Acknowledgements

Significant assistance was provided by NCDOT. The GIS Unit provided the linear referencing system conceptual design and assisted in developing the link to and display from AcrView. The Pavement Management Unit provided the data for New Hanover and Pender counties, office space to work, and access to and use of their computer and software. This assistance is gratefully acknowledged.
TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 Introduction
   1.1 Background
   1.2 Purpose of Study
   1.3 Goals
   1.4 Case Studies
   1.5 Research Approach
   1.6 Context and Study Justification
   1.7 Database Presentation Format

2.0 The Role of Tools
   2.1 Spatial and Non-Spatial Data
   2.2 Data Interoperability
   2.3 Integration Inhibitors
   2.4 Software Inhibitors

3.0 Referencing Systems
   3.1 Topology, Geometry, and Attributes
   3.2 Generic Link Node System Definition
   3.3 Posted Route System Definition
   3.4 Mile Post System Definition
   3.5 Linear Referencing System Definition
   3.6 Linear Referencing System Concepts
   3.7 Referencing System Adoption
   3.8 LRS IDs

4.0 System Architecture

5.0 Database Schema
   5.1 PMU Link Node Referencing System
       5.1.1 Link Node Spatial Topology Tables
       5.1.2 Link Node Non-Spatial Attribute Table
       5.1.3 Observations on the PMU Tables
   5.2 LRS Implementation Example
       5.2.1 LRS Spatial Data
       5.2.2 LRS Non-Spatial Attribute Data
       5.2.3 Other Tables
   5.3 Study Goals
   5.4 Study Generalizability

6.0 General Constraint and Database Table(s) for LRS Algorithms
   6.1 General Constraint
   6.2 Database Table(s)
7.0 Primary Route Algorithms for generating LRS IDs
   7.1 Posted Route Chain Traversal
   7.2 Manual Algorithm
      7.2.1 Manual Algorithm Constraint
   7.3 Matching Posted Route Algorithm
      7.3.1 Matching Posted Route Algorithm Constraint
   7.4 Longest Posted Route Algorithm
      7.4.1 Longest Posted Route Algorithm Constraint
   7.5 Long Route Algorithm
      7.5.1 Long Route Algorithm Constraint
   7.6 Longest Route Algorithm

8.0 Secondary Route Algorithms for generating LRS IDs
   8.1 Posted Route Chain Traversal
   8.2 Manual Algorithm
      8.2.1 Manual Algorithm Constraint
   8.3 Matching County Posted Route Algorithm –
      Stop at County Boundary and New Route
      8.3.1 Matching Posted Route Algorithm Constraint
   8.4 Matching State Posted Route Algorithm –
      No Stop at County Boundary and New Route
      8.4.1 Matching Posted Route Algorithm Constraint
   8.5 Long County Route Algorithm –
      Stop Only at County Boundary
      8.5.1 Long County Route Algorithm Constraint
   8.6 Long State Route Algorithm –
      No Stopping at County Boundary
      8.6.1 Long State Route Algorithm Constraint
   8.7 Longest County/State Route Algorithm

9.0 Algorithm Comparison
   9.1 Quality Measures for LRS Routes
      9.1.1 Length of LRS Routes
      9.1.2 Number of LRS Routes
      9.1.3 Processing Time
      9.1.4 Other Measures
   9.2 Quality Measure Results
      9.2.1 Length of LRS Routes
      9.2.2 Number of LRS Routes
      9.2.3 Processing Time
      9.2.4 Other Measures
   9.3 Evaluation Outcome and Analysis
      9.3.1 Primary Route Algorithm Outcome and Evaluation
      9.3.2 Secondary Route Algorithm Outcome and Evaluation
10.0 Conclusions and Recommendations
   10.1 Algorithm Selection
   10.2 Transition and Correlation
   10.3 Additional Testing
   10.4 Gap Roads
   10.5 Errors in Database
   10.6 LRS Mileposts

11.0 References

Appendix A.0 – New Hanover & Pender Road Maps
Appendix B.0 – Resulting LRS Road Maps
Appendix C.0 – Links Sample Data
Appendix D.0 – Chains Sample Data
Appendix E.0 – Locator Sample Data
Appendix F.0 – Primary Matching Posted Route Algorithm
   F.1 Main Program
   F.2 Procedure DoPrimary
   F.3 Posted Route Traversal Procedures
   F.4 Function CrossCB
Appendix G.0 – Primary Longest Posted Route Algorithm
   G.1 Table Description
   G.2 Main Program
   G.3 Procedure SumLength
   G.4 Procedure UpdateLRS
   G.5 Procedure ChainTraversal
   G.6 Posted Route Traversal Procedures
   G.7 Function CrossCB
Appendix H.0 – Primary Long Route Algorithm
   H.1 Main Program
   H.2 Procedure GrabStartLink
   H.3 Procedure FindConnectedLink
Appendix I.0 – Primary Matching County Route Algorithm
   I.1 Main Program
   I.2 Procedure DoSecondary
   I.3 Posted Route Traversal Procedures
Appendix J.0 – Secondary Matching State Route Algorithm
   J.1 Main Program
   J.2 Procedure DoSecondary
   J.3 Posted Route Traversal Procedures
   J.4 Function CrossCB

Appendix K.0 – Secondary Longest County Route Algorithm
   K.1 Main Program
   K.2 Procedure GrabStartLink
   K.3 Procedure FindConnectedLink

Appendix L.0 – Secondary Longest State Route Algorithm
   L.1 Main Program
   L.2 Procedure GrabStartLink
   L.3 Procedure FindConnectedLink

Appendix M.0 – LRS/Link Node Figures

Appendix N.0 – Secondary Roads with Multiple Posted Routes
LIST OF TABLES

Table 3.1  Link Node System
Table 3.2  Posted Route System
Table 3.3  Milepost System
Table 3.4  LRS System
Table 3.5  Linear Locations
Table 3.6  Point Locations
Table 3.7  Intersections
Table 5.1  Chains Table
Table 5.2  Links Table
Table 5.3  Locator Table
Table 5.4  LRS Table
Table 5.5  Places Table
Table 5.6  Intersections Table
Table 5.7  County Boundaries Table
Table 5.8  Posted Routes Table
Table 5.9  Railroad Crossings Table
Table 5.10  Posted Route Classifier Table
Table 5.11  Pavement Condition Table
Table 5.12  Node Coordinates Table
Table 5.13  Route Type Table
Table 5.14  Counties Table
Table 5.15  Urban Areas Table
Table 5.16  C & U Areas Table
Table 6.1  LRSChains Table
Table 7.1  Primary Algorithm Constraints
Table 8.1  Secondary Algorithm Constraints
Table 9.1  Length of LRS Routes for Primary Roads
Table 9.2  Length of LRS Routes for Secondary Roads
Table 9.3  Number of LRS Routes for Primary Algorithms
Table 9.4  Number of LRS Routes for Secondary Algorithms
Table 9.5  Number of LRS Routes for Primary Algorithms
Table 9.6  Number of LRS Routes for Secondary Algorithms
LIST OF FIGURES

Figure 3.1  Link Node System
Figure 3.2  Posted Route System
Figure 3.3  Milepost System
Figure 3.4  LRS System
Figure 4.1  Overall System Architecture
Figure 5.1  Route Generation System Architecture
Figure 7.1  Primary Posted Route Chain Traversal
Figure 7.2  LRS Configuration for Matching Posted Route Algorithm
Figure 7.3  LRS Configuration for Longest Posted Route Algorithm
Figure 7.4  LRS Configuration for Long Route Algorithm
Figure 8.1  The Fork Problem
Figure 8.2  Secondary Posted Route Chain Traversal
Figure 8.3  LRS Configuration for Matching County Posted Route Algorithm
Figure 8.4  LRS Configuration for Matching State Posted Route Algorithm
Figure 8.5  LRS Configuration for Long County Route Algorithm
Figure 8.6  LRS Configuration for Long State Route Algorithm
Figure 9.1A  Primary – Matching Posted Route Algorithm Lengths
Figure 9.1B  Primary – Matching Posted Route Algorithm Average Lengths
Figure 9.2A  Primary – Longest Posted Route Algorithm Lengths
Figure 9.2B  Primary – Longest Posted Route Algorithm Average Lengths
Figure 9.3A  Primary – Long Route Algorithm Lengths
Figure 9.3B  Primary – Long Route Algorithm Average Lengths
Figure 9.4A  Secondary – Matching County Posted Route Algorithm Lengths
Figure 9.4B  Secondary – Matching County Posted Route Algorithm Average Lengths
Figure 9.5A  Secondary – Matching State Posted Route Algorithm Lengths
Figure 9.5B  Secondary – Matching State Posted Route Algorithm Average Lengths
Figure 9.6A  Secondary – Long County Route Algorithm Lengths
Figure 9.6B  Secondary – Long County Route Algorithm Average Lengths
Figure 9.7A  Secondary – Long State Route Algorithm Lengths
Figure 9.7B  Secondary – Long State Route Algorithm Average Lengths
Figure 9.8  Primary Algorithm Length vs. Number of LRS Routes Comparison
Figure 9.9  Secondary Algorithm Length vs. Number of LRS Routes Comparison
Figure 10.1  Gap Record Example
1.0 Introduction

The need to share information both within and outside transportation agencies is rapidly increasing. As this need increases, the need to maintain road and highway information in a homogeneous manner also increases.

One of the many problems associated with maintaining highway and road information is the lack of a unique way to reference a particular road or section of road. That is, how do we uniquely name a place on a roadway? Or given the node of a place, how do we exactly know where that place is? This problem arises because of the lack of permanent, fixed street and road naming standards. Although using “real-world” names as a means of referencing roadways within an organization may be acceptable, problems occur as names change over time and as non-unique names arise once agencies go outside of the organization and into different jurisdictions [Butler 1996].

This indicates a need for a standardized location referencing system (LRS), that is, a means of accurately describing the location of a physical entity. A linear location referencing system (LLRS) is a means of describing a physical location on a linear network. This is the type of location referencing system most useful in transportation, as the highway system is a linear network [Adams 1995]. The acronym LRS is used herein to encompass both LRS and LLRS.

A linear referencing system utilizes an LRS ID, which is simply a unique individual identification number for roadways that can be used for identification or naming purposes. This identifier would be used in much the same way that a social security number is used for unique identification.

The intent of this study was to explore different referencing systems, their definitions, and the information structures associated with each, as well as to clearly state the study goals. Alternative methods for generating a Linear Referencing System are also presented as well as the constraints imposed on the various methods. The different methods are then compared and the best LRS system is recommended.

1.1 Background

Departments of Transportation (DOTs) collect and store vast amounts of data. Surveys are conducted to gauge accident-prone intersections, pavement conditions, traffic load, etc. Although each individual survey is conducted for a specific purpose, all have a common thread – location.

Location can be described as synergistic, a combination of topology and geometry. Topology is the connectivity of a network – the roadway network in this case. Geometry is a precise location in space. Therefore, the combination of topology and geometry provides a means of referring to a specific place within the network and positioning the network within a larger framework of reference.
The primary goal, when referring to transportation data, is to relate that data back to a specific place. Since all survey data share location as a common thread, the way in which different departments refer to specific locations should be the same. Quite often, each department within DOTs collects and stores locational data differently. This results in significant amounts of money and man-hours being spent on redundant efforts for collecting and storing such information [Rasorf 2000, Rasdorf A1999].

To minimize or eliminate these redundant efforts, DOTs must organize a system that all departments may use to refer to locational data. This paper describes such a system and refers to it as a Linear Referencing System (LRS). An LRS is an approach used to refer to location. Once an agreed upon LRS is accepted as a standard, all departments may use this information to refer to specific places within the roadway network.

1.2 The Purpose of the Study

This study introduces and describes both the link node system and the linear referencing system. It briefly mentions several other referencing systems as well. The purpose of the study is to explore and define several different algorithms, which can be used to generate a linear referencing system based on a preexisting link node system.

That is, we devised a set of algorithms that use as input a network of data currently stored in a link node format. The algorithms generate as output that same network of data, but this time in an LRS format. This study then compares and evaluates each algorithm against a set of predefined criteria that measure their quality or desirability of the different outcomes of each algorithm. The objective of the work is to provide DOT with a picture of what the ideal (for NC DOT) LRS might look like, for various alternative configurations, and to provide a set of measures for each configuration, thus, enabling DOT to select a suitable one for permanent implementation.

1.3 Goals

This study explores several different LRS configuration algorithms. Each algorithm explores various constraints and how they affect the final outcome. After evaluating the outcome of the algorithms, we recommend a strategy for evaluating the design and implementation of a Linear Referencing System, provide measures for impact (based on the testing of various criteria), and recommend implementation tools (such as a geographic Information System (GIS), database, or other tools and environments) and strategies.

1.4 Case Studies

Two sample sets of data from the state of North Carolina Pavement Management Unit database were used as case studies. The data, used for both display and testing of the different algorithms, is from Pender and New Hanover Counties. By choosing this sample set, the number of records that would need to be processed was reduced from
roughly 195,000 for the entire state to about 1,200 for Pender County and about 2,500 for New Hanover. Both data sets contain primary and secondary roads. The smaller sample sets allow for faster processing, ease of checking the preliminary designs, and ease of evaluation of the results.

It should be noted that it is not yet possible to execute an algorithm on a statewide basis for two reasons. First, the data sets for each county’s data are presently being “cleaned” and are not in a state that the algorithms could operate on them successfully. What this means is that the database contains errors that would not permit the algorithms to run properly. Second, even if the first problem were solved, there are still county-to-county data inconsistencies that need to be resolved that would also effectively disable the algorithm. However, this cleanup is underway and two very high quality data sets are emerging: a tabular file containing accurate attribute data and a CAD file containing accurate graphics and linework for the roadway network.

Using these files, and the algorithm developed as a result of the work reported herein, will enable NCDOT to generate its permanent base LRS and standardize all state locational data. This standard will then be adopted for the development of all new data sets. Legacy data sets will either migrate to the new data standard or utilize conversion routines to translate between the two.

North Carolina’s Department of Transportation provided the GIS road coverage and Oracle database tables for Pender and New Hanover County, as well as the attributes associated with the file. The files provided are 95% error free. This represents a measure, meaning that the data have already been “cleaned” as noted above.

1.5 Research Approach

The overall goal of the NCDOT is produce and implement a fully functional linear referencing system, which can be used by a variety of its internal organizational units. The goal of this study is to identify and name the LRS routes comprising that LRS network. However, for this to happen, a research approach had to be developed and a sequence of activities had to be defined and completed.

The first of these activities involved a clear definition and understanding of the existing link node system. An understanding of the problems and limitations associated with such a system was crucial in order to achieve effective problem definition and identification. A link node data set was acquired from North Carolina’s DOT for New Hanover and Pender counties. This provided a clear picture of the existing data and it’s link node format.

A base linear referencing system was previously designed and adopted for NCDOT databases and data sets. In this study, algorithms were designed and developed to build that previously agreed upon LRS. Once the algorithms were clearly defined, they were tested using the data set acquired from the NCDOT. A set of quality assessment
measures was developed and each algorithm was assessed based on these measures. The “best” algorithm was chosen and recommended for implementation.

1.6 Context and Study Justification

As noted earlier, the context of this study is two of North Carolina’s 100 counties. Although this is a small number of counties, the number of records for these two counties is significant enough to generate interpretable results. This is even truer in light of the fact that some very computationally intensive algorithms are used to process this data. In order to run multiple tests with a number of algorithms and get results in a reasonable time frame, we limited the test case size. However, when the study is complete, when the factors affecting the algorithm have been evaluated, and when the final algorithm has been selected, designed, and coded the context will be expanded from the test case to all 100 counties.

1.7 Database Presentation Format

This report uses a relational database schema for representing the data. The schema is designed to work with the data as it currently exits and as it incorporates the proposed linear referencing system.

The format of the proposed design is presented in two ways. One way is purely textual and is achieved by identifying the table name along with the primary key(s) and attributes in an enumerated fashion, as follows:

**TABLE NAME (Primary Key, Attribute 1, Attribute 2, Attribute 3, …, Attribute n)**

Another way is tabular in which the column headings identify the type of data in that column and the rows contain the actual data. Both of these representations are used herein.

<table>
<thead>
<tr>
<th>Table # – TABLENAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary key</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
2.0 The Role of Tools

Geographic Information Systems (GISs) have emerged as useful and powerful tools for storing and using spatial data. They incorporate capabilities that enhance one’s ability to perform complex spatial analysis over some geographic region. Database management systems also have emerged as useful and powerful tools. They store what we refer to as tabular data, or data that describes the attributes and characteristics of other data or of physical objects.

