Exploiting Meshed Routing For MD-Coded Image/Video Content

Uma Parthavi Moravapalle
2007EE50410
IIT Delhi

parthavi.moravapalli@gmail.com

September 13, 2012

Supervisor:
- Dr. Swades De (EE - IITD)
Outline

1 Introduction
2 Theoretical model
3 Route construction
4 Genetic Algorithms
5 Implementation in NS2
6 Conclusion
7 Publications
8 Bibliography
Introduction

Multiple Description Coding

Motivation

- Wireless channel is inherently error prone and unreliable \(\Rightarrow\) difficulty in transmission of content rich multimedia
- Hence the existence of techniques like ARQ, Layered Coding, Multiple Description Coding

How does MDC work?

- Source stream is broken into sub-streams of equal importance; sent over different channels
- Any subset of the original set of descriptions can be used for decoding; Lost descriptions are estimated from those present
- Error resilience achieved due to independent nature of channels
- Popular techniques: Subsampling, Transform coding, Scalar Quantization
Multiple Description Coding

How does MDC work?

- Source stream is broken into sub-streams of equal importance; sent over different channels
- Any subset of the original set of descriptions can be used for decoding; Lost descriptions are estimated from those present
- Error resilience achieved due to independent nature of channels
- Popular techniques: Subsampling, Transform coding, Scalar Quantization
Multiple Description Coding with intermediate recovery

What could happen?

- Possibility of recovering from errors at several stages before the destination
- Quality expected to improve!
A probabilistic point of view

Figure: Equivalent traditional MDC-MPT and MDC intermediate recovery systems

- Probability of finding no descriptions at the end in Figure 1(a) is: $4p^2$
- Probability of finding no descriptions at the end in Figure 1(b) is: $2p^2$
Outline

1. Introduction
2. Theoretical model
3. Route construction
4. Genetic Algorithms
5. Implementation in NS2
6. Conclusion
7. Publications
8. Bibliography
Theoretical Model

- Distortion Measure: MSE/PSNR
- MDC type: MDTC; Ref: [3, 4]
- Number of descriptions: 2; Ref: [8]
- Network: Stochastic directed graph; delay cost: $c_{ij}$, $k$th segment in the $l$th path denoted by $H^l_k$; Segment defined as the set of links in a path between two recovery stages
- Channel: Rayleigh fading $\Rightarrow$ can be explained using two state Markovian dynamics; Ref: [10]

Link Model

![Gilbert Channel](image)

**Figure**: Gilbert Channel
Aggregate Wireless model

Why aggregation?
- If one Gilbert model per link, then $2^{|H_k|}$ possible states!
- Error free transmission $\iff$ all the channels are in Good state
- All channels in a segment are treated as a single Gilbert Channel

Theorem 1
The transition probabilities of the aggregate model are obtained as follows:

$$a_k^l = 1 - \prod_{(i,j) \in H_k^l} (1 - a_{ij}), \quad (1a)$$
$$\phi_k^l = 1 - \prod_{(i,j) \in H_k^l} (1 - \phi_{ij}), \quad (1b)$$
$$b_k^l = \frac{(1 - \phi_k^l)a_k^l}{\phi_k^l} \quad (1c)$$

Uma Parthavi (IIT Delhi)
The big picture: Probabilistic representation of Intermediate recovery system

**Figure:** Intermediate recovery system state transition diagram
States explained...

- S1: Both descriptions are uncorrupted/internally recoverable
- S2: Description 1 is uncorrupted/recoverable and description 2 is corrupted/unrecoverable
- S3: Description 2 is uncorrupted/recoverable and description 1 is corrupted/unrecoverable
- S4: Both the descriptions are corrupted/unrecoverable

If the probability of loss of a description in a segment $H_k$ is $P_k^l$, for $k = 1$ to $N + 1$, $l = 1, 2$, the transition probability matrix $U_k$ is given by:

$$U_k = \begin{pmatrix}
(1 - P_k^1)(1 - P_k^2) & P_k^2(1 - P_k^1) & P_k^1(1 - P_k^2) & P_k^1 P_k^2 \\
0 & 1 - P_k^1 P_k^2 & 0 & P_k^1 P_k^2 \\
0 & 0 & 1 - P_k^1 P_k^2 & P_k^1 P_k^2 \\
0 & 0 & 0 & 1
\end{pmatrix}$$

The transition probability matrix $U$ for the overall path is given by

$$U = \prod_{k=1}^{N+1} U_k$$
End-to-End Distortion

**Figure:** Multiple description coding with two channels and three receivers

Then, the probabilistic distortion $\mathcal{D}$ of the video can be expressed as:

$$\mathcal{D} = P_{00} d_0 + P_{01} d_1 + P_{10} d_2 + P_{11} d_3$$

$P_{ij}$ are given by the elements of first row of $U$, taken in column order.

