CARS: Context Aware Rate Selection for Vehicular Networks

Pravin Shankar
spravin@cs.rutgers.edu

Tamer Nadeem
tamer.nadeem@siemens.com

Justinian Rosca
justinian.rosca@siemens.com

Liviu Iftode
iftode@cs.rutgers.edu

Vehicular networks today

- Ubiquity of WiFi
  - Cheaper, higher peak throughput compared to cellular
- New applications
  - Traffic Management
  - Urban Sensing (eg. Cartel)
  - In-car Entertainment
  - Social Networking (eg. RoadSpeak, MicroBlog)

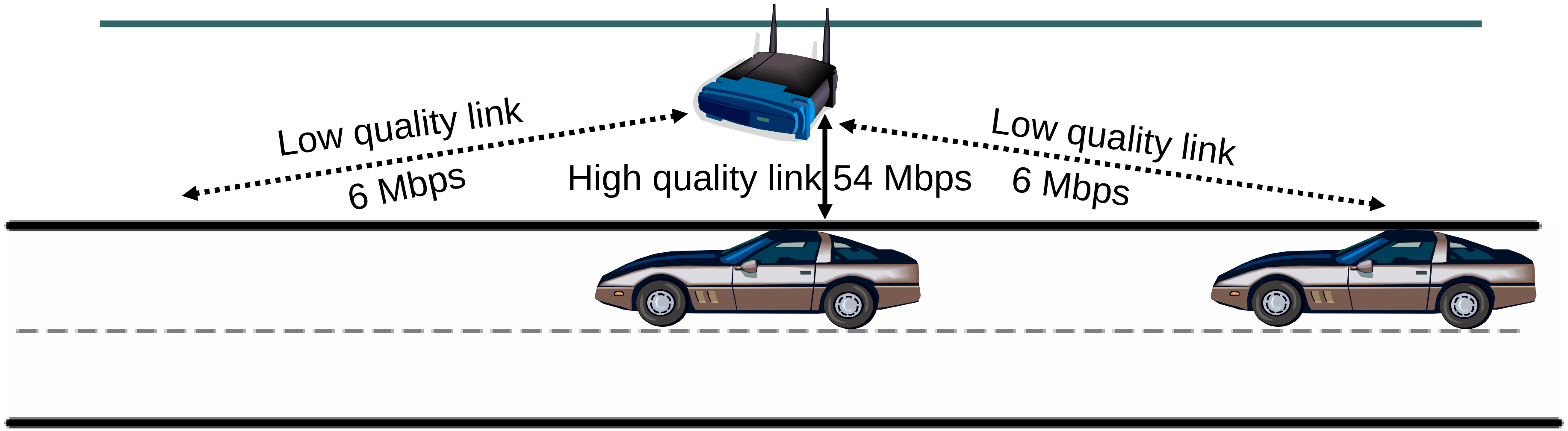
Requirement: High throughput

What is rate selection?

- 802.11 PHY: multiple transmission rates
  - 8 bitrates in 802.11a/g (6 – 54 Mbps)
  - 8 bitrates in 802.11p (3 – 27 Mbps)
- Different modulation and coding schemes

Link Quality \ Bitrate	Low	High
Low	✓	High Error Rate
High	Underutilization	✓

Rate selection problem in vehicular networks



Rate Selection:

Select the best transmission rate based on link quality in real-time to obtain maximum throughput

Outline

- Introduction
- **Existing solutions**
- CARS: Context Aware Rate Selection
- Evaluation
- Conclusion

