Consort: Node-constrained Opportunistic Routing in Wireless Mesh Networks

Xi Fang, Dejun Yang, and Guoliang Xue

Arizona State University
Outline

- Motivation
- Related Work
- System Model
- Problem Formulation and Distributed Algorithm
- Performance Evaluation
- Conclusion
Outline

- Motivation
- Related Work
- System Model
- Problem Formulation and Distributed Algorithm
- Performance Evaluation
- Conclusion
Opportunistic Routing

MORE

Wireless Network
User behavior
Resource competition
What influences node behaviors

- Node *individual* requirement: *node max load constraint*

- Node *social* requirement: *node load balance constraint*
Node social requirement
Problems

- Node-constrained user utility optimization problem
- Node-constrained user profit optimization problem
Outline

• Motivation
• Related Work
• System Model
• Problem Formulation and Distributed Algorithm
• Performance Evaluation
• Conclusion
Related work

Opportunistic routing:

Related work

• Opportunistic routing:
Related work (cont’d)

- **Node load:**
Related work (cont’d)

- **Node load:**
Outline

- Motivation
- Related Work
- **System Model**
- Problem Formulation and Distributed Algorithm
- Performance Evaluation
- Conclusion
System Model

- Opportunistic multipath routing network sub-model
- Node constraint sub-model
- User utility and profit sub-model
System Model

- Opportunistic multipath routing network sub-model
- Node constraint sub-model
- User utility and profit sub-model
Opportunistic multipath routing network sub-model

- Multipath flow conservation constraint

$$\sum_{(u,v) \in E_k} r_k(u,v) - \sum_{(w,u) \in E_k} r_k(w,u) = h_k(u), \forall k \in [1, K]$$
Opportunistic multipath routing
network sub-model (cont’d)

• Coding constraint

\[ b_k(u) \cdot p(u, v) \geq r_k(u, v) \]
Opportunistic multipath routing network sub-model (cont’d)

- MAC broadcasting rate constraint

\[ \sum_{k \in [1, K]} B_k^{(t)}(u) + \sum_{k \in [1, K]} \sum_{v \in R(u)} B_k^{(t)}(v) \leq 1 \]
Opportunistic multipath routing network sub-model (cont’d)

- MAC broadcasting rate constraint

\[
\sum_{k \in [1,K]} B_k^{(t)}(u) + \sum_{k \in [1,K]} \sum_{v \in R(u)} B_k^{(t)}(v) \leq 1
\]

\[
\frac{C}{T} \sum_{t \in [1,T]} \sum_{k \in [1,K]} B_k^{(t)}(u) + \frac{C}{T} \sum_{t \in [1,T]} \sum_{k \in [1,K]} \sum_{v \in R(u)} B_k^{(t)}(v) \leq C
\]
Opportunistic multipath routing network sub-model (cont’d)

- MAC broadcasting rate constraint

\[
\frac{C}{T} \sum_{t \in [1,T]} \sum_{k \in [1,K]} B_k^{(t)}(u) + \frac{C}{T} \sum_{t \in [1,T]} \sum_{k \in [1,K]} \sum_{v \in R(u)} B_k^{(t)}(v) \leq C
\]

\[
b_k(u) = \lim_{T \to \infty} C \times \frac{\sum_{t \in [1,T]} B_k^{(t)}(u)}{T}
\]
Opportunistic multipath routing network sub-model (cont’d)

- MAC broadcasting rate constraint

\[
\frac{C}{T} \sum_{t \in [1,T]} \sum_{k \in [1,K]} B_k^{(t)}(u) + \frac{C}{T} \sum_{t \in [1,T]} \sum_{k \in [1,K]} \sum_{v \in R(u)} B_k^{(t)}(v) \leq C
\]

\[
b_k(u) = \lim_{T \to \infty} C \times \frac{\sum_{t \in [1,T]} B_k^{(t)}(u)}{T}
\]

\[
\sum_{k \in [1,K]} \sum_{v \in R(u)} b_k(v) \leq C
\]
System Model

- Opportunistic multipath routing network sub-model
- Node constraint sub-model
- User utility and profit sub-model
Node constraint sub-model

- Node max load constraint

\[ b(u) \leq \varphi(u), \varphi(u) > 0 \]
Node constraint sub-model

- Node load balance constraint

\[ |b(u) - b(v)| \leq \theta(u, v), \theta(u, v) > 0, \forall v \in A(u) \]
System Model

• Opportunistic multipath routing network sub-model

• Node constraint sub-model

• User utility and profit sub-model
User utility and profit sub-model

- **User utility function:**
  \[ U_k(\lambda_k) \]

