

원전 정보처리계통의 네트워크 부하 감소를 위한 Subscription방식의 Protocol에 관한 연구

오응세, 정연섭, 성찬호, 허태영, 강성곤
전력연구원
대전광역시 유성구 문지동 103-16

요 약

본 논문에서는 차세대 원자력발전소 주제어실에서 감시 및 제어용 디스플레이 기기의 증가에 따른 정보처리계통(Information Processing System)의 네트워크 부하 감소를 위해 주기적 Subscription방식의 통신 프로토콜을 구현하고, 실험을 통하여 실질적인 디스플레이 수와 각종 운전상태에 대해서 기존의 프로토콜에 비해 우수한 성능을 나타냄을 보였다.

1. 서 론

원자력발전소의 정보처리계통(IPS)은 다른 공정제어 시스템에 비해 처리하는 정보의 양이 방대하고, 높은 수준의 신뢰도(reliability)가 요구된다. 복잡하고 방대한 정보를 운전원(operator)에게 효과적으로 신속하고 정확하게 전달해 주기 위해서 제어 시스템과는 별도로 정보처리용 네트워크를 구성하는 것이 일반화되고 있는 추세이다. 운전원 워크스테이션의 CRT로 정보를 전달하는 방법으로는 전체공정 데이터를 주기적으로 CRT에 전달하여 화면을 갱신하는 방법이나 변화된 데이터만 전송하는 이벤트(Event) 전송 방법을 주로 사용하고 있다. 그러나 연결된 디스플레이 기기가 증가하고, 공정데이터의 수가 증가할수록 네트워크 상의 정보 전달속도가 저하되고, 이벤트 전송의 경우 비정상 운전시 데이터의 변화량이 급격히 증가하게 되면 통신 지연 시간 증가를 예측하기 어려운 단점이 있다[4].

따라서 본 논문에서는 이러한 정보처리계통의 네트워크 부하(負荷)를 줄이기 위하여, 발전소 전체 데이터 중 각 CRT화면에서 필요한 데이터만을 선택하는 Subscription 방식과 주기적인 전송방식을 결합한 프로토콜(protocol)을 제안하고자 한다. 각 디스플레이 시스템에 항상 모든 데이터를 전송하는 것보다 현재화면 표시에 필요한 데이터만 전송할 경우에 데이터의 크기가 현저히 줄어들 수 있기 때문이다. 이 프로토콜의 성능을 분석하기 위해 영광 3,4호기 전 규모 시뮬레이터(Full Scope Simulator)에서 공정 데이터를 실시간으로 처리하고 디스플레이 할 수 있는 모의 제어실 정보 계통을 구성하고 기존의 주기적-전체전송 방식과 이벤트-Subscription 방식, 그리고 본 논문에서 제안한 주기적-Subscription 방식에 대하여 데이터 변화량과

클라이언트 수의 변화에 따른 네트워크 부하 지연시간을 측정 해 보았다.

2. 차세대 원전 정보처리 시스템

차세대 원전에서는 모두 디지털 제어시스템을 사용할 예정이므로, 기존의 주제어실 판넬상에 표시되어지던 모든 정보와 제어신호를 컴퓨터를 통해 처리하고 그 결과를 디스플레이에 나타내야 하기 때문에 정보처리계통(그림 1)에서는 많은 데이터를 처리해야하는 부담이 생긴다. 이때, 디스플레이 컴퓨터와 정보처리 컴퓨터는 네트워크로 연결되는데, 네트워크 과부하로 인한 데이터 전달 지연이 발생하지 않도록 하는 통신 프로토콜 설계가 요구되어진다.

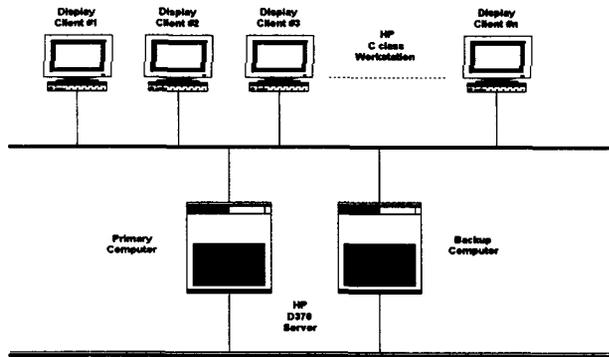


그림 1. 차세대원전 IPS prototype

3. Subscription Protocol

그림 1의 네트워크 구조에서 디스플레이 클라이언트가 차세대원전의 경우 20여개 정도 연결될 것으로 예상되고 이 경우 데이터 서버에서 각 클라이언트로 보내는 데이터의 전체 양도 상당한 분량이 될 것으로 예측되기 때문에, 데이터 전송을 효율적으로 처리하지 않으면 데이터 지연 시간이 늘어날 것으로 예측된다. 따라서 본 논문에서는 이와 같이 클라이언트의 수가 많아짐에 따라 네트워크의 부하를 줄일 수 있는 통신 프로토콜을 제안하고자 한다.

