Robust Rate Adaptation in 802.11 Networks

Starsky H.Y, Hao Yang, Songwu Lu and Vaduvur Bharghavan

Presented by Meganne Atkins

Rate Adaptation

- Definition - The method used to dynamically select the transmission rate based on time-varying and locate independent channel quality

Goal: Optimize the transmission goodput at the receiver

The IEEE 802.11 Standard and Rate Adaptation

- Transmission Rates for 802.11 variations:
  - 802.11b 4 rate options
  - 802.11a 8 rate options
  - 802.11g 12 rate options
- Each Transmission Rate has different modulation and coding schemes

Rate Adaptation is critical to performance, but left undefined

Importance of Rate Adaptation

Transmission Rate ↑ than Loss Ratio ↑

Throughput Decreases

Transmission Rate ↓ than Capacity Utilization ↓

Throughput Decreases

Assumption: Performance is defined as throughput

Related Work

- Rate Adaptation Algorithms Metrics:
  - Probe Packets
    - ARF
    - AARF
    - SampleRate
  - Consecutive successes/losses
    - ARF
    - AARF
    - Hybrid Algorithm
  - Physical Layer metrics
    - Hybrid Algorithm
    - RBAR
    - OAR
  - Long-term statistics
    - ONOE

Commercially Deployed: ARF, SampleRate and ONOE

Issues with Current Algorithms

- Current Metrics are limited in scope
- Simulations do not show flaws in the algorithms
- Performance loss
- 802.11 non-compliant algorithms
  - RBAR

Flawed design guidelines = Flawed algorithms

Current Design Guidelines

1. Decrease Transmission Rate upon severe packet loss
2. Use Probe Packets to assess new rate
3. Use consecutive transmission success/losses to decide rate increase/decrease
4. Use PHY metrics to infer new transmission rate
5. Long-term smothered operation produces best average performance

Experimental Methodology

- Programmable AP Platform
  - Supports 802.11 variations a/b/g
  - Per-frame Control functionality
  - Real-time tracing
  - Transmission rate control functionality
  - Low feedback delay

Experimental Methodology

- Experimental Setup
  - Static/Mobile Clients
  - 802.11 a/b
  - With/Without Hidden Stations

Guideline: Decrease Transmission Rate upon severe packet loss

Channel Conditions Worsen → Lower Transmission Rates

What if a hidden station exists?

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

Goodput decreases!

Many packet loss scenarios exists, algorithms cannot be limited to fading/path loss

Guideline: Use Probe Packets to assess new rate

- Issues
  - Successful probes can be misleading
  - Unsuccessful probes can incur severe penalties

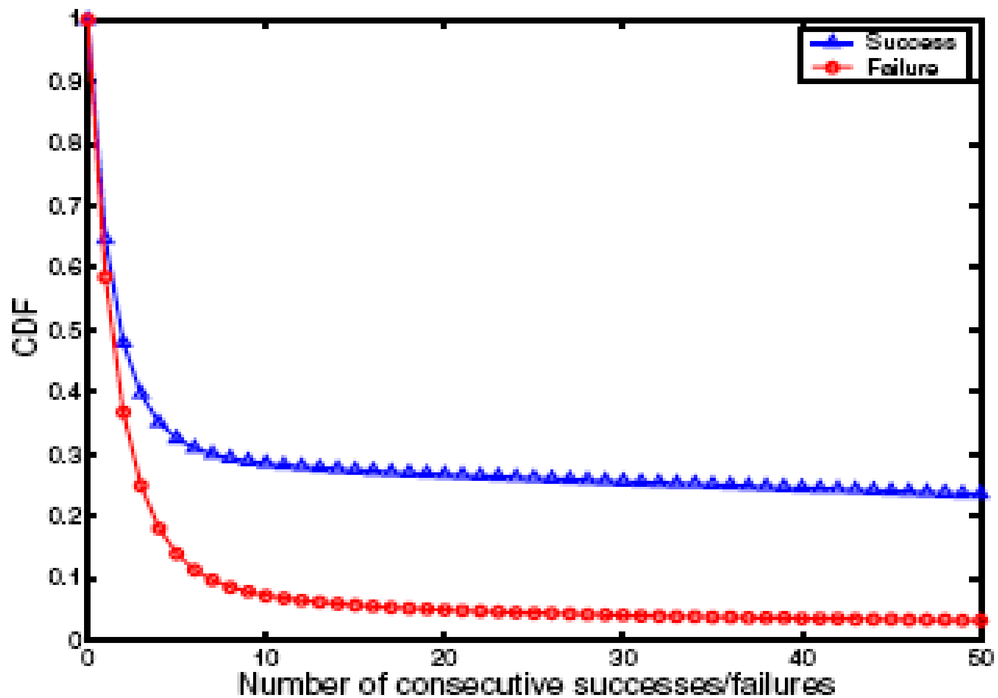
Small # probes ⇨ Inaccurate rate adaptation
Probing is too sensitive