**Corruption Probability of a Description**

$$Pr(errors \geq \tau) = \sum_{n_f=\tau}^{n_t} P(n_f, n_t)$$
**Theoretical model**

### Theorems

#### Theorem 2

End-to-end distortion increases with the number of hops \( r \) in a path.

![Variation of distortion with the number of hops](image)

**Figure**: Variation of distortion with the number of hops. \( F = 25 \text{ dB}, f_d T_s = 0.0004128, L = 1000 \text{ bits}, \tau = 25 \text{ bits}. \)
Theoretical model

Theorems

**Theorem 3**

In the case of a single intermediate recovery stage, for minimal distortion, the intermediate recovery node should be placed equidistant from the source and destination.

![Graph showing distortion (MSE in dB) vs Position of intermediate node]
Theoretical model

Distortion vs Number of recovery nodes

![Graph showing distortion vs number of recovery nodes]

**Advantage:**

9% decrease!

Scheme 1: data rate = 13Kbps, carrier frequency = 1850 MHz  
Scheme 2: data rate = 54 Mbps, carrier frequency = 2.4 GHz
Comparison: Error Threshold

Observations

MDC-IR has greater slope $\Rightarrow$ better error improvement for the same increase in threshold
Comparison: Mobile velocity

Theoretical model
Outline

1 Introduction
2 Theoretical model
3 Route construction
4 Genetic Algorithms
5 Implementation in NS2
6 Conclusion
7 Publications
8 Bibliography
Route construction

Requirements

Route requirements

- **Low delay**: To deliver the video to the end user on time
- **Low Hop count**: To ensure high quality
- **Disjointedness**: To achieve error resilience
- **Meshed**: To facilitate intermediate recovery
- **Low computation**

Define:

Two flow variables $x_{ij}^l$, $l = 1, 2$ for every $(i, j) \in E$, $S = \left\{ i : \sum_{l,j: (i,j) \in E} x_{ij}^l = 2 \right\}$

Delay

$$\text{Delay} = \sum_{k=1}^{N+1} \max \left\{ \sum_{(i,j) \in H_k^1} c_{ij}, \sum_{(i,j) \in H_k^2} c_{ij} \right\}$$
Route Optimization Problem

**OPT-DIST**

Minimize: \( D = DP' \)

s.t. \( \sum_{j:(i,j) \in E} x^l_{ij} - \sum_{j:(i,j) \in E} x^l_{ji} = \begin{cases} 1, & i = s, i \in V, l = 1, 2 \\ -1, & i = t, i \in V, l = 1, 2 \\ 0, & \text{otherwise} \end{cases} \)

\( \sum_{j:(i,j) \in E} x^l_{ij} = \begin{cases} \leq 1, & \text{if } i \neq t, i \in V, l = 1, 2 \\ 0, & \text{if } i = t, i \in V, l = 1, 2 \end{cases} \)

\( x^1_{ij} x^2_{ij} = 0, \forall (i,j) \in E \)

\( |S| = N + 1 \)

\( x^l_{ij} \in \{0, 1\}, \forall (i,j) \in E, l = 1, 2 \)

Delay \( < T \)
Heuristics

**MIN-DELAY**

It is the array containing the shortest pair of disjoint paths from every node to every other node in $G$.

