Abstract

Supervisory Control and Data Acquisition (SCADA) systems are being increasingly used to monitor and control critical infrastructures ranging from computer networks to manufacturing, and proper design of SCADA systems is an important issue for developers of SCADA systems and their users. Current practice seems to center around purchasing a best-of-breed solution and configuring it to meet the expected system goals. In this paper we propose an efficient design approach based on minimum spanning trees and the NFR Framework (NFR standing for Non-Functional Requirements) wherein SCADA designs are captured in spanning trees and minimum spanning trees indicate the optimally weighted designs, and the most suitable among these weighted designs is chosen by employing the NFR Framework, which is a goal-oriented framework for analyzing competing alternatives and choosing the best one based on the goals for the SCADA system. The design algorithm is verified by applying it to a case study.

The case study we considered was the Key Stone oil pipeline project. We propose a sample SCADA design for this pipeline based on our algorithm.
Chapter 1

Introduction

1.1 SCADA

SCADA stands for Supervisory Control and Data Acquisition. SCADA systems monitor and control from remote – for example, the state of all facilities in an airport can be monitored and controlled from a central control room. The software/hardware in the central control room are referred to as the Master of the SCADA system, while the remotely monitored and controlled components such as door sensors, door actuators, door alarms, water monitors/controllers, HVAC monitors/controllers, etc. are all referred to as the remote terminal units (RTU’s). Typically an RTU is a superset of each sensor/controller/actuator in the sense that an RTU can monitor and control a set of sensors/controllers/actuators – therefore, the air-conditioning plant at the airport may be connected to the Master by one RTU, while an airport passenger terminal may be connected to the Master by another RTU. The Master is connected to the RTU’s by means of high speed data communication links such as twisted cables, coaxial cables, optical fibers, or wirelessly.

SCADA gives the business owner a centralized viewpoint of the status of the complete system so that any vulnerabilities or bottlenecks are detected long before they become an issue. SCADA may control facilities in a small geographical area or may be dispersed over a state, country, or internationally; for example, SCADA that monitors a water-treatment plant may cover only a small area, while the SCADA that monitors the water pipelines may be distributed over a much larger geographical area. In the case where a SCADA covers huge geographical areas, the data communication links connecting the Master and the RTU’s may include cellular, satellite, and dedicated circuit technologies [1].
In several systems which employ SCADA, the control system architecture need not necessarily be the same as the system under control: for example, the nervous system of the human body compared to the circulatory system; road network versus the traffic signaling system; and air network in relationship to the air traffic control. However, there are systems wherein the physical architecture of the control system and the system being controlled could be alike: for example, the Internet routing system and the Internet, or home electrical wiring and appliances controlled using X10 protocol [8]. But in the case of oil pipeline systems, the SCADA architecture is typically different both logically and physically from the oil pipeline layout. The most important concerns in deploying SCADA for oil pipelines seem to be reliability, integrity, security, performance and feasibility.

1.2 Design of SCADA

While typically a best-of-breed solution could be used for procuring SCADA systems for a given set of requirements, it would be useful for both SCADA system designers and users if a systematic method could be employed for selecting the appropriate control network design and thereby the most satisfactory SCADA system. In this project we propose a technique for control network design that uses the concept of spanning trees – more specifically, minimum spanning trees [3], to develop potential design candidates that are then evaluated for suitability for system goals using the NFR Framework [4], where NFR stands for Non-Functional Requirements. Since most of the systems being controlled (electrical wiring, network routing, and others in practice) themselves form a spanning tree or sparse graphs [3], the number of different spanning trees for control networks are usually limited – therefore, minimum spanning trees give a good approximation to the SCADA architecture and the NFR Framework provides an ideal platform
for tradeoff analysis and suitability evaluation. The algorithm proposed in this project is evaluated by applying it to a practical case study, the Keystone pipeline project.

1.3 Keystone Gulf Coast Expansion Project

The proposed Keystone Gulf Coast Expansion Project [4] is complementary to the Keystone Pipeline and would serve existing refineries and markets on the U.S. Gulf Coast in Texas. It would link Canadian crude oil with North American ports.
2.1 Why SCADA:

- SCADA system eliminates the need for service personnel to visit each site for inspection, data collection/logging or make adjustments.

- Cheaper and better than Distributed Control systems:
  
  o SCADA systems are different from traditional Distributed Control Systems (DCS) which are generally found in plant sites. SCADA systems cover much larger geographic areas, whereas DCS’s cover the plant site.

- SCADA systems are used to cover much larger geographical areas.

- Some of the applications include Oil & Gas, Electrical & Water Utilities because of their remoteness of Assets.

2.2 Benefits of SCADA Systems:

A SCADA system when applied properly can help industries to save time and money. One reason is that with SCADA, you can eliminate the need for site visits by your personnel for inspection, adjustments and data collection. SCADA software enables you to monitor the operations in real time. It can also make modifications to the system, auto-generate reports and trouble-shoot.

