Scheduling in Wireless

The Central Question

In an interference environment, who can talk in each time slot?
Problem Space

Given: who interfere with whom

- Topology $G = (V, L)$
- Interference model/representation (graph, set, matrix)

Variables: who talks when

- Activation vector $s$
- Contention probability $\lambda, \rho$
- Holding time $\mu$
Problem Space

Goal:
- Stable
- Small delay
- Big utility
- Fair

Stochastic:
- Workload arrival
- Algorithm
- Channel
Problem Taxonomy

---

local or e2e

---

hop-based or SINR-based interference model

---

saturated or non-saturated traffic

---

collision-free or collision-based
Theory-Practice Gap

Simple ones in use (analysis can be challenging):
- Aloha
- CSMA
- RTS/CTS

(Hundreds of) Sophisticated algorithms in theory, based on graph, optimization, game, queueing theories
Max Weight Scheduling

Where We Are In The Tree

- hop-based SINR-based interference model
- local or e2e
- saturated or non-saturated traffic
- collision-free or collision-based
Maximum Weight

- **Tassiulas Ephremides 1992**

The max-weight algorithm is choosing the $s^*(t)$ at each slot $t$:

$$s^*(t) = \arg \max_{s \in S} W(s), \quad W(s) \triangleq \sum_{l \in L} Q_l(t)s_l.$$  

$S$: Set of feasible schedules

$Q_l(t)$: Queue size on link $l$ at time $t$

**Throughput-optimal, Maximum stability region**

- Connections to:

  Prior work: Hajek Sasaki 1988 (known arrivals)

  Graph theory: HP-hard Maximum Weighted Independent Set

  Switching theory

- General yet complex. How to make it simple and distributed?
Approximation: Maximal Weight

Suboptimal matching that can’t be increased by activating more links:

- **Greedy**: the link $l$ with the largest queue length
- **Locally-greedy**: a random link $l$ with a locally-longest queue length

![Link Conflict Graph, Maximal, Locally greedy, Greedy graphs]
Approximation: Maximal Weight

\[ \gamma = 1/2 \ (K = 1) \]: Chaporkar Kar Sarkar 2006, Wu Srikant 2006

\[ 2/3 \ (K = 1, \text{tree}) \]: Sarkar Kar 2006

1: NP-hard in general \((K > 1)\): Sharma, Mazumdar, Shroff 2006

\[ 1/(\text{maximum interference degree}) \] Wu Srikant Perkins 2007, Chaporkar Kar Sarkar 2007: 1/8 for geometric graph

Further approx: Gupta Lin Srikant 2007


Distributed: Israeli Itai 1986, Heopman 2004
Randomization: Pick and Compare

- Centralized: Tassiulas 1998

At each time slot $t$, the $\gamma$-RPC first generates a random schedule $s'(t)$ satisfying $P$, and then schedule $s(t)$ defined in $C$:

$P \quad \exists 0 < \delta \leq 1, \text{ s.t. } \text{Prob}(s'(t) = s|Q(t)) \geq \delta$, for some schedule $s$, where $W(s) \geq \gamma W^*(t)$

$C \quad s(t) = \arg \max_{s = \{s(t-1), s'(t)\}} W(s)$

- Message passing with gossip: P and C can be inaccurate

$\gamma = 1 (K = 1, \text{ not counting complexities})$: Modiano Shah Zussman 2006
Where We Are In The Tree

- hop-based SINR-based interference model
- local or e2e
- saturated or non-saturated traffic
- collision-free or collision-based
Message Passing Random Access

$K = 1$. Each slot starts with constant $M$ minislots for control signals

- Compute $0 \leq x_l(t) \leq 1$ using queue lengths of the interfering neighbors via message passing:

$$x_l(t) = \frac{Q_l(t)}{\max\left[\sum_{k \in L(t(l))} Q_k(t), \sum_{k \in L(r(l))} Q_k(t)\right]}$$

- The link $l$ contends each mini-slot with the probability $p_l = f(x_l(t), M)$ for some $f$ (e.g., $g(M)x/M, 1 - \exp(-g(M)x/M)$)

