



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년11월28일
(11) 등록번호 10-1205518
(24) 등록일자 2012년11월21일

(51) 국제특허분류(Int. Cl.)
H04L 12/28 (2006.01) H04L 29/10 (2006.01)
(21) 출원번호 10-2007-7004806
(22) 출원일자(국제) 2005년07월25일
심사청구일자 2010년07월26일
(85) 번역문제출일자 2007년02월27일
(65) 공개번호 10-2007-0072856
(43) 공개일자 2007년07월06일
(86) 국제출원번호 PCT/US2005/026375
(87) 국제공개번호 WO 2006/014888
국제공개일자 2006년02월09일
(30) 우선권주장
10/948,085 2004년09월23일 미국(US)
60/591,286 2004년07월27일 미국(US)
(56) 선행기술조사문헌
US20020156613 A1*
US20030145117 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
엠케이에스 인스트루먼트 인코포레이티드
미국 매사추세츠 01810 앤도버 수트 201 테크 드
라이브 2
(72) 발명자
카릴 니달
미국 캘리포니아 95128 산호세 아파트 3 사우쓰
윈체스터블르바드 1042
시프나호 쿼터
미국 캘리포니아 95054 산타클라라 #322 칼라일
코트 4513
(뒷면에 계속)
(74) 대리인
박장원

전체 청구항 수 : 총 40 항

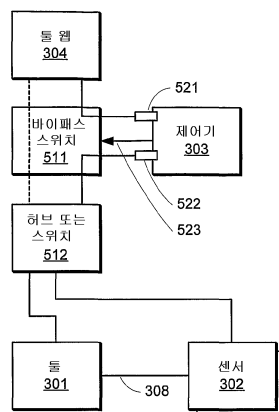
심사관 : 정은선

(54) 발명의 명칭 지능형 제어 디바이스 및 지능형 제어 방법의 오작동 방지스위칭

(57) 요약

본 발명은 톨과 관련된 센서로부터의 데이터 수집과 제어에 대한 것이다. 좀더 상세하게는, 본 발명은 톨, 상기 톨과 관련된 센서들 및 가령 호스트 시스템 또는 분산된 프로세서와 같은 데이터 사용자들 사이에서 통신을 중재하는 제어기의 사용에 관한 것이다. 본 발명의 특별한 양상은 청구항들, 발명의 상세한 설명 및 도면에 기재된다.

대표도 - 도5



(72) 발명자

레브-아미 우지

미국 캘리포니아 94040 마운틴 뷰 아파트 #502 웨
스트 엘 카미노리얼 2400

하다르 론

미국 캘리포니아 95014 쿠파티노 투라 레인 10337

특허청구의 범위

청구항 1

중간 매개 디바이스가 고장난 경우에 제 1 디바이스와 제 2 디바이스 사이에서 투명하게(transparently) 연결을 설정하는 방법으로서,

지능형 제어기가, 상기 제 1 및 제 2 디바이스를 상기 중간 매개 디바이스의 제 1 및 제 2 네트워크 인터페이스에 연결하는 단계와, 여기서 상기 제 1 및 제 2 네트워크 인터페이스는 상기 제 1 디바이스 또는 제 2 디바이스에 관련된 데이터 링크 계층 식별자와는 구별되는 데이터 링크 계층 식별자를 가지며;

정상 동작 모드(normal operating mode)에서, 상기 중간 매개 디바이스가 자신의 연속적인 안정 동작(healthy operation)을 오작동 방지 스위치에 반복적으로 시그널링하는 동안 상기 제 1 디바이스와 상기 제 2 디바이스 사이에서 어드레스된 메시지들을 처리하는 단계와, 여기서 상기 처리된 메시지들은 상기 제 1 및 제 2 디바이스들중 일 디바이스에게는 상기 제 1 및 제 2 디바이스들중 타 디바이스로부터 온 것으로 보임으로써 오작동 방지 모드로의 스위칭에 의해 상기 일 디바이스와 상기 타 디바이스 사이에서 네트워크 세션이 재시작을 필요로 함이 없이 계속되게 하며; 그리고

상기 오작동 방지 스위치가, 상기 중간 매개 디바이스로부터의 안정 동작의 반복되는 시그널링의 중단시 상기 중간 매개 디바이스를 바이패스하는 데이터 연결에 의해 상기 제 1 및 제 2 디바이스를 직접 연결함으로써 오작동 방지 모드로 스위칭하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 2

제 1 항에 있어서,

상기 오작동 방지 스위치를 오작동 방지 모드로 스위칭하는 단계는 가청 신호를 생성하는 것을 포함하는 것을 특징으로 하는 방법.

청구항 3

제 2 항에 있어서,

상기 가청 신호는 릴레이 클릭(relay click)인 것을 특징으로 하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 1 및 제 2 네트워크 인터페이스는 수신한 모든 패킷들을, 상기 중간 매개 디바이스에 어드레스된 패킷들을 처리를 위해 보유하는 필터에 전달하도록 된 것을 특징으로 하는 방법.

청구항 5

제 1 항에 있어서,

상기 메시지들은 패킷들이며,

상기 중간 매개 디바이스에 의해 처리되는 메시지들을 처리하는 단계는:

적어도 일부 패킷들에 대해서, 상기 제 1 디바이스 또는 제 2 디바이스로부터의 패킷들을 제 2 디바이스 또는 제 1 디바이스로 중계(relay)하는 것과; 그리고

상기 중계된 패킷들의 소스 식별 필드 내에서 제 1 또는 제 2 네트워크 인터페이스의 데이터 링크 식별자 대신에 실제 소스의 데이터 링크 식별자를 특정하는 것을 포함하는 것을 특징으로 하는 방법.

청구항 6

제1항에 있어서,

상기 연속적인 안정 동작을 반복적으로 시그널링하는 것은, 상기 오작동 방지 스위치에 메시지를 전송하는 중간 매개 디바이스 상에서 작동하는 데몬에 의해 구현되는 것을 특징으로 하는 방법.

청구항 7

제1항에 있어서,

상기 연속적인 안정 동작을 반복적으로 시그널링하는 것은, 상기 오작동 방지 스위치로부터의 조회들(inquiries)을 경청하고 상기 조회들에 응답하는 중간 매개 디바이스 상에서 작동하는 데몬에 의해 구현되는 것을 특징으로 하는 방법.

청구항 8

제 1 항에 있어서,

하나의 물리적인 통신 링크들을 사용하여 상기 제 1 디바이스에 복수개의 제 2 디바이스들을 연결하는 것을 더 포함하는 것을 특징으로 하는 방법.

청구항 9

제 8 항에 있어서,

상기 하나의 물리적인 통신 링크는, 이더넷 세그먼트인 것을 특징으로 하는 방법.

청구항 10

제 1 항에 있어서,

2개 이상의 물리적인 통신 링크들을 사용하여 상기 제 1 디바이스에 복수개의 제 2 디바이스들을 연결하는 것을 더 포함하는 것을 특징으로 하는 연결을 설정하는 방법.

청구항 11

제 10 항에 있어서,

상기 물리적인 통신 링크들중 하나는 이더넷 세그먼트이며 다른 하나는 직렬 연결인 것을 특징으로 하는 방법.

청구항 12

제 11 항에 있어서,

상기 직렬 연결은, RS-232 또는 RS-422 기준을 따르는 것을 특징으로 하는 방법.

청구항 13

제 1 항에 있어서,

상기 제 1 디바이스와 제 2 디바이스 사이에 직접 통신 세션을 설정하는 것은, 상기 제 1 또는 제 2 디바이스에서 네트워크 어드레스들과 데이터 링크 계층 어드레스들을 상관시키는 임의의 테이블들의 갱신을 요하지 않는 것을 특징으로 하는 방법.

청구항 14

제 1 항에 있어서,

상기 제 1 디바이스와, 제 2 디바이스 및 상기 중간 매개 디바이스의 네트워크 어드레스들 모두는 동일한 네트워크 서브네트 내에 존재하는 것을 특징으로 하는 방법.

청구항 15

제 1 항에 있어서,

상기 제 1 디바이스와 제 2 디바이스 사이에서 직접 통신 세션을 설정하는 것은, 상기 제 1 또는 제 2 디바이스에서 IP 어드레스들과 MAC 어드레스들을 상관시키는 임의의 테이블들의 갱신을 필요치 않는 것을 특징으로 하는 연결을 설정하는 방법.