2.1 Spatial and Non-Spatial Data

Both of these tools are finding ever increasing uses in Departments of Transportation (DOTs) and other transportation and infrastructure areas and applications, and rightly so [Primer 1994, Butler 1996]. They support improved performance of engineering systems by providing new and unique views of those systems through the spatial and attribute data that describe them. But historically GISs have been used primarily for spatial analysis and databases, or files, have been used primarily for data analysis.

Fortunately, the spatial and tabular data worlds are coming closer and closer together. GISs support some limited tabular data management and analysis capabilities. Some Database Management Systems (DBMSs) support limited spatial data management and analysis capabilities. But for large production oriented transportation and infrastructure applications, each tool needs to be used in such a way that its strength is maximized. Quite frankly, neither tool can do the other’s job very well. Thus the ultimate optimal use of these tools is in unison, in an integrated environment, where each uses the same underlying data set to do the type of work it does best.

2.2 Data Interoperability

In a traditional setting, the most common scenario has been that production programs use large data sets without making use of a geographical analysis and display capability. Likewise, GIS departments initially focus on maps and spatial representation development and often do not access engineering attribute data. This is a reality and it must simply be recognized and accounted for or system improvements will not be forthcoming.

What is desired is a way to develop, manage, and maintain data sets that can be used by both GISs and by application programs and can be moved between them with ease. Furthermore, the data sets must be designed in such a way that spatial data is inherently contained within them regardless of whether they reside in the GIS, the DBMS, or elsewhere [NCHRP 1998]. Such data interoperability is a key objective for the NC DOT.

2.3 Integration Inhibitors

One problem in achieving such a seamless integration lies in the fact that most data sets are legacy data sets – they already exist. When they were built, there wasn’t a concept of
seamless GIS/DBMS integration and, most often, there wasn’t even a concept of the need to share the data they contained. Thus, localized decisions were made regarding the spatial referencing system associated with the data. The result is that legacy systems have widely varying spatial referencing systems that actually inhibit data exchange.

Another problem in achieving seamless integration is that GIS and DBMS tools themselves have not historically been integrated. These products have been highly successful commercially as stand-alone products just like many engineering software products. Customers have been so happy to gain the productivity enhancement offered by popular programs (MicroStation for drafting, Oracle for DBMS, etc.) that they were satisfied with stand-alone operation. However, as that became the norm, and as their competitors come up to speed with similar software, customers began seeking a new computing productivity enhancement – integration.

2.4 Software Inhibitors

Integration also is particularly useful where an apparently stand-alone software program turns out to less stand-alone than originally thought. Consider CAD, for example. Its purpose has largely been to automate the process of creating complex sets of architectural and engineering drawings. It has fully met this need and many of the other software products have done the same.

However, some software is limited in its usefulness, although that may not be apparent initially. GIS software is turning out to be such software. GIS software is much like CAD in that it has great value as a presentation tool. Unfortunately, because presentation is not really its purpose, users have discovered that the true utility and value of GIS comes from attaching data to the geographic entities represented by GIS [Miles 1999]. Yet GISs strength lies in its spatial representation and spatial analysis, not in its ability to store and process data.

Thus, there has been a search in recent years for ways to link GISs and DBMSs together to synergistically gain the individual power of each (in some integrated fashion) and some additional combined benefit as well. This is now being achieved and such linkages are commercially available. Add to these new tools a common, agreed upon, standardized linear referencing system and one has a powerful transportation engineering analysis capability [NCHRP 1998, Kiel 1999]. The work done for this study contributes to the creation of that standardized LRS.
3.0 Referencing Systems

Location referencing systems may include geodetic or geographic points of reference. A geodetic referencing system defines placement on the earth’s sphere with latitude, longitude and elevation. A geographic referencing system uses planar coordinates to define a location on the earth’s surface.

Transportation poses a unique problem with respect to spatial representations and location [Adams 1995]. In the case of a state roadway network, for example, the focus of interest is on the linear nature of the network. Location is defined with respect to the lines of the network. Thus, the term linear referencing system is applicable. Furthermore, any interest in points is usually with respect to points on or close to the lines. Thus, the focus is on only a very limited portion of the overall planar space. It is this distinction that differentiates general GIS applications from GIS-Transportation applications.

The idea of linear referencing systems is not new. In fact, many linear and location referencing systems and models exists; among them are the county/route/milepost referencing system, intersection offsets, street addresses, link node system, and linear referencing system. This study focuses mainly on two – link node and linear referencing system (LRS). A brief description of each is provided and will be discussed in this section. Some fundamental principles will be presented as well. For additional details the reader is referred to any number of authors who have documented LRSs including [Dueker 1996], [Vonderohe 1995], and [Kiel 1999].

3.1 Topology, Geometry, and Attributes

A Referencing System is made up of three distinct parts: topology, geometry, and attributes [Rasdorf D1999]. Topology describes the connectivity of the components of the system and geometry describes the precise location in space of the components of the system. Taken together topology and geometry describe the spatial aspect of a component. In most cases that spatial aspect is location or position, but it may also be a measurement such as length or distance as well.

Attribute information, on the other hand, is descriptive information. For example, pavement condition, number of lanes, etc., are used to describe the characteristics of a particular link or segment of a link. The following sections will provide some specific terminology associated with topological, geometric, and attribute information.

3.2 Generic Link Node System Definition

A link node system is a way of defining a network through a series of points and arcs otherwise referred to as nodes and links, respectively [GeoDecisions 1997]. A node is simply defined as a point along an arc that marks the beginning or ending of a link. A
link is defined as an arc that connects two node points. Therefore, nodes are connected through a series of links.

Consider an example. Figure 3.1 displays a link node topology and Table 3.1 defines that same link node topology in tabular form. Each link is provided with a unique identification number, Link ID, and is associated with its FNode and TNode. The FNode (From Node) is a node that defines the “beginning” point of a link. The TNode (To Node) defines the “ending” point of a link. The topology, or connectivity, is defined by matching a link’s TNode to another link’s FNode or vice versa.

Table 3.1: Link Node System

<table>
<thead>
<tr>
<th>Link ID</th>
<th>FNode</th>
<th>TNode</th>
<th>Length (mi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L122</td>
<td>N11</td>
<td>N13</td>
<td>0.85</td>
</tr>
<tr>
<td>L123</td>
<td>N12</td>
<td>N13</td>
<td>1.00</td>
</tr>
<tr>
<td>L124</td>
<td>N13</td>
<td>N14</td>
<td>1.25</td>
</tr>
<tr>
<td>L125</td>
<td>N14</td>
<td>N15</td>
<td>0.75</td>
</tr>
<tr>
<td>L126</td>
<td>N14</td>
<td>N16</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Note that Figure 3.1 and Table 3.1 inherently imply directionality. Words like from and to imply a starting and ending point and, in the example shown, the Node ID and Link ID sequencing increases in order of FNode to TNode. But this is not a necessary constraint on the link node system. It could be that in lieu of FNode and TNode we used “node at one end” and “node at other end” as identifiers. In doing so, directionality
becomes irrelevant and node and link numbering need not imply any sequencing or directionality. Such is the case in the NC DOT Pavement Management Unit database.

### 3.3 Posted Route System Definition

The posted route system is a naming convention widely used by state DOTs. The posted route system refers to roads and highways by the name given to links, segments of links, or multiple connected links. Its advantage is that posted route naming is used in the field on all signs and on maps and is nearly universally understood. Still, the posted route system referred to here retains the use of nodes to assist in delineation of portions of roads.

As mentioned previously, problems arise when organizations use posted route names. These problems are a result of the overlap in road naming conventions. For example, one portion of roadway may have several different posted route names (previous link 123 is now both I-40 and I-85). If one department or organization refers to this portion of roadway as I-40 and another department or organization refers to the same portion of roadway as I-85, the systems may not have the ability to combine the information gathered to form a complete analysis of the route or link. Furthermore, they may be referring to entirely different portions of I-40 and I-85. Finally, as the names of roads change over time, vital historical data may be lost.

**Figure 3.2: Posted Route System**
Table 3.2: Posted Route System

<table>
<thead>
<tr>
<th>Posted Route</th>
<th>FNode</th>
<th>TNode</th>
<th>Length (mi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-85</td>
<td>N11</td>
<td>N13</td>
<td>0.85</td>
</tr>
<tr>
<td>I-40</td>
<td>N12</td>
<td>N13</td>
<td>1.00</td>
</tr>
<tr>
<td>I-40</td>
<td>N13</td>
<td>N14</td>
<td>1.25</td>
</tr>
<tr>
<td>I-85</td>
<td>N13</td>
<td>N14</td>
<td>1.25</td>
</tr>
<tr>
<td>I-40</td>
<td>N14</td>
<td>N15</td>
<td>0.75</td>
</tr>
<tr>
<td>I-85</td>
<td>N14</td>
<td>N16</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Figure 3.2 displays the posted route system and Table 3.2 defines the posted route system in tabular form. What is different here from the link node system is only that the Link ID field in the Link Node system is now replaced with a Posted Route field. All other items in the figure and table remain the same for the Posted Route system as for the Link Node system.

3.4 Milepost System Definition

The milepost system makes use of the posted route system and simply assigns a milepost marker at each node. Figure 3.3 displays the milepost system and Table 3.3 defines the milepost system in tabular form. The milepost marker indicates a running sum of distances along a posted route or assigned LRS. MP1 stands for milepost one and indicates the first milepost marker for a given link. MP1 starts at zero at the beginning of each new posted route. It also starts at zero at each county boundary. MP2 stands for milepost two and denotes the distance to some other location along the roadway such as an intersection.

Figure 3.3 obviously differs from Figures 3.1 and 3.2 in the way the nodes are labeled. Instead of being assigned a node number, the nodes are assigned a milepost marker. Note that at some nodes more than one milepost is shown. For example, the node inside the circle has mileposts MP 0.85 and MP 1.00. This indicates that two different posted routes share the node. Posted route I-85 has a milepost of 0.85 miles and posted route I-40 has a milepost of 1.00 miles at that node.

Table 3.3 is also different from Tables 3.1 and 3.2 in an important way. The use of mileposts in Table 3.3 eliminates the need to store lengths. Lengths are now calculated rather than stored. Yet at the same time, the concept of links is still implied by the new table structure in that each row in the table mimics a link.

A key problem with the milepost system is that it provides not only two names for any portion of pavement, but it also provides two mileposts. The location identified by MP 2.00 on I-40 is the same location as MP 1.85 on I-85. This is a dangerous situation from the perspective of a Geographic Information System (GIS) or database. Furthermore, this system retains the same problems encountered in the posted route system.
3.5 Linear Referencing System Definition

A linear referencing system is similar to the posted route system in that it groups multiple links together with the same name or ID. It is also the similar to the milepost system in that it provides each node with a milepost marker instead of a node number. However, a linear referencing system, unlike the posted route or milepost system, provides a unique and flexible way of referencing links, segments of links, and multiple links within roads and highways. Thus, an LRS provides a means for data transfer between and among organizations. It also allows for historical analysis because once a link is designated with an LRS ID it is permanent.
Figure 3.4 displays the LRS and Table 3.4 defines the LRS in tabular form. Any portion of road has only one identifier (LRS 1, LRS 2, LRS 3). There is no overlap of names or identification. Thus, mileage does not overlap either, except at nodes. For example, the portion of roadway on LRS 1 between MP 1.00 and MP 2.25 had no other possible MP or mileage specification. Additionally, fewer records are needed in the database to describe location as shown in Table 3.4. (Note that it is merely coincidence that LRS 2 and LRS 3 are of the same length).

Milepost markers are assigned to nodes and represent the total length of the LRS route up until that point. The total length of the LRS route is the result of the summation of the individual link lengths that comprise the LRS route. Table 3.4 provides not only the LRS ID, but also MP1 and MP2. MP1 represents an LRS Route’s very first milepost marker, which should always be 0.00. MP2 represents an LRS Route’s very last milepost marker and should always equal the sum of all the individual link lengths that comprise the LRS route.

Figure 3.4: LRS System

<table>
<thead>
<tr>
<th>LRS Route</th>
<th>MP1</th>
<th>MP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRS1</td>
<td>0.0</td>
<td>3.00</td>
</tr>
<tr>
<td>LRS2</td>
<td>0.0</td>
<td>0.85</td>
</tr>
<tr>
<td>LRS3</td>
<td>0.0</td>
<td>0.85</td>
</tr>
</tbody>
</table>
### 3.6 Linear Referencing System Concepts

This section presents the most fundamental conceptual schema for implementing the spatial topology of the recommended base LRS. A centralized approach is recommended for NCDOT, but a distributed approach is also presented so that the reader can better understand the concepts and see how they represent essentially the same solution.

In the database, linear elements, such as pavement type and speed limits, are spatially defined using the base LRS. This requires a unique **LRS ID** for each road in the route system and a **MP1** and **MP2** to mark the beginning and ending locations along the specified road. Table 3.5 shows the general table structure for linear elements. Point features, such as railroad crossings, are spatially defined with the unique **LRS ID** and **Milepost** measurement along that route as shown in Table 3.6.

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>MP1</th>
<th>MP2</th>
<th>Any Linear Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>515</td>
<td>0.00</td>
<td>1.00</td>
<td>X</td>
</tr>
<tr>
<td>623</td>
<td>68.00</td>
<td>69.40</td>
<td>X</td>
</tr>
<tr>
<td>623</td>
<td>69.40</td>
<td>71.50</td>
<td>X</td>
</tr>
</tbody>
</table>

**Table 3.5: Linear Locations**

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>Milepost</th>
<th>Any Point Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>623</td>
<td>69.20</td>
<td>X</td>
</tr>
<tr>
<td>624</td>
<td>106.20</td>
<td>X</td>
</tr>
</tbody>
</table>

**Table 3.6: Point Locations**

Table 3.7 represents an example of the spatial topology of intersections in the LRS route system. Anywhere an LRS route crosses another LRS route an intersection occurs. The easiest way in which to identify an intersection is by using a node identifier. Thus we incorporate one of the characteristics of both the link node and posted route systems. The **Milepost** field is used to represent the distance along the LRS route where the intersection is positioned. Each intersection must be represented by at least two records containing the same **Node ID**. This is because there are at least two LRS routes required to form an intersection and the node will be located at a different milepost distance for each.

Nodes are not the only way that intersections can be represented in the topology. However, they do present the database with an easier query option. Information related to each intersection could be queried using the **Node ID** as the primary key. Nodes are separately identified because they represent a unique and distinct part of the network topology. Table 3.7 is included here to illustrate the concept of nodes and their relationship to the LRS.

Other features that interact with or intersect the roadway, such as railroad crossings, do not need to be assigned nodes because they can be located solely on the basis of the **Milepost** measure shown in Table 3.6. That is, an intersection is a topological composition of two or more intersecting roads; a railroad crossing is simply a place along
a single road. A database designer may choose to name a railroad crossing, thus providing it with an identifier. However, that identification has nothing to do with the topology of the network. This is the key difference between the LRS and the Link Node systems. In the LRS, nodes can only be placed at intersections and represent a topological entity (intersection). In the Link Node system, nodes can be placed anywhere and represent either a topological entity or some other non-topological entity (railroad crossing). The LRS system thus demonstrates simplicity, elegance, and completeness.

### Table 3.7: Intersections

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>Milepost</th>
<th>Node ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>643</td>
<td>32.2</td>
<td>N25</td>
</tr>
<tr>
<td>643</td>
<td>38.9</td>
<td>N89</td>
</tr>
<tr>
<td>745</td>
<td>4.7</td>
<td>N25</td>
</tr>
<tr>
<td>745</td>
<td>7.8</td>
<td>N89</td>
</tr>
</tbody>
</table>

Tables 3.5, 3.6, and 3.7 are intended to conceptually demonstrate how the spatial topology of the LRS works. In actuality there will be an assortment of non-spatial attributes linked to these spatial tables. However, the database will not always use the LRS ID and Mileposts as keys. In some instances, other keys will be required, such as “Station Number.” In such a case, the LRS ID and Milepost fields, while still serving as candidate keys, will also serve as attributes linked to the primary key.

### 3.7 Referencing System Adoption

It should be emphasized that each organizational unit may choose to use a different location referencing method in their data collection and reporting. For example, the Traffic Safety Systems Unit describes accident locations based on a distance from a particular intersection along a roadway (anchor point or intersection offset reference). They currently use parameters such as POSTED ROUTE, INTERSECTION, and DISTANCE to locate accidents.