여기서 제안하고자 하는 Subscription 프로토콜은 일반적으로 흔히 사용되고 있는 모든 데이터의 주기적 전달 방식과는 다른, 디스플레이 클라이언트에서 필요로 하는 데이터만을 주기적(Periodic)으로 전송하여 데이터 전송량을 줄여 전체 네트워크의 효율을 높이는 방법이다. 하나의 CRT화면에 나타나는 데이터 수가 100개를 넘지 않는다고 가정하면 중앙 데이터베이스에서 클라이언트 디스플레이 시스템으로 모든 발전소 데이터를 전송하는 것보다 CRT화면에서 필요로 하는 데이터만을 전송하는 것이 훨씬 효율적이라는 것을 쉽게 예상할 수 있다.

그림 2는 본 논문에서 제안한 Subscription 프로토콜을 정보시스템 프로토타입 상에서 구현된 기능 블록도를 보여 준다. 먼저 네트워크로 연결된 컴퓨터사이의 통신 소프트웨어인 Software Bus[2]의 왼쪽은 CRT화면을 구동하는 클라이언트 컴퓨터 내부의 구조이고 오른쪽은 데이터베이스를 가지는 서버의 구조이다. 그리고 Software와 직접적으로 인터페이스 되는

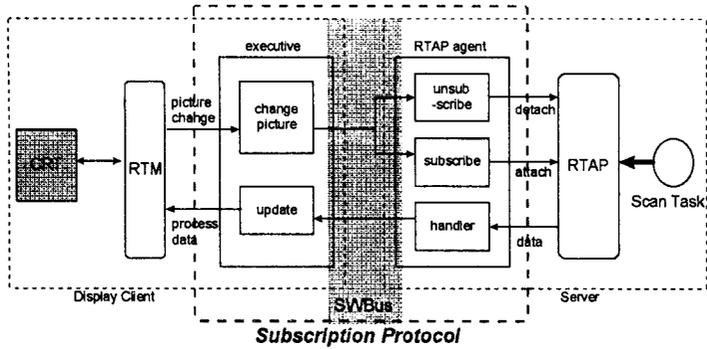


그림 2. 주기적 Subscription 프로토콜의 구현

서버측과 클라이언트측의 모듈이 Subscription 프로토콜을 구현한 예이며 중요한 5개의 함수로 구성되어 있다.

3.1 구현에 사용된 프로그램

제안된 Subscription 프로토콜 구현을 위해 사용된 그래픽사용자 인터페이스와 통신 소프트웨어, 데이터베이스 프로그램은 다음과 같다.

◆ RTM (run time manager)[1]

사용자 인터페이스의 핵심적인 프로그램이며 각종 화면데이터를 관리하고, 사용자로부터 입력을 받아 처리하며, 응용프로그램 인터페이스(API)를 통하여 외부 프로그램과 연결하여 MMI(Man-Machine Interface) 기능을 처리한다.

◆ SWBus (software bus)[2]

네트워크로 연결된 컴퓨터간의 통신을 위해 객체지향의 개념을 바탕으로 원격함수 호출, 원격객체 참조 등을 지역 객체 참조와 동일한 인터페이스로 행할 수 있는 고 수준의 라이브러리이다.

◆ RTAP (real time application platform)[3]

실시간으로 데이터를 수집하고 이를 데이터베이스화하며 각종 데이터베이스간의 연산이 가능하게하고 프로그래머 인터페이스를 바탕으로 응용프로그램을 연결하여 다른 프로그램에 실시간으로 데이터를 전달할 수 있는 환경이다.

◆ ScanTask[3]

RTAP에서 데이터를 수집하는 프로그램으로 공정변수를 주기적으로 읽어서 RTAP의 데이터베이스에 저장한다.

3.2 Subscription 프로토콜 주요 함수 및 기능

앞에서 언급한 것과 같이 Subscription 프로토콜을 구현하기 위하여 주요한 5개의 Software Bus 함수를 구현하였다. 이 Software Bus 함수들은 정보 계통 네트워크 상에서 임의의 호스트에서 동일하게 호출 가능하기 때문에 서버로부터 데이터를 전달받는 기능을 수행한다.