청구항 16

호스트와 툴 사이에 접속된 지능형 제어기의 고장 후에 상기 호스트와 상기 툴 사이에 통신 링크를 설정하는 페일-오버 방법으로서,

상기 툴에 의해 사용되는 제 1 하드웨어 레벨 식별자를 상기 지능형 제어기의 제 1 포트에서 채택하고, 상기 제 1 포트가 상기 제 1 하드웨어 레벨 식별자를 이용하여 어드레스됨을 상기 호스트에게 공표(advertizing)하는 단계와;

상기 호스트에 의해 사용되는 제 2 하드웨어 레벨 식별자를 상기 지능형 제어기의 제 2 포트에서 채택하고, 상기 제 2 포트가 상기 제 2 하드웨어 레벨 식별자를 이용하여 어드레스됨을 상기 툴에게 공표하는 단계와;

상기 호스트 및 툴이 직접적인 공표 통신을 하지 않도록, 상기 지능형 제어기에서 상기 제 1 및 제 2 하드웨어 레벨 식별자들의 공표들(advertisements)을 가로채는 단계와;

상기 지능형 제어기가 신뢰되어야 함을 오작동 방지 스위치로 반복적으로 시그널링하는 단계를 포함하며, 상기 오작동 방지 스위치는 상기 반복적인 시그널링이 실패할 경우 상기 지능형 제어기를 바이패스하는 통신 링크에 의해 상기 호스트 및 상기 툴을 직접적으로 연결하도록 적응되며, 이에 의해 상기 호스트 및 툴은 직접 공표 통신하게 되고 상기 제 1 및 제 2 하드웨어 레벨 식별자들의 공표들은 상기 호스트와 상기 툴 사이의 통신 링크의 설정을 용이하게 하는 것을 특징으로 하는 페일-오버 방법.

청구항 17

제 16 항에 있어서,

상기 제 1 포트를 식별하는 제 1 논리적 IP 어드레스, 상기 툴에 의해 사용되는 제 2 논리적 IP 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 IP 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 IP 어드레스가 모두 서로 다른 것을 특징으로 하는 페일-오버 방법.

청구항 18

제 16 항에 있어서,

상기 제 1 포트를 식별하는 제 1 논리적 네트워크 계층 어드레스, 상기 툴에 의해 사용되는 제 2 논리적 네트워크 계층 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 네트워크 계층 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 네트워크 계층 어드레스가 모두 서로 다른 것을 특징으로 하는 페일-오버 방법.

청구항 19

제 16 항에 있어서,

상기 하드웨어 레벨 식별자는 MAC 어드레스인 것을 특징으로 하는 페일-오버 방법.

청구항 20

제 16 항에 있어서,

상기 하드웨어 레벨 식별자들의 공표들은 표준에 부합하는 ARP 메시지들인 것을 특징으로 하는 페일-오버 방법.

청구항 21

제 16 항에 있어서,

상기 페일-오버 방법은 상기 툴을 모니터링하는 센서에 더 적용되고, 상기 지능형 제어기는 상기 호스트와 상기 센서 사이에 접속되며,

상기 페일-오버 방법은:

상기 지능형 제어기의 제 1 포트에서 상기 센서에 의해 사용되는 제 3 하드웨어 레벨 식별자를 채택하고, 상기 호스트에게 상기 제 1 포트가 상기 제 3 하드웨어 레벨 식별자를 이용하여 어드레스됨을 공표하는 단계와;

상기 센서에게 상기 제 2 포트가 상기 제 2 하드웨어 레벨 식별자를 이용하여 어드레스됨을 공표하는 단계와;

그리고

상기 호스트 및 센서가 직접적인 공표 통신을 하지 않도록, 상기 지능형 제어기에서 상기 제 3 하드웨어 레벨 식별자의 공표들을 가로채는 단계를 포함하여 이루어지며,

상기 반복적인 시그널링이 실패할 경우에 상기 오작동 방지 스위치는 상기 호스트와 상기 센서 사이에서 상기 지능형 제어기를 바이패스하는 직접 통신 링크를 설정하는 것을 특징으로 하는 페일-오버 방법.

청구항 22

제 21 항에 있어서,

상기 제 1 포트를 식별하는 제 1 논리적 IP 어드레스, 상기 톨에 의해 사용되는 제 2 논리적 IP 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 IP 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 IP 어드레스, 상기 센서에 의해 사용되는 제 5 논리적 IP 어드레스가 모두 서로 다른 것을 특징으로 하는 페일-오버 방법.

청구항 23

제 21 항에 있어서,

상기 제 1 포트를 식별하는 제 1 논리적 네트워크 계층 어드레스, 상기 톨에 의해 사용되는 제 2 논리적 네트워크 계층 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 네트워크 계층 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 네트워크 계층 어드레스, 상기 센서에 의해 사용되는 제 5 논리적 네트워크 계층 어드레스가 모두 서로 다른 것을 특징으로 하는 페일-오버 방법.

청구항 24

제 21 항에 있어서,

상기 하드웨어 레벨 식별자는 MAC 어드레스인 것을 특징으로 하는 페일-오버 방법.

청구항 25

제 21 항에 있어서,

상기 하드웨어 레벨 식별자들의 공표들은 표준에 부합하는 ARP 메시지들인 것을 특징으로 하는 페일-오버 방법.

청구항 26

페일-오버 방지 지능형 제어기와 오작동 방지 스위치를, 적어도 하나의 톨과 접속된 호스트 사이의 통신링크들에 도입하는 방법으로서,

상기 오작동 방지 스위치를 상기 호스트, 상기 지능형 제어기 및 상기 톨에 접속하는 단계를 포함하며, 여기서 상기 오작동 방지 스위치가 제 1 상태에 있을 때 상기 호스트 및 상기 톨은 상기 지능형 제어기와 통신하고 상기 호스트 및 상기 톨은 직접적으로 서로 통신하지 않으며, 상기 오작동 방지 스위치가 제 2 상태에 있을 때 상기 호스트 및 상기 톨은 직접적으로 서로 통신하며,

상기 제 1 상태에서, 상기 지능형 제어기는,

상기 호스트와 상기 톨 사이의 메시지들을 가로채고,

상기 가로챈 메시지들내에서, 적어도 하나의 하드웨어 레벨 소스 식별자와 적어도 하나의 하드웨어 레벨 목적지 식별자를 수정하고,

상기 수정된 하드웨어 레벨 식별자들에 대응하도록 상기 가로챈 메시지들 내에서 적어도 하나의 에러 보정 코드를 수정하고, 그리고

수정된 하드웨어 레벨 식별자들 및 수정된 보정 코드를 갖는 상기 가로챈 메시지들을 중계하며; 그리고

상기 제 2 상태에서, 상기 오작동 방지 스위치는 상기 지능형 제어기를 바이패스시키며, 이에 의해 상기 호스트가 상기 톨과 직접적으로 통신하는 것을 특징으로 하는 방법.

청구항 27

제 26 항에 있어서,

상기 지능형 제어기는 적어도 하나의 제 1 포트 및 제 2 포트를 포함하며,

상기 제 1 포트는 상기 호스트와 통신하도록 접속되고 상기 제 2 포트는 상기 툴과 통신하도록 접속되며; 그리고

상기 제 1 포트를 식별하는 제 1 논리적 IP 어드레스, 상기 툴에 의해 사용되는 제 2 논리적 IP 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 IP 어드레스, 및 상기 호스트에 의해 사용되는 제 4 논리적 IP 어드레스 모두는 서로 다른 것을 특징으로 하는 방법.

청구항 28

제 26 항에 있어서,

상기 지능형 제어기는 적어도 하나의 제 1 포트 및 제 2 포트를 포함하며,

상기 제 1 포트는 상기 스위치를 통해 상기 호스트와 통신하도록 접속되고 상기 제 2 포트는 상기 스위치를 통해 상기 툴과 통신하도록 접속되며; 그리고

상기 제 1 포트를 식별하는 제 1 논리적 네트워크 계층 어드레스, 상기 툴에 의해 사용되는 제 2 논리적 네트워크 계층 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 네트워크 계층 어드레스, 및 상기 호스트에 의해 사용되는 제 4 논리적 네트워크 계층 어드레스 모두는 서로 다른 것을 특징으로 하는 방법.

청구항 29

제 26 항에 있어서,

상기 하드웨어 레벨 식별자는 MAC 어드레스인 것을 특징으로 하는 방법.

청구항 30

제 26 항에 있어서,

상기 방법은 상기 툴을 모니터링하는 적어도 하나의 센서에 더 적용되며,

상기 오작동 방지 스위치를 상기 센서에 접속하는 단계 - 상기 오작동 방지 스위치가 제 1 상태에 있을 때 상기 호스트 및 상기 센서는 상기 지능형 제어기와 통신하고 상기 호스트 및 상기 센서는 직접적으로 서로 통신하지 않으며, 상기 오작동 방지 스위치가 제 2 상태에 있을 때 상기 호스트 및 상기 센서는 직접적으로 서로 통신함 - 을 더 포함하여 이루어지며,

상기 제 1 상태에서, 상기 지능형 제어기는,

상기 호스트와 상기 센서 사이의 메시지들을 가로채고,

상기 가로챈 메시지들내에서, 적어도 하나의 하드웨어 레벨 소스 식별자와 적어도 하나의 하드웨어 레벨 목적지 식별자를 수정하고,

상기 수정된 하드웨어 레벨 식별자들에 대응하도록 상기 가로챈 메시지들 내에서 적어도 하나의 에러 보정 코드를 수정하고, 그리고

수정된 하드웨어 레벨 식별자들 및 수정된 보정 코드를 이용하여 상기 가로챈 메시지들을 중계하며; 그리고

상기 제 2 상태에서, 상기 오작동 방지 스위치는 상기 지능형 제어기를 바이패스시키며, 이에 의해 상기 호스트가 상기 센서와 직접적으로 통신하는 것을 특징으로 하는 방법.