As noted earlier, and as described in Section 3.2, the Pavement Management Unit uses a link node referencing system. Thus, there are many alternative referencing systems currently in place, in large databases, throughout NC DOT. The goal is to have these units migrate to the LRS system, replacing their current system. However, if a unit chooses not to adopt the proposed LRS in its data collection and reporting activities, then a conversion mechanism must be developed to convert a particular referencing system into the proposed LRS. These conversion mechanisms are often referred to as filters.

### 3.8 LRS IDs

An LRS ID is a unique integer assigned to a series of links, and chains of links, that is used for identification purposes. The LRS ID is comparable to a social security number in its use. Similar to social security numbers, LRS IDs provide a way to uniquely identify something – continuous chains of links in the pavement database or actual physical
pavement in the field. An LRS ID does not replace a posted route name, just as one’s social security number does not replace one’s name. However, it does provide a unique and efficient way of storing information in a database relative to the named pavement.

Note that in the examples provided in the previous subsections, we used two different representations for LRS IDs: a simple integer (515, 724, etc.) and an alphanumeric value (LRS 1, LRS 2, etc.). The latter is used only for presentation purposes in the text and figures of this documentation. In reality, LRS IDs are arbitrarily assigned permanent integer identifiers.

There is no universal pattern for LRS ID numbering. The algorithm used to select and assign the routes, instead, defines the pattern. However, for secondary roads the NC DOT could use a code embedded in the LRS ID if it chooses to do so. For example, all secondary roads within New Hanover county could be assigned an LRS ID within the range of 1000 – 1999 and all secondary roads within Pender county could be assigned an LRS ID between 2000 – 2999. This would enable the NC DOT to recognize what county the LRS route was in simply by looking at the LRS ID number. This way of “patterning” the LRS IDs becomes especially easy if the algorithm processes secondary roads county-by-county.
4.0 System Architecture

Currently, all relational tables of the Pavement Management Unit are stored in an Oracle database. The tables store both spatial and attribute information. (The details of the PMU database will be fully illustrated in the next section.) This PMU database information is available to other applications such as ArcInfo or Arcview, which are Geographic Information Systems (GISs) that have the ability to store non-spatial information using a relational database format. They provide a limited capability to manipulate tables and values. Since ArcInfo and Arcview are GIS software packages, they obviously allow the user to robustly display the information stored within the tables.

Tables stored in a database management system such as Oracle can seamlessly be linked to ArcInfo or ArcView with a join command through a common identifier (key attribute value) located in both the Oracle table and the ArcInfo or Arcview table. With these tools in mind, the LRS algorithms described herein are implemented in Delphi 5, which is a combination of Pascal and SQL. Delphi provides the elegant programming functionality of Pascal and the database access and manipulation capabilities of SQL. The combination of Arcview and ArcInfo are used to display the roadway network and the LRS IDs. Figure 4.1 illustrates how a database management system is linked to other applications such as GIS.

Figure 4.1: Overall System Architecture
5.0 Database Schema

This section presents North Carolina Department of Transportation’s Pavement Management Unit (PMU) highway network database schema. Pavement Management currently uses the link node referencing system to store all information pertaining to the highway network. In addition to the link node system, the proposed LRS database schema is also provided. The reader is thus afforded the opportunity for a “side-by-side” comparison. Both the link node system and the LRS system have the following components: tables related to topology and geometry, tables related to attribute information, and other tables whose purpose is later explained.

5.1 PMU Link Node Referencing System

Currently the NCDOT Pavement Management Unit stores all highway/roadway information using the concept of a link node referencing system. A node point is placed at each intersection of a road. Node points are also placed at the beginning and ending points of bridges, at railroad crossings, and at county boundaries. Lines or arcs, referred to as links, connect one node to another with a line and approximate the path of the road network.

Once the links and nodes are defined, they are given a statewide unique link number (for all links) and a countywide unique node number (for all nodes). The assignment of these numbers is arbitrary, which means that no numbering pattern can presumed to be followed. The tables listed below are examples of the tables that the Pavement Management Unit uses to implement its link node system. Although the tables represent only a portion of the complete database, they are the only tables used to define the topology and geometry of the roadway network. It is from these tables that the proposed LRS algorithms originate.

5.1.1 Link Node Spatial Topology Tables

As mentioned earlier, topology describes the connectivity of a network. The following table(s) provide the structure of the topological aspects of the system. In addition to the table and the description of its attributes, Figures M.1 and M.2 from Appendix M illustrate the composition of the network.

The CHAINS table provides the topology of the roadway network by connecting each link through the use of Prev_Link and Next_Link. For all intents and purposes this table assembles individual links together into chains of links that are commonly known as routes.
**CHAINS (Link, County, Route, Beg_MP, Prev_Link, Next_Link, MP_Node)**

- **Link** – The *statewide* unique link identification number given to each arc with a beginning and ending node.
- **County** – The county the link is located in.
- **Route** – An 8-digit number that provides information such as route classification (Interstate, US, State, or Secondary road), type of route (Business, Alternate, Regular), Direction, and Posted Route Number.
- **Beg_MP** – Indicates the Posted Route Milepost marker at the beginning node of the link.
- **Prev_Link** – The link number of the link directly preceding the current link.
- **Next_Link** – The link number of the link directly following the current link.
- **MP_Node** – The node number of the Beg_MP marker.

The **LINKS** table provides information about each individual link including its beginning and ending nodes, the county that each is contained within, the length of each node, as well as information about whether it crosses a county boundary. The links are essentially the basic building blocks of routes.

**LINKS (Link, B_Node_Cnty, B_Node, End_Node_Cnty, End_Node, Sec_Length, Gap)**

- **Link** – The *statewide* unique link identification number given to each arc with a beginning and ending node.
- **B_Node_Cnty** – A number representing the county where the beginning node is located.
- **B_Node** – A *countywide* unique number representing the beginning of the link.
- **End_Node_Cnty** – A number representing the county where the ending node is located.
- **End_Node** – A *countywide* unique number representing the ending of the link.
- **Sec_Length** – The length of the link (in miles).
- **Gap** – The county that a link crosses into.

The following data tables provide a better understanding of the type of information stored in the tables defined above. The information in Table 5.1, the **CHAINS** table, is taken from the Link Node Figure M.1 in (Appendix M) and the Posted Routes and Posted MP Figure M.2 (in Appendix M). The information in Table 5.2, the **LINKS** table, is taken from the Link Node Figure M.1 (in Appendix M). Furthermore, Appendices C, D, and E each contain a subset of actual data from the PMU’s **LINKS, CHAINS, and LOCATOR** tables, respectively.

The word “Known” in the **CHAINS** table represents the fact that that links continue beyond the boundary of the figure; we simply have not shown these, but acknowledge their existence. The symbol “-” represents a null value. This means that a value does not exist for that field.
Table 5.1: CHAINS Table

<table>
<thead>
<tr>
<th>Link</th>
<th>County</th>
<th>Route</th>
<th>Beg_MP</th>
<th>Prev_Link</th>
<th>Next_Link</th>
<th>MP_Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Y</td>
<td>30000054</td>
<td>36.80</td>
<td>Known</td>
<td>-</td>
<td>N1</td>
</tr>
<tr>
<td>L2</td>
<td>X</td>
<td>30000054</td>
<td>0.00</td>
<td>-</td>
<td>L3</td>
<td>N12</td>
</tr>
<tr>
<td>L3</td>
<td>X</td>
<td>30000054</td>
<td>0.75</td>
<td>L2</td>
<td>L4</td>
<td>N14</td>
</tr>
<tr>
<td>L4</td>
<td>X</td>
<td>30000054</td>
<td>1.45</td>
<td>L3</td>
<td>Known</td>
<td>N4</td>
</tr>
<tr>
<td>L5</td>
<td>Y</td>
<td>30000078</td>
<td>42.90</td>
<td>Known</td>
<td>-</td>
<td>N2</td>
</tr>
<tr>
<td>L6</td>
<td>X</td>
<td>30000078</td>
<td>0.00</td>
<td>-</td>
<td>L7</td>
<td>N11</td>
</tr>
<tr>
<td>L7</td>
<td>X</td>
<td>30000078</td>
<td>0.65</td>
<td>L6</td>
<td>L8</td>
<td>N13</td>
</tr>
<tr>
<td>L8</td>
<td>X</td>
<td>30000078</td>
<td>1.35</td>
<td>L7</td>
<td>Known</td>
<td>N5</td>
</tr>
<tr>
<td>L9</td>
<td>X</td>
<td>30000115</td>
<td>12.50</td>
<td>Known</td>
<td>L10</td>
<td>N3</td>
</tr>
<tr>
<td>L10</td>
<td>X</td>
<td>30000115</td>
<td>13.20</td>
<td>L9</td>
<td>L11</td>
<td>N4</td>
</tr>
<tr>
<td>L11</td>
<td>X</td>
<td>30000115</td>
<td>13.95</td>
<td>L10</td>
<td>L13</td>
<td>N5</td>
</tr>
<tr>
<td>L13</td>
<td>X</td>
<td>30000115</td>
<td>14.30</td>
<td>L11</td>
<td>Known</td>
<td>N6</td>
</tr>
<tr>
<td>L9</td>
<td>X</td>
<td>30000065</td>
<td>12.50</td>
<td>Known</td>
<td>L10</td>
<td>N3</td>
</tr>
<tr>
<td>L10</td>
<td>X</td>
<td>30000065</td>
<td>13.20</td>
<td>L9</td>
<td>L11</td>
<td>N4</td>
</tr>
<tr>
<td>L11</td>
<td>X</td>
<td>30000065</td>
<td>13.95</td>
<td>L10</td>
<td>L12</td>
<td>N5</td>
</tr>
<tr>
<td>L12</td>
<td>X</td>
<td>30000065</td>
<td>14.30</td>
<td>L11</td>
<td>Known</td>
<td>N6</td>
</tr>
</tbody>
</table>

Table 5.2: LINKS Table

<table>
<thead>
<tr>
<th>Link</th>
<th>B_Node</th>
<th>End_Node</th>
<th>Sec_Length</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Y</td>
<td>N12</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>X</td>
<td>N12</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>L3</td>
<td>X</td>
<td>N14</td>
<td>0.70</td>
<td></td>
</tr>
<tr>
<td>L4</td>
<td>X</td>
<td>N4</td>
<td>1.60</td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td>Y</td>
<td>N2</td>
<td>0.55</td>
<td></td>
</tr>
<tr>
<td>L6</td>
<td>X</td>
<td>N11</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>L7</td>
<td>X</td>
<td>N13</td>
<td>0.70</td>
<td></td>
</tr>
<tr>
<td>L8</td>
<td>X</td>
<td>N5</td>
<td>1.60</td>
<td></td>
</tr>
<tr>
<td>L9</td>
<td>X</td>
<td>N3</td>
<td>0.70</td>
<td></td>
</tr>
<tr>
<td>L10</td>
<td>X</td>
<td>N4</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>L11</td>
<td>X</td>
<td>N5</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>L12</td>
<td>X</td>
<td>N6</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>L13</td>
<td>X</td>
<td>N6</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

5.1.2 Link Node Attribute Table

Attribute data, as mentioned earlier, describes a feature or an entity. Attribute data does not provide information about the connectivity (topology) or precise location (geometry) of an item. However, attribute information is essential in providing descriptive information and is needed to make the database complete and significantly more useful.

The LOCATOR table provides a description of every feature (road, bridge, etc.) at a node in the link node reference system. There is one record for every feature on a node.
**LOCATOR (County, Node, Feature)**

County – A *statewide* unique number that represents the county the link is in.

Node – A *countywide* unique number representing a roadway intersection, county boundary, or bridge beginning or ending.

Feature – Represents every item (road, bridge, etc.) at the node; there is a separate record for each feature at a node.

Table 5.3, the LOCATOR table, provides a better understanding of the type of information stored in the table defined above. The information is taken from Figure 1B in Appendix M.

<table>
<thead>
<tr>
<th>County</th>
<th>Node</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N1</td>
<td>30000054</td>
</tr>
<tr>
<td>X</td>
<td>N12</td>
<td>CB</td>
</tr>
<tr>
<td>X</td>
<td>N12</td>
<td>30000054</td>
</tr>
<tr>
<td>X</td>
<td>N14</td>
<td>RR</td>
</tr>
<tr>
<td>X</td>
<td>N14</td>
<td>30000054</td>
</tr>
<tr>
<td>X</td>
<td>N4</td>
<td>30000054</td>
</tr>
<tr>
<td>X</td>
<td>N4</td>
<td>30000065</td>
</tr>
<tr>
<td>X</td>
<td>N4</td>
<td>30000115</td>
</tr>
</tbody>
</table>

### 5.1.3 Observation on the PMU Tables

Some observations about the Pavement Management Unit’s link node system implementation are in order. At first glance, one may misinterpret the B_Node and End_Node fields in the LINKS table as denoting directionality. However, this is not the case. As mentioned in Section 3.2, the B_Node and End_Node identifiers are assigned arbitrarily and do not provide any directional information.

Still, some general directionality can be determined through the highway naming convention adopted nationwide. This naming convention states that all posted routes with an even number generally run east and west and all posted routes with an odd number generally run north and south.

More specific direction information can be determined from the CHAINS table. This table defines all routes as a chain of multiple links. Following the links in a chain traverses a posted route. A link is listed in the CHAINS table once for each of its assigned posted routes. Therefore, a link may be listed more than once if it lies on the path of more than one posted route.

The Beg_MP provides the mile marker location for one of the nodes of the link in question and is selected as the start (or beginning) node for the link. The Beg_MP value is the sum of all link lengths within the given chain (posted-route) up until that point.
The **MP_Node** is the node identifier where the milepost marker is posted. The **MP_Node** does not necessarily match a link’s **B_Node** (in the **LINKS** table).

It should also be noted that the mile posting of all routes in the PMU link node system begins with 0 and ends with the total length of the route within a **county**. This is unlike the mile posting as seen on highway markers, which begins with 0.00 and ends with the total length of the posted route **statewide**.

In addition to providing information about the beginning point of a link, the **CHAINS** table also provides the network topological information by providing information about a link’s previous and next link. The **Prev_Link** field is assigned a value of null at the beginning of a posted route chain. Likewise, the **Next_Link** field is assigned a value of null at the end of a posted route chain. The **Next_Link** field is assigned a value of null when a link encounters a county boundary. The **Prev_Link** field is assigned a value of null when the chain crosses the county boundary. Normal chain traversal occurs by following **Next_Link** after **Next_Link** until the end of the chain is reached (or **Prev_Link** to the beginning). In the case of county boundaries, however, where no next (or previous) link is identified (even though one exists), traversal can still continue. This is done simply by matching on the **B_Node** and **End_Node** of the links.

Another observation of interest regarding the PMU database lies in the county field. This field holds a numerical value (0-99) for each of the 100 counties in North Carolina. This study used county 64, New Hanover, and county 70, Pender as test cases.

The reader should further note the **LOCATOR** table. At first glance this would appear to be an attribute table providing attribute information about nodes. However, it is actually, subtly, a topological table. It identifies every route that passes through a node.

For example, consider Node 4 Figure M.2 in Appendix M. The **LOCATOR** table tells us that NC 54, NC 115, and NC 65 pass through this node. Using this information one can enter the **LINKS** table and search both the **B-Node** and the **End_Node** fields to find every link that touches this node (we find L3, L4, L9, and L10).

### 5.2 Linear Referencing System Implementation Example

This section provides examples of a set of LRS database tables. Figure M.2 in Appendix M represents a sample route system with posted routes and intersections shown. Nodes, again, are used to represent intersections, county boundaries, railroad crossings, and bridges. Figures M.1 and M.2 show the intersection and county nodes. The following tables illustrate a generic representation of how the topology and some of the attributes might be laid out in NC DOT’s ORACLE database.

#### 5.2.1 LRS Spatial Data

To best illustrate the base spatial topology and geometry, an example roadway network (figures included in Appendix M for graphical illustration) is fully described in database
terms in this section. A set of relational model database tables is presented and used to represent the spatial and geometric data of this roadway network.

All figures represent the same physical roadway networks – a simple example selected for illustration only. The example includes roadways, a county boundary, and a railroad crossing. Figure M.2 shows all of the conventional posted route information. Figure 2 shows the data for the same area in terms of the NCDOT base LRS. Finally, Figure 4 adds information to the network regarding places. Note that the data is intended more to illustrate the database design rather than to portray any actual scenario.

