Utility-Optimal Wireless Random MAC Without Message Passing

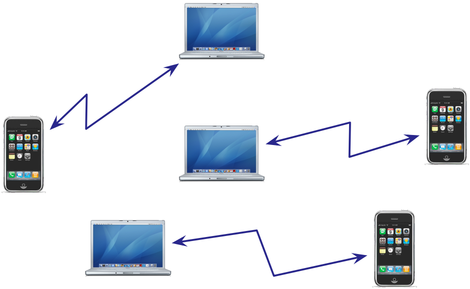
Jianwei Huang

Department of Information Engineering

Acknowledge: A.H. Mohsenian-Rad, M. Chiang, and V.W.S. Wong

ITCSC Seminar
October 2008

Wireless MAC Protocols



- Coordinate multiple wireless users accessing the same channel
  - Centralized: scheduling-based MAC (e.g., cellular network)
  - Distributed: contention-based random MAC (e.g., ad hoc network)

History of Wireless Random MAC

- Studied for 30 years
- Simplicity and practicality
- Many variations
  - Some achieved great success
  - Many are engineering ad hoc designs

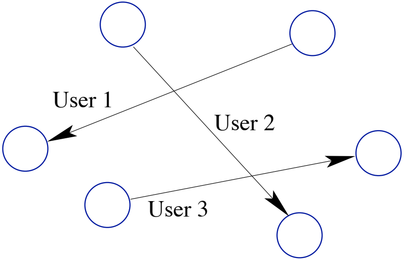
History of Wireless Random MAC

- Studied for 30 years
- Simplicity and practicality
- Many variations
  - Some achieved great success
  - Many are engineering ad hoc designs
- **Our objective**
  - Establish a theoretical framework
  - Design good and simple random access protocols

Our focus: Aloha

Network Model

(A Simple) Network Model



- A set of $\mathcal{N} = \{1, \ldots, N\}$ single-hop users
- Full interference topology
- Each user i
  - Contend the channel with probability $p_i \in \mathcal{P}_i = [P_i^{\min}, P_i^{\max}]$
  - Maximum data rate γ_i
  - Long term average data rate

$$r_i(\mathbf{p}) = \gamma_i p_i \prod_{j \in \mathcal{N} \setminus \{i\}} (1 - p_j)$$

Network Utility Maximization

- Each user i has an increasing and concave utility function $u_i(r_i)$

System Objective: Network Utility Maximization (NUM)

$$\max_{\mathbf{p} \in \mathcal{P}} \sum_{i \in \mathcal{N}} u_i(r_i(\mathbf{p})),$$

Network Utility Maximization

- Each user i has an increasing and concave utility function $u_i(r_i)$

System Objective: Network Utility Maximization (NUM)

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{i\in\mathcal{N}} u_i(r_i(\mathbf{p})),$$

- We will focus on the α-fair utility function:

$$u_i(x) = \begin{cases} (1-\alpha)^{-1}\, x^{1-\alpha}, & \text{if } \alpha \in (0,1)\bigcup(1,\infty), \\ \log x, & \text{if } \alpha = 1. \end{cases}$$

  - $\alpha \to 0$: system throughput maximization
  - $\alpha = 1$: proportional fair allocation
  - $\alpha \to \infty$: max-min fairness

Technical Challenges

- Non-convexity: $\alpha \in (0,1)$ is an open problem
- No centralized controller $\Rightarrow$ need to solve in a distributed fashion
- Contention-based MAC $\Rightarrow$ want no message passing

Technical Challenges

- Non-convexity: $\alpha \in (0,1)$ is an open problem
- No centralized controller $\Rightarrow$ need to solve in a distributed fashion
- Contention-based MAC $\Rightarrow$ want no message passing

We will address these challenges

Distributed Algorithm

Key idea: Localize the Global Optimization Problem

- Intuition: each user optimizes the total network utility $\Rightarrow$ solve NUM
- Challenge: what will be the information needed?