Thus once the system is installed, it reduces operational costs and improves the efficiency of the set-up. SCADA systems are equipped to make immediate corrections in the operational
system, so they can increase the life-period of your equipment and save on the need for costly repairs. It also translates into man-hours saved and personnel enabled to focus on tasks that require human involvement. Further, the auto-generated reporting system ensures compliance with regulatory principles. Other benefits include:

- Reduces operational costs
- Provides immediate knowledge of system performance
- Improves system efficiency and performance
- Increases equipment life
- Reduces costly repairs

2.3 How does the SCADA System work?

The basic SCADA system is shown in Figure 1, where the SCADA master control center is connected to RTU, and RTU is connected to Field Instrument or Sensor.

A SCADA system usually consists of the following subsystems.

- A Human Machine Interface
- A Supervisory Control System
- Remote Terminal Units
Communication Infrastructure

- A Human-Machine Interface or HMI is the apparatus which presents process data to a human operator, and through this, the human operator, monitors and controls the process.
- A supervisory (computer) system collects data on the process and sends commands (control) to the process.
- Remote Terminal Units (RTUs) are connected to sensors in the processes; converting sensor signals to digital data and sends digital data to the supervisory system.
- Communication infrastructure connects the supervisory system to the Remote Terminal Units.

These functions are performed by four kinds of SCADA components:

- **Sensors** (either digital or analog) and **control relays** that directly interface with the managed system.
- **Remote Terminal units (RTUs)**. These are small computerized units deployed in the field at specific sites and locations. RTUs serve as local collection points for gathering reports from sensors and delivering commands to control relays.
- **SCADA Master Units**. These are larger computer consoles that serve as the central processor for the SCADA system. Master units provide a human interface to the system and automatically regulate the managed system in response to sensor inputs.
- The **Communications network** that connects the SCADA master unit to the RTUs in the field.

### 2.4 SCADA system for Oil Pipelines

Oil is transported from the oil wells to the refineries and refineries to various places through the pipelines. Since the oil pipelines travel through thousands of miles, a single system
cannot monitor the entire pipeline. Hence many devices like field instruments, programmable logic units and RTU’s are placed at different places throughout the pipeline. These field instruments and the programmable logic units are connected to RTU’s and the SCADA system is used to monitor all these RTU’s. Various parameters are being sensed by these devices namely pressure, temperature of the oil flowing inside the pipeline, density of the liquid flowing etc.

Since there are many RTU’s located at different places throughout the pipeline and since all these RTU’s communicate with the SCADA master system, rapid exchange of data takes place. The RTU’s sends the data to the SCADA master system and the SCADA master system checks if everything is working well or if there are any changes to be made. If there are any changes to be made, the SCADA master sends a message to the respective RTU and it performs the required action.

Any minor change in the pipeline can be detected by the SCADA master system and the required action can be taken. The application of SCADA has reduced huge amount man power requirements in the field which in turn has reduced the expenditure. Figure 2, is a brief high level architecture of a typical SCADA system for oil pipelines. In Figure 2 we can observe the sensors are connected to RTU’s using wired connections and then RTU’s are connected to SCADA master control center using wireless connection.

![Figure 2: The SCADA System for oil pipelines](image)
Chapter 3

MINIMUM SPANNING TREES

3.1 Spanning Trees

Given a connected, undirected graph $G = (V, E)$ where $V$ is the set of all vertices (RTU’s) in the network, and $E$ the set of all possible interconnections between pairs of nodes, then the acyclic subset $T \subseteq E$ that connects all of the vertices is called the spanning tree. In Figure 3 the spanning tree for the graph is shown with bold edges.

![Figure 3: A spanning tree (heavy edges) of a rigid graph](image)

![Figure 4: The minimum spanning tree of a graph, each edge is labeled with its weight](image)

3.2 Minimum Spanning Trees

If for each edge $(u, v) \in E$, we assign a weight $\omega(u, v)$ specifying the cost (in terms of units of length or money or any other) to connect $u$ and $v$, then the acyclic subset $T \subseteq E$ that connect all of the vertices and whose total weight $\omega(T) = \sum (u, v) \in T \omega(u, v)$, is minimized is referred to as the minimum spanning tree. In Figure 4 minimum spanning tree for the graph is shown with bold edges.
Figure 5: Typical oil pipeline SCADA system architecture

Consider the physical pipeline layout shown on top of Figure 5. This pipeline (could be water, natural gas, oil, or any other fluid) has a pump for increasing the fluid pressure, a valve
controller for directing the fluid along the correct outlet, and three pressure sensors for monitoring the pressure at the outlet. One possible control architecture for this system is shown in the middle of Figure 5 (architecture A) wherein the control links follow the actual system – this could mean that the medium of the pipe is used for transmitting electrical signals or the electrical cables follow the path of the pipe (for example, telephone signals that parallel rail lines or X10 [8] signaling that uses the wires for transmitting control signals as well). Possible control architecture for this system is shown in the bottom of Figure 5 (architecture B) wherein all the control links connect directly to the control center. Similarly, other control architectures are possible that connect the controlled entities (RTU’s) with their control center.

