SCADA Topology 알고리즘 분석

최영민, 유현정, 박용조, 김성학
한국전력거래소

Analysis of SCADA Topology Algorithm

Choi, Young min Yu, Hyun-Jung Park, Yong Jo Kim, Sung Hak
Korea Power Exchange

Abstract - 최근 들어 전력 계통은 점차 복잡해지고 계통 규모 역시 빠른 속도로 성장하고 있다. 이러한 환경하에서 전력계통의 안정적, 경제적 운영을 담당하고 있는 한국전력거래소는 EMS(Energy Management System)를 통해 실시간 전력계통에 대한 정확한 판단을 기반으로 전력계통의 안정성과 경제성 확보에 주력하고 있다. EMS의 다양한 기능 중 스카다(SCADA) 기능은 단순히 취득 데이터를 처리하는 기능뿐 아니라 인텔리전트한 기능을 탑재하고 있는데 이중 대표적 기능이 스카다 토폴로지(SCADA Topology)라 할 수 있다. 스카다 토폴로지는 그 이름에서 알 수 있듯이 스카다에서 실시간 취득, 처리되는 아날로그와 스테이터스 데이터를 기반으로 전력계통에서 운용하는 각종 전력설비(발전기, 송전선로, 변압기, 조상설비 등)에 대한 가압 또는 무압 여부를 스캔주기(2초)내에 결정하는 기능을 말한다. 현재 전력거래소는 100인치 화면 16장을 연결한 Rear Projector에 전국전력계통도 화면을 제작하여, 한눈에 전력계통의 가압 또는 무압상태를 표시, 실시간 감시에 활용하고 있다. 그럼에도 국내에서는 스카다 토폴로지에 대해 소개된 논문이나 소개가 부재한 형편임을 감안하여, 본고는 스카다 토폴로지의 기본 알고리즘을 분석하여 국내 스카다 기능의 선진화에 기여하고자 한다.

1. 서 론

현재와 같이 계통규모가 점차로 증가하고 복잡해지고 있는 상황에서 전력계통의 정확한 감시는 전력계통의 안정적 운영을 위한 필수사항이라 할 수 있다. 하지만, EMS는 아날로그 4만개, 스테이터스 2만개의 정보를 처리하고 있어, 감시자가 수동으로 전력기기의 가압여부를 판단하기엔 정보의 양이 너무 방대하다. 또한, 국내 전력계통망이 환상망으로 구성되어 있는 점, 아날로그 정보의 비정상 취득 가능성 역시 전력기기의 가압여부를 수동으로 판단하는 것은 매우 어려운 일임을 반증한다. 이런 문제점을 해소하기 위해 EMS는 "스카다 토폴로지" 기능을 탑재하여 상기의 문제를 해결하고 있다. 이 기능은 토폴로지(연결관계) 처리를 매우 단순화시키는 알고리즘을 통해 전력기기의 가압여부를 신속히 판단하여 실시간 경보로 처리, 감시자에게 전력기기의 가압 또는 무압 정보를 제공한다. 이러한 기능은 소위 인텔리전트 알람기능과 매우 유사한 측면을 가지고 있는데, 실제 전력계통의 다수 고장시 발생하는 무수한 알람, 즉 수십개 차단기의 개폐알람을 전력기기 알람으로 응축하는 효과를 지니기 때문이다.

본론에서는 스카다 토폴로지 프로그램에서 사용하는 자료구조(객체)와 BUS와 ISNALD의 구성 그리고 가압판단 등에 대한 내용을, 가상 계통을 예로 들어 설명하도록 한다.

2. 본 론

2.1 스카다 토폴로지 자료구조(Objects)

스카다 토폴로지 프로그램의 알고리즘을 설명하기 위해서 가장 먼저 제시해야 할 것은 "자료구조"로서, 간략한 설명과 함께 객체(Object)와 속성(Property)을 표형태로 제시하였다.