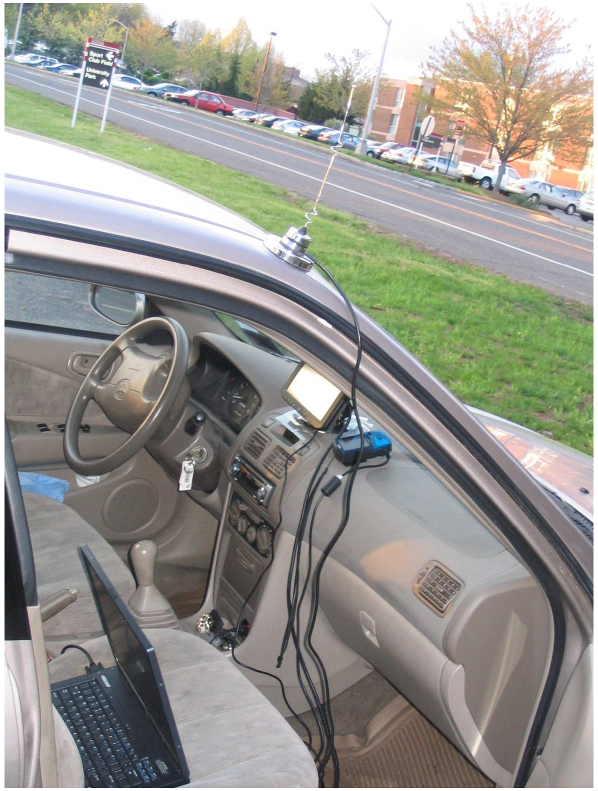
Existing rate selection algorithms

- ARF (1996), RBAR (2001), OAR(2004), AMRR (2004), ONOE (2005), SampleRate (2005), RRAA (2006) (and many more…)
- Basic scheme in all existing algorithms
  - **Estimation:** Use physical layer or link layer metrics to estimate the link quality
  - **(Re)Action:** Switch to lower/higher rate

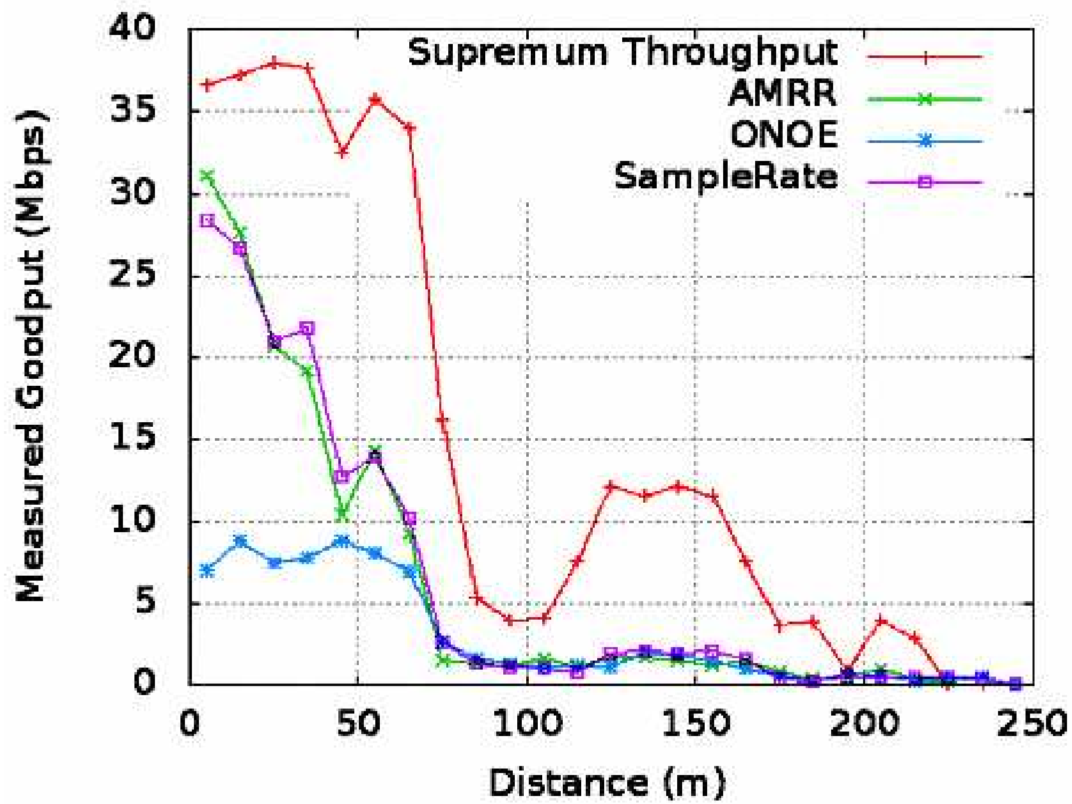
Question: How well do these algorithms work in vehicular environments?

