# Wireless Random Medium Access Control: Reverse and Forward Engineering


Jianwei Huang

Department of Information Engineering

The Chinese University of Hong Kong



Ack: R. Calderbank, M. Chiang, J. Lee, A. Mohsenian-Rad, A. Tang, V. Wong


EPFL SRI
June 2009

# The Chinese University of Hong Kong (CUHK)


# Research Overview

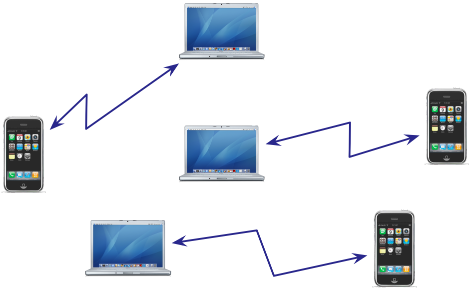
- **Wireless Communications & Networking**
  - Cognitive Radio
  - Cooperative Communications
  - OFDM/CDMA Networks
  - Wireless Multimedia
  - Wireless MAC
- **Network Management & Economics**
  - Pricing & Revenue Management
  - Service Provider Competitions
  - Network Disruption Management
  - Robust Network Optimization

# Key Methodologies

- Game Theory & Microeconomics
- Nonlinear Optimization
- Queueing & Stochastic Control

# Wireless Random MAC

# 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: Aloha, CSMA, ...
  - Many are engineering ad hoc designs

# History of Wireless Random MAC

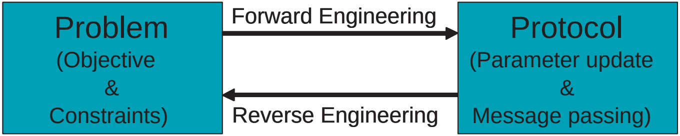
- Studied for 30 years
- Simplicity and practicality
- Many variations
  - Some achieved great success: Aloha, CSMA, ...
  - Many are engineering ad hoc designs
- What we will talk about
  - **Reverse Engineering**: understand the math behind existing protocols
  - **Forward Engineering**: design better protocols

# Part I: Reverse Engineering MAC

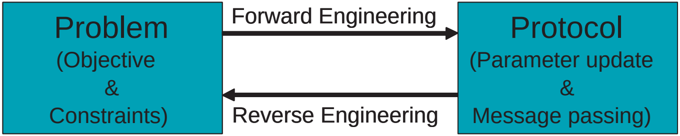
# Reverse Engineering

- Given the solution, what is the problem?
- Know what works, what doesn't, why it works, how to improve?
- Provide the missing piece (on MAC) for existing layers 2-4 protocols on rigorous mathematical foundation

# Reverse Engineering



# Reverse Engineering




- Related works:
  - Layer 4: TCP/AQM [Kelly-Maulloo-Tan98, Low03, Kunniyur-Srikant03, ...] NUM
  - Layer 3: BGP [Griffin-Shepherd-Wilfong02] SPP
  - Layer 2: MAC (contention avoidance in random access)

# TCP/AQM

- Network Utility Maximization (NUM) problem
  - Utility of each user depends on its own data rate
  - Adequate feedback from the network

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s:l\in L(s)} x_s \le c_l, \ \ \forall l, \\ & \mathbf{x}^{min} \preceq x \preceq \mathbf{x}^{max}. \end{array}$$

# TCP/AQM

- Network Utility Maximization (NUM) problem
  - Utility of each user depends on its own data rate
  - Adequate feedback from the network

$$
\begin{array}{ll}
\text{maximize} & \sum_s U_s(x_s) \\
\text{subject to} & \sum_{s:l\in L(s)} x_s \le c_l, \;\; \forall l, \\
 & \mathbf{x}^{min} \preceq x \preceq \mathbf{x}^{max}.
\end{array}
$$

- Reverse engineering provides

# TCP/AQM

- Network Utility Maximization (NUM) problem
  - Utility of each user depends on its own data rate
  - Adequate feedback from the network