Guideline: Use consecutive transmission success/losses to decide rate increase/decrease

- Statistically the success rate for this method is sub par
  - After 10 consecutive success (28.5%)
  - After back-to-back failures (36.8%)

Statistics are not substantial enough to base an algorithm on

Guideline: Use consecutive transmission success/losses to decide rate increase/decrease (cont'd)

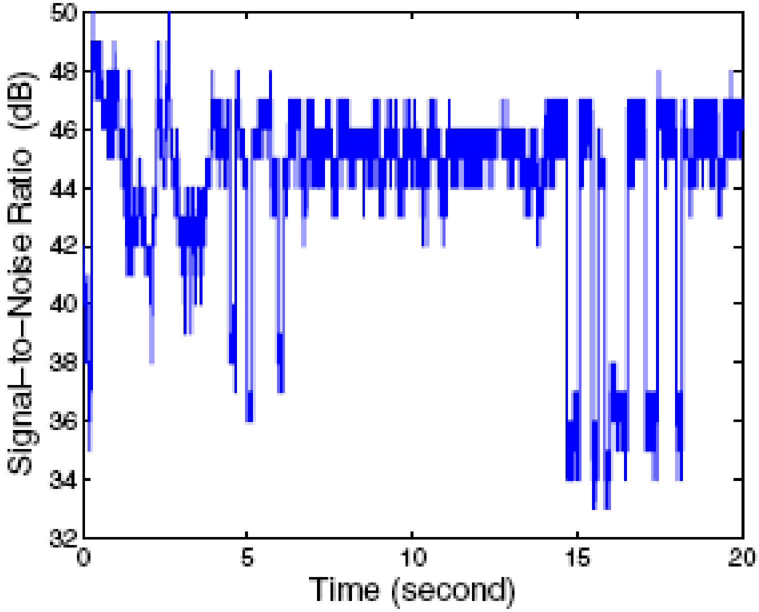


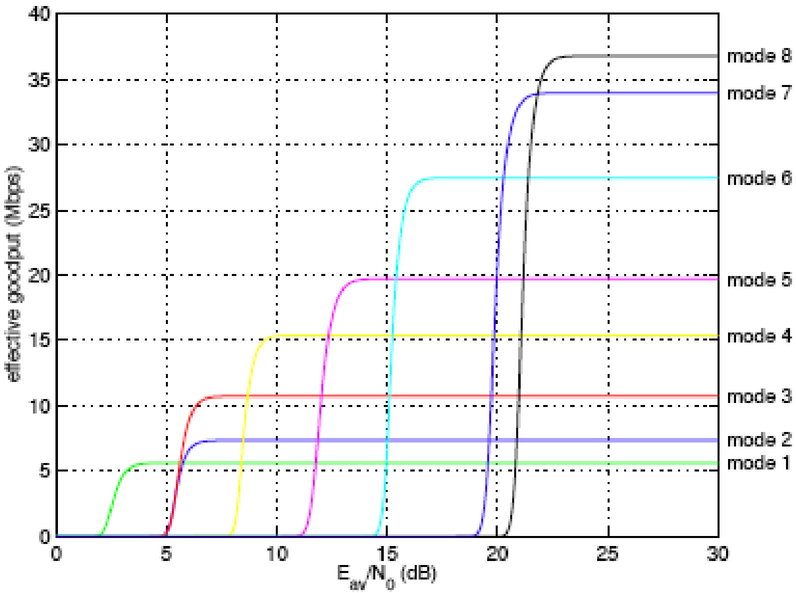
Frame Retry –Turned Off   Rate Adaptation – Turned Off