청구항 31

제 30 항에 있어서,

상기 지능형 제어기는 적어도 하나의 제 1 포트 및 제 2 포트를 포함하며, 상기 제 1 포트는 상기 오작동 방지 스위치를 통해 상기 호스트와 통신하도록 접속되고 상기 제 2 포트는 상기 오작동 방지 스위치를 통해 상기 툴 및 상기 센서와 통신하도록 접속되며; 그리고

상기 제 1 포트를 식별하는 제 1 논리적 IP 어드레스, 상기 틀에 의해 사용되는 제 2 논리적 IP 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 IP 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 IP 어드레스 및 상기 센서에 의해 사용되는 제 5 논리적 IP 어드레스 모두는 서로 다른 것을 특징으로 하는 방법.

청구항 32

제 30 항에 있어서,

상기 지능형 제어기는 적어도 하나의 제 1 포트 및 제 2 포트를 포함하며, 상기 제 1 포트는 상기 오작동 방지 스위치를 통해 상기 호스트와 통신하도록 접속되고 상기 제 2 포트는 상기 오작동 방지 스위치를 통해 상기 틀 및 상기 센서와 통신하도록 접속되며; 그리고

상기 제 1 포트를 식별하는 제 1 논리적 네트워크 계층 어드레스, 상기 틀에 의해 사용되는 제 2 논리적 네트워크 계층 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 네트워크 계층 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 네트워크 계층 어드레스 및 상기 센서에 의해 사용되는 제 5 논리적 네트워크 계층 어드레스 모두는 서로 다른 것을 특징으로 하는 방법.

청구항 33

제 30 항에 있어서,

상기 지능형 제어기는 적어도 하나의 제 1 포트, 제 2 포트 및 제 3 포트를 포함하며, 상기 제 1 포트는 상기 오작동 방지 스위치를 통해 상기 호스트와 통신하도록 접속되고 상기 제 2 포트는 상기 오작동 방지 스위치를 통해 상기 틀과 통신하도록 접속되며 상기 제 3 포트는 상기 오작동 방지 스위치를 통해 상기 센서와 통신하도록 접속되며; 그리고

상기 제 1 포트를 식별하는 제 1 논리적 IP 어드레스, 상기 틀에 의해 사용되는 제 2 논리적 IP 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 IP 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 IP 어드레스 및 상기 센서에 의해 사용되는 제 5 논리적 IP 어드레스 모두는 서로 다른 것을 특징으로 하는 방법.

청구항 34

제 30 항에 있어서,

상기 지능형 제어기는 적어도 하나의 제 1 포트, 제 2 포트 및 제 3 포트를 포함하며, 상기 제 1 포트는 상기 오작동 방지 스위치를 통해 상기 호스트와 통신하도록 접속되고 상기 제 2 포트는 상기 오작동 방지 스위치를 통해 상기 틀과 통신하도록 접속되며 상기 제 3 포트는 상기 오작동 방지 스위치를 통해 상기 센서와 통신하도록 접속되며; 그리고

상기 제 1 포트를 식별하는 제 1 논리적 네트워크 계층 어드레스, 상기 틀에 의해 사용되는 제 2 논리적 네트워크 계층 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 네트워크 계층 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 네트워크 계층 어드레스 및 상기 센서에 의해 사용되는 제 5 논리적 네트워크 계층 어드레스 모두는 서로 다른 것을 특징으로 하는 방법.

청구항 35

중간 매개 디바이스가 고장난 경우에 제 1 디바이스와 제 2 디바이스 사이에서 직접 연결을 보장하도록된 오작동 방지 스위치를 갖는 오작동 방지 시스템으로서,

상기 제 1 및 제 2 디바이스에 각각 링크되어 통신하는 제 1 및 제 2 네트워크 인터페이스를 갖는 중간 매개 디바이스와, 상기 제 1 및 제 2 네트워크 인터페이스는 상기 제 1 디바이스 또는 제 2 디바이스에 관련되는 임의의 데이터 링크 계층식별자들과는 구별되는 데이터 링크 계층 식별자를 가지며;

상기 중간 매개 디바이스와 통신하며 우발적으로(contingently) 상기 제 1 및 제 2 디바이스와 링크되어 통신하는 오작동 방지 스위치와, 상기 오작동 방지 스위치는 상기 제 1 및 제 2 디바이스가 상기 제 1 및 제 2 네트워크 인터페이스와 통신하는 정상 동작 모드에 적응하며 상기 중간 매개 디바이스를 바이패스하여 상기 제 1 및 제 2 디바이스가 직접 통신하는 오작동 방지 모드에 적응하며;

상기 제 1 및 제 2 네트워크 인터페이스들중 일 네트워크 인터페이스로부터, 상기 제 1 및 제 2 디바이스들중

일 디바이스로부터 패킷들을 수신하고 상기 패킷들을 상기 제 1 및 제 2 네트워크 인터페이스들중 타 네트워크 인터페이스를 통해 상기 제 1 및 제 2 디바이스들중 타 디바이스에 중계하고 그리고 상기 중간 매개 디바이스의 안정 동작을 상기 오작동 방지 스위치로 반복적으로 통신하도록 된, 상기 중간 매개 디바이스 상의 로직 및 리소스들과; 그리고

오작동 방지 모드로 스위칭함으로써 상기 중간 매개 디바이스로부터의 안정 동작에 대한 상기 반복된 신호의 중단에 대해 응답하도록 된, 상기 오작동 방지 스위치 상의 로직 및 리소스들을 포함하여 구성되며,

상기 중계된 패킷들은 상기 일 디바이스에게는 상기 타 디바이스로부터 온 것으로 보임으로써 오작동 방지 모드로의 스위칭에 의해 상기 일 디바이스와 상기 타 디바이스 사이에서 네트워크 세션이 재시작을 필요로 함이 없이 계속되게 하는 것을 특징으로 하는 오작동 방지 시스템.

청구항 36

제 35 항에 있어서,

상기 오작동 방지 스위치를 오작동 방지 모드로 스위칭하는 것은 가청 신호를 생성하는 것을 더 포함하는 것을 특징으로 하는 오작동 방지 시스템.

청구항 37

제 36 항에 있어서,

상기 가청 신호는 릴레이 클릭인 것을 특징으로 하는 오작동 방지 시스템.

청구항 38

제 35 항에 있어서,

상기 중간 매개 디바이스 상의 로직 및 리소스들은 상기 중간 매개 디바이스로 어드레스된 패킷들을 처리하기 위해 존속하는 필터를 더 포함하며, 상기 필터는 상기 제 1 및 제 2 네트워크 인터페이스와 접속하는 것을 특징으로 하는 오작동 방지 시스템.

청구항 39

제 35 항에 있어서,

상기 중간 매개 디바이스 상의 로직 및 리소스들은 상기 오작동 방지 스위치로 메시지를 전송하는 상기 중간 매개 디바이스 상에서 구동되는 데몬을 더 포함하는 것을 특징으로 하는 오작동 방지 시스템.

청구항 40

제 35 항에 있어서,

상기 중간 매개 디바이스 상의 로직 및 리소스들은 오작동 방지 스위치로부터의 조회를 요청하고, 상기 조회에 응답하는 중간 매개 디바이스 상에서 구동되는 데몬을 더 포함하는 것을 특징으로 하는 오작동 방지 시스템.

명세서

기술 분야

[0001]

본 발명은, 제어기 호스트(또는 제 1 디바이스)와 툴(또는 제 2 디바이스) 사이에 도입된 제어기의 오작동이 방지된(failsafe) 동작에 관한 것이다. 특히, 본 발명은 제 1 및 제 2 디바이스 사이에서 데이터 전달을 중재하는 제어기의 고장시에 제 1 및 제 2 디바이스 사이에 직접 연결을 설정하는 것에 관한 것이다. 본 발명이 유용하게 적용될 수 있는 환경 중의 하나는 호스트 및 툴들/센서들을 구비한 반도체 제조 팹(fab)으로서, 툴, 상기 툴과 관련된 센서들 및 데이터 사용자들(가령, 호스트 시스템 또는 분산된 프로세서) 사이에서 통신을 중재하는 제어기를 사용함으로써, 반도체 제조 팹 사이에 지능형 제어기가 부가된다.