2.1.1 Breaker 객체

Breaker 객체는 차단기(단로기)를 표현하기 위한 자료구조이며, near(또는 far)Node 속성은 차단기가 연결된 오른쪽(위) 또는 왼쪽(아래)의 연결된 Node 정보를 표현한다.

<표 1> Breaker 객체 속성

속 성	설 명	속 성	설 명
ID	이름	nearND	near Node포인터
isClosed	개폐여부	farND	far Node포인터
isUnknown	정상취득여부		

2.1.2 Measurement 객체

Measuerment 객체는 모든 아날로그가 아닌 전압 측정값만을 대상으로 표현한다.

<표 2> Measuerment 객체 속성

속 성	설 명	속 성	설 명
ID	이름	ND	Node 포인터
Value	측정값(kV)	Thres	가압판단 임계값

2.1.3 Device 객체

Device 객체는 전력기기(발전기, 선로 등)를 표현하는 객체로 단일터미널 기기(발전기, 조상설비 등)의 경우 nearNode와 farNode를 같게 설정한다.

<표 3> Device 객체 속성

속 성	설 명	속 성	설 명
ID	이름	nearND	near Node 포인터
farND	far Node포인터		

2.1.4 Station 객체

Station객체는 전국 발전소와 변전소를 표현한다.

<표 4> Station 객체 속성

속 성	설 명	속 성	설 명
ID	이름	numOfNode	Station에 속한 Node객체수

2.1.5 Node 객체

Node객체는 전기적 연결상태(Topology)를 표현하며, 모델링과정에서 정의한다. Node객체 속성중 Bus와 next는 동적으로 할당한다.

<표 5> Node 객체 속성

속 성	설 명	속 성	설 명
ID	이름	next	다음 Node에 대한 포인터
Bus	Bus 객체포인터		

2.1.6 Bus 객체

Bus 객체는 Node객체의 상위개념으로 전기적 연결상태를 표현하며, 프로그램 수행시 동적으로 생성한다.

<표 6> Bus 객체 속성

속 성	설 명	속 성	설 명
ND	Node 객체포인터	next	다음 Bus에 대한 포인터
Island	Island 객체 포인터	Station	Node가 속한 Station포인터

2.1.7 Island 객체

Island 객체는 Bus객체의 상위개념으로 전기적 연결상태를 표현하며, 프로그램 수행시 동적으로 생성한다.

<표 7> Island 객체 속성

속 성	설 명
Bus	Bus객체 포인터
numOfLive	가압으로 판단한 측정값 객체수
numOfDead	무압으로 판단한 측정값 객체수
numOfTotal	아일랜드에 속한 측정값 객체 총수
isLive	아일랜드 가압여부
isDead	아일랜드 무압여부
isUnknown	판단불가

2.1.8 System-Wide 파라미터

System-Wide 파라미터는 퍼센트값으로 ISLAND의 최종 가압 또는 무압상태를 결정하는 데 사용된다.

<표 8> System 객체 속성

속 성	설 명	속 성	설 명
DeadThres	무압 Threshold	LiveThres	가압 Threshold

상기 제시한 객체 중 BUS와 ISLAND 객체는 프로그램 수행과정에서 동적으로 만들어지며, 그 외의 객체들은 모델링과정에서 정적으로 정의한 객체들이다. 또한 상기 객체들은 스카다 토폴로지의 알고리즘만을 설명하기 위해 필요한 최소한 객체와 속성만을 나타내었으며, 실제 응용프로그램을 제작하기 위해서는 더 많은 객체와 속성들이 연구되어야 함을 전제한다.