**procedure FINDPATH**($G$, $s$, $t$, $N$)

1. $minDelay \leftarrow -1$
2. $path \leftarrow NULL$
3. **for all** $(v_1, \ldots, v_N) \in V^N, v_i \neq v_j, \forall i \neq j, v_i \neq s, t$ **do**
   1. $currPath \leftarrow \{H[s][v_1], H[v_1][v_2], \ldots, H[v_N][t]\}$
   2. $currDelay \leftarrow Delay(currPath)$
   3. **if** $minDist == -1 || currDelay \leq minDelay$ **then**
      1. $path \leftarrow currPath$
      2. $minDelay \leftarrow currDelay$
4. **return** $path$

**Theorem 4**

Complexity: $O(n^{N+2}) + nS(n, m)$, $S(n, m)$ is $O\left(\min\{m + n \log n, m \log \log C, m + (n \log C)^{0.5}\}\right)$; $C$ is the largest edge cost.
Performance of Routing strategies

Observations

- More braidedness $\implies$ more number of hops
- Works for high network density (more scope of finding right paths)
- Good for up to one recovery stage

Figure: Performance in intermediate recovery versus network size.
Similarity Index

Similarity index at a delay threshold is defined as the percentage of connections having delay less than or equal to that threshold.

Observations:
- Less randomness $\Rightarrow$ Min. distortion is same as Min. delay
- More randomness $\Rightarrow$ Min. distortion doesn't min. delay
- Need alternate strategies
Outline

1. Introduction
2. Theoretical model
3. Route construction
4. Genetic Algorithms
5. Implementation in NS2
6. Conclusion
7. Publications
8. Bibliography
Representation

Challenge: How to do: Represent, mutate, crossover select??

Genome Representation

Apart from the source, destination and intermediate nodes, each gene stores the corresponding paths as follows:

<table>
<thead>
<tr>
<th>Path1a</th>
<th>Path1b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path2a</td>
<td>Path2b</td>
</tr>
</tbody>
</table>

Initial Population

- All possible intermediate nodes covered
- For each intermediate node, consider shortestpathpair(src,inter) U shortestpathpair(inter,dest)
- Other possible path pairs are generated by: deleting a randomly selected link from the shortest path pair; and repeating above step
Genetic Algorithm Description:

Mutation:

CrossOver:
Genetic Algorithm Description:

Mutation:
Randomly select one path

CrossOver:
Genetic Algorithm Description:

Mutation:
Randomly select a link in that path

CrossOver:
Genetic Algorithm Description:

Mutation:
Delete that link from original network; delete all links from the other path; find the shortest path to destination;

CrossOver:
Genetic Algorithm Description:

Mutation:

CrossOver:
Genetic Algorithm Description:

Mutation:

CrossOver:
Select one path from each of the genes
Genetic Algorithm Description:

Mutation:

CrossOver:
Find out two common nodes in them
Genetic Algorithm Description:

Mutation:

CrossOver:
Exchange the path between those two nodes
Genetic Algorithms

**Mutation Algorithm**

```plaintext
procedure MUTATE(Genome g)
    path ← Random(g.Path11, g.Path12, g.Path21, g.Path22)
    (deli, delj) ← ran(pathlength(path))
    Delete(G, (deli, delj))
    for all \((i, j) \in \text{Disj(path)}\) do
        Delete(G, (i, j))

    apppath ← Dijkstra(deli, t)
    path ← Trim(path, (deli, delj)) \cup apppath
    Add(G, (deli, delj))
    for all \((i, j) \in \text{Disj(path)}\) do
        Add(G, (i, j))
```

**procedure** DISJ(path)

Returns the disjoint subpath of the other description

---

**Theorem 5:**

Complexity of Mutation operator is \(S(n, m)\), the complexity of a single Dijkstra’s computation.
Cross over algorithm

**procedure** CROSSOVER(Genome $g_1$, Genome $g_2$)

```plaintext
path1 ← Random($g_1$.Path11, $g_1$.Path12, $g_1$.Path21, $g_1$.Path22)
p path2 ← Random($g_2$.Path11, $g_1$.Path12, $g_1$.Path21, $g_1$.Path22)
[sub1, sub2] ← FindCommon(path1, path2)
if [sub1, sub2] ≠ [NULL, NULL] then
    path1 new ← Subpath(path1, path1.source, sub1) ∪ Subpath(path2, sub1, sub2) ∪ Subpath(path1, sub2, path1.dest)
    path2 new ← Subpath(path2, path2.source, sub2) ∪ Subpath(path1, sub1, sub2) ∪ Subpath(path2, sub2, path2.dest)
    path1 ← path1 new
    path2 ← path2 new
```

**Theorem 6:**

Complexity of Crossover operator is $O(N_1 + N_2)$, where $N_1$ and $N_2$ are the number of edges in subpaths chosen for cross over.
### Results