- **User profit function:**
  \[ P_k(\lambda_k, \vec{b}_k) = U_k(\lambda_k) - \sum_{u \in V_k \setminus d_k} \sigma_k(u)b_k(u) \]
Outline

- Motivation
- Related Work
- System Model
- Problem Formulation and Distributed Algorithm
- Performance Evaluation
- Conclusion
Node-constrained User Utility Optimization

System 1: Maximize \( \sum_{k \in [1, K]} U_k(\lambda_k) \)

s.t. \( \sum_{(u,v) \in E_k} r_k(u,v) - \sum_{(w,u) \in E_k} r_k(w,u) = h_k(u), u \in V_k, k \in [1, K], \) 

\( \sum_{k \in [1,K]} b_k(u) + \sum_{k \in [1,K]} \sum_{v \in R(u)} b_k(v) \leq C, \forall u \in V_k \setminus s_k, \) 

\( b_k(u) \cdot p(u,v) \geq r_k(u,v), \forall (u,v) \in E_k, k \in [1, K], \) 

\( \sum_{k \in [1,K]} b_k(u) \leq \varphi(u), \forall u \in V_k \setminus d_k, \) 

\( | \sum_{k \in [1,K]} b_k(u_1) - \sum_{k \in [1,K]} b_k(u_2) | \leq \theta(u_1, u_2), \) 

\( \forall u_1 \neq u_2, u_1 \in V_k \setminus d_k, u_2 \in V_k \setminus d_k, u_2 \in A(u_1), \)

over: \( r_k(u,v) \in [0, C], \forall (u,v) \in E_k, k \in [1, K], \) 

\( b_k(u) \in [0, C], \forall u \in V_k \setminus d_k, k \in [1, K]. \)
Node-constrained User Profit Optimization

System 2: Maximize \( \sum_{k \in [1,K]} \left( U_k(\lambda_k) - \sum_{u \in V_k \setminus d_k} \sigma_k(u)b_k(u) \right) \),

s.t. \( \sum_{(u,v) \in E_k} r_k(u,v) - \sum_{(w,u) \in E_k} r_k(w,u) = h_k(u), \ u \in V_k, k \in [1,K], \)

\( \sum_{k \in [1,K]} b_k(u) + \sum_{k \in [1,K]} \sum_{v \in R(u)} b_k(v) \leq C, \ \forall u \in V_k \setminus s_k, \)

\( b_k(u) \cdot p(u,v) \geq r_k(u,v), \ \forall (u,v) \in E_k, k \in [1,K], \)

\( \sum_{k \in [1,K]} b_k(u) \leq \varphi(u), \ \forall u \in V_k \setminus d_k, \)

\( \sum_{k \in [1,K]} b_k(u_1) - \sum_{k \in [1,K]} b_k(u_2) \leq \theta(u_1, u_2), \)

\( \forall u_1 \neq u_2, u_1 \in V_k \setminus d_k, u_2 \in V_k \setminus d_k, u_2 \in A(u_1), \)

over: \( r_k(u,v) \in [0, C], \ \forall (u,v) \in E_k, k \in [1,K], \)

\( b_k(u) \in [0, C], \ \forall u \in V_k \setminus d_k, k \in [1,K]. \)
What we can achieve

- In each iteration, *each user* and *each node* individually adjusts its own behavior.
- Algorithm *converges* to the optimal.
- The *amount of constraint violation*, and *the gap between the optimal solution and our solution* in each iteration are computable.
High-level framework

- **Primal problem:**

  Minimize $f(\bar{x})$, s.t. $g(\bar{x}) \leq 0$, over: $\bar{x} \in \bar{X}$

- **Dual problem:**

  Maximize $q(\bar{\delta}) = \inf_{\bar{x} \in \bar{X}} (L(\bar{x}, \bar{\delta})) = \inf_{\bar{x} \in \bar{X}} (f(\bar{x}) + \bar{\delta}^T g(\bar{x}))$

  s.t. $\bar{\delta} \succeq 0$, over: $\bar{\delta} \in \mathbb{R}^\rho$
Based on our modified approximate dual subgradient method, in each iteration:

1. minimize \( L(\overrightarrow{x}, \overrightarrow{\delta}^{(i)}) + \sum_{n=1}^{N} \epsilon_n \| x_n - x_n^{(i-1)} \|^2 \) over \( \overrightarrow{x} \in \overrightarrow{X} \)

2. update \( \overrightarrow{\delta}^{(i+1)} = [\overrightarrow{\delta}^{(i)} + \eta g(\overrightarrow{x}^{(i)})]^+ \)

Primal solution is approximated by averaging:

\[
\hat{x}^{(i)} = \frac{1}{i} \sum_{m=0}^{i-1} \overrightarrow{x}^{(m)}, \quad i \geq 1
\]
Why distributed algorithm is possible?