Fixed Transmission Rate – For highest throughput

Realistic scenarios =
Randomly distributed loss behaviors

Guideline: Use PHY metrics to infer new transmission rate





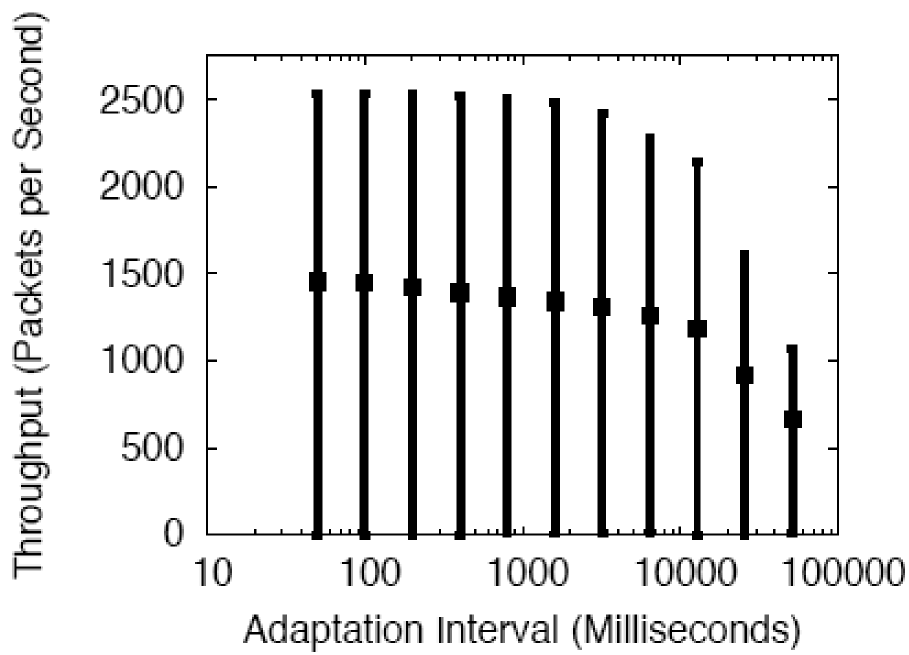
(a) MSDU size : 2000 octets

Metrics can not be directly used
to estimate transmission rates

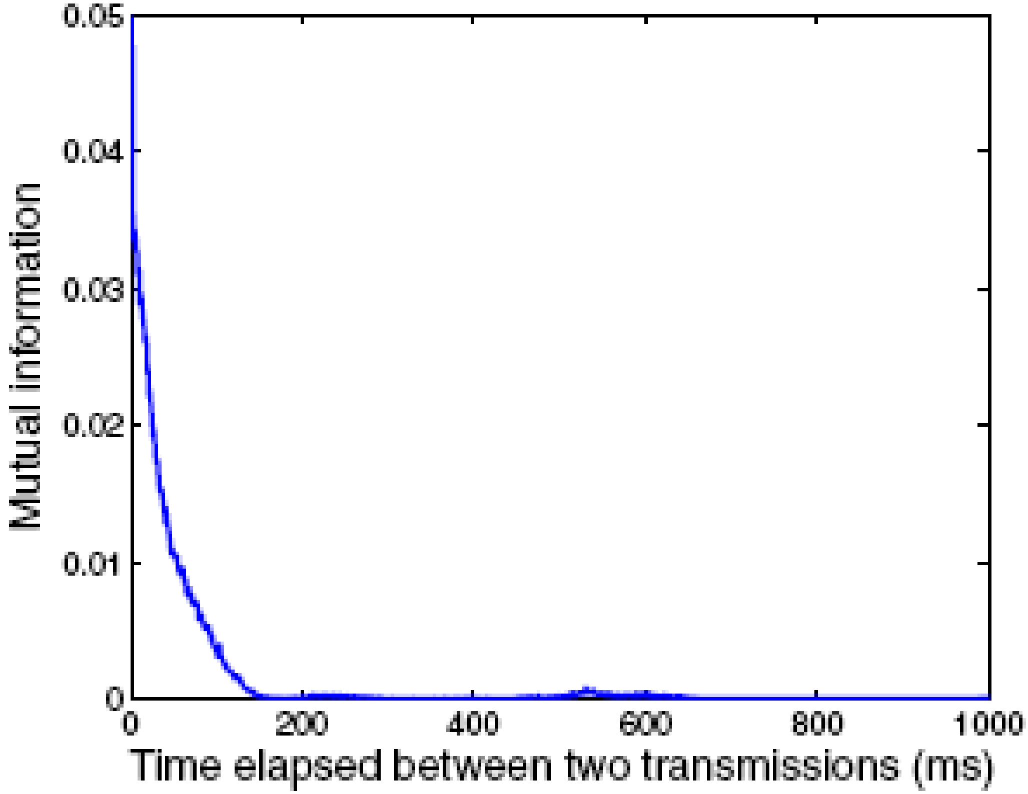
Guideline: Long-term smothered operation produces best average performance

- Issues
  - Long-term rate estimation and rate change over time =! Best average performance

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



Guideline: Long-term smothered operation produces best average performance (Cont'd)