- Successfully contended link transmits during the time slot $1/3 - 1/M$: Lin Rasool 2006

$1/2 - 1/\sqrt{M}$: Joo Shroff 2007

$1/2 - \log(2M)/2M$: Gupta Lin Srikant 2007

Further study: Marbach Eryilmaz Ozdaglar 2007, Joo Lin Shroff 2008
3D Tradeoff

Throughput

Delay

Complexity

O(2^L)

Maximum Weight
Greedy/Maximal

1

RPC (\gamma=1)

\gamma-RPC

stretching
Our Path

Where We Are In The Tree

- hop-based SINR-based interference model
- local or e2e
- saturated or non-saturated traffic
- collision-free or collision-based
Notation

Interference (0-1 matrix): \( A \)
Schedule (0-1 vector): \( s \)
Set of feasible schedules: \( S(A) \)
Time fraction of activation: \( \pi_s \)
Throughput: \( x_l \)
maximize \( \sum_l U_l(x_l) \)

subject to

\[
\begin{align*}
    x_l & \leq \sum_{s \in S: s_l=1} \pi_s, \quad \forall l \\
    \pi_s & \geq 0, \quad \forall s \\
    \sum_{s \in S} \pi_s & = 1
\end{align*}
\]

variables \( \{x_l, \pi_s\} \)
Reverse Engineering Exponential Backoff

- Reverse engineer as a game (derive utility function)
- Nash equilibrium exists but suboptimal
- Existing protocol is stochastic subgradient
- Converges under conditions on how interfered the topology is

Lee Chiang Calderbank 2007
Reverse Engineering Exponential Backoff

- Contrast to reverse engineering of TCP congestion control into NUM
- Self interests not aligned
- How to align them? Maybe with the help of message passing?
Problem Statement

$L_{out}(n)$: set of logical links where node $n$ is transmitter

$N(l)$: set of nodes whose transmission collide with that on $l$

Each link with a utility function $U_l(x_l)$ and fixed rate $c_l$

$$x_l = c_l p_l \prod_{k \in N(l)} (1 - P^k)$$

Optimization over variables $(p, P)$:

**maximize**

$$\sum_l U_l(c_l p_l \prod_{k \in N(l)} (1 - P^k))$$

**subject to**

$$x_l^{min} \leq c_l p_l \prod_{k \in N(l)} (1 - P^k) \leq x_l^{max}, \forall l$$

$$\sum_{l \in L_{out}(n)} p_l = P^n, \forall n$$

$$P^{min} \leq P^n \leq P^{max}, \forall n, 0 \leq p_l \leq 1, \forall l$$
How Distributed Can Solution Be

- Step 1: log change of variable to decouple
- Step 2: dual decomposition
- Step 3: $\alpha \geq 1$ utility function to ensure global optimality

- How to make it converge faster?
  Stepsize-free algorithm

- How to reduce message passing to zero?
  Learn from historical record of collisions
  Optimal for fully-interferred topology and sufficient number of nodes

Mohsenian-Rad Huang Chiang Wong 2009
Motivation

- Message passing undesirable
- Not fully distributed, overhead
- Security
- Synchronization
- Slotted Aloha version well-understood

What can be achieved without message passing?

Can we actually use it?
Theme 1: Simplicity-driven Design

How distributed is distributed?
- How often? Time-complexity
- How far? Space-complexity
- How many bits per message? Bit-complexity

Optimality-driven design:
Must have optimality proof, find simplest protocol

Simplicity-driven design:
Zero message passing, find best performance protocol
Related Work

- Jiang Walrand 2008
- Rajagopalan Shah 2008
- Liu Yi Proutiere Chiang Poor 2008
- Zhang Shroff 2010
- Ni Srikant 2009
- Eryilmaz Marbach Ozdaglar 2008
- Kelly 1987
- Hajek 1988
Algorithm and Theory
Wireless Scheduling: How Good Can CSMA Be?