Existing schemes + vehicular networks: Experiment

- Outdoor experiments comparing
  - SampleRate [2005]
  - AMRR [2004]
  - ONOE [2005]
- 5 runs per rate algorithm
- 5 runs per fixed rate
- Slow Mobility: 25 mph
- Metrics
  - Average goodput
  - Supremum goodput (maximum among all runs for all rates)



Existing schemes + vehicular networks: Results



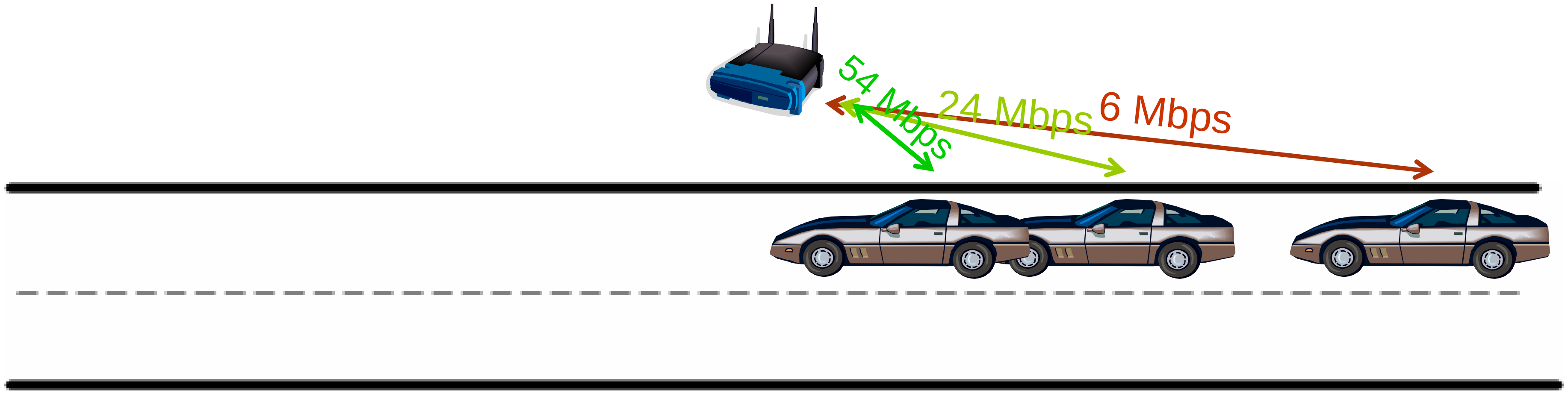
Underutilization of link capacity

Existing schemes + vehicular networks: Analysis

- Rapid change in link quality due to distance, speed, density of cars
- Problems:

1. Estimation delay
2. Sampling requirement
3. Collisions vs. channel errors

Problem 1: Estimation delay



Link conditions change faster than the estimation window - the rate adaptation *lags* behind

Problem 2: Sampling Requirement

- When an idle client starts transmitting, there are no recent samples in the estimation window
- Packet scheduling causes bursty traffic
- Results in anomalous behavior

Problem 3: Collisions vs. errors

- Hidden-station induced losses should not trigger rate adaptation [CARA06, RRAA06]
- Lower rate prolongs packet transmission time, aggravating channel collisions
- Use of RTS/CTS causes additional overhead