Large Sampling periods do not lead to more accurate rate estimations

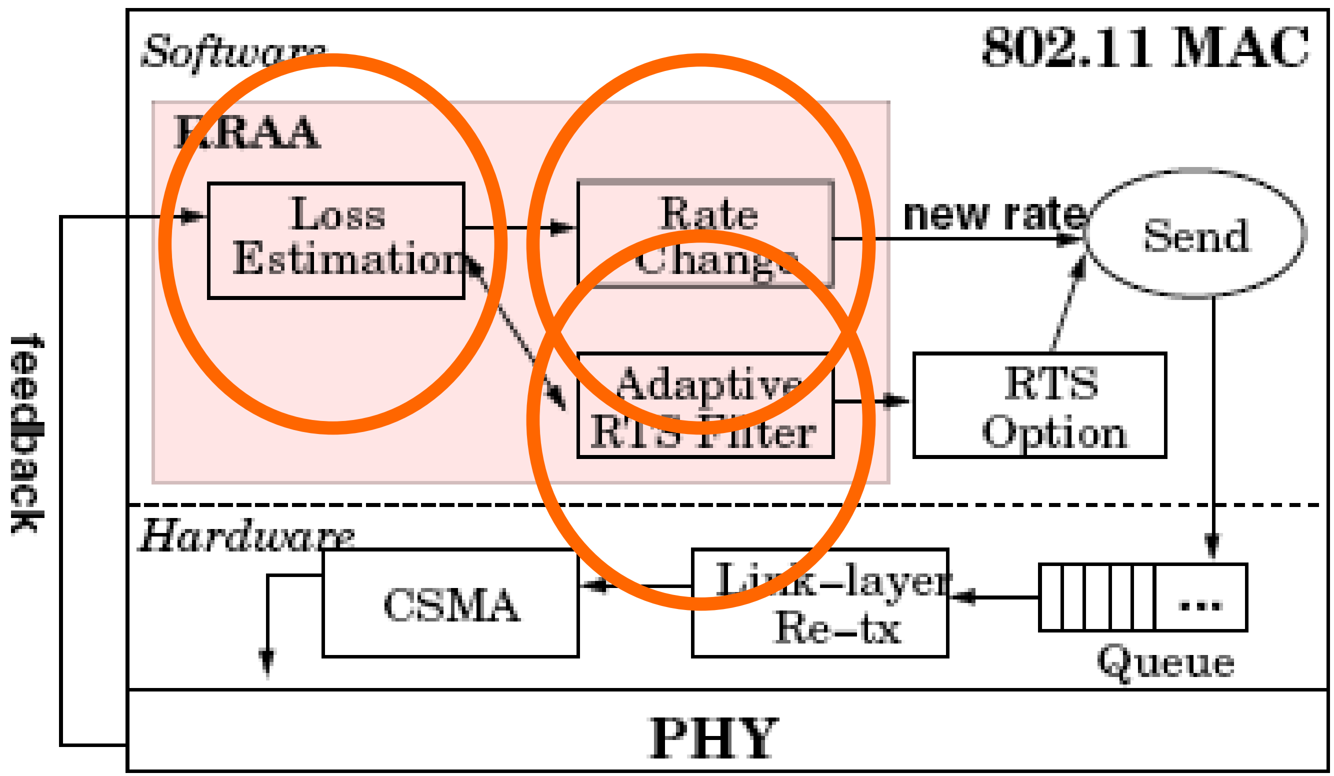
Robust Rate Adaptation Algorithm Goals

- Improve performance
- Manage varying dynamics
- Easy to implement
- Fit the IEEE 802.11 Standard

RRAA Design

- Short-term loss ratio
  – Assess
  – Adapt
- Adaptive RTS
  – Leverage
  – Filter

RRAA Design -Modules



RRAA-BASIC – Loss Estimation

```
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--;
```

$$P = \frac{\#\_lost\_frames}{\#\_transmitted\_frames}$$

RRAA-BASIC – Rate Change

```
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--;
```

$$P^*(R) = 1 - \frac{Throughput(R\_)}{Throughput(R)} = 1 - \frac{tx\_time(R)}{tx\_time(R\_)}$$

Estimation Window Size

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

```
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--;
```

Miscellaneous Issues

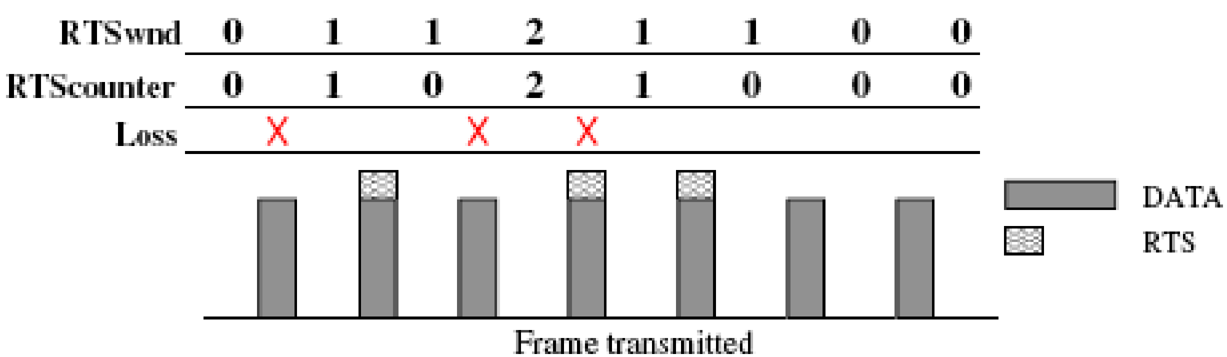
- Idle Stations
  - Refresh the window
- Multiple Active Stations
  - More stations, shorter estimation windows
- Variable Packet Size
  - Packet groupings

