Reverse Engineering MAC: A Non-Cooperative Game Model

Jianwei Huang

Information Engineering
The Chinese University of Hong Kong

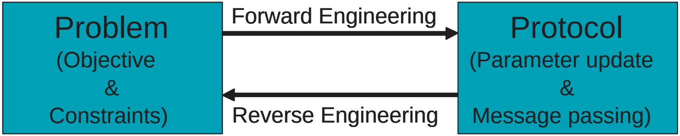
Joint work with J.-W. Lee, A. Tang, M. Chiang and A. R. Canderbank

Summary

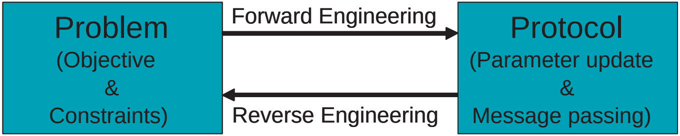
Reverse engineering:
Given the solution, what is the problem?
Then, know what works, what doesn't, why it works, how to improve

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) [This Paper]

TCP/AQM

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

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

TCP/AQM

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

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

- 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

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

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

MAC Reverse Engineering

- Utility depends on its own transmission and other links transmissions through collisions, which can not be completely controlled by the link itself.

MAC Reverse Engineering

- Utility depends on its own transmission and other links transmissions through collisions, which can not be completely controlled by the link itself.
- Inadequate feedback from the network

MAC Reverse Engineering

- Utility depends on its own transmission and other links transmissions through collisions, which can not be completely controlled by the link itself.
- Inadequate feedback from the network

Reverse engineer to non-cooperative game

MAC Reverse Engineering

- Utility depends on its own transmission and other links transmissions through collisions, which can not be completely controlled by the link itself.
- Inadequate feedback from the network

Reverse engineer to non-cooperative game

- Questions:

MAC Reverse Engineering

- Utility depends on its own transmission and other links transmissions through collisions, which can not be completely controlled by the link itself.
- Inadequate 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 and other links transmissions through collisions, which can not be completely controlled by the link itself.
- Inadequate 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 and other links transmissions through collisions, which can not be completely controlled by the link itself.
- Inadequate 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?
  - Does the Nash Equilibrium (NE) exist? If so, is the NE unique and stable?

Different Work

Game to MAC:

- 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:
  - Politeness: p_l^{max}
  - Backoff multiplier $\beta \in (0, 1)$
  - p_l^{min}

Persistence Probabilistic Model of Protocol

- Protocol parameters:
  - Politeness: $p_l^{\max}$
  - Backoff multiplier $\beta \in (0, 1)$
  - $p_l^{\min}$

- 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 Probabilistic Model of Protocol

- Protocol parameters:
  - Politeness: $p_l^{\max}$
  - Backoff multiplier $\beta \in (0,1)$
  - $p_l^{\min}$
- 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 $0 < \beta_l < 1$
- Define $L_{to}(l)$ as set of links causing interferences to link l

Persistence Probability Update

$$
\begin{aligned}
p_l(t+1) \;=\; & \max\{p_l^{min}, p_l^{max}\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=0\}} \\
& +\beta_l p_l(t)\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=1\}} \\
& +p_l(t)\mathbf{1}_{\{T_l(t)=0\}}\}
\end{aligned}
$$

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

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

- $C_l(t)$: event that there is a collision to link l's transmission at time slot t

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

MAC Game

- A deterministic updating rule approximating the average behavior

$$
\begin{aligned}
p_l(t+1) &= \max\{p_l^{min}, p_l^{max} p_l(t) \prod_{n \in L_{to}(l)} (1 - p_n(t)) \\
&\quad + \beta_l p_l(t) p_l(t) \left(1 - \prod_{n \in L_{to}(l)} (1 - p_n(t)) \right) \\
&\quad + p_l(t)(1 - p_l(t))\},
\end{aligned}
$$

- The strategy such that each link tries to maximize its utility U_l based on strategies of other links

MAC Game

- A deterministic updating rule approximating the average behavior

$$
\begin{aligned}
p_l(t+1) &= \max\{p_l^{min}, p_l^{max} p_l(t) \prod_{n \in L_{to}(l)} (1 - p_n(t)) \\
&\quad + \beta_l p_l(t) p_l(t) \left(1 - \prod_{n \in L_{to}(l)} (1 - p_n(t))\right) \\
&\quad + p_l(t)(1 - p_l(t))\},
\end{aligned}
$$

- The strategy such that each link tries to maximize its utility U_l based on strategies of other links
- Define MAC game as

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

  - E: set of players (links)
  - $A_l = \{p_l | p_l^{\min} \leq p_l \leq p_l^{\max}\}$: 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

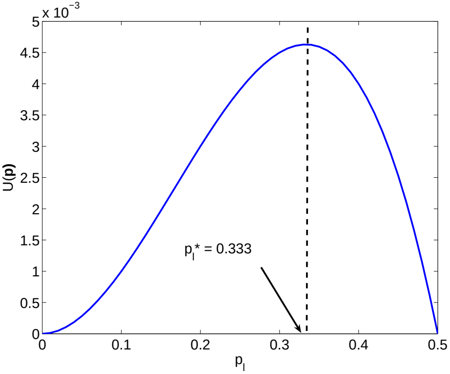
MAC Game

- 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



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 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.

