Wireless Network Inference and Optimization: Algorithm Design and Implementation

Yalin E. Sagduyu, Senior Member, IEEE, Yi Shi, Senior Member, IEEE, Anthony Fanous, Member, IEEE, and Jason H. Li, Member, IEEE

Abstract—This paper addresses the problem of joint inference and optimization in wireless networks. An optimization framework based on information-geometric network inference is developed and implemented (using a real radio emulation testbed) with scalable solutions to infer the end-to-end rate distributions of stochastic network flows from link rate measurements. The proposed low-complexity solutions apply when the underlying network inference (network tomography) problem can be decomposed to smaller-size subproblems that are solved independently by partially inferring only the flow rates of interest. The solutions are extended to infer flow rates jointly with link loss rates when retransmissions are considered over unreliable wireless links. By using the inferred distributions of flow rates, an inference mechanism is presented to optimize the network performance. First, the distributions of flow rates are inferred from the average rate measurements on selected links. Then, the distributions of all links rates are computed from the inferred distributions of flow rates. Finally, these inference results are used with network optimization. The weighted sum of link outage probabilities is minimized by adapting power control or routing decisions in mobile wireless access. This approach is iterated between network inference and optimization, providing link outage and end-to-end throughput gains compared to static approaches with fixed network (inference and optimization) parameters. The joint network inference and optimization framework is implemented with real configurable radios and the performance is verified with hardware-in-the-loop emulation test results that are obtained with actual radio transmissions over emulated channels.

Index Terms—Network inference, network tomography, network optimization, wireless networks, routing, power control, emulation, testbed.

1 INTRODUCTION

We consider the problem of joint inference and optimization in wireless networks. Network inference (tomography) has the objective of learning network features, e.g., end-to-end flow rates, from a limited number of measurements, e.g., rates measured on selected links [1]–[7]. Since multiple flow rates possibly share a single link and the dimension of observations is typically smaller than the dimension of variables, this is a statistical inference problem for underdetermined systems. There have been several solutions typically customized for wired networks [1]–[7]. When applied to a wireless network, there is a need to infer statistical distributions of flow rates subject to the potential effects of measurement noise, wireless channel properties, network dynamics, and underlying network execution protocols.

In this paper, we follow an information-geometric optimization approach for network inference. Information geometry provides a systematic mechanism to infer the probability distributions from incomplete observations [8], [9] by maximizing the entropy of the unknown probability space (i.e., by following the maximum entropy principle). The underlying optimization problem aims to minimize the distance to prior distributions subject to the constraints imposed by the measurements. This approach has been applied to wireless setting in [10] with the focus on distributed algorithms in small-scale networks. Maximum entropy models have been also used to infer flow rates with given distributions [11] or under linear independence of flows [12] in wired networks.

Wireless networks have unknown/arbitrary distributions of flows with different source-destination pairs. Wireless measurements (e.g., spectrum sensing [13]) are typically prone to errors. In addition, with diverse mobile wireless applications, mobility can further increase network dynamics. Therefore, low-complexity solutions that can quickly infer fast-changing network dynamics are needed while mitigating measurement noise effects. Consequently, a systematic mechanism is necessary to utilize this inferred statistical information in improving the network performance. We design a network inference and optimization framework to achieve this goal.

The contributions of the paper are given as follows:

1) We exploit network structure and then develop low-complexity algorithms for large-sized networks with certain structures that can scale up with the number of unknown flows and consider possible measurement noise in a convex optimization framework. We provide a systematic mechanism to decompose the network inference problem on a partial routing topology to subproblems, each inferring a subset of unknown flow rate distributions. In addition, we provide a partial inference mechanism than can infer a subset of unknown flow rate distributions of interest. We specify the smaller solution space with...
Fig. 1. The flowchart for joint network inference and optimization.

reduced complexity for both mechanisms and verify the performance. Then, we extend our approach to infer end-to-end flow rates jointly with link loss rates in a wireless network of unreliable links that necessitate retransmissions of failed packets.

2) We develop the capability to optimize the network performance by using the inference results. First, we measure the average rates on selected links and obtain the distributions of flow rates. Then, unlike classical network optimization solutions on static networks, e.g., rate control in [14], we compute the distributions of rates on all links by aggregating the inferred flow rates according to the routing matrix. Based on the inferred distributions, we minimize the weighted sum of link outage probabilities by varying network parameters such as power control or adaptive routing parameters. The optimized network operation updates the channel capacity constraints (under power control) or the routing matrix (under adaptive routing) and then network inference is repeated based on the updated routing matrix. We iterate between network inference and optimization until the system converges with the optimal power control or routing solution. This procedure is shown in Fig. 1. Our results show that this joint inference and optimization approach provides significant outage and throughput performance gains compared to static approach with fixed network parameters.

3) We implement and validate the joint network inference and optimization approach in a realistic wireless emulation testbed environment for multiple runs. This capability provides high-fidelity radio tests and reliable performance comparison of different schemes under identical channel conditions. We measure link rates, infer flow rates, and update routing operation to minimize the weighted sum of link outage probabilities. Our testbed results show fast adaptation to network topology changes and verify throughput gains over static solutions via real-time video transmissions.

The rest of this paper is organized as follows. Section 2 introduces the network inference approach to determine the end-to-end flow rates based on the link rate measurements. Low complexity solutions are presented in Section 3. The approach is extended to joint inference of flow and link loss rates in Section 4. Section 5 integrates network inference with network optimization based on adaptive routing or power control and provides a joint solution. This joint approach of network inference and optimization is implemented and tested in a realistic wireless emulation testbed environment in Section 6. Section 7 summarizes the contributions.