Adaptive RTS Filter - Design

- Suppressing hidden-station-induced loss options
  - – Turn RTS on (every frame)
    - Large Overhead
  - – RTS on frame loss / RTS off frame success
    - RTS oscillations

Adaptive RTS Filter - Scheme

```
RTSWnd = 0;
RTScounter = 0;
while ture do
    rcv_tx_status(last_frame);
    if(!RTSOn and !Success) then
        RTSWnd++;
        RTScounter = RTSWnd;
    elseif(RTSOn xor Success) then
        RTSWnd = RTSWnd/2;
        RTScounter = RTSWnd;
    if(RTScounter > 0) then
        TurnOnRTS(next_frame);
        RTScounter--;
```



Integrating RRAA-BASIC & A-RTS

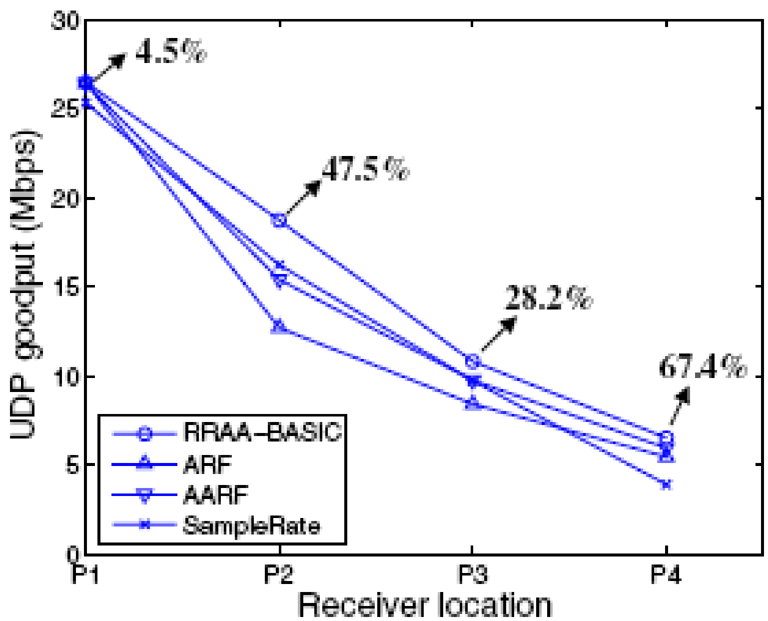
- RRAA-BASIC
  - Channel Fluctuations
- A-RTS
  - Hidden terminals

```
while true do
   rcv_tx_status(last_frame);
   A-RTS();
   if(!RTSFail) then
      RRAA_BASIC();
      if(RTSWnd > 3) then
         fix_re_tx_rate();
```

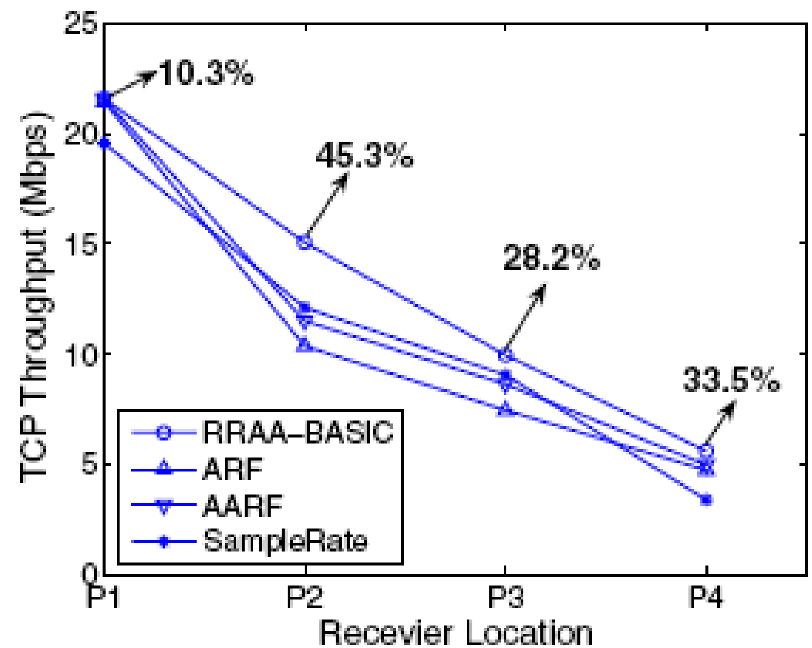
Performance Evaluation – Static Clients

- Other Algorithms
  - ARF
  - AARF
  - SampleRate
- UDP and TCP
- 802.11 a/ b Channels