배경 기술

[0002]

무어의 법칙(Moore's)에 따르면 가격이 하락함에도 컴퓨터 성능의 지수적인 성장이 예고되고 있다. 프로세싱 능

력의 이와같은 역동적인 성장은, 자칫 혹자로 하여금 반도체 디바이스 제조가 석유를 탐사하는 것(wild-catting for oil)과 유사한 모험이 가득한 사업이라고 생각하게 만들 수도 있다. 하지만, 사실은 정 반대이다. 왜냐하면, 반도체 제조 장비들은 매우 고가이며 그리고 제조 공정들은 미세한 오류에도 매우 민감하기 때문에, 반도체 디바이스 제조는 매우 조심스러운 사업이다. 새로운 장비에 대한 자격 싸이클(qualification cycles)들 및 기준들과 구형 장비에 대한 변경은 매우 시간이 걸리며 또한 필요한 일이다. 제조에 투입되기 전에, 심지어 아주 사소한 변화에 대해서도 광범위하게 검토된다.

[0003] 반도체 제조의 랩에서 사용되는 중요한 구성요소들은, 튜들[가령, 증착 챔버들, 리액터들(reactors)], 상기 튜들을 모니터링하는 센서들[가령, FTIR 센서들, 매스 스펙트로그래프(mass spectrographs), 써모커플(thermocouples)] 및 상기 튜의 동작에 대한 센서들로부터의 데이터를 저장하며 분석하는 호스트들 또는 분산된 프로세서들을 포함한다.

[0004] 본원의 선출원에서는 센서들로부터 데이터를 청취(listening)하고 가령, 인터넷상의 TCP/IP 과 같이 고속이며 에러에 강한 기술들을 사용하여 이를 호스트들 또는 분산된 프로세서들로 제공하는 명백한 방법이 개시되어 있다. 상기 선출원은 Uzi Lev-Ami 와 Yossef Ilan Reich 에 의해 발명되어 "Method and Apparatus for Monitoring Host to Tool Communication" 라는 명칭으로 2001년 8월 22일자로 미국에 출원되었으며(출원번호 09/935,213), 이는 본 명세서에 참조로 편입된다. 상기 선출원에는 광학적으로 분리된 커넥터(connector)를 사용하여 튜 또는 센서로부터의 직렬(serial) 통신을 도청할 수 있는 리스닝 포트(listening post)에 대해 개시되어 있다. 상기의 도청 방식을 사용함으로써, 튜들 또는 센서들에 대한 변경을 요구하지 않는 낮은 위험부담하에서 랩 통신들(fab communications) 및 데이터 수집 인프라 스트럭처가 업그레이드(upgrade)될 수 있다. 상기 업그레이드 실행 가능성은 현재 사용중인 통신 인프라 스트럭처를 해체하지 않고도 증명될 수 있다.

[0005] 랩 기구류 및 백엔드 분석 능력(backend analysis capabilities)에서의 다음 변형은, 센서들의 분석 특성들을 변화시키거나 대체하지 않고도, 제어기들의 한쪽 편인 튜들 및 센서들 사이와 제어기들의 다른 한쪽 편인 튜 호스트들(또는 분산된 프로세서들) 사이의 중재(mediate) 통신에 지능형 제어기들을 추가하는 것을 포함할 것이다. 증가된 프로세서 성능과 감소된 저장소자의 가격으로 인해, 종래의 랩 환경에서는 실제적이지 않았던 구성들이 가능해지고 있다. Uzi Lev-Ami, Guenter Sifnatsch 및 Mark Attwood 에 의해 발명되어 2004년 4월 7일자로 "Controller and Method to Mediate Data Collection from smart Sensor for Fab Application"라는 명칭으로 미국에 출원된 2번째 선출원(출원번호 10/819,903)에서는 다양한 능력들을 갖는 지능형 제어기가 개시되어 있다.

[0006] 중간 매개 디바이스들이 고장나거나 또는 정지된다면, 서로 다른 디바이스들을 직접적으로 링크하는 오작동 방지된 스위치를 이용하여, 지능형 제어기의 분석적 피쳐들을 보완하거나 또는 직접적으로 통신가능한 2개의 서로 다른 디바이스 사이에서 통신을 중재하는 임의의 다른 네트워크가 부착된 디바이스를 보완하는 기회가 발생한다. 좀더 우수하며, 좀더 신뢰성 있고 좀더 확실있는 중간 매개 디바이스의 설치가 필요할 수도 있다.

발명의 상세한 설명

[0007] 본 발명은 튜들과 관련된 센서들로부터의 데이터 수집 및 제어에 관한 것이다. 특히, 본 발명은 튜, 상기 튜와 관련된 센서들 및 데이터 사용자들(가령, 호스트 시스템 또는 분산된 프로세서들) 사이에서 통신을 중재하기 위하여 제어기를 사용하는 것에 관한 것이다. 본 발명의 특정 양상들은 청구항들, 명세서 및 도면에 개시되어 있다.

실시예

[0014] 아래의 상세한 서술들은 도면들을 참조하여 설명된다. 본 발명을 예시하기 위해 바람직한 실시예가 기술될 것이며, 이는 청구항들에 의해 정의되는 본 발명의 범위를 제한하지 않는다. 당 업계의 당업자들은 아래에서 설명될 기술내용에 의해 다양한 동등한 변형물들을 식별할 수 있을 것이다.

도입부

[0016] 반도체 랩들 및 파운드리 업체들(foundries)의 튜들 및 다른 자동화된 장비 또는 반-자동 장비에 의해 사용되는 SECS 메시지 프로토콜들, 통신 인프라스트럭처 및 호스팅 모드들은, 통신 속도와 프로세서 속도가 상대적으로 제한되었던 수년전 부터 개발되어 왔다. 랩 어플리케이션을 위한 SECS 메시지 프로토콜들은 낮은 속도의 직렬 통신들(serial communications)을 이용하도록 설계되었다. 이러한 메시지 프로토콜들은 구조적 메시지들(structured messages)을 포함하고 있는바, 이들 구조적 메시지들은 저 속도 통신들에서도 빠르게 전달될 수 있

다. 구조적 메시지들은 과거에나 지금에나 번역(translate)되기 어려우며 이해하기도 어렵다. 이러한 어려움은, 첫번째 메시지가 응답을 위한 문맥(context)을 세팅하며 응답 메시지인 두번째 메시지가 상기 문맥(context)을 반복하지 않는 때에 더욱 가중된다. 즉, 문맥에 민감한 응답(context-sensitive response)은 상응하는 문맥 설정 메시지(context-setting message)와 짝을 이룰 때에만 의미가 있다. 통신들은 전형적으로, RS 232 또는 이와 동등한 직렬 통신들을 통해, 단말들 및 시분할(time-sharing) 컴퓨터들에 대해 모델들 및 전화선들과 유사한 전용 채널들을 따라 이루어진다. 호스트 시스템들은 메인 프레임들(mainframes), 미니 컴퓨터들, 또는 워크 스테이션들 상에서 작동한다. 호스트 시스템들은 전형적으로 단일체(monolithic)이며, 랩 내의 모든 툴들 또는 중요 툴들의 세트를 모니터링 및 제어한다. 호스트 시스템들은 툴들 및 센서들과 인터페이스하기 위해 어댑터들(adapters)에 의존한다. 호스트 시스템들은 툴들 및 센서들로부터 데이터를 수신하며 상기 툴들로 제어명령들을 발행하는 것이 일반적이다. 또한, 호스트 시스템들은 종종 상당한 분량의 통신 메시지들을 수신 및 생성한다.

[0017] '툴 호스트' 라는 용어는 툴 제어 호스트들 및 좀더 제한되거나 또는 좀더 유연성 있는 분산된 프로세서들을 모두 포함하는 넓은 범위의 의미로 사용된다. 툴 호스트들은, 포괄적이며 통합된 툴 제어 기능들을 갖는 호스트들 및 분산된 프로세서들에서 작동하고 좀더 제한되며 작업에 특정화된 기능들을 갖는 호스트들을 포함한다. 툴 호스트들은 Consilium 사의 FAB 300(TM) 소프트웨어와 같은 제품들을 포함하며, 이는 고객별로 특정한 비즈니스 프로세스들의 중앙집중화된 정의에 의해 구동되는 단일의 포괄적인 공장 제어 시스템을 제공한다. 이러한 카테고리 고리의 툴 호스트들은, 각기 다른 벤더(vendors)들로부터 공급된 툴들을 제어하도록 설계된 전통적인 생산 수행 시스템들을 대체하도록 설계되었다. 전통적인 생산 수행 시스템들로부터 툴 호스트의 스펙트럼(spectrum)의 반대편 끝단에서, 콤포넌트 프로세스들은, 포괄적인 관리 시스템들을 필요로 하지 않고 다양한 특정한 기능들을 처리하도록 분산된 프로세서들 상에서 작동할 수도 있다. 스펙트럼을 따라, Consilium 사의 FAB 300(TM) 소프트웨어와 같은 제품은, 소정의 목적들(purposes)에 대해서는 툴 제어 호스트로 간주될 수도 있으며, 다른 목적에 대해서는 분산된 프로세서 상에서 구동되는 프로세스로 간주될 수도 있다.