However, an important point to note is that all control architectures form a spanning tree – more importantly, a minimum spanning tree, where the minimization is performed according to a dominant criteria. For example, if cost of links (connecting components of the control network) were the most important criterion, the minimum spanning tree could be different from that obtained if the length of the links were the most important criterion. For example, in Figure 5, we could say that control architecture A, which is a spanning tree, is minimized from the perspective of cost of securing the installation while control architecture B, another spanning tree, is minimized from the perspective of the length of the links. For a given graph there can be different minimum spanning trees based on the minimization criterion and therefore it becomes important to choose from competing spanning trees or go for a hybrid of two or more spanning trees. This choice is guided by technical factors such as RTU technologies, communication technologies, data gathering techniques, telemetry and controls, and protocols. For this purpose the NFR Framework [4] provides an excellent tool to choose among alternatives and this Framework is discussed next.
Chapter 4

Non Functional Requirements Frame Work Approach

4.1 What is NFR approach?

The NFR Framework is briefly described in this section. The NFR Framework, where NFR stands for Non-Functional Requirements, requires the following interleaving tasks, which are iterative:

1. Develop the NFR softgoals and their decomposition.
2. Develop operationalizing softgoals and their decomposition.
3. Determine contributions between operationalizing softgoals and NFR softgoals.
4. Develop goal criticalities.
5. Evaluation and analysis.

The graph that results from the application of above steps is called the Softgoal Interdependency Graph (SIG). The NFR Framework uses the concept of goal satisfying. The notion of goal satisfying, of the NFR Framework assumes that decisions taken during the development process usually contribute only partially (or against) a particular goal, rarely “accomplishing” or “satisfying” goals in a clear-cut sense. Consequently any model is expected to satisfy NFRs within acceptable limits, rather than absolutely. There are different degrees of satisfying and the degrees are indicated by arrows annotated with + or – symbols. In the NFR Framework, each requirement (either system requirement or software requirement) is called an NFR softgoal (depicted by a cloud), while each potential configuration of a SCADA system is called an operationalizing softgoal (depicted by a dark cloud). The rationale for various decisions is captured by yet another softgoal – the claim softgoal (depicted by a cloud with dashed border).
The partial ontology of the NFR Framework is given in Figure 6 and the steps to use the NFR Framework are described below.

During the NFR goal decomposition process, the NFR softgoals are decomposed into their constituent softgoals based on the domain – in this paper we treat the requirements for the SCADA system as NFR softgoals. This decomposition is not unique and depends on what the people performing the decomposition consider important for the domain at that point in time. The softgoals are named using the following convention:

Type [Topic1, Topic2, ……]

Where Type is an NFR and Topic is the system or domain to which the Type applies. The NFR softgoals can be related to each other by three contributions: AND-contribution (indicated by single arc), OR-decomposition (indicated by double arc), and a refinement (only one child NFR softgoal).

![Figure 6: Partial Ontology of the NFR Framework](image)

The various architectures for a SCADA system and their elements are considered as operationalizing softgoals in the NFR Framework. Some NFR softgoals may be declared critical or priority softgoals by annotating them with ‘!’ marks, and this is done during step 3 of the NFR Framework application process. During NFR softgoal satisfaction determination, we determine the contributions made by the operationalizing softgoals to the various NFR softgoals – these contributions can be of four types: MAKE, HELP, HURT, and BREAK. These contributions have the following ranking:

**MAKE > HELP > HURT > BREAK.**

During the final step – the evaluation and analysis step – we apply the propagation rules of the NFR Framework to determine to what extent the model satisfies the NFR softgoals. While detailed propagation rules may be seen in, in this project based on IEEE paper we will be using the following simplified propagation rules:

R1. If most of the contributions received by a leaf NFR softgoal are positive (MAKE or HELP) then that leaf NFR softgoal is considered satisfied.

R2. If most of the contributions received by a leaf NFR softgoal are negative (BREAK or HURT) then that leaf NFR softgoal is considered denied or not satisfied.

R3. In the case of priority softgoals, or when there is a tie between positive and negative contributions, the system architect or the developer can take the decision based on or a variation of R1 and R2

R4. In the case of AND-contribution, if all the child softgoals are satisfied then the parent NFR softgoal is satisfied; else the parent softgoal is denied.
R5. In the case of OR-contribution, if at least one child softgoal is satisfied then the parent NFR softgoal is satisfied; else the parent softgoal is denied.

R6. In the case of refinement (only one child) the parent is satisfied if the child is satisfied; and the parent is denied if the child is denied.

Upon applying these propagation rules, if the root NFR softgoals are satisfied, then the goals of that SCADA system have been met to a large extent. Throughout the SIG development, the rationales for the various contributions are captured by claim softgoals.
Chapter 5
Algorithm for Efficient SCADA Design

The algorithm for optimal design of SCADA systems is given below:

**Step 1.** Draw the graph \( G = (V, E) \) of the required SCADA system – this graph will have the control centers and the RTU’s as nodes and the connections between the nodes as edges.

**Step 2.** Associate weights for edges in the graph of Step 1 – the weights may relate to real cost, perceived cost, length, or another measure; here perceived cost relates to cost of intangibles: for example, reliability of links may be related to cost of products used or to the strength of the signal used or to the amount of energy used, each of which may be interpreted in terms of a dollar value.

**Step 3.** For each weighted graph \( G \), determine the minimum spanning tree.

**Step 4.** Determine the suitability or otherwise of each minimum spanning tree for the goals of the SCADA system using the NFR Framework.