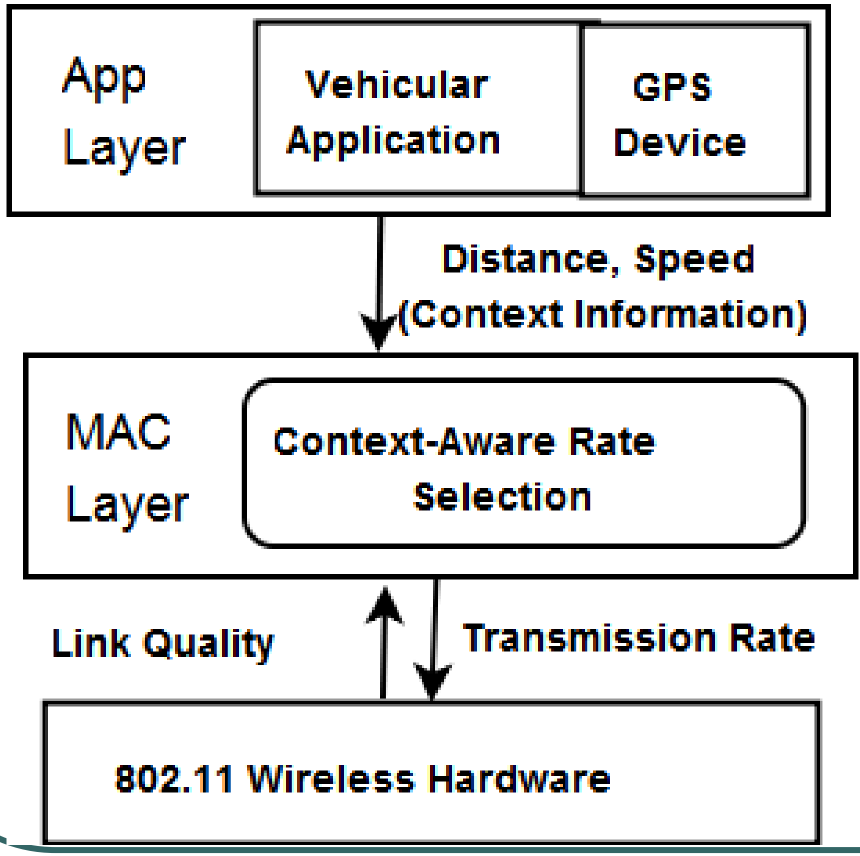
Outline

- Introduction
- Existing solutions
- **CARS: Context Aware Rate Selection**
- Evaluation
- Conclusion

CARS at a glance

- Rapid change in link quality due to distance, speed (context)
- Vehicular nodes already have this context information
- Use this cross-layer information at the link layer to estimate link quality and perform *proactive* rate selection

CARS: reactive + proactive



Link Quality: Error Function

$E_C = f(distance, speed, bitrate, len)$

- Proactive
- Predicted error as a function of context information

$E_H = f(bitrate, len)$

- Reactive
- Short-term loss statistics from estimation window

Proactive rate selection using E_c

$E_C = f(distance, speed, bitrate, len)$

- **Model** link error rate as a function of context information and transmission rate
  - Empirically derived using data from outdoor experiments
- Simple model is sufficient because of discrete rates in 802.11
- Context recalculation frequency = 100 ms

CARS Algorithm

Function *CARS_GetRate*

Input: ctx, α, len

Output: *rate*

for all $rate_i$ **do**

$\quad PER_i = \alpha.E_C(ctx, rate_i, len) + (1 - \alpha).E_H(rate, len)$

$\quad Thr_i = f(rate_i, PER_i, num\_retr)$

end for

Return $rate_i$ with maximum Thr_i

CARS Implementation

- The CARS algorithm was implemented on the open-source MadWifi wireless driver
  - ~ 520 lines of C code
- Context information obtained from TrafficView [2004]
  - Generic */proc* interface:
  - Any other app can be extended to provide a similar interface
- Extensively tested by means of vehicular field trials and simulations

Outline

- Introduction
- Existing solutions
- CARS: Context Aware Rate Selection
- **Evaluation**
- Conclusion

CARS Evaluation

- **Effect of Mobility:** How does CARS adapt to fast changing link conditions? (Field trial)
- **Effect of Collisions:** How robust is CARS to packet losses due to collisions? (Field trial)
- **Effect of Density of Vehicles:** How does the throughput improvement scale over large number of vehicles? (Simulation study)