Local Optimization Problem

User i's Local NUM Problem

$$\max_{p_i \in \mathcal{P}_i} \sum_{j \in \mathcal{N}} u_j(r_j(p_i, \mathbf{p}_{-i})),$$

- Objective: total network utility
- Variable: user i's transmission probability p_i
- Parameter: other users' transmission probabilities

$$\mathbf{p}_{-i} = (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_N)$$

Optimal Solution of Local Optimization

$$p_i^*(\mathbf{p}_{-i}) = f_i(\mathbf{p}_{-i}) = \left[1/\left(1 + \sqrt[\alpha]{v_i(\mathbf{p}_{-i})}\right)\right]_{P_i^{\min}}^{P_i^{\max}},$$

Optimal Solution of Local Optimization

$$p_i^*(\mathbf{p}_{-i}) = f_i(\mathbf{p}_{-i}) = \left[1 / \left(1 + \sqrt[\alpha]{v_i(\mathbf{p}_{-i})} \right) \right]_{P_i^{\min}}^{P_i^{\max}},$$

- $v_i(\mathbf{p}_{-i}) = {\gamma_i}^{\alpha-1} \sum_{j \in \mathcal{N} \setminus \{i\}} m_j$

Optimal Solution of Local Optimization

$$p_i^*(\mathbf{p}_{-i}) = f_i(\mathbf{p}_{-i}) = \left[1/\left(1+\sqrt[\alpha]{v_i(\mathbf{p}_{-i})}\right)\right]_{P_i^{\min}}^{P_i^{\max}},$$

- $v_i(\mathbf{p}_{-i}) = {\gamma_i}^{\alpha-1}\sum_{j\in\mathcal{N}\setminus\{i\}} m_j$
- $m_j = (1/\gamma_j)^{\alpha-1}(1/p_j-1)^{\alpha-1}, \quad \forall j\in\mathcal{N}$

Optimal Solution of Local Optimization

$$p_i^*(\mathbf{p}_{-i}) = f_i(\mathbf{p}_{-i}) = \left[1/\left(1 + \sqrt[\alpha]{v_i(\mathbf{p}_{-i})}\right)\right]_{P_i^{\min}}^{P_i^{\max}},$$

- $v_i(\mathbf{p}_{-i}) = {\gamma_i}^{\alpha-1} \sum_{j \in \mathcal{N} \setminus \{i\}} m_j$
- $m_j = (1/\gamma_j)^{\alpha-1} (1/p_j - 1)^{\alpha-1}, \quad \forall j \in \mathcal{N}$
- If each j announces the message m_j, then user i can calculate $p_i^*(\mathbf{p}_{-i})$.

Local Algorithm for User i

Initialize p_i and $\mathbf{m} = (m_1, \cdots, m_N)$.
repeat
    Transmit with probability p_i.
    At a randomly chosen time, Update

$$p_i = \left[1 \Big/ \left(1 + \sqrt[\alpha]{{\gamma_i}^{\alpha-1} \sum_{j \in \mathcal{N} \setminus \{i\}} m_j} \right) \right]_{P_i^{\min}}^{P_i^{\max}}.$$

    At a random chosen time, update and broadcast

$$m_i = (1/\gamma_i)^{\alpha-1} (1/p_i - 1)^{\alpha-1}.$$

until the user decides to leave the network.

Algorithm Properties

Theorem

Under proper technical conditions and for any α-fair utility function:

1. **Uniqueness**: the algorithm has a unique fixed point.
2. **Optimality**: it is also the unique global optimal solution of NUM problem.
3. **Convergence**: the algorithm globally and asynchronously converges.
4. **Robustness**: convergence is robust to any bounded message delay/loss.

Key Maths

1. **Uniqueness**: contraction mapping or monotonic mapping

$$\mathbf{f}(\mathbf{p}) = (f_i(\mathbf{p}_i), \forall i)$$

2. **Optimality**: fixed point set of algorithm = KKT point set of NUM
3. **Convergence** & **Robustness**:
   - Synchronous convergence
   - Box condition

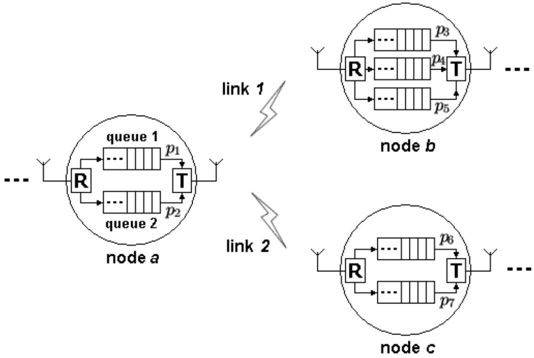
Key Maths

1. **Uniqueness**: contraction mapping or monotonic mapping

$$\mathbf{f}(\mathbf{p}) = (f_i(\mathbf{p}_i), \forall i)$$

2. **Optimality**: fixed point set of algorithm = KKT point set of NUM
3. **Convergence** & **Robustness**:
   - Synchronous convergence
   - Box condition
4. No convexity is required
   - The proposed algorithm works with enough contention level