2 INFORMATION-GEOMETRIC NETWORK INFERENCE

The classical problem of network inference (tomography) is to determine all the end-to-end flow rates based on the link rate measurements [1]. For a network with \( m \) nodes, the number of possible flows is \( m(m - 1) \), while the number of measured links is usually less than \( m(m - 1) \), unless the network is fully connected and all links can be measured. Therefore, this is a statistical inference problem for underdetermined systems. Let \( \mathbf{X} \) denote the vector of end-to-end information flow rates, where its \( j \)-th element \( x_j \) is the rate of the \( j \)-th source-destination pair. Let \( \mathbf{Y} \) denote the vector of link-level rate measurements, where its \( i \)-th element \( y_i \) is the traffic rate on link \( i \). We can regard both \( \mathbf{X} \) and \( \mathbf{Y} \) as random variables. The randomness in \( \mathbf{X} \) may be due to the stochastic packet traffic, whereas the randomness in \( \mathbf{Y} \) may be due to its dependency on \( \mathbf{X} \) and the measurement noise \( \mathbf{N} \).

We assume that flow rates take values from a discrete set \( \mathcal{X} \). The size \( |\mathcal{X}|^{|\mathbf{X}|} \) determines the resolution of the network inference problem, where \( |\mathcal{X}| \) is the size of \( \mathcal{X} \) and \( |\mathbf{X}| \) is the size of \( \mathbf{X} \). Link rate measurements are expressed in terms of end-to-end flow rates as

\[
\mathbf{Y} = \mathbf{AX} + \mathbf{N},
\]

where \( \mathbf{A} \) is the routing matrix (\( A_{i,j} \) is the fraction of the \( j \)-th flow on the \( i \)-th link) and \( \mathbf{N} \) is the measurement noise.

The dimension of \( \mathbf{Y} \) is usually smaller than that of \( \mathbf{X} \). Therefore, it is an underdetermined system, where solutions cannot be uniquely determined. Hence, we pursue a statistical inference approach, where we infer the probability distribution \( \mathbf{p}_{\mathbf{X}} \) for \( \mathbf{X} \) based on the moving average of \( k \) measurements \( \mathbf{Y}_k \), a (prior) distribution \( \mathbf{q}_{\mathbf{X}} \) on \( \mathbf{X} \), and an expectation \( \mathbf{N} \) over a (prior) distribution on \( \mathbf{N} \). Initially, \( \mathbf{q}_{\mathbf{X}} \) can be chosen as a uniform distribution with the maximum entropy (i.e., maximum uncertainty). That is, \( \mathbf{p}_{x_j} = 1/|\mathcal{X}| \)
for each $x_j \in \mathcal{X}$. Similarly, we can assume small values of $\tilde{N}$ initially.

As we will discuss in Section 5 in detail, the attempt to better capture and optimization of information flows may require inference of distributions rather than average values. With measurements $Y_k$, the distribution $p_X$ can be inferred by minimizing the sum of Kullback-Leibler (KL) divergence [16] (distance from prior distributions) and estimation error on noise as follows.

\[
\begin{align*}
\text{minimize} & \quad D(p_X || q_X) + |N - \tilde{N}| \\
\text{subject to} & \quad AE_{p_X}[X] + \tilde{N} = Y_k \\
& \quad D(p_X || q_X) = \sum_{x_{j=1}}^{[|X|]} \sum_{x \in \mathcal{X}} p_{x_j,x} \ln \frac{p_{x_j,x}}{q_{x_j,x}} \\
& \quad E_{p_X}[X] = \left( \sum_{x \in \mathcal{X}} x p_{x_j,x} \right)_{j=1} \\
& \quad \sum_{x \in \mathcal{X}} p_{x_j,x} = 1, j \in \{1, |\mathcal{X}|\} \\
\text{variables} & \quad 0 \leq p_{x_j,x} \leq 1, j \in \{1, |\mathcal{X}|\}, x \in \mathcal{X} \\
& \quad 0 \leq \tilde{n}_i \leq N_{\max}, i \in \{1, |Y|\},
\end{align*}
\]

where $p_{x_j,x}$ is the probability of $x_j = x$, $\tilde{n}_i$ is the expectation of measurement noise, and $N_{\max}$ is an upper bound on the expected value of measurement noise.\(^1\) This optimization is a convex program with a convex objective function and linear constraints.\(^2\) The optimization problem (1) is solved iteratively by updating $Y_k$ with new measurements.

The number of variables for distribution is $|X||\mathcal{X}|$ and the number of variables for expectation is $|Y|$. The number of equality constraints is $|X|+|Y|$. Both $|X|$ and $|Y|$ increase with network size and the resulting problem becomes computationally difficult. In the next section, we discuss low-complexity solutions for large-sized networks by reducing the solution space.

To illustrate the performance in this preliminary case (as solutions to (1)), we apply a two-path routing algorithm, where data from a source node can be transmitted along two different paths to the destination, each with 50% of data. There are 15 nodes in the network. Source-destination pairs and paths of the two flows generated are shown in Table 1 and Fig. 2. We consider a special case that each flow can be either off (with rate 0) or on (with rate 1), i.e., $\mathcal{X} = \{0, 1\}$. We randomly generate on/off events with some randomly selected probabilities $p_{x_1,0} = 0.3323$ and $p_{x_2,0} = 0.4018$, where $p_{x_1,0}$ is the probability that flow $x_1$ is off. Similarly, we use $p_{x_1,1}$ to denote the probability that flow $x_1$ is on. Network inference algorithm, as solutions to (1), ensures that the simulated probabilities over the first $k$ random events will converge to these values.

We show the inferred distribution $p_X$ in Figure 3 (the straight lines show the actual probabilities). We find that the inferred distribution $p_X$ is the same as the simulated probabilities after several estimations and thus it will converge to the actual flow rate distributions. The complexity of solving a linear program is $O(n^3)$, where $n$ is the number of variables in the standard form [17]. Therefore, low-complexity solutions to network inference problems are needed.

### 3 Low-Complexity Network Inference

We present two complexity reduction techniques for network inference in this section. Each of them requires some condition such that it can be successfully applied. We note that these techniques are enhancements for the network inference and optimization framework proposed in this paper, i.e., the entire framework can still be run even if these techniques cannot be applied.