[0018] 상기에서 인용된 출원에서, 하나 이상의 툴 호스트들 및 하나 이상의 툴들 사이의 유선 통신 채널들을 모니터링 할 수 있는 제거가능한(removable) 청취(listening) 디바이스가 설명되어 있다. 상기 청취 디바이스는 수동적이며, 이는 상기 청취 디바이스에 의해 발생하는 잡음으로부터 통신 채널을 보호하기 위한 표준 격리 디바이스(standard isolation device)를 선택적으로 포함할 수 있다. 이러한 격리 디바이스는 광 차폐기(optical isolator), 고임피던스 증폭기 또는 상기 청취 디바이스로부터 유선 통신 채널을 효과적으로 격리시키는 임의의 다른 구성요소들을 포함할 수 있다. 상기 유선 통신 채널은 RS 232, RS 422, CAN-compliant 채널이 될 수도 있으며, 또는 전에 언급된 임의의 통신 채널일 수도 있다.

[0019] 제어기 및 스마트 센서들

[0020] 또 다른 선출원에 개시된 방법에서는 지능형 제어기들 및 스마트한 문맥-인식 센서들(context-aware sensors)을 사용한다. 지능형 제어기는 툴 및/또는 상기 툴이 공정중인 작업품들(workpieces)(가령, 웨이퍼들 또는 레티클들)의 상태를 알고 있다. 이러한 타입의 제어기들은 툴 및 작업품의 상태 정보에 반응하는 스마트 센서와 통신한다. 툴 및 작업품의 상태가 변화할 때에 명령들을 재구성하는데 의존하는 대신에, 상기 스마트 센서들은 상태 변화들을 청취하고 이에 응답한다. 상기 센서들은, 명령들의 재구성을 요구하는 대신에, 미리 프로그램되어진 방법에 따라 상태 변화에 대해 반응한다.

[0021] 새로운 지능형 제어기들은 툴들, 센서들, 제어기들 및 데이터 사용자들에 대한 동작 모델을 변경한다. 한 방법 또는 다른 방법을 통해 상기 제어기들은 툴들 및 작업품들(workpieces)의 상태를 알고 있다. 제어기는 툴을 제어하는 명령들을 도청할 수도 있으며 또는 중계할 수도 있다. 대안적으로, 상기 툴은 자신의 상태를 제어기에게 알릴 수도 있다. 또는, 상기 제어기는 툴의 상태에 대해 주기적으로 문의할 수도 있으며 또는 후속 문의를 필요로 한다고 인식되는 이벤트들에 대한 응답으로 툴의 상태에 대해 문의할 수도 있다. 제어기들은 센서들과 상태 정보를 통신한다. 상기 상태 정보는 툴 또는 작업품과 관련될 수 있다. 센서들은 상기 상태 정보에 대해 반응하도록 미리 구성되어 있다. 상기 상태 정보에 응답하여, 센서들은 데이터 수집 계획을 채용할 수도 있으며, 스스로 수치를 조정하거나, 출력 레인지를 설정하거나, 현재의 상태 정보와 데이터를 관련시킬 수도 있다. 상기 제어기들은 센서들로부터 수집된 데이터를 데이터 사용자들에게 전한다. 상기 데이터 사용자는 메인 프레임상에서 동작하는 전통적인 툴 호스트일 수도 있으며 또는 분산된 프로세서들에서 동작하는 새로운 소프트웨어일 수도 있다. 상기 데이터 사용자는 모노리식 시스템(monolithic system)일 수 있으며 또는 독립적으로 또는 협동적으로 작동하는 연합 패키지들(confederated packages) 일 수도 있다. 또한 상기 제어기들은 센서들로부터의 데이터를 모니터링 할 수도 있고, 관심있는 이벤트들을 식별할 수도 있으며, 이미 수집된 데이터를 추가로 요구할 수도

있고, 모니터링된 데이터에 응답하여 센서들의 수집계획을 변경할 수도 있다.

[0022] 다양한 작동 환경들(Various Operating enviroment)

[0023] 도1은 본 발명의 양상들이 특히 유용하게 사용될 수 있는 환경을 도시한 도면이다. 도1에는 프로세스 챔버(125)와, 프로세스 챔버로 입력되는 다양한 입력들 및 프로세스 챔버로부터의 다양한 출력들, 또한 센서들(plus sensors), 제어 채널들 및 제어기들이 도시되어 있다. 상기 챔버(125)는 가령, 증착, 세정(cleaning), 에칭(etching), 이온주입, 애싱(ashing) 등등과 같은 다양한 반응들(reactions)을 위해 사용될 수 있다. 도1에는 도시되지 않은 다른 타입들의 툴들 역시, 본 발명의 양상에 의해 이득을 얻을 수 있다.

[0024] 인터넷, 가상 사설 네트워크(Virtual Private Network : VPN) 또는 광역 네트워크(wide area network)(112)를 경유하여 접근할 수도 있는 랩 네트워크(111)는, 제어기, 방화벽(firewall) 또는 또 다른 커넥터(162)를 통해 툴 네트워크(112)로의 제어된 액세스를 갖는다. 도1의 툴 네트워크에서는, 프로세스 챔버(125)와 링(ring) 형태로 연결된 제어기들 및 센서들이 도시되어 있다. 해당 기술분야의 당업자라면, 이러한 구조(architecture)가 단지 예시적인 것일 뿐이라는 것을 알 것이다. 링(ring) 보다는 직렬 통신들, 이더넷 또는 계층적 통신수단들(tiered communications)이 랩(fab) 내에서 사용될 수도 있음을 이해할 것이다.

[0025] 반응 챔버(125)로 입력되는 가스성 입력들(gaseous inputs)은 가스 박스 압력 변환기(gas box pressure transducers)(113)와 질량 유량 제어기(Mass Flow Controller : MFC)(114)를 통과하는 가스를 포함한다. 몇몇 가스는 오존 발생기(133)를 통과할 수도 있다. 다른 가스들 및 가스 혼합물들은 반응성 가스 발생기(reactive gas generator)(115)와 가스 성분 모니터(gas composition monitor)(117)를 통과할 수도 있다. 반응성 가스 발생기(115)는 프로세스 챔버(125)의 내부 또는 외부에서 플라즈마를 생성할 수도 있다. 가스 성분 모니터(117)는 상기 반응성 가스 발생기와 직렬 또는 병렬로 연결될 수도 있다. 질량 유량 제어기(114)는, 반응성 가스 발생기(115)와 가스 성분 모니터(117)와 가스를 주고받으며, 그리고 직접적으로 또는 궁극적으로(ultimately) 프로세스 챔버(125)와 가스를 주고 받는다. 가스성 입력 디바이스들(gaseous input devices)(113, 114, 133, 115, 및 117)은 하나 이상의 디지털 제어기들(142), 챔버 제어기들(152), 및 연결 포인트들(162)과 통신한다. 이러한 통신은 전형적으로 제어 및 텔레메트리(telemetry) 둘 모두를 포함한다. 이러한 디바이스들은, 디바이스들의 작동에 반응하거나 또는 가스성 입력 및/또는 출력에 반응하는 제어기들 및 센서들 모두를 포함할 수 있다.

[0026] 다른 입력들은 물질 전달 시스템(134), 냉각 서브시스템(145), 다양한 파워 인젝터들(power injectors)(153, 154, 155)을 포함할 수 있다. 반응 챔버(125)는 증착 챔버, 식각장비(etcher), 열 처리장치(thermal processor), 또는 다른 타입의 반응기일 수 있다. 물질 전달 시스템(134)은 반응 챔버의 타입에 의존하여, 예를 들면, 증착에 필요한 물질들을 작업품들(136)에 공급할 수 있다. 냉각 서브시스템(145)은 챔버(125)내의 온도를 조절하는데 도움을 줄 수 있으며, 이는 대부분의 화학 반응들이 온도에 민감한 속도에서 진행되기 때문이다. 챔버로 공급되는 파워는 마이크로 와트 파워(153), 플라즈마를 생성하는데 이용되는 RF 파워(154), 및 DC 파워를 포함할 수 있는데, DC 파워는 플라즈마를 생성하거나 챔버를 가열하거나 가스들 또는 챔버로 공급되는 다른 물질들을 가열하는데 사용된다. 가령, 가스성 입력들과 같은 다른 입력들은, 하나 이상의 디지털 제어기들(142), 챔버 제어기들(152), 및 연결 포인트들(162)과 통신한다. 이러한 통신은 전형적으로 제어 및 텔레메트리(telemetry) 둘 모두를 포함한다. 이러한 디바이스들은, 디바이스들의 작동에 반응하거나 또는 가스성 입력 및/또는 출력에 반응하는 제어기들 및 센서들 모두를 포함할 수 있다.