**CSMA:** Carrier Sense Multiple Access:
When to contend, and How long to hold the channel

Adaptive CSMA without message passing:
Adjust contention and holding time $(\lambda, \mu)$
Theme 2: Timescale Assumption

Timescale separation assumption:
Network state converges to stationary distribution before parameter update

Real system does not obey this assumption
Algorithm

Update “virtual queue length” based on service rate

No message passing needed:

\[ q_l[t + 1] = \left[ q_l[t] + \frac{b[t]}{q_l[t]} \left( U_l'[q_l[t]] - D_l[t] \right) \right]_{q_{min}}^{q_{max}} \]

Adjust Poisson contention rate or exponential holding time

\[ \frac{\lambda_l[t + 1]}{\mu_l[t + 1]} = \exp(q_l[t + 1]) \]
Performance

Algorithm converges to \( \lim_{t \to \infty} q[t] = q^* \) such that \( x(q^*) \) solves

\[
\begin{align*}
\text{maximize} & \quad V \sum_l U_l(x_l) - \sum_s \pi_s \log \pi_s \\
\text{subject to} & \quad x_l \leq \sum_{s: s_l = 1} \pi_s, \quad \forall l \\
& \quad \pi_s \geq 0, \quad \forall s \\
& \quad \sum_s \pi_s = 1
\end{align*}
\]

Approximation error bounded by \( \log |S|/V \)

Pick \( V \) large enough and grows \( \mathcal{O}(L) \)
Proof Outline

A stochastic subgradient algo. modulated by a Markov chain

Step 1: show averaging over fast timescale is valid
Interpolation of discrete \( q \) converges a.s. to a continuous \( q \) solving a system of ODE

Step 2: show the resulting averaged process converges
The system of ODE describes the trajectory of subgradient solving the dual of the approximation problem

Step 3: standard results in convex optimization and duality to show convergence and optimality
Step 1 Illustration

$q[t]$

Original system

$\bar{q}[t]$

Continuous interpolation

$\tilde{q}[t]$

ODE system

equal for large $t$, i.e.,

$\bar{q}[t] = \tilde{q}[t]$, a.s.

for large $t$

Trajectories with the service rate by CSMA being stationary (i.e., averaged)
Given sequence $x_n$ of random real numbers, and random variable $Y_n$,

$$x_{n+1} = x_n + b_n h(x_n, Y_n)$$

$h$ is bounded, continuous, Lipschitz (to first variable)

$Y_n$ is Markov chain whose kernel evolves in time and depends on $x_n$:

$$\text{Prob}[Y_{n+1} = z | Y_n = y, x_n = x] = p(z|y, x)$$

Kernel $p$ of a stationary, ergodic Markov chain with stationary distribution $\pi_x$

Let $\bar{x}$ be interpolated $x$, and $\tilde{x}^s$ be solution to the following ODE:

$$\frac{dx(t)}{dt} = \sum_y \pi_x(t)(y)h(x(t), y), \quad \tilde{x}^s(0) = \bar{x}(s)$$

Then, a.s.,

$$\lim_{s \to \infty} \sup_{t \in [s, s+T]} |\bar{x}(t) - \tilde{x}^s(t)| = 0$$
Initial Experimentations

(Small, Fully Connected)
Theme 3: Falsifiable Theory

- Implement and deploy theory-driven scheduling algorithm on top of conventional hardware
- Discover, quantify, and bridge the gaps between theory and practice in wireless scheduling
Implementation over 802.11 madwifi driver

Network Layer

UO-CSMA (Overlay)

MAC Adaptor

802.11 (Substrate)

FIFO Queue

Wireless (Radio)
Work-Around Solutions

• Ensuring correct holding time
  • Reduce high priority beaconing
  • MAC prioritization by AIFS and CW_{min}, CW_{max}
  • NAV option via overhearing
• Making CW=0 feasible
  • CW can only be a power of 2
  • Similar transmission chance as CW=1
**Basic Performance**

Throughput Deviation (%)

- OPT
- SIM
- EXP(V=500)
- 802.11 DCF

Values:
- OPT: 0%
- SIM: 4%
- EXP(V=500): 6.6%
- 802.11 DCF: 14.9%
Theory Predictions: Stepszie Choice

Flow 9

Fixed step size

Flow 9

Decreasing step size

Flow 10

Flow 10
Theory Predictions: V Choice

![Graph showing utility and queue size changes with V parameter changes.]
Theory Prediction: \( W() \) Choice

