Spring 2004 Inaugural Issue

stanford ECJ

an electrical engineering and computer science research journal

Faculty Reviewed
Groundbreaking Research

ieee.stanford.edu/ecj

STANFORD

SCHOOL OF ENGINEERING

A message from the Dean of Engineering

One of the great benefits for students at a research university like Stanford is the opportunity to participate in leading edge research. While not all students will take advantage of this opportunity, for those that do, the results can be magical. All too often, for example, an undergraduate engineering or science education can seem like an endless series of lectures, homework assignments and exams. There is much more to engineering and science than this. Working on the frontiers of a discipline by spending a summer in a faculty research group or by doing a special project with a faculty member can provide tremendous insight into what a career might be like. Sometimes these research experiences result in published work, in technical journals or in magazines like this EE/CS Research Journal. I hope you enjoy reading about the research projects of the students who have written for this magazine. Behind each of the articles is a story of an exciting opportunity to explore the frontiers of a field.

Jim Plummer, Dean
School of Engineering

Contents

The ECJ showcases the top and latest research being done at the union of electrical engineering and computer science disciplines by Stanford graduate and undergraduate students.

ECJ is a faculty reviewed technical journal available both online and in print, serving to create awareness across disciplines within the campus technical community, prepare current researchers for the publication and review process of larger journals, and inspire future research by Stanford community members.

the Staff

Sponsors

Estimation-Pruning (EP) Algorithm for Point-to-Point Travel Cost Minimization in a Non-FIFO Dynamic Network

Keyvan Mohajer, Almir Mutapcic, and Majid Emami

Abstract. This paper presents the estimation-pruning (EP) algorithm for finding the best path (with minimum cost) from a source to a destination in a dynamic network that does not necessarily obey the first-in-first-out (FIFO) property. The EP algorithm consists of two steps. The first step is the forward or the estimation step in which a bound on the traveling cost of each possible path is calculated. The second step is the backward or the pruning step in which the paths that are unlikely to produce the best route are eliminated. The resulting network is then expanded in time and is converted to a static network, which is used to find the best route.

Index Terms: dynamic network routing, estimation-pruning algorithm, Dijkstra algorithm, network optimization, FIFO property.

I. Introduction

THE problem of finding the best route from a source to a destination in a weighted network has received a great deal of attention in the past few decades [1], [2]. This problem arises in many contexts; for example, data packets traveling through a computer network and vehicles traveling on a road network would ideally choose the optimum path according to some predefined criteria. For the vehicle routing problem, commercially available tools already exist that allow a user to specify a source and a destination and receive the corresponding route [3]–[5]. These tools mainly focus on optimizing the travel time of the route. In some instances, however, the user may desire to optimize the travel cost based on other factors such as safety, distance, standard deviation, avoidance of speed traps, and other considerations. In fact the cost of each link in the network can be computed according to a weighted combination of the above factors specified by the user. The problem gets more complicated once we start dealing with a dynamic network (a network in which the link travel costs are time dependent [6]–[8]). Moreover, when the defined travel cost does not have the FIFO property (defined below), the situation gets significantly more difficult. This paper presents a novel algorithm (The Estimation-Pruning Algorithm) to find a cluster of routes in the network that have the best bound on their travel cost. This algorithm has very good running time characteristics.

The next section in this paper describes the nature of non-FIFO dynamic networks in detail and introduces the idea of time expansion of a dynamic network. The basic ideas of the estimation and the pruning steps in the EP algorithm are explained in sections III and IV respectively. Section V presents the EP algorithm itself. The running time analysis is described in section VI and section VII provides an extension to the basic EP algorithm of section V for more accurate results at the expense of longer processing time using a tuning parameter. Finally the simulation results are presented in section VIII.

The authors are Ph.D. candidates with the Department of Electrical Engineering, Stanford University, Stanford, CA. Email: {keyvan, almirm, memami}@stanford.edu

II. Non-FIFO Dynamic Networks and Time Expansion

Let $G = (V, E)$ be a directed network, where $V = 1, 2, \ldots, N$ is the set of nodes and $E = 1, 2, \ldots, M$ is the set of directed links. The cost of traveling from node i to node j in a dynamic network is a function of time and is represented as $C_{ij}(t)$. We will assume that $C_{ij}(t)$ is a discrete function of time and stays constant for periods of Δt. Furthermore, we will assume that there is at most one link from node i to node j, although this is not required and is only assumed for simplicity of this presentation. In addition, the time it takes to travel from node i to node j in a dynamic network is also a function of time and is represented by $T_{ij}(t)$. The parameter $T_{ij}(t)$ is not necessarily the same as $C_{ij}(t)$ since the traveling cost could be a function of traveling time and other factors.

Consider the simple example of Fig. 1. Assume that the user can leave node A between times t_1 and $t_2 = t_1 + 3\Delta t$ and would like to reach node C with the minimum total traveling cost.

Depending on when the user departs from node A, the cost of going from node A to node B could be $C_{AB}(t_1)$, $C_{AB}(t_1 + \Delta t)$, $C_{AB}(t_1 + 2\Delta t)$ or $C_{AB}(t_1 + 3\Delta t)$. Similarly, the cost of going from node A to node C depends on when the user departs from node A. The cost of going from node B to node C depends on when the user arrives at node B. For instance, If the user leaves node A at time $t_1 + \Delta t$, she will get to node B at time $t_1 + \Delta t + T_{AB}(t_1 + \Delta t)$ and the cost of going from B to C would be $C_{BC}(t_1 + \Delta t + T_{AB}(t_1 + \Delta t))$.

In the process of finding the best route with minimum cost, one should consider all possibilities. To do this, one can expand the dynamic network in time by replicating each node to convert it to a static network. Fig. 2 shows the corresponding time expanded network of Fig. 1.

In this case, node A has been replicated four times to account for the fact that the user can leave during four different time intervals. Note that in each case, the user can arrive at node C in two different time slots (depending on which path is chosen). Therefore, if the user wants to leave node C, the time-expanded version of Fig. 1 would have $4 \times 2 = 8$ different replications. The time expansion technique described in [9] increases the size of the

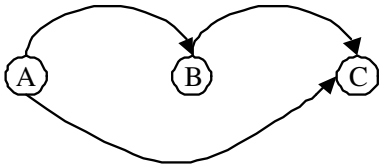


Fig. 1. Sample Network

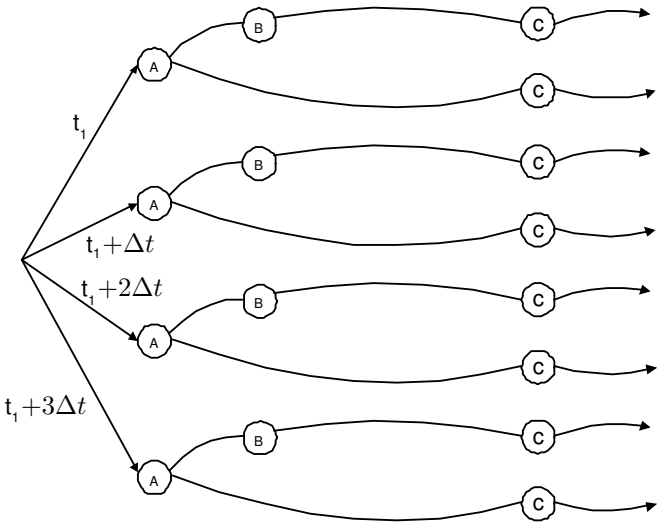


Fig. 2. Time expanded network

dynamic network by the number of available time intervals. As Δt becomes smaller and the number of time intervals becomes increasingly large, this time expansion becomes intractable.

Intelligent algorithms have been developed to deal with such problems [9]–[12]. Most of these techniques, however, use the assumption that the network has the FIFO property. This property states that if path 1 enters link L with a smaller total cost compared to the total cost of path 2, then path 1 also leaves link L with a smaller total cost. This assumption is reasonable if the traveling cost is the same as the traveling time (hence the name FIFO). As mentioned above, this is not generally the case. An example of a non-FIFO problem is that of searching for the safest route.

The goal of the EP algorithm is to estimate an upper bound and a lower bound on the total traveling cost of going from the source to the destination through all possible paths before the time expansion process (the estimation step), and then eliminate the paths that are unlikely to have the minimum cost (the pruning step). The output of the EP algorithm is a small number of candidate paths, which can be expanded in time in a tractable manner.

III. Basic Ideas of the Estimation (Forward) Step

We will define some new parameters as follows:

$Cf_{min}(i)$: The lower bound on the cost of traveling from source to node i

$Cf_{max}(i)$: The upper bound on the cost of traveling from source to node i

$T_{min}(i)$: The lower bound on the time of arrival at node i

$T_{max}(i)$: The upper bound on the time of arrival at node i.

Our goal in this step is to find the above parameters for all the nodes until we reach the destination node. Again, consider the simple example of Fig. 1. Assume that we already know $Cf_{min}(A)$ and $Cf_{max}(A)$. We also know that $T_{min}(A) = t_1$ and $T_{max}(A) = t_2 = t_1 + 3\Delta t$. We can now estimate the same parameters for node B as follows:

$$\begin{aligned} Cf_{min}(B) &= Cf_{min}(A)+ \\ &\quad \min\{C_{AB}(t_1^*) \mid T_{min}(A) < t_1^* < T_{max}(A)\} \\ Cf_{max}(B) &= Cf_{max}(A)+ \\ &\quad \max\{C_{AB}(t_2^*) \mid T_{min}(A) < t_2^* < T_{max}(A)\} \\ Tf_{min}(B) &= T_{min}(A)+ \\ &\quad \min\{T_{AB}(t_3^*) \mid T_{min}(A) < t_3^* < T_{max}(A)\} \\ Tf_{max}(B) &= T_{max}(A)+ \\ &\quad \max\{T_{AB}(t_4^*) \mid T_{min}(A) < t_4^* < T_{max}(A)\} \end{aligned} \quad (1)$$

These bounds on the costs and the arrival times may never be achieved, but they are absolute bounds and are never exceeded. This example suggests that, after initialization at the source node, one can proceed node to node toward the destination and calculate the cost and arrival time bounds on each node. If more than one branch enter a particular node, the minimum of the lower bounds and the maximum of the upper bounds are recorded (the latter is due to the non-FIFO property). This process continues until the destination node is reached. At this point, we can compute the bounds on the cost for entering the destination node through the last link. If k links enter the destination node at the end, we will have K bounds on the cost and arrival time for each link. We are now ready to traverse the network backward and prune the possible paths that have unpromising bounds on their traveling cost.

IV. Basic Ideas of the Pruning (Backward) Step

So far, in the forward step, we have been keeping the minimum of the lower bounds and the maximum of the upper bounds. In the backward step, we will keep the minimum of the lower bounds and the *minimum* of the upper bounds.

We shall define:

$Cb_{min}(i)$: The lower bound on the cost of traveling from node i to the destination.

$Cb_{max}(i)$: The upper bound on the cost of traveling from node i to the destination

These parameters can be updated from node to node in a similar way as in the forward step. Consider the situation in Fig. 3, where the destination node D can be reached from any of the K nodes.

For each node entering the destination node, we can calculate:

$$\begin{aligned} C_{min}^i &= Cf_{min}(i) + Cb_{min}(i) \\ C_{max}^i &= Cf_{max}(i) + Cb_{max}(i) \end{aligned}$$

Where, C_{min}^i is the lower bound on the cost of going from source to destination through node i, and C_{max}^i is the upper bound on the cost of going from source to destination through node i.

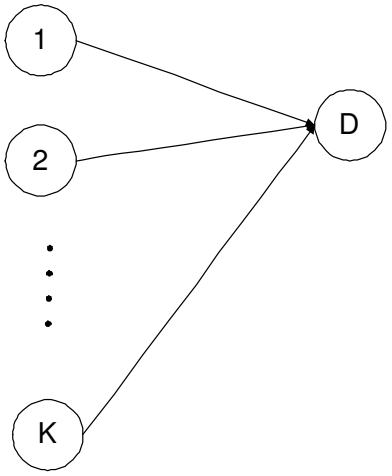


Fig. 3. K nodes entering destination

Fig. 4 shows a possible outcome of what these bounds may look like for $K = 5$ nodes entering the destination.

The function of the pruning step should now be clear. Compare the bounds of node 1 and node 4. Since $C^1_{max} < C^4_{min}$, it is always better to get to node D through node 1 as opposed to node 4. This is the best case when the intervals within the bounds of different nodes do not overlap. In the basic EP algorithm presented in the next section, the pruning step always chooses at most two nodes in each stage. These two nodes are the nodes corresponding to the smallest minimum and the smallest maximum (which may happen to correspond to the same node). All other nodes are discarded. In the example of Fig. 4, only nodes 1 and 2 are selected, and this ensures that the final route will have a cost bounded by C^1_{min} and C^2_{max}. Section VII extends this to a more generic version of the EP algorithm that allows selecting the best $2p$ nodes at each pruning stage. By default, $p = 1$.

V. The EP Algorithm

A. Definitions

$\mathcal{V}$: List of all the nodes in the network (provided)
$\mathcal{Q}$: List of all the nodes to be processed by the forward step
S: Source node
D: Destination node

$T_{min}(i)$: Lower bound on the time of arrival at node i
$T_{max}(i)$: Upper bound on the time of arrival at node i
$Cf_{min}(i)$: Lower bound on travel cost from S to node i
$Cf_{max}(i)$: Upper bound on travel cost from S to node i
$Cb_{min}(i)$: Lower bound on travel cost from node i to D
$Cb_{max}(i)$: Upper bound on travel cost from node i to D
C^i_{min}: Lower bound on cost from S to D through node i
C^i_{max}: Upper bound on cost from S to D through node i
$C_{ij}(t)$: Travel cost from node i to node j at time t
$T_{ij}(t)$: Travel time from node i to node j at time t

$\mathcal{A}(i)$: List of nodes that are directly linked to from node i
$\mathcal{F}(i)$: List of nodes entering i collected in the forward step
$\mathcal{B}(i)$: List of nodes from $\mathcal{A}(i)$ kept in the backward step

B. Initialization

We initialize some of the above variables as follows:

$\mathcal{Q} = \mathcal{V}$

foreach i *in* $\mathcal{V}$ **do**
  $T_{min}(i) = \infty$
  $T_{max}(i) = 0$
  $Cf_{min}(i) = \infty$
  $Cf_{max}(i) = 0$
  $Cb_{min}(i) = \infty$
  $Cb_{max}(i) = \infty$
end

$T_{min}(S) = t_1$ (entered by user)
$T_{max}(S) = t_2$ (entered by user)
$Cf_{min}(S) = 0$
$Cf_{max}(S) = 0$
$Cb_{min}(D) = 0$
$Cb_{max}(D) = 0$

$\mathcal{F}(i) = NULL$
$\mathcal{B}(i) = NULL$

C. Forward (Estimation) Step

while D *is in* $\mathcal{Q}$ **do**
  $i =$ node with the smallest $Cf_{min}(i)$ from $\mathcal{Q}$
  remove i from $\mathcal{Q}$
  if $i = D$ **then**
    break
  end
  foreach *node* j *in intersection of* $\mathcal{A}(i)$ *and* $\mathcal{Q}$ **do**
    add node i to $\mathcal{F}(j)$
    if $T_{min}(j) > T_{min}(i) + \min\{T_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$ **then**
      $T_{min}(j) = T_{min}(i) + \min\{T_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$
    end
    if $T_{max}(j) < T_{max}(i) + \max\{T_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$ **then**
      $T_{max}(j) = T_{max}(i) + \max\{T_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$
    end
    if $Cf_{min}(j) > Cf_{min}(i) + \min\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$ **then**
      $Cf_{min}(j) = Cf_{min}(i) + \min\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$
    end
    if $Cf_{max}(j) < Cf_{max}(i) + \max\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$ **then**
      $Cf_{max}(j) = Cf_{max}(i) + \max\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\}$
    end
  end
end

D. Backward (Pruning) Step

$\mathcal{L}_{current} = NULL$
$\mathcal{L}_{next} = D$
while $\mathcal{L}_{next} \neq NULL$ **do**
  $\mathcal{L}_{current} = \mathcal{L}_{next}$
  $\mathcal{L}_{next} = NULL$
  foreach *node* j *in* $\mathcal{L}_{current}$ **do**
    $C^j_{min} = \min_{i \epsilon \mathcal{F}(j)}\{Cf_{min}(i) + \min\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\} + Cb_{min}(j)\}$
    $MIN(j) = \arg\min_{i \epsilon \mathcal{F}(j)}\{Cf_{min}(i) + \min\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\} + Cb_{min}(j)\}$
    $C^j_{max} = \min_{i \epsilon \mathcal{F}(j)}\{Cf_{max}(i) + \max\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\} + Cb_{max}(j)\}$
    $MAX(j) = \arg\min_{i \epsilon \mathcal{F}(j)}\{Cf_{max}(i) + \max\{C_{ij}(t^*) \mid T_{min}(i) < t^* < T_{max}(i)\} + Cb_{max}(j)\}$
  end
  $n = \arg\min_{j \epsilon \mathcal{L}_{current}}\{C^j_{min}\}$
  $m = MIN(n)$
  if $Cb_{min}(m) > Cb_{min}(n) + \min\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$ **then**
    $Cb_{min}(m) = Cb_{min}(n) + \min\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$
  end
  if $Cb_{max}(m) > Cb_{max}(n) + \max\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$ **then**
    $Cb_{max}(m) = Cb_{max}(n) + \max\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$
  end
  add n to $\mathcal{B}(m)$
  if $m \neq S$ **then**
    add m to $\mathcal{L}_{next}$
  end
  $n = \arg\min_{j \epsilon \mathcal{L}_{current}}\{C^j_{max}\}$
  $m = MAX(n)$
  if $Cb_{min}(m) > Cb_{min}(n) + \min\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$ **then**
    $Cb_{min}(m) = Cb_{min}(n) + \min\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$
  end
  if $Cb_{max}(m) > Cb_{max}(n) + \max\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$ **then**
    $Cb_{max}(m) = Cb_{max}(n) + \max\{C_{mn}(t^*) \mid T_{min}(m) < t^* < T_{max}(m)\}$
  end
  if n *not already in* $\mathcal{B}(m)$ **then**
    if $m \neq S$ *AND* m *is not already in* $\mathcal{L}_{next}$ **then**
      add m to $\mathcal{L}_{next}$
    end
  end
end

E. Route Optimization

The best route in the pruned network can now be computed using the following recursive algorithm:

function Route Optimization
$opt\_route = NULL$
$opt\_cost = \infty$
call `Propagate(`*S, NULL*`)`
output opt_route and opt_cost

function Propagate($node$, $path$)
$path = path + node$
if $node = D$ **then**
  foreach $Tmin(S) \leq t \leq Tmax(S)$ **do**
    find the best cost path
  end
  if *lower cost is achieved* **then**
    update opt_route and opt_cost
  end
  return
end
else
  foreach n *in* $\mathcal{B}(node)$ **do**
    call `Propagate(`*n, path*`)`
  end
end

VI. Complexity Analysis

The forward step in the EP algorithm has the same complexity as the Dijkstra algorithm, $O(N^2)$ [13]. The number of pruning iterations is at least equal to the number of nodes on the best path from the source to the destination. In the worst case, when the number of nodes in $\mathcal{F}$ linearly depends on N (as opposed to a constant), the pruning step could take at most $O(N^2)$. However, in typical networks the pruning step converges a lot faster than $O(N^2)$.

VII. Extension to the Algorithm

The basic EP algorithm presented in the previous section may eliminate the most optimal route in the pruning step. An extension of the EP algorithm could take a parameter p as the input, and select the best $2p$ nodes in each stage of the backward step (the factor of 2 is for the min and max criteria). This parameter is a tuning parameter that introduces a trade-off between the running time and the performance of the algorithm.

VIII. Simulation Results

The EP algorithm discussed in the previous sections has been implemented in Java for the purpose of concept testing and comparison with other methods. The main performance benchmark in the following simulation results is the Dijkstra algorithm adopted to compute optimal costs for static approximations of a dynamic network. The first approximation uses minimum cost across all the time bins as the link cost and we refer to this implementation as MIN_{DA}. The second approximation uses the average cost across all the time bins as the link cost and we refer to it as AVG_{DA}. After running MIN_{DA} and AVG_{DA}, the actual dynamic cost of their output route is calculated by traversing the dynamic network along these routes. In addition, the absolute minimum cost route for the given dynamic network is computed by exhaustive search.

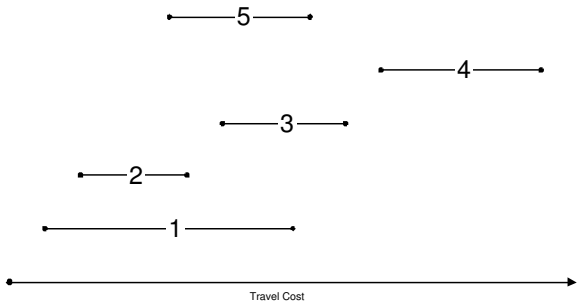


Fig. 4. Cost bound comparison for different paths

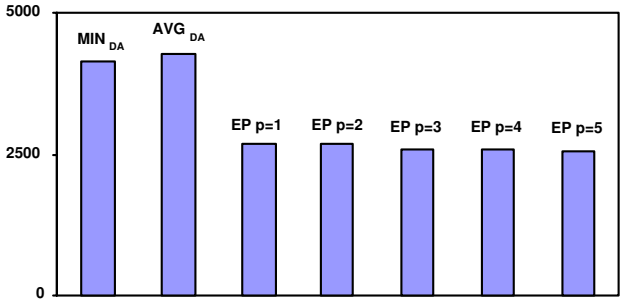


Fig. 5. Route Cost

All algorithms were tested and compared on randomly generated networks with different setups. The network variables are: number of nodes, number of links leaving each node, average travel time and cost on each link, standard deviation for time and cost of each link, number of time bins, and the neighborhood size. The neighborhood parameter was introduced to model the node clusters in the real road networks (*e.g.*, an urban area is heavily connected within itself, but different areas are sparsely connected between each other). Higher time and cost values were also introduced for certain time bins in order to simulate time characteristics in the real world (*e.g.*, rush hour conditions on the highways).

A numerical comparison was performed on a sample network with 1000 nodes, 3 links leaving each node, 100 node neighborhood, 24 time slots (hourly model), and uniformly distributed times and costs with some additional rush hour penalties at specified time bins. Obtained results are presented in Table I and Fig. 5.

The EP algorithm achieves significantly better results than the Dijkstra algorithms. For realistic road network of this simulation with rush hour traffic patterns, EP outperforms both Dijkstra approximations by 35-40%.

The EP routing cost in Table I is very close to the exhaustive search optimal routing cost even with pruning parameter $p = 1$ and as p is increased the EP cost rapidly converges to the optimal cost point.

TABLE I
SAMPLE RESULTS

Algorithm	Achieved Cost
MIN_{DA}	4135.58
AVG_{DA}	4290.27
EP ($p = 1$)	2680.96
EP ($p = 2$)	2680.96
EP ($p = 3$)	2597.58
EP ($p = 4$)	2597.58
EP ($p = 5$)	2559.41
EP ($p = 6$)	2559.41
EP ($p = 7$)	2559.41
Exhaustive Search	2559.41

IX. Conclusion

In this paper we have considered the problem of finding the best point-to-point route with minimum cost through a non-FIFO dynamic network. We introduced a novel algorithm (EP) that estimates the lower and upper bounds on all the paths through the network and uses these bounds to prune the paths with higher costs. The algorithm then finds the optimal route for the pruned subset of network paths.

The basic ideas of the EP algorithm are presented together with pseudo-code. The experimental results verified the value of this new algorithm. The future directions of research include performance evaluation of the EP algorithm on dynamic networks of various nature as well as investigation of the real-time and distributed implementations.

References

[1] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
[2] N. Christofides, "Vehicle routing," in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, Eds. New York: John Wiley and Sons, 1985.
[3] W. H. Randolph. (2002, Feb.) Vehicle Routing Software Survey, paper and online edition of OR/MS Today. [Online]. Available: http://www.lionhrtpub.com/orms/surveys/Vehicle_Routing/vrss.html
[4] University of Maryland, College Park, MD. (2003) The Dynasmart website. [Online]. Available: http://www.dynasmart.com/
[5] Massachusetts Institute of Technology, Cambridge, MA. (2003) The DynaMIT website. [Online]. Available: http://mit.edu/its/
[6] M. Gendreau and J.-Y. Potvin, "Dynamic vehicle routing and dispatching," in *Fleet Management and Logistics*, T. Crainic and G. Laporte, Eds. Kluwer, 1998, pp. 115–126.
[7] H. N. Psaraftis, "Dynamic vehicle routing: Status and prospects," *Annals of Operations Research*, vol. 61, pp. 143–164, 1995.
[8] W. Powell, P. Jaillet, and A. Odoni, "Stochastic and dynamic networks and routing," in *Handbooks in Operations Research and Management Science, Network Routing*, M. Ball, T. Magnanti, C. Monma, and G. Nemhauser, Eds. Amsterdam: Elsevier, 1995, vol. 8, pp. 141–296.
[9] I. Chabini and S. Lan, "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," *IEEE Trans. Intell. Transport. Syst.*, vol. 3, pp. 60–74, Mar. 2002.
[10] I. Chabini and S. Gao, "Optimal routing policy problems in stochastic time-dependent networks. I. Framework and taxonomy," in *Proc. The IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, pp. 549–554.
[11] ——, "Optimal routing policy problems in stochastic time-dependent networks. II. Exact and approximation algorithms," in *Proc. The IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, pp. 555–559.
[12] D. J. Bertsimas and D. Simchi-Levi, "A new generation of vehicle routing research: Robust algorithms, addressing uncertainty," *Operations Research*, vol. 44, pp. 286–304, 1996.
[13] T. H. Cormen (Editor), C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, Sept. 2001.

Real-Time Lens Distortion Correction: 3D Video Graphics Cards Are Good for More than Games

Michael R. Bax

Abstract. Optical lens systems suffer from non-linear radial distortion. Image-based applications require distortion compensation for the accurate location, registration and measurement of image features, and a real-time capability is desirable for interactive systems. A texture-mapping graphics accelerator can use corresponding meshes of distorted and undistorted nodes along with the dynamic distorted image to render high-quality distortion-corrected images at video framerates. Mesh generation, an error analysis, and performance results are presented. The proposed polar-based method is shown to have both more accuracy than a conventional grid-based approach and greater speed than the traditional method of using the CPU to transform each pixel individually.

Index Terms: lens distortion, correction, compensation, real-time, texture mapping.

I. Introduction

THE optical lens systems used in imaging equipment ranging from film cameras to endoscopes suffer from distortion artefacts, which detract from the quality of the images produced (see Fig. 1). In applications such as photogrammetry, computer vision, and medical imaging, the determination of and compensation for distortion is required to enable accurate location, measurement and registration of features in images [1]–[5].

While distortion correction may be applied offline in many cases, a real-time capability is desirable for systems that must interact with the environment or with a user in real time. It is possible to implement radial lens distortion correction at video framerates on current workstation central processing units (CPU's) [6], but correcting full-resolution colour video images in real time can require more processor power than is available.

In cases where the raw CPU power is not exceeded, the processor may nevertheless be concurrently required for other important real-time tasks such as navigation or tracking, making it impractical to spare much CPU time; distortion correction should ideally be performed with minimal impact on the CPU load.

II. Lens Distortion

Infinite series are needed to fully model non-linear lens distortion [1], [2], but in practice it is normally sufficient to model only the dominant radial distortion (also known as barrel or pincushion distortion) using a single parameter, κ_1 [3]. This is the model used here; it is assumed that the value of κ_1 is known.

The radial lens distortion is modelled as

$$r_u = r_d(1 + \kappa_1 r_d^2), \tag{1}$$

where r_u is the correct, undistorted radial distance to a point from the optical centre of the image, and r_d is the distorted radius. This relationship is illustrated in Fig. 2.

Given r_u, calculation of r_d requires the solution of this cubic equation. Applying Viète's substitution $r_d = w - \frac{1}{3\kappa_1 w}$ leads to

The author is with the Image Guidance Laboratories at Stanford. Email: mbax@stanford.edu.

(a) Distorted image

(b) Undistorted image

Fig. 1. Radial lens distortion: before and after.