The following tables are used to provide the tabular data for the information shown in the figures located in Appendix M. Following these table descriptions are the actual spatial data tables that contain all the data represented in the figures located in Appendix M.

The **LRS** table, Table 5.4, defines the LRS routes. LRS routes are identified by each route’s **LRS ID**, the route’s beginning milepost (**Beg_MP**), and the route’s ending milepost (**End_MP**). Thus the route is defined in its entirety, but the definition does not include intermediate nodes or segments of the LRS route (which are defined in a separate **PLACES** table).

**LRS (LRS ID, Beg_MP, End_MP)**
- **LRS ID** – the statewide unique LRS identification number assigned to the road
- **Beg_MP** – the beginning LRS milepost value, which will always be 0 since it marks the beginning milepost for each LRS route
- **End_MP** – the ending LRS milepost value, which will always be the LRS route’s total length

The **PLACES** table, Table 5.5, allows for the identification of arbitrary segments of any road. These are identified by the road’s **LRS ID**, beginning milepost (**MP1**) and ending milepost (**MP2**). Thus, each table entry identifies a linear segment of roadway by establishing a relationship between places and the LRS referencing system. The **Place ID** is the name given to each of these roadway segments. A place may occur at any arbitrary location on the roadway network.

**PLACES (LRS ID, MP1, Measured From 1, MP2, Measured From 2, Place ID)**
- **LRS ID** – the statewide unique LRS identification number assigned to the road
- **MP 1** – the beginning LRS milepost value
- **Measured From 1** – The LRS milepost of the intersection/node from which the beginning milepost was measured
- **MP 2** – the ending LRS milepost value
- **Measured From 2** – the LRS milepost of the intersection/node from which the ending milepost was measured
- **Place ID** – the unique label/name given to the segment of road under consideration
The **INTERSECTIONS** table, Table 5.6, stores the nodes that represent an intersection. It gives the node identification number for each intersection, the LRS that the intersection occurs on, and the milepost it occurs at. Thus, nodes may be queried to reveal information about each intersection. Note that in the Figures in Appendix M, nodes have been placed at the ends of each route as they are drawn. These routes are actually continuous even though nodes are shown. This is only for demonstration purposes and will not occur in the actual database representation, except for roads crossing the boundaries of the State.

**INTERSECTIONS** (Node ID, LRS ID, LRS MP, Posted Route, PR MP, County Name)
- **Node ID** – name/identification given to the point of intersection
- **LRS ID** – the statewide unique LRS identification number assigned to the road
- **LRS MP** – the milepost of the intersection using LRS values
- **Posted Route** – the name (number) of the posted route
- **PR MP** – the posted milepost of intersection on the posted route
- **County Name** – the county in which the intersection is located

The **COUNTY BOUNDARIES** table, Table 5.7, stores the nodes that represent county boundaries. This table gives the posted route, posted route milepost and county name for each county boundary, depending upon the **LRS ID**, the **LRS MP** and the **Node ID** of the boundary intersection.

**COUNTY BOUNDARIES** (Node ID, LRS ID, LRS MP, Posted Route, PR MP, County Name)
- **Node ID** – name/identification given to the point of intersection
- **LRS ID** – the statewide unique LRS identification number assigned to the road
- **LRS MP** – the milepost of the intersection using LRS values
- **Posted Route** – the name (number) of the posted route
- **PR MP** – the posted milepost of intersection on the posted route
- **County Name** – the county in which the intersection is located

Note that a node is being assigned to county boundaries here. This is not in line with a topological entity, but was requested by NC DOT. Although it is not *pure*, it is not harmful.

The **POSTED ROUTES** table, Table 5.8, stores the names of posted routes that are associated with a particular LRS route. The **POSTED ROUTES** table establishes the definition of roadway segments in terms of posted routes and posted mileposts. That is, this table establishes the relationship between the LRS and the posted route referencing system. There may be one or more posted routes associated with a given LRS route. **MP1** and **MP2** fields are used to record the LRS measure along the LRS route for which the specified posted route designation applies.

**POSTED ROUTES** (LRS ID, Posted Route, Posted MP1, County Name 1, Posted MP2, County Name 2)
- **LRS ID** – the statewide unique LRS identification number assigned to the road
Posted Route – the name (number) of the posted route
Posted MP 1 – the beginning posted milepost of the route for this segment
County Name 1 – the name of the county where the beginning milepost is located
Posted MP 2 – the ending posted milepost of the route for this segment
County Name 2 – the name of the county where the ending milepost is located

The RAILROAD CROSSINGS table, Table 5.9, catalogs the locations of all railroad crossings in the sample route-system. The Milepost field is used to reference this point feature. Any number of additional attribute fields could be added to this table or a unique identifier could be added for each crossing.

**RAILROAD CROSSINGS (LRS ID, Milepost)**

- LRS ID – the statewide unique LRS identification number assigned to the road
- Milepost – the milepost of the railroad intersection using LRS values

Data Tables 5.4 through 5.9 represent the type information stored in the tables described above. All information is taken directly from Figures M.3 and M.4 in Appendix M. The reader is referred to these figures.

The reader should note that the MP1 values in Table 5.4 do not necessarily start with 0.00. This reflects what is depicted on Figure M.3 in Appendix M and would not be the case with actual LRS routes. All LRS routes will start with 0.00, but Table 5.4 is merely the tabular representation of Figure M.3 (in Appendix M).

**Table 5.4: LRS Table**

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>MP 1</th>
<th>MP 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>515</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>624</td>
<td>105.00</td>
<td>108.50</td>
</tr>
<tr>
<td>742</td>
<td>25.00</td>
<td>27.25</td>
</tr>
<tr>
<td>623</td>
<td>68.00</td>
<td>71.50</td>
</tr>
</tbody>
</table>

**Table 5.5: PLACES Table**

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>MP 1</th>
<th>Measured From 1</th>
<th>MP 2</th>
<th>Measured From 2</th>
<th>Place ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>623</td>
<td>68.00</td>
<td>68.00</td>
<td>71.50</td>
<td>71.50</td>
<td>2</td>
</tr>
<tr>
<td>742</td>
<td>25.00</td>
<td>25.00</td>
<td>26.80</td>
<td>26.80</td>
<td>4</td>
</tr>
<tr>
<td>742</td>
<td>26.45</td>
<td>26.45</td>
<td>26.80</td>
<td>26.80</td>
<td>11</td>
</tr>
<tr>
<td>742</td>
<td>26.80</td>
<td>26.80</td>
<td>27.25</td>
<td>27.25</td>
<td>12</td>
</tr>
<tr>
<td>623</td>
<td>68.00</td>
<td>68.00</td>
<td>68.55</td>
<td>68.55</td>
<td>15</td>
</tr>
<tr>
<td>624</td>
<td>105.00</td>
<td>105.00</td>
<td>105.45</td>
<td>105.45</td>
<td>19</td>
</tr>
<tr>
<td>742</td>
<td>25.00</td>
<td>25.00</td>
<td>25.70</td>
<td>25.70</td>
<td>22</td>
</tr>
<tr>
<td>624</td>
<td>105.45</td>
<td>105.45</td>
<td>106.90</td>
<td>106.90</td>
<td>30</td>
</tr>
<tr>
<td>624</td>
<td>106.90</td>
<td>106.90</td>
<td>108.50</td>
<td>108.50</td>
<td>31</td>
</tr>
<tr>
<td>623</td>
<td>68.55</td>
<td>68.55</td>
<td>69.90</td>
<td>69.90</td>
<td>32</td>
</tr>
<tr>
<td>623</td>
<td>69.90</td>
<td>69.90</td>
<td>71.50</td>
<td>71.50</td>
<td>33</td>
</tr>
<tr>
<td>742</td>
<td>25.70</td>
<td>25.70</td>
<td>26.45</td>
<td>26.45</td>
<td>34</td>
</tr>
</tbody>
</table>
### Table 5.6: INTERSECTIONS Table

<table>
<thead>
<tr>
<th>Node ID</th>
<th>LRS ID</th>
<th>LRS MP</th>
<th>Posted Route</th>
<th>PR MP</th>
<th>County Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>624</td>
<td>105.00</td>
<td>54</td>
<td>36.8</td>
<td>Y</td>
</tr>
<tr>
<td>N4</td>
<td>624</td>
<td>106.90</td>
<td>54</td>
<td>1.45</td>
<td>X</td>
</tr>
<tr>
<td>N8</td>
<td>624</td>
<td>108.50</td>
<td>54</td>
<td>3.05</td>
<td>X</td>
</tr>
<tr>
<td>N2</td>
<td>623</td>
<td>68.00</td>
<td>78</td>
<td>42.90</td>
<td>Y</td>
</tr>
<tr>
<td>N5</td>
<td>623</td>
<td>69.90</td>
<td>78</td>
<td>1.35</td>
<td>X</td>
</tr>
<tr>
<td>N9</td>
<td>623</td>
<td>71.50</td>
<td>78</td>
<td>2.95</td>
<td>X</td>
</tr>
<tr>
<td>N6</td>
<td>515</td>
<td>0.00</td>
<td>115</td>
<td>14.30</td>
<td>X</td>
</tr>
<tr>
<td>N10</td>
<td>515</td>
<td>1.00</td>
<td>115</td>
<td>15.30</td>
<td>X</td>
</tr>
<tr>
<td>N3</td>
<td>742</td>
<td>25.00</td>
<td>65</td>
<td>12.50</td>
<td>X</td>
</tr>
<tr>
<td>N3</td>
<td>742</td>
<td>25.00</td>
<td>115</td>
<td>12.50</td>
<td>X</td>
</tr>
<tr>
<td>N4</td>
<td>742</td>
<td>25.70</td>
<td>65</td>
<td>13.20</td>
<td>X</td>
</tr>
<tr>
<td>N4</td>
<td>742</td>
<td>25.70</td>
<td>115</td>
<td>13.20</td>
<td>X</td>
</tr>
<tr>
<td>N5</td>
<td>742</td>
<td>26.45</td>
<td>65</td>
<td>13.95</td>
<td>X</td>
</tr>
<tr>
<td>N5</td>
<td>742</td>
<td>26.45</td>
<td>115</td>
<td>13.95</td>
<td>X</td>
</tr>
<tr>
<td>N6</td>
<td>742</td>
<td>26.80</td>
<td>65</td>
<td>14.30</td>
<td>X</td>
</tr>
<tr>
<td>N6</td>
<td>742</td>
<td>26.80</td>
<td>115</td>
<td>14.30</td>
<td>X</td>
</tr>
<tr>
<td>N7</td>
<td>742</td>
<td>27.25</td>
<td>6</td>
<td>14.75</td>
<td>X</td>
</tr>
</tbody>
</table>

### Table 5.7: COUNTY BOUNDARIES Table

<table>
<thead>
<tr>
<th>Node ID</th>
<th>LRS ID</th>
<th>LRS MP</th>
<th>Posted Route</th>
<th>PR MP</th>
<th>County Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>N12</td>
<td>624</td>
<td>105.45</td>
<td>54</td>
<td>0.00</td>
<td>X</td>
</tr>
<tr>
<td>N12</td>
<td>624</td>
<td>105.45</td>
<td>54</td>
<td>37.25</td>
<td>Y</td>
</tr>
<tr>
<td>N11</td>
<td>623</td>
<td>68.55</td>
<td>78</td>
<td>0.00</td>
<td>X</td>
</tr>
<tr>
<td>N11</td>
<td>623</td>
<td>68.55</td>
<td>78</td>
<td>43.45</td>
<td>Y</td>
</tr>
</tbody>
</table>

### Table 5.8: POSTED ROUTES Table

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>Posted Route</th>
<th>Posted MP1</th>
<th>County Name 1</th>
<th>Posted MP2</th>
<th>County Name 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>515</td>
<td>115</td>
<td>14.30</td>
<td>X</td>
<td>15.30</td>
<td>X</td>
</tr>
<tr>
<td>742</td>
<td>115</td>
<td>12.50</td>
<td>X</td>
<td>14.30</td>
<td>X</td>
</tr>
<tr>
<td>742</td>
<td>65</td>
<td>12.50</td>
<td>X</td>
<td>14.75</td>
<td>X</td>
</tr>
<tr>
<td>623</td>
<td>78</td>
<td>42.90</td>
<td>Y</td>
<td>2.95</td>
<td>X</td>
</tr>
<tr>
<td>624</td>
<td>54</td>
<td>36.80</td>
<td>Y</td>
<td>3.05</td>
<td>X</td>
</tr>
</tbody>
</table>

### Table 5.9: RAILROAD CROSSINGS Table

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>Milepost</th>
</tr>
</thead>
<tbody>
<tr>
<td>623</td>
<td>69.20</td>
</tr>
<tr>
<td>624</td>
<td>106.20</td>
</tr>
</tbody>
</table>
A number of observations are in order. First, the general topology of the roadway network is captured using these tables and the concepts embodied in the NCDOT base LRS. The tables are compact and implement efficiently. They centralize topology and geometry to enhance maintenance. They also contain appropriate references to adjust the geometry if corrections need to be made and they allow this to be done without jeopardizing the integrity of the database. Finally, they contain the appropriate correlation with the county/route/milepost referencing scheme. This, then, is a direct representation that allows appropriate transformations between the posted route and LRS database representations.

As a reminder, the data contained in these tables is shown for illustrative purposes only. All data is derived from the example network shown in the figures.

5.2.2 LRS Non-Spatial Attribute Data

The previous tables stored information fully describing the topology and geometry of the route system. Tables 5.10 through 5.12 illustrate the storage of non-spatial attribute information on this network.

The POSTED ROUTE CLASSIFIER table, Table 5.10, reports the type, or classification of the posted route.

**POSTED ROUTE CLASSIFIER (Posted Route, Classifier)**

- Posted Route – the posted name (number) of the posted route
- Classifier – the type of route (interstate, us, state, etc.)

<table>
<thead>
<tr>
<th>Posted Route</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>115</td>
<td>state</td>
</tr>
<tr>
<td>78</td>
<td>state</td>
</tr>
<tr>
<td>54</td>
<td>state</td>
</tr>
<tr>
<td>65</td>
<td>state</td>
</tr>
</tbody>
</table>

The PAVEMENT CONDITION table, Table 5.11, stores the pavement condition found along each route in the sample route-system for the distributed approach. Pavement condition is a linear attribute so it uses an MP1 and an MP2 as a spatial delineation.

**PAVEMENT CONDITION (LRS ID, MP1, MP2, Pavement Condition)**

- LRS ID – the statewide unique LRS identification number assigned to the road
- MP1 – the beginning posted milepost of the route for this segment
- MP2 – the ending posted milepost of the route for this segment
- Pavement Condition – an attribute value used to describe the condition of the pavement for the segment defined
Table 5.11: PAVEMENT CONDITION Table

<table>
<thead>
<tr>
<th>LRS ID</th>
<th>MP1</th>
<th>MP2</th>
<th>Pavement Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>515</td>
<td>0.00</td>
<td>1.00</td>
<td>Good</td>
</tr>
<tr>
<td>623</td>
<td>68.00</td>
<td>69.40</td>
<td>Moderate</td>
</tr>
<tr>
<td>623</td>
<td>69.40</td>
<td>71.50</td>
<td>Good</td>
</tr>
<tr>
<td>624</td>
<td>105.00</td>
<td>106.70</td>
<td>Good</td>
</tr>
<tr>
<td>624</td>
<td>106.70</td>
<td>108.50</td>
<td>Excellent</td>
</tr>
<tr>
<td>742</td>
<td>25.00</td>
<td>26.45</td>
<td>Poor</td>
</tr>
<tr>
<td>742</td>
<td>26.45</td>
<td>26.80</td>
<td>Moderate</td>
</tr>
<tr>
<td>742</td>
<td>26.80</td>
<td>27.25</td>
<td>Poor</td>
</tr>
</tbody>
</table>

Finally, one additional attribute table is illustrated. This table deals with GPS data; it identifies those points within the overall highway network whose precise geometric location is known. The form of this data is shown below in the NODE COORDINATES table, Table 5.12.