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s:l\in L(s)} x_s \le c_l, \ \ \forall l, \\ & \mathbf{x}^{min} \preceq x \preceq \mathbf{x}^{max}. \end{array}$$

- Reverse engineering provides
  - Better understanding: existence, uniqueness, optimality and stability, counter-intuitive behaviors
  - Systematic design: scalable price signal, control laws with better stability properties

# MAC Reverse Engineering

- Utility depends on
  - Its own transmission
  - Other links transmissions: not locally controllable
- Inadequate binary feedback from the network

# MAC Reverse Engineering

- Utility depends on
  - Its own transmission
  - Other links transmissions: not locally controllable
- Inadequate binary feedback from the network

Reverse engineer to non-cooperative game

# MAC Reverse Engineering

- Utility depends on
  - Its own transmission
  - Other links transmissions: not locally controllable
- Inadequate binary feedback from the network

Reverse engineer to non-cooperative game

- Questions:

# MAC Reverse Engineering

- Utility depends on
  - Its own transmission
  - Other links transmissions: not locally controllable
- Inadequate binary feedback from the network

Reverse engineer to non-cooperative game

- Questions:
  - What kind of utility functions do users have?

# MAC Reverse Engineering

- Utility depends on
  - Its own transmission
  - Other links transmissions: not locally controllable
- Inadequate binary feedback from the network

Reverse engineer to non-cooperative game

- Questions:
  - What kind of utility functions do users have?
  - What does the MAC protocol do for the game?

# MAC Reverse Engineering

- Utility depends on
  - Its own transmission
  - Other links transmissions: not locally controllable
- Inadequate binary feedback from the network

Reverse engineer to non-cooperative game

- Questions:
  - What kind of utility functions do users have?
  - What does the MAC protocol do for the game?
  - Nash Equilibrium (NE): existence? uniqueness? stablility?

# Related Work on MAC Games

- MacKenzie, Wicker 2003
- Jin, Kesidis 2004
- Altman et. al. 2005
- Yuen, Marbach 2005
- Wang, Krunz, Younis 2006

- This is different: Reverse engineering
- Discover, not impose, utility and game

# Persistence Probabilistic Model of Protocol

- Protocol parameters:
  - Transmission probability range: $p_l \in \left[p_l^{\text{min}}, p_l^{\text{max}}\right]$
  - Backoff multiplier $\beta \in (0, 1)$

# Persistence Probabilistic Model of Protocol

- Protocol parameters:
  - Transmission probability range: $p_l \in \left[p_l^{\min}, p_l^{\max}\right]$
  - Backoff multiplier $\beta \in (0, 1)$

- Protocol description: link $l$ transmits with a probability $p_l$
  - If success, set $p_l = p_l^{\max}$ for the next transmission
  - If failure, set $p_l = \max\{p_l^{\min}, \beta_l p_l\}$, where $0 < \beta_l < 1$

# Persistence Probability Update

- $\mathbf{1}_a$: indicator function of event $a$
- $T_l(t)$: link $l$ transmits at time slot $t$

$$\text{Prob}\{T_l(t) = 1|\mathbf{p}(t)\} = p_l(t)$$

- $C_l(t)$: a link collides with link $l$ given it transmits at time slot $t$

$$\text{Prob}\{C_l(t) = 1|\mathbf{p}(t)\} = 1 - \prod_{n \in L_{to}(l)} (1 - p_n(t))$$

  - $L_{to}(l)$: set of links causing interferences to link $l$

# Persistence Probability Update

- $\mathbf{1}_a$: indicator function of event $a$
- $T_l(t)$: link $l$ transmits at time slot $t$

$$\mathrm{Prob}\{T_l(t) = 1|\mathbf{p}(t)\} = p_l(t)$$

- $C_l(t)$: a link collides with link $l$ given it transmits at time slot $t$

$$\mathrm{Prob}\{C_l(t) = 1|\mathbf{p}(t)\} = 1 - \prod_{n \in L_{to}(l)} (1 - p_n(t))$$

  - $L_{to}(l)$: set of links causing interferences to link $l$

- Probability update: $p_l(t+1) = \max\left\{p_l^{min}, p_l'(t+1)\right\}$, with

$$p_l'(t+1) = p_l^{max}\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=0\}} + \beta_l p_l(t)\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=1\}} + p_l(t)\mathbf{1}_{\{T_l(t)=0\}}$$

# Deterministic Approximation of Average Behavior

- Probabilities keep changing.
- How about average behavior?