Extensions

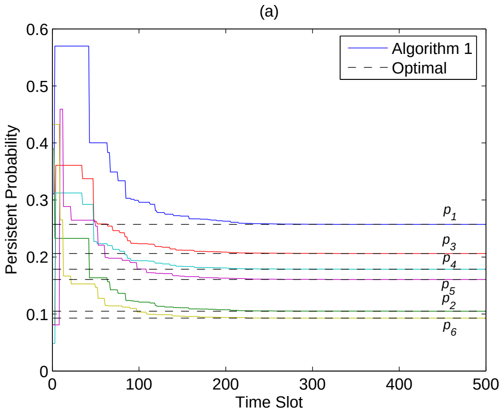


- Can also be extended to general interference case
  - A node can have multiple outgoing links
  - A link may only interfere with a subset of other links
- All previous results go through.

Performance Evaluation

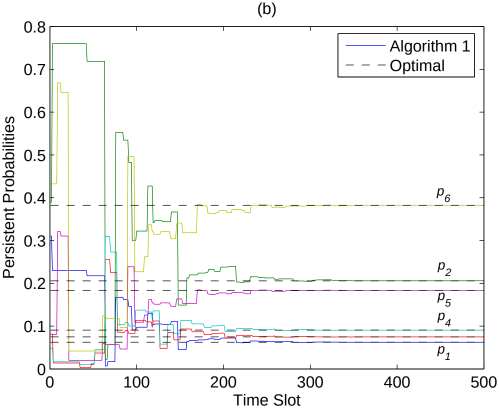
Convergence and Optimality

- 3 nodes and 6 links; $\alpha = 2$ (convex NUM)

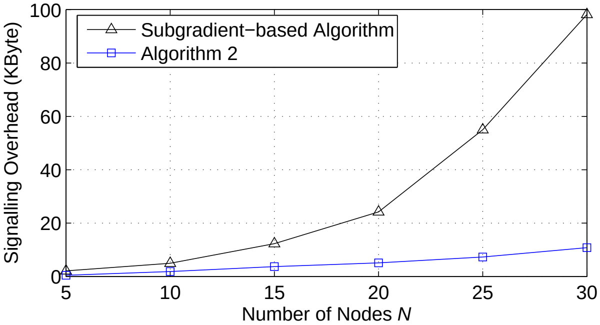


Convergence and Optimality

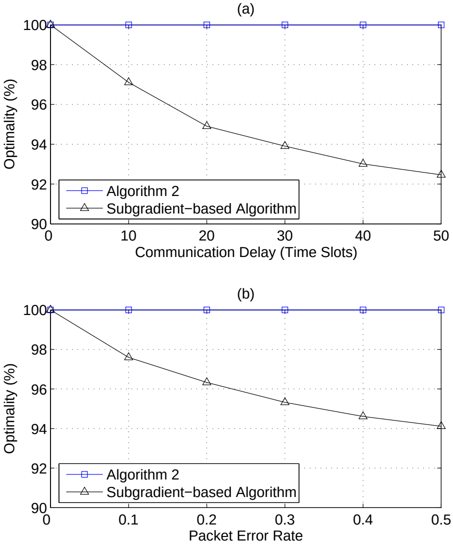
- 3 nodes and 6 links; $\alpha = 0.6$ (non-convex NUM)



Signalling Overhead



Impact of Delay and Message Loss



Eliminating Message Passing

What Messages Tell Us

- Recall message

$$m_j = (1/\gamma_j)^{\alpha-1} (1/p_j - 1)^{\alpha-1}, \quad \forall j \in \mathcal{N}$$

What Messages Tell Us

- Recall message

$$m_j = (1/\gamma_j)^{\alpha-1} (1/p_j - 1)^{\alpha-1}, \quad \forall j \in \mathcal{N}$$

- Maximum data rate γ_j: need to announce once
- Contention probability p_i: can we learn by local observation?

What Can We Observe?

- User i can observe four types of events:
  - **Idle**: no one transmits:

$$p_i^{\text{idle}} = \prod_{j \in \mathcal{N}} (1 - p_j)$$

  - **Busy**: user i does not transmit, but at least one other user does

$$p_i^{\text{busy}} = (1 - p_i) - p_i^{\text{idle}}$$

  - **Success**: user i's transmission succeeds

$$p_i^{\text{succ}} = \frac{p_i}{1 - p_i} p_i^{\text{idle}} (1 - p_{i,i}^{\text{err}})$$

  - **Failure**: user i's transmission fails

$$p_i^{\text{fail}} = p_i - \left(\frac{p_i}{1 - p_i} p_i^{\text{idle}} \right) (1 - p_{i,i}^{\text{err}})$$

Do We Have Enough Information to Calculate p_j?