NODE COORDINATES (Node ID, N, E, Z)

Node ID – name/identification given to the point of intersection
N – the northing coordinate value location of the node
E – the easting coordinate value location of the node
Z – the elevation of the node

Table 5.12: NODE COORDINATES Table

<table>
<thead>
<tr>
<th>Node ID</th>
<th>N</th>
<th>E</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>121</td>
<td>412</td>
<td>312</td>
</tr>
<tr>
<td>N2</td>
<td>115</td>
<td>513</td>
<td>336</td>
</tr>
<tr>
<td>N3</td>
<td>201</td>
<td>378</td>
<td>420</td>
</tr>
<tr>
<td>N4</td>
<td>203</td>
<td>459</td>
<td>378</td>
</tr>
<tr>
<td>N5</td>
<td>207</td>
<td>488</td>
<td>245</td>
</tr>
<tr>
<td>N6</td>
<td>209</td>
<td>569</td>
<td>365</td>
</tr>
<tr>
<td>N7</td>
<td>214</td>
<td>642</td>
<td>398</td>
</tr>
<tr>
<td>N8</td>
<td>298</td>
<td>449</td>
<td>425</td>
</tr>
<tr>
<td>N9</td>
<td>299</td>
<td>502</td>
<td>403</td>
</tr>
<tr>
<td>N10</td>
<td>296</td>
<td>523</td>
<td>397</td>
</tr>
<tr>
<td>C1</td>
<td>136</td>
<td>414</td>
<td>357</td>
</tr>
<tr>
<td>C2</td>
<td>145</td>
<td>516</td>
<td>437</td>
</tr>
</tbody>
</table>

In this table, a subset of LRS nodes would be represented: those where the precise North Carolina grid coordinate positions are known. This table establishes a link between the geometry of the lines in the stored network (by such tools as ARC/INFO and MicroStation) and their actual location within the state or within the geodetic realm. These coordinate values may also establish a connection with GPS data collection devices, as these are increasingly used within NCDOT and elsewhere.
5.2.3 Other Tables

The next five table formats are not actually needed for the database. However, they provide NCDOT with archival historical information, even though the data represented are no longer used as they once were. The last table, C & U AREA NAMES, is a combination of the URBAN AREAS and COUNTIES tables.

The ROUTE TYPE table, Table 5.13, associates a code with each type of roadway classifier.

**ROUTE TYPE (Classifier Code, Classifier Description)**
- Classifier Code – numerical value given that relates to a specific type of route
- Classifier Description – the type of route (interstate, us, state, etc.)

The COUNTIES table, table 5.14, establishes a linkage between the county name and a standard numerical NC County ID.

**COUNTIES (County ID, County name)**
- County ID – identification number given to a specific county
- County name – name of the county of interest

The URBAN AREAS table, Table 5.15, gives the urban area name, or city name, for a numerically designated NC urban area ID.

**URBAN AREAS (Urban ID, Urban Area Name)**
- Urban ID – identification number for a specific urban area, or city
- Urban Area Name – name of the city or urban area

The C & U AREAS table, Table 5.16 gives the name for a county or urban area when given a specific county or urban ID.

**C & U AREAS (C or U ID, C or U Area Name)**
- C or U ID – identification number for a specific county or urban area
- C or U Area Name – name of a specific county or urban area

These are the tables that are considered to be redundant in the database. They are provided only for DOT use to obtain historical numerical codes, which previously serve as keys to access data. In the new database design they are not needed. For example, previously one would indicate that a roadway segment was an interstate roadway segment by having the numerical value 1 appear in its ID number. Today, if something is an interstate, we can just state that it is an interstate. Therefore, the use of use of numerical codes is not needed. Thus, the lay person or the casual user can now understand and use the database as well as the person who knows all the codes.
Table 5.13: ROUTE TYPE Table

<table>
<thead>
<tr>
<th>Classifier Code</th>
<th>Classifier Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>interstate</td>
</tr>
<tr>
<td>2</td>
<td>us</td>
</tr>
<tr>
<td>3</td>
<td>state</td>
</tr>
<tr>
<td>4</td>
<td>secondary</td>
</tr>
<tr>
<td>5</td>
<td>local/city</td>
</tr>
<tr>
<td>6</td>
<td>state park</td>
</tr>
<tr>
<td>7</td>
<td>federal domain</td>
</tr>
<tr>
<td>8</td>
<td>ramps</td>
</tr>
<tr>
<td>9</td>
<td>projected</td>
</tr>
</tbody>
</table>

Table 5.14: COUNTIES Table

<table>
<thead>
<tr>
<th>County ID</th>
<th>County Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>031</td>
<td>X</td>
</tr>
<tr>
<td>047</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 5.15: URBAN AREAS Table

<table>
<thead>
<tr>
<th>Urban ID</th>
<th>Urban Area Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>Charlotte</td>
</tr>
<tr>
<td>107</td>
<td>Raleigh</td>
</tr>
</tbody>
</table>

Table 5.16: C & U AREAS Table

<table>
<thead>
<tr>
<th>C or U ID</th>
<th>C or U Area Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>031</td>
<td>X</td>
</tr>
<tr>
<td>047</td>
<td>Y</td>
</tr>
<tr>
<td>103</td>
<td>Charlotte</td>
</tr>
<tr>
<td>107</td>
<td>Raleigh</td>
</tr>
</tbody>
</table>

5.3 Study Goal

Section 5.1 has provided the structure of Pavement Management’s existing link node system. The goal of this study is to devise ways to move from the existing system described in Section 5.1 to the linear referencing system described in Section 5.2 (as shown in Figure 5.1). This move from the link node system to the LRS system can be accomplished by utilizing the existing link node tables in the Pavement Management Unit database.

The link node database contains all the information needed to generate the LRS routes and LRS tables. Therefore, the primary study goal is to use the existing system to create the new LRS system, i.e., to generate the LRS routes. Once the LRS routes have been generated, all related LRS tables can be created.
The work described herein makes use of the link node database tables to create the LRS routes by generating LRS IDs for all links in the roadway network. This study does not create new database tables with the new LRS. Figure 5.1 shows how the LRS algorithms move from the link node system to the LRS.

![Figure 5.1: Route Generation System Architecture](image)

It should be emphasized that it was not a goal of this study to determine whether there is a need for the new LRS system. In a study prior to this it was already concluded that a base LRS system was needed. This work builds on that recommendation by providing insight into how to do it. What this study does is provide the DOT with several LRS configurations to choose from. In addition to multiple configurations, this report also provides an interpretation of the analysis results of the various configurations so that the process of choosing an approach is well founded.

### 5.4 Study Generalizability

Other states could use this base LRS definition. But, how would they do it? In North Carolina the Pavement Management Unit had a link node database that was available and could be used to generate the LRS routes. Given the results of the study, the selected algorithm can be run using the link node database as input and generating the LRS routes and IDs as output.

Other states might choose a manual approach. In North Carolina the total road mileage is approximately 78,000 miles and the number of roads in the link node database is approximately 195,000. With these large numbers a manual approach is not feasible. However, for a state with fewer roads, a manual approach may indeed be a reasonable approach.

Other ways might exist to create the base LRS. The NC DOT study started with a link node referencing system. Other location referencing systems could be used as the starting point as well. If another state had a different referencing system implemented, that system could be used to generate the LRS.
Finally, the LRS can be generated using a Geographic Information System (GIS), if the roadway network is stored in a GIS. An analyst could either do this manually or by writing route generation code using the GIS itself. Especially for a small state that had a GIS already implemented, a manual GIS approach would be excellent. For a large state that had a GIS already implemented, the GIS algorithm approach would work well.
6.0 General Constraints and Database Table(s) for LRS Algorithms

The following two subsections provide general information that affects all the algorithms described in Sections 7.0 and 8.0. The general constraints section describes general roadway description constraints that must be accounted for in all LRS algorithms. They enable us to fully understand how the current system works so that our algorithms are properly constructed and their result is meaningful. The general database table section (Section 6.2) defines the PMU table that was used as input to the LRS algorithms.

6.1 General Constraint

The algorithms that were developed to generate alternative LRS route configurations needed to take into account a number of specific predefined constraints; the constraints limit how the configurations were created. These constraints place restrictions on the definition of the precedence of the road classification scheme for generating LRS IDs, and on the defined coverage area for LRS routes. The constraints apply to all LRS algorithms.

The algorithms described in Section 7.0 are divided into primary route algorithms and secondary route algorithms. A route is classified as a primary route if its posted route number denotes an Interstate, U.S., or State route. A secondary route is any other state-maintained roadway. Primary posted routes are uniquely named throughout the United States for Interstate and U.S. routes and are uniquely named statewide for State routes. This means that Interstate and U.S. posted route numbers do not change when crossing state lines and State posted route names do not change when crossing county lines. The Inset Map of New Hanover and Pender Counties located in Appendix A illustrates these observations.

Secondary posted routes are uniquely named countywide. Secondary posted route names can, and often do, change when crossing county lines. For example, refer to the Inset Map of New Hanover and Pender Counties located in Appendix A. Secondary route SR1336 changes to SR1572 upon crossing the county boundary. The Inset map also demonstrates a unique case where the secondary route SR1002 remains the same when crossing the county boundary. Additionally, secondary posted route names may also change at roadway intersections within the same county.

Another factor that affects the algorithm design is the way primary and secondary posted route names are assigned to a segment of pavement. Primary roads may have multiple posted route names assigned to the same pavement whereas a secondary posted route name is unique to a section of pavement. For example, on the Pender County road map located in Appendix A, US17 is also NC210 between the two NC210 segments.

Secondary roads are supposed to be assigned a single posted route name for a particular section of pavement. Out of the 143,729 secondary roadway segments in the entire 100 county PMU database, only 67 links were assigned multiple secondary posted route names. These are most likely database errors. For example, in Appendix N, links
215210 and 215209 are assigned both posted route IDs of 40001319 and 40001439. This is more than likely a data entry error. As mentioned previously, PMU is in the process of “cleaning” the link node database. (It should be noted that for New Hanover and Pender Counties there are no instances of dual naming.)

It is also necessary to specify road classification precedence with respect to Interstate, US, State, and Secondary roadways. Any link defined as an Interstate link, which came from the conventional naming of a road, takes precedence over US, State, and Secondary links. Any link classified as US takes precedence over State and Secondary links. And, finally, State links take precedence over Secondary links. Therefore, all Interstate links are assigned an LRS ID before US, State, and Secondary links; all US links are assigned an LRS ID before State and Secondary links; and, all State links are assigned an LRS IDs before Secondary links.

Finally, the LRS coverage area must be defined. An LRS system can be configured on either a countywide or statewide basis. For primary roads posted routes are defined statewide and, therefore, primary LRS routes should not stop at county lines, but continue on to the state boundary. However, secondary routes should be treated differently. Algorithms have been developed to test both scenarios – crossing the county boundary and not crossing the county boundary. The results of applying a set of quality measures to each test set will mandate which approach is best suited for the LRS system.

6.2 Database Table

The algorithms discussed in Sections 7.0 and 8.0 use a table with the same format as Table 6.1, the LRSCHAINS table, shown below. This table is composed of the attributes B_Node, End_Node, and Sec_Length from the PMU LINKS table and all of the attributes from the PMU CHAINS table (see Section 5.1.1 for a detailed explanation of these two tables). Joining the two tables based on the Link attribute and deleting unnecessary columns created Table 6.1. (The Route number has been truncated to fit the space provided.)

<table>
<thead>
<tr>
<th>Link</th>
<th>County</th>
<th>Route</th>
<th>Beg_MP</th>
<th>Prev_Link</th>
<th>Next_Link</th>
<th>MP_Node</th>
<th>B_Node</th>
<th>End_Node</th>
<th>Sec_Length</th>
<th>LRS_ID</th>
<th>LRS_MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Y</td>
<td>54</td>
<td>36.8</td>
<td>-</td>
<td>L2</td>
<td>N1</td>
<td>N1</td>
<td>N12</td>
<td>0.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>X</td>
<td>54</td>
<td>0.00</td>
<td>-</td>
<td>L3</td>
<td>N12</td>
<td>N12</td>
<td>N14</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In addition to the attributes that were included as a result of joining the two PMU tables, two new attributes, the LRS ID and the LRS MP fields, were also added. The LRS ID field is used to assign the unique LRS identifier (number) to each link. Links that contain the same LRS ID comprise an LRS route. The LRS MP is the milepost marker for the LRS route. Each link is assigned an LRS MP, which is the summed link lengths within the LRS route up until that point. The reader is referred to Section 3.5 for a more detailed description of LRS ID and LRS MP.
The goal of the algorithms is to use the data from the LRSCHAINS table to identify LRS routes, assign LRS IDs, and to accumulate the mileage (LRS MP) from the information provided in the table. The end result, then, is a complete LRS system that is integrated with the current link node system. At a later time, the link node system can be abandoned and dropped. Thus, the algorithms do two things: (1) they determine LRS routes and (2) they provide an automated conversion from the PMU link node system to the LRS system.
7.0 Primary Algorithms for Generating LRS IDs

The algorithms described below have been broken into primary posted route algorithms and secondary posted route algorithms. Separate algorithms for primary and secondary posted routes have been developed due to the distinct contrasts between them that were described in Section 6.0. The primary posted route algorithms are described in this section and the secondary posted route algorithms are described in the Section 8.0.

Each algorithm uses a different subset (mix) of the constraints and/or order of application of the constraints. Table 7.1, Primary Route Algorithm Constraints, provides a list of the major constraints along with the primary posted route test algorithms. An “X” in a cell indicates the use of the specific constraint (row) in the design and implementation of that particular algorithm (column). The reader is referred to Section 6.0 for a further explanation of these constraints.

### Table 7.1: Primary Algorithm Constraints

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Manual (7.2)</th>
<th>Matching Posted Route (7.3)</th>
<th>Longest Matching Posted Route (7.4)</th>
<th>Long Route (7.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedence Rule</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Longest Route</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Follow Single Posted Route</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cross County Boundary</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Continue on New Posted Route</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

#### 7.1 Posted Route Chain Traversal

Consider Figure 7.1, Primary Posted Route Chain Traversal, for example. In this figure four a sequence of examples are presented that illustrate the values of the link, next link, and previous link fields in the table as the selected “link” moves down the chain.

The first example illustrates the case where the first link in the chain is arbitrarily selected. If this is the case, there will be no previous link to this one (previous link = null). In this case the algorithm follows the chain to its end and is complete.

The second example shows the situation at an intermediate link in the chain. As one can see, this link has both a previous and a next link since it is in the middle of the chain. In this case the algorithm follows the chain to its end, returns to the starting link, and follows the chain in the reverse direction back to its beginning.

The third and fourth examples illustrate a county boundary. If the link lies just before a county boundary it has no next link (next link = null). Likewise, if it falls after a county boundary it has no previous link (previous link = null).
As the reader knows, it is not factually correct in the third example to say that there is no next link. That is simply how the Pavement Management Unit database treats a chain when it encounters a county boundary. But, as the figure shows, the chain continues across the boundary. Thus, in the algorithm, the chain must be followed across this boundary if the continuance of the chain is sought.

**Figure 7.1: Primary Posted Route Chain Traversal**

7.2 Manual Algorithm

The manual algorithm simply involves choosing the LRS routes by hand. This means that, an analyst looks at a map, chooses the LRS routes, then designates the LRS IDs. Computer generation techniques are not used. Although this algorithm is not a feasible approach for large statewide systems, it can be used for smaller systems and, hence, should be mentioned as an option.
7.2.1 Manual Algorithm Constraint

The constraints for the Manual Algorithm can change depending on the desired outcome. The user may choose to select LRS routes based on length, based on matching the posted route numbers, or based on some other arbitrary assignment. Since a computer program is not used to generate the LRS routes, the enforcement and outline of constraints is not as rigid as it would be with computer LRS route generation. The following subsections present several options for generating LRS IDs; any of these options can be adopted into the Manual Algorithm by following the predefined constraint. The difference lies in the way the LRS routes are generated, which is manually for this algorithm.

There is some debate about the feasibility of the manual generation of LRS IDs for primary roads. Clearly, it is feasible for the two county test case used in this study. I would also suggest that it is feasible for a statewide LRS ID generation for primary roads.

This could be done by using hard copy maps, marking them with color markers, and noting the assigned LRS ID directly on the map(s). Alternatively, a more automated approach would be to use the GIS Unit’s primary road coverage and work with it manually. A series of mouse clicks would enable the analyst to select a collection of roadway segments to be included in each LRS route. Once selected, another menu operation would enable the assignment of an LRS ID to that route, color it on the monitor so that the analyst could clearly see that it was already assigned, and write the LRS ID into the PMU database as the LRS ID for every link in that route. This last step is possible because the GIS/DBMS linkage has already been established between the GIS linework and the PMU database.

Note that this is still referred to as a manual approach because all of the decision making regarding the LRS route selection is still entirely in the hands of the analyst. Yet, with the use of a GIS, and carefully following the guidelines and constraints set forth herein, this approach shows significant promise and could be seriously considered.