2.2 가상계통도와 모델링

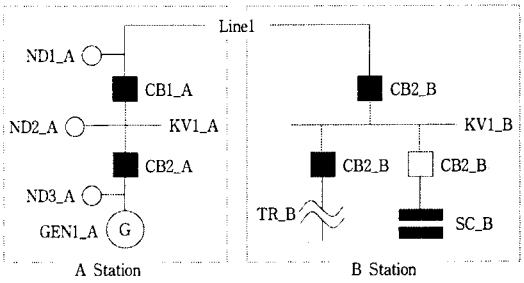
위에서 설명한 객체를 가상 계통도를 통해 모델링하는 예시를 이번 장에서 설명하겠으며, 본고에서는 이 가상계통도를 통해 앞으로 전개하고자 한다.

2.2.1 모델링

다음은 <그림 1>의 가상계통도를 기준으로 2.1절에서 제시한 객체를 모델링한 예시이다.

<표 9> 주요 객체 모델링 예

객체명	Breaker 객체 (총 5개)
객체예	Breaker(1).ID : CB1_A Breaker(1).isClosed : True Breaker(1).isUnknown : False Breaker(1).nearND : 1(ND1_A) Breaker(1).farND : 2(ND2_A)
객체명	Measurement 객체 (총 2개)
객체예	Measurement(1).ID : KV1_A Measurement(1).Value : 155 Measurement(1).Node : 2(ND2_A)
객체명	Device 객체 (총 4개)
객체예	Device(1).ID : GEN1_A Device(1).nearND : 3(ND3_A)
객체명	Node 객체 (총 7개)
객체예	Node(2).ID : ND2_A Node(2).Bus : Null Node(2).Next : Null
객체명	Station 객체 (총 2개)
객체예	Station(1).ID : A Station(1).numOfNode : 3



<그림 1> 가상 계통도

2.2.2 Bus의 구성

Bus는 Node의 상위개념으로, Bus 구성 알고리즘은, 1단계로 Station객체에 속하는 모든 Node에 1:1로 대응하는 Bus객체 생성 및 Bus와 Node간 연결관계 및 속성을 초기화한다. 2단계로 차단기상태가 닫힘일 경우, far Node의 Bus 속성을 near Bus로 통합하고, near Node의 next 속성을 far Node로 하여 Node객체의 연결관계를 정의한다. 이것을 의사코드 (psedo code)로 나타내면 다음과 같고, <표 10>는 상기의 알고리즘으로 Bus 구성시 Bus 및 Node 객체의 속성값 변화를 보여주고 있다.

```
Loop Node(i) in Station
  Bus(i).Node = i
  Node(i).Bus = i
  Node(i).next = 0
End Loop

Loop Breaker(i) in Station
  If Breaker(i).isClosed
    nearND = Breaker(i).nearND
    farND = Breaker(i).farND
    nearBus = Node(nearND).Bus
    farBus = Node(farND).Bus
    Node(Bus(farBus).Node).Bus = nearBus
    Node(nearND).next = Bus(farBus).Node
    Bus(farBus).Node = 0
End Loop
```

<표 10> Bus 및 Node객체의 속성값 변화표

객체	속성	1단계	2단계	객체	속성	1단계	2단계
ND1_A (Node(1))	Bus	1	1	Bus(1)	ND	1	1
	next	0	2				
ND2_A (Node(2))	Bus	2	1	Bus(2)	ND	2	0
	next	0	3				
ND3_A (Node(3))	Bus	3	1	Bus(3)	ND	3	0
	next	0	0				
ND1_B (Node(4))	Bus	4	4	Bus(4)	ND	4	4
	next	0	5				
ND2_B (Node(5))	Bus	5	4	Bus(5)	ND	5	0
	next	0	6				
ND3_C (Node(6))	Bus	6	4	Bus(6)	ND	6	0
	next	0	0				
ND4_B (Node(7))	Bus	7	7	Bus(7)	ND	7	7
	next	0	0				