[0027] 센서들은 챔버의 상태들(conditions)에 반응하거나 또는 챔버로부터의 배출(exhaust)에 대해서 동작한다. 챔버의 상태들에 대해 반응하는 센서들은, 창(126)을 통해 챔버(125) 내부를 주시하여 막(film)의 두께, 패턴들 및 다른 특성들(예를 들면, EPI-Online(TM))을 관찰하는 웨이퍼 모니터(116)와, 식각 공정 제어를 위해 간섭 필터 또는 간섭계(interferometer)를 구비한 광학 발광 모니터(optical emission monitor)와 같은 공정 모니터(127)와, 압력 변환기(137)를 포함할 수 있다. 챔버(125)로부터의 배출에 대해서 동작하는 센서들은, 누설 검출기(146), 진공 게이지(157) 및 배출 모니터(158)를 포함할 수 있다. 이러한 센서들은 압력 조절기(148) 및 제어 밸브(147)와 상호작용할 수 있으며, 진공과 관련된 구성요소들 및/또는 서브 시스템들과 상호작용할 수 있다. 또한, 이러한 센서들은 도1에는 도시되어 있지 않은 펌프 및/또는 배출 가스 세정기(exhaust gas scrubber)와도 상호작용할 수도 있다. 이러한 센서들은, 하나 이상의 디지털 제어기들(142), 챔버 제어기들(152), 및 연결 포인트들(162)과 통신한다. 이러한 통신은 전형적으로 제어 및 텔레메트리(telemetry) 둘 모두를 포함한다. 상기 센서들과 통신하는 디바이스들, 가령, 147, 148, 156은 제어기들 및 센서들 모두를 포함할 수 있다.

[0028] 도1에는 도시되어 있지 않지만, 중간 매개 디바이스가 가령, 랩 사이드(fab side) 네트워크와 같은 제 1 디바이스(또는 제 1 네트워크)와 가령, 툴 네트워크와 같은 제 2 디바이스(또는 제 2 네트워크) 사이에 삽입될

때마다, 오작동 방지(failsafe)) 스위치가 적용될 수도 있는 바, 상기 제 1 디바이스와 제 2 디바이스는 중간매개 디바이스가 없이도 서로 통신할 수 있다. 툴 또는 센서로부터 기록들(readings)을 수집하고, 상기 기록들을 저장하고, 요청될 때마다 상기 기록들을 활용가능하게 만드는 제어기가 그 일례가 될 수 있다. 센서로부터의 데이터를 이용가능하도록 보완하는 제어기는 다른 일례가 될 수 있다. 하나의 포맷에서(예를 들면 SECS와 같은) 다른 포맷(예를 들면, tagged XML 포맷과 같은)으로 데이터를 변환하는 제어기는 또 다른 일례가 될 수 있다. 해당 기술분야의 당업자라면, 제 1 및 제 2 디바이스들(또는 네트워크들)에 대한 오작동 방지 직접 연결에 의해, 중간 매개 디바이스의 고장이 해결될 수도 있는 또 다른 상황들이 있을 수도 있음을 충분히 인식할 수 있을 것이다.

[0029] 제어기 배치 및 아키텍처

[0030] 도2는 툴들, 센서들 그리고 툴 호스트와 통신하는 제어기에 대한 블록도이다. 이러한 블록도는 또한, 호스트(211)를 제 1 디바이스로 간주하고, 센서(227) 또는 기기(instrument)(237)를 제 2 디바이스로 간주하고, 상기 제어기(200)를 중간 매개 디바이스로 간주한다면, 일반화될 수 있다. 도시된 실시예에서, 제어기(200)의 구성은 2개의 SEC/GEM 인터페이스 포트(212, 216)와 2개의 네트워크 인터페이스 포트(232, 236)를 포함한다. 상기 제어기는, SECS MUX(215)를 포함하여 SECS(SEMI Equipment Communication Standard) 프로토콜을 통해 통신하는 논리회로들(logic) 및 리소스들(resources)을 포함한다. 또한, 상기 제어기는, 데이터 사용자와 통신하기 위한 랩 사이드(fab side) 인터페이스(234)를 구현하는 논리회로들 및 리소스들을 더 포함하며 툴들, 센서들 및 기기들과 통신하기 위한 툴 사이드 인터페이스(235)를 구현하는 논리회로들 및 리소스들을 더 포함한다. SECS MUX(215)와 인터페이스들(234, 235)은 데이터 수집 및 발행(publishing) 리소스들(225)에 논리적으로 연결되어 있다. 상기 제어기의 랩 사이드(fab side)에서는, SECS-I, HSMS 또는 SECS의 후속 수정판 또는 계승자인 SECS 순응 통신채널(SECS compliant 통신채널)을 통해 전통적인 툴 호스트(211)가 인터페이스 포트(212)에 연결될 수도 있다. 랩이 아닌 환경하에서도 본 발명이 적용될 수도 있는바, 그런 환경에서는, 가령, 의료용 기기 또는 수치제어되는 머신 툴과 같은 모니터링되는 툴을 연결하기 위해 다른 프로토콜들이 사용될 수 있다. 또한, 상기 제어기는 SECS와는 별개인 랩 사이드 프로토콜을 경유하여 연결될 수도 있는바, 네트워크에 의해 저장소(231) 및 보고 리소스들(reporting resources)(221)로 운반된다. 제어기(200)의 툴 및 센서 사이드에서는, 툴 또는 툴 클러스터에 대한 SEC/GEM 툴 인터페이스(217)가 인터페이스 포트(216)에 연결될 수도 있다. 또한, 상기 제어기는, 센서들(227), 기기들(237) 및 다른 디바이스들을 호스트하며 잠재적으로는 툴들(217)을 호스트(host)하는 네트워크(226)와 연결된 인터페이스 포트(236)에 연결될 수도 있다. 비록, 이 도면에서는 네트워크를 경유하여 센서들에 연결되어 있는 제어기가 도시되어 있지만, 상기 제어기는 SECS-I 또는 또 다른 프로토콜, 센서들을 연결하는 직렬-기반의(serial-based) 프로토콜을 대안적으로 사용할 수 있다.

[0031] 도3은 툴들, 센서들 및 툴 호스트와 통신하기 위해 하나의 타입의 통신 채널을 사용하는 제어기에 대한 블록도이다. 도4는 복수개의 타입의 통신 채널들을 사용하는 것을 도시한 도면이다. 도3에는, SECS가 지배하는 시나리오가 도시되어 있다. 상기 제어기(303)는 툴(301) 및 센서(302) 둘 다와 통신하기 위해 SECS 프로토콜(307, 309)을 사용한다. 툴(301)과 센서(302)간의 접속(308)은, 통상적으로 감지되는 임의의 타입의 에너지 또는 힘과 관련될 수 있는 바, 도1에 대한 설명에서 식별되었던 임의의 센세이션들(sensations)을 포함한다. 도4에서는, SECS와 네트워크 통신들의 좀더 복잡한 조합이 도시되어 있다. 이러한 시나리오에서는 오직 하나의 SECS 통신 채널(307)만이 제어기(303)와 호스트(301) 사이에 존재한다. 툴 사이드 네트워크(409)는 제어기를, 센서들(302A-B), 관리된 스위치(managed switch)(413) 및 네트워크 첨부 저장소(Network Attached Storage)(415)에 연결한다. 랩 사이드 네트워크(419)는, 가령 전통적인 툴 호스트 또는 분산 프로세서들과 같은 분석 소프트웨어(423)와 제어기(303)를 연결하며, 가령 인터넷, 가상 사설 네트워크 또는 전용 인터넷 네트워크와 같은 확장된 네트워크(425)와 제어기(303)를 연결하며, 제어기(303)에 의해 발행된 데이터를 유지하거나 또는 저장하는 데이터 베이스(427)와 제어기(303)를 연결한다. 대안적으로, 상기 데이터 베이스(427)는 제어기(303) 내에 상주할 수도 있다.