# Deterministic Approximation of Average Behavior

- Probabilities keep changing.
- How about average behavior?
- A deterministic updating rule approximating the average behavior

$$p_l(t+1) = \max\left\{p_l^{min}, \bar{p}_l'(t+1)\right\},$$

with

$$\bar{p}_l'(t+1) = p_l^{max} p_l(t) \prod_{n \in L_{to}(l)} (1 - p_n(t)) + \beta_l p_l(t) p_l(t) \left(1 - \prod_{n \in L_{to}(l)} (1 - p_n(t))\right) + p_l(t)(1 - p_l(t))$$

  - Each user $l$ tries to maximize its utility $U_l$ given other links' strategies

# MAC Game

- Define MAC game as

$$[\mathcal{E}, \{A_l\}_{l\in\mathcal{E}}, \{U_l\}_{l\in\mathcal{E}}]$$

  - $\mathcal{E}$: set of players (links)
  - $A_l = \{p_l | p_l^{\min} \le p_l \le p_l^{\max}\}$: action set of link $l$
  - $U_l$: utility function of link $l$

# MAC Game

- We know
  - $S(\mathbf{p}) = p_l \prod_{n \in L_{to}(l)} (1 - p_n)$: probability of transmission success
  - $F(\mathbf{p}) = p_l (1 - \prod_{n \in L_{to}(l)} (1 - p_n))$: probability of transmission failure

# MAC Game

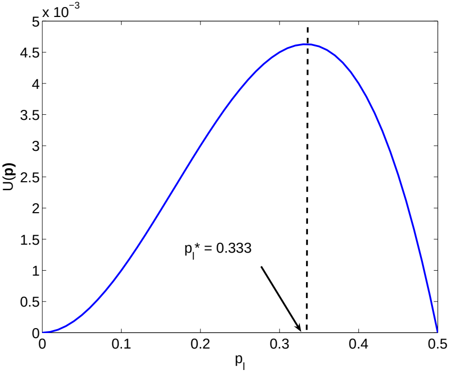
- We know
  - $S(\mathbf{p}) = p_l \prod_{n \in L_{to}(l)} (1 - p_n)$: probability of transmission success
  - $F(\mathbf{p}) = p_l (1 - \prod_{n \in L_{to}(l)} (1 - p_n))$: probability of transmission failure
- Theorem: Utility function turns out to be expected net reward:

$$U_l(\mathbf{p}) = R(p_l) S(\mathbf{p}) - C(p_l) F(\mathbf{p})$$

where

  - $R(p_l) = p_l (\frac{1}{2} p_l^{max} - \frac{1}{3} p_l)$: reward for transmission success
  - $C(p_l) = \frac{1}{3}(1 - \beta_l) p_l^2$: cost for transmission failure

# Utility Function




Dependence of a utility function on its own persistence probability ($\beta_l = 0.5$, $p_l^{max} = 0.5$, and $\prod_{n \in L_{to}(l)}(1 - p_n) = 0.5$)

# MAC protocol as a stochastic subgradient algorithm

- Is it a gradient-based maximization of $U_l(\mathbf{p})$ over $p_l$?
  - No, that requires explicit message passing among links

# MAC protocol as a stochastic subgradient algorithm

- Is it a gradient-based maximization of $U_l(\mathbf{p})$ over $p_l$?
  - No, that requires explicit message passing among links
- MAC maximizes $U_l$ using stochastic subgradient ascent method (using only local information on success and collision):

$$p_l(t+1) = \max\{p_l^{min}, p_l(t) + v_l(t)\}$$

where

$$\mathrm{E}\{v_l(t)|\mathbf{p}(t)\} = \frac{\partial U_l(\mathbf{p})}{\partial p_l}|_{\mathbf{p}=\mathbf{p}(t)}$$

# Existence and Uniqueness of Nash Equilibrium

- Theorem: there always exits a Nash equilibrium in the MAC game, which can be characterized by

$$p_l^* = \frac{p_l^{max} \prod_{n \in L_{to}(l)} (1 - p_n^*)}{1 - \beta_l (1 - \prod_{n \in L_{to}(l)} (1 - p_n^*))}, \ \forall l$$

  - Proof: Fixed point theorem in the compact strategy interval.