2.2.3 Island의 구성

Island는 Bus의 상위개념으로, Island를 구성하기 위한 알고리즘은, 1단계 Bus객체에 1:1로 대응하는 Island객체 생성 및 속성을 초기화한다. 2단계로 모든 Device객체를 기준으로 Device→Node→Bus 정보를 기반으로 near와 far Bus의 아일랜드가 다른 경우, far Bus측 Island를 near Bus의 아일랜드로 재설정하고, near Bus의 next속성을 far Bus로 한다. 이것을 의사코드(psedo code)로 나타내면 다음과 같고, <표 11>은 상기의 알고리즘으로 Island 구성시 Island 및 Bus 객체의 속성값 변화를 보여주고 있다.

```
Loop Bus(i)
  Island(i).Bus = i
  Bus(i).Island = i
  Bus(i).next = 0
End Loop

Loop Device(i)
  nearBus = Node(Device(i).nearND).Bus
  farBus = Node(Device(i).farND).Bus
  If neaBus.Island not equal farBus.Island
    Bus(farBus).Island = Bus(nearBus).Island
    Bus(nearBus).next = farBus
End Loop
```

<표 11> Island 및 Bus객체의 속성값 변화표

객체	속성	1 단계	2 단계	객체	속성	1 단계	2 단계
Bus(1)	Island	1	1	Island(1)	Bus	1	1
	next	0	4				
Bus(4)	Island	4	1	Island(4)	Bus	4	1
	next	0	0				
Bus(7)	Island	7	7	Island(7)	Bus	7	7
	next	0	0				

2.2.4 Island의 가압여부 및 Device 가압여부 결정

지금까지 차단기 상태를 기준으로 Node, Bus와 Island 객체의 연관관계의 정의 즉, 토폴로지의 구성을 살펴보았다. 이번 장은 Island의 가압여부로, 이것은 Measurement객체의 Node 속성을 기준으로 Bus→Island의 관계를 이용, Value속성값과 Thres값의 비교를 통해 결정하며, 이것을 다시 System객체의 임계값과 비교를 통해 최종 판단을 하게된다. 이를 의사코드로 나타내면 다음과 같고, 각 전력기기의 가압여부는 Device객체의 Node속성을 기반으로 Island의 연관관계를 통해 최종 판단이 가능하게 된다.

```
Loop Measurement(i)
  island_idx = Bus(Node(Measurement(i).Node).Bus).Island
  Island(island_idx).numOfTotal++
  If (Measurement(i).Value >= Measurement(i).Thres)
    Island(island_idx).numOfLive++
  Else Island(island_idx).numOfDead++
End Loop

Loop Island(i)
  LiveRate = Island(i).numOfLive/Island(i).numOfTotal * 100
  DeadRate = Island(i).numOfDead/Island(i).numOfTotal * 100
  If LiveRate >= System.LiveThred
     Island(i).isLive = True
  Else If DeadRate >= System.DeadThred
     Island(i).isDead = True
  Else Island(i).isUnknown = True
End Loop
```

3. 결 론

본 논문에서는 스카다 토폴로지의 자료구조, 이를 기반한 토폴로지의 구성을 통해 전력기기의 가압 또는 무압여부를 판단하는 알고리즘을 소개하였다. 스카다 토폴로지 기능은 실시간 처리라는 요구사항을 감안하여 완벽한 계통 토폴로지 구성보다는 각 객체들의 임계값을 통해 문제를 해결한 솔루션이다. 현재 전력거래소의 운영경험에 비추면, 스카다 토폴로지 기능은 거의 100% 신뢰할 만한 정보를 제공하고 있다. 서론에서 제시했던 알람처리나 아날로그의 비정상 상태에 대한 내용을 구체적으로 나타내진 못했지만, 본 논문에서 제시한 내용만으로도 충분히 유추가 가능할 것이며, 발 변전소 단위의 가압여부까지 처리할 수 있을 것이다.

앞으로 본 논문의 스카다 토폴로지 알고리즘을 현재 운용중인 스카다 시스템에 응용한다면 국내 전력IT 기술의 진화를 가져올 것으로 판단한다.