[0032] 네트워크로의 오작동 방지 스위치의 도입

[0033] 도5는 도3에 도시된 네트워크에 오작동 방지 스위치를 추가한 것을 도시한 도면이다. 2개의 새로운 구성요소가 도입되었는 바, 바이패스 스위치(511)와 허브(또는 스위치)(512)가 그것이다. 다양한 구성예들에서, 바이패스 스위치(511)만이 새로울 것이다. 이러한 네트워크 구성에서는, 툴들과의 통신은 네트워크 프로토콜을 경유한다. 이러한 통신 프로토콜은, 만일 상기 툴이 네트워크 프로토콜을 이전에 사용하지 않았다면은, 프로토콜 어댑터를 허브 또는 스위치(512)와 툴(501) 사이에 도입함으로써 구성될 수도 있다. 대안적으로, 2개의 통신 채널들이 바이패스 스위치(511)에 의해 제어될 수 있는데, 예를 들면, SECS 트래픽을 운반하는 직렬 연결과 다른 트래픽을

운반하는 이더넷 연결(Ethernet connection)이다. 도3에 도시된 바와같이, 제어기(303)는 정상 동작 모드에서 톨(301) 및 센서(302) 둘 다와 통신한다. 상기 센서는 톨에 접속되어 톨의 상태 또는 성능의 몇몇 양상을 측정하거나 기록한다. 톨 웹 또는 호스트(304)로부터의 메시지들은 바이패스 스위치(511)를 경유하여 제어기(303)로 전송되며, 제어기로부터의 메시지들은 바이패스 스위치(511)를 경유하여 허브 또는 스위치(512)로 전송된다. 상기 허브 또는 스위치(512)는 디바이스들을 상호연결해 주는 통상적인 네트워크 구성요소이다. 상기 제어기(303)는 적어도 2개의 네트워크 인터페이스(521, 522)를 포함한다. 이러한 인터페이스들은, 톨 웹(304) 및 톨(301) 또는 센서(302)와 통신하는 바이패스 스위치(511)를 통하여 효과적으로 접속되어 있다. 제어기(303)와 바이패스 스위치(511) 사이의 추가적인 연결(523)은, 바이패스 스위치를 정상 모드(오작동 방지가 아님, non-failsafe)로 유지하는데 필요한 심박 신호(heartbeat signal) 또는 또 다른 생생한 신호(keep alive signal)를 운반한다. 이 도면에서, 2가지 동작모드들이 도시되어 있는 바, 제어기의 정상 동작은 실선으로 표시되며, 바이패스 동작은 점선으로 표시된다. 비록 도면에는, 바이패스 스위치를 관통하는 2가지의 연결들이 도시되어 있지만은, 이러한 연결들은 단일 유선 또는 단일 연결상의 논리적인 동작 모드들일 수도 있으며, 바이패스 스위치(511)와 톨웹(304) 사이 또는, 바이패스 스위치(511)와 허브(또는 스위치)(512) 사이에서 한 쌍의 연결들일 필요는 없다. 다시한번 말하자면, 바이패스 스위치는 하나 이상의 연결들을 제어할 수 있다.

[0034] 도6은 도4에 도시된 네트워크에 오작동 방지 스위치가 추가된 것을 도시한 도면이다. 도6에서, 409에 의해 바이패스 스위치(511)에 연결된 허브 또는 스위치는, 바이패스 스위치로부터 톨(301) 및 센서들(302A-B)로의 멀티-드롭(multi-drop) 네트워크 연결에 의해 전형적으로 수반되어 있다. 전술한 바와같은 바이패스 스위치 및 제어기 사이의 3개의 연결들은, 도6에도 도시되어 있다.

[0035] 오작동 방지 동작

[0036] 도5로 돌아가서, 이제 오작동 방지 동작에 대한 일 실시예를 설명한다. 제어기가 작동불능이 되어, 중간 매개 제어기가 없이도 서로 통신할 수 있는 2개의 디바이스들 사이에서 제어기가 더 이상 적절한 역할을 수행하지 못하는 경우, 상기 디바이스들[예를 들면, 호스트(304) 및 톨(301)] 사이에서 연결이 직접 설정된다. 일 실시예에서, 간단한 마이크로컨트롤러가 내장된(embedded) 스위치가 사용된다. Shore Micro 사는 이와같은 기능을 하는 적어도 하나의 스위치를 공급한다. 상기 스위치는, 스위칭 동작을 할때 재확인(reassuring) "click"을 만들어내는 중계기(relay)를 포함한다. 스위치내의 마이크로컨트롤러는, 직렬, 병렬, USB, 파이어와이어, 이더넷 등등과 같은 연결(523)을 통해 제어기로부터의 심박 신호(heartbeat signal)를 주목하고 있다. 오작동 방지 스위치에 제공되는 로직 및 리소스들에 좌우하여, 심박신호(heartbeat signal) 이외의 다른 프로토콜들이 사용될 수도 있다. 예를 들어, 상기 오작동 방지 스위치는, 제어기로 주기적으로 조회하고 에러 상태 징조들(error condition indications) 또는 시간 초과된 응답(response time-out)에 반응한다. 이러한 조회들은 정기적일 수도 있으며, 많은 트래픽, 적은 트래픽 및/또는 관심있는 다른 동작 조건들에 따라 적응적일 수도 있다. 마이크로컨트롤러에 의해 제어되는 스위치는, 제어기(303)를 호스트(304) 회로내로 끼어들게 하거나 또는 떼어내는 2극쌍 스위치(double pole double throw switch)와 유사하다. 정상 모드에서, 상기 제어기는 회로내에 물리적으로 연결되어 있다. 오작동 방지 모드에서, 상기 호스트 및 톨/센서들은 직접적으로 연결되어 있다. 대안적으로, 중계기의 재확인 클릭을 포기하면서 전자적인 스위칭 수단이 채용될 수도 있다.

[0037] 심박 및 오작동 방지 연결을 구현하는 소프트웨어는, 패킷 주소들을 설정하는 디바이스 드라이버(device driver), 뒤범벅된 모드(promiscuous mode)에서 네트워크 인터페이스가 구동될 때 패킷들을 포획하기 위한 커널 후크(kerner hook), 와치 독 데몬(watch dog daemon), 및 사용자 스페이스 프로그램(user space program)을 포함할 수도 있다. 하기에서 설명되는 부분은 리눅스를 지향한 것이다. BSD 변형물들(variants), UNIX 변형물들(variants), 또는 윈도우와 같은 다른 운영 시스템들도 사용될 수 있다. 시스템은, 가령 Java의 실시간 버전과 같은 가상 머신상에서 작동하도록 쓰여질 수 있는 바, 낮은 레벨의 네트워킹 기능들을 제어하기 위해 소프트웨어를 하나의 운영 시스템에서 다른 운영시스템으로 이식하는데 있어서, 소프트웨어에 대한 작은 변경만을 필요로 한다.

[0038] 상기 디바이스 드라이버는 TCP/IP 스택에 접속되어(hook), 마치 제어기가 연결되어 있지 않은 것처럼 상기 제어기를 투명하게 만든다. 디바이스 드라이버는, 중간 매개 제어기 대신에 상기 톨 또는 상기 호스트로부터 패킷들이 온것처럼 보여지게 하기 위해 패킷들 내의 소스 주소들과 정보를 변경함으로써, 이를 가능케 한다. 상기 디바이스 드라이버는 커널 스페이스내에서 사용자 스페이스로부터 TCP/IP hooks 로의 데이터를 복사한다.

[0039] 상기 제어기는, 하나의 네트워크 인터페이스에서 다른 네트워크 인터페이스로 브리징(bridging)함으로써 제어기에 의해 이러한 패킷들이 처리되는 일 없이 HSMS 포트로 어드레스되지 않는 선택된 패킷들을 다룰수도 있다. 예

를 들어, 핑(ping) 또는 ICMP 패킷은 제어기에 의해 처리되는 일 없이 통과할 수 있다. 이러한 선택적인 브리징은 패킷 타입 및/또는 포트에 적용될 수 있다.