**Step 5.** Implement the most appropriate minimum spanning tree determined from the NFR Framework as the optimal SCADA design implementation; if more than one minimum spanning tree is determined to be appropriate then a hybrid design may be required.

5.1 **Explanation of algorithm with an example**

First we draw the graph of the expected SCADA system – here the RTU’s on the physical layout are treated as nodes and any control centers to which the RTU’s need to communicate are added to the graph as new nodes. Then add edges between nodes using wired and wireless
technologies – that is one graph has wired edges and the other has wireless edges (both of these follow the physical topology), the third has wired connections from each RTU direct to a control center, and the fourth has wireless connection from each RTU direct to that control center. All the four types of graphs are weighted according to length and cost and minimum spanning trees are determined for each graph. Each of the minimum spanning trees is evaluated for suitability using the NFR Framework and the most suitable set of minimum spanning trees are considered for implementation; if the set has more than one element then a hybrid design, which is the combination of two minimum spanning trees formed from different criteria like cost and length, may be the most optimal.

5.2 Application of algorithm to sample SCADA architectures

Our algorithm was applied to the SCADA requirements shown in Figure 5. For Step 1, we created the graph as shown in the first part (a) of Figure 7; here we consider the physical layout in Figure 5 and all possible wired connections to the nodes in the physical layout and this graph was weighted (Step 2) with the cost of the links given in Table 1.

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Equipment Used</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Wired</td>
<td>HP Ethernet 10GBase cable</td>
<td>$7900.00 per mile</td>
</tr>
<tr>
<td>2. Wireless</td>
<td>Wireless Extender zBoost YX610 Dual Band PCS and Cellular 800 / 1900 MHz 60db Amplifier Repeater Signal Booster kit</td>
<td>$604.00 each</td>
</tr>
</tbody>
</table>

Table1: Equipment used for connecting RTU’s and Master Control center and their cost.

For wired connections, cost of Ethernet cable per mile multiplied by distance in a typical installation. For wireless connections, one wireless booster is used for every 5 miles and the
weighted graph (Step 2) is shown in Figure 7, part (c). For Step 3, we implemented Prim’s algorithm for obtaining MST’s in Java programming language as shown in Appendix A. The two MST’s obtained are shown in part (b) and (d) of Figure 7 as MST A (for wired connections) and MST B (for wireless connections). For Step 4, we developed the Softgoal Interdependency Graph (SIG) and we apply the NFR evaluation for each SCADA design and final step we select the best suitable design as efficient design SCADA design.

![MST’s of Different Control Architectures of SCADA Systems](image)

**Figure 7:** MST’s of Different Control Architectures of SCADA Systems
Chapter 6
Case Study: Proposed Keystone Gulf Coast Expansion Project

6.1 Details of Keystone pipeline project

The Keystone pipeline project [4] is approximately 1,980-mile (3,200-kilometre), 36-inch crude oil pipeline that would begin at Hardisty, Alberta and extend southeast through Saskatchewan, Montana, South Dakota and Nebraska. TransCanada Company is working on this pipeline project. It would incorporate a portion of the Keystone Pipeline to be constructed through Kansas to Cushing, Oklahoma, before continuing through Oklahoma to a delivery point near existing terminals in Nederland, Texas to serve the Port Arthur, Texas marketplace. Also proposed is an approximate 50-mile (80-kilometre) pipeline to the Houston, Texas marketplace.

The Keystone Pipeline will be operated as one integrated pipeline system which will include the proposed Keystone XL pipeline project. The Keystone Pipeline has secured additional firm, long-term contracts totaling 380,000 barrels per day for an average term of approximately 17 years. With these commitments from shippers, TransCanada will proceed with the necessary regulatory applications in Canada and the U.S. for approvals to construct and operate an expansion and extension of the Keystone Pipeline system that will provide additional capacity of 500,000 barrels per day from Western Canada to the U.S. Gulf Coast in 2012.

When completed, the expansion will increase the commercial design of the Keystone Pipeline system from 590,000 barrels per day to approximately 1.1 million barrels per day. With the additional contracts, the Keystone Pipeline has now secured long-term commitments for 910,000 barrels per day for an average term of approximately 18 years. These commitments
represent approximately 83 per cent of the commercial design of the system. The Keystone

![Keystone Pipeline Map](image)

**Figure 8: Keystone project pipeline Map**

Pipeline system is expected to result in a capital investment of approximately US$12 billion between 2008 and 2012. Figure 8 shows the geographical location of pipeline from Hardisty (CANADA) to Houston and Port Arthur (USA).

6.2 Proposed SCADA architecture designs

In this project we are proposing a SACDA design using our NFR framework technique. Whole SCADA system will be controlled from one central location which will be the central
master of the SCADA system. As the pipeline covers huge geographical area, we have chosen to plant sub master stations, which in turn connect to the central master controller. Figure 9 shows the design for the SCADA system.

![Diagram of SCADA system architecture](image)

**Figure 9: Typical oil pipeline SCADA system architecture**

The sensors which take pressure or temperature readings will be connected to the RTU’s. We apply this efficient SCADA design algorithm for connecting RTU’s to the sub-master stations and sub-master stations to central master station. We have designed three architectures for the SCADA design. One of the architecture would be selected after applying to our efficient SCADA design algorithm.