# Existence and Uniqueness of Nash Equilibrium

- Theorem: there always exits a Nash equilibrium in the MAC game, which can be characterized by

$$p_l^* = \frac{p_l^{max} \prod_{n \in L_{to}(l)} (1 - p_n^*)}{1 - \beta_l (1 - \prod_{n \in L_{to}(l)} (1 - p_n^*))}, \ \forall l$$

  - Proof: Fixed point theorem in the compact strategy interval.

- The Nash equilibrium may not be unique in general.

# Convergence of Nash Equilibrium

- Define the best response function as

$$p_l^*(t+1) = \arg \max_{p_l^{\min} \leq p_l \leq p_l^{\max}} U_l(p_l, p_{-l}^*(t))$$

- Theorem: connsider best response updates with $\mathbf{p}^*(0) = \mathbf{p}_{\min}$, then,
  - $\mathbf{p}^*(2t+1) \rightarrow \mathbf{p}'$ and $\mathbf{p}^*(2t) \rightarrow \mathbf{p}''$ as $t \rightarrow \infty$.
  - If $\mathbf{p}' = \mathbf{p}''$, then $\mathbf{p}'$ is a Nash equilibrium.

# Convergence of Nash Equilibrium

- Define the best response function as

$$p_l^*(t+1) = \arg \max_{p_l^{\min} \le p_l \le p_l^{\max}} U_l(p_l, p_{-l}^*(t))$$

- Theorem: connsider best response updates with $\mathbf{p}^*(0) = \mathbf{p}_{\min}$, then,
  - $\mathbf{p}^*(2t+1) \to \mathbf{p}'$ and $\mathbf{p}^*(2t) \to \mathbf{p}''$ as $t \to \infty$.
  - If $\mathbf{p}' = \mathbf{p}''$, then $\mathbf{p}'$ is a Nash equilibrium.
  - Proof: S-modular theory.

# Convergence of Nash Equilibrium

- Assume all links have the same $p^{\max} < 1$ and $p^{\min} = 0$
- Theorem: define $K = \max_l |L_{to}(l)|$, then if

$$\frac{p^{max} K}{4\beta(1 - p^{max})} < 1$$

  - The Nash equilibrium is unique
  - The best response iteration globally converges to NE

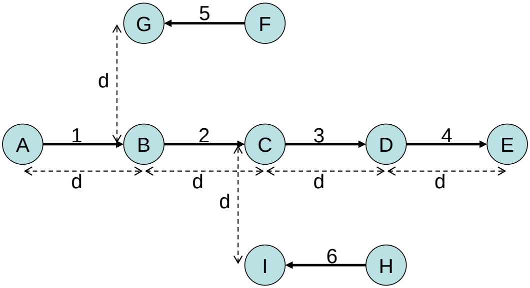
# Convergence of Nash Equilibrium

- Assume all links have the same $p^{\max} < 1$ and $p^{\min} = 0$
- Theorem: define $K = \max_l |L_{to}(l)|$, then if

$$\frac{p^{max} K}{4\beta(1 - p^{max})} < 1$$

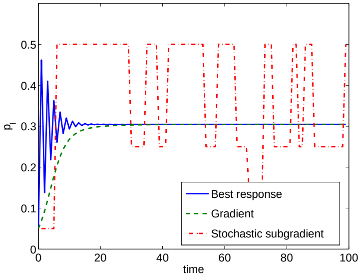
  - The Nash equilibrium is unique
  - The best response iteration globally converges to NE
  - Proof: Properly bounding the matrix norm of the Jacobian. Show it is a contraction mapping.
- Easier to converge: small $K$, small $p^{\max}$, large $\beta$