◆ `changePicture()`

CRT화면을 구동하는 RTM에서 호출하는 함수로써 네트워크를 통하여 서버측의 `unsubscribe()`와 `subscribe()`함수를 부름으로써 현재 각 화면당 등록된 변수를 바꾸어 주는 역할을 담당한다.

◆ `subscribe()`

화면 정보에 대한 공정변수 리스트를 구해내고 이 각각의 변수에 대하여 데이터베이스에 등록한다. 등록된 변수에 대해서는 주기적으로 외부 `handler` 함수가 데이터베이스로부터 현재 값을 읽어내 이를 등록된 클라이언트로 전송하게 된다.

◆ `unsubscribe()`

데이터 베이스에 등록되어 있으면 그 변수를 해제한다. 이 함수에서 고려해야 할 것은 복수 개의 클라이언트에서 동일 변수를 등록한 경우를 고려하여 해제하여야 한다. 이 기능을 위해서 각 변수 별로 `counter`를 두어 `subscribe()` 함수에서 등록할 경우 `counter`를 증가시키고 `unsubscribe()`와 마찬가지로 화면 정보에 대한 공정변수 리스트를 구해내고 이 각각의 변수가 `subscribe()` 함수에서 해제될 경우, `counter`를 감소시켜 이 값이 0이 될 경우에 한해서 실제로 데이터 베이스에 해제를 요구하게 된다.

◆ `handler()`

데이터베이스에 등록된 변수의 값을 주기적으로 읽어서 클라이언트측으로 보내는 함수이다.

◆ `update()`

서버측의 `handler()`에서 불러지며 등록된 변수의 값을 전송 받아 화면의 데이터를 갱신하는 함수이다.

3.3 데이터 전달 과정

디스플레이 클라이언트의 CRT 상에서 화면전환버튼 혹은 실행 버튼을 눌러 원하는 계통의 화면을 선택하거나 특정 기능을 수행시키게 되면, RTM은 연결된 통신 프로그램 내의 `changePicture()`라는 함수를 부르게 된다. 이 함수가 호출되면 먼저 보관하고 있던 과거 화면정보에 대하여 RTAP의 `unsubscribe()`라는 함수를 호출하여 현재 연결된 데이터들에 대한 접속을 끊는다. 한편, 이미 전환된 화면에 존재하는 데이터 값은 현재의 값이 갱신되어 있지 않기 때문에 운전원을 혼란시킬 우려가 있으므로, 주기적인 전달 방식과는 별도로 요구된 화면의 전체 데이터를 데이터베이스로부터 읽어내고 디스플레이 클라이언트로 보내어 현재 화면에 표시되고 있는 값을 현재 값으로 우선적으로 갱신한다. 이 과정이 끝나고 나면 현재 바뀌어진 화면정보에 대해 `subscribe()`라는 함수를 통해 새로운 화면에 대해서 데이터베이스 등록을 시도하게 된다. 이 일련의 과정을 통하여 화면전환이 완료된 후 RTAP은 데이터베이스에 등록된 변수의 값을 `handler()` 함수를 통하여 주기적으로 읽어들이며, 디스플레이 클라이언트의 `update()`함수를 호출하여 새로운 값들이 CRT에 나타나게 한다.

4. 프로토콜의 성능 분석

본 논문에서 제안한 프로토콜과 기존 방식간 비교를 위하여 발전소 정보 시스템의 프로토타입 상에서 주기적-전체전송 방식과 이벤트-subscription 방식[4], 그리고 주기적-subscription 방식을 사용한 전송 방식에 대한 성능을 비교 분석하였다. 실험에서는 전체 공정 데이터를 약 20,000개, 각 화면당 데이터는 100개로 추정하였다. 성능 측정의 지표로써 클라이언트의 갯수 증가에 따른 IPS와 CRT간 네트워크 통신 지연 시간을 측정하였고, 이벤트처리 방식에서 나타날 수 있는 비정상상태의 네트워크 전송 지연을 관찰할 수 있도록 데이터 변화량에 따른 통신 지연 시간을 동시에 측정하였다.

4.1 주기적-전체전송 방식

클라이언트에서 IPS에 있는 데이터 전달 함수를 호출함으로써 공정 데이터를 한꺼번에 블록화하여 전달받도록 되어 있다. 그 결과로 클라이언트의 수에 비례하여 통신 지연이 일어났고, 최대 32개의 클라이언트의 경우 50초 이상의 지연이 발생하였다. 이것은 서버측에서 각 클라이언트로 공정 데이터를 반복적으로 전달하는 과정에서 발생하는 누적된 지연시간임을 유추해 볼 수 있으며 실제 원자력발전소에 적용하기에는 비현실적인 값이다.