Lagrangian analysis

\[ L(\vec{r}, \vec{b}, \vec{\alpha}, \vec{\beta}, \vec{\mu}, \vec{\omega}) = -\sum_{k \in [1,K]} U_k(\lambda_k) + \sum_{k \in [1,K]} \sum_{(u,v) \in E_k} \beta_k(u,v) r_k(u,v) \]
\[ + \sum_{k \in [1,K]} \sum_{u \in V_k \setminus s_k} \alpha(u) b_k(u) + \sum_{k \in [1,K]} \sum_{u \in V_k} \sum_{v \in R(u), v \neq s_k} \alpha(v) b_k(u) \]
\[ - \sum_{k \in [1,K]} \sum_{(u,v) \in E_k} \beta_k(u,v) b_k(u) p(u,v) + \sum_{u \in V} \mu(u) \sum_{k \in [1,K]} b_k(u) \]
\[ + \sum_{u_1, u_2 \in V : u_2 \in A(u_1)} \omega(u_1, u_2) \sum_{k \in [1,K]} (b_k(u_1) - b_k(u_2)) - C \sum_{u \in V} \alpha(u) \]
\[ - \theta(u_1, u_2) \sum_{u_1, u_2 \in V : u_2 \in A(u_1)} \omega(u_1, u_2) - \sum_{u \in V} \mu(u) \varphi(u). \]
Why distributed algorithm is possible? (cont’d)

- User $k$’s behavior:

Minimize $-U_k(\lambda_k) + \sum_{(u,v) \in E_k} \beta_k(u,v)r_k(u,v)$,

subject to $\sum_{(u,v) \in E_k} r_k(u,v) - \sum_{(w,u) \in E_k} r_k(w,u) = h_k(u), u \in V_k$

over: $r_k(u,v) \in [0, C], \forall (u,v) \in E_k$. 

Why distributed algorithm is possible? (cont’d)

- Node $u$’s behavior for user $k$:

Minimize $b_k(u) \left( \alpha(u)_{u \neq s_k} + \sum_{v \in R(u), v \neq s_k} \alpha(v) - \sum_{(u,v) \in E_k} \beta_k(u, v)p(u, v) \right. $

\[ + \sum_{v \in A(u)} \omega(u, v) - \sum_{v \in A(u)} \omega(v, u) + \mu(u) \right), \]

over: $b_k(u) \in [0, C]$, $u \in V_k \setminus d_k$. 
In each iteration:

1. minimize $L(\vec{x}, \vec{\delta}^{(i)}) + \sum_{n=1}^{N} \epsilon_n \|x_n-x_n^{(i-1)}\|^2$ over $\vec{x} \in \overline{X}$

2. update $\vec{\delta}^{(i+1)} = [\vec{\delta}^{(i)} + \eta g(\vec{x}^{(i)})]^+$
User $k$ updates its behavior in the $i$-th iteration:

\[
\text{Minimize } -U_k(\lambda_k^{(i)}) + \sum_{(u,v) \in E_k} \beta_k^{(i)}(u,v)r_k^{(i)}(u,v),
\]

s.t. \[\sum_{(u,v) \in E_k} r_k^{(i)}(u,v) - \sum_{(w,u) \in E_k} r_k^{(i)}(w,u) = h_k^{(i)}(u), u \in V_k\]

over : \[r_k^{(i)}(u,v) \in [0, C], \forall (u,v) \in E_k.\]
Equivalent flow-path formulation
Flow-path formulation:

\[
\begin{align*}
\text{Minimize} & \quad -U_k\left(\sum_{\pi \in P} \gamma_k(\pi)\right) + \sum_{\pi \in P} \kappa_k(\pi) \gamma_k(\pi), \\
\text{over:} & \quad \sum_{\pi \in P} \gamma_k(\pi) \in [0, C],
\end{align*}
\]
Flow-path formulation:

\[
\text{Minimize} \quad -U_k \left( \sum_{\pi \in P} \gamma_k(\pi) \right) + \sum_{\pi \in P} \kappa_k(\pi) \gamma_k(\pi),
\]

over:

\[
\sum_{\pi \in P} \gamma_k(\pi) \in [0, C],
\]

\[
\text{Minimize} \quad -U_k(\Gamma_k) + \kappa_k^{\text{min}} \Gamma_k, \quad \text{over: } \Gamma_k \in [0, C]
\]
Node $u$ updates its behavior for user $k$:

Minimize \[
\left( \alpha^{(i)}(u)_{u \neq s_k} + \sum_{v \in R(u), v \neq s_k} \alpha^{(i)}(v) - \sum_{(u, v) \in E_k} \beta^{(i)}_k(u, v) p(u, v) \right)
 + \mu^{(i)}(u) + \sum_{v \in A(u)} \omega^{(i)}(u, v) - \sum_{v \in A(u)} \omega^{(i)}(v, u) \right) b_k^{(i)}(u)
 + \epsilon \| b_k^{(i)}(u) - b_k^{(i-1)}(u) \|^2,
\]
over: $b_k^{(i)}(u) \in [0, C]$, $u \in V_k \setminus d_k$. 
In each iteration:

1. minimize \( L(\overrightarrow{x}, \overrightarrow{\delta}^{(i)}) + \sum_{n=1}^{N} \epsilon_{n} \|x_{n} - x_{n}^{(i-1)}\|^2 \) over \( \overrightarrow{x} \in \overrightarrow{X} \)

2. update \( \overrightarrow{\delta}^{(i+1)} = [ \overrightarrow{\delta}^{(i)} + \eta g(\overrightarrow{x}^{(i)}) ]^{+} \)
Updating Lagrange Multipliers

\[ \forall u \in V, \alpha^{(i)}(u) = [\alpha^{(i-1)}(u) + \eta M^{(i-1)}_u]^{+}, \]
\[ \forall (u, v) \in E_k, \beta_k^{(i)}(u, v) = [\beta_k^{(i-1)}(u, v) + \eta C^{(i-1)}_{(k,u,v)}]^{+}, \]
\[ \forall u \in V, \mu^{(i)}(u) = [\mu^{(i-1)}(u) + \eta G^{(i-1)}_u]^{+}, \]
\[ \forall u \in V \text{ and } v \in A(u), \omega^{(i)}(u, v) = [\omega^{(i-1)}(u, v) + \eta H^{(i-1)}_{(u,v)}]^{+}, \]

where

\[ M^{(i-1)}_u = \sum_{k \in [1,K], u \neq s_k} b_k^{(i-1)}(u) + \sum_{k \in [1,K]} \sum_{v \in R(u), u \neq s_k} b_k^{(i-1)}(v) - C, \]
\[ C^{(i-1)}_{(k,u,v)} = r_k^{(i-1)}(u, v) - b_k^{(i-1)}(u) \cdot p(u, v); \]
\[ G^{(i-1)}_u = \sum_{k \in [1,K]} b_k^{(i-1)}(u) - \varphi(u) \]
\[ H^{(i-1)}_{(u,v)} = \sum_{k \in [1,K]} (b_k^{(i-1)}(u) - b_k^{(i-1)}(v)) - \theta(u, v), \]
Optimality and Convergence

- An upper bound on the amount of constraint violation of our primal solution $\hat{x}^{(i)}$

\[ \| g(\hat{x}^{(i)})^+ \| \leq \frac{B}{i\eta} \]

- This property states that the amount of constraint violation of the primal solution diminishes to zero at the rate $1/i$ as the number of iterations $i$ increases.
A lower and upper bound on $f(\hat{x}^{(i)})$:

$$f^* - \frac{B^2}{i\eta} \leq f(\hat{x}^{(i)}) \leq f^* + \frac{||\overrightarrow{\delta}^{(0)}||^2}{2i\eta} + \frac{\eta L^2}{2} + \epsilon KC^2|V|$$

This property states that Consort converges to the optimal with the rate $1/i$. 

Optimality and Convergence (cont’d)
Outline

- Motivation
- Related Work
- System Model
- Problem Formulation and Distributed Algorithm
- Performance Evaluation
- Conclusion
Simulation Setup

- **Nodes**
  - We randomly distributed 36 nodes in a 1000m×1000m square region

- **Users**
  - 10, 20, and 30 users with sources and destinations being randomly selected
Performance Comparison

- Utility and profit

Utility

Profit
Performance Comparison (cont’d)

- Network violation ratio
Performance Comparison (cont’d)

- Fairness index
Outline

- Motivation
- Related Work
- System Model
- Problem Formulation and Distributed Algorithm
- Performance Evaluation
- Conclusion
Conclusions

- We have studied how to allocate network resource to optimize the total *utility* or *profit* of multiple simultaneous users in a WMN subject to node constraints by using opportunistic routing.
- We have formulated these two problems as two convex programming systems, and have presented a distributed iterative algorithm framework.
- We have proved its convergence, and have also provided bounds on the amount of constraint violation and the gap between our solution and the optimal solution in each iteration.
THANK YOU!
Questions?