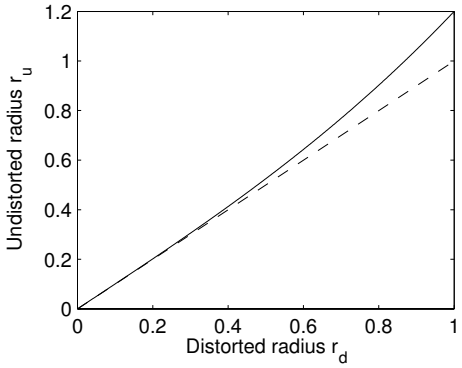


Fig. 2. The radial lens distortion model.

the quadratic form

$$\left(w^3\right)^2 - \frac{r_u}{\kappa_1} w^3 - \frac{1}{27\kappa_1^3} = 0 \tag{2}$$

and its solution

$$w = \sqrt[3]{\frac{r_u}{2\kappa_1} \pm \sqrt{\frac{r_u^2}{4\kappa_1^2} + \frac{1}{27\kappa_1^3}}}. \tag{3}$$

The value of r_d is found by substituting either of these roots for w.

III. Distortion Correction

A. Individual pixel resampling

The direct approach to reversing lens distortion is to resample the image using the inverse mapping. Simply choosing the closest integral pixel location for each point (nearest neighbour) results in poor visual quality, but bilinear interpolation between the four adjacent pixel values gives a satisfactory result at a reasonable additional computational cost.

Let $f(\boldsymbol{u})$ and $g(\boldsymbol{u})$ denote the value of the pixel at the location $\boldsymbol{u}$ in the input and output images respectively, and let $\boldsymbol{u}_0$ be the optical image centre. The sampling position in the input image is therefore

$$\boldsymbol{u}_f = \boldsymbol{u}_0 + \frac{\boldsymbol{u}_g - \boldsymbol{u}_0}{1 + \kappa_1 r_d^2}. \tag{4}$$

Let $\boldsymbol{\Delta} = (\Delta_1, \Delta_2)$ be the fractional part of the sampling position, and let $\boldsymbol{u}_{f_{xy}} = \boldsymbol{u}_{f_{00}} + (x, y)$ denote an adjacent pixel location. Bilinear interpolation gives

$$\begin{aligned} \alpha &= \boldsymbol{u}_{f_{00}} + \Delta_2 \left(\boldsymbol{u}_{f_{01}} - \boldsymbol{u}_{f_{00}}\right) \\ \beta &= \boldsymbol{u}_{f_{10}} + \Delta_2 \left(\boldsymbol{u}_{f_{11}} - \boldsymbol{u}_{f_{10}}\right) \\ g(\boldsymbol{u}_g) &= \alpha + \Delta_1 \left(\beta - \alpha\right). \end{aligned} \tag{5}$$

Pixel-by-pixel resampling at video framerates is accurate but CPU-intensive. Even a fast processor has little time for other tasks, and if the image size and/or colour-depth are too large it will fail to keep up. Faster approaches such as simple linear compensation using image scaling [6] suffer from significant residual error. Dedicated, special-purpose hardware solves this problem [8], [9] but is typically inflexible and relatively expensive.

B. Local affine transformation with texture mapping

Inexpensive yet powerful PC 3D video accelerators, or graphics processing units (GPU's), have given the workstations the capability of rendering hundreds of millions of texture-mapped pixels per second; GPU-based image-processing algorithms can free the CPU for other tasks.

The GPU accepts one or more images (known as textures), a list of 3D triangle vertex co-ordinates $\boldsymbol{x} = (x, y, z)$, and a corresponding list of 2D texture co-ordinates $\boldsymbol{u} = (u, v)$. The rendering process computes the displayed position of each vertex, then interpolates between the corresponding texture co-ordinates to determine the texture location to sample in order to fill the pixels in each triangle.

The goal of texture-mapping in a GPU is to warp images fast — but it can also be used to *unwarp* distorted images!

1) Cartesian-based vertex mapping: The straightforward approach to exploiting the GPU is to tessellate the distorted input image into a set of triangles in a uniformly-spaced grid [10], [11], as shown in Fig. 3(a).

The original grid intersection positions become the list of texture co-ordinates $\boldsymbol{u}_i$, and the distortion-corrected positions form a list of triangle vertex positions $\boldsymbol{x}_i$ for rendering. Following the one-time calculation of these lists, a stream of distorted images is fed to the GPU as a sequence of replacement textures.

The trade-off here is accuracy for speed. The vertices are correctly located, but the interior of each triangle undergoes an affine transformation. As a result, the interpolated sampling positions introduce distortion-correction errors in the *positions* of input image content within the corrected output image.

The worst-case error may however be reduced by orienting the triangles in each quadrant to minimise their lengths in the radial direction, as shown in Fig. 3(b).

2) Polar-based vertex mapping: The lens distortion model is parametrically polar. Tessellating the distorted image on a polar basis is a natural fit, facilitating analysis and permitting straightforward optimisation of the vertex locations: this yields a more efficient use of triangles and avoids localised peaks in the error field. It also allows the triangles to be more equilateral in shape, which for a given triangle area reduces the affine transformation error.

The approach proposed here is to divide the input image into concentric annuli and apply a radially affine transform to approximate the lens distortion correction function. In order to tessellate the annuli into triangles for rendering, the perimeters of the rings are broken up into piecewise-linear segments.

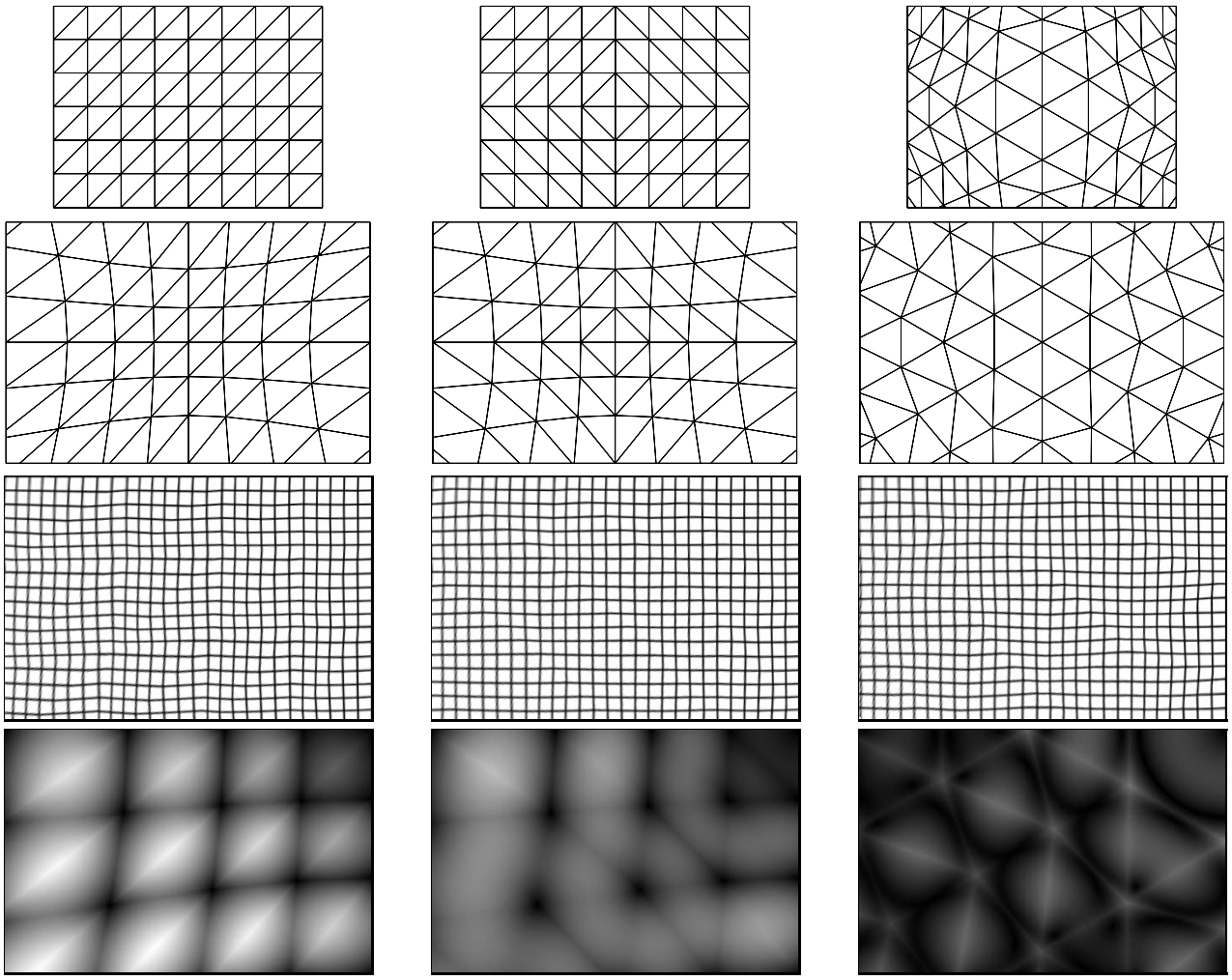
If the number of segments (and hence triangles) is constant for each perimeter, the maximum dimension of the triangles grows in size as the radial distance increases. In order to offset this, additional triangles must be introduced. In this method the additional triangles are introduced along 6 equispaced radius vectors as shown in Fig. 3(c), reducing the deformation from equilateral form of the propagated triangles.

As a result, the ith complete annulus from the image centre adds $6i$ vertices and $6(2i-1)$ triangles; the first n complete annuli contain $3n(n+1)+1$ vertices describing $6n^2$ triangles. Note however that these numbers apply only to circular images such as those from an endoscope. In general the borders of a rectangular image will clip much of the outer annuli, and triangles falling outside the borders of the output image need not be rendered.

The remaining element is the determination of the width of a given annulus. With a Cartesian-based vertex approach it is simple to determine the number of triangles; the polar-based case on the other hand lends itself more directly to minimising the number of triangles subject to a given goal of maximum distortion-correction error. This is achieved if each annulus has the greatest width that does not cause the intra-annulus error to exceed the specified limit.

3) Analysis: In order to maximise the width of a given annulus without causing its internal distortion-correction error to exceed the desired threshold when undergoing affine transformation, the maximum error within its boundaries must be known.

Consider the ith annulus, with its interior perimeter at the radius r_{d_i} and of radial width w_{d_i} in the lens-distorted input



(a) Simple grid tessellation (b) Improved grid tessellation (c) Polar tessellation

Fig. 3. From top to bottom: input image texture co-ordinates; output image vertex co-ordinates; lower-left quadrant of a distortion-corrected output image; lower-left quadrant of the output image pixel error (black is zero). The input image was divided into a similar number of triangles in each case.

image. A point within this annulus has radius

$$r_d(\alpha) = r_{d_i} + \alpha w_{d_i} \tag{6}$$

for some $\alpha \in [0, 1)$. The correct distortion-free radius for this point in the output image is therefore

$$r_u(\alpha) = (r_{d_i} + \alpha w_{d_i}) \left(1 + \kappa_1 (r_{d_i} + \alpha w_{d_i})^2\right), \tag{7}$$

but affine transformation of this point results in the actual output image radius

$$\begin{aligned}\tilde{r}_u(\alpha) = {} & (1-\alpha) r_{d_i} \left(1 + \kappa_1 {r_{d_i}}^2\right) \\ & + \alpha (r_{d_i} + w_{d_i}) \left(1 + \kappa_1 (r_{d_i} + w_{d_i})^2\right).\end{aligned} \tag{8}$$

The error is

$$\begin{aligned}e(\alpha) &= \tilde{r}_u(\alpha) - r_u(\alpha) \\ &= \kappa_1 {w_{d_i}}^2 \alpha (1-\alpha) (\alpha w_{d_i} + 3 r_{d_i} + w_{d_i}),\end{aligned} \tag{9}$$

which reaches its maximum where

$$\begin{aligned}\frac{d}{d\alpha} e(\alpha) &= \kappa_1 {w_{d_i}}^2 \left(3 r_{d_i} + w_{d_i} - 6 r_{d_i} \alpha - 3 w_{d_i} \alpha^2\right) \\ &= 0,\end{aligned} \tag{10}$$

giving

$$\alpha_{e_{\max}} = \frac{2s - 6 r_{d_i}}{6 w_{d_i}}, \quad s = \sqrt{9 r_{d_i}^2 + 9 w_{d_i} r_{d_i} + 3 w_{d_i}^2}. \tag{11}$$

Substituting $\alpha_{e_{\max}}$ into (9) gives

$$e_{\max} = \frac{(2s + 3 r_{d_i}) (s - 3 r_{d_i})^2 \kappa_1}{27}. \tag{12}$$

This does not have a neat closed-form solution for w_{d_i} given r_{d_i} and $e_{\max}$, but it is straightforward to solve numerically.

Unfortunately, the annulus is not transformed monolithically; if it were simply scaled radially to match the lens-distorted inner and outer perimeters in the output image, the result in (12) would be the only error. The annulus is however tessellated into triangles for rendering, each of which undergoes individual affine transformation.

Non-radially aligned lines within the triangles are mapped to straight lines in the output image; exact distortion compensation would transform them to curves. This introduces another error component with approximately the same range as e_α. Both kinds

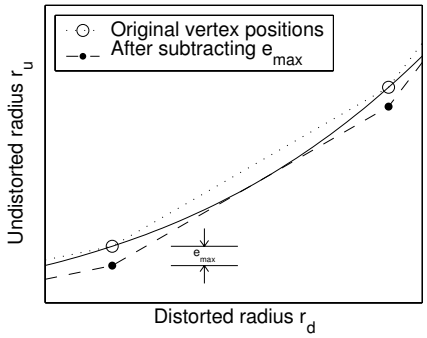


Fig. 4. Vertex position adjustment: the dotted line shows the initial affine approximation of the non-linear curve, and the dashed line shows the approximation after reducing the radii of the vertices by $e_{\max}$ to cancel tessellation error.

of error are centrifugal, increasing the radii of points in the output image beyond their corrected values.

These two sources of error have approximately the same range. The additional error introduced by triangular tessellation may therefore be counterbalanced by moving the previously-computed output image vertices a distance of $e_{\max}$ towards the image centre, as illustrated in Fig. 4. This in effect negates the range of e_α so that it tends to cancel the tessellation error component.

IV. Implementation

A C framework using the Microsoft DirectX 9.0 API was written to compare lens distortion correction using:

- individual pixel resampling by the CPU
- texture-mapped affine transformation by the GPU.

Both methods used bilinear interpolation. A look-up table was used to optimise CPU-based resampling. Ping-pong frame buffers in 24-bit packed RGB format were used to simulate an incoming colour video stream of 640×480 pixels (the full scanline resolution of an NTSC signal, reformatted for square pixels on a display with a 4:3 aspect-ratio). The display depth was set to 32-bit XRGB format, and a pair of texture maps were used to allow the display to use one while the other was updated with incoming video data.

The value of κ_1 was set to 0.2 (scaled for a normalised maximum inscribed radius of 1 in the input image). This value is representative of endoscope telescopes such as the 0° Karl Storz 50200 A.

The resolution of the output image was set to 866×574. This is the largest rectangular region that is entirely filled when a 640×480 image is distortion-corrected using $\kappa_1 = 0.2$ with no loss of information (1:1 sampling of the centre of the input image and oversampling elsewhere).

Three increasingly demanding targets for $e_{\max}$ were chosen: a maximum desired error of 1, 0.1, and 0.01 pixels respectively. A triangle mesh was designed using the analysis in section III-B.3 and those triangles not visible in the output image were culled. The remaining triangles were totalled and Cartesian-based meshes were selected to match the number of output image triangles as closely as possible while constraining the row height to column width ratio in order to obtain approximately square cells. The grids were divided into triangles in two ways: using first the simpler approach with triangles of constant orientation, and second the improved orientation of each diagonal to maximise its angle of intersection with the radius vector.

The pixel error at each output image pixel was determined by measuring the deviation of the actual sampling position from its correct ideal location in the input image. Real-time performance measurements were made on a 900 MHz Intel Pentium III computer with a 32 MB NVIDIA GeForce2 Go GPU and 512 MB of system RAM, running Microsoft Windows XP. Each test was run for 40 seconds; the data from the first 10 seconds was discarded to allow performance stabilisation. Elapsed times were recorded using a high-resolution hardware counter with sub-millisecond accuracy.

V. Results

As expected the error-free CPU-based method of individually resampling each pixel was too slow to keep up with the NTSC video rate of 30 frames per second when processing this amount of data (see Table I). The real-time correction of radial lens distortion in, and the subsequent display of, the equivalent of a 24-bit colour NTSC video stream demanded more power than was available from the CPU.

All the texture-mapping methods comfortably exceeded the video threshold of 30 frames per second. The optimum trade-off of pixel error against triangle count appears to be in the region of $e_{\max} = 0.1$; at the lower tolerance of $e_{\max} = 0.01$ pixels the framerate began to fall.

In each group of meshes with similar numbers of triangles, the Cartesian-based vertex mapping with improved triangle orientation lowered maximum pixel error in the input image by approximately 20% compared to a simple grid.

Polar-based vertex mapping performed substantially better than either of the other mapping methods, beating the simple and improved methods by about 50% and 40% respectively. The largest measured error was within 20% of the design $e_{\max}$ in each case, improving significantly for higher values of $e_{\max}$.

VI. Conclusion

If using a simple Cartesian-based tessellation, a useful improvement in accuracy can be easily achieved through a simple layout optimisation. If still greater accuracy is required, polar-based tessellation can more than double the improvement made by the Cartesian-based optimisation.

The GPU tested here is not a recent model, yet the texture-mapping methods described in this paper easily surpass the video threshold of 30 frames per second. This demonstrates a real-time performance capability on inexpensive, commodity hardware.

Newer GPU's feature programmable pixel shaders, which could be used to implement a real-time version of the per-pixel lens distortion compensation algorithm used on the CPU. Although there would be no pixel error with this approach, it is not expected to be as fast as conventional texture-mapping since it cannot be optimised and implemented directly at the hardware level in a fixed-function pipeline. In addition, pixel shader programming imposes an additional layer of complexity and the API's have yet to stabilise.

An efficient multi-threaded implementation of the pixel resampling algorithm running on a fast dual-CPU workstation can be

TABLE I

PERFORMANCE RESULTS FOR THE DISTORTION-CORRECTION METHODS DISCUSSED IN SECTION III. THE CARTESIAN-BASED METHOD IS LISTED FIRST WITHOUT, AND THEN WITH, QUADRANT-BASED HYPOTENUSE ORIENTATION; THE METHODS ARE GROUPED BY TRIANGLE COUNT.

Lens distortion correction method	Design $e_{\max}$ [pixels]	Number of triangles	Maximum error [pixels]	RMS error [pixels]	Time [ms]	Frames per second
Individual pixel resampling	—	—	0	0	168	6
Cartesian-based vertex mapping (simple)	—	258	1.888	0.879	26	39
Cartesian-based vertex mapping (improved)	—	256	1.495	0.729	26	39
Polar-based vertex mapping	1.00	254	0.902	0.351	26	39
Cartesian-based vertex mapping (simple)	—	2026	0.231	0.106	25	39
Cartesian-based vertex mapping (improved)	—	2020	0.188	0.088	25	39
Polar-based vertex mapping	0.10	2052	0.105	0.036	25	39
Cartesian-based vertex mapping (simple)	—	19494	0.024	0.011	28	36
Cartesian-based vertex mapping (improved)	—	19472	0.019	0.009	28	36
Polar-based vertex mapping	0.01	19452	0.012	0.004	28	36

expected to exceed the 30 frames-per-second threshold; this would however dominate the available CPU time. This is undesirable in environments where the CPU is simultaneously required for other tasks.

The bus between the display subsystem and the rest of the computer system is potentially a secondary bottleneck; since lens distortion generally compresses the periphery of an image, transmitting the smaller distorted image over this bus is more efficient. Although doing so translates to a 38% reduction in bandwidth for NTSC colour video images with $\kappa_1 = 0.2$, AGP bandwidth is an order of magnitude greater; this is only likely to be an issue for very large or highly-distorted images.

There are significant additional potential benefits of using the GPU texture-mapping technique if further manipulation of the distortion-corrected image is required. Once the input image has been uploaded into texture memory, the power of the GPU is available for additional rotation, scaling, translation, masking, blending and other operations with very little to no additional load. In applications such as augmented reality, virtual endoscopy, and other areas where the GPU either already is or can be used to project or composite images from multiple sources, lens distortion correction comes as a virtually resource-free bonus.

References

[1] C. C. Slama, C. Theurer, and S. W. Henriksen, Eds., *Manual of Photogrammetry*, 4th ed. Amer. Soc. Photogrammetry and Remote Sensing, Jan. 1980.

[2] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 10, pp. 965–980, Oct. 1992.

[3] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics Automat.*, vol. 3, no. 4, pp. 323–344, Aug. 1987.

[4] W. E. Smith, N. Vakil, and S. A. Maislin, "Correction of distortion in endoscope images," *IEEE Trans. Med. Imag.*, vol. 11, no. 1, pp. 117–122, Mar. 1992.

[5] H. Haneishi, Y. Yagihashi, and Y. Miyake, "A new method for distortion correction of electronic endoscope images," *IEEE Trans. Med. Imag.*, vol. 14, no. 3, pp. 548–555, Sept. 1995.

[6] R. Shahidi, M. R. Bax, C. R. Maurer, Jr., J. A. Johnson, E. P. Wilkinson, B. Wang, J. B. West, M. J. Citardi, K. H. Manwaring, and R. Khadem, "Implementation, calibration and accuracy testing of an image-enhanced endoscopy system," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 1524–1535, Dec. 2002.

[7] M. R. Bax, R. Khadem, J. A. Johnson, E. P. Wilkinson, and R. Shahidi, "Image-enhanced endoscopy calibration and image registration accuracy testing," in *Proc. SPIE Conf. Med. Imag.: Visualization, Image-Guided Procedures, and Display*, S. K. Mun, Ed., vol. 4681, May 2002, pp. 119–126.

[8] F. M. Waltz, "Implementation of real-time perspective correction," in *Proc. SPIE Conf. Automated Inspection and High Speed Vision Architectures*, vol. 849, Cambridge, Massachusetts, USA, Nov. 1988, pp. 179–83.

[9] A. G. J. Nijmeijer, M. A. Boer, C. H. Slump, M. M. Samson, M. J. Bentum, G. J. Laanstra, H. Snijders, J. Smit, and O. E. Herrmann, "Correction of lens-distortion for real-time image processing systems," in *Proc. IEEE Workshop on VLSI Signal Processing*, L. D. J. Eggermont, P. Dewilde, E. Deprettere, and J. Van Meerbergen, Eds., Veldhoven, the Netherlands, Oct. 1993, pp. 316–324.

[10] D. A. Butler and P. K. Pierson, "A distortion-correction scheme for industrial machine-vision applications," *IEEE Trans. Robotics Automat.*, vol. 7, no. 4, pp. 546–551, Aug. 1991.

[11] J. C. A. Fernandes, M. J. O. Ferreira, J. A. B. C. Neves, and C. A. C. Couto, "Fast correction of lens distortion for image applications," in *Proc. IEEE Int. Symp. Industrial Electronics*, vol. 2, July 1997, pp. 708–712.

Using Self-Consistent Naive-Bayes to Detect Masquerades

Kwong H. Yung

Abstract. To gain access to account privileges, an intruder masquerades as the proper account user. This paper proposes a new strategy for detecting masquerades in a multiuser system. To detect masquerading sessions, one profile of command usage is built from the sessions of the proper user, and a second profile is built from the sessions of the remaining known users. The sequence of the commands in the sessions is reduced to a histogram of commands, and the naive-Bayes classifier is used to decide the identity of new incoming sessions. The standard naive-Bayes classifier is extended to take advantage of information from new unidentified sessions. On the basis of the current profiles, a newly presented session is first assigned a probability of being a masquerading session, and then the profiles are updated to reflect the new session. As prescribed by the expectation-maximization algorithm, this procedure is iterated until the probabilities and the profiles are consistent. Experiments on a standard artificial dataset demonstrate that this self-consistent naive-Bayes classifier beats the previous best-performing detector and reduces the missing-alarm rate by 40%.

Index Terms: naive-Bayes classifier, self-consistent update, expectation-maximization algorithm, semisupervised learning; masquerade detection, anomaly detection, intrusion detection.

I. Introduction

THIS paper presents a simple technique for identifying masquerading sessions in a multiuser system. Profiles of proper and intruding behavior are built on the sessions of known users. As new, unidentified sessions are presented, the profiles are adjusted to reflect the credibility of these new sessions. Based on the expectation-maximization algorithm, the proposed *self-consistent* naive-Bayes classifier markedly improves detection rates.

A. Motivation

Unauthorized users behave differently from proper users. To detect intruders, behaviors of proper users are recorded and deviant behaviors are flagged. Typically, each user account is assigned to a single proper user with a specified role. Hence, future sessions under that user account can be compared against the recorded profile of the proper user. This approach to detecting masquerades is called anomaly detection.

More generally, intrusion detection is amenable to two complementary approaches, misuse detection and anomaly detection. Misuse detection trains on examples of illicit behavior; anomaly detection trains on examples of proper behavior. Because examples of intrusions are generally rare and novel attacks cannot be anticipated, anomaly detection is typically more flexible. In the context of masquerade detection, anomaly detection is especially appropriate because each user account is created with an assigned proper user, whose behavior must not deviate far from the designated role.

B. Previous Approaches

The reference dataset used in this study has been studied by many researchers. Schonlau and his colleagues [8] presented the dataset in full detail and reviewed six techniques used to analyze the dataset. Maxion and Townsend [6] later achieved better results by using the naive-Bayes classifier with updating, which serves as the basis for comparison. Yung [10] markedly improved detection rates but relied on confirmed updates. This paper also substantially improves over [6] and achieves results competitive with [10].

Both Maxion and Townsend [6] and Schonlau and his colleagues [8] tried to correct for the nonstationarity of user behavior by updating the user profiles. In [8], detectors with updating had fewer false alarms and generally outperformed the versions without updating. In [6], the naive-Bayes classifier with updating performed uniformly better than the naive-Bayes classifier without updating, by a wide margin.

The results of [6] and [8] demonstrate clearly that updating the detector produced significant improvements by lowering the number of false alarms. As new sessions are presented to the detector, the detector must classify the new sessions and also update the profile with these new sessions. In both [6] and [8], the detector considers each new session separately in sequence and must decide immediately how to classify the new session. Once the detector decides that the new session is proper, the detector then updates the profile of the proper user to include the new session, as if the new session had been part of the training set.

C. Early Limitations

There are at least two limitations to the updating scheme used in [6] and [8]. Principally, the detector must make a concrete decision whether to update the proper profile with the new session. In many cases, the judgment of a session is not always clear. Yet the detector is forced to make a binary decision before the detector can continue, because future decisions depend on the decision for the current case.

Furthermore, all future decisions depend on the past decisions, but the converse does not hold. In principle, the detector can defer the decision on the current session until additional new sessions from a later stage are studied. Yet in both [6] and [8], the detector is forced to make an immediate, greedy decision

for each new session, without considering the information from future cases. Scores for previous sessions cannot be revised as new sessions are encountered. Hindsight cannot improve the detector's performance on earlier sessions, because backtracking is not allowed.

D. New Strategy

Without a concrete optimality criterion, there is no basis on which to judge the performance of the greedy sequential strategy. In general, however, the best greedy strategy is not always the best strategy overall. By ignoring strategies that involve backtracking, the greedy search may fail to make decisions most consistent with the entire dataset.

This paper presents an alternative strategy for identifying masquerading sessions. To start, an optimality criterion is defined by specifying a concrete objective function. Moreover, each new session is assigned a score indicating how likely that session is a masquerading session. Instead of a binary decision, the detector assigns to the new session a probability between 0 and 1. This new session is then used to update the detector in a nonbinary fashion. The technique presented in this paper is an extension of the naive Bayes classifier used in [6] and will be called the *self-consistent naive-Bayes* classifier.

E. Outline of Paper

Following this introduction, Section II provides a brief background on the standard naive-Bayes classifier used for detecting masquerading sessions. Section III then explains the EM-algorithm used in the self-consistent naive-Bayes classifier. Section IV presents results of experiments on a standard artificial dataset. Section V discusses the advantages of the self-consistent naive-Bayes classifier as well as potential for future work. Finally, Section VI summarizes this paper's main conclusions.

II. Background

The self-consistent naive-Bayes classifier is an extension of the usual naive-Bayes classifier. Basically, the EM-algorithm is used to assign probabilities to new unidentified sessions. By incorporating a new session in a nonbinary fashion, the strategy for updating the profile is no longer greedy but rather optimizes the log-likelihood function of an underlying probabilistic model. The details the probabilistic model are explained in Section III. There the formalism of the EM-algorithm is used to generate a procedure for maximizing the log-likelihood in the presence of incomplete data.