---

\(^1\) $D(\cdot)$ function is undefined if $q_{x_j,x} = 0$. In this case, we define $D(\cdot)$ by adding a small number to the denominator.

\(^2\) For general measurements $T(X)$ (other than average links rates in the form $AX$), the link measurement constraint is changed to $E_{p_X}[T(X)] + N = Y_k$ and an additional link capacity constraint $AE_{p_X}[X] \leq C$ is introduced, where $C$ is the set of link capacities. This new problem is still convex.
3.1 Network Inference Decomposition

The first complexity reduction technique is based on “divide and conquer”, which is an effective approach to reduce complexity. The basic idea is that if we can decompose a problem into subproblems and each of them can be solved independently, then the overall complexity can be reduced as long as the complexity is super-linear with the problem size. The network inference problem has a super-linear complexity. In many instances, the network inference problem can be decomposed into subproblems and each of them can be solved independently. A given routing matrix \( A \) may be decomposable, i.e., we may find that some flows only use some links that are not used by other flows. For example, the network connectivity graph may have several connected components. Then a source node in a connected component can only use links in this component to transmit its data while another source node not present in this connected component will not use any link in this connected component.

We identify such flows by starting with a group \( F[1] \) of a single flow. We also have a group \( L[1] \) of links used by this flow. We then add any flow into \( F[1] \) that shares some links in \( L[1] \) and then update \( F[1] \) to include new links used by flows in \( F[1] \). This process continues until we cannot add any flow into \( F[1] \). If \( F[1] \) does not include all flows, then we can have a subproblem based on \( F[1] \) and \( L[1] \). We continue with the same process to find other subproblems. If we reorder flows and links based on these groups of flows and links, we can decompose the routing matrix into

\[
A = \text{diag} \{ A[1], A[2], \ldots, A[C] \}, \tag{2}
\]

where \( C \) is the number of subproblems. Denote flow rates in the \( c \)-th group as \( X[c] \), link measurements in the \( c \)-th group as \( Y[c] \), and noise in the \( c \)-th group as \( N[c] \). Then, the optimization problem in (1) can be rewritten as follows.

\[
\text{minimize} \quad \sum_{c=1}^{C} D(p_{X[c]}||q_{X[c]}) + |N[c] - \bar{N}[c]|
\]

subject to \( A[c] = E_{p_{X[c]}|X[c]}[X[c]] + N[c] = (Y[c])_k \), \( c \in [1, C] \)

\[
D(p_{X[c]}||q_{X[c]}) = \sum_{x \in X} p_{x_i,x} \ln \left( \frac{p_{x_i,x}}{q_{x_i,x}} \right), \quad c \in [1, C]
\]

\[
E_{p_{X[c]}|X[c]}[X[c]] = \left( \sum_{x \in X} k_{x_j,x} x_{j=1} \right)_{c=1,\ldots,C}, \quad c \in [1, C]
\]

\[
\sum_{x \in X} p_{x_i,x} = 1, j \in [1, |X|]
\]

\[
\text{variables} \quad 0 \leq p_{x_i,x} \leq 1, j \in [1, |X|], x \in X
\]

\[
0 \leq \bar{n}_i \leq N_{\text{max}}, i \in [1, |Y|]. \tag{3}
\]

The above problem can be decomposed into \( C \) independent subproblems, where the \( c \)-th subproblem is as follows.

\[
\text{minimize} \quad D(p_{X[c]}||q_{X[c]}) + |N[c] - \bar{N}[c]|, \quad \sum_{x \in X} p_{x_i,x} \ln \left( \frac{p_{x_i,x}}{q_{x_i,x}} \right), c \in [1, C]
\]

subject to \( A[c] = E_{p_{X[c]}|X[c]}[X[c]] + N[c] = (Y[c])_k \)

\[
E_{p_{X[c]}|X[c]}[X[c]] = \left( \sum_{x \in X} k_{x_j,x} x_{j=1} \right)_{c=1,\ldots,C}, \quad c \in [1, C]
\]

\[
\sum_{x \in X} p_{x_i,x} = 1, j \in [1, |X|]
\]

\[
\text{variables} \quad \{ p_{X[c]}, \bar{N}[c] \} \tag{4}
\]

Due to smaller problem sizes, the total complexity to solve these subproblems is much less than solving the problem (1).

We illustrate the performance by inferring the rates of 5 randomly generated flows with source-destination pairs and relays shown in Table 2 and Fig. 4. We consider a special case that each flow \( j \) can be either on (with rate 1) or off (with rate 0). We randomly generate these on/off events by the following (randomly selected) off probabilities: \( p_{x_i,0} = 0.1837, p_{x_2,0} = 0.9306, p_{x_3,0} = 0.8837, p_{x_4,0} = 0.1020, \) and \( p_{x_5,0} = 0.7714 \). By following the decomposition procedure, we obtain two subproblems by having the first three flows in one group and the last two flows in the other group. Probabilities \( p_{x_i,0}, j = 1, 2, 3, \) are inferred by the first subproblem and probabilities \( p_{x_i,0}, j = 4, 5, \) are inferred by the second subproblem. Note that the original problem has 5 flows, 7 links, and thus \( 5 \times 2 + 7 = 17 \) variables. We decompose it into two subproblems. Problem 1 has 3 flows, 3 links, and thus \( 3 \times 2 + 3 = 9 \) variables. Problem 2 has 2 flows, 4 links, and thus \( 2 \times 2 + 4 = 8 \) variables. Instead of one 17-dimensional problem, we solve two independent problems, one 9-dimensional and one 8-dimensional, with total complexity reduced compared to the single higher-dimensional problem.

Figure 4 shows how the probabilities in the decomposed case of two subproblems are inferred over time. We do not show probability \( p_{x_1,1} \) in Figure 5, since \( p_{x_1,1} = 1 - p_{x_1,0} \). The results verify the convergence to actual probabilities (straight lines) as the number of measurements increases.