If one adopts a manual approach it is true that the LRS routes are not generated using the PMU database, but the chosen routes can easily be assigned to the PMU database. As a result, all parties’ objectives are met. The reason they are met so easily is because of the seamless integration between the GIS linework and the PMU and Roadway inventory databases. This results from an incorporation of the link node referencing system directly into the GIS. Ultimately, this incorporation can and will be abandoned once the LRS is created, but until that time the current seamless integration is extremely useful.

7.3 Matching Posted Route Algorithm

The Matching Posted Route Algorithm closely resembles the roadway network as seen on maps and other printed materials. That is, the LRS routes are chosen to “match” the posted routes. The only difference lies in the order in which the LRS routes are assigned. The following subsections explain the different constraint used to generate the LRS routes.
7.3.1 Matching Posted Route Algorithm Constraint

The Matching Posted Route algorithm chooses routes based on the posted route numbering. With this method LRS IDs are assigned in a sequential manner to posted routes in the order of lowest to highest posted route number. Therefore, all links that make up I-40 are assigned LRS IDs before I-95, etc. The matching algorithm does NOT take the length of the route into consideration before assigning LRS IDs. The classification precedence, however, still stands.

Figure 7.2 illustrates an example of the resulting route configuration created by using the constraints described above and the existing link node system. As the figure shows, I-40 is assigned an LRS route before I-85. Note that the only applied constrains were matching the posted route and ascending order. Length was not considered in any way.

7.4 Longest Posted Route Algorithm

The Longest Posted Route Algorithm also generates LRS IDs based on the Posted Route Numbers. Therefore, similar the to Matching Posted Route Algorithm discussed in Section 7.3, the Longest Posted Route LRS system would closely resemble the roadway network as labeled on maps and other published materials with the longest posted route LRS IDs generated first.
7.4.1 Posted Route Algorithm – Longest Route Constraint

The Posted Route Algorithm can have different constraints depending on the method that one chooses to implement. The only strict constraints defined by the Posted Route Algorithm are that it must follow the posted routes and it must assign the longest posted route the current LRS ID. This is clearly different from assigning the LRS ID to the one whose posted route number is numerically the lowest. Thus, an entirely different processing order occurs for the records in the database table. The following paragraphs provide a brief description of the Longest Posted Route Algorithm constraints.

The Posted Route Algorithm generates all possible route traversals (while maintaining adherence to posted route numbers) and calculates the total length of each. Once the longest posted route is found, an LRS ID is assigned to all links within this route.

Figure 7.3: LRS Configuration for Longest Posted Route Algorithm

Figure 7.3 illustrates an example of the resulting route configuration created by using the constraints described above (for the Longest Posted Route algorithm) and the existing link node system. As the figure shows, I-85 is assigned an LRS route before I-40 because it results in a longer LRS route. Note the difference between Figure 7.3 and the Figure 7.2.

7.5 Long Route Algorithm

The Long Route Algorithm chooses LRS IDs based on something referred to as a “long” route. The Long Route algorithm’s main objective is to produce “long” LRS routes while
adhering to the posted route paths. However, a single LRS may be made up of more than one posted route. Therefore, the LRS routes will closely resemble the roadway posted routes, but they will not match the roadway posted routes.

### 7.5.1 Long Route Algorithm Constraint

The goal of this algorithm is to generate long LRS routes. This algorithm does enforce adherence to Posted Route numbering, but the LRS route is allowed to “pick up” another posted route once the end of the current posted route is encountered. The LRS route then follows the new posted route to its end.

This algorithm enforces the precedence rule. Therefore, all LRS routes reside within their own posted route category. For example, an LRS route cannot consist of an interstate and US route (all interstate or all US or all NC is allowable). It also mandates that LRS routes maintain certain directionality, so that an LRS route should not circle back on its self or form a loop.

Figure 7.4 illustrates an example of the resulting route configuration created by using the constraints described above (for the Long Route algorithm) and the existing link node system. As the figure shows, LRS 1 starts by selecting I-40 and then continues on I-85 once the end of the posted route chain (I-40) is encountered. The reader should note that Figure 7.4 has been changed to include I-70 so that this algorithm can be effectively illustrated. When the algorithm encounters a choice between I-70 and I-85 it currently is programmed to arbitrarily choose. Alternatively, it could have been programmed to use specific selection criteria in making this choice. It could have selected the route with the next longest link, the route with the lowest posted route number, or some other criteria.

**Figure 7.4: Resulting LRS Configuration for Long Route Algorithm**
7.6 Longest Route Algorithm

Throughout this study we imply that one goal is to obtain the longest LRS routes possible. Yet there is no longest route algorithm. This is because we believe the difference between the long route and longest route algorithm results would not be substantial. The reason for this, as previously stated, is that primary posted routes are all already largely as long as they possibly can be. That is, they tend to all run from state border to state border. As a result, because of the anticipated complexity of the Longest Route Algorithm, because its results are not expected to be significantly different, and because (very importantly) it would deviate from following posted routes, this algorithm was not developed.
8.0 Secondary Algorithms for Generating LRS IDs

As mentioned earlier, LRS routes for primary roads run from state boundary to state boundary. However, for secondary posted routes, names may change once the county line is crossed. Therefore, the Secondary Posted Route Algorithm has two options: (1) cross the county line and continue assigning the LRS ID based on the new secondary route name or (2) stop at the county line and start assigning a new LRS ID to all links until another county line is encountered. The algorithms defined below explore both options.

Secondary roads are also unique (from primary) in the way that posted route names and milepost markers are assigned. Secondary roads are arbitrarily named, meaning that even and odd posted route numbers give no indication of direction (north, south, east, or west). Additionally, posted route milepost markers do not necessarily increase from west to east and south to north, as is the case with primary roads. In fact, there is absolutely no pattern to follow when using the secondary posted route numbers or milepost. Therefore, this makes it virtually impossible to create an algorithm (that follows the posted route numbering) to move in a specified direction. As a result, when assigning long LRS routes, the possibility that LRS routes will create spirals or circular patterns may exist.

How does this affect the results of a particular algorithm? Figure 8.1 illustrates what is referred to as the “fork” problem. This situation arises when a posted route number reaches its end and the algorithm must make a decision to select the link to its left or to its right. Ideally, we would like it to choose the link (and follow the posted route number) that enables us to move from west to east and south to north and that provides the longest possible LRS route. However, in this situation there is no way to logically choose the next link (left or right) with respect to directionality.

What happens is that the algorithm first checks to see if one of the links is assigned the *same* posted route number. If it is, the LRS route continues along that link because it is the same posted route (and road) as the current posted route path. However, if no links exist with the *same* posted route number, then the algorithm arbitrarily chooses a link and follows that series of links to the end of the new posted route. The process is repeated until a dead end or state boundary is encountered.

Although this approach most likely will not result in the *longest* routes, the rules and quality measures mandate that looping, spiraling, and meandering be avoided if possible. *Longest* routes would surely produce routes that loop, spiral, or meander. Therefore, *long* really makes more sense for secondary routes compared to *longest.*
Figure 8.1: The Fork Problem

Each algorithm uses a different subset (mix) of the constraints introduced in Section 6.0 and/or the order of application of the constraints. Table 8.1 identifies the constraints embedded in each algorithm. An “X” in a cell indicates the use of the specific constraint (row) in the particular algorithm (column) under consideration. The N.A. in the first row of the table stands for not applicable. Since all secondary roads are within the same classification, the precedence rule is not needed. The reader is referred to Section 6.0 for further explanation of each constraint.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Manual (8.2)</th>
<th>Matching County Posted Route (8.3)</th>
<th>Matching State Posted Route (8.4)</th>
<th>Long County Route (8.5)</th>
<th>Long State Route (8.6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedence Rule</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>Longest Route</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow Single Posted Route</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cross County Boundary</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Continue on New Route</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

8.1 Secondary Posted Route Chain Traversal

The posted route chain traversal procedure for the secondary Matching Posted Route Algorithm is very similar to the Primary Posted Route Chain Traversal discussed in Section 7.1. However, there are several differences worth mentioning. Secondary posted route names often change not only when crossing county boundaries, but also at roadway intersections.
Figure 8.2 illustrates how the secondary posted route data is stored when the posted route name changes at a roadway intersection. In addition to the posted route name change at roadway intersections, posted route names change when crossing county boundaries. However, this differentiation does not change the way the data is stored in comparison to primary posted routes; the Next_Link field is still null when crossing county boundaries.

![Figure 8.2: Secondary Posted Route Chain Traversal](image)

8.2 Manual Algorithm

As before, the manual algorithm simply involves choosing the LRS routes by hand. This can mean a fully manual approach or an approach automated through the use of a GIS. However, we did not use a manual approach for LRS route generation. Our opinion is that this manual approach is not feasible for large statewide systems such as NC DOTs. It may be used for smaller systems and, hence, should be mentioned as an option, but it is not recommended for consideration by NC DOT.

8.2.1 Manual Algorithm Constraint

Again, the constraints for the Manual Algorithm could change depending on the desired outcome. The user could choose to select LRS routes based on length, based on matching the posted route numbers, or based on some other arbitrary assignment. Since a computer program is not used to generate the LRS routes, the enforcement and adoption of certain constraints is not as rigid as it would be with computer LRS route generation. The following subsections present several options for generating LRS Ids. Any of these
options could be incorporated into the Manual Algorithm by following the predefined constraint. The difference between most of these algorithms lies in the approach they take to LRS route generation, which is manually for this algorithm.

The key advantage of a manual approach is that a person can distinguish the fork problem described above in Section 8.0. In an automated approach there is no way to provide the computer with any guidance for secondary roads, therefore, any choice is completely arbitrary. An analyst, however, can move west to east and south to north by visual inspection. An analyst can make logical choices that result in good LRS routes being identified. In fact, the results should be excellent. However, the problem size is intractable; it is simply too big.

8.3 Matching County Posted Route Algorithm – Stopping at County Boundary and New Route

The Matching Posted Route Algorithm follows the posted route and assigns LRS routes by following only the posted route chain. It ends when the posted route ends. Therefore, the Matching Posted Route Algorithm closely resembles the roadway network as seen on maps and other printed materials. It stops when a county boundary is encountered and when the posted route ends, wherever that may be. This algorithm results in the shortest LRS routes and matches the existing posted route system as closely as any algorithm discussed so far. The following subsections explain the different constraint used to generate the LRS routes using this approach.

8.3.1 Matching County Posted Route Algorithm Constraint

The Matching Posted Route algorithm (stopping at county boundaries) arbitrarily chooses routes based on posted route numbering. That is, LRS IDs are assigned in a sequential manner based on an ascending ordering of the posted route number. Therefore, all links that make up SR1340 are assigned LRS IDs before SR1495. However, the Matching Posted Route algorithm assigns all secondary routes within each county an LRS ID before moving onto the next county. This means that all secondary routes within Pender County are assigned an LRS ID before any secondary routes are assigned in New Hanover, or vice versa. The Matching Posted Route algorithm does NOT take the length of the route into consideration before assigning LRS IDs, although it could. We feel that this would not make much difference because nearly all secondary roads are short anyway, they are unique to pavement (no duplicate names for any links or portions of pavement), and they do not overlap. As a result, any processing order for a matching algorithm would result in nearly the same routes being assigned anyway.

As mentioned previously, secondary posted route names often change when crossing county boundaries. This algorithm stops at county boundaries, regardless of whether the name actually changes.

Figure 8.3 illustrates an example of the resulting route configuration created by using the constraints described above (for the Matching County Posted Route algorithm) and the
existing link node system. In this figure the algorithm is processing the county to the left of the County Boundary (C.B.) line first. In doing so, it selects SR 1215 as the lowest numbered posted route and assigns LRS 1 to this route. But note that it does not cross the county boundary to SR 1110. The second lowest posted route is SR 1310, which is assigned LRS 2. Note that it too stops at the county boundary and does not cross onto SR 1310 in the county to the right. Finally, after LRS 3 is assigned to the next sequential SR 1572, the algorithm moves into the right county and assigns the LRS IDs in a like manner.

**Figure 8.3: Configuration for Matching County Posted Route Algorithm**

---

**8.4 Matching State Posted Route Algorithm – No Stopping at County Boundary**

Similar to the Matching Posted Route Algorithm described in section 8.4, this algorithm follows the posted route and assigns LRS routes by following only the posted route chain. Therefore, the Matching Posted Route Algorithm closely resembles the roadway network as seen on maps and other printed materials. However, this algorithm does not stop when a county boundary is encountered, regardless of whether the posted route name changes. The following subsections explain the different constraints used to generate the LRS routes for this algorithm.

**8.4.1 Matching State Posted Route Algorithm Constraint**

Aside from crossing county boundaries, the Matching Posted Route algorithm (no stop at county boundaries) shares the same constraints as the Matching Posted Route algorithm
(stop at county boundaries). Although this algorithm follows the posted route numbering while within the county, when the route crosses the county boundary, the posted route number is allowed to change (only if the new route shares the same county boundary node) and the new posted route is followed to its end as a continuation of the original route.

Figure 8.4 illustrates an example of the resulting route configuration created by using the constraints described above (for the Matching State Posted Route algorithm) and the existing link node system. In this figure, the county on the left of the county boundary demarcation line was again processed first. Since SR 1215 is numerically lower than SR 1310 it was assigned LRS 1. Note that the algorithm then carried that numbering across the county boundary and onto SR 1110 stopping at the end of the route. The second lowest posted route, SR 1210, was then selected and assigned LRS 2. Unlike the county algorithm, however, this state algorithm carries LRS 2 across the county boundary. At that point SR 1310 in the right county was assigned LRS 2.

**Figure 8.4: Configuration for Matching State Posted Route Algorithm**

8.5 Long County Route Algorithm – Stop Only at County Boundary

The Long County Route Algorithm chooses LRS IDs by following a posted route number to its end. Once the end of a route has been reached, and LRS IDs have been assigned to all links in that posted route chain, the algorithm selects a new route that continues on from the end of the current route. This route is followed it to its end (assigning the same
LRS ID to all links along the way). This process is repeated until a particular route “dead ends” or until a route reaches a county boundary. By not stopping at the end of a route and by continuing along another route’s chain, the resulting LRS routes become longer than when using a purely matching approach. This algorithm still matches a route, but it can extend out beyond the boundaries of a starting route encompassing others posted routes into it.

In the Matching Posted Route algorithms, one LRS route is exactly the same as one posted route. Using this Long Route algorithm each LRS route may encompass one or more posted routes. The overall benefit of this algorithm is longer and fewer LRS routes.

8.5.1 Long County Route Algorithm Constraint

The goal of this algorithm is to generate long LRS routes within a county. This algorithm does enforce adherence to posted route numbering, but it is allowed to “pick up” a new posted route and continue along its chain once the end of a posted route chain has been reached. Since this algorithm will encounter the “fork” problem (described in Section 8.0), the algorithm makes a choice to arbitrarily select any of the connecting links (at the intersection) and continue along the new posted route’s chain. (Note that instead of selecting a connecting link arbitrarily, we may have chosen either the longest link or the one with the lowest posted route number). This means that an LRS route may circle back on itself, form loops, or meander wildly. This problem can only be detected and corrected by visually displaying the LRS route and manually correcting such instances. Finally, this algorithm forces the termination of LRS routes upon encountering the county boundary or encountering a “dead end” within the county. Thus, the resulting LRS routes are contained within a county and do not cross county boundaries.

Figure 8.5 illustrates an example of the resulting route configuration created by using the constraints described above (for the Long County Route algorithm) and the existing link node system. As the figure shows, the algorithm begins processing in the county to the left of the county boundary (C.B.) line. It selects SR1215 first, as this posted route has the lowest numerical number, and assigns is LRS 1. Note that it does not cross the county boundary onto SR 1110. But it is permitted to be long, so it extends onto SR 1310 where it arbitrarily chooses the left link and continues on to the end. The secondary routes SR 1310 and SR 1572 are then assigned LRSs 2 and 3, respectively. Finally, the algorithm processes the roads in the county on the right in a like manner.
8.6 Long State Route Algorithm – No Stopping at County Boundary

The Long State Route Algorithm chooses LRS IDs by following a posted route number to its end. Once the end of a route has been reached and LRS IDs have been assigned to all links in that posted route chain, the algorithm selects a new route and follows it to its end (assigning the same LRS ID to all links along the way). This process is repeated until a particular route “dead ends” or until a route reaches a state boundary.