Do We Have Enough Information to Calculate p_j?

- No.

Do We Have Enough Information to Calculate p_j?

- No.
- Why: we need more user specific information.

Do We Have Enough Information to Calculate p_j?

- No.
- Why: we need more user specific information.
- How: differentiate the sender in the **busy** slot.
  - Check the MAC address in the packet header

How to Estimate the Probability p_j

- User i can decode user j's transmission in a busy slot

$$p_{i,j}^{\text{decd}} = p_j\left(\prod_{l\in\mathcal{N}\setminus\{j\}}(1-p_l)\right)(1-p_{i,j}^{\text{err}})$$

How to Estimate the Probability p_j

- User i can decode user j's transmission in a busy slot

$$p_{i,j}^{\text{decd}} = p_j(\prod_{l \in \mathcal{N} \setminus \{j\}} (1 - p_l))(1 - p_{i,j}^{\text{err}})$$

  - Still not locally observable

How to Estimate the Probability p_j

- User i can decode user j's transmission in a busy slot

$$p_{i,j}^{\text{decd}} = p_j \left(\prod_{l \in \mathcal{N} \setminus \{j\}} (1 - p_l)\right)(1 - p_{i,j}^{\text{err}})$$

  - Still not locally observable

- Denote the number of time slots between adjacent such events: $n_{i,j}^{\text{decd}}$

How to Estimate the Probability p_j

- User i can decode user j's transmission in a busy slot

$$p_{i,j}^{\mathrm{decd}} = p_j(\prod_{l \in \mathcal{N} \setminus \{j\}} (1 - p_l))(1 - p_{i,j}^{\mathrm{err}})$$

  - Still not locally observable

- Denote the number of time slots between adjacent such events: $n_{i,j}^{\mathrm{decd}}$
  - Locally observable

How to Estimate the Probability p_j

- User i can decode user j's transmission in a busy slot

$$p_{i,j}^{\text{decd}} = p_j\left(\prod_{l \in \mathcal{N} \setminus \{j\}} (1 - p_l)\right)(1 - p_{i,j}^{\text{err}})$$

  - Still not locally observable

- Denote the number of time slots between adjacent such events: $n_{i,j}^{\text{decd}}$
  - Locally observable

- Estimate the decoding probability

$$p_{i,j}^{\text{decd}} = 1/(1 + \bar{n}_{i,j}^{\text{decd}})$$

How to Estimate the Probability p_j

- Define the number of time slots between adjacent idle time slots: $n_{i,j}^{\text{idle}}$

How to Estimate the Probability p_j

- Define the number of time slots between adjacent idle time slots: $n_{i,j}^{\text{idle}}$
- Estimation the idle probability

$$p_i^{\text{idle}} = 1/(1 + \bar{n}_i^{\text{idle}})$$

How to Estimate the Probability p_j

- Define the number of time slots between adjacent idle time slots: $n_{i,j}^{\text{idle}}$
- Estimation the idle probability

$$p_i^{\text{idle}} = 1/(1 + \bar{n}_i^{\text{idle}})$$

- Finally, we estimate user j's transmission probability p_j

$$1/p_j - 1 = \left((1 + \bar{n}_{i,j}^{\text{decd}})/(1 + \bar{n}_i^{\text{idle}})\right) \, (1 - \tilde{p}_{i,j}^{\text{err}}).$$

How to Estimate the Probability p_j

- Define the number of time slots between adjacent idle time slots: $n_{i,j}^{\text{idle}}$
- Estimation the idle probability

$$p_i^{\text{idle}} = 1/(1 + \bar{n}_i^{\text{idle}})$$

- Finally, we estimate user j's transmission probability p_j

$$1/p_j - 1 = \left((1 + \bar{n}_{i,j}^{\text{decd}})/(1 + \bar{n}_i^{\text{idle}})\right)\ (1 - \tilde{p}_{i,j}^{\text{err}}).$$

  - $\tilde{p}_{i,j}^{\text{err}}$ is typically very small due to low rate of header info.