Although the self-consistent naive-Bayes classifier is a significant step beyond the simple naive-Bayes classifier used by Maxion and Townsend [6], three elements introduced there still apply here: the classification formulation, the bag-of-words model, and the naive-Bayes classifier. These three strategies are not unique to detecting masquerading sessions, but rather they appear in the far more general context of intrusion detection and text classification. Below, a brief review of the three elements appears in the context of identifying masquerading sessions.

A. Classification Formulation

As in much of intrusion detection, examples of normal behavior are plentiful, but intrusions themselves are rare. In the context of detecting masquerading sessions, many examples of proper sessions are available. In the training data, however, no masquerading sessions are known. Moreover, masquerading sessions in the test data are expected to be rare.

Naturally, proper sessions in the training data are used to build the profile of proper sessions. To create a profile for masquerading sessions, the sessions of *other* known users in the training data are used. For the dataset of [8], this approach is particularly appropriate, because the artificially created masquerading sessions do indeed come from known users excluded from the training set.

Each new session is compared against the profiles of the proper sessions and against the masquerading sessions. The anomaly-detection problem has been reformulated as a classification problem over the proper class and the artificially created intruder class. This reformulation allows many powerful techniques of classification theory to be used on the anomaly-detection problem.

Even in a more general context of anomaly detection, the profile of the proper sessions is not defined only by the proper sessions, but also by the sessions of other users. The extent of the proper profile in feature space is most naturally defined through both the proper cases and the intruder cases. So the classification formulation is potentially useful even for other anomaly-detection problems.

B. Bag-of-Words Model

The so-called bag-of-words model is perhaps the most widely used model for documents in information retrieval. A text document can be reduced to a bag of words, by ignoring the sequence information and only counting the number of occurrences of each word. For classification of text documents, this simple model often performs better than more complicated models involving sequence information.

Let $\mathbf{C} = \{1, 2, \ldots, C\}$ be the set of all possible commands. User session number s is simply a finite sequence $c_s = (c_{s1}, c_{s2}, \ldots, c_{sk}, \ldots, c_{sz_s})$ of commands. In other words, session number s of length z_s is identified with a sequence $c_s \in \mathbf{C}^{z_s}$. In the bag-of-words model, the sequence information is ignored. So c_s is reduced from the sequence (c_{sk}) into the multi-set $\{c_{sk}\}$.

As demonstrated in [6], the bag-of-words model outperforms many more complicated models attempting to take advantage of the sequence information. In particular, the techniques reviewed in [8] and earlier techniques based on a Markov model of command sequences achieved worse results.

C. Naive-Bayes Classifier

In the field of text classification, the simple naive-Bayes classifier is perhaps the most widely studied classifier, used in conjunction with the bag-of-words model for documents. The naive-Bayes classifier is the Bayes-rule classifier for the bag-of-words model, under the typical uniform loss function for misclassification.

Suppose that each user has a distinct pattern of command usage. Let c_s denote the sequence of commands in session number s.

By Bayes inversion formula, the posterior probability $P(u|c_s)$ of user u given the sequence c_s is

$$P(u|c_s) = \frac{P(c_s|u)P(u)}{P(c_s)} \propto P(c_s|u)P(u), \tag{1}$$

where $P(u)$ is the prior probability for user u, and $P(c_s|u)$ is the probability that the sequence c_s was generated by user u. In practice, c_s is assigned to the user $u_0 = \text{argmax}\ \{P(c_s|u)P(u) : u = 1, 2, \ldots, U\}$, among U different users. In other words, the session c_s is assigned to the user u_0 who most likely generated that session.

In the naive-Bayes model, each command is assumed to be chosen independently of the other commands. User u has probability p_{uc} of choosing command c. Because each command is chosen independently, the probability $P(c_s|u)$, that user u produced the sequence $c_s = (c_{s1}, c_{s2}, \ldots, c_{sk}, \ldots, c_{sz_s})$ of commands in session number s, is simply

$$P(c_s|u) = \prod_{k=1}^{z_s} P(c_{sk}|u) = \prod_{k=1}^{z_s} p_{uc_{sk}} = \prod_{c=1}^{C} p_{uc}^{n_{sc}}, \tag{2}$$

where $n_{sc} = \sum_{k=1}^{z_s} 1_{\{c_{sk}=c\}}$ is the total count of command c in session s.

III. Theory

The simple naive-Bayes classifier is typically built on a training set of labeled documents. Nigam and his colleagues [7] demonstrated that classification accuracy can be improved significantly by incorporating labeled as well as unlabeled documents in the training set. The algorithm for training this new naive-Bayes classifier can be derived from the formalism of the EM-algorithm.

This section continues the development of the probabilistic model introduced in Section II. First, the maximum-likelihood formalism is used to estimate the model parameters necessary for implementing the naive-Bayes classifier. Then the complete log-likelihood is introduced to incorporate new unidentified sessions. Finally, an iterative hill-climbing procedure for optimizing the log-likelihood is generated from the EM-algorithm.

A. Maximum-Likelihood Estimation

In Section II, the bag-of-words model for documents was introduced. The session was defined to be a sequence of commands chosen independently from the set of possible commands. Then the naive-Bayes classifier was derived from the simple Bayes inversion formula. In practice, the parameters of the probabilistic model must be estimated from the data itself.

From now on, only two distinct classes of sessions are considered, the proper class and the masquerading class. The indicator *variable* $1_s = 1$ exactly when session s is a masquerading session. Let $1 - \epsilon$ and ϵ be the prior probabilities that a session is proper and masquerading, respectively. Moreover, let p_c and p'_c be the probability of command c in a proper and masquerading session, respectively.

B. Likelihood from Test Sessions

The log-likelihood L_s of a test session s is simply, up to an additive constant

$$L_s = (1 - 1_s)(\log(1 - \epsilon) + \sum_{c=1}^{C} n_{sc} \log p_c) + 1_s(\log \epsilon + \sum_{c=1}^{C} n_{sc} \log p'_c) \tag{3}$$

where n_{sc} is the total count of command c in session s.

Assuming that all test sessions are generated independently of each other, the cumulative log-likelihood L^t after t test sessions is, up to an additive constant

$$L^t_+ = \sum_{s=1}^{t} L_s \tag{4}$$

$$= w^t_+ \log(1 - \epsilon) + \sum_{c=1}^{C} n^t_{+c} \log p_c + w'^t_+ \log \epsilon + \sum_{c=1}^{C} n'^t_{+c} \log p'_c \tag{5}$$

where

$$w^t_+ = \sum_{s=1}^{t} (1 - 1_s), \tag{6}$$

$$w'^t_+ = \sum_{s=1}^{t} 1_s, \tag{7}$$

$$n^t_{+c} = \sum_{s=1}^{t} (1 - 1_s) n_{sc}, \tag{8}$$

$$n'^t_{+c} = \sum_{s=1}^{t} 1_s n_{sc}. \tag{9}$$

Here w^t_+ and $w'^t_+ = t - w^t_+$ are the cumulative numbers of proper and masquerading sessions, respectively; n^t_{+c} and n'^t_{+c} are the cumulative counts of command c amongst proper sessions and masquerading sessions, respectively, in the t total observed test sessions.

C. Likelihood from Training Sessions

Now let n^0_{+c} denote the total count of command c among proper sessions in the training set. Likewise, let n'^0_{+c} denote the total count of command c among masquerading sessions in the training set. Letting r denote a session in the training set,

$$n^0_{+c} = \sum_r (1 - 1_r) n_{rc}, n'^0_{+c} = \sum_r 1_r n_{rc}. \tag{10}$$

Assuming that the sessions in the training set are generated independently, the log-likelihood L^0_+ of the proper and masquerading sessions in the training set is

$$L^0_+ = \sum_{c=1}^{C} n^0_{+c} \log p_c + \sum_{c=1}^{C} n'^0_{+c} \log p'_c. \tag{11}$$

This log-likelihood L^0_+ is useful for providing initial estimates of p_c and p'_c but provides no information about ϵ.

D. Posterior Likelihood

Rare classes and rare commands may not be properly reflected in the training set. To avoid zero estimates, smoothing is typically applied to the maximum-likelihood estimators. This smoothing can also be motivated by shrinkage estimation under Dirichlet priors on the parameters ϵ, p_c, and p'_c.

Here the parameters ϵ, p_c, and p'_c are drawn from known prior distributions with fixed parameters, and a simple standard Bayesian analysis is applied. Suppose that

$$(1-\epsilon,\epsilon) \sim \text{Beta}(\beta,\beta'), \quad (12)$$

$$p \sim \text{Dirichlet}(\alpha), \quad (13)$$

$$p' \sim \text{Dirichlet}(\alpha'), \quad (14)$$

where α, α', β, and β' are taken to be known fixed constants specified in advance. Then the cumulative posterior log-likelihood $\tilde{L}^t$ is, up to an additive constant

$$\begin{aligned}\tilde{L}^t &= (\beta-1+w_+^t)\log(1-\epsilon)+ \\ &\quad \sum_{c=1}^{C}(\alpha_c-1+n_{+c}^0+n_{+c}^t)\log p_c + \\ &\quad (\beta'-1+w_+'^t)\log\epsilon + \\ &\quad \sum_{c=1}^{C}(\alpha'_c-1+n_{+c}'^0+n_{+c}'^t)\log p'_c. \end{aligned} \quad (15)$$

Here w_+^t, $w_+'^t$, n_{+c}^t, and $n_{+c}'^t$, defined in Equations 6–9, are cumulative quantities determined by the t available test sessions; n_{+c}^0 and $n_{+c}'^0$, defined in Equation 10, are the fixed quantities determined by the training sessions.

E. Shrinkage Estimators

Equation 15 gives the cumulative posterior log-likelihood $\tilde{L}^t$ after observing t test sessions, in addition to the training sessions. Shrinkage estimators are just the maximum-likelihood estimators calculated from the posterior log-likelihood $\tilde{L}^t$. So cumulative shrinkage estimators $\hat{\epsilon}^t$, $\hat{p}_c^t$ and $\hat{p}_c'^t$ for ϵ, p_c and p'_c after $t>0$ sessions are

$$\hat{\epsilon}^t = \frac{\beta'-1+w_+'^t}{\beta-1+\beta'-1+t}, \quad (16)$$

$$\hat{p}_c^t = \frac{\alpha_c-1+n_{+c}^0+n_{+c}^t}{\sum_{v=1}^{C}(\alpha_v-1+n_{+v}^0+n_{+v}^t)}, \quad (17)$$

$$\hat{p}_c'^t = \frac{\alpha'_c-1+n_{+c}'^0+n_{+c}'^t}{\sum_{v=1}^{C}(\alpha'_v-1+n_{+v}'^0+n_{+v}'^t)}. \quad (18)$$

F. Complete Log-Likelihood

Initially, the training set is used to build the naive-Bayes classifier. As a new unidentified session is presented to the classifier, that new session is scored by the classifier and used to update the classifier. This scoring and updating procedure is repeated until convergence. As each new session is presented, the classifier is updated in a self-consistent manner.

For a session s in the training set, the identity is known. Specifically, $1_s=0$ for a proper session, and $1_s=1$ for a masquerading session in the training set. For a new unidentified session, 1_s is a missing variable that must be estimated from the available data.

G. EM-Algorithm for Naive-Bayes Model

The EM-algorithm [1] is an iterative procedure used to calculate the maximum-likelihood estimator from the complete log-likelihood. Each iteration of the EM-algorithm includes two steps, the expectation step and the maximization step.

In the expectation step, the indicators 1_s are replaced by their expectations, calculated from the current estimates of model parameters. In the maximization step, the new estimates of the model parameters are calculated by maximizing the expected log-likelihood. For each stage t, these two-steps are repeated through multiple iterations until convergence. Below the iterations of the EM-algorithm during a fixed stage t are described.

Let $\theta^t=(\epsilon^t,p_c^t,p_c'^t)$ denote the estimate of $\theta=(\epsilon,p_c,p'_c)$ at stage t, after observing the first t test sessions. For each known session r in the training set, 1_r is a known constant. So n_{+c}^0 and $n_{+c}'^0$ of Equation 10 remain fixed even as θ^t changes.

For each *unidentified* test session $s=1,2,\ldots,t$, the expectation step estimates $E[1_s|c_s;\theta^t]=P(1_s=1|c_s;\theta^t)$ via the Bayes inversion formula as

$$P(1_s=1|c_s;\theta^t)=\frac{P(1_s=1;\theta^t)P(c_s|1_s=1;\theta^t)}{P(c_s;\theta^t)}, \quad (19)$$

where

$$\begin{aligned}P(c_s;\theta^t) &= P(1_s=0;\theta^t)P(c_s|1_s=0;\theta^t)+ \\ &\quad P(1_s=1;\theta^t)P(c_s|1_s=1;\theta^t). \end{aligned} \quad (20)$$

For the expectation step of the current iteration, the estimate $\hat{\theta}^t$ from the maximization step of the previous iteration is used for θ^t.

The maximization step calculates updated estimate $\hat{\theta}^t$ of the parameter θ^t. The usual cumulative shrinkage estimators of Equations 16–18 are used. However, the counts w_+^t, $w_+'^t$, n_{+c}^t, and $n_{+c}'^t$ are now no longer known constants but rather are random variables. These random variables are replaced by their estimates from the expectation step above. In other words, the maximization step of the current iteration uses the estimates from the expectation step of the current iteration.

H. Sequential Processing of Sessions

Table I shows abstractly how the self-consistent update is used to process the incoming stream of test sessions. The algorithm processes in separate stages each new unidentified test session in sequence, from the first incoming session to the last incoming session. At stage t after sessions $s=1,2,\ldots,t$ have arrived, the EM-algorithm is run until convergence. In other words, at each stage t, multiple iterations of the expectation step and maximization step occur.

At the end of each *stage*, new consistent scores and models found after the EM-algorithm converges. These consistent scores and models are kept for each stage and used as the starting point for the next stage. For example, the opening iteration of the EM-algorithm for stage $t+1$ uses the closing values of the final iteration of the EM-algorithm in stage t. Estimates from the training set are used to initialize the very first iteration of the EM-algorithm at stage 1, after the first new test session arrives.

TABLE I

SELF-CONSISTENT UPDATE APPLIED TO SEQUENCE OF TEST SESSIONS.

- **Inputs:**
  - Target user u. Cut-off threshold h *not* necessary.
  - Initial training set $\mathbf{R}$ of identified sessions from all users.
  - Streamed sequence $\mathbf{S}$ of new unidentified test sessions for user u.
- Build initial classifier model $\hat{\theta}^0 = (\hat{\epsilon}^0, \hat{p}^0, \hat{p}'^0)$ for user u by using training set $\mathbf{R}$.
- Loop over each new session $t = 1, 2, \ldots, S$ in test sequence $\mathbf{S}$:
  - Start initial $\hat{\theta}^t = \hat{\theta}^{t-1}$.
  - Repeat until consistency:
    - E-step: Use Bayes inversion formula to calculate new scores $P(1_s = 1|c_s; \hat{\theta}^t)$, for $s = 1, 2, \ldots, t$, under current $\hat{\theta}^t$.
    - M-step: Update model $\hat{\theta}^t$, using new scores.
    - Keep new model $\hat{\theta}^t$ of stage t from current iteration for the next iteration.
  - Keep new scores for sessions $1, 2, \ldots, t$ under score set t.
- **Outputs:**
  - Classifier model $\hat{\theta}^S$ for target user u built on training set $\mathbf{R}$ and updated by the sequence $\mathbf{S}$ of sessions.
  - Multiple score sets, $t = 1, 2, \ldots, S$. Score set t contains scores of test sessions $s = 1, 2, \ldots, t$, updated based on information from all sessions $s = 1, 2, \ldots, t$ observed at stage t.

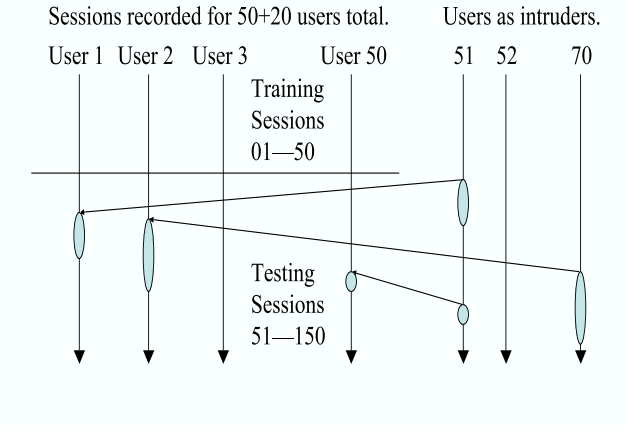


Fig. 1. Splicing of sessions.

I. Interpretation of Self-Consistency

From the computational perspective, the EM-algorithm allows the unidentified test sessions to be used. Since each new test session's identity is unknown, using that test session to update the profile blindly is risky. Instead, the EM-algorithm imputes a best guess on the identity of the unidentified session. This best guess acts as a placeholder and allows sequential classification to continue.

By imputing a probabilistic score for the identity of each unidentified test session, the EM-algorithm makes full use of the information in the unlabeled test sessions. In essence, the EM-algorithm improves masquerade detection by carefully incorporating the unlabeled test data in a meaningful way.

J. Multistep Paradigm and Backtracking

The multistep update is not as obvious as the one-step update but is indeed more intuitively appealing and logically sound. Because concept drift is unavoidable in masquerade detection, the detection rate depends not only on the underlying model but also the update scheme. The careful choice of update scheme is critical for the ongoing detection of masquerades. As a specific example of the multistep update, the self-consistent update based on the EM-algorithm has a firm statistical foundation and offers concrete optimality properties.

The backtracking in the multistep update naturally requires more computational effort. Additionally, because new scores are assigned to all previous sessions as each new session arrives in each successive stage, different modes of evaluation are possible.

IV. Results

The technique proposed in this paper is tested on a standard dataset previously analyzed by other authors using different techniques. The dataset contains command logs from 50 users on a UNIX multiuser system. First a brief overview of the reference dataset is provided. Then the experiments performed are described in detail. Finally, the results are compared with [6].

A. Reference Dataset

The dataset contains sequences of actual user commands, divided into artificial sessions of 100 commands each. For privacy reasons, the command logs were stripped of the command options and arguments, leaving only the commands themselves. Each original recording was declared to be free of intrusions. To create masquerading sessions, blocks of sessions from foreign users were spliced into the test set. This process of creating artificial masquerading sessions is illustrated in Figure 1. All intrusions appear only in discrete session units.

Originally command data was collected over a total of 70 users. From these 70 users, 20 users were selected at random to serve as intruders, as shown in Figure 1. Their sessions appear in the final dataset only as masquerading sessions for the 50 known users.

The first 50 sessions of the 50 users are declared to be free of intruders. These 50 sessions for the 50 users constitute the training set. The remaining 100 sessions for each of the 50 users form the test set. Sessions from the 20 excluded users were injected into the test portions of the 50 known users. The extra sessions in the test portions are then removed, leaving just 100 test sessions for each of the 50 known users.

To avoid artificially optimistic results, the following details of the data creation were *not* used as part of the classification procedures proposed in this report. Essentially, once an intruder has broken into a user account, he is likely to return on the very next session. In creating the reference dataset, a proper session is followed by a masquerading session with 0.01 probability. A masquerading session is followed by another masquerading session from the same intruder with 0.80 probability. A chain of masquerading sessions is taken from that intruder's log in a contiguous sequence. The exact details of this random splicing process are described in full by Schonlau and his colleagues [8].

B. Experimental Design

The details for one single experiment are described below. The example shows how masquerading sessions on User 12 were

detected. For other users, the same analysis applies.

The full training set was divided into two classes. The proper class consisted of 50 sessions for User 12. The masquerade class consisted of the 50 sessions for each of the other 49 users. So the masquerade class was built on a total of 2450 sessions.

The test set for user 12 had 100 sessions, all purportedly proper sessions. The classifier for user 12 was then run against each of the 100 sessions in a sequential fashion. As a new session is added, the scores of the previous sessions are also updated. At the end of the 100 sessions, the final set of all 100 scores were recorded.

The ROC curve for user 12 was generated by thresholding against the *final* set of scores, after all test sessions were presented sequentially. In this case, a session's score is the probability that the session was generated from the proper profile. Therefore, sessions with lower scores were flagged before sessions with higher scores.

C. Classifier Settings

A self-consistent naive-Bayes classifier was built for each individual user. The algorithm is outlined in Subsections III-G and III-H. The initial naive-Bayes was built on the training set. As each new session is presented, that new session is first scored by the previous classifier. Afterwards, all the unidentified sessions are used to construct the self-consistent naive-Bayes. This procedure is repeated for each new session.

In the full self-consistent naive-Bayes classifier presented in Section III, the fraction ϵ of masquerading sessions was adjusted in each iteration. In the results presented here, the fraction $\epsilon = 0.50$ was held fixed and was not estimated as part of the EM-algorithm. Keeping the prior weight $(1-\epsilon, \epsilon)$ fixed at $(0.50, 0.50)$ allowed for faster convergence and did not adversely affect the final results.

For each new session, the EM-algorithm was used to adjust the fractions p_c and p'_c. The Dirichlet parameters $\alpha_c = 1.01$ and $\alpha'_c = 1.01$ for all $c \in \mathbf{C}$ were chosen to match the parameters used in [6]. The EM-algorithm was iterated until the change between iteration i and $i+1$, measured by the quantity $||p^{(i+1)} - p^{(i)}||^2 + ||p'^{(i+1)} - p'^{(i)}||^2$ averaged over the total number of estimated parameters, was less than some tolerance, set to be 2.2×10^{-16}. In practice, the algorithm was not sensitive to the precise tolerance.

D. Composite ROC Curves

A separate classifier was created for each user and used to score that user's test sessions. The scores of test sessions from one user can be compared because those scores were assigned by that user's classifier. However, scores from different users cannot easily be compared because those scores are assigned from different classifiers.

To evaluate a strategy's success over the entire dataset of 50 users, a composite ROC curve can be useful. There are several common ways to integrate the individual 50 ROC curves into one single curve, but all methods are at best arbitrary.

In this paper, the scores from the 50 different classifiers were taken at face value. The 5000 total test sessions from the 50 user were sorted, and sessions with the lowest scores were flagged first. Because the scores were probability values, this global thresholding strategy has some merit. This method for constructing the composite ROC curve was also used in [6] and [8]. Because the same methodology was used in these previous papers, the composite ROC curves can be meaningfully compared.

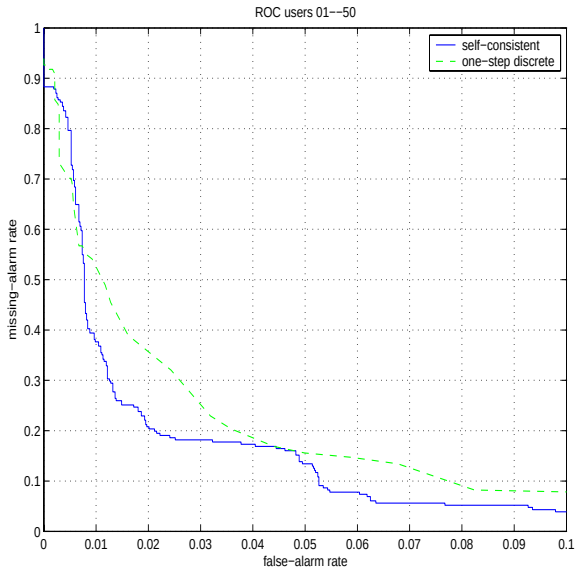


Fig. 2. ROC curve over users 01–50, less than 0.10 false-alarm rate.

E. Experimental Results

Two ROC curves are used to compare the self-consistent naive-Bayes classifier to the one-step discrete adaptive naive-Bayes classifier of [6]. These curves together demonstrate that the self-consistent naive-Bayes classifier offers significant improvements. All results reported here are based on offline evaluation, after all the test sessions have been presented sequentially. Online evaluation and deferral strategies are discussed in a separate paper.

Figure 2 shows in fine detail the composite ROC curves of all 50 users, for false-alarm rate 0.00–0.10. Indeed, the self-consistent naive-Bayes outperforms the adaptive naive-Bayes uniformly for all but the small portion of false-alarm rate 0.00–0.01. In particular, for false-alarm rate 0.013, the self-consistent naive-Bayes reduces the missing-alarm rate of the adaptive naive-Bayes by 40%.

Figure 3 shows the ROC curve for user 12 alone. As noted by the other authors [6], [8], user 12 is a challenging case because the masquerading sessions in the test set appear similar to the proper sessions. On user 12, the adaptive naive-Bayes performs worse than random guessing, but self-consistent naive-Bayes encounters little difficulty. In fact, the 50 ROC curves over each individual user show that self-consistent naive-Bayes typically outperforms adaptive naive-Bayes.

V. Discussion

The self-consistent naive-Bayes classifier outperforms the adaptive naive-Bayes classifier in all practical circumstances, where only low false-alarm rates can be tolerated. For false-alarm rate 0.013, the self-consistent naive-Bayes classifier lowers the missing-alarm rate by 40%.

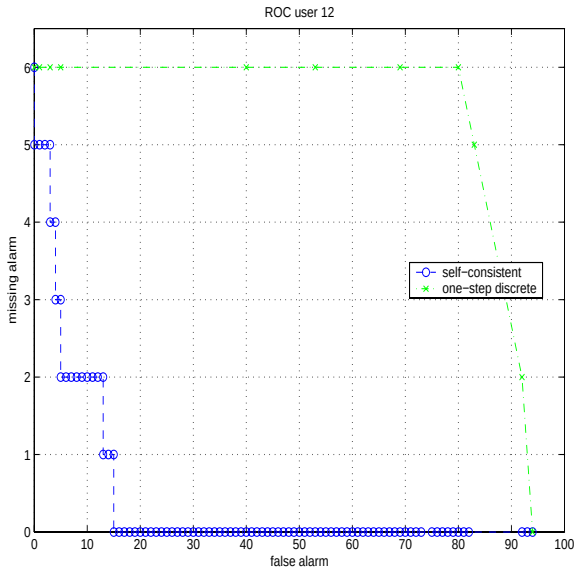


Fig. 3. ROC curve user 12

A. Parallel User Sessions

In a more realistic scenario, all 50 users would generate new sessions in parallel. In such a situation, changes to one user would affect other users directly. Because self-consistent naive-Bayes can change scores of earlier sessions as well as later sessions, the self-consistent naive-Bayes will benefit most from the additional information of other users. Although the adaptive naive-Bayes can use the additional information, only later sessions benefit because scores of earlier sessions cannot be changed.

B. Long-Term Scenario

Unlike the adaptive naive-Bayes, the self-consistent naive-Bayes is not forced to make a binary decision as each new session is presented. This advantage will become more dramatic as the number of users and the number of sessions increase. As the classifier learns from more cases, mistakes from earlier decisions are propagated onwards and magnified. Because the adaptive naive-Bayes is forced to make a binary decision immediately, it is also more likely to make errors. In a long-term scenario, the improvements of the self-consistent naive-Bayes classifier are expected to become far more pronounced.

C. Computation Time

Although the self-consistent naive-Bayes classifier offers a great number of advantages, more computation is required. The iterative procedure used to calculate the probabilities and to update the profile must be run until convergence. This additional effort allows the self-consistent naive-Bayes classifier to assign scores in a nongreedy fashion. Naturally, searching through the space of models requires more effort than the simple greedy approach.

In practice, convergence is often quick because most sessions have probability scores at the extremes, near 0 or 1. In the context of sequential presentation, only one single session is presented at a time. Computing the score of a single session requires little additional effort. Moreover, the set of scores from one stage can be used as the starting point to calculate scores for the next stage. This incremental updating approach relieves the potential computational burden.

D. Nonstationary User Behavior

For some users, the sessions in the test set differ significantly from the sessions in the training set. Because user behavior changes unpredictably with time, a detector based on the old behavior uncovered in the training set can raise false alarms. This high rate of false alarms can render any detector impractical, because there are not enough resources to investigate these cases.

In certain real world applications, however, identifying changed behavior may well be useful because a proper user can misuse his own account privileges for deviant and illicit purposes [5]. If the detector is asked to detect all large deviations in behavior, then flagging unusual sessions from the proper user is acceptable. This redefinition is especially useful to prevent a user from abusing his own privileges.

E. Future Directions

The iterative nature of the EM-algorithm is quite intuitive. In fact, many instantiations [2]–[4], [9] of the EM-algorithm existed well before the general EM-algorithm was formulated in [1]. Moreover, the self-consistency paradigm can be applied even to classifiers not based on a likelihood model. A self-consistent version can be constructed for potentially any classifier, in the same way that a classifier can be updated by including the tested instance as part of the modified training set.