By not stopping at the end of a route and by continuing along another route’s chain, the resulting LRS routes become longer than when using a purely matching approach. This algorithm still matches a route, but it can extend out beyond the boundaries of a starting route encompassing others posted routes into it.

In the Matching Posted Route algorithms, one LRS route is exactly the same as one posted route. Using this Long Route algorithm each LRS route may encompass one or more posted routes. The overall benefit of this algorithm is longer and fewer LRS routes. This algorithm differs from Long County Routes Algorithm in one respect only – it stops at state boundaries instead of at county boundaries.
8.6.1 Long State Route Algorithm Constraint – No Stopping at County Boundary

The goal of this algorithm is to generate long LRS routes across the entire state. This algorithm does enforce adherence to posted route numbering, but it is allowed to “pick up” a new posted route and continue along its chain once the end of a posted route chain has been reached. Since this algorithm will encounter the “fork” problem (described in Section 8.0), the algorithm makes a choice to arbitrarily select any of the connecting links (at the intersection) and continue along the new posted route’s chain. Thus, an LRS route may circle back on itself, form loops, or meander. This problem can only be detected and corrected by visually displaying the LRS route and manually correcting such instances. Finally, this algorithm forces the termination of LRS routes upon encountering the state boundary or encountering a “dead end” within the state. Thus, the resulting LRS routes are allowed to stretch from one end of the state to another.

Figure 8.6 illustrates an example of the resulting route configuration created by using the constraints described above (for the Long State Route algorithm) and the existing link node system. As the figure shows, the algorithm began processing in the county to the left of the county boundary (C.B.) line, selecting SR 1215 as LRS 1. It then continues across the county boundary to include SR 1110 in LRS 1. In the other direction, it discovered a new route, SR 1310 and arbitrarily selected the left link. After completing LRS 1, it selected SR 1310 as the next lowest numbered link and assigned it LRS 2, crossing the county boundary in the process to extend the route. Finally, LRS 3 is assigned to SR 1572.

Figure 8.6: Configuration for Long State Route Algorithm
8.7 Longest State Route Algorithm

Again, as it did for primary roads, the question arises of why a Longest Route algorithm for secondary roads was not developed. The answer is because we know that such an algorithm would generate infeasible routes. To generate the longest route involves generating all possible routes, calculating the length for each, selecting the longest and assigning its links the correct LRS ID, incrementing the LRS ID counter, and repeating the process.

First of all, because these are about 195,000 link records in the statewide database, the processing time it would take to generate every possible LRS route for one single pass through the loop would be prohibitive. Even worse, the longest theoretical LRS route (if the algorithm generates all possible routes as it should) would always involve nearly all the links and would loop and meander all across the state. Thus, the longest route would always be guaranteed to be an undesirable outcome.

If we had knowledge of directionality we could largely avoid loops and meandering, but because the PMU database does not provide a way to determine directionality for secondary roads, this problem cannot be overcome. Therefore a Longest Route algorithm could not and was not developed.
9.0 Algorithm Comparison

The previous section explored several options for producing the LRS routes. Each method was bound by its own set of constraints. These constraints greatly impact the outcome of the LRS routes. The following subsections will compare the resulting LRS routes based on a set of measures that we selected as having a bearing on the quality of the algorithm. In Section 9.1 we simply list and define all of the measures used. In Section 9.2 we report on the values of each measure for each algorithm. Finally, in Section 9.3 we interpret the results and come to some overall determination of how the measures enable us to decide which algorithm best meets our needs.

9.1 Quality Measures for LRS Routes

The following subsections provide a description of all quality measure and how values were obtained for each of them. Most of the quality measures were obtained by querying and manipulating the LRS tables for each algorithm.

9.1.1 Length of LRS Routes

The length of the routes is a potential measure of differences between the algorithms. For each algorithm the length of each LRS route was calculated by summing the length of each link within the route. Once the length of each route was calculated the longest, shortest, average, and median of each LRS route was calculated. Finally, we determined the average length for the shortest 10% of all routes, the next shortest 10% of all routes, etc. These averages tell us how short the short routes were, etc.

9.1.2 Number of LRS Routes

The number of LRS routes was determined by identifying the minimum and maximum LRS IDs for each algorithm, subtracting the two, and adding 1. The total number of LRS Routes is important because it determines a maximum number of records that define all of the names (LRS IDs) used in the database. Since LRS ID is a key field, the fewer LRS IDs the faster the processing. In addition to providing the total number of LRS routes for each algorithm, the number of longest routes, long routes, short routes, and shortest routes is also provided. Section 9.2 will provide a more detailed explanation of these terms.

9.1.3 Processing Time

Processing time can be interpreted two ways. There is the processing time that it takes to run an algorithm and create an LRS route configuration. There is also the processing time it takes to use an LRS route configuration to run a query and retrieve data. This latter measure is especially important when running queries and retrieving data from the entire state file.
9.1.4 Other Measures

There are two more quality measures that are taken into account: ease of algorithm development and visual inspection. The ease of algorithm development is completely subjective in nature and refers to the difficulties (or obstacles encountered) when creating the different algorithms.

In a general sense, we seek LRS routes whose orientation is largely east to west and north to south. Visual inspection is a good way to verify whether or not we have achieved this in our configurations. That is, when looking at and studying a picture one can “see” if the components of the picture exhibit the desired characteristics (constraints).

Visual inspection is a visual analysis of an algorithm’s performance through the inspection of GIS generated maps that display the LRS Routes resulting from each algorithm. Appendix B contains such maps and includes one map for each algorithm (total of seven) that display the routes and one map for each of the secondary algorithms (total of four) that displays how long each such secondary route is.

9.2 Quality Measure Results

The following subsections provide results for the quality measures described above. Each quality measure defines a distinct characteristic of the results of each algorithm. In addition to statistical information, graphs and charts are provided along with a detailed description of what each means. Based on the results provided in these sub-sections, an evaluation of each algorithm is formulated and, ultimately, a recommendation is made.

9.2.1 Length of LRS Routes

The following tables and graphs illustrate, for each algorithm, the values of various measures related to the length of LRS routes. These values show the differences between each algorithm and provide a means to evaluate the results. Included in this section are two tables that provide the length of the longest LRS route, the length of the shortest LRS route, the average length of all LRS routes, and the median length of all LRS routes. Following the two tables are a graph and a bar chart for each algorithm that display additional length data for the routes.

In Table 9.1 and Table 9.2 the column headings represent the algorithm names. The rows represent the statistical measures that were calculated from the resulting algorithms. The longest and shortest represent the absolute longest and shortest LRS route. The average totals all LRS lengths and divides by the total number of LRS routes, which results in the average length for each algorithm. The median is the absolute middle, meaning that an equal number of LRS routes have a length greater than this value and an equal number have a length less than this value.
Table 9.1: Length of LRS Routes for Primary Algorithms

<table>
<thead>
<tr>
<th>Length of LRS Route Measures</th>
<th>Manual</th>
<th>Matching Posted Route</th>
<th>Longest Posted Route</th>
<th>Long Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longest</td>
<td>N.A.</td>
<td>35.84</td>
<td>55.34</td>
<td>37.65</td>
</tr>
<tr>
<td>Shortest</td>
<td>N.A.</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Average</td>
<td>N.A.</td>
<td>13.51</td>
<td>15.27</td>
<td>15.97</td>
</tr>
<tr>
<td>Median</td>
<td>N.A.</td>
<td>7.66</td>
<td>4.90</td>
<td>12.56</td>
</tr>
</tbody>
</table>

Table 9.2: Length of LRS Routes for Secondary Algorithms

| Length of LRS Route Measures | Manual | Matching County Posted Route | Matching State Posted Route | Long County Route | Long State Route |
|-----------------------------|--------|-------------------------------|----------------------------|-------------------|----------------|---|
| Longest                     | N.A.   | 19.7                          | 19.7                       | 21.65             | 21.65          |
| Shortest                    | N.A.   | 0.01                          | 0.01                       | 0.01              | 0.01           |
| Average                     | N.A.   | 0.643                         | 0.64                       | 0.75              | 0.75           |
| Median                      | N.A.   | 0.25                          | 0.24                       | 0.27              | 0.26           |

In Figures 9.1 through 9.7 the vertical axis of each bar chart or graph represents the length of an LRS Route. The horizontal axis is more complicated. For the graph, we are plotting a length value for each and every LRS Route. The horizontal axis, then, extends from zero to the largest number of LRS Routes for that algorithm and the result appears to be a continuous string of data points that approximate a curve. To better illustrate the lengths and the number of routes that fall within various length groupings, a logarithmic scale was used to plot the points.

For the bar chart, the approach is different. First, we sort the data by length from the shortest to the longest LRS route. Then the total number of LRS routes is divided into tenths. For example, if there is a total of 100 LRS routes, then each bar represents the average length of 10 LRS routes. Therefore, one bar is plotted for each one-tenth of the total number of LRS Routes. The vertical axis represents the average length of each bar (one-tenth of the total number of routes).
**Figure 9.1A: Primary – Matching Posted Route Algorithm Lengths**

![Graph showing the length of LRS routes against the number of LRS routes.](image)

**Figure 9.1B: Primary – Matching Posted Route Algorithm Average Lengths**

![Bar chart showing the average length of LRS routes against the 1/10th number of LRS routes.](image)
Figure 9.2A: Primary – Longest Posted Route Algorithm Lengths

Length of LRS Routes

Number of LRS Routes

Figure 9.2B: Primary – Longest Posted Route Algorithm Average Lengths

Average Length of LRS Routes

1/10th Number of LRS Routes

Each 1/10th = 2 Routes
Figure 9.3A: Primary – Long Route Algorithm Lengths

Length of LRS Routes

Number of LRS Routes

LRS Length in Miles

Figure 9.3B: Primary – Long Route Algorithm Average Lengths

Average Length of LRS Routes

1/10th Number of LRS Routes

Each 1/10th = 2 Routes

Average Length in Miles
Figure 9.4A: Secondary – Matching County Posted Route Algorithm Lengths

Length of LRS Routes

Number of LRS Routes

Figure 9.4B: Secondary – Matching County Posted Route Algorithm Average Lengths

Average Length of LRS Routes

Each 1/10th = 138 Routes

1/10th Number of LRS Routes
Figure 9.5A: Secondary – Matching State Posted Route Algorithm Lengths

Length of LRS Routes

![Graph showing the relationship between the number of LRS Routes and their length in miles.](image)

Figure 9.5B: Secondary Matching State Posted Route Algorithm Average Lengths

Average Length of LRS Routes

![Graph showing the average length of LRS Routes for each 1/10th number of routes.](image)

Each 1/10th = 138 Routes
Figure 9.6A: Secondary – Long County Route Algorithm Lengths

Length of LRS Routes

Figure 9.6B: Secondary – Long County Route Algorithm Average Lengths

Average Length of LRS Routes

Each 1/10th = 118 Routes
Figure 9.7A: Secondary – Long State Route Algorithm Lengths

Length of LRS Routes

Number of LRS Routes

Figure 9.7B: Secondary – Long State Route Algorithm Average Lengths

Average Length of LRS Routes

Each 1/10th = 118 Routes

1/10th Number of LRS Routes
9.2.2 Number of LRS Routes

The following tables and graphs illustrate, for each algorithm, the values of various measures related to the number of LRS routes. These values show the differences between each algorithm and provide a means to evaluate the results.

In Tables 9.3 and 9.4 the column headings represent the algorithm names. The rows are separated into a measures section and a length cutoffs section. The measures identify what is being measured. They are broken into the number of longest, long, short, and shortest routes. These numbers specify the number of routes that fall between 100 and 75 percent (longest), 75 and 50 percent (long), 50 and 25 percent (short), and, finally, 25 and 0 percent (shortest) of the maximum length of the longest LRS route for each algorithm. The Length Cutoffs represent the actual lengths that define each category (100%, 75%, 50%, and 25%) for each individual algorithm.

This length measure (Tables 9.3 and 9.4) is a relative length measure and it is useful only for understanding the distribution of LRS route lengths within any given algorithm. It is not a useful measure for comparing algorithms to each other.

Table 9.3: Number of LRS Routes for Primary Algorithms

<table>
<thead>
<tr>
<th>Number of LRS Route Measures</th>
<th>Manual</th>
<th>Matching Posted Route</th>
<th>Longest Posted Route</th>
<th>Long Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of LRS Routes</td>
<td>N.A.</td>
<td>26</td>
<td>23</td>
<td>22</td>
</tr>
<tr>
<td>Number of Longest Routes</td>
<td>N.A.</td>
<td>7 (27%)</td>
<td>1 (4%)</td>
<td>6 (27%)</td>
</tr>
<tr>
<td>Number of Long Routes</td>
<td>N.A.</td>
<td>2 (8%)</td>
<td>7 (31%)</td>
<td>3 (14%)</td>
</tr>
<tr>
<td>Number of Short Routes</td>
<td>N.A.</td>
<td>3 (11%)</td>
<td>1 (4%)</td>
<td>3 (14%)</td>
</tr>
<tr>
<td>Number of Shortest Routes</td>
<td>N.A.</td>
<td>14 (54%)</td>
<td>14 (61%)</td>
<td>10 (45%)</td>
</tr>
</tbody>
</table>

Length Cutoffs

| 100 – 75 % (Longest) | N.A. | 35.84 – 26.88 | 55.34 – 41.51 | 37.65 – 28.24 |
| 75 – 50 % (Long)     | N.A. | 26.88 – 17.92 | 41.51 – 27.67 | 28.24 – 18.83 |
| 50 – 25 % (Short)    | N.A. | 17.92 – 8.96  | 27.67 – 13.84 | 18.83 – 9.41 |
| 25 – 0 % (Shortest)  | N.A. | 8.96 – 0.00   | 13.84 – 0.00  | 9.41 – 0    |
Table 9.4: Number of LRS Routes for Secondary Algorithms

<table>
<thead>
<tr>
<th>Number of LRS Route Measures</th>
<th>Manual</th>
<th>Matching County Posted Route</th>
<th>Matching State Posted Route</th>
<th>Long County Route</th>
<th>Long State Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Routes</td>
<td>N.A.</td>
<td>1380</td>
<td>1379</td>
<td>1186</td>
<td>1184</td>
</tr>
<tr>
<td>Number of Longest Routes</td>
<td>N.A.</td>
<td>1 (0.1%)</td>
<td>1 (0.1%)</td>
<td>5 (0.4%)</td>
<td>4 (0.3%)</td>
</tr>
<tr>
<td>Number of Long Routes</td>
<td>N.A.</td>
<td>5 (0.4%)</td>
<td>5 (0.4%)</td>
<td>5 (0.4%)</td>
<td>5 (0.4%)</td>
</tr>
<tr>
<td>Number of Short Routes</td>
<td>N.A.</td>
<td>30 (2.2%)</td>
<td>31 (2.2%)</td>
<td>16 (1.4%)</td>
<td>19 (1.6%)</td>
</tr>
<tr>
<td>Number of Shortest Routes</td>
<td>N.A.</td>
<td>1344 (97.3%)</td>
<td>1342 (97.3%)</td>
<td>1160 (97.8%)</td>
<td>1156 (97.6%)</td>
</tr>
</tbody>
</table>

Length Cutoffs

<table>
<thead>
<tr>
<th></th>
<th>Manual</th>
<th>Longest Posted Route</th>
<th>Longest Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 – 75 % (Longest)</td>
<td>N.A.</td>
<td>19.7 – 14.78</td>
<td>21.65 – 16.24</td>
</tr>
<tr>
<td>50 – 25 % (Short)</td>
<td>N.A.</td>
<td>9.85 – 4.93</td>
<td>10.83 – 5.41</td>
</tr>
<tr>
<td>25 – 0 % (Shortest)</td>
<td>N.A.</td>
<td>4.93 – 0.00</td>
<td>5.41 – 0.00</td>
</tr>
</tbody>
</table>

Tables 9.5 and 9.6 identify the total number of routes within a certain fixed length range. For the primary route algorithms, the length ranges are broken into four equal length ranges. For secondary routes, the length ranges are broken into 10 equal length ranges. Although these two tables provide a means for comparison, it should be noted that the total number of LRS routes is different for each algorithm.

This length measure (Tables 9.5 and 9.6) is an {absolute} length measure and it is useful for understanding the overall length distribution of all the algorithms. These tables of results provide a measure for comparing the algorithms to each other.