The proposed architectures for establishing central SCADA control system were at:

1. Billings (Montana) using **wireless connections** to sub master stations.
2. Pierre (South Dakota) using both **wireless and wired connections** to various sub master stations.
3. Oklahoma City (Oklahoma) using **wired connections** to sub master stations.

Geographically these master control cities are located at top, middle and bottom areas of the pipeline. The sub-master stations were chosen to be major cities/tows closer to pipeline and left and right of the pipeline alternatively.

Ten sub master stations were chosen to connect to RTU’s, those sub master stations were located at:

1. Calgary (Alberta State, Canada).
2. Regina (Saskatchewan State, Canada).
3. Dickinson (North Dakota State, USA)
4. Rapid city (South Dakota State, USA)
5. Sioux city (Iowa State, USA)
6. Lincoln (Nebraska State, USA)
7. Topeka (Kansas State, USA)
8. Dodge city (Kansas State, USA)
9. Dallas (Texas State, USA)
10. Austin (Texas State, USA)

**6.3 Application of algorithm to proposed SCADA architectures**

**Step 1:** we draw the graph of the expected SCADA system – here the sub master stations on the physical layout are treated as nodes and any control centers to which the sub master stations need to communicate are added to the graph as new nodes. Then add edges between nodes using wired and wireless technologies – that is one graph has wired edges and the other has wireless edges (both of these follow the physical topology), the third has wired and wireless
connections from each sub master stations to control center. Now we show all the possible connections between various sub master stations and control center in a graph for three architectures.

**Step 2:** Each control architecture graphs are weighted and minimum spanning trees were determined by using Prim’s algorithm in Java programming language as given in Appendix A.

One of the possible associated weights between the nodes is Distance/Cost. The cost for connecting nodes in each graph is determined by the communication medium used. The equipment cost per unit is given in Table 2.

![Figure 10(a): Control Architecture A, weighted graph with cost as criteria](image-url)
Table 2: Equipment used for connecting RTU’s and Master Control center and their cost.

For wired connections, cost of Ethernet cable per mile multiplied by distance in a typical installation. For wireless connections, one wireless booster is used for every 5 miles.

```
<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Equipment Used</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Wired</td>
<td>Belkin RJ-45 CAT5e 10/100 Mbps Ethernet Cable with Snagless Design.</td>
<td>$4800.00 per mile</td>
</tr>
<tr>
<td>2. Wireless</td>
<td>Wireless Extender zBoost YX610 Dual Band PCS and Cellular 800 / 1900 MHz 60db Amplifier Repeater Signal Booster kit.</td>
<td>$604.00 each</td>
</tr>
</tbody>
</table>
```

Figure 10(b): Control Architecture B, weighted graph with cost as criteria
Figure 10(a) shows possible connections between various sub master stations and control center at Oklahoma City (Oklahoma, USA) in a graph and cost associated to connect each node.

Figure 10(b) shows possible connections between various sub master stations and control center at Pierre (South Dakota, USA) in a graph and cost associated to connect each node.

Figure 10(c): Control Architecture C weighted graph with cost as criteria
The Control Architecture B is hybrid design which is the combination of both wired and wireless connections in between the nodes.

**Wireless connections in the Control Architecture B:** Pierre(USA) to Regina(Canada), Pierre(USA) to Rapid city(USA), Pierre to Calgary(Canada), Dickinson(USA) to Lincoln (USA), Rapid city (USA) to Dodge city(USA), Sioux city(USA) to Dallas(USA), Topeka(USA) to Dodge city(USA), Dodge city(USA) to Pierre(USA), Topeka(USA) to Dallas(USA), Pierre(USA) to Austin(USA), Pierre(USA) to Topeka(USA).

**Wired connections in the Control Architecture B:** Calgary (Canada) to Rapid city(USA), Dickinson(USA) to Rapid city (USA), Pierre(USA) to Sioux city(USA), Dickinson(USA) to Sioux city(USA), Sioux city(USA) to Lincoln (USA), Lincoln(USA) to Dodge city(USA), Dallas(USA) to Austin(USA), Dodge City (USA) to Austin (USA), Calgary(Canada) to Dickinson(USA).

Figure 10(c) shows possible connections between various sub master stations and control center at Billings City (Montana, USA) in a graph and cost associated to connect each node.

**Step 3:**

We developed a Java language program that used the Prim’s method as shown in Appendix A, to determine minimum spanning trees (MST’s) for each Control Architecture graphs and the MST’s were obtained. The MST’s were shown in part (a); (b) and (c) of Figure 11 as MST A (for wired connections) and MST B (both wired and wireless connections) and MST C (for wireless connections).
The minimum cost required to connect all the sub master stations to central master location at Oklahoma City was $13,579,200.00
The minimum cost required to connect all the sub master stations to central master location at Pierre was $618,000.00.
Figure 11(c): MST for Control Architecture C

The minimum cost required to connect all the sub master stations to central master location at Billings (USA) was $419,417.00
Chapter 7

Application of the NFR approach for Evaluation of Efficient SCADA Design

This chapter primarily focuses on the selecting the best MST Control Architecture from three different SCADA systems and evaluates them using NFR approach. The following steps are followed along this approach. This chapter uses symbols, terms and definitions from previous chapters. The final step involves in choosing the efficient SCADA design for the keystone pipeline project.