The naive-Bayes classifier relies only on command frequencies. As a user's behavior changes over time, the profile built from past sessions become outdated. An exponential-weighing process can be applied to counts from past sessions. For the naive-Bayes classifier, this reweighting of counts is especially simple. Such an extension becomes even more valuable in realistic scenarios, in which the classifier is used in a sequential context over a long period of time.

VI. Summary

In previous approaches to detecting masquerading sessions, the detector was forced to make a binary decision about the identity of a new session. The self-consistent naive-Bayes does not make a binary decision but rather estimates the probability of a session being a masquerading session. Moreover, past decisions can be adjusted to accommodate newer sessions. Experiments prove that this sensible extension markedly improves over more restrictive adaptive approaches, by reducing the missing-alarm rate by 40%.

The self-consistent naive-Bayes classifier extends the usual naive-Bayes classifier by taking advantage of information from unlabeled instances. New instance are assigned probabilistic labels, and then the profiles are updated in accordance with the assigned probabilities of the new instances. This procedure is iterated until convergence to a final set of probabilities which are consistent with the updated profile.

By its very nature, the self-consistent naive-Bayes classifier is adaptive to the new sessions. Moreover, information from

new sessions is also used to adjust scores of previous new sessions. In this way, the scores of sessions are assigned in a self-consistent manner. As a specific instance of the EM-algorithm, the self-consistent naive-Bayes classifier finds a model optimal with respect to its likelihood given the available data.

VII. Acknowledgments

This research project was funded in part by the US Department of Justice grant 2000-DT-CX-K001. Jerome H. Friedman of the Stanford University Statistics Department provided invaluable advice through many helpful discussions. Jeffrey D. Ullman of the Stanford University Computer Science Department introduced the author to the field of intrusion detection and offered insightful critiques throughout the past three years. The author is grateful for their guidance and support.

References

[1] Arthur P. Dempster, Nam M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society, Series B*, 39(1), 1–38, 1977.

[2] N. E. Day. "Estimating the components of a mixture of two normal distributions." *Biometrika*, **56**, 463–474, 1969.

[3] Victor Hasselblad. "Estimation of parameters for a mixture of normal distributions." *Technometrics*, **8**, 431–444, 1966.

[4] Victor Hasselblad. "Estimation of finite mixtures of distributions from the exponential family." *Journal of American Statistical Association*, **64**, 1459–1471, 1969.

[5] Vernon Loeb. "Spy case prompts computer search." *Washington Post*, 05 March 2001, page A01.

[6] Roy A. Maxion and Tahlia N. Townsend. "Masquerade Detection Using Truncated Command Lines." *International Conference on Dependable Systems and Networks (DSN-02)*, pp. 219–228, Washington, DC, 23–26 June 2002. IEEE Computer Society Press, Los Alamitos, California, 2002.

[7] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, Tom Mitchell. "Text classification from labeled and unlabeled documents using EM." *Machine Learning*, 39:2:103–134, 2000.

[8] Matthias Schonlau, William DuMouchel, Wen-Hua Ju, Alan F. Karr, Martin Theus, and Yehuda Vardi. "Computer intrusion: detecting masquerades." *Statistical Science*, 16(1):58–74, February 2001.

[9] John H. Wolfe. "Pattern clustering by multivariate mixture analysis." *Multivariate Behavioral Research*, **5**, 329–350, 1970.

[10] Kwong H. Yung. "Using feedback to improve masquerade detection." *Lecture Notes in Computer Science: Applied Cryptography and Network Security*. Proceedings of the ACNS 2003 Conference. LNCS **2846**, 48–62. Springer-Verlag, October 2003.

Compact Optical Mirrors with Broadband Polarization-Insensitive Reflectivity using Coupled Photonic Crystal Slabs

Wonjoo Suh, Virginie Lousse, and Shanhui Fan

Abstract. It was recently demonstrated that a photonic crystal slab can function as a mirror for externally incident light with near-complete reflectivity over a broad wavelength range for transverse magnetic (*TM*) polarized light. We propose a coupled photonic crystal structure that enables broadband reflection over a sizeable angular range for both transverse electric (*TE*) and *TM* polarized light. We expect such mirror to play important roles in optical devices where polarization independent high reflectivity is needed.

Index Terms: Guided Resonance, Optical Filters, Sensors, Photonic Crystals.

DIELECTRIC optical mirrors with near 100% reflectivity play important roles in optical devices including filters, sensors, switches, and laser structures. The conventional way of obtaining high-reflectivity dielectric mirrors for applications including optical Micro-Electrical-Mechanical-Systems (MEMS) is to use multi-layer films, since metallic mirrors are extremely lossy at optical frequencies. Also, in optical MEMS research, reducing the displacement of mechanical tuning is of great interest since such reduction directly increases the response speed and decreases the actuation force. However, the multilayer nature makes it hard to combine these devices with MEMS technology for sensors and tunable filters applications, since up to 100 dielectric layers is often required for extremely high reflectivity.

In this paper, we demonstrate that broad-band near 100% reflection for normally incident light can be obtained by using a two dielectric layers of photonic crystal slabs. The spectral function of the proposed structure provides a wide angular range of reflection for both *TE* and *TM* polarizations, which is of practical importance when dealing with finite beam sizes. When the beam size is finite, unlike plane waves there are a range of angular components in the incident beam and therefore it is crucial that high reflection for a wide range of angular components is created for various applications.

A photonic crystal slab, which is a building block for our structure, consists of a square lattice of air holes in a high-dielectric film as shown in Figure 1. The operating mechanism of the proposed structure is a guided resonance [1]–[3]. A guided resonance mode in a photonic crystal slab is excited by externally incident light. A typical spatial distribution of the power density in electric fields for a guided resonance mode is shown in Figure 2.

From Figure 2(b), which shows the vertical slice of the power distribution, we can clearly see that the majority of the power is concentrated inside the photonic crystal slab. Although the resonances are strongly confined within the slab, the periodic index contrast provides the phase matching mechanisms that allow these modes to couple into radiation modes and possess a finite lifetime. The interaction between these guided resonance modes and the external radiation gives rise to interesting reflection and transmission properties. The resonance frequency and the lifetime of the guided resonance are determined by the structure of the photonic crystal, and therefore we can design the desired reflection and transmission properties by engineering the photonic crystal structure, i.e., the radius of the airholes, the thickness of the photonic crystal slabs and the dielectric constant of the photonic crystal slab.

The authors are with the Department of Electrical Engineering at Stanford University. Email: wjsuh@stanford.edu

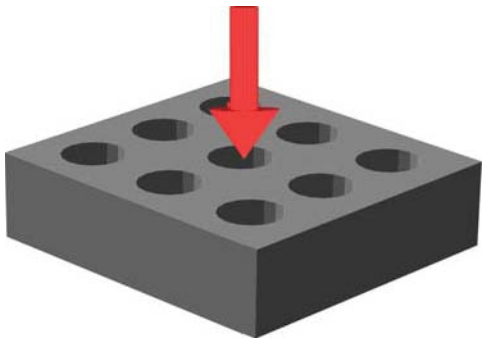


Fig. 1. Schematic of a photonic crystal slab consisting of a square lattice of air holes in a high-dielectric film. The arrow represents the direction of the incident light.

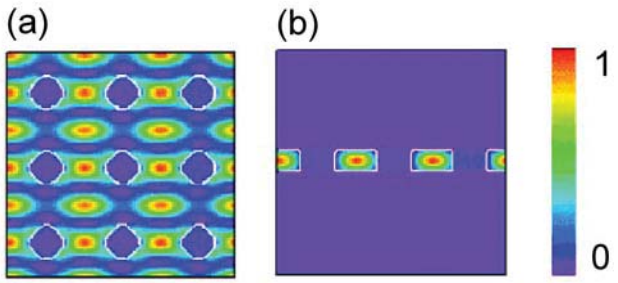


Fig. 2. Spatial distribution of the power in electric fields on (a) a horizontal slice and (b) a vertical slice for normally incident light.

It has recently been shown that when in a photonic crystal slab with dielectric constant of 12, a thickness of $0.55a$, and radius of air holes of $0.4a$, where a is the lattice constant, we can obtain a broadband reflection of over 99% [4]. Also, it has been reported that the filter function has a wide angular dependency for *TM* polarization near $0.53(c/a)$ [5]. However, broadband reflection with a wide angular range is not created

for *TE* polarized light near $0.53(c/a)$ for this structure. For *TE* polarization, near 100% reflection with a wide angular range is rather created near $0.45(c/a)$. Therefore, if we design the second slab to have near 100% reflection with a wide angular range near $0.53(c/a)$ for *TE* polarization, we would be able to create near 100% reflection for a wide angular range for both *TE* and *TM* polarized light by coupling two different photonic crystal slabs that compensate each other. This can be theoretically shown using the transfer-matrix formalism. When we define r and t as the reflection and transmission coefficient of a single photonic crystal slab, the transfer matrix T for a single slab reduces to the following form [6]:

$$T = \begin{pmatrix} t + \frac{rr^*}{t^*} & -\frac{r^*}{t^*} \\ -\frac{r}{t} & \frac{1}{t} \end{pmatrix} = \begin{pmatrix} \frac{1}{t^*} & -\frac{r^*}{t^*} \\ -\frac{r}{t} & \frac{1}{t} \end{pmatrix}. \quad (1)$$

When there are two photonic crystal slabs separated by a displacement of h, the total transfer matrix becomes:

$$T_{total} = \begin{pmatrix} \frac{e^{-j\phi}}{t_1^* t_2^*} + \frac{r_2 r_1^* e^{+j\phi}}{t_2 t_1^*} & -\frac{r_2^* e^{-j\phi}}{t_1^* t_2^*} - \frac{r_1^* e^{+j\phi}}{t_2 t_1^*} \\ -\frac{r_1 e^{-j\phi}}{t_1 t_2^*} - \frac{r_2 e^{+j\phi}}{t_1 t_2} & \frac{r_1 r_2^* e^{-j\phi}}{t_1 t_2^*} + \frac{e^{+j\phi}}{t_1 t_2} \end{pmatrix}, \quad (2)$$

where $\phi = \frac{\omega}{c}h$ is the phase shift that the wave acquires as it propagates through the air cavity that two photonic crystal slabs are forming. Also, r_1, t_1 and r_2, t_2 are the reflection and transmission coefficients of the first and second photonic crystal slabs, respectively. From the transfer matrix of the total system, we can extract the total transmitted power density of the coupled slab structure as follows:

$$|t_{total}|^2 = \frac{|t_1 t_2|^2}{|r_1 r_2|^2 + 2|r_1 r_2| \cos\left\{\arg(\frac{r_2}{r_1 t_2^*})\right\} + 1}. \quad (3)$$

From Eq. (3), we can clearly see that when the transmission through either of the two slabs is zero, the total transmission is zero and hence we obtain near 100% reflection.

As a physical realization of the theoretical analysis, using a first principles finite-difference time-domain (FDTD) simulation [7], we consider the two slab structure schematically shown in Figure 3. The first slab has a dielectric constant of 12, radius of $0.4a$ and thickness of $0.55a$. From Figure 4, we can see that near frequency $0.45(c/a)$, where we have the first resonant reflection, the reflection spectrum has a wide angular range of reflection for *TE* polarized light, whereas near frequency $0.53(c/a)$, where we have the second resonant reflection, the reflection spectrum has a wide angular range of reflection for *TM* polarized light.

Using this information, we design the second slab such that it would completely compensate for the angular dependency of the reflectivity in the first slab. This can be done by shifting the first resonance, which creates a wide angular range of reflection for *TE* polarized light, to frequency $0.53(c/a)$. By choosing a radius of $0.35a$ and a thickness of $0.2a$, with the same dielectric constant as the first slab, we obtain a wide angular range of reflection for *TE* polarized light near frequency $0.53(c/a)$. When we couple these two different photonic crystal slabs, we indeed create near 100% reflection for a wide angular range for both *TE* and *TM* polarized light as shown in Figure 5. Two different slabs compensate each

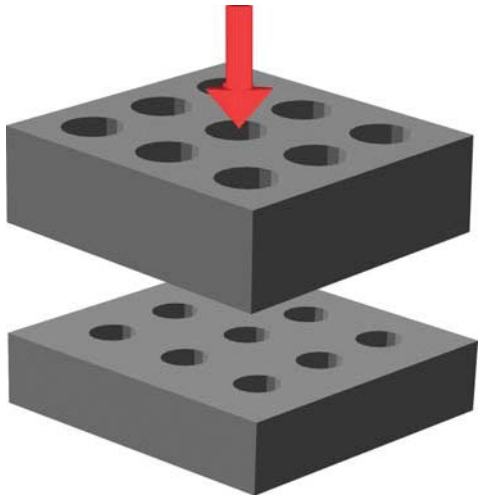


Fig. 3. Schematic of a photonic crystal filter consisting of two coupled photonic crystal slabs. The arrow represents the direction of the incident light. The first slab has a dielectric constant of 12, radius of $0.4a$ and thickness of $0.55a$. On the other hand, the second slab has the same dielectric constant as the first slab, but has a radius of $0.35a$ and a thickness of $0.2a$.

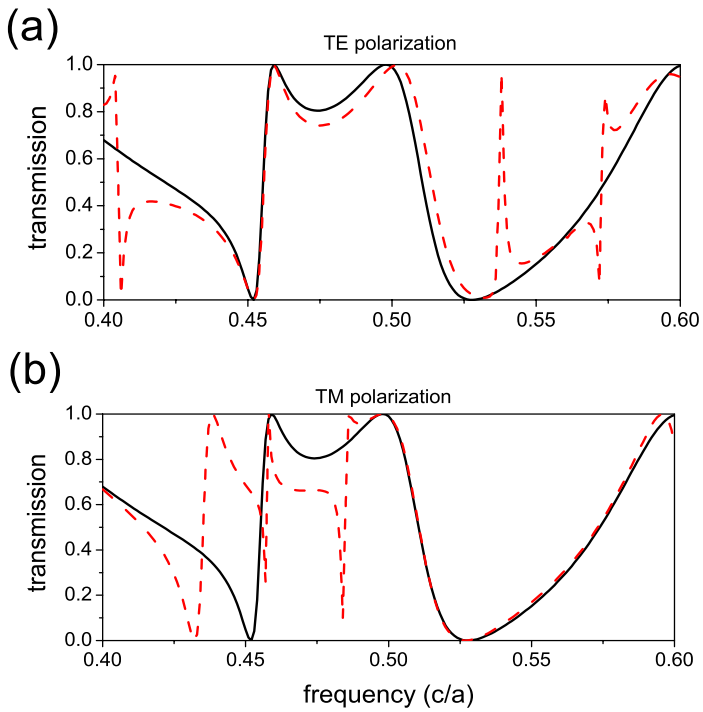


Fig. 4. Transmission spectrum for various incident angles through a photonic crystal slab with dielectric constant of 12, radius $0.4a$ where a is the lattice constant, and thickness $0.55a$. (a) shows the transmission when incident light has *TE* polarization and (b) shows the transmission when incident light has *TM* polarization. Both in (a) and (b), solid line is the transmission upon normally incident light, while the dashed line is with incident angle 1° near frequency $0.53(c/a)$.

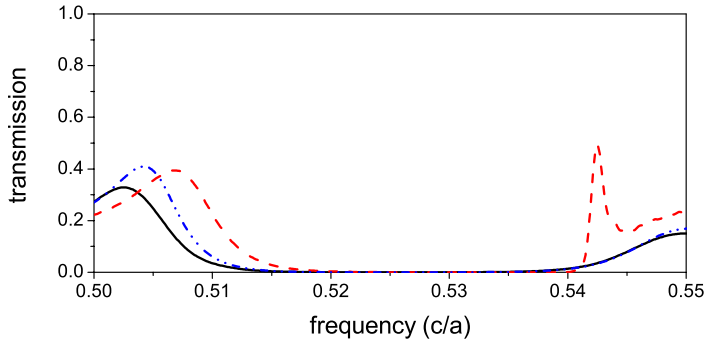


Fig. 5. Transmission spectrum for coupled photonic crystal slabs. The first slab has a dielectric constant of 12, a radius of $0.4a$, where a is the lattice constant, and a thickness of 0.55a. The second slab has the same dielectric constant as the first slab, with a radius $0.35a$ and thickness of $0.2a$. The solid line is for normal incident light, the dashed line is for *TM* polarization, and the dash-dotted line is for *TE* polarization with incident angle 11^o near frequency $0.53(c/a)$.

other to give near 100% reflection for an angular range up to 11^o for both *TE* and *TM* polarized light as predicted in the theory.

As final remarks, the photonic crystal slab mirror, which consists of only two single dielectric layers, is far more compact than traditional multi-layer film structures commonly used, where the use of up to 100 dielectric layers is often required to accomplish a Q-factor of a few thousands with a desired line shape. In addition, such a structure represents an attractive application of photonic crystal structures, as the structure functions with a relatively large optical aperture, and does not suffer from the insertion loss problems that often limit the practical use of 2-D photonic crystal waveguides. We therefore expect these novel and compact devices to be useful in optical communication systems.

This work was supported by the US Army Research Laboratories under Contract No. DAAD17-02-C-0101. The computational time was provided by the NSF National Resource Allocation Committee (NRAC), and the IBM-SUR program.

References

[1] M. Kanskar, P. Paddon, V. Pacradouni, R. Morin, A. Busch, J. F. Young, S. R. Johnson, J. MacKenzie, and T. Tiedje, "Observation of leaky slab modes in an air-bridged semiconductor waveguide with a two-dimensional photonic lattice," *Appl. Phys. Lett.*, vol. 70, no. 11, pp. 1438-40, 1997.

[2] V. N. Astratov, I. S. Culshaw, R. M. Stevenson, D. M. Whittaker, M. S. Skolnick, T.F. Krauss, and R.M. De La Rue, "Resonant coupling of near-infrared radiation to photonic band structure waveguides," *J. Lightwave Technol.*, vol. 17, no. 11, pp. 2050-7, 1999.

[3] Shanhui Fan and J. D. Joannopoulos, "Analysis of guided resonances in photonic crystal slabs," *Phys. Rev. B*, vol. 65, no. 23, pp. 235112-8, 2002.

[4] W. Suh, M. F. Yanik, O. Solgaard, and S. Fan, "Displacement-sensitive photonic crystal structures based on guided resonance in photonic crystal slabs," *Appl. Phys. Lett.*, vol. 82, no. 13, pp. 1999-2001, 2003.

[5] V. Lousse, W. Suh, O. Kilic, S. Kim, O. Solgaard, and S. Fan, "Angular and polarization properties of a photonic crystal slab mirror," *Opt. Express*, vol. 12, pp. 1575-1582, 2004.

[6] H. A. Haus, *Waves and fields in optoelectronics*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[7] K. S. Kunz and R. J. Luebbers, *The finite difference time domain method for electromagnetics*, CRC Press, Boca Raton, 1993; A. Taflove and S. C. Hagness, *Computational electrodynamics : the finite-difference time-domain method*, 2nd ed. Artech House, Boston, 2000.

Rate-Distortion Optimized Video Streaming with Rich Acknowledgments

Jacob Chakareski

Abstract. We consider an unconventional procedure for communicating to the server the receipt of media packets for Internet video streaming. Instead of separately acknowledging each media packet as it arrives, we periodically send to the server a single acknowledgment packet, denoted *rich acknowledgment*, that contains information about all media packets that have arrived at the client by the time the rich acknowledgment is sent. We investigate rate-distortion optimized sender-driven streaming that employs rich acknowledgments. Performance gains of up to 1.3 dB for streaming packetized video content are observed over rate-distortion optimized sender-driven systems that employ conventional acknowledgments.

Index Terms: rate-distortion optimization, video streaming, media communication and networking, rich acknowledgments.

I. Introduction

We consider the problem of rate-distortion optimized video streaming from a server to a client over a lossy packet network using rich feedback from the client. Packets may be lost in the network due to congestion or erasures. In addition, packets arriving late are also considered lost. Currently, in sender-driven transmission schemes employed for streaming media the client replies with an acknowledgment packet whenever a media packet arrives. The purpose of the acknowledgment packet is to inform the server that the client has received the corresponding media packet and that the server does not need to consider retransmitting that media packet again.

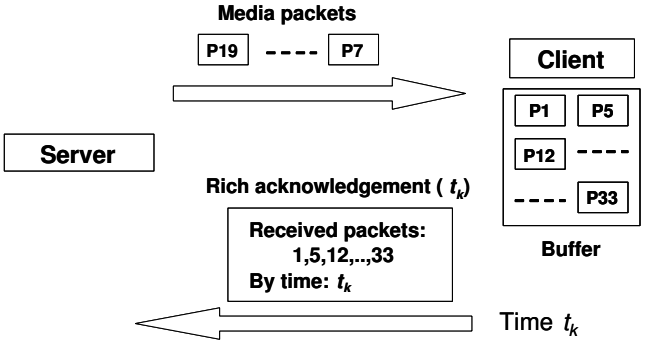


Fig. 1. Streaming on demand using rich acknowledgments.

In the present work, we employ an unconventional procedure for communicating to the server the receipt of media packets. Instead of separately acknowledging each media packet as it arrives, we periodically send to the server a single acknowledgment packet, denoted henceforth *rich acknowledgment*, that contains information about all media packets that have arrived at the client by the time the rich acknowledgment is sent. This information in essence reflects the current state of the client's buffer, i.e., which packets have been received by the client thus far. The proposed scenario of streaming with rich acknowledgments is illustrated in Figure 1.

It should be noted that the concept of rich acknowledgments is not new and has been introduced in the networking community under the name of vector acknowledgments. However, to the best of our knowledge rich acknowledgments have not been explored yet in media streaming. The purpose of vector acknowledgments is to allow a TCP sender to perform selective retransmission of lost data packets, which is not provided by the acknowledgment scheme employed by TCP, called cumulative ack (CACK) [1]. In essence, vector acks are binary maps that describe the correctly received or missing data in the receiver's buffer and have been adopted into several proposed feedback schemes [2–4]. In addition, vector acknowledgements have been included as an option in the recently proposed Datagram Congestion Control Protocol (DCCP) [5], which implements a congestion-controlled, unreliable flow of datagrams suitable for use by applications such as streaming media, Internet telephony and on-line games. Another alternative for providing selective retransmission in TCP is the selective ack option (SACK) [6], which allows a receiver to communicate simultaneously the identities of several contiguous blocks of successfully received data.

The contributions of this paper can be summarized as follows. First, it introduces the concept of rich feedback, i.e., rich acknowledgments in streaming of packetized media. Second, it provides a framework for rate-distortion optimized scheduling of the packet transmissions that employs rich acknowledgments as a feedback scheme. Rate-Distortion Optimized (RaDiO) packet scheduling is one of the latest advances in media streaming. In this approach, the media server or the client is equipped with a rate-distortion optimization framework for scheduling the packet transmissions such that a constraint on the average transmission rate is met while minimizing at the same time the average end-to-end distortion. In [7–9] the authors have introduced a framework for distortion-rate optimized scheduling of the transmissions of packetized media and applied it to the scenario of sender-driven streaming. In [10] the authors have employed this framework to study the scenario of receiver-driven transmission over best-effort networks. Rate-distortion optimized streaming over lossy packet networks to wireless clients, again in a sender-driven scenario, has been studied in [11–13]. In addition, in [14, 15] a streaming system is introduced, called *RaDiO Edge*, centered around a proxy server, located at the edge of the backbone network, and equipped with a RaDiO packet scheduling procedure and a hybrid receiver/sender driven transmission. Finally, in [16] the authors present a framework for rate-distortion optimized sender-driven streaming with path diversity, while in [17] the authors consider rate-distortion optimized receiver-driven streaming with server diversity. All these works have demonstrated a substantial improvement in performance over current state of the art solutions.

Media packets are typically characterized by different deadlines, importances and interdependencies. Using this information and the framework presented in this paper, the sender is able to transmit its media packets based on the rich acknowledgments it receives in a rate-distortion optimized way, that is, minimizing the expected end-to-end distortion subject to a constraint on the expected transmission rate. Such a rate-distortion optimized transmission algorithm, or transmission policy, results in unequal error protection provided to different portions of the media stream.

We present the major ideas in our paper as follows. In Section II, we define our abstractions of the encoding, packetization, and communication processes. In Section III, we explain the proposed protocol or algorithm for streaming packetized media that employs rich acknowl-

The author is with the Information Systems Laboratory at Stanford. E-mail: cakarz@stanford.edu

edgments as a feedback scheme. Next, in Section IV we show how the entire media presentation can be transmitted in a rate-distortion optimized way, using as a building block an algorithm for rate-distortion optimized transmission of a single media packet. This algorithm is the subject of Section V. In Section VI, we report our experimental results. Finally, concluding remarks are provided in Section VII.

II. Source and Channel Characterizations

A. Media Source Model

In a streaming media system, the encoded data are packetized into *data units* and are stored in a file on a media server. All of the data units in the presentation have interdependencies, which can be expressed by a directed acyclic graph. Associated with each data unit l is a size B_l, a decoding time $t_{DTS,l}$, a set of data units $\mathcal{N}_c^{(l)}$ and an importance $\Delta d_l^{(l_1)}$. Specifically, the size B_l is the size of the data unit in bytes. $t_{DTS,l}$ is the *delivery deadline* by which data unit l must arrive at the client, or be too late to be usefully decoded. Packets containing data units that arrive after the data units' delivery deadlines are discarded. Finally, $\mathcal{N}_c^{(l)}$ is the set of data units that the receiver considers for error concealment in case data unit l is not decodable by the receiver on time, while $\Delta d_l^{(l_1)}$, for $l_1 \in \mathcal{N}_c^{(l)}$, is the reduction in reconstruction error (distortion) for the media presentation if data unit l is not decodable and is concealed with data unit l_1 that is received and decoded on time.

B. Packet Loss Probabilities

The forward and the backward channel on a network path between a server and a client are characterized as independent time-invariant packet erasure channels with random delay. Hence, they are completely specified with the probabilities of packet loss ϵ_F and ϵ_B, and the probability densities of the transmission delay p_F and p_B, respectively. This means that if the media server sends a packet on the forward channel at time t, then the packet is lost with probability ϵ_F. However, if the packet is not lost, then it arrives at the client at time t', where the forward trip time $FTT = t' - t$ is randomly drawn according to the probability density p_F. Therefore, we let $P\{FTT > \tau\} = \epsilon_F + (1-\epsilon_F)\int_\tau^\infty p_F(t)dt$ denote the probability that a packet transmitted by the server at time t does not arrive at the client application by time $t+\tau$, whether it is lost in the network or simply delayed by more than τ. Then similarly, $P\{BTT > \tau\} = \epsilon_B + (1-\epsilon_B)\int_\tau^\infty p_B(t)dt$ denotes the probability that a rich acknowledgment packet transmitted by the client at time t does not arrive at the server by time $t+\tau$, whether it is lost in the network or simply delayed by more than τ.

III. Media Communication using Rich Acknowledgments

A media session starts when a client requests a presentation from the media server. Once the request packet is received, the media server starts sending packets with data units from the presentation at discrete transmission opportunities $t_i, t_{i+1}, \ldots$. Note that the transmission decisions regarding when and how often each of the data units will be sent to the client are completely determined by the server's transmission policy. The client periodically monitors the status of its buffer at every t_i and returns to the server this information via a single acknowledgment packet, i.e., a rich acknowledgment. The buffer information basically informs the server what data units have arrived at the client by the time (t_i) the rich acknowledgment was transmitted. Based on this information and the optimization framework presented in the next section, the server then dynamically decides at every transmission opportunity t_i what is at that moment the best transmission policy for every data unit in the presentation.

Note that in practice the server computes a sliding window of data units $\mathcal{W}_i$ at every t_i. Only the data units from $\mathcal{W}_i$ are considered for transmission at t_i. Therefore, at every t_i the client needs to send to the server information on the arrival status only for the data units in $\mathcal{W}_i$. In essence, this information comprises a binary vector $\boldsymbol{r}^i$ of length w_i, where $\boldsymbol{r}_j^i = 1(0)$ means that the j^{th} data unit from $\mathcal{W}_i$ has (not) arrived at the client by t_i and w_i is the length of the window $\mathcal{W}_i$ in data units. Then, from the time stamp t_i of a received rich acknowledgment the server can recompute the transmission window at that point (t_i) and thus can easily determine to which data units the information in $\boldsymbol{r}^i$ refers to. Finally, note that the client needs to know how the server computes its transmission window at every t_i. This knowledge can be provided to the client by the server at the beginning of the streaming session or it is simply fixed and known ahead of time.

