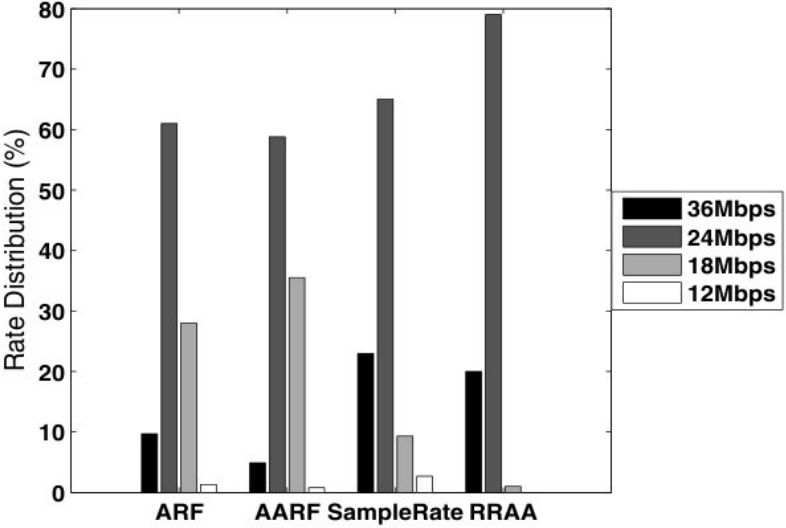
RRAA Performance Evaluation: UDP 802.11a Static



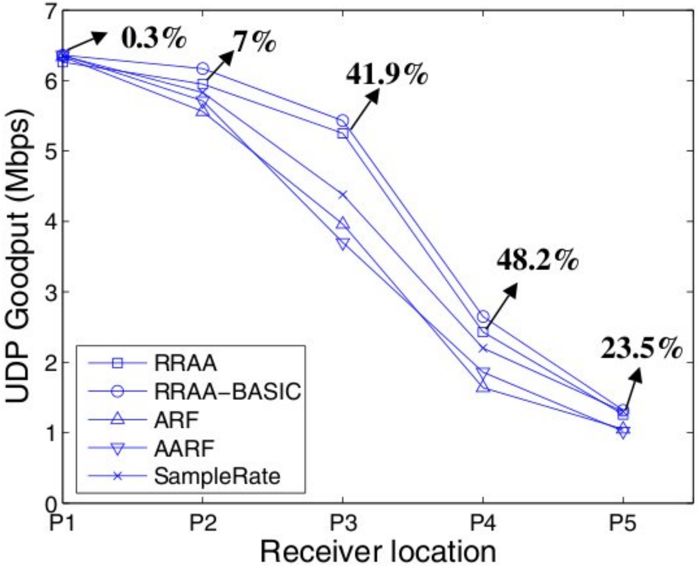
RRAA Performance Evaluation: TCP 802.11a Static



RRAA Performance Evaluation: 802.11a Rate Distribution



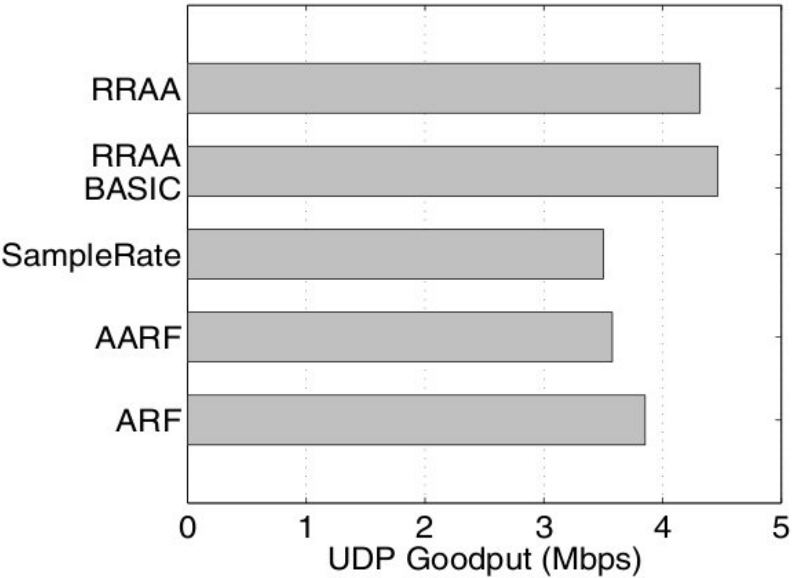
RRAA Performance Evaluation: UDP 802.11b Static