Step 4: We developed the Softgoal Interdependency Graph (SIG) for the three MST’s. In each SIG we decomposed the goals for the SCADA system (represented by the NFR softgoal Goals [SCADA]) into six child NFR softgoals: Feasibility[SCADA], Reliability [SCADA], Integrity [SCADA], Security [SCADA], Performance [SCADA] and Cost[SCADA]; this is an AND-decomposition as indicated by a single arc which means all these goals need to be satisfied. The rationale for this decomposition is that these characteristics are most important for oil pipeline SCADA (as discussed in the Introduction). We also indicated three of these softgoals (Reliability and Security and Cost as high priority by ‘!’ symbols). Each of these NFR softgoals (Reliability, Integrity, Security, Performance) was decomposed into corresponding child softgoals for nodes and links, Where as Feasibility NFR softgoal is decomposed into Resources and Approvals (as indicated by the topics in the softgoal names).

Feasibility[SCADA] is AND-decomposed into Resources and Approval, Reliability[SCADA] is AND-decomposed (as indicated by the single arc) into Reliability[Nodes] and Reliability[Links], Integrity[SCADA] is OR-decomposed (as indicated by the double arc) into Integrity[Nodes] and Integrity[Links], Security[SCADA] is AND-decomposed into
Security[Nodes] and Security[Links], and Performance[SCADA] is AND-decomposed into Speed[Nodes] and Speed[Links].

7.1 Application of NFR approach for evaluation of SCADA design at Oklahoma City

Figure 12(a) shows the SIG graph for MST A (i.e.) SCADA design at Oklahoma City and justifications (claim softgoals) for the contributions (HELP or +, MAKE or ++, HURT or --, and BREAK or -) made by the MST A were determined.

Figure 12(a): SIG for Control Architecture A (i.e. at Oklahoma City)
7.2 Application of propagation rules to SIG graph of Control Architecture at Oklahoma City

As shown in Figure 12(a) NFR propagation rules are applied to SIG graph of Control Architecture at Oklahoma City

1. By R1:

   The leaf Feasibility factor [Resources] is MAKE satisfied as Oklahoma City will have all the resources required.

   The leaf Feasibility factor [Approvals] is HURT satisfied as it might be difficult to get approvals in the city due to heavy industrialization.

   The leaf Reliability factor [Nodes] is MAKE satisfied due to high quality equipments.

   The leaf Reliability factor [Links] is MAKE satisfied as wired links are more reliable.

   The leaf Integrity factor [Nodes] is MAKE satisfied if we use single vendor.

   The leaf Integrity factor [Links] is Break satisfied if we use multiple vendors due to synchronizing or integrity problems.

   The leaf Security factor [Nodes] is MAKE satisfied as tight physical and data security is maintained.

   The leaf Security factor [Links] is MAKE satisfied as wired links are more reliable and tight physical security.

   The leaf Speed factor [Nodes] is MAKE satisfied due to high quality components usage at nodes.

   The leaf Speed factor [Links] is HURT satisfied as large distances affect the speed.

   The leaf Cost is HURT satisfied due to high cost in digging and establishing wired networks, which involve huge cost of Ethernet cables and other equipment.
2. By R4, Feasibility factor is HURT satisfied
3. By R4 Reliability factor is MAKE satisfied
4. By R5 Integrity factor is MAKE Satisfied
5. By R4, Security factor is MAKE satisfied.
6. By R4, Performance factor is HURT satisfied
7. By R4, the overall the SCADA design is HURT satisfied, due to Performance and Feasibility factors, as those were disadvantages.

7.3 Application of NFR approach for evaluation of SCADA design at Pierre

![Diagram of SCADA design at Pierre](image_url)

Figure 12(b): SIG for Control Architecture B (i.e. at Pierre)
7.4 Application of propagation rules to SIG graph of Control Architecture at Pierre

As shown in Figure 12(b) NFR propagation rules are applied to SIG graph of Control Architecture at Pierre.

1. By R1:

The leaf Feasibility factor [Resources] is MAKE satisfied as Oklahoma City will have all the resources required.

The leaf Feasibility factor [Approvals] is MAKE satisfied as it would be less difficult to get approvals in a town/county due to less population and more vacant space.

The leaf Reliability factor [Nodes] is MAKE satisfied due to high quality equipments.

The leaf Reliability factor [Links] is MAKE/HURT satisfied as this design has combination of wireless and wired links. Wired networks are more reliable and wireless networks are less reliable.

The leaf Integrity factor [Nodes] is MAKE satisfied if we use single vendor.

The leaf Integrity factor [Links] is Break satisfied if we use multiple vendors due to synchronizing or integrity problems.

The leaf Security factor [Nodes] is MAKE satisfied as tight physical and data security is maintained.

The leaf Security factor [Links] is MAKE/HURT satisfied as this design has combination of wireless and wired links. Wired links are more reliable and tight physical security and wireless links might have some security issues.