Performance Evaluation – Static Clients

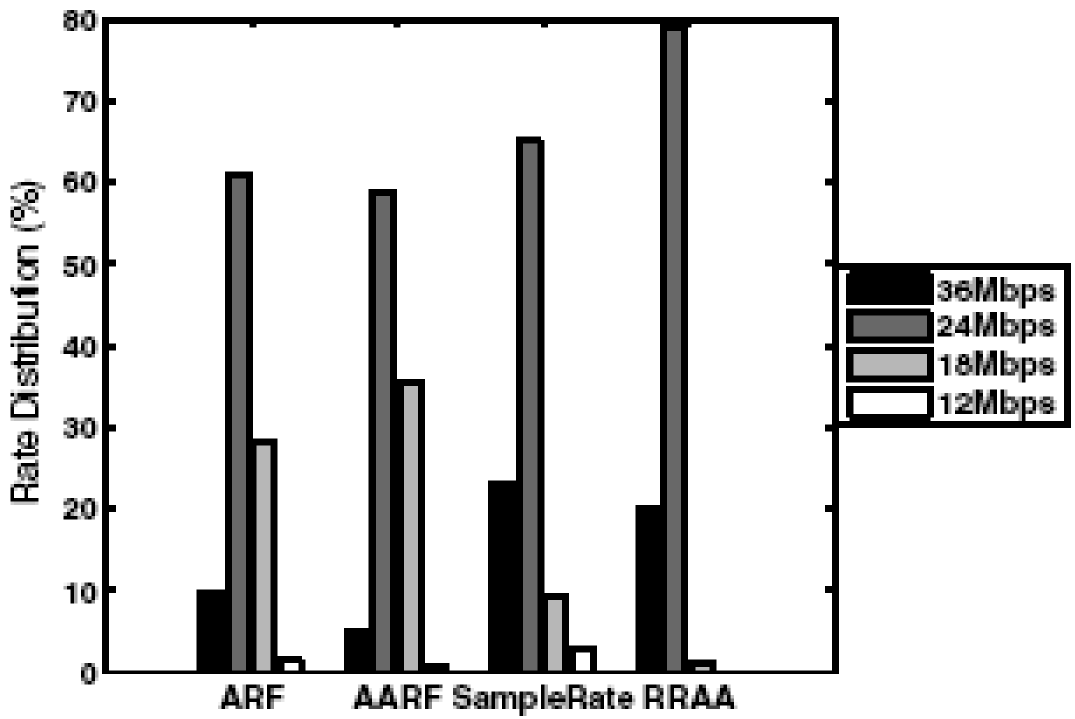


(a) UDP goodput in 802.11a.