- Changing weight function with fixed step size
  - \( W(x) = x \)
  - Converges in 80 sec
  - Small backlog

- \( W(x) = \log\log(x) \)
  - Converges in 40 sec
  - Large backlog
Theory-Practice Gaps

Theory

Simulation

Implementation

Legacy

collision

SIR based receiving with capture
asymmetric sensing
control overhead

time-slotted interference
asynchronous transmission
backoff granularity
imperfect channel holding
Theory-Practice Gaps

Assumed away:
- asymmetry, overhead, control granularity

Modeled simplistically:
- imperfect sensing
- SIR collision model with capture
- imperfect holding

Analyzed loosely:
- convergence speed
- transient behavior
- parameter choice
Further Experimentations
**Basic Setup**

(a) Fixed testbed node  
(b) Mobile testbed node

<table>
<thead>
<tr>
<th>Transmission rate</th>
<th>2 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet size</td>
<td>1000 bytes</td>
</tr>
<tr>
<td>Traffic pattern</td>
<td>Concurrent, fully-backlogged flows</td>
</tr>
<tr>
<td>Weight function $W(x)$</td>
<td>$x$</td>
</tr>
<tr>
<td>$V$</td>
<td>200</td>
</tr>
<tr>
<td>Utility function</td>
<td>$\log(x)$</td>
</tr>
</tbody>
</table>
Basic Setup

Baseline of **Fully Connected**
Pick V s.t. oCSMA throughput is similar to 802.11
Flow in the Middle (all asyn. transmission)

oCSMA solves the Flow-in-the-Middle problem. Aggressiveness adaptation generates enough transmission opportunities for the flow in the middle
**Fundamental Difference**

802.11 DCF:

collision -> backoff

oCSMA:

collision -> under-service -> aggressive
With high collision probabilities, symmetric increase of contention from both flows still lead to self-sustaining loop of performance degradation.
Disadvantaged flow has incomplete information and cannot take full advantage of transmission opportunities of oCSMA due to short silence time.
Do We Know Interference?

Capture when we think it should collide
Collide when we think it shouldn’t
The Real Life of Interference

signal strength

time
**Channel Variations: Downlink**

![Diagram of a network topology](image)

**Figure 2.** Atomic topologies used to separate factors related to channel asymmetry. In the diagram, vertices represent network nodes, dotted lines represent the ability of nodes to carrier sense each other, and arrows represent asymmetric transmission effects. In each topology, an edge from a vertex indicates that a carrier-sense transmission over the corresponding side link occurs. The start of transmission of side flows is synchronized with each other. Therefore, the central transmitter defers its transmissions in favor of the flows that occur earlier.

**Figure 3.** Experimental testbed hardware.

**Table 1.** Channel Variations:

<table>
<thead>
<tr>
<th>RSSI Difference (dBm)</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput (kbps)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highest signal strength flow</td>
<td>1400</td>
<td>1200</td>
<td>1200</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>Lowest signal strength flow</td>
<td>1400</td>
<td>1200</td>
<td>1200</td>
<td>800</td>
<td>800</td>
</tr>
</tbody>
</table>

(a) 802.11

(b) Optimal CSMA

**Graph 1.** Throughput (kbps) vs. Difference in RSSI (dBm) for 802.11 and Optimal CSMA.

**Graph 2.** Performance comparison of 802.11 and Optimal CSMA under varying channel conditions.

oCSMA does not change 802.11 behavior: per-flow queues but still sharing the same transmitter
**Channel Variation: Uplink**

Channel variation is a key aspect to be considered in the analysis of Uplink scenarios.

**Figure 1.** Atomic topologies used to separate topological factors and study them. 

- (a) AP-DL: An access point with two clients in CS range of each other.
- (b) HT: An access point with uplink traffic and clients in CS range of each other.
- (c) IA: An access point with uplink traffic and downlink traffic.
- (d) FIM: A Flow-in-The-Middle topology where transmitters and receivers span over two links.