IV. Rate-distortion optimized policy selection

Suppose there are L data units in the media presentation. Let $\pi_l \in \Pi$ be the transmission policy for data unit $l \in \{1, \ldots, L\}$ and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ be the vector of transmission policies for all L data units. Π is a family of policies defined precisely in the next section.

Any given policy vector $\boldsymbol{\pi}$ induces an expected distortion $D(\boldsymbol{\pi})$ and an expected transmission rate $R(\boldsymbol{\pi})$ for the media presentation. We seek the policy vector $\boldsymbol{\pi}$ that minimizes $D(\boldsymbol{\pi})$ subject to a constraint on $R(\boldsymbol{\pi})$. This can be achieved by minimizing the Lagrangian $D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$ for some Lagrange multiplier $\lambda > 0$, thus achieving a point on the lower convex hull of the set of all achievable distortion-rate pairs.

We now compute expressions for $R(\boldsymbol{\pi})$ and $D(\boldsymbol{\pi})$. The expected transmission rate $R(\boldsymbol{\pi})$ is the sum of the expected number of bytes transmitted for each data unit $l \in \{1, \ldots, L\}$, $R(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l)$, where B_l is the number of bytes in data unit l and $\rho(\pi_l)$ is the expected number of transmitted bytes per source byte (under policy π_l), called the *expected cost*. The expected distortion $D(\boldsymbol{\pi})$ can be expressed in terms of the probability $\epsilon(\pi_l)$ that data unit l does not arrive at the receiver on time (under policy π_l), called the *expected error*. We borrow the expression for $D(\boldsymbol{\pi})$ from [16]

$$D(\boldsymbol{\pi}) = D_0 - \sum_l \sum_{l_1 \in \mathcal{N}_c^{(l)}} \Delta d_l^{(l_1)} \prod_{j \in \mathcal{A}(l_1)} (1 - \epsilon(\pi_j)) \times \prod_{l_2 \in \mathcal{C}(l,l_1)} \left(1 - \prod_{l_3 \in \mathcal{A}(l_2) \setminus \mathcal{A}(l_1)} (1 - \epsilon(\pi_{l_3}))\right) \quad (1)$$

where D_0 is the expected reconstruction error for the presentation if no data units are received. $\mathcal{A}(l_1)$ is the set of ancestors of l_1, including l_1. $\mathcal{C}(l, l_1)$ is the set of data units $j \in \mathcal{N}_c^{(l)} : j > l_1$ that are not mutual descendants, i.e., for $j, k \in \mathcal{C}(l, l_1) : j \notin \mathcal{D}(k), k \notin \mathcal{D}(j)$, where $\mathcal{D}(j)$ is the set of descendants of data unit j. "\" denotes the operator "set difference".

Finding a policy vector $\boldsymbol{\pi}$ that minimizes the expected Lagrangian $J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$, for $\lambda > 0$, is difficult since the terms involving the individual policies π_l in $J(\boldsymbol{\pi})$ are not independent. Therefore, we employ an iterative descent algorithm, called Iterative Sensitivity Adjustment (ISA), in which we minimize the objective function $J(\pi_1, \ldots, \pi_L)$ one variable at a time while keeping the other variables constant, until convergence [7]. It can be shown that the optimal individual policies at iteration n, for $n = 1, 2, \ldots$, are given by

$$\pi_l^{(n)} = \arg\min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \lambda B_l \rho(\pi_l), \quad (2)$$

where $S_l^{(n)} = \sum_{l_1 \,:\, l \in \mathcal{N}_c^{(l_1)}} S_{l,l_1}^{+(n)} - S_{l,l_1}^{-(n)} = S_l^{+(n)} - S_l^{-(n)}$ can be regarded as the *sensitivity* to losing data unit l, i.e., the amount by which the expected distortion will increase if data unit l cannot be recovered at the client, given the current transmission policies for the other data

units. Note that differently from [7], the sensitivity here consists of two nonnegative terms $S_l^{+(n)}$ and $S_l^{-(n)}$. The first term increases the sensitivity associated with data unit l in case l is in the ancestor set of data unit l_2 used for concealment of a data unit l_1. On the other hand, the second term reduces the sensitivity associated with l in case l is not in the ancestor set of l_2. This result is intuitive and allows us to better model the situations where data unit l is irrelevant for concealment of another data unit. Expressions for $S_{l,l_1}^{+(n)}$ and $S_{l,l_1}^{-(n)}$ are easily obtained from (1) by grouping terms.

The minimization (2) is now simple, since each data unit l can be considered in isolation. Indeed the optimal transmission policy $\pi_l^* \in \Pi$ for data unit l minimizes the "per data unit" Lagrangian $\epsilon(\pi_l)+\lambda'\rho(\pi_l)$, where $\lambda' = \lambda B_l/S_l^{(n)}$. In the next section, we show how to find π^* for the family of transmission policies Π corresponding to sender-driven streaming with rich acknowledgments.

V. Computing the Optimal Transmission Policy

For transmitting a single data unit on the forward channel, we assume that there are N discrete transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to the data unit's delivery deadline t_{DTS} at which the server considers transmitting a packet for the data unit. The server need not transmit a packet at every transmission opportunity. The server does not transmit any packets after a rich acknowledgment is received confirming the receipt of the data unit at the client. In addition, as explained in Section III, the client sends a rich acknowledgment packet to the server on the backward channel at every t_i notifying the server about the state of its buffer, i.e., which data units have arrived at the client by t_i. Note that at the same time this also informs the server which data units have not arrived at the client by t_i. Therefore, the information provided by a received rich acknowledgment is much richer than that provided by a received conventional acknowledgment.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the server takes an action a_i, where $a_i = 1$ if the server sends a packet and $a_i = 0$ otherwise. Then, at the next transmission opportunity t_{i+1}, the server makes an observation o_i, where o_i is the set of rich acknowledgments received by the server in the interval $(t_i, t_{i+1}]$. For example, $o_i = \{NAK(t_1), ACK(t_3)\}$ means that during the interval $(t_i, t_{i+1}]$, the rich acknowledgment sent at t_1 arrived at the server informing that the data unit has not been received by the client by t_1 ($NAK(t_1)$) and that the rich acknowledgment sent at t_3 arrived at the server informing that the data unit has been received by the client by t_3 ($ACK(t_3)$). Note that for the purposes of our algorithm it is irrelevant for the server to distinguish the transmission times of the rich acknowledgments received within $(t_i, t_{i+1}]$ confirming that the data unit has arrived at the client by their respective transmission times. As explained above, once the server receives a confirmation that the data unit has arrived at the client, it stops sending any packets afterwards, regardless of the transmission times of the rich acknowledgments that brought that confirmation. Therefore, we drop the timing notation on these rich acknowledgments and simply use ACK to denote the event that at least one rich acknowledgment has been received confirming the receipt of the data unit due to previous transmissions. In addition, we denote the event $o_i = \{NAK(t_1), ACK\}$ simply as $o_i = ACK$ since receiving any NAKs together with at least one ACK will not affect the transmission actions that the server considers afterwards. Again, as we explained earlier, the server does not transmit any packets with the data unit after an ACK has been received, regardless of any number of NAKs that have also arrived during $(t_i, t_{i+1}]$. Finally, we denote the event $o_i = \{NAK(t_1), NAK(t_3)\}$ simply as $o_i = NAK(t_3)$ since not receiving the data unit by t_3 implies that the data unit was certainly not received by t_1. In other words, we denote the events o_i, when multiple NAKs are received as an observation, using only the most recently sent NAK.

The history, or the sequence of action-observation pairs $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_i, o_i)$ leading up to time t_{i+1}, determines the state q_{i+1} at time t_{i+1}, as illustrated in Figure 2. Therefore, a state represents uniquely this sequence of action-observation pairs. If the final observation o_i includes an ACK, then q_{i+1} is a final state. In addition, any state at time $t_N = t_{DTS}$ is a final state. Final states in Figure 2 are indicated by double circles.

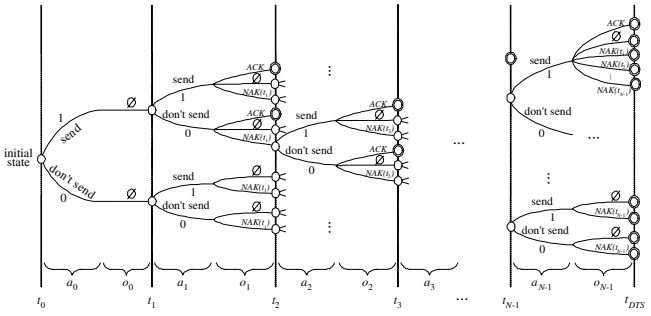


Fig. 2. State space for a Markov decision process.

The action a_i taken at a non-final state q_i determines the transition probability $P(q_{i+1}|q_i, a_i)$ to the next state q_{i+1}. Formally, a policy π is a mapping $q \mapsto a$ from non-final states to actions. Thus any policy π induces a Markov tree with transition probabilities between states $P_\pi(q_{i+1}|q_i) \equiv P(q_{i+1}|q_i, \pi(q_i))$, and consequently also induces a probability distribution on final states. Let q_F be a final state with history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{F-1}, o_{F-1})$, and let $q_{i+1} = q_i \circ (a_i, o_i)$, $i = 1, \ldots, F-1$, be the sequence of states leading up to q_F. Then q_F has probability $P_\pi(q_F) = \prod_{i=0}^{F-1} P_\pi(q_{i+1}|q_i)$, transmission cost $\rho_\pi(q_F) = \sum_{i=0}^{F-1} a_i$, and error $\epsilon_\pi(q_F) = 0$ if o_{F-1} contains an ACK and otherwise $\epsilon_\pi(q_F)$ is equal to the probability that none of the packets transmitted under policy π results in successful decoding by time t_{DTS}, given q_F. For example, if q_F is the second state from the top at time t_{DTS} in Figure 2, then a packet with the data unit was sent at every transmission opportunity $t_0, t_1, \ldots, t_{N-1}$ and no rich acknowledgments were received. In that case, $\epsilon_\pi(q_F) = \prod_{i=0}^{N-1} P\{FTT > t_{DTS} - t_i\}$. Therefore, Π is a collection of all possible trees in the state space described in Figure 2.

We can now express the expected cost and error for the Markov tree induced by policy π: $\rho(\pi) = E_\pi \rho_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\rho_\pi(q_F)$, $\epsilon(\pi) = E_\pi \epsilon_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\epsilon_\pi(q_F)$. As stated earlier we are interested in finding the policy π^* that minimizes the expected Lagrangian cost

$$J(\pi) \equiv \epsilon(\pi) + \lambda'\rho(\pi) = \sum_{q_F} P_\pi(q_F) J_\pi(q_F), \tag{3}$$

where $J_\pi(q_F) \equiv \epsilon_\pi(q_F) + \lambda'\rho_\pi(q_F)$. We compute π^* using a dynamic programming [18–21] algorithm as follows. Let

$$J_\pi(q_i) = \begin{cases} \epsilon_\pi(q_F) + \lambda'\rho_\pi(q_F) \\ \qquad \text{if } q_i \text{ is final } (i = F), \\ \sum_{q_{i+1}} P(q_{i+1}|q_i, \pi(q_i)) J_\pi(q_{i+1}) \\ \qquad \text{otherwise} \end{cases}$$

be the expected Lagrangian of all paths through q_i (under π). Then let

$$J^*(q_i) = \begin{cases} \epsilon_\pi(q_F) + \lambda'\rho_\pi(q_F) \\ \qquad \text{if } q_i \text{ is final } (i = F), \\ \min_{a_i} \sum_{q_{i+1}} P(q_{i+1}|q_i, a_i) J^*(q_{i+1}) \\ \qquad \text{otherwise.} \end{cases} \tag{4}$$

By induction, $J^*(q_i) \le J_\pi(q_i)$ for all q_i and all π, with equality if $\pi = \pi^*$, where

$$\pi^*(q_i) = \arg\min_{a_i} \sum_{q_{i+1}} P(q_{i+1}|q_i, a_i) J^*(q_{i+1}) \tag{5}$$

for all non-final states q_i. Thus the optimal policy (minimizing (3)) can be computed efficiently using (4) and (5).

Next, we provide the actual expressions for $\epsilon_\pi(q_F), \rho_\pi(q_F)$ and $P(q_{i+1}|q_i, a_i)$ used in the equations above. As given before, the transmission cost $\rho_\pi(q_F) = \sum_{i=0}^{F-1} a_i$. Now, if no NAKs have been received along the path that leads to q_F in the state space from Figure 2, then the transmission error is $\epsilon_\pi(q_F) = \prod_{i\,:\,a_i=1} P\{FTT > t_{DTS} - t_i\}$. On the other hand, if at least one NAK has been received, then the transmission error is

$$\epsilon_\pi(q_F) = \prod_{i\,:\,i<j,\ a_i=1} P\{FTT > t_{DTS} - t_i | FTT > t_j - t_i\} \times \prod_{i\,:\,j\le i,\ a_i=1} P\{FTT > t_{DTS} - t_i\},$$

where $j \in \{1, \ldots, N-1\}$ is the index of the most recently sent NAK that has arrived at the server, i.e., $NAK(t_j)$. Finally, we need to differentiate 4 possible cases for the transition probability $P(q_{i+1}|q_i, a_i)$. As mentioned earlier, $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_i, o_i)$ is the history, or the sequence of action-observation pairs that leads to state q_{i+1} at time t_{i+1}. Then, we can have

(a) no NAK received along the path that leads to q_i, i.e., $NAK \notin \bigcup_{j=0}^{i-1} o_j$ and no NAK received in $(t_i, t_{i+1}]$, i.e., $NAK \notin o_i$

$$P(q_{i+1}|q_i, a_i) = \prod_{l=1}^{i} P\{BTT > t_{i+1} - t_l | BTT > t_i - t_l\}, \quad (6)$$

(b) no NAK received along the path that leads to q_i, i.e., $NAK \notin \bigcup_{j=0}^{i-1} o_j$ and at least one NAK received during $(t_i, t_{i+1}]$, i.e., $NAK \in o_i$

$$P(q_{i+1}|q_i, a_i) = P\{t_i - t_k < BTT \le t_{i+1} - t_k | BTT > t_i - t_k\} \times \prod_{l>k}^{i} P\{BTT > t_{i+1} - t_l | BTT > t_i - t_l\} \times \prod_{l\,:\,l<k,\ a_l=1} P\{FTT > t_k - t_l\}, \quad (7)$$

(c) at least one NAK received along the path that leads to q_i, i.e., $NAK \in \bigcup_{j=0}^{i-1} o_j$ and no NAK received during $(t_i, t_{i+1}]$, i.e., $NAK \notin o_i$

$$P(q_{i+1}|q_i, a_i) = \prod_{l>j}^{i} P\{BTT > t_{i+1} - t_l | BTT > t_i - t_l\}, \quad (8)$$

(d) at least one NAK received along the path that leads to q_i, i.e., $NAK \in \bigcup_{j=0}^{i-1} o_j$ and at least one NAK received during $(t_i, t_{i+1}]$, i.e., $NAK \in o_i$

$$P(q_{i+1}|q_i, a_i) = P\{t_i - t_k < BTT \le t_{i+1} - t_k | BTT > t_i - t_k\} \times \prod_{l>k}^{i} P\{BTT > t_{i+1} - t_l | BTT > t_i - t_l\} \times \prod_{l\,:\,l<k,\ a_l=1} P\{FTT > t_k - t_l | FTT > t_j - t_l\}, \quad (9)$$

where j in (8) and (9) is the index of the most recently sent NAK received up to t_i, i.e., $NAK(t_j) \in \bigcup_{l=0}^{i-1} o_l$. Similarly, k in (7) and (9) is the index of the most recently sent NAK received during $(t_i, t_{i+1}]$, i.e., $NAK(t_k) \in o_i$. Note that in (9), it necessarily holds $j < k$ and $P\{FTT > t_k - t_l | FTT > t_j - t_l\} = P\{FTT > t_k - t_l\}$ for $j \le l$.

VI. Experimental Results

Here, we investigate the end-to-end distortion-rate performance for streaming packetized video content using different algorithms. The videos used in the experiments are two-layer SNR scalable representations of the image sequences *Foreman* and *Mother and Daughter*, henceforth denoted *MaD*. Using H.263+ [22] 130 frames of QCIF *Foreman* have been encoded into a base layer and an enhancement layer with corresponding rates of 32 and 64 Kbps. Similarly, 130 frames of QCIF *MaD* have been encoded into two layers with rates 32 and 69 Kbps, respectively. For both videos the frame rate is 10 fps and the size of the Group of Pictures (GOP) is 10 frames, consisting of an I frame followed by 9 consecutive P frames. Two RaDiO streaming systems are employed in the experiments. *Conv. ACK* is a system that performs RaDiO scheduling using conventional acknowledgments [7–9]. *Rich ACK* is the system presented in this work, which also performs RaDiO packet scheduling, but using rich acknowledgments. The Lagrange multiplier λ is fixed for the entire presentation for both systems. Performance is measured in terms of the luminance peak signal-to-noise ratio (Y-PSNR) in dB of the end-to-end perceptual distortion, averaged over the duration of the video clip, as a function of the average transmission rate (Kbps) on the forward channel. In the experiments we use $T = 100$ ms as the time interval between transmission opportunities and 600 ms for the playback delay.

The forward and the backward channel on the network path between the server and the client are specified as follows. Packets transmitted on these channels are dropped at random, with a drop rate $\epsilon_F = \epsilon_B = \epsilon = 10$ %. Those packets that are not dropped receive a random delay, where for the forward and the backward delay densities p_F and p_B we use identical shifted Gamma distributions with parameters (n, α) and right shift κ, where $n = 2$ nodes, $1/\alpha = 25$ ms, and $\kappa = 50$ ms for a mean delay of $\kappa + n/\alpha = 100$ ms and standard dev. $\sqrt{n}/\alpha \approx 35$ ms.

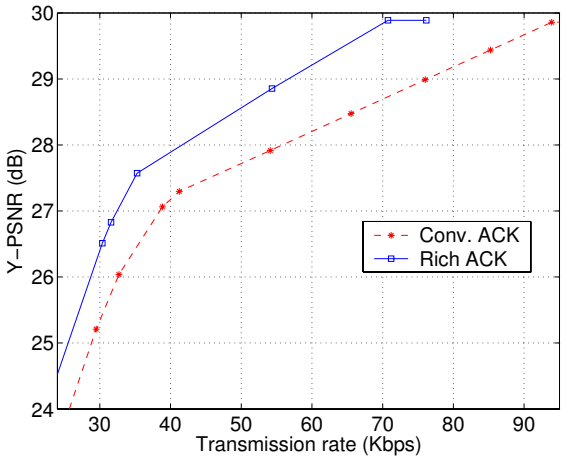


Fig. 3. Rich vs. Conv. ACKs for streaming *Foreman*.

It can be seen from Figure 3 that given the selected simulation parameters using rich acknowledgments can improve performance for streaming *Foreman*. *Rich ACK* performs consistently better than *Conv. ACK* over all transmission rates under consideration. The performance gains reach up to 1.3 dB for a transmission rate of 70 Kbps, or equivalently, transmission rate savings of 27 % are observed for the given PSNR of 29.9 dB. The difference in performance between the two systems is due to the fact that rich acknowledgments can make up for losses of individual acknowledgment packets in the *Conv. ACK* system, as shown next. In addition, they also provide the server with a much richer knowledge of the state of the client's buffer than that provided by conventional acknowledgments, as explained on the beginning of Section V. Consequently, the server is able to exploit this information to its benefit and therefore to provide enhanced performance over a rate-distortion optimized system that only employs conventional acknowledgment packets. In essence, the transmission policies computed based on the knowledge provided by rich acknowledgments are more efficient in a rate-distortion sense than those computed based on conventional acknowledgments.

We observe a similar situation for streaming *MaD* as seen from Figure 4. *Rich ACK* outperforms *Conv. ACK* over the whole range of available transmission rates. The performance gains in this case reach up

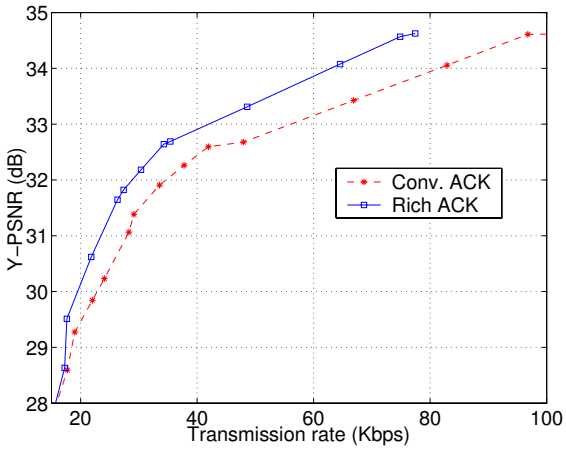


Fig. 4. Rich vs. Conv. ACKs for streaming *MaD*.

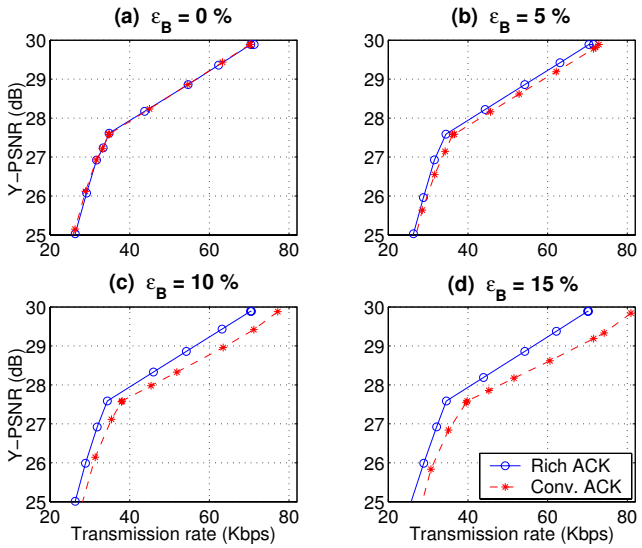


Fig. 5. Y-PSNR (dB) vs. Transmission rate (Kbps) for streaming *Foreman*.

to 0.9 dB for transmission rate of 70 Kbps, or equivalently, transmission rate savings of 22.2 % are observed for the given PSNR of 34.3 dB. Note, however, that the performance gains in this case are not as large as those for *Foreman*. This is due to the nature of the sequence *MaD*, which exhibits comparably less motion than *Foreman*. Therefore, the drop in quality incurred by a lost or late packet is not as significant for *MaD* as is for *Foreman* since error concealment can be performed more successfully in the case of the former.

Next, we compare the efficiencies of the two streaming systems in terms of the transmission rate on the forward channel, as the packet loss rate increases on the backward channel and remains fixed on the forward channel. The forward and backward delay densities p_F and p_B are specified with the following parameter values: $n = 1$ node, $1/\alpha = 25$ ms, and $\kappa = 20$ ms for a mean delay of $\kappa + n/\alpha = 50$ ms and standard deviation $\sqrt{n}/\alpha \approx 25$ ms. Figure 5 shows the Y-PSNR for the *Foreman* sequence as a function of the transmission rate on the forward channel, when $\epsilon_B = 0\%$, 5%, 10%, and 15%, while $\epsilon_F = 10\%$. It can be seen that as the backward channel degrades, the transmission rate on the forward channel of *Rich ACK* remains approximately constant (for any given Y-PSNR), while the transmission rate of *Conv. ACK* increases. Thus the gap between them, which is essentially zero when $\epsilon_B = 0\%$, increases as the packet loss rate on the backward channel increases. This is because as the backward channel degrades, *Conv. ACK* increasingly and unnecessarily retransmits information over the forward channel, due to loss of acknowledgment packets on the backward channel. Note that *Rich ACK* avoids this by acknowledging every arriving media packet multiple times by consecutive rich acknowledgments.

We find that the performance of both *Rich ACK* and *Conv. ACK* can be accurately predicted from the distortion-rate function of the source and the values of ϵ_F and ϵ_B. When the number of retransmission opportunities before the playout deadline is sufficiently large, both systems transmit close to channel capacity. As is well known [23], the capacity of an erasure channel with erasure probability ϵ is $1-\epsilon$. Thus an optimal system transmits $1/(1-\epsilon)$ channel packets for every data unit. In particular, since *Conv. ACK* continues to transmit packets until it receives an acknowledgment, which it receives with probability $(1-\epsilon_F)(1-\epsilon_B)$ for each transmitted packet, then *Conv. ACK* transmits on average $1/[(1-\epsilon_F)(1-\epsilon_B)]$ packets per data unit over the forward channel. On the other hand, *Rich ACK* transmits on average only $1/[(1-\epsilon_F)(1-\epsilon_B^K)]$ packets per data unit on the forward channel, where K is the number of rich acknowledgments sent after the data unit arrives at the client. Note that for the range of values for ϵ_B considered here, this can be approximately written as $1/(1-\epsilon_F)$ even for very small values of K. Therefore, rich acknowledgments essentially cancel out the effect of packet loss on the backward channel.

To confirm that these factors accurately predict performance, we normalize the transmission rates on the forward channel for all the graphs in Figure 5 with $1/[(1-\epsilon_F)(1-\epsilon_B)]$ for *Conv. ACK* and with $1/(1-\epsilon_F)$ for *Rich ACK* and we plot them (in gray) in Figure 6, along with the

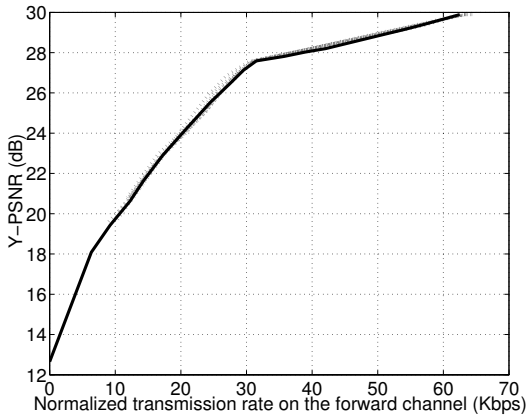


Fig. 6. Y-PSNR (dB) vs. Normalized transmission rate (Kbps) for *Foreman*.

distortion-rate function of the source (in bold). It can be seen that all graphs lie essentially on top of the distortion-rate function for the source, meaning that we can effectively synthesize the graphs in Figure 5 by multiplying the rate-distortion function by the appropriate factors for *Conv. ACK* and *Rich ACK*.

Finally, we study the transmission rates of acknowledgments for both systems which simply comprise the amount of acknowledgment data sent to the client during a streaming session. In the case of *Conv. ACK* each acknowledgment packet contains only a header and no payload. On the other hand, a rich acknowledgment packet sent at transmission opportunity t_i consists of both a header and a payload of size w_i bits, where w_i is the number of data units in the transmission window of the server at t_i, as explained in Section III. In our experiments, the header size in both systems is set to 320 bits. In Figure 7, we show the transmission rates of *Conv. ACK* and *Rich ACK* on the backward channel as a function of the transmission rate on the forward channel. It can be seen that for transmission rates greater than 40 Kbps, the rate of acknowledgments for *Conv. ACK* becomes larger than the corresponding

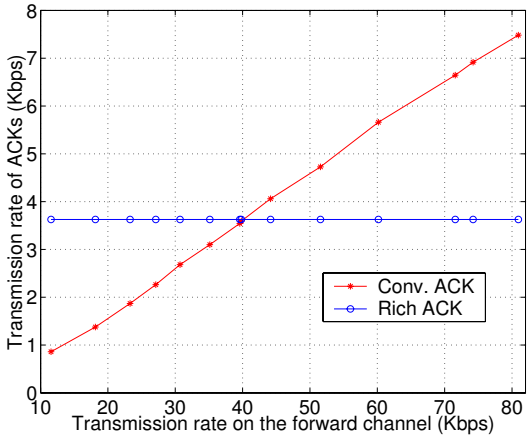


Fig. 7. Rate of ACKs (Kbps) vs. Transmission rate (Kbps) for *Foreman*.

rate for *Rich ACK*. For example, for transmission rates of 80 Kbps, *Rich ACK* saves 50 % of the bandwidth on the backward channel. Furthermore, note that the range of transmission rates $\geq$ 40 Kbps is also the range over which *Rich ACK* provides the most significant improvement in Y-PSNR performance, as shown in Figure 3.

We observed analogous results regarding the efficiencies of the two streaming systems for streaming *MaD*, as a function of the packet loss rate on the backward channel, and also in terms of the transmission rates of acknowledgements. These results are not included here due to space constraints. Nonetheless, the same discussion and conclusions that were exposed above for the case of *Foreman* also apply for *MaD*.