The leaf Speed factor [Nodes] is MAKE satisfied due to high quality components usage at nodes.

The leaf Speed factor [Links] is MAKE satisfied as wireless links are speeder.
The leaf Cost is HELP satisfied due to modest cost involved, by making use of wired networks where establishing wireless networks is not feasible. Thus decreasing the establishment cost.

2. By R4, Feasibility factor is MAKE satisfied

3. By R3 Reliability factor is MAKE satisfied, where according to propagation rule R3 (the system architect or the developer can take the decision based on or a variation of R1 and R2)

4. By R5 Integrity factor is MAKE Satisfied

5. By R3, Security factor is HELP satisfied, where according to propagation rule R3 (the system architect or the developer can take the decision based on or a variation of R1 and R2)

6. By R4, Performance factor is MAKE satisfied, wireless has higher speed than wired networks, this design has both wired and wireless links.

7. By R4, the overall the SCADA design is HELP satisfied.
7.5 Application of NFR approach for evaluation of SCADA design at Billings

Figure 12(c): SIG for Control Architecture C (i.e. at Billings)

7.6 Application of propagation rules to SIG graph of Control Architecture at Billings

As shown in Figure 12(c) NFR propagation rules are applied to SIG graph of Control Architecture at Billings.
1. By R1:

   The leaf Feasibility factor [Resources] is MAKE satisfied as Oklahoma City will have all the resources required.

   The leaf Feasibility factor [Approvals] is HURT satisfied as it might be difficult to get approvals in the city due to heavy industrialization.

   The leaf Reliability factor [Nodes] is MAKE satisfied due to high quality equipments.

   The leaf Reliability factor [Links] is HURT satisfied as wireless links are less reliable.

   The leaf Integrity factor [Nodes] is MAKE satisfied if we use single vendor.

   The leaf Integrity factor [Links] is Break satisfied if we use multiple vendors due to synchronizing or integrity problems.

   The leaf Security factor [Nodes] is MAKE satisfied as tight physical and data security is maintained.

   The leaf Security factor [Links] is HURT satisfied as wireless links are less reliable and more secure mechanisms are required.

   The leaf Speed factor [Nodes] is MAKE satisfied due to high quality components usage at nodes.

   The leaf Speed factor [Links] is MAKE satisfied as wireless links are faster than wired links.

   The leaf Cost is MAKE satisfied due to low cost than wired networks, where it eliminates Cables.

2. By R4, Feasibility factor is HURT satisfied

3. By R4 Reliability factor is HURT satisfied

4. By R5 Integrity factor is MAKE Satisfied
5. By R4, Security factor is HURT satisfied.

6. By R4, Performance factor is MAKE satisfied

7. By R4, the overall the SCADA design is HURT satisfied, due to Reliability, Integrity and Feasibility factors, as those were disadvantages.

7.7 Case Study Results:

Figure 13: Efficient SACADA design selected by applying NFR approach
Control Architecture selected for connecting the sub-master stations and the central master control center was shown in Figure 13.

From the three SIG graphs shown in figure 12 (a) (b) (c), we can compare the overall weight age of three SCADA designs. Clearly, the SCADA design at Pierre (USA) has better design using the combination of wired and wireless networks. Even though the cost for Billings (USA) is less when compared to other SCADA designs, by considering other factors such as Security and Feasibility factors makes the SCADA architecture B which is at Pierre(USA) chosen for having master control center.
Chapter 8

Conclusion and Future Work

Current approaches to designing SCADA systems typically focus around purchasing best-of-breed solutions – however, it will be useful to both developers of SCADA systems and their users if a systematic approach to designing such systems were available. In this paper we propose an efficient technique for designing such SCADA systems based on minimum spanning trees and the NFR Framework. Typical SCADA solutions conform to a spanning tree structure [3] simply because the systems monitored and controlled are typically sparse networks such as electrical wiring, oil pipelines, railway lines and the like; therefore, minimum spanning trees that form when spanning trees are minimized based on a criterion such as cost or length are natural candidates for SCADA control architectures. However, how do we ensure optimal solutions? For this we use the NFR Framework [4], a qualitative framework that allows selection among competing designs using a goal-oriented approach to evaluate amongst tradeoffs. We verified our algorithm by applying to a case study and applied to keystone pipeline project and came up with efficient SCADA design.

There are several directions for further research. Most importantly we need to define the parameters for creating the graph in Step 1 of the algorithm and for weighting the graph in Step 2 so that too many possibilities are not created for the MST’s. We need to implement the design in a simulated system, if not the actual system, to verify the algorithm. However, we believe our approach provides an efficient means for designing SCADA systems.
Chapter 9

Bibliography


Appendix A

Prim’s Algorithm implementation in Java Programming Language:

```java
import java.io.*;
class Prim {
    public static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    static int [][] G;
    static int [][] t;
    static int[] near;
    static int n;
    static int mincost = 0;
    static int k, l;

    public static void main(String[] args) throws IOException {
        System.out.println("Prim's Algorithm");
        System.out.print("Enter the number of the vertices: ");
        n = Integer.parseInt(br.readLine());
        G = new int[n+1][n+1];
        t = new int[n+1][3];
        near = new int[n+1];
        System.out.print("If edge between the following vertices enter its distance (not more than 7000) else 0:
        
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                {
                    if((i!=j)&&(i<j))
                        {
                            System.out.print(i+" and "+j+": ");
                            G[j][i] = G[i][j] = Integer.parseInt(br.readLine());
                            if(G[i][j] == 0 )
                                G[j][i] = G[i][j] = 7001;
                        }
                    if(i==j)
                        G[i][j]=7001;
                }
```