How to Estimate the Message m_j

- The message can be estimate as

$$m_j^i(t) = (1/\gamma_j)^{\alpha-1} \left((1 + \bar{n}_{i,j}^{\text{decd}}(t))/(1 + \bar{n}_i^{\text{idle}}(t)) \right)^{\alpha-1}$$

  - In general, the estimation may not be accurate
  - Inaccuracy in $\bar{n}_{i,j}^{\text{decd}}(t)$ and $\bar{n}_{i,j}^{\text{idle}}(t)$
  - No knowledge of $\tilde{p}_{i,j}^{\text{err}}$

How to Estimate the Message m_j

- The message can be estimate as

$$m_j^i(t) = (1/\gamma_j)^{\alpha-1} \left((1 + \bar{n}_{i,j}^{\text{decd}}(t))/(1 + \bar{n}_i^{\text{idle}}(t)) \right)^{\alpha-1}$$

  - In general, the estimation may not be accurate
  - Inaccuracy in $\bar{n}_{i,j}^{\text{decd}}(t)$ and $\bar{n}_{i,j}^{\text{idle}}(t)$
  - No knowledge of $\tilde{p}_{i,j}^{\text{err}}$

- If $m_j(t)$ is the accurate value of the message, then

$$m_j^i(t) = \beta_j^i(t)\, m_j(t)$$

  - Under-estimate ($\beta_j^i(t) < 1$) or over-estimate ($\beta_j^i(t) > 1$)
  - With proper estimation stepsizes, $\beta_j^i(t)$s eventually converge (but may not equal to 1)

Modified Distributed Algorithm

- Replace message passing with local estimation
- Properties:

Theorem

Under proper technical conditions and for any α-fair utility function:

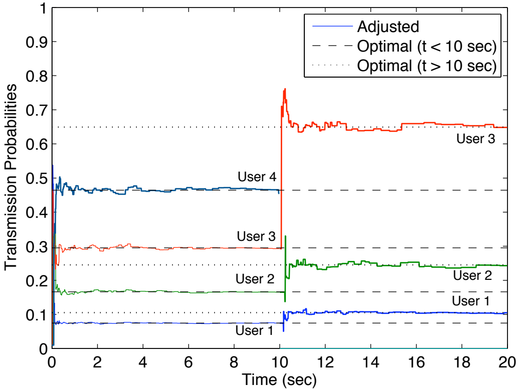
1. **Uniqueness**: the algorithm has a unique fixed point.
2. **Convergence**: the algorithm globally and asynchronously converges.
3. **Optimality**: reaches optimal solution if local estimations are asymptotically accurate.

Key Maths

- The message estimations are stochastic.
- The mapping $\mathbf{f}(\mathbf{p}) = (f_i(\mathbf{p}_i), \forall i)$ is time varying.
- What we prove: the contention probability vector $\mathbf{p}(t)$ converges to the limit of a series of stochastically changing fixed points.
  - The key difficulty is to satisfy the box condition.

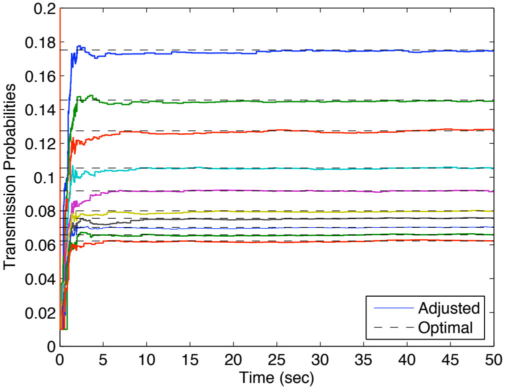
Channel and User Population Changes

- 0-10 secs: 4 users with perfect channel
- 10-20 secs: 3 users with imperfect channel



More Users

- 10 users, $\alpha = 2$, no message passing



Conclusion

- **Topic**: wireless random MAC algorithm design
- **Key ideas**:
  - Stepsize-free updates
  - Learning instead of talking
- **Algorithm**: a family of simple algorithms
- **Performance**: optimal, fast, robust, and low overhead

References

- A.H. Mohsenian-Rad, J. Huang, M. Chiang and V.W.S. Wong, "Utility-Optimal Random Access: Reduced Complexity, Fast Convergence, and Robust Performance," *IEEE Trans. Wireless Commun.*, accepted.
- A.H. Mohsenian-Rad, J. Huang, M. Chiang and V.W.S. Wong, "Utility-Optimal Random Access: Optimal Performance Without Frequent Explicit Message Passing," *IEEE Trans. Wireless Commun.*, accepted.