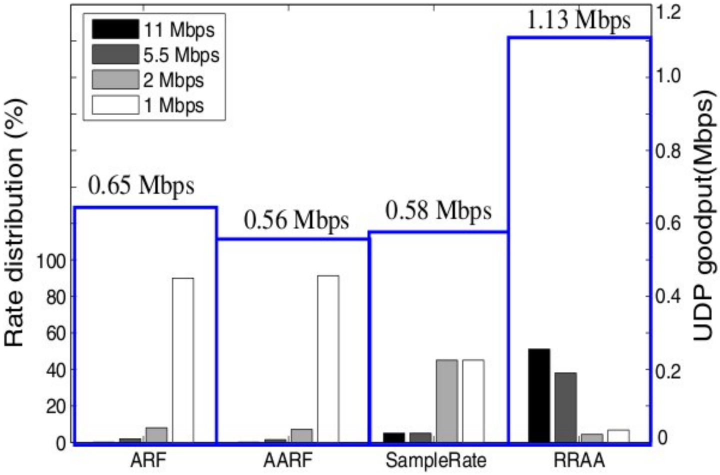
RRAA Performance Evaluation

- 802.11a
  - The RRAA-BASIC algorithm outperforms the other RA algorithms, in both UDP and TCP
  - RRAA-BASIC transmits 79% of its packets at 24Mbps
  - The others transmit only 59%∼66% of their packets at this rate
  - RRAA-BASIC only reduces its transmission rate when its loss ratio threshold has been reached
  - This proves RRAA-BASIC is more robust to channel losses
- 802.11b
  - RRAA achieves 0.3%∼48.2% throughput gain compared to the other algorithms

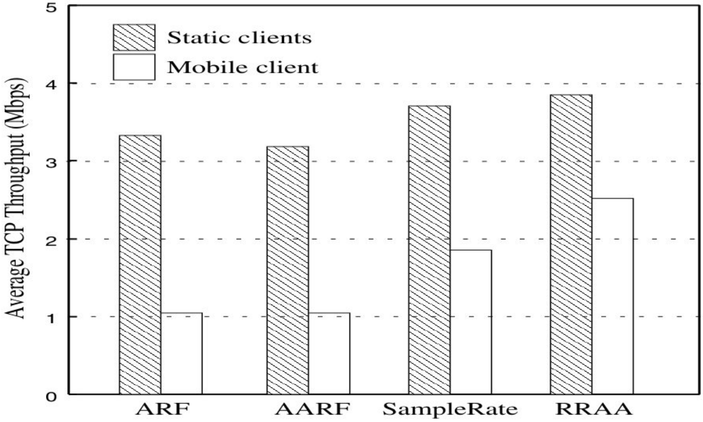
RRAA Performance Evaluation: Mobile Client



RRAA Performance Evaluation: Hidden Terminals



RRAA Performance Evaluation: Field Trials



RRAA Performance Evaluation

- Mobile Client
  - This scenario gauges the responsiveness of the RA algorithms
  - RRAA provides throughput improvements of 10%∼27.6%
- Hidden Terminals
  - Evaluates the ability of the algorithm to quickly infer collision losses and adjust the rate accordingly
  - RRAA provides throughput gains of 101% and 74% due to its adaptive RTS filter mechanism
- Field Trials
  - Conducted to understand how the RA algorithms perform under realistic scenarios
  - Both static and mobile settings were tested in an busy office setting for a duration of 6 hours
  - RRAA achieves throughput gains of 15.3% for static and 142.7% for mobile

Conclusions

- Rate adaptation offers an effective means to facilitate system throughput improvement in 802.11
- Five common design guidelines were critiqued
- A new Robust Rate Adapatation Algorithm(RRAA) was proposed

- Key insight: RA algorithms must infer different loss behaviors and adapt accordingly
- RRAA was compared with ARF, AARF and SampleRate
- Experimental analysis showed the superiority of RRAA over the other RA algorithms