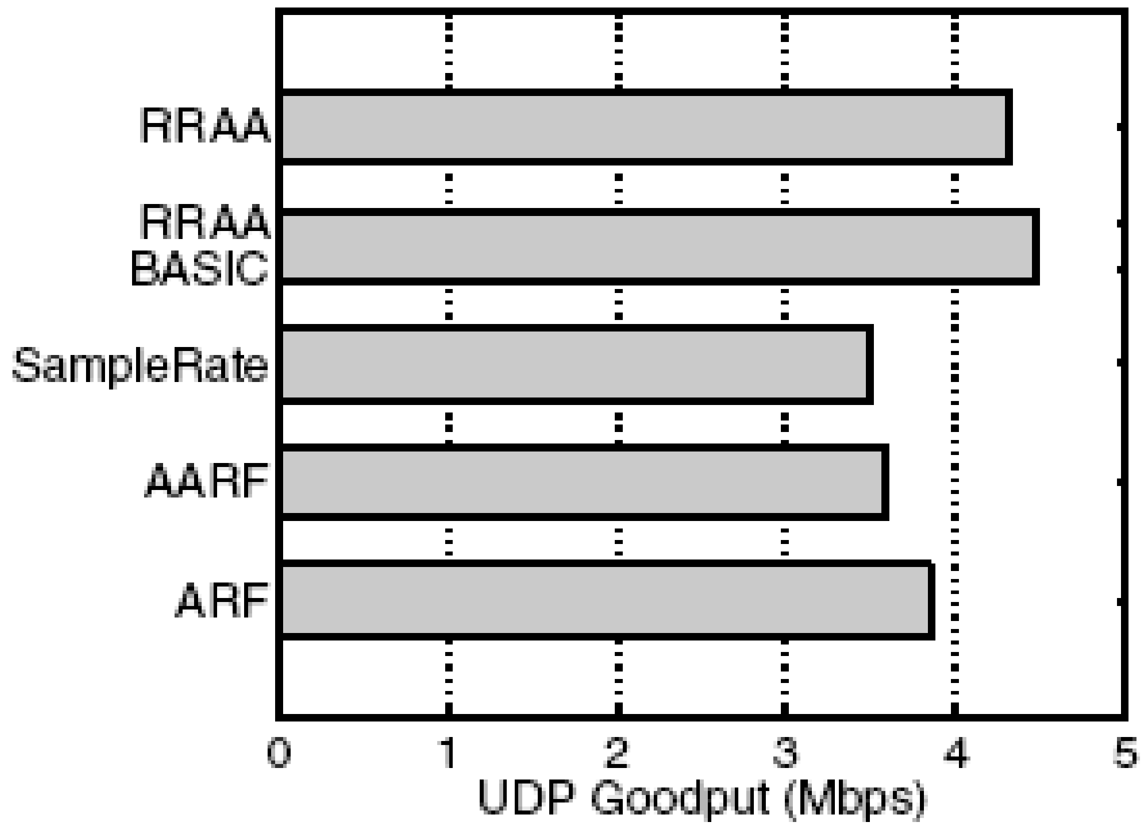


(b) TCP throughput in 802.11a.