Effect of mobility: Setup

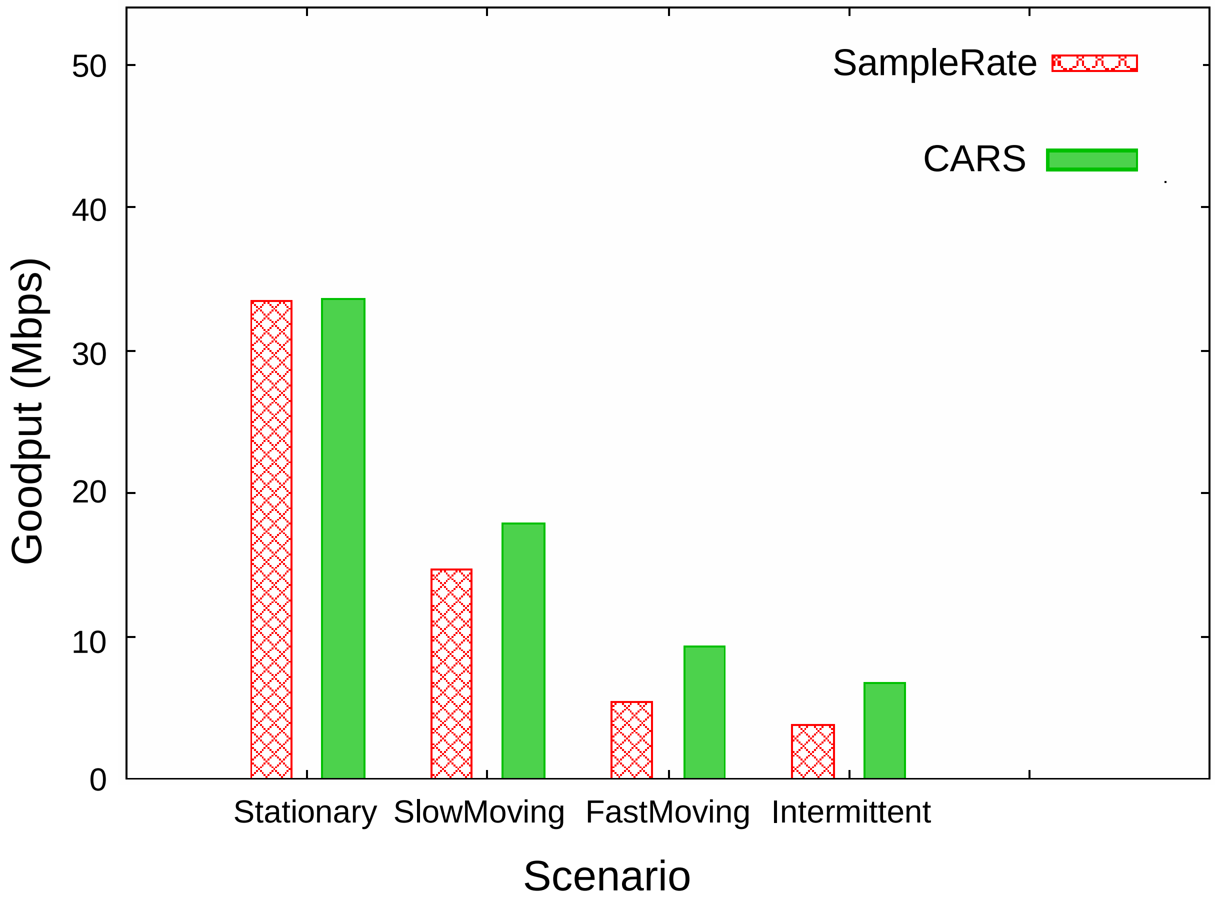
Scenarios

- **Stationary**: Base case
  - Cars are stationary next to each other.
- **SlowMoving**: A simple moving scenario
  - Cars are driving around the Rutgers campus: ~25mph speeds
- **FastMoving**: A more stressful moving scenario
  - Cars are driving on New Jersey Turnpike: ~70mph speeds in high car/truck traffic conditions
- **Intermittent**: A scenario with intermittent connectivity
  - Cars move in and out of each other's range periodically - Hot-spot scenario