VII. Conclusions

We have presented a system for rate-distortion optimized sender-driven streaming of packetized media with rich acknowledgments. The system consists of two major components. A feedback scheme that periodically returns to the sender a single acknowledgment packet, called rich acknowledgment, that informs the sender of the state of the receiver's buffer at the time when the rich acknowledgment was transmitted. The second component of our system is an optimization framework that enables the sender to optimize in a rate-distortion sense its transmission policies based on the knowledge provided by received rich acknowledgments and on the knowledge of the media source and the communication channels. The computation of the optimal policies is done using a Markov decision tree with finite horizon N, associated with the transmission scenario under consideration. The proposed system provides significant gains in performance over rate-distortion optimized systems that employ conventional acknowledgements as a feedback scheme. Our experimental results demonstrate that the proposed framework performs favorably over a large range of conditions considered for the feedback (backward) channel. The gains in performance increase as the packet loss rate on the backward channel increases. An additional advantage of streaming with rich acknowledgments is that it provides significant savings in transmission bandwidth on the backward channel. Finally, it was also shown that the performance of the proposed system is quite insensitive to variations in the quality of the feedback channel. This can be exploited to reduce the frequency of ack packets in order to conserve bandwidth or energy, which can be extremely useful for two classes of networks: networks with asymmetric links where the relative cost of sending an ack is higher and wireless sensor networks that have stringent power constraints.

References

[1] W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Boston, MA: Addison-Wesley, 1994.

[2] B. Doshi, P. Johri, A. Netravali, and K. Sabnani, "Error and flow control performance of a high speed protocol," *IEEE Trans. Communications*, vol. 41, no. 5, pp. 707–720, May 1993.

[3] J. C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," in *Proc. Conf. on Computer Communications (INFOCOM)*, vol. 3. San Francisco, CA, USA: IEEE, Mar. 1996, pp. 1414–1424.

[4] H.-S. W. So, Y. Xia, and J. Walrand, "A robust acknowledgement scheme for unreliable flows," in *Proc. Conf. on Computer Communications (INFOCOM)*, vol. 3. New York, NY, USA: IEEE, June 2002, pp. 1500–1509.

[5] E. Kohler, M. Handley, S. Floyd, and J. Padhye, "Datagram Congestion Control Protocol (DCCP)," IETF, http://www.ietf.org/internet-drafts/draft-ietf-dccp-spec-05.txt, Internet-Draft, Oct. 2003.

[6] M. Mathis, J. Madhavi, S. Floyd, and A. Romanow, "TCP selective acknowledgment options," IETF, http://www.ietf.org/rfc/rfc2018.txt, Tech. Rep. RFC 2018, Oct. 1996, proposed standard.

[7] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, 2001, submitted.

[8] ——, "Rate-distortion optimized streaming of packetized media," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-2001-35, Feb. 2001.

[9] ——, "Rate-distortion optimized sender-driven streaming over best-effort networks," in *Proc. Workshop on Multimedia Signal Processing*. Cannes, France: IEEE, Oct. 2001, pp. 587–592.

[10] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *Proc. Int'l Packet Video Workshop*, Pittsburg, PA, Apr. 2002.

[11] J. Chakareski, P. Chou, and B. Aazhang, "Computing rate-distortion optimized policies for streaming media to wireless clients," in *Proc. Data Compression Conference*. Snowbird, UT: IEEE Computer Society, Apr. 2002, pp. 53–62.

[12] J. Chakareski and P. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 3. Orlando, FL: IEEE, May 2002, pp. 2513–2516.

[13] ——, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-2002-81, Aug. 2002, publicly available at ftp://ftp.research.microsoft.com/pub/tr/TR-2002-81.ps.

[14] J. Chakareski, P. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," in *Proc. Workshop on Multimedia Signal Processing*. St. Thomas, US Virgin Islands: IEEE, Dec. 2002, pp. 49–52.

[15] ——, "Computing rate-distortion optimized policies for hybrid receiver/sender driven streaming of multimedia," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, vol. 2. Pacific Grove, CA: IEEE, Nov. 2002, pp. 1310–1314.

[16] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conference*. Snowbird, UT: IEEE Computer Society, Mar. 2003, pp. 203–212.

[17] ——, "Server diversity in rate-distortion optimized streaming of multimedia," in *Proc. Int'l Conf. Image Processing*, vol. 3. Barcelona, Spain: IEEE, Sept. 2003, pp. 645–648.

[18] R. Bellman, *Dynamic Programming*. Princeton Univesity Press, 1957.

[19] R. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton Univesity Press, 1962.

[20] D. Bertsekas, *Dynamic Programming*. Prentice Hall, 1987.

[21] R. Parker and R. Rardin, *Discrete optimization*. Academic Press, 1988.

[22] Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *ITU-T Recommendation H.263 Version 2*, Feb. 1998.

[23] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 1991.

Frequency Coordination in the Amateur Radio Emergency Service

Leif J. Harcke, Kenneth S. Dueker, and David B. Leeson

Abstract. The selection of a new frequency for disaster communications use by the Stanford University community is examined in detail. Allocating frequencies in a voluntary radio service is constrained by band overcrowding, limited receiver selectivity, and small or non-existent budgets for regional coordination efforts. When presented with the challenge, individuals from the student radio club and the electrical engineering department developed a three phase plan to model, test, and deploy the use of a new narrowband VHF FM simplex channel. Computer modeling of the radio propagation environment in the San Francisco Bay Area identified several candidate frequencies for further study. Passive monitoring eliminated some of the initial choices by highlighting those exhibiting co-channel interference. An extended period of testing resulted in the final selection of a new channel. The frequency selected at the completion of the study has been in continual use since the fall of 2002 with no reports of co- or adjacent channel interference, demonstrating the effectiveness of the exercise.

I. Introduction

MANY people are familiar with the commercial radio services defined and regulated by the Federal Communications Commission. These include broadcast television and radio, and mobile telephone, data, and paging systems. Less well known radio services include land-mobile voice and data services used by construction firms, taxis, and delivery companies, and local government emergency services such as police, fire, and ambulance. The amateur radio service is often unnoticed among the above popular services, since amateur license grantees are primarily individuals, not corporations or civil agencies. The amateur service is defined by Title 47, Part 97 of the Code of Federal Regulations as having several fundamental purposes. The first principle listed in 47CFR97 is:

(a) Recognition and enhancement of the value of the amateur service to the public as a voluntary noncommercial communication service, particularly with respect to providing emergency communications [1].

Local and regional emergency communication networks are often overstressed and fail during large disastrous events. Several days after the onset of a disaster, temporary networks may be deployed by the Federal Emergency Management Agency (FEMA) or the National Guard to restore lost capacity. But local networks typically fail in the first few hours following a disaster, and the absence of communications capability for 24 to 48 hours can lead to unnecessary loss of life and/or increased damage to property. The causes of network failure include power outages exceeding equipment backup battery lifetimes, broken or downed wires, and overloading due to heavy traffic demands. A more subtle failure mechanism is the inability of emergency services from disparate locales to operate on the same frequency due to regulatory restrictions on co-channel operation which then become implemented in hardware. More succinctly, town A's fire company radios can't talk to town B's fire company radios when both towns respond to a large regional emergency.

Cellular telephone networks, while seemingly ideal for interagency communications, suffer from the fact that the network is designed to support one-to-one traffic, not one-to-many traffic necessary for deploying forces. Public safety communications generally uses a combination of point-to-point systems as well as "dispatch" radio systems, in which everyone on the network can hear the prime transmitting user. This avoids the need for relaying of messages and the corresponding possibility for misinterpretation. Civilian users tend to overload the cellular network beyond its statistical capabilities with welfare traffic. Even so-called push-to-talk cellular services do not provide true handset to handset capability. Both handsets must have access to usable base stations to make the connection.

Since the amateur service is composed of licensed individuals, they are often "first responders" in the event of a disaster, and are in the unique position provide local, regional, and national communications capabilities for the first crucial hours when communication networks fail and until backup systems can be installed and brought on-line. Once main networks are restored, amateur service operations are usually continued in support of volunteer relief agencies such as the Red Cross which do not maintain their own networks. This paper is primarily concerned with emergency communication planning in the amateur service for the Stanford campus, which is located in the San Francisco Bay Area of the state of California (Figure 1). The area is made up of nine large counties surrounding the bay and hosts a population of approximately 5.6 million people in the San Francisco–Oakland–San Jose extended metropolitan areas. The topography varies from sea level to near 1200 meters, and is a challenge for both broadcast service and land-mobile radio communications.

Two major disasters in the late 1980's and early 1990's, the 1989 Loma Prieta earthquake and the 1991 Oakland Hills fires, caused the cities and counties of the Bay Area to extensively review their emergency contingency plans. Though there are many aspects to city and county emergency planning, one overriding

Manuscript received April 18, 2004; revised May 17, 2004. L.J. Harcke and D.B. Leeson are with the Space, Telecommunications, and Radioscience Laboratory of the Electrical Engineering Department, Stanford University, Stanford, CA 94305. Email: {lharcke|leeson}@stanford.edu K.S. Dueker is with PowerFlare Corp., Palo Alto, CA 94301. Email: kdueker@powerflare.com

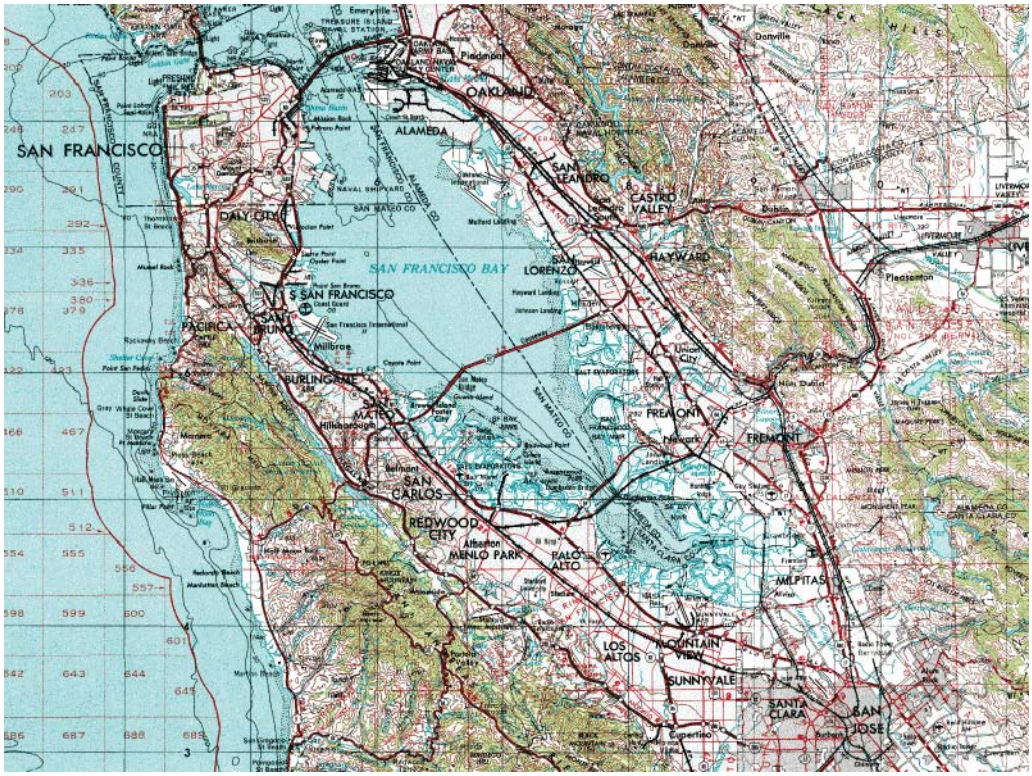


Fig. 1. San Francisco Bay Area topographic map. The Stanford University campus is located at the southwest corner of the bay near the town of Palo Alto. Significant mountain ranges (850m) lie several kilometers to the south and west of the campus, while hilly terrain (150-250m) is common in the immediate vicinity (Image credit: U.S. Geological Survey).

concern after the Oakland Hills fires was the inability of emergency services from different counties to communicate at the scene of the incident. This eventually led to the establishment of a California state-wide emergency communications plan which aspires to provide seamless inter-operation of communication networks at local, regional, and state levels.

The California Office of Emergency Services sponsors a one-day, state-wide emergency drill annually, while county and city emergency services drill more frequently on their own schedules. The amateur radio service participates in the annual state wide drill with other agencies, and drills independently at other times. Typical amateur service drills in the Bay Area are carried out weekly at the city level, monthly in support of regional hospitals, and quarterly on a county-wide basis. Most civil authorities recognize the capabilities of the amateur radio service, and amateur radio stations are installed in many city and county emergency operations centers. In some California counties, amateur radio service volunteers are categorized as "emergency responders," issued identification, and become unpaid employees of the county when activated for drills or real emergencies. In case of injury, these volunteers are covered by workman's compensation laws when deployed in hazardous situations.

Technical standards in both the civil services and the amateur radio service tend to follow a lowest common denominator approach. Though digital voice technologies provide increased capacity in commercial radio networks, disparate budget allocations between various city and county governments mean that the baseline communication capability for inter-operation is narrowband frequency modulation (FM) on the VHF and UHF radio bands. For example, Alameda County upgraded its communications facilities to a county-wide 800 MHz digital trunking system after the Oakland Hills fires, while Santa Clara County continues to use analog FM for most public safety radio services. An umbrella organization, the Association of Public-Safety Communications Officials – International, Inc. sponsors Project 25 in conjunction with the telecommunications industry as a migration path to all-digital communications in the public emergency network [2], but nationwide deployment of the technology is many years in the future. The amateur radio service mirrors the public networks in the adoption of narrowband FM modulation for its baseline local and regional networks. The selectivity of equipment produced for the amateur service tends to be lower than radios produced for government emergency services, which limits the re-use of adjacent-channel frequencies.

Frequency allocation is handled quite differently in the civil and amateur radio services. City, county, and state agencies are permanently allocated frequency channels for their operations by the FCC. In the amateur radio service, frequency bands are identified, but no explicit channelization or allocations are imposed by the FCC. Because of the wide range of band allocations from shortwave through microwave, operating frequency choice in the amateur service is mainly a technology choice based upon propagation considerations, modulation formats, and equipment capabilities, rather than a regulatory choice. All frequency coordination of both automated retransmission (repeater) stations and point to point (simplex) stations is on a voluntary basis by

self-organized authorities. When disputes arise, the FCC usually defers to the decision of the local or national amateur frequency coordinating authority [3], [4]. The availability of inexpensive portable equipment and the good propagation characteristics over hilly terrain make the 147 MHz VHF amateur radio band quite popular as well as a site of frequent contention. The subject of the remainder of this paper is frequency coordination of a new VHF simplex channel for Stanford amateur radio emergency service use. The planning, modeling and testing which led to this new coordination took place in August of 2002, and the frequency has been used regularly for drill purposes since the fall of that year.

II. Radio Emergency Services at Stanford

The Stanford University campus is located at the north end of Santa Clara County directly on the border with San Mateo County. The populations of these two counties are 1.6 million and 700 thousand people, respectively. The core Stanford campus is 8 square kilometers in area and houses 14 thousand residents [5]. The daytime population swells to over 20 thousand due to student and staff commuters. The Stanford University Amateur Radio Emergency Service is included in the university emergency response plan. Specific facilities maintained for use by amateur service volunteers include a 450 MHz FM voice repeater at 30m elevation on campus to provide local coverage, and a 1300 MHz repeater at 850m elevation approximately 8 kilometers from campus to provide regional coverage. One 147 MHz simplex channel is identified for local tactical use.

A decade after the Oakland Hills fires, the attacks on New York and Washington, D.C. in September 2001 prompted an additional review of local Bay Area emergency planning. On the university campus, the Stanford Campus Residential Leaseholders [6], the homeowners association, undertook a major overhaul of their emergency plan in conjunction with the campus police and fire services. As part of the review, the need for a tactical radio communications facility was identified. Two candidate radio services were proposed to meet this need, the family radio service (FRS) and the amateur radio service. The family radio service was created in the mid 1990's by the FCC for recreational use and is part of the FCC's Citizens Band services. FRS uses 1/2 watt narrowband FM transmissions in the 460 MHz band. Ground testing with typical FRS hand-held units revealed that the terrain and vegetation of the campus (the Stanford sports team mascot is the Tree) prevented FRS radios from providing the needed coverage. Testing with 147 MHz amateur service hand-held radios with 5 watt capacity provided coverage that was acceptable. An additional advantage of the amateur service is the ability to use much higher transmit power levels for base station radios if necessary. Though amateur licensees may use up to 1.5 kilowatts, voice FM base stations in the 147 MHz amateur service typically use 50 to 100 watts. The disadvantage of using the amateur service for this system is that radio operators need to pass the FCC license examination, whereas FRS and citizens band systems may be operated without a license. In the end, the technical benefits of the amateur service outweighed the regulatory hurdles, and a core group of radio operators were trained and licensed.

The additional loading which the residential civil defense program would place on the existing Stanford amateur radio emergency service frequencies led to an overhaul of the existing frequency allocations. Up to that time, the 440 MHz on-campus repeater served as the prime communication channel for regular drills, with a fall back plan to operate on the repeater frequency in simplex mode if the repeater, which does have battery backup, went down for other reasons. The existing 147 MHz simplex channel had been underutilized for many years, and in the meantime, the national frequency coordination authority had reallocated the band segment containing the Stanford frequency for space satellite to Earth station links [3]. Periodic review of the voluntary amateur band plans results in reallocation of existing frequencies. Though in an emergency, amateurs may forgo any voluntary band plans and operate on any frequency as needed, the presence of weekly and quarterly drill traffic on a satellite band would not be proper. Another solution would have been to drill on a secondary channel, and switch to the prime channel in the satellite band for actual emergencies. There is a strong desire to drill on the frequencies which will be used in an actual emergency, so that operators will have radios pre-programmed and configured, and won't have to remember new procedures when under the stress of an actual deployment. It was decided that the Stanford University Amateur Radio Emergency Service should seek a new permanent 147 MHz simplex channel for regular drills and emergency use. Simplex frequencies in the amateur service are designated as first come first serve. In the amateur emergency service, channels are typically chosen by reviewing a chart of frequencies claimed for use by surrounding towns, and picking a channel in a somewhat ad hoc fashion. Use of the channel during regional drill exercises will then reveal conflicts with other users. It is up to the users of simplex channels to resolve co-channel and adjacent channel interference issues with other users. Wishing to minimize potential conflicts from the start, the Stanford amateur radio emergency service approached the university's student radio club and the electrical engineering department for technical advice on selection of a new VHF simplex channel.

III. Planning and Propagation Modeling

Taking into account the budget (nil) and time constraints (one month) for channel selection, a three phase plan for project completion was developed. Phase one involved using public domain software and databases to model the VHF propagation environment of the Bay Area and identify the best candidate frequencies for Stanford operations. Phase two involved passive monitoring of selected frequencies during co-channel user's drills to reject poor candidates. Phase three involved use of the best candidate frequency for regular weekly drills, with the goal of identifying real conflicts with co- and adjacent channel users. At the end of the trial period, the candidate frequency would be presented to the amateur radio emergency service coordinators for Santa Clara and San Mateo counties for official adoption.

From the outset of phase one, it was recognized that a simple free-space propagation model would not accurately represent the radio environment of the Bay Area, and that the effects of terrain should be included in the modeling and selection process. As mentioned in the introduction, the topography variation is quite significant, and shadowing effects can allow frequency re-use over shorter spatial scales than free-space considerations alone would indicate (Figure 2).

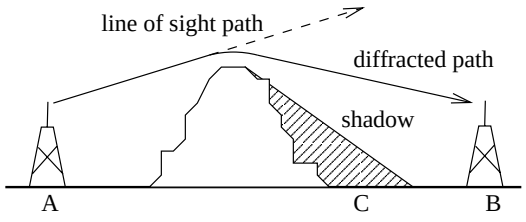


Fig. 2. Terrain effects on VHF radio propagation. The mountain prevents line of sight propagation from point A to point B. Knife-edge diffraction over the mountain allows station A to communicate with station B. Point C lies in the terrain shadow of station A. Different response teams located at points A and C may operate co-channel without experiencing interference.

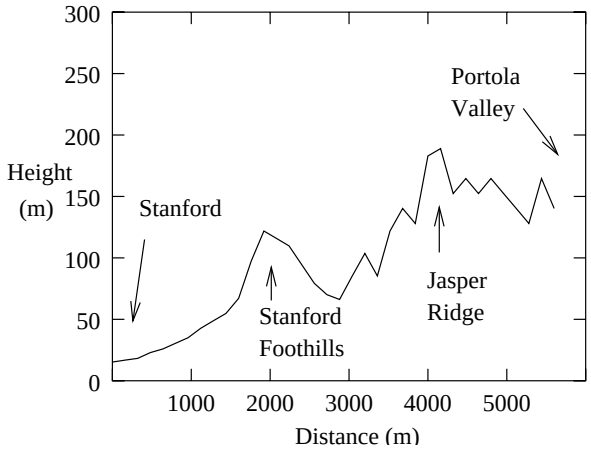


Fig. 3. Digital terrain profile. This data set, showing elevation above sea level in meters between the Stanford campus and the neighboring town of Portola Valley, demonstrates the challenging local propagation environment.

Propagation modeling that includes terrain effects is a FCC requirement for radio transmission facility licensing in the broadcast, business, and public safety radio bands. There are several commercially available software packages which help radio engineers site new towers for broadcast and other radio services, but the costs of these packages can be prohibitive for regular use in the amateur service, where funding sources are limited. Many regional amateur frequency coordinating authorities employ propagation models when reviewing applications for fixed repeater stations [4]. The most common model employed for terrain modeling in the land-mobile radio services is the irregular terrain model (ITM) developed in the late 1960's by the U.S. Department of Commerce [7]. The model combines both physical considerations such as knife edge diffraction and empirical data such as tropospheric propagation loss over typical path lengths. The computerized version of the model is also known as the Longley-Rice model after its primary developers at the National Bureau of Standards. Since the FCC requires the use of Longley-Rice for some radio service sitings, most commercial companies supplying propagation modeling software include ITM as a baseline mode, and develop a value-added product by incorporating graphical user interfaces and terrain database integration into their packages. The bare bones model is available in Fortran source code form from the Department of Commerce (C++ source is available as of November 2003) [8].

The irregular terrain model can be operated in two modes, area prediction mode and point to point mode. As the current study is concerned with co-channel and adjacent channel interference from other cities in the Bay Area, the point to point mode was used. Using ITM in point to point mode requires several inputs, including the frequency of operation, the distance between the radio terminals, the height of the transmitting antennas, ground conductivity, atmosphere refractivity, and a digital topography profile along the line between the terminals (Figure 3). Though no explicit channelization of the amateur service radio bands is required by the FCC, due to population density and repeater stations, the 147 MHz simplex band in the Bay Area is effectively divided into twenty six 15 kHz channels. To promote coordination, frequency usage by city and county are maintained in an on-line database by the amateur radio emergency service [9]. The physical location of each city or county radio facility was taken to be the latitude and longitude coordinates reported in the master U.S. Geological Survey catalog of place names [10]. An average antenna height of 10 meters above local terrain was assumed, based on typical amateur base station installations. Average to poor ground conductivity and sea-level atmosphere were assumed. The GLOBE digital topography database from the National Geophysical Data Center at 1km postings was used for terrain profiles [11]. The interface between ITM and GLOBE was provided by the Institute for Telecommunication Sciences of the U.S. Department of Commerce [8].

Programming for the model proceeded as follows. Since ITM was available only as Fortran source with a command line interface, and the input data existed in a number of disparate databases, a driver program was written in Perl to extract needed information from the databases, run the propagation model, and sort and format the results for printing. All databases were queried to produce the needed input data in flat, comma separated variable (CSV) files in standard ASCII encoding. Since the computational requirements for ITM are quite low by modern standards, a brute force search of the parameter space was employed. The driver program looped over all 26 possible frequencies for simplex operation, and computed and stored the path loss from each town using that frequency to Stanford. These results were then sorted in oder of predicted attenuation and written to disk in fixed column, tabular format. The programming effort resulted in 100 lines of Fortran and 150 lines of Perl. Program execution time was less than two seconds on a 450 MHz Pentium II processor running version 2.4 of the Linux operating system.

Several conclusions can be drawn from viewing the output of the program (Table I). First, the major source of path loss is the $1/d^2$ component of the standard Friis free-space transmission formula where d is the distance to the remote city. Second, the ranking does not follow the $1/d^2$ law exactly, since the terrain loss can be significant even over short distances. Compare the nearby city of La Honda, located 15km away in the Santa Cruz mountains, to San Rafael, located 70km away at the north end of the bay. The La Honda frequency would be a better choice than the San Rafael frequency for co-channel operation, since the total attenuation of a signal arriving from La Honda is greater than that of a signal arriving from San Rafael. The 500m elevation change of the intervening terrain between Stanford and La Honda causes

TABLE I
PREDICTED ATTENUATION FOR VHF BAND RADIO SIGNALS

Freq. (MHz)	Location	Dist. (km)	Space Loss (dB)	Ht. Chg. (m)	Terrn. Loss (dB)	Total Loss (dB)
146.580	Los Gatos	27.8	104.7	50	43.5	148.2
146.520	La Honda	15.1	99.4	507	48.3	147.7
146.490	Millbrae	27.5	104.6	68	40.2	144.8
147.420	San Rafael	69.0	112.6	65	31.7	144.3
146.565	Oakland	43.3	108.5	22	31.8	140.3
146.550	San Francisco	45.1	108.9	46	26.6	135.4
146.460	Cupertino	16.4	100.1	24	34.3	134.3
146.505	S. San Francisco	33.3	106.3	32	20.7	127.0
147.510	Santa Clara	20.3	101.9	34	24.4	126.3
147.570	Foster City	17.5	100.7	31	24.2	124.9

17 dB greater attenuation than the terrain between Stanford and San Rafael. The choice of La Honda over San Rafael may seem counterintuitive to those who only consider free-space loss, but it demonstrates the utility of including terrain modeling.

Some additional manual interpretation of the model output was necessary, as the program does not completely capture the dynamics of amateur radio service simplex operation in the Bay Area. The top program choice, 146.58 MHz, was already allocated to a digital packet radio service. The second program choice, 146.52 MHz, is designated by the national frequency coordination authority as a nation-wide hailing frequency [3]. The fourth program choice, 147.42 MHz, had the town of Los Altos Hills, only 3 kilometers from the campus, as an adjacent channel user, and was rejected on the basis of the frequency selectivity characteristics of radios manufactured for use in the amateur service. This left two choices in the top five, 146.49 and 146.565 MHz, which were identified as candidates for phase two passive monitoring.

IV. Theory into Practice

Phase two passive monitoring took place during the first two weeks of August 2002. The primary base station for Stanford amateur radio emergency service operations is located at 30m elevation on campus. As part of the emergency communications plan, this facility may call upon a high frequency (3 to 30 MHz) radio facility in the hills behind campus at 150 meters elevation for state-wide relaying of traffic. Testing was carried out at the secondary facility since the facilities must be able to communicate via the tactical VHF simplex frequency in the event of an emergency. Passive monitoring of the 146.565 Oakland amateur radio emergency service weekly drill traffic led to the conclusion that co-channel operation would result in significant interference, as Oakland is essentially a straight shot across the bay from Stanford. Monitoring of the 146.49 MHz Millbrae frequency revealed that co-channel operation would be possible without significant interference.

In late August and early September 2002, phase three of the plan was put into action, and regular weekly Stanford drills commenced on 146.49 MHz. This new VHF tactical frequency was also used during quarterly regional drills. No significant co-channel or adjacent channel interference was reported over the next few months of drill operations, and in January 2003 the frequency choice was presented to regional amateur radio emergency service officials for formal adoption.

Several weaknesses in the use of the irregular terrain propagation model were identified by the testing phase. First, since a large number of Bay Area cities are located on the edge of the bay, line-of-sight over water propagation should be accounted for in the model by changing the ground conductivity when the majority of the signal path is determined to be over water. It should be noted that the model prediction worked well for the actual frequency chosen at the end of the study, since the propagation path from Stanford to Millbrae is over land. Second, the Bay Area terrain is quite variable, which can require cities to use antenna supports which exceed the 10m assumption in the model in order to "see over" neighboring hills. The databases could be augmented by polling city radio officers to obtain actual antenna heights. Third, 1km postings are almost too coarse for this type of work. An additional enhancement to the model would be the use of finer resolution topographic data. Public domain digital elevation maps at 30m postings are available from the U.S. Geological Survey, and could be incorporated into a future upgrade to the software. Many commercial packages make use of these 30m data sets.