Table 9.5: Number of LRS Routes for Primary Algorithms

<table>
<thead>
<tr>
<th>Number of LRS Route Measures</th>
<th>Manual</th>
<th>Matching Posted Route</th>
<th>Longest Posted Route</th>
<th>Longest Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of LRS Routes</td>
<td>N.A.</td>
<td>26</td>
<td>23</td>
<td>22</td>
</tr>
<tr>
<td>Routes &gt; 45 miles</td>
<td>N.A.</td>
<td>0 (0%)</td>
<td>1 (4%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Routes 45 – 30 miles</td>
<td>N.A.</td>
<td>5 (19%)</td>
<td>6 (26%)</td>
<td>6 (27%)</td>
</tr>
<tr>
<td>Routes 30 – 15 miles</td>
<td>N.A.</td>
<td>4 (15%)</td>
<td>1 (4%)</td>
<td>4 (18%)</td>
</tr>
<tr>
<td>Routes 15 – 0 miles</td>
<td>N.A.</td>
<td>17 (66%)</td>
<td>15 (66%)</td>
<td>12 (55%)</td>
</tr>
</tbody>
</table>
Table 9.6: Number of LRS Routes for Secondary Algorithms

<table>
<thead>
<tr>
<th>Number of LRS Route Measures</th>
<th>Manual</th>
<th>Matching County Posted Route</th>
<th>Matching State Posted Route</th>
<th>Long County Route</th>
<th>Long State Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Routes</td>
<td>N.A.</td>
<td>1380</td>
<td>1379</td>
<td>1186</td>
<td>1184</td>
</tr>
<tr>
<td>Routes &gt; 10 miles</td>
<td>N.A.</td>
<td>4 (0.3%)</td>
<td>4 (0.3%)</td>
<td>12 (1.0%)</td>
<td>12 (1.0%)</td>
</tr>
<tr>
<td>Routes 10 – 9 miles</td>
<td>N.A.</td>
<td>5 (0.4%)</td>
<td>5 (0.4%)</td>
<td>0 (0.0%)</td>
<td>0 (0.0%)</td>
</tr>
<tr>
<td>Routes 9 – 8 miles</td>
<td>N.A.</td>
<td>3 (0.2%)</td>
<td>4 (0.3%)</td>
<td>1 (0.1%)</td>
<td>4 (0.3%)</td>
</tr>
<tr>
<td>Routes 8 – 7 miles</td>
<td>N.A.</td>
<td>4 (0.3%)</td>
<td>4 (0.3%)</td>
<td>3 (0.3%)</td>
<td>4 (0.3%)</td>
</tr>
<tr>
<td>Routes 7 – 6 miles</td>
<td>N.A.</td>
<td>5 (0.4%)</td>
<td>6 (0.4%)</td>
<td>9 (0.8%)</td>
<td>6 (0.6%)</td>
</tr>
<tr>
<td>Routes 6 – 5 miles</td>
<td>N.A.</td>
<td>14 (1.0%)</td>
<td>12 (0.9%)</td>
<td>5 (0.4%)</td>
<td>4 (0.3%)</td>
</tr>
<tr>
<td>Routes 5 – 4 miles</td>
<td>N.A.</td>
<td>8 (0.6%)</td>
<td>8 (0.6%)</td>
<td>11 (0.9%)</td>
<td>10 (0.9%)</td>
</tr>
<tr>
<td>Routes 4 – 3 miles</td>
<td>N.A.</td>
<td>21 (1.5%)</td>
<td>21 (1.5%)</td>
<td>16 (1.4%)</td>
<td>14 (1.3%)</td>
</tr>
<tr>
<td>Routes 3 – 2 miles</td>
<td>N.A.</td>
<td>31 (2.2%)</td>
<td>30 (2.2%)</td>
<td>35 (3.0%)</td>
<td>36 (3.0%)</td>
</tr>
<tr>
<td>Routes 2 – 1 miles</td>
<td>N.A.</td>
<td>88 (6.4%)</td>
<td>87 (6.3%)</td>
<td>95 (8.0%)</td>
<td>95 (8.0%)</td>
</tr>
<tr>
<td>Routes 1 – 0 miles</td>
<td>N.A.</td>
<td>1197 (86.7%)</td>
<td>1198 (86.8%)</td>
<td>999 (84.1%)</td>
<td>999 (84.3%)</td>
</tr>
</tbody>
</table>

9.2.3 Processing Time

Processing time to create the LRS route configuration has been identified as being different from the processing time required to use a configuration to respond to queries. To use a configuration requires that a fully functional LRS be in place. Clearly, we do not yet have this, so processing time for use cannot and has not been tested. However, this is a key measure since, ultimately, the LRS system should improve efficiency and more rapidly retrieve data via queries etc.

Processing time to generate the LRS was not tested either. Because this will be a one-time operation (once an algorithm is selected), it is felt that this type of processing time is not critical. Therefore, unless a significant obstacle is encountered, no further investigation of the measure is warranted.


9.2.4 Other Measures

The measures described above can be observed or calculated directly from data generated from the resulting LRS route configurations by the algorithms. The following measures are more subjective in their nature. The way they are interpreted is left up to the reader. These measures include ease of algorithm development and visual inspection (see Appendix K).

9.2.4.1 Ease of Algorithm Development

For the most part, all programming and algorithms were straight-forward. Manually generating an LRS route configuration was deemed impractical on a statewide basis for secondary roads.

9.2.4.2 Visual Inspection

Visual inspections of LRS routes involve displaying each LRS Route configuration using a GIS to create map-like printouts (the reader is referred to Appendix B for the maps generated by this study). This is an essential part of the evaluation process. It allows one to look at the printouts and detect errors not only in the algorithm, but also in the database itself. As stated previously, the database file that was used in this study was not entirely cleaned. That is, it was not entirely correct. Visual inspection allows one to more easily identify these errors and mistakes, which would otherwise be detected only by laboriously inspecting the LRS system in tabular form. Thus, the route generation process can turn out to be a means to improve the integrity of the databases.

For example, consider the Primary Road LRS Routes for the Matching Posted Route Algorithm in Appendix B. LRS Route 14 stops at the New Hanover/Pender County boundary and LRS Route 16 continues into New Hanover. At first glance this appears to be a mistake since it is in fact the same posted route and, therefore, should be assigned the same LRS ID. However, the tabular data has been inspected and a mistake in the database was detected. The algorithm correctly processed the database table, but because the county boundary node for the link was given the wrong number the algorithm stopped at the county boundary. This happened because, there was no county boundary node to match with to continue the LRS route.

Referring to the same figure, LRS Route 19 appears to have “jumped” from the upper right corner down to a lower section of the county. Again, this is due to the structure of the database. Within the database, a fictitious link exists along the county boundary for any posted route that crosses into another county and then crosses back into the original county. Therefore, the algorithm used this “link” and continued assigning the LRS ID across it even though this result should not occur.

Visual inspection is especially critical for inspecting the “long” secondary route algorithms described in Sections 8.5 and 8.6. As explained previously, the LRS routes created with these algorithms may form loops or spirals or very odd shapes because the
generation of the LRS routes is completely arbitrary. The reader is referred to Section 11.0 for further discussion on this issue. In fact, through visual inspection, an early mistake in the Long Secondary Route Algorithms was detected. It was discovered that the algorithm chose both forks when encountering the “fork” problem instead of only choosing one. This problem was resolved, but visual inspection enabled us to easily detect the problem and make the required adjustments to the algorithm.

9.3 Evaluation Outcome and Analysis

The following two subsections look at the results from the previous section and provide some comparative analysis of these results. The following two sub-sections are separated into a comparison between the primary algorithms and a comparison between the secondary algorithms. From this evaluation a useful set of conclusions can be drawn and a preliminary recommendation can be made.

9.3.1 Primary Route Algorithm Outcome and Evaluation

Regarding the number of primary LRS routes, the difference between the Matching and Long Route algorithms was not significant (22 to 23). However, the Longest Posted Route Algorithm does produce fewer LRS routes than the matching Posted Route Algorithm (23 to 26). Although the number is not significantly different, one must remember that our test case was small in comparison to the entire state. Therefore, this seemingly insignificant change in number may prove to be a more substantial change if tested statewide.

In addition to a decrease in the total number of LRS routes generated by the Longest Posted Route algorithm, the average length also increased slightly. With the goal that number of LRS routes should remain low while the length should remain high, it appears that the Longest Posted Route algorithm satisfies these measures more adequately than the Matching Posted Route algorithm.
9.3.2 Secondary Route Algorithm Outcome and Evaluation

Regarding the number of secondary LRS routes, differentiation occurs between the long and the matching algorithms. Long Route Algorithms reduce the number of LRS routes compared to the Matching Posted Route Algorithms (1200 vs 1400). Therefore, if NC DOT desires fewer LRS Routes, the Long Route Algorithms would be better choices as opposed the Matching Posted Route Algorithm(s). Furthermore, the length of the longest LRS route for each algorithm increases for routes generated with the Long County/State Route Algorithm(s) as opposed to the Matching Posted Route Algorithms (22 miles vs. 20 miles). Finally, the average length of the LRS routes generated with the Long County/State Route Algorithm(s) increases when compared the average length generated with the Matching Posted Route Algorithms (0.75 miles vs. 0.65 miles).

Thus, with the goal of having long routes, the Long County/State Route Algorithms are the algorithms that would produce this result. The only concern is that “long” will not guarantee directionality (west to east and south to north) for secondary roads. The LRS routes may be spiral, circular, or any other odd shape. Correcting such a situation would require manual intervention via visual inspection and adjustment.

The difference between the algorithms crossing a county boundary versus not crossing a county was insignificant with this test case. The two test case counties did not adequately permit testing the difference between crossing a county boundary and not crossing the
county boundary. New Hanover and Pender counties share a boundary that is distinguished by a river. Therefore, very few secondary roads cross the county boundary; in fact, only two.

As noted earlier, both the manual algorithm and the longest secondary LRS algorithms were not attempted in this study. It was concluded that the longest secondary routes would surely generate routes that loop and spiral, which is not desirable. The “fork” problem (defined in section 8.0) prevents the generation of guided, criteria-based algorithm results. Thus, based on the LRS constraints and the desirable outcome, long secondary LRS routes provided a more sensible alternative to longest secondary LRS routes.

Figure 9.9: Secondary Algorithm Length vs. Number of LRS Routes Comparison

![Secondary Algorithm LRS Length Comparison](image-url)
10.0 Conclusions and Recommendations

The North Carolina Department of Transportation LRS model (Kiel 1999) suggested that long LRS routes and that keeping the number of LRS routes to a minimum is significantly important. Combining these two goals with the results of the testing conducted herein we draw a number of conclusions and make the following preliminary recommendations.

• Preliminary Algorithm Recommendations
  — Primary - Longest Posted Route
  — Secondary - Long State Route or Long County Route

• Notes
  — Manual algorithms not tested
  — Secondary State Route algorithms not effectively tested with the current data set

• More testing needed
  — Include the Manual algorithms

• Better test case
  — Four-county group
  — Internal to state
  — No river bordering the county boundary

• Results carefully reanalyzed
  — If consistent, go with recommended algorithms
  — If not consistent, a different recommendation may be in order

• Execute algorithms statewide
  — Resolve problems due to database inaccuracy

• LRS Mileposts need to be calculated
10.1 Algorithm Selection

The recommended algorithms are:

Primary – Longest Posted Route Algorithm
Secondary – Long State Route Algorithm or Long County Route Algorithm

From the test cases provided by the NC DOT (New Hanover and Pender Counties) these algorithms produce the longest routes overall and have the fewest LRS routes compared to all other algorithms tested. However, it must be noted that we still need to more comprehensively prove this. Our conclusions stand to reason, but we believe that a new test case is essential to verifying them. Secondary routes must also be further tested for circles, spirals, and meandering.

However, these recommendations are not made lightly. An examination of Table 9.1 for primary roads shows consistency with what we would have expected to find. The Matching Posted Route algorithm and Longest Posted Route algorithm are both algorithms that force the LRS routes onto posted routes. Interstate and US routes are generally statewide in length. Thus, to match them is to automatically get a long route. Additionally, a similar result occurs for US and NC routes.

10.2 Transition and Correlation

It is easiest to do things tomorrow the same way they are done today: Change requires effort. The more drastic the change the greater the effort, the higher the learning curve, and the longer the time to greater productivity.

With this in mind, it is most expeditious to examine the LRS in terms of the posted route system. The posted route system provides the highest comfort level, the greatest familiarity, and the largest implementation base in databases for the Department of Transportation. To the extent that the LRS can “somewhat” mirror the posted route system without sacrificing any of the advantages of the LRS approach, it should do so. For the recommended Longest Posted Route algorithm for primary roads this can readily be achieved.

However, it would appear that the Long State Route algorithm recommendation for secondary roads will not do so. Since this algorithm does not appear to be a “matching” algorithm the correlation seems weak. But in reality, the secondary LRS routes do match the secondary posted routes. They are even longer and, in fact, may encompass more than one secondary route in each LRS route. As a result, the transition from posted routes to LRS routes should be relatively easy and natural for those working at the level of the base referencing system. For all others, the LRS is invisible, and the underlying changes will be completely unknown to them anyway.
10.3 Additional Testing

Although the study provided some insight on which algorithms produced the best LRS routes based on the quality measures, the test counties did not produce the results necessary to adequately compare all quality measures. More testing is needed to ensure that the algorithms produce the anticipated results. It is recommended that the algorithms be tested on a four-county group, internal to the state, with no river bordering the county boundary. This would verify the county boundary tests more accurately than the New Hanover and Pender test cases. It is critical to have an adequate number of secondary roads that cross county boundaries to obtain a more accurate assessment of the algorithms and the results.

Once the algorithms have been tested using the suggested four county grouping, the results of the quality measures should be carefully analyzed. If these results are the same as the results from this study, then the best algorithm(s) should be chosen and run on the entire state. However, if the results do not produce the same outcome as this study, all algorithms should be studied and appropriate changes should be made.

10.4 Gap Roads

The Pavement Management Unit (PMU) database contains “gap” records. A gap in a road in County A occurs when the road leaves County A, enters County B, and then returns to County A.

Figure 10.1: Gap Record Example
As expected, the PMU database creates the appropriate links for the roads in both counties for the portions of the road that lie within each respective county. In addition, a *fictitious* record is inserted into the County A database. This record (L5 in the example shown in figure 10.1) enables a user of the County A database records to know which links are a part of the road in question.

The gap record creates a problem for the algorithms outlined herein. Consider the example (Figure 10.1). We wish to assign an LRS ID to the chain of links identified by L1, L2, L3, and L4. However, as the algorithm proceeds from L1 it encounters both L2 and L5 as lying on the same route and sharing the same county boundary node. How does it know to choose L2 rather than L5? The algorithms currently do not. Thus, either the gap records must be removed or the algorithms must be modified. This is because the algorithm may assign the LRS route incorrectly to L5 and assign a different LRS route to L2 and L3.

### 10.5 Errors in Database

It is recommended that all 100 counties be “cleaned” before the LRS IDs are generated. That is, timely completion of the linework/database cleanup would facilitate LRS route generation. Still, errors will remain, as will discrepancies between the database and the linework. The LRS route generation algorithms can aid in identifying these errors and discrepancies.

Recall the discussion in Section 9.2.4.2 regarding the assignment of two LRS Routes (14 and 16 with the Matching Posted Route Algorithm) to US 421. Visual inspection of the maps clearly identified the problem. An inspection of the database revealed an incorrectly named node. Thus, the LRS route generation process aided in the identification of a database discrepancy.

This is a good result. It provides an additional check and an opportunity to improve the database, the linework, or both *prior* to permanently establishing the LRS, its routes, and the databases.

It is recommended that the selected algorithm be run statewide on the database after full cleaning. At this time a complete examination of the results should be made to identify discrepancies. These should be resolved. After this *final* cleaning, the algorithm can be rerun to establish the *permanent* LRS routes.
10.6 LRS Mileposts

In Sections 3.5 and 6.2 we alluded to outlined the procedure for determining the LRS milepost values for all of the nodes in the link node database. Determining the value of the LRS mileposts if critical. With both the routes and the mileposts we have all the information needed to completely describe the topology (routes) and geometry (mileposts) of the transportation network. Although the determination of the mileposts was outside the scope of this study, doing so is essential for completing the LRS.
11.0 References


Rasdorf, W., Shuller, E., Poole, R., Abudayyeh, O., and Robson, F., “Information Management at State Highway Departments: Issues and Needs,” Journal of Transportation Engineering, American Society of Civil Engineers, Volume 126, Number 2, Pages 134-142 (March/April 2000).