**Theorem 5,6,7: for one path pair**

\[ O(n^2 M) \], where \( M \) is the max. num. of nodes in subpaths in crossovers;

**Why use these?**

**Table:** Performance of different routing strategies

<table>
<thead>
<tr>
<th>Routing strategy</th>
<th>Delay</th>
<th>Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN-DIST</td>
<td>34.8112</td>
<td>32.1909</td>
</tr>
<tr>
<td>MIN-DELAY</td>
<td>18.119</td>
<td>34.535</td>
</tr>
<tr>
<td>GA</td>
<td>21.9597</td>
<td>32.7261</td>
</tr>
</tbody>
</table>

- MIN-DELAY - Fast but poor
- MIN-DIST - good but slow
- GA - intermediate
**Figure:** Variation of delay with network density for different routing strategies

[Graph showing delay variation with network density for different routing strategies: MINDELAY, MINDIST, GA]
**Figure:** Variation of distortion with network density for different routing strategies

**Observations**

Significantly low distortion for nearly same delay!
Implementation

**Figure:** Intermediate recovery system for video
## Simulation Parameters

**Table:** Parameters used for simulation

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video sequence</td>
<td>foreman</td>
</tr>
<tr>
<td>Number of frames</td>
<td>400</td>
</tr>
<tr>
<td>Resolution</td>
<td>176x144 (QCIF)</td>
</tr>
<tr>
<td>Number of descriptions</td>
<td>2</td>
</tr>
<tr>
<td>Frame rate</td>
<td>7.5/sec</td>
</tr>
<tr>
<td>MTU size</td>
<td>1024B</td>
</tr>
<tr>
<td>MAC type</td>
<td>802.11</td>
</tr>
<tr>
<td>Max. Velocity</td>
<td>70m/sec</td>
</tr>
<tr>
<td>Transm. power</td>
<td>0.2818 (250m)</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1e5</td>
</tr>
<tr>
<td>Frequency</td>
<td>914MHz</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>3/4</td>
</tr>
<tr>
<td>Distance between consecutive nodes</td>
<td>150m</td>
</tr>
</tbody>
</table>
Some plots

![Graph showing PSNR vs. velocity for different recovery stages](image-url)

- **One intermediate recovery stage**
- **No intermediate recovery**
Sample Videos

Without intermediate recovery

(a) MDC-MPT  (b) MDC with intermediate recovery

Figure: Particular frame of foreman sequence
Outline

1. Introduction
2. Theoretical model
3. Route construction
4. Genetic Algorithms
5. Implementation in NS2
6. Conclusion
7. Publications
8. Bibliography
Key Contributions

- End-to-end distortion of MDC-MPT systems was studied in a random wireless ad hoc network using physical layer channel parameters.
- Intermediate recovery achieves good performance gains in comparison to the traditional multipath system.
- A cross-layer optimization problem was then formulated to construct meshed multiple paths.
- A simple heuristic routing strategy was proposed that gives good quality paths with intermediate recovery stages.
- It was observed that, the two minimization criteria are not jointly achieved if the link costs vary widely.
- Alternate routing strategies (Genetic algorithms) were explored to avoid the problems associated with heuristic routing.
- Using simulations in NS2, it was proved that intermediate recovery technique indeed achieves considerable PSNR improvement.
Outline

1 Introduction
2 Theoretical model
3 Route construction
4 Genetic Algorithms
5 Implementation in NS2
6 Conclusion
7 Publications
8 Bibliography
Accepted

Uma Parthavi Moravapalle and Swades De, *Mesh Routing for Error Resilient Delivery of Multiple-Description Coded Image/Video Content*, in 21st International Conference on Computer Communication Networks, Munich, Germany, July-August 2012 (to appear)

A category, EIC = 0.69 (Among all CS related conferences)
Outline

1 Introduction
2 Theoretical model
3 Route construction
4 Genetic Algorithms
5 Implementation in NS2
6 Conclusion
7 Publications
8 Bibliography
References I


## Bibliography

**Thank you!**

<table>
<thead>
<tr>
<th></th>
<th>Summer 2011</th>
<th>Semester 1a</th>
<th>Semester 1b</th>
<th>Semester 2a</th>
<th>Semester 2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature reading</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theoretical framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing heuristics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genetic Algorithms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>