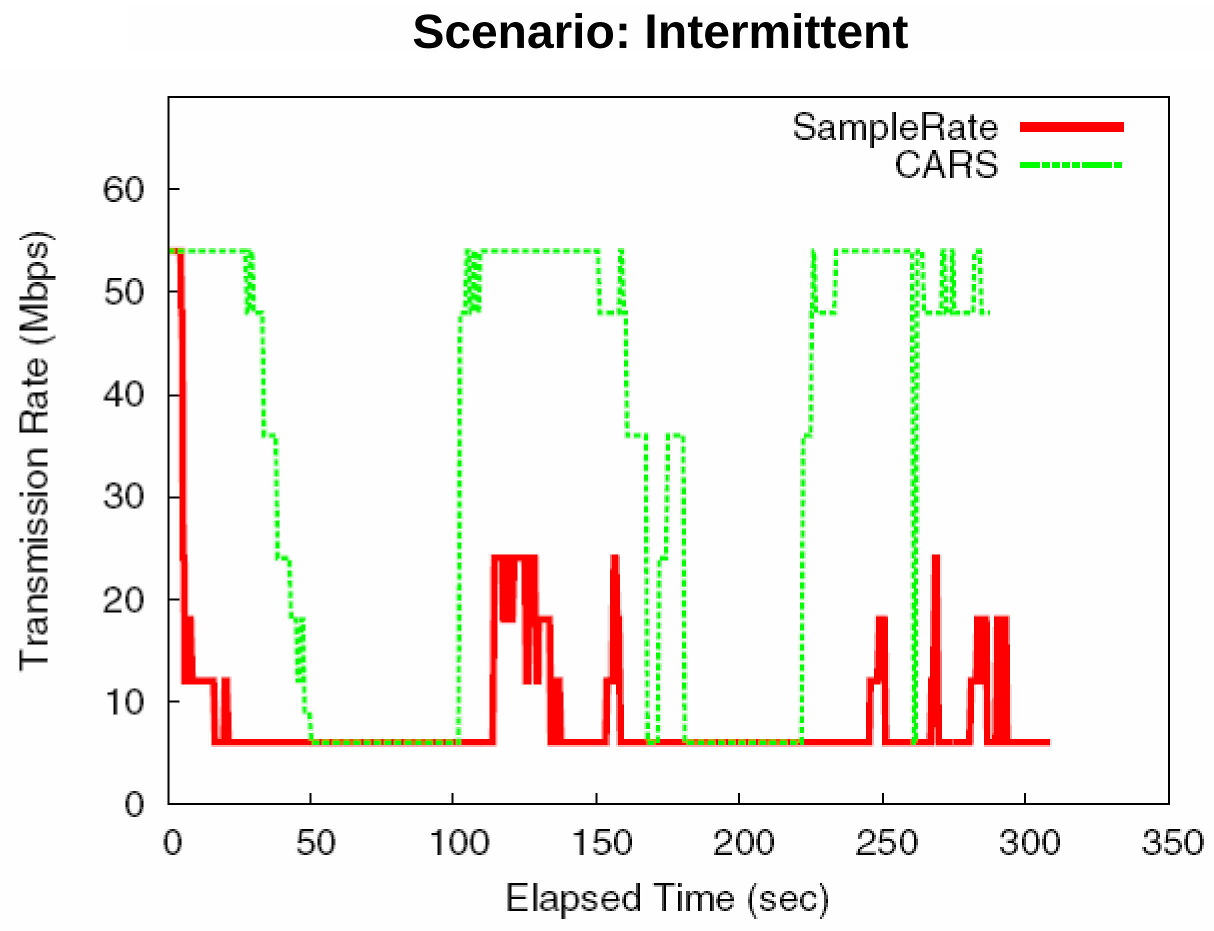
Workload:

- UDP traffic from TX to RX using iperf
- Duration of experiment - 5 minutes

Effect of mobility: Results



Effect of mobility: Analysis

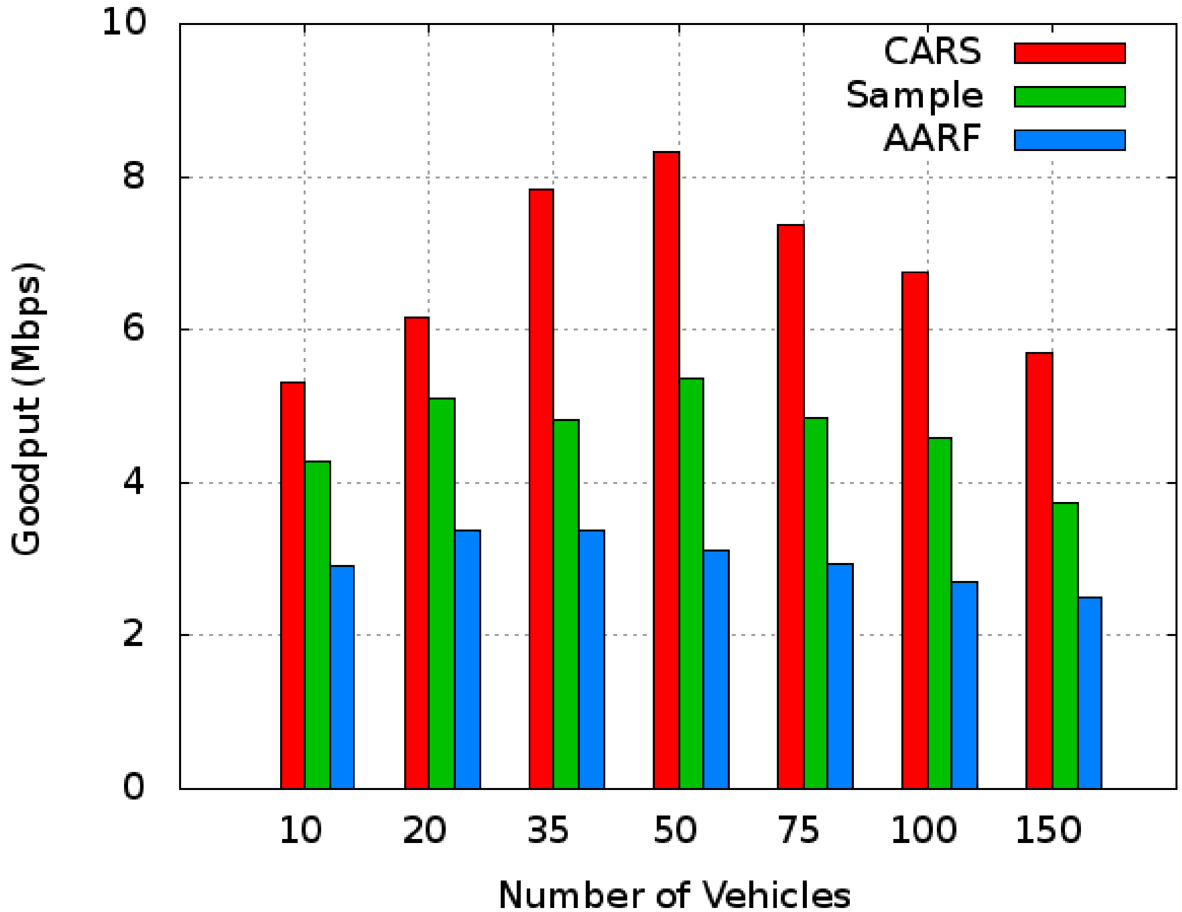


Reactive vs. Proactive

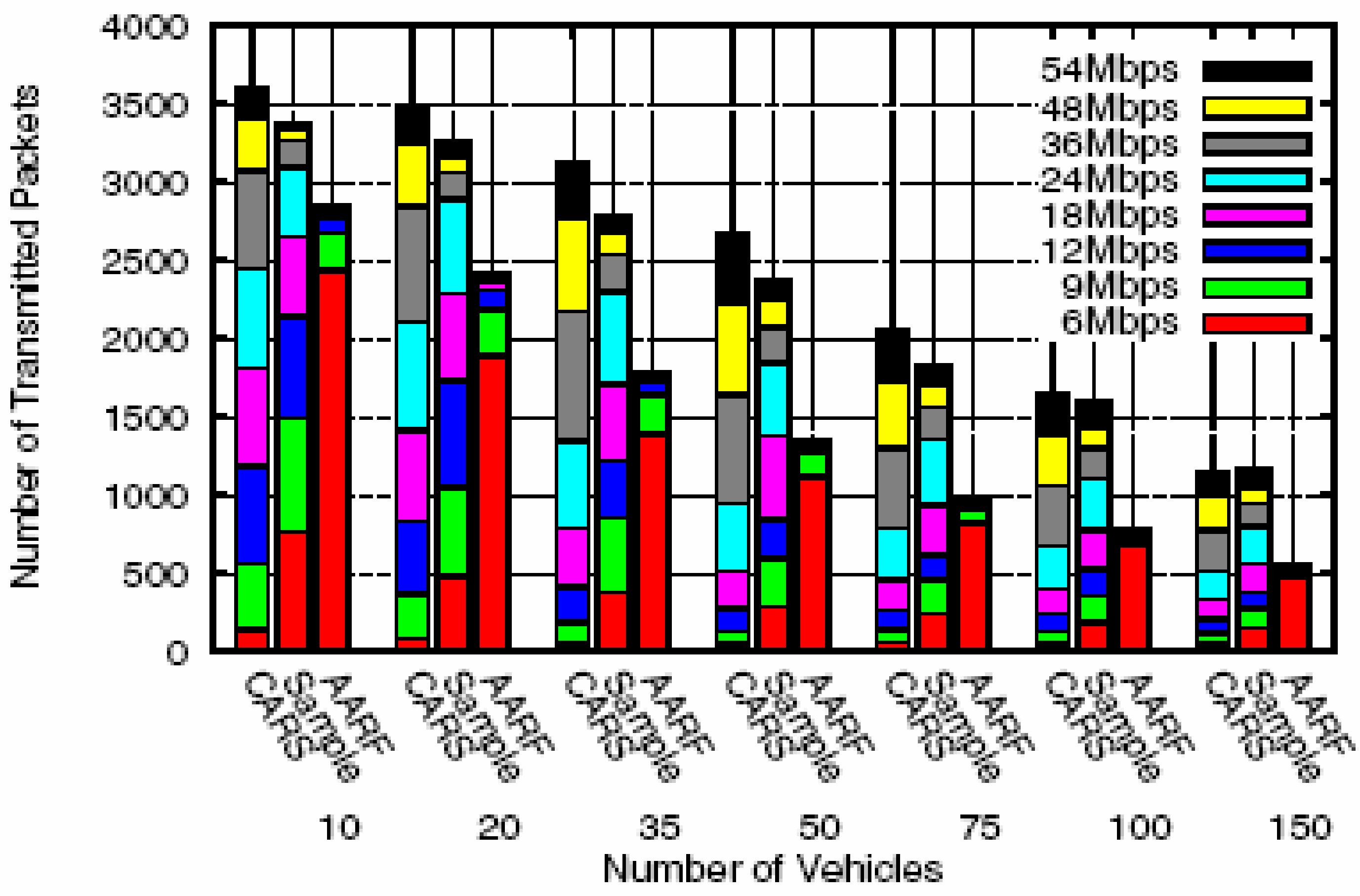
Effect of vehicle density - Setup

- Hotspot scenario:
  - Road of length 5000 m with multiple lanes
  - Base station in the middle of the road
- Workload:
  - Video stream: 1500 packets of size 1000 bytes each
  - UDP: transmission rate 100 packets per second
  - RTS/CTS disabled
  - Max_retransmits: 4
- ns-2 with microscopic traffic generator
  - Compared CARS with AARF and SampleRate

Effect of vehicle density - Results



Effect of vehicle density - Analysis



Outline

- Introduction
- Existing solutions
- CARS: Context Aware Rate Selection
- Evaluation
- **Conclusion**

Conclusion

- Existing rate adaptation algorithms under-utilize vehicular network capacity
- CARS: uses context information to perform *fast* rate selection
- Significant goodput improvement over existing algorithms