3.2 Partial Network Inference

Another complexity reduction technique is to remove any unnecessary variables. As a result, problem size can be decreased and thus its complexity can also be reduced.
Among all possible flow rates, we may only want to determine a small number of flow rates. We can still solve a complete problem to obtain distributions of all flow rates, which include distributions of flow rates of no interest. But since we do not want to determine all flow rates, those flow rates of no interest are unnecessary variables in the optimization problem. Thus, a better approach with much lower complexity can be developed as follows. For those flow rates that we are not interested in, we combine them together with the measurement noises aggregated as new noise terms. We obtain the following problem, which does not have any unnecessary variables.

\[
\begin{align*}
\text{minimize} & \quad D(\mathbf{p}_X \| q_X) + |\tilde{M} - \tilde{M}| \\
\text{subject to} & \quad A E \mathbf{p}_X [\hat{X}] + \tilde{M} = \tilde{Y}_k \\
& \quad D(\mathbf{p}_X \| q_X) = \sum_{j=1}^{5} \sum_{x \in \mathcal{X}} p_{x_j,x} \ln \frac{p_{x_j,x}}{q_{x_j,x}} \\
& \quad E_{\mathbf{p}_X} [\hat{X}] = (\sum_{x \in \mathcal{X}} x p_{x_j,x})_{j=1}^{5} \\
& \quad \sum_{x \in \mathcal{X}} p_{x_j,x} = 1, j \in [1, |\hat{X}|] \\
\text{variables} & \quad 0 \leq p_{x_j,x} \leq 1, j \in [1, |\hat{X}|], x \in \hat{X} \\
& \quad 0 \leq \hat{n}_i \leq N_{\text{max}}, i \in [1, |\hat{Y}|], \\
\end{align*}
\]  

(5)

where \( \hat{X} \) are the flow rates of interest, \( \tilde{M} \) includes both the measurement noise and the flow rates of no interest combined, \( A \) is the routing sub-matrix for flows of interest, and \( \tilde{Y} \) are the remaining measurements after removing measurements on links that are not used by flows of interest. We also need to determine the upper bound \( N_{\text{max}} \) for \( M \) based on routing matrix \( A \). The number of variables is \( |\hat{X}| + |\hat{Y}| \) and the number of constraints is \( |\hat{X}| + |\hat{Y}| \). This new problem has a smaller size and lower complexity than the full problem of network inference (1).

We evaluate the performance by inferring the rates of the first two flows out of 5 randomly generated flows with source-destination pairs and relays shown in Table 3. The full convex program has 5 flows and 9 links, and thus has \( 5 \times 2 + 9 = 19 \) variables. Since we only want to infer flows 1 and 2, we do not need to consider links (5,4), (6,4), (6,8), (10,6), since there is no flow 1 or 2 on these links. The number of remaining links is 5 and the number of remaining flows is 2. Thus, the reduced problem has 2 flows and 5 links. The number of variables is \( 2 \times 2 + 5 = 9 \). Instead of a 19-dimensional problem, we solve a 9-dimensional problem and the complexity is reduced. Figure 6 shows how these probabilities are inferred over time and the results verify convergence to the actual probabilities (straight lines).

### 4 Joint Inference of Flow and Link Loss Rates

For transmissions over a wireless link, the link loss probability may not be negligible (even after applying channel coding that imposes a particular link rate). Thus, we need to develop the actual link rate formula (with consideration of link loss) in the network inference problem.

Suppose that the data rate over a link \( i \) is \( z_i \), link loss probability is \( \varepsilon_i \), and they are independent from each other. Each packet needs to be retransmitted, if the transmission fails. Therefore, the total number of transmissions to deliver one packet on a link is a geometric random variable with the expected value \( \frac{1}{1-\varepsilon_i} \). Then the expected traffic rate (including both transmissions and retransmissions) is \( y_i = \frac{z_i}{1-\varepsilon_i} \).

This way, we obtain the rate constraint as

\[
AE_{\mathbf{p}_X} [x_i] = (Y_k)_i, 
\]  

(6)

for any link \( i \), where \( (Y_k)_i \) is the \( k \)-th measurement on link \( i \)'s rate. We can re-write constraint (6) as

\[
AE_{\mathbf{p}_X} [X] + \text{diag}(\varepsilon_1, \ldots, \varepsilon_5) \ Y_k = Y_k. 
\]  

(7)

Considering the measurement noise, we can express the measurement constraint as

\[
AE_{\mathbf{p}_X} [X] + \text{diag}(\varepsilon_1, \ldots, \varepsilon_5) \ Y_k + \tilde{N} = Y_k. 
\]  

(8)
Then, we formulate the following network inference problem.

\[
\begin{align*}
\text{minimize} & \quad D(p_X || q_X) + |N - \hat{N}| + |\bar{\varepsilon} - \bar{\bar{\varepsilon}}| \\
\text{subject to} & \quad D(p_X || q_X) = \sum_{j \in L} \sum_{x \in X} p_{x,j} \ln \frac{p_{x,j}}{q_{x,j}} \\
& \quad AE_{p_X}[X] + \text{diag}(\bar{\varepsilon}_1, ..., \bar{\varepsilon}_|Y|) Y_k + \hat{N} = Y_k \\
& \quad E_{p_X}[X] = (\sum_{x \in X} x p_{x,j,x})_{j=1} \\
& \quad \sum_{x \in X} p_{x,j} = 1, j \in [1, |X|] \\
& \quad \text{variables} \quad 0 \leq p_{x,j} \leq 1, j \in [1, |X|], x \in X \\
& \quad 0 \leq \bar{n}_i \leq N_{\max}, 0 \leq \bar{\varepsilon}_i \leq \varepsilon_{\max}, i \in [1, |Y|],
\end{align*}
\]

where \(\bar{\varepsilon}\) is the expectation of \(\varepsilon\), \(\bar{\varepsilon}_i\) is the previous estimation on \(\varepsilon_i\), and \(\varepsilon_{\max}\) is an upper bound on link loss rate. We evaluate the performance by inferring link loss probabilities together with rates of two flows randomly generated. The corresponding source-destination pairs and relays are shown in Table 4. Each link has some error probability within [0, 0.05]. Figure 7 shows how the flow rate distributions are inferred as solutions to (9) over time under link loss events and the results verify convergence to the actual probabilities (straight lines).