Propagation modeling is a great aid for planning purposes, but occasionally co- or adjacent interference occurs in the real world and must be dealt with. An effective technical solution if the interference is not too strong is the use of the continuous-tone coded squelch system (CTCSS). This system mixes a low frequency (60 to 250 Hz) tone into the audio of the transmitting station. The receiver squelch is set to open only when the proper tone is present on the demodulated audio. CTCSS encoders and decoders are standard issue on both family radio service and amateur radio service equipment. The system is effective in allowing moderate channel re-use while maintaining back compatibility with unequipped radios. The Stanford amateur radio emergency service routinely drills using this tone feature, so that operators are trained and ready if interference becomes an issue during an actual deployment.

V. Conclusions

In this paper, the role that the amateur radio service plays in augmenting civil authority communication networks in times of disaster has been highlighted. The San Francisco Bay Area has heightened sensitivity to the need for disaster communications due to its large population and experiences with the 1989 Loma Prieta earthquake and 1991 Oakland Hills fires. In the wake of the 2001 terror attacks on the east coast, the Stanford homeowners association reviewed its tactical communication requirements and requested the assistance of the student radio club and the electrical engineering department in meeting its communication needs. Preliminary ground tests in the 460 MHz UHF band determined the inability of the family radio service (FRS) to provide coverage of the campus area. Further testing found that equipment designed for the amateur radio service could provide the needed coverage due to higher power transmitters and use of VHF frequencies which have better propagation characteristics across the heavily vegetated campus. The required licensing and training of community members as amateur radio service operators was not seen as an impediment to the adoption of the amateur radio service for this need.

Frequency coordination in a voluntary radio service can be simplified by defining a three phase plan which makes use

of modeling, passive monitoring, and active testing to identify usable channels. Computer modeling of the radio propagation environment selects the best candidate frequencies and greatly reduces the number of channels which must be monitored during on-air testing. The availability of public domain terrain propagation models and geographic databases allows volunteers in the amateur service to use the same engineering tools as professionals to complete a frequency utilization study. Terrain shadowing effects permit co-channel operations that free-space propagation considerations alone would discount. Passive monitoring of candidate frequencies will eliminate those with significant co-channel interference. An active testing phase is necessary to identify conflicts with co- or adjacent channel users before the final adoption of the best candidate channel can take place.

Adoption of this plan for the voluntary coordination of a new VHF simplex frequency for Stanford University emergency communications needs has been a success. The plan was executed on time and within budget constraints. The frequency selected at the completion of the study has been in continual use since the fall of 2002 with no reports of co- or adjacent channel interference, demonstrating the effectiveness of the exercise. The methodologies used in this study can be readily applied by other groups in the amateur radio service, and the lessons learned can be extended to other radio services.

Acknowledgment

The authors thank the volunteer members of both the Stanford Amateur Radio Club [12] and the Amateur Radio Emergency Service [13] for support of the testing phase of this project. R.T. Tidd and L.W. Carr provided frequency coordination contact points for San Mateo and Santa Clara counties, respectively.

References

[1] "Amateur Radio Service," U.S. Code of Federal Regulations, Title 47, Part 97, Section 97.1. [Online]. Available: http://www.access.gpo.gov/nara/cfr/waisidx_03/47cfr97_03.html

[2] Project 25. Association of Public Safety Communications Officials International. [Online]. Available: http://www.apcointl.org/frequency/project25/index.html

[3] Band plans. American Radio Relay League, Inc. [Online]. Available: http://www.arrl.org/FandES/field/regulations/bandplan.html#2m

[4] Band plans. The Northern Amateur Relay Council of California, Inc. [Online]. Available: http://www.narcc.org/Rptr_Lists/Bandplan.html

[5] U.S. Census. [Online]. Available: http://www.census.gov/

[6] Stanford Campus Residential Leaseholders, Inc. [Online]. Available: http://www.scrl.org/

[7] A. G. Longley and P. L. Rice, "Prediction of tropospheric radio transmission over irregular terrain: a computer method," Environmental Science Services Administration, U.S. Department of Commerce, Boulder, Colorado, Tech. Rep. ERL 79-ITS 67, 1968.

[8] Irregular terrain model. Institute for Telecommunication Sciences, U.S. Department of Commerce. [Online]. Available: http://elbert.its.bldrdoc.gov/itm.html

[9] R. T. Tidd. Northern California ARES/RACES/ACS/SAR/VIP frequencies. [Online]. Available: https://www.quickbase.com/db/6sa4rmya

[10] Geographic names information system. U.S. Geological Survey. [Online]. Available: http://geonames.usgs.gov/

[11] The global land one-km base elevation project. National Geophysical Data Center, National Oceanic and Atmospheric Administration. [Online]. Available: http://www.ngdc.noaa.gov/seg/topo/globe.shtml

[12] Stanford Amateur Radio Club. [Online]. Available: http://www-w6yx.stanford.edu/w6yx/

[13] Stanford Amateur Radio Emergency Service. [Online]. Available: http://www-suares.stanford.edu/suares/

Capacity of Fading Broadcast Channels with Quality-of-Service Constraints

Chris T. K. Ng

Abstract. For wireless fading broadcast channels, different capacity configurations such as ergodic, outage, minimum rate, or limited-jitter can be represented by a triplet of quality-of-service parameters from each user: maximum rate, minimum rate, and shortage probability. For each fading state, the channel capacity is obtained from evaluating a finite set of possible extreme points. Optimal power allocation strategy across fading states is shown to be water-filling with rates determined by the effective noise of the corresponding fading states, then combined with constant rate power allocation. Shortage capacity can be similarly obtained by removing the minimum rate constraints when the system is in one of the shortage fading states.

Index Terms: Broadcast channels, capacity region, quality-of-service constraints, fading channels.

I. Introduction

SUPPORTING real-time, multiuser, and heterogeneous traffic such as combinations of multimedia, data, and control signals over wireless communication channels has been an active research topic. In multiuser wireless communications, a broadcast channel refers to a transmitter sending independent information to multiple users simultaneously, as in the downlink communication from a base station to a group of mobile terminal users. The received signal strength fluctuates randomly over time due to scattering, and the constructive and destructive interference of the transmitted signal. This effect of time-varying received signal strength is called fading. For instance, in a favorable fading state when the received signal strength is high, data can be sent at a high rate using little transmit power. On the contrary, in a fading state when signal strength is low, data transmission rate suffers, or it requires large transmit power to boost the transmission rate to an acceptable level. Different applications have different tolerance against such random channel fluctuations. Hence in addition to an average transmission rate, quality-of-service (QoS) parameters such as delay bound, minimum throughput, maximum jitters or outage percentage can be used to fully specify the communications requirements of an application. For example, file downloads might require high average transmission rates, but otherwise have no delay constraints. On the other hand, voice-streaming applications need only moderate average rates, yet impose stringent delay constraint requirements.

The capacity of a channel determines the maximum rate data can be sent to the receiver with arbitrarily small probability of error. Ergodic capacity is the long term average transmission rate over all fading states. Outage capacity is likewise defined, with the additional requirement that the transmitter has to maintain a constant rate, except for a given fraction of time when transmission is allowed to be suspended (typically during unfavorable fading states). As a special case, zero-outage capacity allows no transmission suspension, and thus requires a constant rate at all times. Finally, minimum rate capacity is the average rate achieved with the constraint that a given minimum rate for each user has to be maintained over all fading states.

The author is a PhD student in the Department of Electrical Engineering at Stanford. Email: Chris.Ng@stanford.edu

While capacity for fading broadcast channels with different constraints are known: ergodic capacity [1], [2], outage capacity [3], and minimum rate capacity [4], this analysis focuses on the scenario where each user is able to specify a set of parameters that represents the QoS requirements on the channel. The user specifies a maximum rate needed by the applications, a minimum rate required, and a shortage probability over which the minimum rate stipulation might be waived. Effectively, when in shortage, the system resorts to a best-service resource provisioning scheme, in which minimum rates are no longer guaranteed. Similar to [3], there is also a system-wide parameter to specify whether common shortage mode, where all users declare shortage simultaneously, or independent shortage mode, where users declare shortage separately, is used.

This model is motivated by the desire to support heterogeneous applications with different QoS requirements over a common wireless channel. For example, delay-insensitive applications such as batch file transfers seek to maximize ergodic capacity with little QoS constraints. On the other hand, an adaptive audio/video streaming application would operate with a minimum rate needed for legible communication up to a maximum rate in which the source content is encoded. Any rate above this maximum does not provide additional utility. Whereas for wireless control applications, a fixed rate (therefore fixed delay) is much more desirable than raw capacity, hence the user would wish to set the minimum rate the same as the maximum rate to eliminate jitters.

The QoS requirement parameters are explained in Section II. Next, Section III presents the discrete-time memoryless broadcast channel system model used in the analysis. Section IV considers the scenario where the broadcast channel is static (i.e., no fading), while Section V extends the analysis results for a fading channel. Effects of shortage on capacity and power allocation is investigated in Section VI. Numerical results are presented in Section VII, and conclusions follow in Section VIII.

II. Quality-of-Service Constraints

Each user will specify the QoS requirements in a triplet of maximum rate R^m, minimum rate R^n, and shortage probability q.

Let (R_j^m, R_j^n, q_j) be the QoS parameters given by user j, with $0 \leq R_j^n \leq R_j^m$ and $0 \leq q_j \leq 1$. Specifying a maximum rate means user j derives no additional utility beyond the rate R_j^m. The maximum rate may represent system bottlenecks (e.g., limited encoder/decoder processing rate, buffer access speed) such that even if additional power is available, the system still cannot transmit to the user at a higher rate.

On the other hand, setting a minimum rate with shortage probability means user j must receive rate equal to or above R_j^n with probability $1 - q_j$. With the remainder probability q_j, the system can declare shortage and the minimum rate is no longer guaranteed. Shortage capacity differs from outage capacity [3] in that when system is in shortage, while it is short of providing a minimum rate guarantee, transmission might still take place at a lower rate. Whereas when system is in outage, transmission is suspended completely. If transmission power is set to zero when system is in shortage, then shortage capacity reduces to outage capacity.

Besides the QoS requirements triplet (R_j^m, R_j^n, q_j) from each user, a system-wide parameter shortage mode is also to be specified. In common shortage mode, all users are required to declare shortage simultaneously, whereas independent shortage mode allows each user to have different shortage fading states. Naturally, if common shortage is specified, it is assumed that q_j's of all users are the same: $q_1 = \ldots = q_M = q$.

The QoS parameters can be set to represent a variety of channel capacity configurations. For example, having a minimum rate of zero and maximum rate of infinity leads to the ergodic capacity, while setting minimum rate the same as maximum rate represents zero-outage capacity. Different capacity configurations represented by the QoS parameters are shown in Table I.

Capacity Configuration	*QoS Parameters*
Ergodic [1], [2]	$(R^m, R^n, q) = (\infty, 0, 0)$
Zero-outage [3]	$R^m = R^n$, $q = 0$
Minimum rate [4]	$(R^m, R^n, q) = (\infty, r_{min}, 0)$
Outage [3]	$R^m = R^n$, $q = p_{out}$
Limited-jitter	$R^m - R^n = r_{jitter}$

TABLE I

CHANNEL CAPACITY CONFIGURATIONS REPRESENTED BY QOS PARAMETERS.

At first, it might not be intuitive how having a maximum rate parameter would represent a constraint to the system. Since a capacity region encompasses all achievable rates, it would appear that the maximum rate requirements would simply crop the portion of the capacity region that is beyond the respective maximum rates. However, since the maximum rate constraint limits the transmission rate in *every* fading state (instead of limiting the *average* transmission rate), it effectively restricts the transmitter from taking advantage of favorable fading states to transmit at high data rates when noise levels are low.

For example, Fig. 1 shows the ergodic, zero-outage, and minimum rate capacity regions of a two-user fading broadcast channel. The minimum rate requirements are R_1^n and R_2^n, for user 1 and user 2, respectively. As described in [4], the minimum rates have to lie within the zero-outage capacity region; otherwise, the requirements are not achievable. Compared to the ergodic capacity, imposing minimum rate constraints can only reduce the capacity region. In Fig. 2, it shows the QoS constraint capacity region for the same fading broadcast channel. The minimum rate constraints R_1^n and R_2^n are the same, but in this case there is also a maximum rate constraint R_1^m for user 1. The maximum rate constraint R_1^m lies completely outside the ergodic region. However, since capacity is the average rate over all fading states, without the constraint it is possible for the transmitter to send to user 1 at rate higher than R_1^m in some fading states. The maximum rate constraint imposes a limit on the transmission rate under those scenarios, and therefore has the effect of further reducing the capacity region.

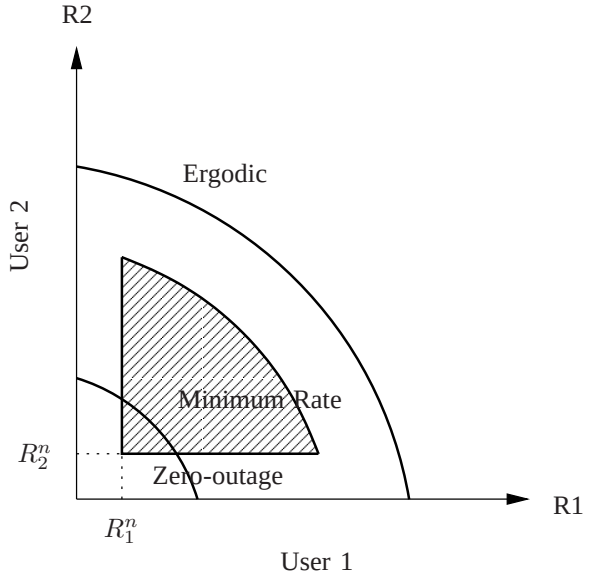


Fig. 1. Ergodic, zero-outage, and minimum rate capacity regions.

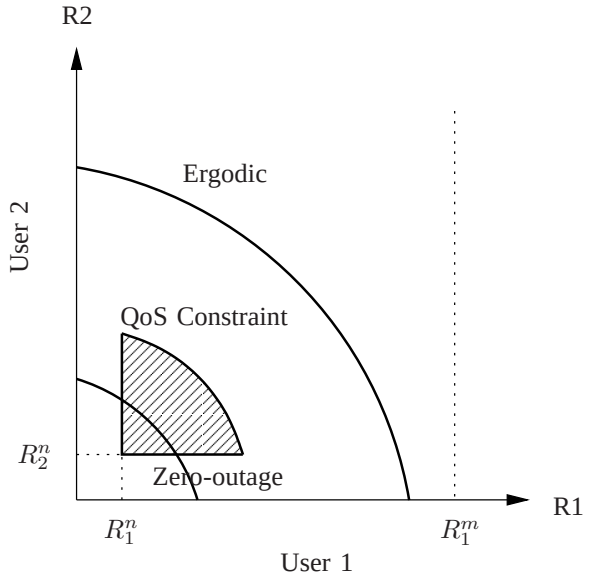


Fig. 2. Ergodic, zero-outage, and QoS constraint capacity regions.

III. System Model

The analysis adopts the same system model as defined in [2]–[4]. It assumes a discrete-time flat-fading Gaussian broadcast channel, with perfect channel state information at the transmitter and at all receivers. The transmitter varies its transmit power (thus controlling the transmission rate) under an average power constraint, and superposition coding with successive decoding [5] is used.

At each time instance i, the transmitter sends a signal $X_j(i)$ with power $p_j(i)$ to user j over bandwidth B. The average total transmission power over time cannot exceed $\overline{P}$. To characterize the channel, let v_j be the noise density of user j's channel, and $\sqrt{g_j(i)}$ its time-varying channel gain at time i. Then combine the channel gain and noise density to form the time-varying noise density $n_j(i) = v_j/g_j(i)$; since $n_j(i)$ incorporates the time-varying channel state, it will also be referred to as a fading state. Signals to all M users are sent simultaneously, hence the received signal for user j becomes

$$Y_j(i) = \sum_{k=1}^{M} X_k(i) + z_j(i) \tag{1}$$

where $z_j(i)$ is a Gaussian random variable with zero mean and variance $n_j(i)B$. It is assumed that the noise density vector $\mathbf{n}(i) = (n_1(i),\ldots, n_M(i))$ are known to the transmitter and all receivers at each time instance i.

Using superposition coding with successive decoding, the rate for user j at fading state $\mathbf{n}$ is

$$R_j(\mathbf{n}) = B\log\left(1 + \frac{p_j(\mathbf{n})}{n_j B + \sum_{i=1}^{M} p_i(\mathbf{n})\,\mathbf{1}[n_j > n_i]}\right) \tag{2}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Since B is simply a constant scaling factor, it can be assumed to be 1 without loss of generality. This analysis will focus on the case where the channel has two users ($M = 2$).

IV. Additive White Gaussian Noise Channel

First consider an additive white Gaussian noise (AWGN) channel where there is no fading. In this case, $n1$, $n2$ are constants with respect to time. Furthermore, assume $n_2 > n_1$ (user 2 has noisier channel); in the case of $n_1 > n_2$, the derivation remains valid with the subscripts 1, 2 reversed. Let user 1 and user 2 has QoS requirements $(R_1^m, R_1^n, 0)$ and $(R_2^m, R_2^n, 0)$, respectively (since channel has no fading, set shortage probability q_1 and q_2 to 0). To determine channel capacity, it is needed to maximize the achievable rates subject to the total power and QoS constraints:

$$\max_{p_1,p_2}\ \mu_1\log\left(1+\frac{p_1}{n_1}\right) + \mu_2\log\left(1+\frac{p_2}{p_1+n_2}\right) \tag{3}$$
$$\text{subject to: } p_1 + p_2 \le P,\ (R_1^m, R_1^n, 0),\ (R_2^m, R_2^n, 0)$$

where p_1 is the power allocated to user 1, and p_2 the power allocated to user 2. The proportion factors μ_1 and μ_2 allow achievable rates of user 1 and user 2 to be weighted differently; $0 < \mu_1 < 1$, and $\mu_2 = 1 - \mu_1$. For instance, $\mu_1 = \mu_2 = 1/2$ corresponds to the case when the metric of interest is the sum-rate $R_1 + R_2$, and both users' rates are weighted equally. Note that from the definition $\mu_1 \neq 0$ and $\mu_2 \neq 0$: This is to simplify the boundary conditions in the maximization; as each factor can be arbitrarily close to zero, no generality is lost.

The channel capacity is a function of (p_1, p_2), the noise density vector $\mathbf{n} = (n_1, n_2)$, and the total available power P. In an AWGN channel, $\mathbf{n}$ is known and remains constant. Suppose total power P is available for transmission, then the problem is to find out how to optimally divide up P between the two users; i.e., what is p_1^* and p_2^*? Let the function $F_A(P, \mathbf{n})$ denote this optimal power allocation between the two users. Given P and $\mathbf{n}$, $F_A(\cdot)$ returns the optimal p_1^* and p_2^*, which can then be substituted back in (3) to obtain the capacity achieved under optimal power allocation. Therefore, the optimal power allocation function is

$$(p_1^*, p_2^*) = F_A(P, \mathbf{n}), \tag{4}$$

and the respective achieved capacity is

$$R_C(P, \mathbf{n}) = \max_{p_1,p_2} R((p_1, p_2), P, \mathbf{n}) \tag{5}$$
$$= R(F_A(P, \mathbf{n}), P, \mathbf{n}). \tag{6}$$

Note that $R((p_1, p_2), P, \mathbf{n})$ represents an achievable rate with *some* power allocation p_1 and p_2. When the power allocation is optimal, capacity is therefore achieved and is denoted by $R_C(\cdot)$. Since power allocation is already determined, $R_C(\cdot)$ is only a function of P and $\mathbf{n}$.

Successive decoding allows user 1 to subtract out the signal power of user 2. Therefore, user 1 is not affected by user 2. To user 2, however, user 1's signal power p_1 is effectively noise. For user 2 to maintain a constant rate R_2^m, as derived in [4], its power p_2 needs to be accordingly increased whenever p_1 is increased:

$$p_2 = (e^{R_2^m} - 1)p_1 + n_2(e^{R_2^m} - 1). \tag{7}$$

Likewise, if a constant rate of R_2^n is to be maintained:

$$p_2 = (e^{R_2^n} - 1)p_1 + n_2(e^{R_2^n} - 1). \tag{8}$$

For convenience, define the following constants:

$$P_1^m \triangleq n_1(e^{R_1^m} - 1)$$
$$P_1^n \triangleq n_1(e^{R_1^n} - 1)$$
$$G_m \triangleq e^{R_2^m} - 1$$
$$G_n \triangleq e^{R_2^n} - 1$$
$$P_2^m \triangleq G_m(P_1^m + n_2)$$
$$P_2^n \triangleq G_n(P_1^n + n_2).$$

Then the constraints of (3) can be expressed in terms of the following five inequalities:

$$p_1 + p_2 \le P \tag{9}$$
$$p_1 \le P_1^m \tag{10}$$
$$p_1 \ge P_1^n \tag{11}$$
$$p_2 \le G_m(p_1 + n_2) \tag{12}$$
$$p_2 \ge G_n(p_1 + n_2) \tag{13}$$

Note that all of the constraints are linear in p_1, p_2. Let Ω denote the region over p_1, p_2 where all constraints are met. Since $P \ge 0$ and $0 \le R_j^n \le R_j^m$ for each user $j = 1, 2$, it implies $P_1^m \ge P_1^n$, $G_m \ge G_n$ and $P_2^m \ge P_2^n$. These conditions lead to a contiguous and closed region Ω on the p_1-p_2 plane, as shown in Fig. 3. Let

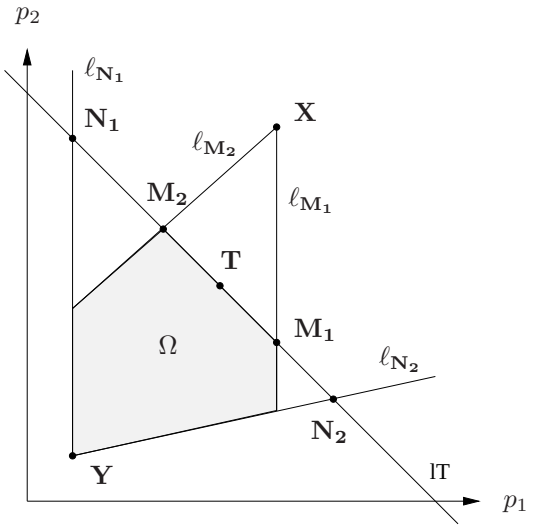


Fig. 3. QoS constraints on power allocation for user 1, 2.

the lines ℓ_T, ℓ_{M_1}, ℓ_{N_1}, ℓ_{M_2} and ℓ_{N_2} denote the boundaries of Ω, which correspond, respectively, to constraints (9)–(13) with equality being taken, as summarized in Table II below.

Line	*Constraint*	*Constraint Type*
ℓ_T	(9)	total average power
ℓ_{M_1}	(10)	maximum rate for user 1
ℓ_{N_1}	(11)	minimum rate for user 1
ℓ_{M_2}	(12)	maximum rate for user 2
ℓ_{N_2}	(13)	minimum rate for user 2

TABLE II
CONSTRAINT BOUNDARIES.

An extreme point is defined by the intersection of two boundaries. Note that a point (p_1, p_2) can be represented as a vector $\mathbf{p} = p_1\,\mathbf{i} + p_2\,\mathbf{j}$, where $\mathbf{i}$, $\mathbf{j}$ are unit vectors in the p_1, p_2 directions, respectively. Six extreme points $\mathbf{M_1}$, $\mathbf{N_1}$, $\mathbf{M_2}$, $\mathbf{N_2}$, $\mathbf{X}$ and $\mathbf{Y}$ are of particular interest; their definitions are tabulated in Table III.

Point	*Intersection*	*Constraint Type*
$\mathbf{M_1}$	ℓ_T, ℓ_{M_1}	maximum rate for user 1
$\mathbf{N_1}$	ℓ_T, ℓ_{N_1}	minimum rate for user 1
$\mathbf{M_2}$	ℓ_T, ℓ_{M_2}	maximum rate for user 2
$\mathbf{N_2}$	ℓ_T, ℓ_{N_2}	minimum rate for user 2
$\mathbf{X}$	ℓ_{M_1}, ℓ_{M_2}	minimum rate for both users
$\mathbf{Y}$	ℓ_{N_1}, ℓ_{N_2}	minimum rate for both users

TABLE III
CONSTRAINT BOUNDARY EXTREME POINTS.

The channel achievable rate function

$$R(\mathbf{p}) = \mu_1 \log\left(1 + \frac{p_1}{n_1}\right) + \mu_2 \log\left(1 + \frac{p_2}{p_1 + n_2}\right), \qquad (14)$$

being differentiable everywhere, is differentiable on the closed region Ω. Therefore, $R(\mathbf{p})$ must take on an absolute maximum on Ω, which can be found as follows:

1) First calculate the the gradient of $R(\mathbf{p})$:

$$\nabla R(\mathbf{p}) = \frac{\partial R}{\partial p_1}(\mathbf{p})\,\mathbf{i} + \frac{\partial R}{\partial p_2}(\mathbf{p})\,\mathbf{j} \qquad (15)$$

$$= \left(\frac{\mu_1}{p_1 + n_1} - \frac{\mu_2 p_2}{(p_1 + n_2)(p_1 + p_2 + n_2)}\right)\mathbf{i} + \frac{\mu_2}{p_1 + p_2 + n_2}\,\mathbf{j}. \qquad (16)$$

The gradient, defined everywhere, is never $\mathbf{0}$. In particular, since $\mu_2 > 0$, its $\mathbf{j}$-component will always be positive. Therefore, the maximum point of $R(\mathbf{p})$ is on one of the boundaries of Ω.

2) Consider the boundaries ℓ_{M_2} and ℓ_{N_2}. Define unit vectors $\mathbf{u_m}$, $\mathbf{u_n}$, respectively, to have the same directions as ℓ_{M_2}, ℓ_{N_2} (in the increasing p_1-p_2 polarity):

$$\mathbf{u_m} = \frac{\mathbf{i} + G_m\,\mathbf{j}}{\sqrt{(G_m)^2 + 1}}, \qquad (17)$$

$$\mathbf{u_n} = \frac{\mathbf{i} + G_n\,\mathbf{j}}{\sqrt{(G_n)^2 + 1}}. \qquad (18)$$

The directional derivatives of $R(\mathbf{p})$ with respective to $\mathbf{u_m}$ and $\mathbf{u_n}$, respectively, are given by the dot products:

$$R'_{\mathbf{u_m}}(\mathbf{p}) = \nabla R(\mathbf{p}) \cdot \mathbf{u_m} \qquad (19)$$

$$= \frac{1}{\sqrt{(G_m)^2 + 1}}\,\frac{\mu_1}{p_1 + n_2}, \qquad (20)$$

$$R'_{\mathbf{u_n}}(\mathbf{p}) = \nabla R(\mathbf{p}) \cdot \mathbf{u_n} \qquad (21)$$

$$= \frac{1}{\sqrt{(G_n)^2 + 1}}\,\frac{\mu_1}{p_1 + n_2}. \qquad (22)$$

Since both $R'_{\mathbf{u_m}}(\mathbf{p})$ and $R'_{\mathbf{u_n}}(\mathbf{p})$ are always positive, $R(\mathbf{p})$ is monotonically increasing in the directions of $\mathbf{u_m}$ and $\mathbf{u_n}$. Therefore, if the maximum point lies on ℓ_{M_2} or ℓ_{N_2}, it has to be one of the extreme points $\mathbf{M_2}$, $\mathbf{N_2}$, or $\mathbf{X}$.

3) Next consider the boundaries ℓ_{M_1} and ℓ_{N_1}. Both of them are parallel to the p_2 axis, hence the derivative of $R(\mathbf{p})$ in the direction of $\mathbf{j}$ is

$$R'_{\mathbf{j}}(\mathbf{p}) = \nabla R(\mathbf{p}) \cdot \mathbf{j} \qquad (23)$$

$$= \frac{\mu_2}{p_1 + p_2 + n_2}. \qquad (24)$$

Similarly, since the directional derivative $R'_{\mathbf{j}}(\mathbf{p})$ is always positive, $R(\mathbf{p})$ is monotonically increasing in the direction of $\mathbf{j}$. Therefore, if the maximum point lies on ℓ_{M_1} or ℓ_{N_1}, it has to be one of the extreme points $\mathbf{M_1}$ or $\mathbf{N_1}$.