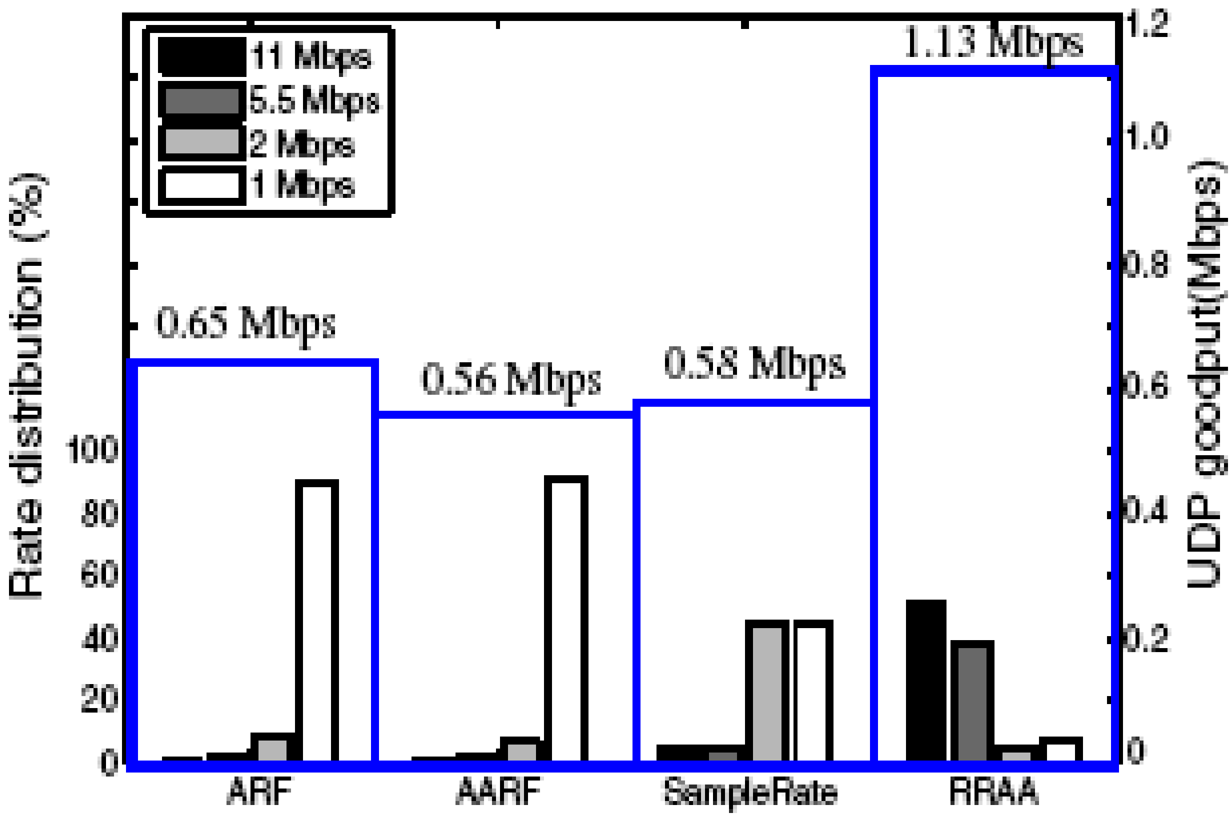
Performance Evaluation – Static Client



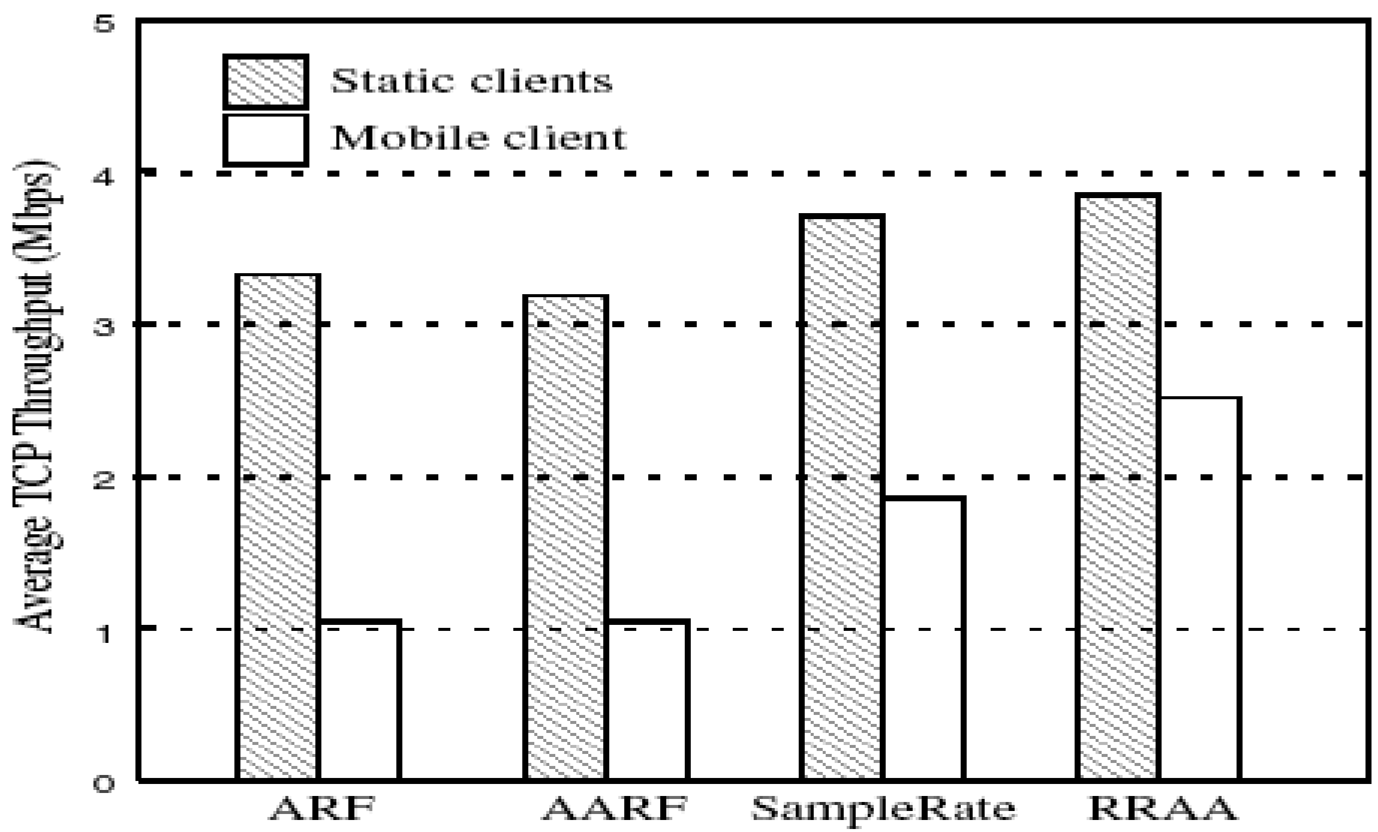
Performance Evaluation– Mobile Clients



Performance Evaluation – Hidden Stations



Performance Evaluation– Field Trials



Conclusions

- Rate Adaptation Algorithms
  - Differentiate between loss behaviors
  - Adapt to realistic scenarios
  - Handle hidden stations

PROBLEMS

- Number of Work Stations
- RTS failure/ Data transmission Failure

Work Cited

- S. H.Y, H. Yang, S. Lu and V.Bharghavan. Robust Rate Adaptation in 802.11 Networks
- Chart (Slide 15) - Figure 3.5 from J.Bicket. Bit-rate Selection in Wireless Networks. MIT Master's Thesis, 2005
- Chart (Slide 14) – MSDU from D.Qiao, S.Choi and K.Shin.Goodput Analysis and Link Adaptation for 802.11a Wireless LANs.IEEE TMC, 1(4), October 2002