5 Joint Network Inference and Optimization

There have been many efforts on optimizing various aspects of wireless networks (see e.g., [20] and references therein). These efforts typically focused on optimization of network parameters for a given network status (e.g., routing [21], network coding [22], and relaying [23]). In a real system, network status is not given and thus needs to be inferred. In this section, we study how we can use the network inference results to optimize the dynamic network performance.

We develop a joint network inference and optimization framework to dynamically optimize the network performance by using the network inference results through the iterative algorithm shown in Figure 1. We assume wireless networks with slow dynamics such that the iterative algorithm can converge before the next system status update. We discussed the network inference step in the previous sections. After the network inference step, we have obtained the distributions of end-to-end flow rates \(X\). According to the routing matrix, we can determine the link rate distributions \(Y\). We want to minimize the weighted sum of the link outage probabilities \(w_i \cdot P[y_i > C_i]\), where \(C_i\) is the capacity of the link \(i\), and \(w_i\) is the weight of link \(i\). This, in turns, improves the end-to-end throughput and also enhances other link performances: for instance, if the rate over a link is much lower than its capacity, then the delay over that link will be small. To minimize the weighted sum of link outage probabilities, we can change either the transmission power on each link (and thus \(C_i\) is changed) or the routing matrix (and thus \(y_i\) is changed). We study both approaches subsequently. After the network optimization step, the current network behavior is updated with the optimal network parameters and network inference is run in the next iteration.

When link outage probabilities are small, minimizing the total weighted outage probability is equivalent to maximizing the total throughput by setting

\[
w_i = \sum_{j \in [1, |X|]} a_{j,i} r_j,
\]

where \(a_{j,i}\) is the fraction of flow \(j\)'s data transmitted on link \(i\) and \(r_j\) is the expected transmission rate at the source of flow \(j\). To see this, we can identify all paths for each flow \(j\) and the fraction \(w_{j,p}\) of flow \(j\)'s data transmitted on path \(p\). Then we have

\[
a_{j,i} = \sum_{p \in \mathcal{L}_{j,p}} w_{j,p},
\]

where \(\mathcal{L}_{j,p}\) is the set of links in path \(p\) of flow \(j\). For the outage probability \(P_{out}^i\) on link \(i\), the expected total throughput is

\[
\sum_j r_j \sum_p w_{j,p} \prod_{i \in \mathcal{L}_{j,p}} (1 - P_{out}^i) \\
\approx \sum_j r_j \sum_p w_{j,p} (1 - \sum_{i \in \mathcal{L}_{j,p}} P_{out}^i) \\
= \sum_j r_j \sum_p w_{j,p} - \sum_j r_j \sum_i w_{j,p} \sum_{i \in \mathcal{L}_{j,p}} P_{out}^i \\
= \sum_j r_j - \sum_i P_{out}^i \sum_j r_j \sum_{i \in \mathcal{L}_{j,p}} w_{j,p} \\
= \sum_j r_j - \sum_i P_{out}^i \sum_j r_j a_{j,i} \\
= \sum_j r_j - \sum_i P_{out}^i w_i,
\]

where the approximation (i.e., \(\prod_{i \in \mathcal{L}_{j,p}} (1 - P_{out}^i) \approx 1 - \sum_{i \in \mathcal{L}_{j,p}} P_{out}^i\)) holds due to small link outage probabilities. The fourth equation holds since \(\sum_j w_{j,p} = 1\), the fifth equation holds by (10), and the last equation holds by our setting on \(w_i\). Since \(\sum_j r_j\) is a constant, the above
The network topology used for performance evaluation of 8, 14, 15, 11, 8, 10, 11, 14, 5, 4, 12, 13, 1, 4, 8, 9, and links from our measurements. According to routing

P = total power constraint

e outage probabilities over all the links while satisfying some transmission powers to minimize the sum of the weighted

We now consider an application that nodes adjust their transmission powers to minimize the sum of the weighted outage probabilities to each link, as shown in Figure 8(b). If the outage rate of each link is assumed to be a known function of the transmission power, we can solve (11) via exhaustive search over discretized

\[ P_{\text{max}} = 3P, \quad \bar{P} = -90 \text{dBm}. \]

The transmission powers are selected from \([-98, -85] \text{ dBm}\) (with resolution 0.3 dBm). The transmission powers to each link are measured and the new power allocation is determined. Figure 9 shows the optimized power allocation results over time that are obtained from joint network inference and power control with quick convergence. For given power levels, the end-to-end throughput is defined as the expected sum rate of flows delivered without any link outage on paths between source-destination pairs. Enabled by network inference, the performance gain for outage probability and throughput is shown in Figure 10, where we compare two cases: (i)

\[ \text{minimize } \sum \alpha_{ij} P_{ij} \left( y_{ij} > C_i(P_i) \right) \]

subject to

\[ \sum \beta_i P_i \leq P_{\text{max}} \]

\[ E[y_{ij}] \leq C_i(P_i) \quad \forall i \]

variables \{P_i, \forall i\}. \quad (11)

For (11), the distribution of \(\{y_i, \forall i\}\) is obtained through either network inference or measurement, which depends on transmission powers \(P_i\) through link outage objectives and link capacity constraints. Meanwhile, the power \(P_i\) is optimized depending on \(P_{y_{ij}}\).