**Figure 2.** Experimental evaluation using oCSMA.

- (a) 802.11: The throughput distribution for highest and lowest signal strength flows under different difference in RSSI (dBm).
- (b) Optimal CSMA: The throughput distribution for highest and lowest signal strength flows under different difference in RSSI (dBm).

**Graphs:**
- The throughput (kbps) is shown on the y-axis, and the difference in RSSI (dBm) is shown on the x-axis.
- Error bars represent the 95% confidence intervals.

**Observations:**
- oCSMA does not take advantage of channel quality differences.
- The scenarios with HTs are dominated by physical layer capture over collision.
- In scenarios with hidden terminals and asymmetry, oCSMA manifests as an asymmetric interference relation.
- Throughput experiments are performed using a transmitter and two clients in relation to the difference in RSSI measured at each of them.
oCSMA cannot resolve starvation due to capture effect in channel asymmetry, as *silent time not long enough*
Coupling with TCP

<table>
<thead>
<tr>
<th><strong>PHY</strong></th>
<th>802.11a, 5.805 GHz band, 6 Mbps rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TCP versions</strong></td>
<td>Reno, Tahoe, NewReno, SACK</td>
</tr>
<tr>
<td><strong>TCP MSS</strong></td>
<td>512, 1024 bytes</td>
</tr>
<tr>
<td><strong>TCP CWND\text{max}</strong></td>
<td>16, 64, 256 kB</td>
</tr>
<tr>
<td><strong>Step size, Weight function</strong></td>
<td>( b[t] = 0.01, W(x) = x )</td>
</tr>
</tbody>
</table>

**Table III**

We have also performed experiments for HT and IA scenarios. The results are similar to those in UBC+oCSMA. An inter-flow's optimal oCSMA provably guarantees optimality. However, TCP is a window-based mechanism, where transmissions are clocked by acks. Its rate is controlled by AIMD (Additive Increase Multiplicative Decrease), the bounded number of in-queue packets. The problem of TCP+oCSMA is that oCSMA's efficiency is sufficiently exploited, thus the total throughput increases. To demonstrate, consider the FC scenario in Fig. 1a. Fig. 10 shows how oCSMA+TCP behaves for bounded CWND and physical queue size. We observe that the channel is under-utilized due to TCP's starvation in another flow even for small values of CWND.

Small CWND and large RTT starves the flow in the middle. This is in stark contrast to bounded contention aggressiveness, thereby little competition for the channel leads to fairness. However, with large buffer sizes, fairness is almost guaranteed, which, however, comes at the cost of under-utilization of channel. With 256 kB buffer size, fairness is almost guaranteed, which, however, comes at the cost of under-utilization of channel. With 256 kB buffer size, fairness is almost guaranteed, which, however, comes at the cost of under-utilization of channel.

To pull back to optimality, oCSMA needs high congestion level, increasing RTT and hurting TCP injection of packets.
Large buffer size of oCSMA, and the MAC design principles behind it, in conjunction with low channel quality due to the lack of knowledge about oCSMA solves this problem creating frequent transmission terminal scenarios, physical layer capture tends to dominate communication becomes impossible; 2) in practical hidden protocol to a state of extreme contention aggressiveness where with implementation on hardware. The key findings include; realistic wireless network conditions, through experimentation even with small service rate, significantly different from those completely starved because of small CWND and large RTT under-utilized due to bounded contention aggressiveness, but oCSMA and UBC. With small buffer size, the channel is inadequately packet injection due to ack-clocking. Throughput (kbps) TCP conflicts with oCSMA. FIM topology.

Small buffer size leads to under-utilization. Large buffer size leads to starvation: middle flow sees inadequate packet injection due to ack-clocking.
Next Steps

- Re-analysis
  - Sensing, interference
  - RF, TCP impact
- Re-design
  - Message passing come to help?
  - Sufficient statistics from history of collision?
- Re-implement
Discrete time-slot

More realistic than Poisson clock model
Collision (in addition to algorithmic inefficiency)

Form a sequence of systems converging to Poisson model
Scale both contention probability and channel holding time
Need Virtual Sensing

Efficiency-Fairness Tradeoff without virtual sensing:

Utility gap: $\delta$

bound on suboptimality

Short-term fairness: $\beta$

$1 / \text{ave. number of periods of no transmission}$

$\beta \leq \frac{\delta}{C_1 \exp(C_2/\delta)}$
**Redesign: First Step**

**Reservation phase**: virtual carrier sensing by oCSMA

**Data phase**: increased channel holding time

Avoid collision/overlap
3.4 Information asymmetry

In IA topology, RTS/CTS signaling significantly helps improving the throughput of a disadvantaged flow. This enhancement is manifested when the oCSMA is used. More importantly, increasing holding time achieves almost fair throughput distribution between two flows, which is exactly what we expect in the IA scenario.

3.5 Flow in the middle

In addition to the FC case, we also check whether the redesigned oCSMA is working or not in FIM topology. Fig. 7 demonstrates that the redesigned oCSMA have better fairness and efficiency than the pure oCSMA.

3.6 Discussion

There may be some weaknesses due to enlarged data transmission. Typically, the short-term fairness is getting worse as holding time increases. This, for instance, could make TCP flow have bad throughput due to abruptly increased RTT. Moreover, when there are many contending hidden nodes in the network and the number of contenders is not reflected to the access probability, we cannot be sure how much holding time extension is helpful.

4. REFERENCES


3.5 Flow in the Middle

In addition to the FC case, we also check whether the redesigned oCSMA is working or not in FIM topology. Fig. 7 demonstrates that the redesigned oCSMA have better fairness and efficiency than the pure oCSMA.

3.6 Discussion

There may be some weaknesses due to enlarged data transmission. Typically, the short-term fairness is getting worse as holding time increases. This, for instance, could make TCP flow have bad throughput due to abruptly increased RTT. Moreover, when there are many contending hidden nodes in the network and the number of contenders is not reflected to the access probability, we cannot be sure how much holding time extension is helpful.

4. REFERENCES


3.4 Information asymmetry

In IA topology, RTS/CTS signaling significantly helps improving the throughput of a disadvantaged flow. This enhancement is manifested when the oCSMA is used. More importantly, increasing holding time achieves almost fair throughput distribution between two flows, which is exactly what we expect in the IA scenario.

3.5 Flow in the middle

In addition to the FC case, we also check whether the redesigned oCSMA is working or not in FIM topology. Fig. 7 demonstrates that the redesigned oCSMA have better fairness and efficiency than the pure oCSMA.

3.6 Discussion

There may be some weaknesses due to enlarged data transmission. Typically, the short-term fairness is getting worse as holding time increases. This, for instance, could make TCP flow have bad throughput due to abruptly increased RTT. Moreover, when there are many contending hidden nodes in the network and the number of contenders is not reflected to the access probability, we cannot be sure how much holding time extension is helpful.

4. REFERENCES


Hidden Terminal w. Capture

Figure 3: Throughput comparison in FC topology

Figure 4: Throughput comparison in symmetric HT topology

Figure 5: Throughput comparison in HT with physical capture topology

Figure 6: Throughput comparison in IA topology

Figure 7: Throughput comparison in FIM topology

3.4 Information asymmetry
In IA topology, RTS/CTS signaling significantly helps improving the throughput of a disadvantaged flow. This enhancement is manifested when the oCSMA is used. More importantly, increasing holding time achieves almost fair throughput distribution between two flows, which is exactly what we expect in the IA scenario.

3.5 Flow in the middle
In addition to the FC case, we also check whether the redesigned oCSMA is working or not in FIM topology. Fig. 7 demonstrates that the redesigned oCSMA have better fairness and efficiency than the pure oCSMA.

3.6 Discussion
There may be some weaknesses due to enlarged data transmission. Typically, the short-term fairness is getting worse as holding time increases. This, for instance, could make TCP flow have bad throughput due to abruptly increased RTT. Moreover, when there are many contending hidden nodes in the network and the number of contenders is not reflected to the access probability, we cannot be sure how much holding time extension is helpful.

Completing the model-theory-implement-data-model loop pays off

References


The Whole Loop

Language → Model → Data → Implement → Deploy → Language

Theory → Model → Data → Implement → Deploy → Theory