클라이언트 수	1	2	4	8	16	32
전송지연(sec)	1.51	3.13	6.34	12.9	25.6	56.2

4.2 이벤트-Subscription 방식

이벤트 방식의 성능을 측정하기 위하여 전체 공정 데이터를 10%, 50%, 100% 변화시키면서 전송 지연 시간을 측정하였다. 그 결과 데이터의 변화량이 30%일 경우에는 1초 이내의 전송지연 시간을 보였지만 변화량이 커질수록 시간이 증가되어 100%의 변화에서는 최대 3초의 지연을 보였다. 발전소 시뮬레이터에서 수집한 자료를 참조해 보면, 전체 공정 데이터 변화량이 정상 상태 하에서는 30%를 초과하지 않기 때문에, 정상 운전 상태의 경우에는 다른 어떤 전송방식 보다 실제 변화한 데이터의 표시 지연 시간이 적다고 볼 수 있다. 왜냐하면 주기적 방식은 실제 데이터의 값이 변화하더라도 1주기 후에 실제 전송이 일어나는 반면, 이벤트 방식은 데이터의 값 변화시 바로 통신이 일어나서 이 값이 클라이언트로 전달되기 때문이다. 하지만, 변화량이 50%를 초과하는 비정상상태라든가 정상 상태 하에서도 subscribe한 데이터들만 집중적으로 변할 때 지연시간이 증가됨을 피할 수 없게된다.(그림 3 참조)

4.2 주기적-Subscription 방식

이 방식으로 측정한 시간은 이벤트-subscription 방식에서 데이터의 변화량이 10%일 경우와 비슷한 지연시간을 보여준다. 실제 데이터 전달량만을 고려하였을 때, 이벤트-subscription 방식에서 100%변화할 경우와 동일 하지만, 지연시간이 그 보다 현격히 줄어든 것은 이벤트 방식에서 데

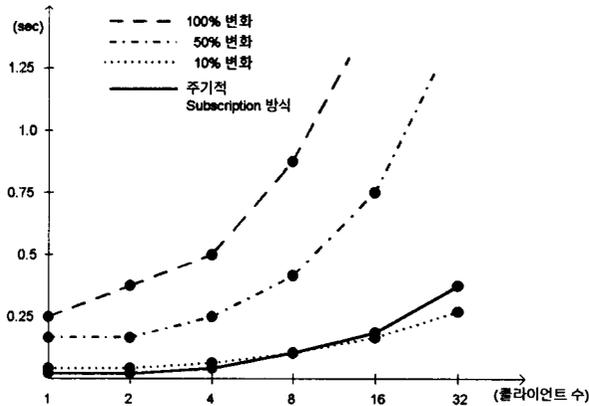


그림 3. Subscription 방식의 전송 지연

이터의 변화를 감지하기 위하여 데이터베이스 계산 과정이 없고, 데이터베이스 내에서 이벤트 처리 과정을 거치지 않기 때문이다. 따라서 이 프로토콜을 사용하였을 경우 주기적인 특성상으로 발생하는 전달 지연(2전송/1초)을 감안하더라도 최대 1초 이내의 전달 지연을 보임으로써 현실적으로 가장 적합한 전송 방식이라 볼 수 있다.

5. 결론

차세대 원전의 정보처리계통(IPS)은 발전소의 모든 정보를 처리하는 중요한 역할을 담당하고 있으며, 운전원에게 신속, 정확한 정보를 제공해야만 한다. 그러나 많은 양의 데이터를 처리하고 전달하는데 있어서 네트워크 부하에 따른 지연시간을 고려하지 않을 수 없다. 네트워크 부하 감소를 위한 방법으로 본 논문에서 소개한 주기적-Subscription 프로토콜은 이러한 네트워크 부하의 감소뿐만 아니라, 무엇보다도 지연시간에 대한 신뢰도를 향상시킬 수 있는 한 방법임을 보였으며, 이 프로토콜을 정보처리계통의 프로토타입에 적용한 결과 상당히 만족된 결과를 얻었다. 나아가 이러한 결과를 바탕으로 Subscription 프로토콜에 대한 보완과 검증을 거친다면 차세대 원전 정보처리계통에 적합한 프로토콜을 선정하는데 도움이 될 수 있을 것이라 생각된다.

참고문헌

- [1] OECD Halden Reactor Project, "The Picasso-3 Reference Manual", Revision R1.4, March 1995
- [2] OECD Halden Reactor Project, "The Software Bus Communication System", Revision R1.0, May 1996
- [3] HEWLETT PACKARD, "RTAP/Plus Programming", Release A.06.60, January 1995
- [4] 성찬호, 정봉준, "원전 주제어실 Network 부하 감소를 위한 Protocol 개발", 한국자동제어학술회의, 1997