4) If the maximum point does not lie on ℓ_{M_2}, ℓ_{N_2}, ℓ_{M_1}, or ℓ_{N_1}, then it has to be on the boundary ℓ_T within the open interval between points $\mathbf{M_1}$ and $\mathbf{M_2}$. Denote such point as $\mathbf{T}$. Then optimization (3) reduces to a standard one-dimensional maximization problem. The derivative of the channel capacity along ℓ_T with respect to p_1 is

$$\frac{dR_{\ell_T}}{dp_1}(p_1) = \frac{(\mu_1 - \mu_2)p_1 + \mu_1 n_2 - \mu_2 n_1}{(p_1 + n_1)(p_1 + n_2)}. \qquad (25)$$

Since $n_2 > n_1$, the derivative is always positive for $\mu_1 \geq \mu_2$. In such case $R_{\ell_T}(p_1)$ is monotonically increasing in the

direction of p_1, so $\mathbf{M_1}$ or $\mathbf{N_2}$ are the possible maximum points rather than $\mathbf{T}$. For $\mu_1 < \mu_2$, $\mathbf{T}$ is given by

$$\mathbf{T} = \left(\frac{\mu_1 n_2 - \mu_2 n_1}{\mu_2 - \mu_1}, P - \frac{\mu_1 n_2 - \mu_2 n_1}{\mu_2 - \mu_1} \right). \quad (26)$$

5) Finally, when available total power equals the minimum power requirement $P = P_1^n + P_2^n$. All of the above operating points close in to meet at the minimum rate extreme point $\mathbf{Y}$.

Therefore, there are only seven potential maximum point candidates: $\mathbf{M_1}$, $\mathbf{N_1}$, $\mathbf{M_2}$, $\mathbf{N_1}$, $\mathbf{T}$, $\mathbf{X}$ and $\mathbf{Y}$. Thus the maximum of $R(\mathbf{p})$ can be obtained by evaluating the candidate points that lie on the boundaries of Ω, and choose the one that yields the largest value, i.e.,

$$F_A(P, \mathbf{n}) = \mathbf{v} \text{ such that } R(\mathbf{v}) \geq R(\mathbf{v}') \ \forall \ \mathbf{v}' \neq \mathbf{v}, \quad \text{where } \mathbf{v}, \mathbf{v}' \in \{\mathbf{M_1}, \mathbf{N_1}, \mathbf{M_2}, \mathbf{N_2}, \mathbf{T}, \mathbf{X}, \mathbf{Y}\} \cap \Omega. \quad (27)$$

The coordinates of the possible maximum points are presented in Table IV. To evaluate the channel capacity at the possible maximum points, substitute the coordinates into (14) to obtain:

$$R(\mathbf{M_1}) = \mu_2 \log \left(1 + \frac{P - P_1^m}{P_1^m + n_2} \right) + \mu_1 R_1^m \quad (28)$$

$$R(\mathbf{N_1}) = \mu_2 \log \left(1 + \frac{P - P_1^n}{P_1^n + n_2} \right) + \mu_1 R_1^n \quad (29)$$

$$R(\mathbf{M_2}) = \mu_1 \log \left(1 + \frac{P - G_m n_2}{(G_m + 1) n_1} \right) + \mu_2 R_2^m \quad (30)$$

$$R(\mathbf{N_2}) = \mu_1 \log \left(1 + \frac{P - G_n n_2}{(G_n + 1) n_1} \right) + \mu_2 R_2^n \quad (31)$$

$$R(\mathbf{T}) = \mu_2 \log \left(1 + \frac{P - \frac{\mu_2 n_1 - \mu_1 n_2}{\mu_1 - \mu_2}}{\frac{\mu_2}{\mu_1 - \mu_2}(n_1 - n_2)} \right) + \mu_1 \log \left(1 + \frac{\mu_2 n_1 - \mu_1 n_2}{(\mu_1 - \mu_2) n_1} \right) \quad (32)$$

$$R(\mathbf{X}) = \mu_1 R_1^m + \mu_2 R_2^m \quad (33)$$

$$R(\mathbf{Y}) = \mu_1 R_1^n + \mu_2 R_2^n \quad (34)$$

Then the channel capacity associated with the optimal power allocation is

$$R_C(P, \mathbf{n}) = R(F_A(P, \mathbf{n}), P, \mathbf{n}), \quad (35)$$

which is given by one of the equations in (28)–(34).

V. Fading Channel

In this section, zero-shortage capacity of a fading channel will be studied. Effects of shortage are analyzed in Section VI. For a given total power $P(\mathbf{n})$ and noise density $\mathbf{n}$, optimally allocating the power between the users is solved in Section IV; the power allocation between user 1 and user 2 is given by (27) and the capacity achieved is the maximum of (28)–(34). Hence, the remaining step is to determine the optimal power allocation across fading states $\mathbf{n}$.

In a fading channel, n_1, n_2 are random variables with known joint probability distribution. Consider based on the noise density vector $\mathbf{n} = (n_1, n_2)$, the total power $P(\mathbf{n})$ can be varied. Partition the noise density vector into states $\{S_{\mathbf{M_1}}, S_{\mathbf{N_1}}, S_{\mathbf{M_2}}, S_{\mathbf{N_2}}, S_{\mathbf{T}}, S_{\mathbf{X}}, S_{\mathbf{Y}}\}$, where $S_{\mathbf{v}}$ corresponds to the values of $\mathbf{n}$ that results in the channel capacity taking on maximum at $\mathbf{v}$ for a given total power P, i.e.,

$$\mathbf{n} \in S_{\mathbf{v}} \text{ if } F_A(P, \mathbf{n}) = \mathbf{v}, \quad \text{where } \mathbf{v} \in \{\mathbf{M_1}, \mathbf{N_1}, \mathbf{M_2}, \mathbf{N_2}, \mathbf{T}, \mathbf{X}, \mathbf{Y}\}. \quad (36)$$

Therefore, the channel capacity for a given noise density state $S_{\mathbf{v}}$ and total power P is given by (28)–(34). For states $S_{\mathbf{X}}$ and $S_{\mathbf{Y}}$, the channel has constant rates. For the other states, the channel capacity has the following form:

$$R_C(P(\mathbf{n}), \mathbf{n}) = A_{\mathbf{v}} \log \left(1 + \frac{P(\mathbf{n}) - B_{\mathbf{v}}}{C_{\mathbf{v}}} \right) + D_{\mathbf{v}} \quad (37)$$

where $A_{\mathbf{v}}$, $B_{\mathbf{v}}$, $C_{\mathbf{v}}$ and $D_{\mathbf{v}}$ are parameters specific to the noise density state $S_{\mathbf{v}}$. The values of these parameters are tabulated in Table V.

To find the optimal power allocation strategy for each $\mathbf{n}$, i.e., to determine $P(\mathbf{n})$, it is needed to maximize the following:

$$\max_{P(\mathbf{n})} \ \mathrm{E}_{\mathbf{n}} \left[R((p_1(\mathbf{n}), p_2(\mathbf{n})), P(\mathbf{n}), \mathbf{n}) \right] \quad (38)$$

$$= \max_{P(\mathbf{n})} \mathrm{E}_{\mathbf{n}} \left[\mu_1 \log \left(1 + \frac{p_1(\mathbf{n})}{n_1} \right) + \mu_2 \log \left(1 + \frac{p_2(\mathbf{n})}{p_1(\mathbf{n}) + n_2} \right) \right]$$

$$\text{subject to: } \mathrm{E}_{\mathbf{n}}[P(\mathbf{n})] \leq \overline{P}, \quad (39)$$

where $p_1(\mathbf{n}) + p_2(\mathbf{n}) \leq P(\mathbf{n})$.

For $S_{\mathbf{X}}$ and $S_{\mathbf{Y}}$, power allocation is dictated by the maximum and minimum rates:

$$P_{\mathbf{X}}(\mathbf{n}) = P_1^m + P_2^m \quad (40)$$

$$P_{\mathbf{Y}}(\mathbf{n}) = P_1^n + P_2^n \quad (41)$$

where $P_{\mathbf{X}}(\mathbf{n})$, $P_{\mathbf{Y}}(\mathbf{n})$ are the power allocation strategy for states $S_{\mathbf{X}}$, $S_{\mathbf{Y}}$, respectively. For the other states, substitute (37) in (38) to obtain:

$$\max_{P(\mathbf{n})} \mathrm{E}_{\mathbf{n}} \left[A_{\mathbf{v}} \log \left(1 + \frac{P(\mathbf{n}) - B_{\mathbf{v}}}{C_{\mathbf{v}}} \right) + D_{\mathbf{v}} \right] \quad \text{subject to: } \mathrm{E}_{\mathbf{n}}[P(\mathbf{n})] \leq \overline{P}. \quad (42)$$

Using Lagrangian method, form

$$J(P(\mathbf{n})) = A_{\mathbf{v}} \log \left(1 + \frac{P(\mathbf{n}) - B_{\mathbf{v}}}{C_{\mathbf{v}}} + D_{\mathbf{v}} \right) - \lambda(\mathrm{E}_{\mathbf{n}}[P(\mathbf{n})] - \overline{P}), \quad (43)$$

then set $\frac{\partial J}{\partial P(\mathbf{n})}$ to zero to obtain:

$$P(\mathbf{n}) = \begin{cases} A_{\mathbf{v}} \frac{1}{\lambda} - (C_{\mathbf{v}} - B_{\mathbf{v}}) & A_{\mathbf{v}} \frac{1}{\lambda} \geq C_{\mathbf{v}} - B_{\mathbf{v}} \\ 0 & \text{else} \end{cases} \quad (44)$$

where $\frac{1}{\lambda}$ is the water-filling level. Define effective noise $n_{\mathbf{v}}^*(n_1, n_2) = C_{\mathbf{v}} - B_{\mathbf{v}}$. The values of $A_{\mathbf{v}}$ and $n_{\mathbf{v}}^*$ for each state are summarized in Table VI. Notice that states S_{M_1}, S_{N_2} and S_T all have $A_{\mathbf{v}} = \mu_2$ and the same effective noise $n_{\mathbf{v}}^* = n_2$. Therefore, there are only three water-filling equations:

$$P_{\mathbf{M_1}, \mathbf{N_1}, \mathbf{T}}(\mathbf{n}) = \lfloor \mu_2 \frac{1}{\lambda} - n_2 \rfloor_+ \quad (45)$$

$$P_{\mathbf{M_2}}(\mathbf{n}) = \lfloor \mu_1 \frac{1}{\lambda} - (G_m + 1) n_1 + G_m n_2 \rfloor_+ \quad (46)$$

$$P_{N_2}(\mathbf{n}) = \lfloor \mu_1 \frac{1}{\lambda} - (G_n + 1) n_1 + G_n n_2 \rfloor_+ \quad (47)$$

Point	p_1	p_2	*Constraint Type*
$\mathbf{M_1}$	P_1^m	$P - P_1^m$	maximum rate for user 1
$\mathbf{N_1}$	P_1^n	$P - P_1^n$	minimum. rate for user 1
$\mathbf{M_2}$	$\frac{1}{G_m+1}(P - G_m n_2)$	$\frac{G_m}{G_m+1}(P + n_2)$	maximum rate for user 2
$\mathbf{N_2}$	$\frac{1}{G_n+1}(P - G_n n_2)$	$\frac{G_n}{G_n+1}(P + n_2)$	minimum rate for user 2
$\mathbf{T}$	$\frac{\mu_1 n_2 - \mu_2 n_1}{\mu_2 - \mu_1}$	$P - \frac{\mu_2 n_1 - \mu_1 n_2}{\mu_1 - \mu_2}$	total average power
$\mathbf{X}$	P_1^m	P_2^m	maximum rate for both users
$\mathbf{Y}$	P_1^n	P_2^n	minimum rate for both users

TABLE IV

POSSIBLE MAXIMUM POINTS.

State	$A_\mathbf{v}$	$B_\mathbf{v}$	$C_\mathbf{v}$	$D_\mathbf{v}$
$S_{\mathbf{M_1}}$	μ_2	P_1^m	$P_1^m + n_2$	$\mu_1 R_1^m$
$S_{\mathbf{N_1}}$	μ_2	P_1^n	$P_1^n + n_2$	$\mu_1 R_1^n$
$S_{\mathbf{M_2}}$	μ_1	$G_m n_2$	$(G_m+1)n_1$	$\mu_2 R_2^m$
$S_{\mathbf{N_2}}$	μ_1	$G_n n_2$	$(G_n+1)n_1$	$\mu_2 R_2^n$
$S_\mathbf{T}$	μ_2	$\frac{\mu_2 n_1 - \mu_1 n_2}{\mu_1 - \mu_2}$	$\frac{\mu_2}{\mu_1 - \mu_2}(n_1 - n_2)$	$\mu_1 \log\left(1 + \frac{\mu_2 n_1 - \mu_1 n_2}{(\mu_1 - \mu_2) n_1}\right)$

TABLE V

RATE FUNCTION PARAMETERS FOR EACH NOISE DENSITY STATE.

State	$A_\mathbf{v}$	$n_\mathbf{v}^*(n_1, n_2)$
$S_{\mathbf{M_1}}$	μ_2	n_2
$S_{\mathbf{N_1}}$	μ_2	n_2
$S_{\mathbf{M_2}}$	μ_1	$(G_m+1)n_1 - G_m n_2$
$S_{\mathbf{N_2}}$	μ_1	$(G_n+1)n_1 - G_n n_2$
$S_\mathbf{T}$	μ_2	n_2

TABLE VI

WATER-FILLING PARAMETERS FOR EACH NOISE DENSITY STATE.

Next, it is needed to determine which power allocation equation to use for a given fading state $\mathbf{n}$. Consider the power allocation $P_\mathbf{v}(\mathbf{n})$ in the direction n_1 for each state. Effective noise of states $S_{\mathbf{M_1}}$, $S_{\mathbf{N_1}}$, $S_\mathbf{T}$ does not depend on n_1, so allocated power is simply the constant $\mu_2 \frac{1}{\lambda} - n_2$, whereas states $S_{\mathbf{M_2}}$ and $S_{\mathbf{N_2}}$ have linear power allocation with slopes, respectively, $-(G_m+1)$ and $-(G_n+1)$, as illustrated in Fig. 4.

Observe that as given in Table IV, as n_1 increases, T moves from $\mathbf{N_2}$ towards $\mathbf{M_2}$; therefore, in the direction of increasing n_1 the states can only appear in this order: $\{S_{\mathbf{N_2}}, S_\mathbf{T}, S_{\mathbf{M_2}}\}$. Coupled with the fact that slope of $P_{\mathbf{M_1},\mathbf{N_1},\mathbf{T}}(n_1) \geq$ slope of $P_{\mathbf{N_2}}(n_1) \geq$ slope of $P_{\mathbf{M_2}}(n_1)$ with respect to n_1, the water-filling power allocation can be represented compactly by

$$P_W(\mathbf{n}) = \min\{\max\{P_{\mathbf{N_2}}(\mathbf{n}), P_{\mathbf{M_1},\mathbf{N_1},\mathbf{T}}(\mathbf{n})\}, P_{\mathbf{M_2}}(\mathbf{n})\}. \quad (48)$$

Alternatively, in the direction of n_2, increasing n_2 while keeping n_1 fixed implies the state occurrence order $\{S_{\mathbf{M_2}}, S_{\mathbf{N_1}}\}$ and $\{S_{\mathbf{M_1}}, S_{\mathbf{N_2}}\}$. With slope of $P_{\mathbf{N_1}}(n_2) >$ slope of $P_{\mathbf{M_2}}(n_2)$, and slope of $P_{\mathbf{M_1}}(n_2) >$ slope of $P_{\mathbf{N_2}}(n_2)$, it leads to the same conclusion. Finally, combine with the maximum and minimum rate constraints, the optimal power allocation strategy is:

$$P^*(\mathbf{n}) = \min\{\max\{P_W(\mathbf{n}), P_\mathbf{Y}(\mathbf{n})\}, P_\mathbf{X}(\mathbf{n})\}. \quad (49)$$

The water-filling level $\frac{1}{\lambda}$ can be determined by choosing the maximum water-filling level allowed by the average total power

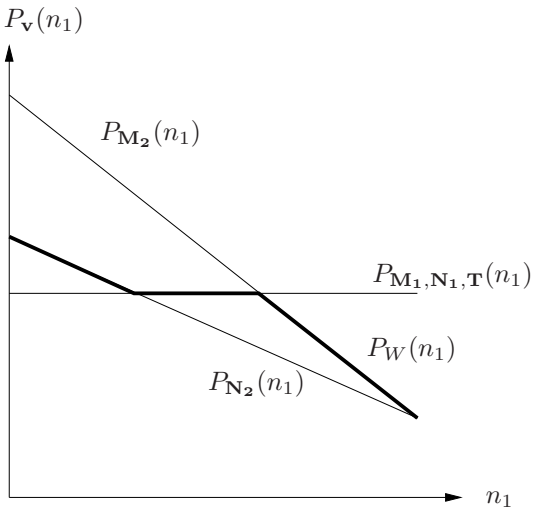


Fig. 4. Power allocation for each noise density state $S_\mathbf{v}$ in the direction of n_1, with the combined water-filling power allocation $P_W(n_1)$ shown in bold.

constraint:

$$\lambda = \lambda^* \text{ such that } \max_{\lambda^*} \mathrm{E}_\mathbf{n}[P(\mathbf{n}, \lambda^*)] \leq \overline{P}, \quad (50)$$

and the channel capacity achieved is simply the weighted average of the capacity over the fading states:

$$R^* = \mathrm{E}_\mathbf{n}[R(P^*(\mathbf{n}))]. \quad (51)$$

This is illustrated in Fig. 5, where the dashed line denotes a lower water-filling level than the one in bold.

VI. Shortage Probability

A. Common Shortage

In the common shortage mode, when the system declares shortage for a certain fading state $\mathbf{n}$, the minimum rate constraints

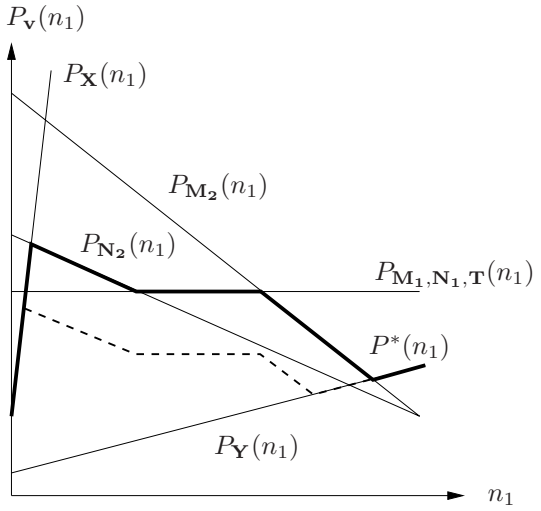


Fig. 5. Water-filling combined with constant rate power allocation. The dashed line denotes a lower water-filling level than the one in bold.

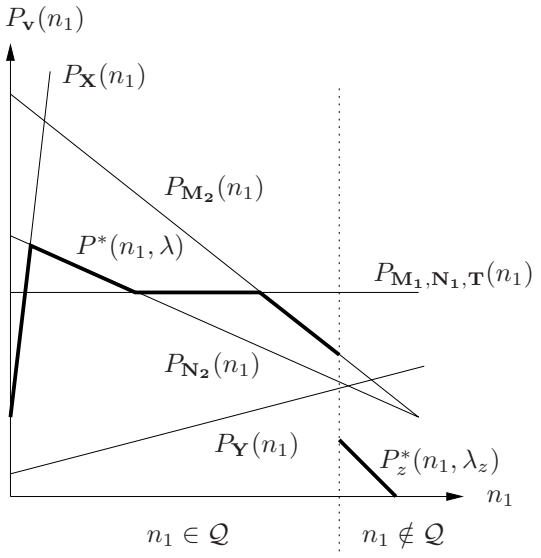


Fig. 6. Power allocation with shortage probability.

for all M users are removed. Naturally, all users have the same shortage probability q. Effectively, this is same optimization problem as before with minimum rates $R_j^n = 0\ \forall j = 1, \ldots, M$. Therefore, similar procedures from Section IV can be used to deduce the optimal power allocation.

1) First calculate the zero-shortage power allocation function as previously derived. Write $P^*(\mathbf{n}, \lambda)$ to denote power allocation in terms of its water-filling level λ, and $P^*(\mathbf{n})$ when λ is determined by the average total power constraint (50).
2) Repeat the same calculations with the minimum rate constraints removed, i.e., set $R_j^n = 0\ \forall j$; denote such power allocation function obtained as $P_z^*(\mathbf{n}, \lambda_z)$, and when the water-filling level λ_z is determined, as $P_z^*(\mathbf{n})$.
3) Select the set of shortage fading states $\mathcal{Q}$ to contain the fading states $\mathbf{n}$ that results in maximum power savings while not exceeding the specified shortage probability q:

$$\max_{\mathcal{Q}} \sum_{\mathbf{n}\in\mathcal{Q}} P^*(\mathbf{n}) - P_z^*(\mathbf{n}) \text{ such that } \sum_{\mathbf{n}\in\mathcal{Q}} \Pr(\mathbf{n}) \le q \quad (52)$$

4) As shown in Fig. 6, the final shortage capacity power allocation strategy $P_{\mathcal{Q}}(\mathbf{n})$ is obtained from combining $P^*(\mathbf{n}, \lambda)$ and $P_z^*(\mathbf{n}, \lambda_z)$ over their respective associated domain of $\mathbf{n}$.

$$P_{\mathcal{Q}}^*(\mathbf{n}, \lambda, \lambda_z) = \begin{cases} P^*(\mathbf{n}, \lambda) & \mathbf{n} \notin \mathcal{Q}, \\ P_z^*(\mathbf{n}, \lambda_z) & \mathbf{n} \in \mathcal{Q}. \end{cases} \quad (53)$$

The water-filling levels λ, λ_z need to be recalculated to satisfy the average total power constraint (50). However, since the removal of minimum rate constraints represents a discontinuous change of channel capacity with respective to $\mathbf{n}$, the water-filling levels λ and λ_z are independent. After constraint (50) is applied, there still is an extra degree of freedom over division of power between λ and λ_z. Unfortunately, as λ and λ_z have no closed form analytic expressions, standard optimization techniques cannot be applied. Notwithstanding, in practice when the shortage probability q specified is small, λ_z is often zero. This is because small q limits $\mathcal{Q}$ to only contain fading states with large noise density $\mathbf{n}$, over which the optimal power allocation strategy simply is to suspend transmission. In this case shortage capacity reduces to outage capacity.

B. Independent Shortage

In independent shortage mode, each user specifies a shortage probability $q_j,\ j = 1, \ldots, M$, over which the minimum rate constraint for that user is not guaranteed, i.e., $R_j^n = 0$. Calculating the optimal power allocation for independent shortage capacity is similar to that of common shortage. However, it is needed to compare the zero-shortage power allocation function to that of each of the $2^M - 1$ permutations of different users in shortage. Moreover, it is needed to divide the power among the $2^M - 1$ water-filling levels. As described in Section VI-A, as there is no closed form expression for the channel capacity, standard optimization techniques cannot be readily applied. Further research work is needed to devise the optimal power allocation strategy for this scenario.

VII. Numerical Simulations

In this section simulation results are presented. In all plots it is assumed that the average total transmission power $\overline{P}$ is 10mW, and the channel has a bandwidth B of 100kHz. In Fig. 7, the fading state distribution is: (n_1B, n_2B) = (1×10^{-4}mW, 1mW) with probability $1/2$, and (n_1B, n_2B) = (1mW, 1×10^{-4}mW) with probability $1/2$. The channel has a large 40dB SNR fluctuation between the fading states for each user. As expected, imposing a minimum rate constraint of 300kbps for each of the users reduces the capacity region. If each user also specifies a maximum rate constraint of 1200kbps, capacity region is further reduced. Note that the 1200kbps maximum rate actually lies outside of the ergodic capacity region. However, it still represents a limitation on the system since the transmitter can no longer send at arbitrarily high rates (within transmit power constraint) in a favorable fading state.

In Fig. 8, the channel has the fading distribution: (n_1B, n_2B) = (0.001mW, 0.1mW) with probability $1/2$, and (n_1B, n_2B)

= (0.1mW, 0.001mW) with probability $1/2$. In this case, the channel has a relatively smaller 20dB fluctuation in SNR between the fading states for each user. Therefore, imposing a minimum rate of 300kbps does not reduce the capacity region drastically. In fact, unlike the previous scenario, specifying a maximum rate constraint of 1200kbps for each user does not reduce the minimum rate capacity. It implies that even without the maximum rate constraint, in every fading state the transmission rate for each user does not exceed 1200kbps, thus having the maximum rate constraint does not represent any additional limitation to the transmitter. When a more restrictive maximum rate of 900kbps is imposed, it can be observed that only then the capacity region is reduced moderately.

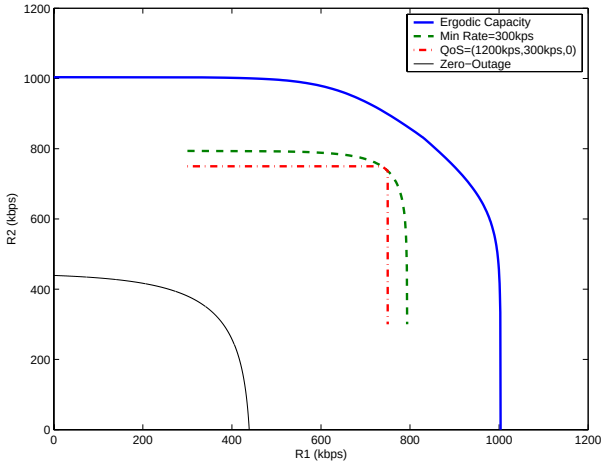


Fig. 7. Capacity regions of a symmetric two-user fading broadcast channel, with 40dB SNR difference between fading states.

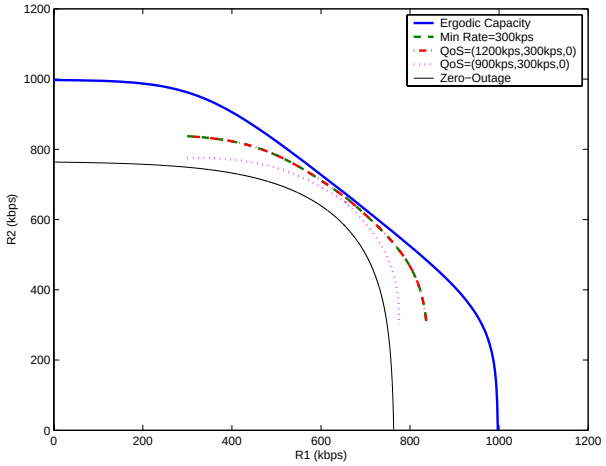


Fig. 8. Capacity regions of a symmetric two-user fading broadcast channel, with 20dB SNR difference between fading states.

VIII. Conclusion

In this analysis it has been shown that a variety of channel capacity configurations, e.g., ergodic, outage, minimum rate, limited-jitter, or a heterogeneous combination of them, can be represented by a triplet of QoS parameters from each user. Optimal power allocation between the users can be readily determined by evaluating a finite set of possible extreme points. The resulting channel capacity with respect to total power is either constant, or has a common form that is logarithmically proportional to the total power relative to an effective noise. This common capacity expression allows power allocation across fading states to be optimized, which is shown to be water-filling with rates determined by the effective noise of the corresponding fading states, then combined with constant rate power allocation. Finally, it is shown that shortage capacity can be similarly obtained by removing the minimum rate constraints when the system is in one of the shortage fading states. Therefore, the QoS parameter model provides a unifying framework in which channel capacity configurations with heterogeneous requirements from different users can be analyzed to obtain the respective channel capacity and optimal power allocation strategy.

Acknowledgment

The author would like to thank Prof. A. Goldsmith and Nihar Jindal for helpful discussions on the topic.

References

[1] D. Hughes-Hartog, "The capacity of the degraded spectral gaussian broadcast channel," Ph.D. dissertation, Stanford University, 1975.

[2] L. Li and A. J. Goldsmith, "Capacity and optimal resource allocation for fading broadcast channels—Part I: Ergodic capacity," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1083–1102, March 2001.

[3] ——, "Capacity and optimal resource allocation for fading broadcast channels—Part II: Outage capacity," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1103–1127, March 2001.

[4] N. Jindal and A. Goldsmith, "Capacity and optimal power allocation for fading broadcast channels with minimum rates," in *Proceedings of IEEE Globecom*, November 2001, pp. 1292–1296.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

ieee.stanford.edu/ecj