A Measurement-Based Algorithm to Maximize the Utility of Wireless Networks

Julien Herzen

joint work with
Adel Aziz, Ruben Merz, Seva Shneer and Patrick Thiran

September 19th, 2011
Context

Inefficient situations in wireless LANs

- **Example**, performance anomaly:

  ![Diagram](image)

  - **Intuition**: Send slightly fewer packets at 1 Mb/s, so that the flow at 11 Mb/s can send many more
• **Formalization:** Capture the *efficiency* and the *fairness* of the network using a utility function

\[ U = \sum_{i} u_i(x_i), \]

\( x_i \): throughput of flow \( i \)

**Examples**

\[ U_{\text{max}} = \sum_{i} x_i \]

\[ U_{\text{prop}} = \sum_{i} \log x_i \]
Network Stack

- Backward compatibility → runs on top of IEEE 802.11
- Congestion control → throttle each flow
- One limiter per IP source in the network
How to throttle the flows?

- Find the rate allocation $\rho$ that maximizes the utility $U$
- **Problem: We do not know the feasible rate region!**
  - hard to predict or measure

$$U = \sum \log x_i$$
Decide at the gateway

The gateway knows

- The throughput achieved by the flows: \( x \)
- The current utility of the network: \( U(x) = \sum u_i(x_i) \)
- If \( x = \rho \), then \( \rho \) belongs to the rate region

Measure and Decide loop

- Measure each flow
- Decide
- Broadcast decision
Model

At time slot $n$:

- Measured throughput: $x[n] \in \mathbb{R}^F_+$
- Rate allocation vector: $\rho[n] \in \mathbb{R}^F_+$
- Last stable rate allocation: $r[n] \in \mathbb{R}^F_+$
- Utility function: $U(x) = \sum_i u_i(x_i)$
- Level set: $L(\mu[n]) = \{x[n] : U(x[n]) = \mu[n], x[n] \in \mathbb{R}^F_+\}$
Step 1 - Start from IEEE 802.11 allocation

- $\rho[0] \leftarrow x[0]$
- Current level set: $L(U(x[0]))$
- Remember allocation $r[0] \leftarrow x[0]$
Step 2 - Enhance phase

Time step $n$:

- If $x[n-1] = \rho[n-1]$:  
  - Obtain a new target utility $\mu[n]$ by a **full size gradient ascent**
- Else:
  - Obtain a new target utility $\mu[n]$ by halving the size of the gradient ascent
- Go to Explore phase (next slide)

\[
U = \sum \log x_i
\]
Step 3 - Explore phase

- If $x[n - 1] = \rho[n - 1]$: 
  - Remember $r[n] = \rho[n - 1]$
  - Go to Enhance phase
- Else:
  - Keep target utility: $\mu[n] = \mu[n - 1]$
  - Pick $\rho[n]$ randomly in $L(\mu[n])$
  - Repeat explore phase at most $N$ times, then move to Enhance phase (and reduce the size of the gradient ascent)

\[ U = \sum \log x_i \]
Step 3 - Explore phase

- If $x[n-1] = \rho[n-1]$:  
  
  - Remember $r[n] = \rho[n-1]$
  
  - Go to Enhance phase

- Else:
  
  - Keep target utility: $\mu[n] = \mu[n-1]$
  
  - Pick $\rho[n]$ randomly in $L(\mu[n])$
  
  - Repeat explore phase at most $N$ times, then move to Enhance phase (and reduce the size of the gradient ascent)
Step 3 - Explore phase

- If $x[n-1] = \rho[n-1]$:
  - Remember $r[n] = \rho[n-1]$
  - Go to Enhance phase

- Else:
  - Keep target utility: $\mu[n] = \mu[n-1]$
  - Pick $\rho[n]$ randomly in $L(\mu[n])$
  - Repeat explore phase at most $N$ times, then move to Enhance phase (and reduce the size of the gradient ascent)

$$U = \sum \log x_i$$
Step 3 - Explore phase

- If \( x[n - 1] = \rho[n - 1] \):
  - Remember \( r[n] = \rho[n - 1] \)
  - Go to Enhance phase
- Else:
  - Keep target utility: \( \mu[n] = \mu[n - 1] \)
  - Pick \( \rho[n] \) randomly in \( L(\mu[n]) \)
  - Repeat explore phase at most \( N \) times, then move to Enhance phase (and reduce the size of the gradient ascent)
Optimality result

Assumptions
- Fixed rate region $\Lambda[n] = \Lambda$
- Coordinate-convex rate region
  - Much weaker than convexity!

Theorem
The Enhance & Explore algorithm guarantees that, for any initial rate allocation $r[0]$, the utility of the last stable rate allocation $r[n]$ converges to the maximal utility for $n \to \infty$. 
Practical implementation

- Based on Click\textsuperscript{[1]} with MultiflowDispatcher\textsuperscript{[2]}

- Creation of 4 new Click elements
  - MFQueue
  - MFLeakyBucket
  - EEadapter
  - EEscheduler

- Evaluation with
  - Asus routers
  - ns-3

[2] Schiöberg et al., SyClick, 2009
Experimental results

- Deployment map:

- Without E&E:

- With E&E ($U_{prop}$):

![Diagram showing experimental results for UDP and TCP traffic with and without E&E.](image)
Experimental results

- Deployment map:

- Without E&E:

- With E&E ($U_{prop}$):

![Diagram](image-url)
Simulation results

**ns-3 simulator**

- Re-use of the same *Click* elements
- More controlled environment
- Possible estimation of the rate region
- Computation of optima

![Diagram showing simulation results with rate region boundary and optima for different flow rates.](image-url)
Simulation results

- Adaptivity to time-varying traffic
- Cyclic validation of last stable allocation $r[n]$
Conclusion

Problem

- Inefficient and/or unfair situations in WLANs
- Capture efficiency and fairness using a utility function
  - The feasible rate region is unknown!

Solution

- Successive decisions and measurements by the GW
- Optimal for a fixed rate region
- When rate region changes, keeps adapting
- More details in [1], with an extension to multi-hop networks

Future work:

- Downlink traffic
- Rate adaptation