This problem has non-convex objective and constraint functions. There is no general low complexity algorithm for non-convex optimization problems and each of them can be solved via customized approaches, e.g., decomposition, duality [18] and approximation [19]. For numerical results, we simply solve (11) via exhaustive search over discretized

\(P_i\) values. We consider fixed routing parameters \(\alpha = 0.1, \beta = 0.1, \gamma = 0.5\) in Figure 8(b). Rates over links (1, 3, 2, 3, 3, 4) are measured and transmission powers over all links are optimized. To establish a realistic capacity term \(C_i(P_i)\), we run 200 test packet transmissions with actual radios, RouterStation Pros, over a wireless network emulator (the details of network emulation are presented in Section 6) and measure the capacity.

The transmission powers are selected from \([-98, -85] \text{ dBm}\) (with resolution 0.3 dBm). The transmission powers to each link are measured and the new power allocation is determined. Figure 9 shows the optimized power allocation results over time that are obtained from joint network inference and power control with quick convergence. For given power levels, the end-to-end throughput is defined as the expected sum rate of flows delivered without any link outage on paths between source-destination pairs. Enabled by network inference, the performance gain for outage probability and throughput is shown in Figure 10, where we compare two cases: (i)

\[ \text{minimize } \sum \alpha_{ij} P_{ij} \left( y_{ij} > C_i(P_i) \right) \]

subject to

\[ \sum \beta_i P_i \leq P_{\text{max}} \]

\[ E[y_{ij}] \leq C_i(P_i) \quad \forall i \]

variables \{P_i, \forall i\}. \quad (11)

For (11), the distribution of \(\{y_i, \forall i\}\) is obtained through either network inference or measurement, which depends on transmission powers \(P_i\) through link outage objectives and link capacity constraints. Meanwhile, the power \(P_i\) is optimized depending on \(P_{y_{ij}}\).

This problem has non-convex objective and constraint functions. There is no general low complexity algorithm for non-convex optimization problems and each of them can be solved via customized approaches, e.g., decomposition, duality [18] and approximation [19]. For numerical results, we simply solve (11) via exhaustive search over discretized

\(P_i\) values. We consider fixed routing parameters \(\alpha = 0.1, \beta = 0.1, \gamma = 0.5\) in Figure 8(b). Rates over links (1, 3, 2, 3, 3, 4) are measured and transmission powers over all links are optimized. To establish a realistic capacity term \(C_i(P_i)\), we run 200 test packet transmissions with actual radios, RouterStation Pros, over a wireless network emulator (the details of network emulation are presented in Section 6) and measure the capacity.

The transmission powers are selected from \([-98, -85] \text{ dBm}\) (with resolution 0.3 dBm). The transmission powers to each link are measured and the new power allocation is determined. Figure 9 shows the optimized power allocation results over time that are obtained from joint network inference and power control with quick convergence. For given power levels, the end-to-end throughput is defined as the expected sum rate of flows delivered without any link outage on paths between source-destination pairs. Enabled by network inference, the performance gain for outage probability and throughput is shown in Figure 10, where we compare two cases: (i)

\[ \text{minimize } \sum \alpha_{ij} P_{ij} \left( y_{ij} > C_i(P_i) \right) \]

subject to

\[ \sum \beta_i P_i \leq P_{\text{max}} \]

\[ E[y_{ij}] \leq C_i(P_i) \quad \forall i \]

variables \{P_i, \forall i\}. \quad (11)

For (11), the distribution of \(\{y_i, \forall i\}\) is obtained through either network inference or measurement, which depends on transmission powers \(P_i\) through link outage objectives and link capacity constraints. Meanwhile, the power \(P_i\) is optimized depending on \(P_{y_{ij}}\).

This problem has non-convex objective and constraint functions. There is no general low complexity algorithm for non-convex optimization problems and each of them can be solved via customized approaches, e.g., decomposition, duality [18] and approximation [19]. For numerical results, we simply solve (11) via exhaustive search over discretized

\(P_i\) values. We consider fixed routing parameters \(\alpha = 0.1, \beta = 0.1, \gamma = 0.5\) in Figure 8(b). Rates over links (1, 3, 2, 3, 3, 4) are measured and transmission powers over all links are optimized. To establish a realistic capacity term \(C_i(P_i)\), we run 200 test packet transmissions with actual radios, RouterStation Pros, over a wireless network emulator (the details of network emulation are presented in Section 6) and measure the capacity.

The transmission powers are selected from \([-98, -85] \text{ dBm}\) (with resolution 0.3 dBm). The transmission powers to each link are measured and the new power allocation is determined. Figure 9 shows the optimized power allocation results over time that are obtained from joint network inference and power control with quick convergence. For given power levels, the end-to-end throughput is defined as the expected sum rate of flows delivered without any link outage on paths between source-destination pairs. Enabled by network inference, the performance gain for outage probability and throughput is shown in Figure 10, where we compare two cases: (i)

\[ \text{minimize } \sum \alpha_{ij} P_{ij} \left( y_{ij} > C_i(P_i) \right) \]

subject to

\[ \sum \beta_i P_i \leq P_{\text{max}} \]

\[ E[y_{ij}] \leq C_i(P_i) \quad \forall i \]

variables \{P_i, \forall i\}. \quad (11)

For (11), the distribution of \(\{y_i, \forall i\}\) is obtained through either network inference or measurement, which depends on transmission powers \(P_i\) through link outage objectives and link capacity constraints. Meanwhile, the power \(P_i\) is optimized depending on \(P_{y_{ij}}\).

This problem has non-convex objective and constraint functions. There is no general low complexity algorithm for non-convex optimization problems and each of them can be solved via customized approaches, e.g., decomposition, duality [18] and approximation [19]. For numerical results, we simply solve (11) via exhaustive search over discretized

\(P_i\) values. We consider fixed routing parameters \(\alpha = 0.1, \beta = 0.1, \gamma = 0.5\) in Figure 8(b). Rates over links (1, 3, 2, 3, 3, 4) are measured and transmission powers over all links are optimized. To establish a realistic capacity term \(C_i(P_i)\), we run 200 test packet transmissions with actual radios, RouterStation Pros, over a wireless network emulator (the details of network emulation are presented in Section 6) and measure the capacity.