---

42
prims();
System.out.println("\n\nSolution : \n\n");

for (int i = 1; i<=n; i++)
{
    if ( (t[i][1]!=0) && (t[i][2] !=0) )
        System.out.println(t[i][1] + " -" +t[i][2]);
}
System.out.println("\n\n\nMinimum cost incurred is: "+ mincost);

static void prims()
{
    getMinKL();
    mincost = G[k][l];
    t[1][1] = l;
    t[1][2] = k;
    for(int i=1; i<=n; i++)
        near[i] = (G[i][l]<G[i][k])?l:k;
    near[k] = near[l] = 0;

    for(int i=2; i<n; i++)
    {
        int j = getMin();
        t[i][1] = j; t[i][2] = near[j];
        mincost = mincost+G[j][near[j]];
        near[j] =0;
        for (int k=1; k<=n; k++)
            if( (near[k] !=0) && G[k][ near[k] ]> G[k][j] )
                near[k] =j;
    }
}

static int getMin()
{
    int j=1;
    for(int i=1;i<=n;i++)
        if(near[i] !=0)
            j = i;
break;
}

for(int i=1;i<=n;i++)
    if(near[i] !=0)
        if(G[j][near[j]] > G[i][near[i]])
            j =i;
    return j;

static void getMinKL()
{
    int k1 = 1, l1 = 2;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            if((i!=j)&&(i<j))
            {
                if((G[i][j] < G[k1][l1]) && G[i][j] !=0 )
                {
                    k1 = i;
                    l1 = j;
                }
            }
        }
    if(G[k1][l1] !=0 )
    {
        k =k1; l=l1;
    }
}
A.1 Finding minimum spanning tree for control architecture A at Oklahoma City

Figure 14 shows the design for control architecture A at Oklahoma City, with all the possible connections between sub-master stations and master station. The graph is weighted with distance as criteria. The interaction with the Java program for calculating MST given below:
Enter the number of the vertices: 11

If edge between the following vertices enters its distance else 0:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 and 2:</td>
<td>1361</td>
<td>3 and 4:</td>
</tr>
<tr>
<td>1 and 3:</td>
<td>0</td>
<td>3 and 5:</td>
</tr>
<tr>
<td>1 and 4:</td>
<td>0</td>
<td>3 and 6:</td>
</tr>
<tr>
<td>1 and 5:</td>
<td>0</td>
<td>3 and 7:</td>
</tr>
<tr>
<td>1 and 6:</td>
<td>493</td>
<td>3 and 8:</td>
</tr>
<tr>
<td>1 and 7:</td>
<td>0</td>
<td>3 and 9:</td>
</tr>
<tr>
<td>1 and 8:</td>
<td>270</td>
<td>3 and 10:</td>
</tr>
<tr>
<td>1 and 9:</td>
<td>0</td>
<td>3 and 11:</td>
</tr>
<tr>
<td>1 and 10:</td>
<td>185</td>
<td>4 and 5:</td>
</tr>
<tr>
<td>1 and 11:</td>
<td>0</td>
<td>4 and 6:</td>
</tr>
<tr>
<td>2 and 3:</td>
<td>416</td>
<td>4 and 7:</td>
</tr>
<tr>
<td>2 and 4:</td>
<td>0</td>
<td>4 and 8:</td>
</tr>
<tr>
<td>2 and 5:</td>
<td>699</td>
<td>4 and 9:</td>
</tr>
<tr>
<td>2 and 6:</td>
<td>0</td>
<td>4 and 10:</td>
</tr>
<tr>
<td>2 and 7:</td>
<td>0</td>
<td>4 and 11:</td>
</tr>
<tr>
<td>2 and 8:</td>
<td>0</td>
<td>5 and 6:</td>
</tr>
<tr>
<td>2 and 9:</td>
<td>0</td>
<td>5 and 7:</td>
</tr>
<tr>
<td>2 and 10:</td>
<td>0</td>
<td>5 and 8:</td>
</tr>
<tr>
<td>2 and 11:</td>
<td>0</td>
<td>5 and 9:</td>
</tr>
</tbody>
</table>

Solution:

The resultant graph will have edges between the below nodes:

11-10
1-10
8-1
9-8
7-9
6-9
5-7
4-5
3-4
2-3

Minimum distance incurred is: 2829 Miles

Minimum cost incurred is: Distance * cost of Ethernet cable per mile i.e. $13,579,200.00
Figure 15, shows the minimum spanning graph constructed from the results of the algorithm where the cost for connecting various edges is obtained by multiplying with Ethernet Cable cost per mile and distance. Similarly the MST B (Figure 11(b)) and MST C(Figure 11(c)) are obtained.

Figure 15: MST for Control Architecture A