Uniqueness of Nash Equilibrium

- The Nash equilibrium may not be unique in general.
- Example
  - Two links interfering with each other
  - $p_1^{\max} = p_2^{\max} = p_{\max} = 1$
  - Infinite many Nash equilibria

$$\max\{p^{min}, \frac{1-p^{max}}{1-\beta p^{max}}\} \leq p_1^* \leq \min\{1, \frac{1-p^{min}}{1-\beta p^{min}}\}$$

$$p_2^* = \frac{1-p_1^*}{1-\beta p_1^*}$$

Uniqueness and 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.

Uniqueness and 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.

Uniqueness and 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 the unique equilibrium

Uniqueness and Convergence of Nash Equilibrium

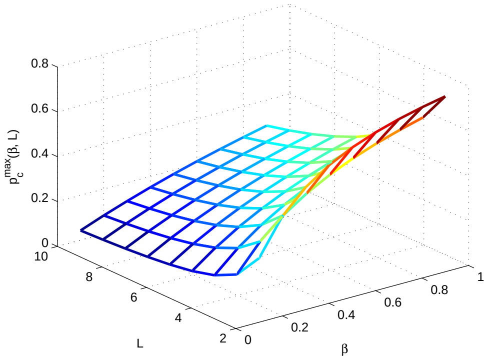
- Assume all links have the same $p^{\max} < 1$ and $p^{\min} = 0$
- Theorem: define $K = \max_l |L_{to}(l)|$, then if

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

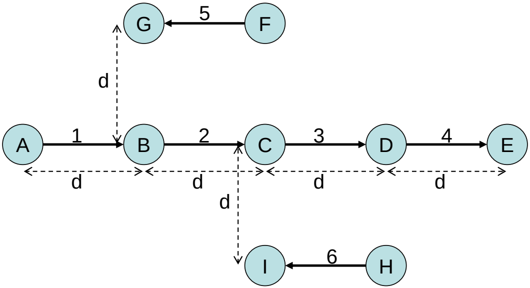
  - The Nash equilibrium is unique
  - The best response iteration globally converges to the unique equilibrium
  - Proof: Properly bounding the matrix norm of the Jacobian. Show it is a contraction mapping.
- How polite is necessary? Critical value: $p_c^{\max}$

Uniqueness and Convergence of Nash Equilibrium

- All links interfere with each other

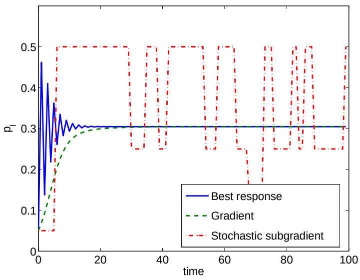


Network Topology

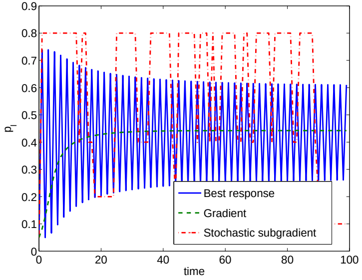


A network with Six Links

Convergence



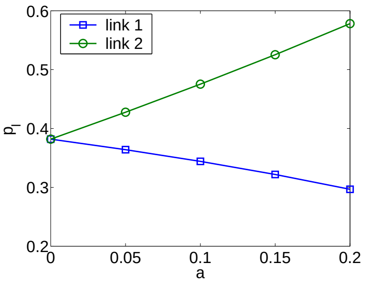
$p_l^{max} = 0.5$

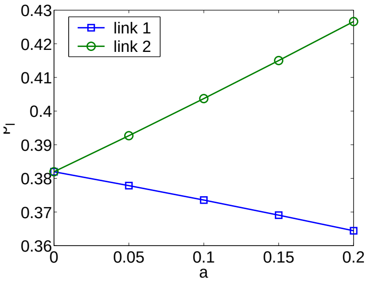


$p_l^{max} = 0.8$

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

Fairness





$$\beta_1 = \beta_2 = 0.5$$
$$p_1^{max} = 0.5, p_2^{max} = 0.5 + a$$

$$p_1^{max} = p_2^{max} = 0.5$$
$$\beta_1 = 0.5, \beta_2 = 0.5 + a$$

Conclusions

- Reverse engineering for MAC protocol
  - Reverse engineered as a non-cooperative game
  - Utility function discovered: expected net reward
  - NE always exists. It is unique and stable if the protocol is polite enough and backoff smooth enough.
  - Sequential subgradient = Best response

Conclusions

- Reverse engineering for MAC protocol
  - Reverse engineered as a non-cooperative game
  - Utility function discovered: expected net reward
  - NE always exists. It is unique and stable if the protocol is polite enough and backoff smooth enough.
  - Sequential subgradient = Best response
- Future work
  - Bounding efficiency loss, Gradient play convergence
  - From reverse engineering to forward engineering (design)
  - Union of session level stochastic and utility-optimal protocol