The transmission powers are selected from \([-98, -85] \text{ dBm}\) (with resolution 0.3 dBm). The transmission powers to each link are measured and the new power allocation is determined. Figure 9 shows the optimized power allocation results over time that are obtained from joint network inference and power control with quick convergence. For given power levels, the end-to-end throughput is defined as the expected sum rate of flows delivered without any link outage on paths between source-destination pairs. Enabled by network inference, the performance gain for outage probability and throughput is shown in Figure 10, where we compare two cases: (i)
inference and power control, and (ii) no inference and fixed power. We show the ratio between the performance of case (i) and the performance of case (ii). After convergence, the weighted sum outage probability is 13% smaller and the throughput is 16% larger than the case of equal power allocation among links (the default case without knowledge on network parameters).

### 5.2 Application 2 – Network Optimization with Adaptive Routing

Next, we consider another application that the link rates are adjusted by varying the routing matrix in adaptive routing. With a new routing matrix \( \hat{A} \), the new link rates are \( Y(\hat{A}) = \hat{A}X \). The network optimization problem can be formulated as follows.

\[
\begin{align*}
\text{minimize} & \quad \sum_i w_i \Pr[y_i(\hat{A}) > C_i] \\
\text{subject to} & \quad E[y_i(\hat{A})] \leq C_i, \forall i \\
\text{variables} & \quad \{\hat{A}\}.
\end{align*}
\]

For (12), \( \Pr[y_i(\hat{A})] \) is obtained through network inference results \( X \) and new routing decisions \( \hat{A} \). Therefore, network inference and optimization are carried out jointly through the iterative algorithm shown in Figure 1. Note that for the network in Figure 8(b), routing matrix \( \hat{A} \) can be determined by three parameters \( \alpha, \beta, \) and \( \gamma \). Given the non-convex objective and constraint functions, we solve (12) via exhaustive search over discretized parameter values.

Numerical evaluation is performed over the network topology in Figure 8(b) with fixed \( \alpha = 0.6, \beta = 0.5 \) and variable \( \gamma \in [0, 1] \) to be optimized with resolution 0.01. The average rates over links (1,2), (2,3) and (2,4) are measured. The transmission powers are fixed to generate link capacities 0.7, 0.8, 0.8, 0.8, 0.55 for links (1,2), (1,3), (2,3), (2,4), and (3,4), respectively. Link rate distributions are inferred and routing parameter is optimized jointly. Transmission on each link is 1Mbps. The weighted sum of outage probabilities is optimized iteratively with incoming link rate measurements and inferred link rates, and then the new routing matrix is determined.

Figure 11 shows how the flow splitting parameter \( \gamma \) is optimized over time. This optimal value is obtained (with quick convergence) from joint network inference and adaptive routing. The end-to-end throughput is defined as the expected sum rate of flows delivered between source-destination pairs. Enabled by network inference, the performance gains for outage probability and throughput are shown in Figure 12, where we compare two cases: (i) inference and power control, and (ii) no inference and fixed power.
ence and adaptive routing, and (ii) no inference and fixed routing. We show the ratio between the performance of case (i) and the performance of case (ii). After convergence, the weighted sum outage probability is 30% smaller and the throughput is 19% larger than the case of equal flow splitting (γ = 0.5).

6 Emulation Testbed Implementation of Joint Network Inference and Optimization

We implemented the joint network inference and optimization approach in a realistic wireless emulation testbed environment and then evaluated its performance with real radios.

6.1 Testbed

Due to dynamic network topology and time varying channels, it is very challenging to run different schemes under the same wireless environment and thus to compare them fairly. Since a scheme’s performance is affected by wireless environment, we cannot simply identify the best scheme by comparing their performances under wireless setup that we cannot repeat. In this emulation study, we adopt a radio frequency network emulator simulator tool, RFnest\textsuperscript{TM} ([15] (RFnest\textsuperscript{TM} is a COTS product previously developed by Intelligent Automation, Inc.), to overcome this challenge. RFnest\textsuperscript{TM} repeats tests for different schemes under the same wireless environment and provides a full system verification. Each radio performs real transmissions over a channel (controlled by RFnest\textsuperscript{TM} ) with previously recorded channel conditions.

To run multiple schemes under the same wireless environment, RFnest\textsuperscript{TM} first records wireless channel conditions and then replays the same measured channel conditions for all schemes. By doing so, RFnest\textsuperscript{TM} keeps the benefits of emulation, i.e., real radio transmission over real (but controlled) channels, and provides the same wireless environment for multiple schemes such that we can perform a fair and reliable comparison under realistic wireless network characteristics. We note that RFnest\textsuperscript{TM} goes beyond a channel emulator in the sense that multi-access and broadcast configuration are generated and multiple nodes with unique IDs can coexist in a network environment running the entire protocol stack, including all real protocols from the PHY to application layers. This capability allows reliable test of how the presented scheme interacts with the existing protocols at other network layers (rather than separating them in simulations).

In this emulation environment, we removed the radio antennas and connected the radios (RouterStation Pros) with RF cables over RFnest\textsuperscript{TM} that can digitally control channel attenuation. Then, real signals are sent over emulated channels with physical-layer signal interactions between radios. We control channel characteristics by changing node locations in an interactive GUI. This emulation capability fills the gap between software simulations and fixed testbed experiments. Software simulations are typically based on simplistic wireless propagation models (e.g., circular transmission/reception ranges) and often ignore radio hardware characteristics. On the other hand, experiments in fixed testbeds cannot be easily controlled to establish same or similar channel conditions and they are very hard to repeat because wireless channels are continuously changing. Instead, emulation capability of RFnest\textsuperscript{TM} uses real radios but can still control and repeat wireless channel conditions in a programmable testbed setting.