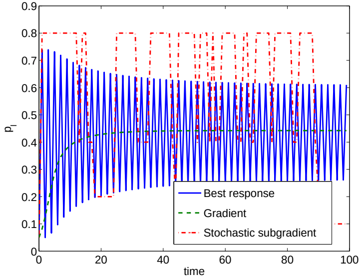
# Network Topology




A network with six links

# Convergence




$p_l^{max} = 0.5$ $\qquad\qquad$ $p_l^{max} = 0.8$

Comparison of trajectories of $p_l(t)$ in the network

# Summary

- **Topic**: Reverse engineering MAC as a non-cooperative game
- **Discovery**: Utility function as expected net reward
- **Properties**:
  - NE always exists.
  - Unique & stable: if users polite enough and backoff smooth enough.
  - Sequential subgradient = Best response

# Part II: Forward Engineering MAC

# Forward Engineering

- How to design a better algorithm

# Forward Engineering

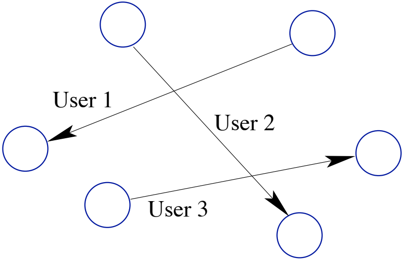
- How to design a better algorithm
- Simple, Optimal, Fast, and Robust

# Forward Engineering

- How to design a better algorithm
- Simple, Optimal, Fast, and Robust
- Overcome performance bottlenecks of many previous algorithms

# Our focus: Aloha

# (A Simple) Network Model




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

# Previous Work

- Lee, Chiang, Canderbank 2007
- Wang, Kar 2006
- Chen, Low, Doyle 2005
- Gupta, Stolyar 2006

# Previous Work

- Lee, Chiang, Canderbank 2007
- Wang, Kar 2006
- Chen, Low, Doyle 2005
- Gupta, Stolyar 2006

- Several performance bottlenecks

# Technical Challenges / Performance Bottlenecks

- Non-convexity: $\alpha \in (0, 1)$ is an open problem even centrally
- No centralized controller $\Rightarrow$ need to be distributed and asynchronous
- Wireless lossy channels $\Rightarrow$ messages may get delayed and dropped
- Channels can be time varying $\Rightarrow$ demand fast convergence

# Technical Challenges / Performance Bottlenecks

- Non-convexity: $\alpha \in (0, 1)$ is an open problem even centrally
- No centralized controller $\Rightarrow$ need to be distributed and asynchronous
- Wireless lossy channels $\Rightarrow$ messages may get delayed and dropped
- Channels can be time varying $\Rightarrow$ demand fast convergence

**We will address these challenges**

# 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 $j$ broadcasts the message $m_j \Rightarrow$ user $i$ can calculate $p_i^*(\mathbf{p}_{-i})$.

# Local Algorithm for User $i$

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

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

5: At a randomly chosen time, update and broadcast

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

6: **until** the user decides to leave the network.

# Algorithm Properties

**Theorem**

Under proper technical conditions and for any $\alpha$-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.

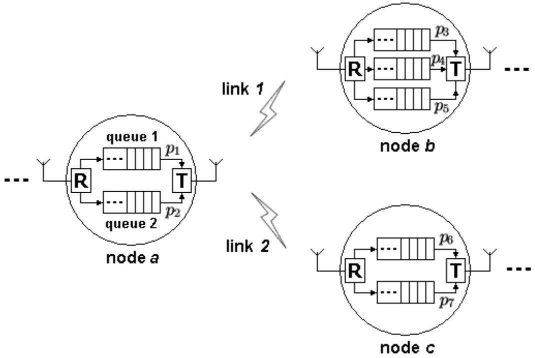
# Key Maths

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
4. No convexity is required
   - The proposed algorithm works with enough contention level