We implemented and tested the joint network inference and routing optimization in this realistic wireless emulation environment (enabled by RFnest\textsuperscript{TM} ) with real radio transmissions. As illustrated in Figure 13, our testbed platform consists of four main components: radio frequency network emulator simulator tool, RFnest\textsuperscript{TM}, configurable RF front-ends (RouterStation Pro from Ubiquiti), software simulators running higher-layer protocols on PC hosts, and digital switch. This testbed environment allows repeatable experimentation with actual radios under identical channel scenarios. We executed wireless tests at 2.462GHz and generated mobility by changing the locations of nodes through RFnest\textsuperscript{TM} GUI (shown in Figure 14) where the model imposes that the signal power decays as $d^{-\alpha}$ over distance $d$ with path loss coefficient $\alpha = 4$ such that different signal strengths are achieved at different locations.

6.2 Emulation Test Results

We consider joint network inference and adaptive routing with the same network setting as in Section 5. The transmit power is set to 8dBm and transmission rate is set to 1Mbps. Periodically, radios make test transmissions and link loss rates are computed. This link information is used to compute the link capacity that is fed to the optimization problem as the link constraint. Figure 14 shows the interactive GUI to set up the emulation network topology and collect network measurements. The emulated channel properties are shown in Table 6 for the initial network topology. The results on the
achieved end-to-end throughput (moving average) at node 4 are shown in Figure 15. It is clear that adaptive routing with inference can achieve a significantly better performance than static routing in the initial topology setting.

In case of dynamic network topology (and thus dynamic channel status), our algorithm can quickly infer flow rates and adapt routing with optimal flow splitting to achieve the best throughput. To test this capability, we move node 4 further away from others (at the time 160s in Fig. 15) and then back to the original position, and we observe how the throughput changes with node mobility. In all cases, Figure 15 shows that joint network inference and optimization provides significant throughput gains over static routing solutions ($\gamma = 0.5$). As shown in Figure 14, we also validate the performance with real-time video transmissions and observed the effects of throughput gains (video quality improvement) with adaptive routing based on network inference (since our goal is to improve throughput, we do not consider other video performance metrics).

7 CONCLUSION

We studied the problem of wireless network inference and optimization with low-complexity solutions. We formulated an information-geometric optimization framework to infer the distributions of end-to-end flows rates from link rate measurements while mitigating measurement noise. We showed how network inference can be decomposed to subproblems with lower complexity that are solved independently and can be reformulated to partially infer a subset of network flow rates. When retransmissions are used over unreliable wireless links, we extended our approach to jointly infer distributions of flow and link loss rates. We presented a systematic mechanism of using network inference results to optimize the network performance. For this purpose, we applied either power control or adaptive routing to minimize the overall outage probabilities in the network. This was shown to have similar effects as maximizing the throughput. Our results demonstrated significant outage and throughput improvements compared to static solutions with fixed network operation. We verified these results by implementing our joint network inference and optimization approach with real radios and RFnest™ and executing tests for adaptive and static routing under the same realistic wireless network emulation environment.

REFERENCES


Yalin E. Sagduyu (S’02–M’08–SM’15) received the B.S. degree from Bogazici University, Turkey, in Electrical and Electronics Engineering and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Maryland at College Park. Dr. Sagduyu is currently an Associate Director of Network and Security at Intelligent Automation Inc., Rockville MD. His research interests are in the areas of design and optimization of wireless networks, cognitive radio networks, network coding, information theory, security, game theory, social networks, and Big Data analytics. He authored more than 80 papers on network design, analysis and implementation. Dr. Sagduyu co-organized ACM MobiCom 2013 Workshop on Cognitive Radio Architectures for Broadband, IEEE CNS 2014 Workshop Cognitive Radio and Electromagnetic Spectrum Security, and IEEE ICNP 2014 Workshop on Cognitive Radio Architectures for Broadband. He served as the Track Co-Chair for “MAC and Cross-Layer Design Track” at the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2014. He has served as a TPC member of major IEEE and ACM conferences.

Yi Shi (S’02–M’08–SM’13) received his Ph.D. degree in computer engineering from Virginia Tech, Blacksburg, VA, in 2007. He is a Senior Research Scientist in Intelligent Automation Inc., Rockville MD. His current research focuses on algorithm design and optimization for wireless communication system, cognitive radio networks, MIMO, cooperative, sensor and ad hoc networks, and social networks. He co-organized four IEEE and ACM workshops and has been a TPC member of 60 major IEEE and ACM conferences. He authored one book, five book chapters, and more than 110 papers on network design and analysis. He was a recipient of IEEE Communications Surveys and Tutorials Exemplary Editor for the year of 2014, IEEE INFOCOM 2008 Best Paper Award, the only IEEE INFOCOM 2011 Best Paper Award Runner-Up, and ACM WUWNet 2014 Best Student Paper Award.

Anthony Fanous (S’08–M’14) received the B.Sc. degree in electrical engineering from Alexandria University, Egypt in 2007 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland at College Park in 2012 and 2013, respectively. His research interests include cognitive radio networks and cross-layer design of wireless networks with focus on cooperation and Network Coding. Dr. Fanous is currently a Senior Engineer at Qualcomm Inc., San Diego CA. He is a member of the Honor society of Phi Kappa Phi.
Jason H. Li (M’02) received the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park. He is currently the Senior Director of the Networks and Security Group with Intelligent Automation Inc. (IAI), Rockville, MD. Before joining IAI, he was a Researcher with Hughes Network Systems, Germantown, MD. He is the author of more than 40 publications in the area of networks, protocols, security, and multiagent systems. His research interests include computer networks, networks and systems security; cyber security analysis; network management and control; distributed systems; and intelligent software agents. Dr. Li is a member of the Association for Computing Machinery (ACM), the Advanced Computing Systems Association (USENIX), and the Armed Forces Communications and Electronics Association. He has served on numerous Technical Program Committees for major IEEE/ACM conferences on networks and security-related technologies.