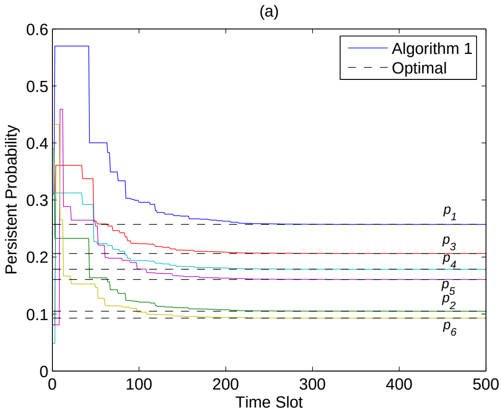
# Extensions




- 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.

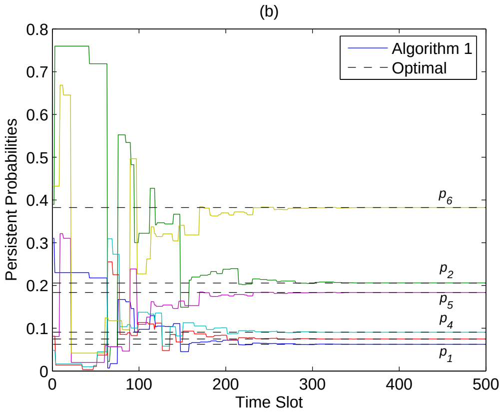
# Convergence and Optimality

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

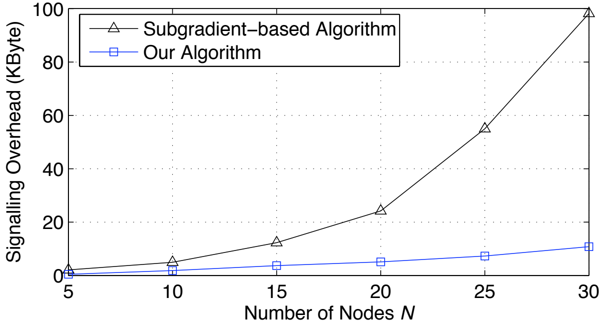


# Convergence and Optimality

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

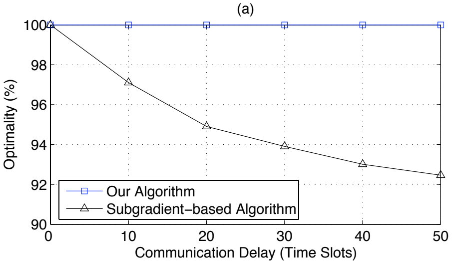


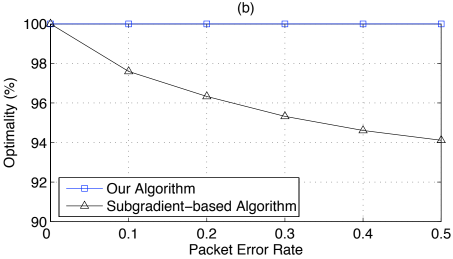
# Signalling Overhead




Subgradient-based algorithm: J. Lee, M. Chiang, and R. Calderbank, "Utility-optimal random-access control," IEEE Trans. Wireless Comm., 2007.

# Impact of Delay and Message Loss



# Summary

- **Topic**: Forward engineering MAC as an optimization problem
- **Algorithm**: distributed asynchronous updates with limited message passing
- **Properties**: simple, optimal, fast, and robust
- **Extension**: the same algorithm can work without any explicit message passing

# Related Journal Publications

- **Reverse Engineering**: J.-W. Lee, A. Tang, J. Huang, M. Chiang and A. R. Calderbank, "Reverse Engineering MAC: A Non-Cooperative Game Model," *IEEE Journal on Selected Areas in Communications*, Aug. 2007
- **Forward Engineering I**: 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 Transactions on Wireless Communications*, Feb. 2009
- **Forward Engineering II**: 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 Transactions on Wireless Communications*, March 2009

# Contact


www.ie.cuhk.edu.hk/∼jwhuang

jwhuang@ie.cuhk.edu.hk