- [0040] 커널 후크는, 네트워크 인터페이스가 뒤범벅된 모드에서 동작할 때, 적어도 소정의 패킷들을, 본질적으로 네트워크 인터페이스가 나타나 있는 어드레스로 향하지 않는 스니핑 패킷들(sniffing packets)을, 세이브(save)할 수 있는 몇줄 안되는 코드이다.
- [0041] 변형되지 않은 리눅스 커널(예를 들면, 4.2.26)은, 비록 뒤범벅된 모드에서 인터페이스를 구동하더라도, 변형되지 않은 리눅스 커널(레이어 2, 또는 MAC 레이어)으로 어드레스되지 않은 패킷들은 포획하지 않는다. 그것은 ip_rcv function 내의 내용을 덤프(dump)한다. 상기 커널 후크는, 처리되어질 변형되지 않은 커널의 부분이 아닌 평선(function)으로 소정의 패킷들을 전달한다.
- [0042] 와치 독 데몬들은 스위치와 통신하며 심박신호를 송출한다. 만일, 스위치가 상기 데몬으로부터의 심박신호를 듣지 못한다면, 상기 스위치는 연결에 대한 오작동 방지를 수행할 것이다[(상기 중계기를 스위칭함(switch the relay)]. 스위치에 대한 파워 업(power up)이후에 오류가 있는 연결은 오작동 방지된다. 상기 심박 데몬은 syslog에 기록한다.
- [0043] 사용자 스페이스 프로그램은, 제어기 자신의 포트 및 그 사이에 제어기가 위치하는 제 1 및 제 2 디바이스들(예를 들면, 툴/센서 그리고 호스트)을 위한 IP 또는 다른 어드레스들을 셋업한다.
- [0044] 리눅스에서, 구성 정보(configuration information)를 위한 사용자 스크립트(userscript)와 스타트업(startup)은 다른 구성요소들을 함께 붙여있다(glue). 상기 소프트웨어는 샘플 구성 파일과 번들되어 있다. 상기 구성 파일은 호스트 IP, 툴 IP, 와치 데몬 타이머를 트리거링하는 시간 간격, 와치 독 데몬과 오작동 방지 스위치가 통신하기 위한 com port 번호, 및 향후 향상에 대한 옵션 모드를 포함하고 있다. 리눅스 4.2.26 으로 구현되었을 때, syslog 및 네트워크 이후 및 easi 이전에 스타트업 스크립트는 바람직하게 작동한다. 스타트업 스크립트들에 있어 전형적인 것과 같이 상기 스크립트는 데몬 동작을 개시하고 정지시킨다. 상기 스크립트는 명명될 수도 있으며, 예를 들면, hsmfsd start 로 명명될 수 있다. 상기 스크립트는 다른 스크립트들보다 길 수도 있는데, 이는 상기 스크립트가 상기 툴 및 상기 호스트 둘 다의 MAC 어드레스들을 발견하고 있으며, 그리고 필요하다면 적절한 경로들을(routes) 부가하고 있기 때문이다. 발견을 성공시키기 위하여, 상기 제 1 및 제 2 디바이스들(예를 들면, 상기 툴 및 상기 호스트)은 스타트업(startup) 에서 제어기에 연결된다. 대안적으로, 기 결정된 IP 어드레스들이 로드(load)될 수 있으며 어떤 발견(discovery)이 없이도 사용될 수 있다. 또한, 상기 스타트업 스크립트는, 제어기의 IP 어드레스들이 네트워크 인터페이스들인 eth0 및 eth1 과 어떻게 관련되어 있는지를 상기 TCP/IP 스택에게 알려주기 위해서, 경로(route)를 부가한다.
- [0045] 제어기 또는 다른 중간 매개 디바이스가 적당하게 부가되어 있는 가장 간단한 구성은 툴 및 호스트를 포함한다. 이러한 간단한 케이스에서, 상기 툴 및 상기 호스트의 IP 어드레스들은 알려져 있으며 구성 테이블 내에 존재한다. 상기 제어기가 상기 hsmfsd start 스크립트를 구동할 때, 스위치 와치독 데몬이 시작된다. 만일, 상기 블루박스가 스위치가 온라인을 유지하도록 하는 심박신호들 또는 명령들을 계속해서 전송하지 못한다면, 이러한 데몬은 오작동 방지 매커니즘의 나머지에 대해서는 아무것도 알지 못하며 단지 상기 스위치(Shore Micro Switch)가 오작동 방지(호스트가 툴과 물리적으로 연결됨)로 스위칭하도록 야기한다.
- [0046] 상기 제어기가 정상적으로 작동하는 한, 상기 스위치는 온라인(online)을 유지하며, 블루박스를 통해 상기 툴과 상기 호스트를 연결한다. 상기 스타트업 스크립트는, 툴 및 호스트의 MAC 어드레스를 설정하도록 스탠다드 ARP 프로토콜을 사용할 수도 있으며, 로컬 ARP 테이블(local ARP table)을 갱신할 수 있다. ARP 프로토콜에 관한 좀더 상세한 내용은 다음의 RFCs 에 의해 설명되는바, 일반적인 ARP #826, PROXY ARP #1027, DHCP ARP #2131 및 UNARP #1868. 만일, 상기 호스트 또는 상기 툴이 제 2 네트워크 인터페이스(예를 들면, eth1) 상에서 응답한다면, 호스트 경로가 그 IP 주소와 함께 제 2 인터페이스에 부가된다.
- [0047] 이러한 실시예에서, 제어기 네트워크 인터페이스들 처럼 상기 툴 및 상기 호스트는 동일한 네트워크 서브네트 상에 존재한다. 상기 제어기는 메시지들을 라우팅(route)할 수 있도록, 어떤 인터페이스가 상기 호스트에 연결되는지 그리고 어떤 인터페이스가 상기 툴에 연결되는지를 알 수 있거나 또는 인지하도록 구성된다. TCP/IP 스택에 대한 디폴트 경로(default route)가 사용될 수도 있으며, 그렇지 않으면, 동일한 네트워크 서브네트(호스트 및 툴 모듈을 포함하는)에 대한 모든 메시지들이 상기 동일한 네트워크 인터페이스로 라우팅되기 때문에 작동하지 않을 것이다.
- [0048] 상기 모듈들이 로드되며, 이후에 IP 어드레스들을 수반한 어플리케이션이 구성 파일로부터 획득되며 이러한 IP

어드레스들에 대한 MAC 어드레스들이 로컬 ARP 테이블로부터 획득된다. 이제, 상기 호스트 및 상기 툴은 블루박스를 통해 연결되었다. EMX 셋업하에서 상기 블루박스 구성은 수동적이어서 호스트로부터의 연결을 받아들일 것이며 툴의 IP 어드레스로 툴과 연결되는 점에서는 능동적이다. 상기 수동적 및 능동적 HSMS 셋업은 공통으로 동일한 포트를 가져야 한다.

[0049] 디바이스 드라이버에 의해 유지되며 커널내에서 TCP/IP 스택 후크에 첨부되는 테이블은, 상기 툴에 대한 엔트리 및 상기 호스트에 대한 또 다른 엔트리를 가질 것이다. 제 2 디바이스, 예를 들어 툴/센서, 에 대한 테이블 엔트리들은 다음과 같을 수 있다.

Entry Number	Sequence by the app
Protocol	0x6 for TCP
Flags	1 means the MAC address is valid
Destination MAC address	Taken from startup script
Destination IP address	Taken from the startup script
Destination Port	Taken from configuration
Source IP address	Unknown
Source port	Unknown

[0050]

[0051] 제 1 디바이스, 예를 들면 호스트, 에 대한 테이블 엔트리는 다음과 같을 수 있다.

Entry Number	Sequence by the app
Protocol	0x6 for TCP
Flags	1 means the MAC address is valid
Destination MAC address	Taken from startup script
Destination IP address	Taken from the startup script
Destination Port	Unknown
Source IP address	Taken from configuration
Source port	Taken from configuration

[0052]

[0053] TCP/IP에 대해, 소스 IP, 소스 포트, 목적지 IP, 목적지 포트로 구성된 4-튜플(4-tuple)에 의해, 고유한 연결은 전형적으로 식별된다. 이러한 점은 상기 연결을 네트워크내에서 보편적인 것으로 만들 것이며 또한 고유한 것-네트워크 프로토콜들은 상기 4-튜플단지 하나의 연결만을 지원함-으로 만들 것이다. 또한 이용가능한 것은 소위 하프 어쏘시에이션(half-association)이라 불리우는 것으로, 목적지 IP 및 포트를 포함하며, 여기서 상기 포트는 서비스를 참조하며 상기 IP는 연결이 만들어지는 머신(machine)이다. 예를 들어, www.mksinst.com 은 포트 80이며 도메인 네임 서버(Domain Name Server)가 www.mksinst.com 이라는 이름으로부터 변화(resolve)시키는 어떤 IP 어드레스이다. ISP 사용자의 경우, ISP의 종단(end)의 서버는 당신의 IP를 간직할 것이며 로컬 포트를 할당할 것인 바, 이는 고유한 4-튜플을 만든다. 어떤이가 브라우저의(browser) 또 다른 윈도우를 오픈할 때마다 동일한 목적지 IP 및 포트, 동일한 소스 IP, 고유성 요구사항(uniqueness requirement)을 만족시키기 위해 할당된 서로 다른 소스 포트의 또 다른 고유한 4-튜플이 생성된다.

[0054] 실제적인 작동에서, 호스트는 예를 들면 툴 또는 센서에 연결하기를 시도할 것이며, 상기 제어기는 뒤범벅된 모드에서 동작하는 네트워크 인터페이스를 이용하여 호스트로부터의 하나 이상의 메시지들을 가로챈다. 입력 후크(input hook)에 첨부된 제 1 필터가 적용되며, 이는 상기 패킷이 제어기 상의 임의의 네트워크 인터페이스들에 속하는지를 결정한다. 상기 제 1 필터는 인터페이스에 속하는 임의의 패킷을 그 인터페이스로 제공하고 종료한다(exit). 만일, 상기 패킷이 제어기의 네트워크 인터페이스들에 속하지 않는다면, 상기 필터는 목적지 포트 및 IP를 전술한 디바이스 드라이버 테이블에서 검색한다. 만일, 상기 목적지가 디바이스 드라이버 테이블에서 발견된다면, 상기 필터는 패킷의 목적지를 제어기와 관련된(controller-related) 어드레스로 변경하며 이를 TCP/IP 스택으로 전달한다. 상기 패킷이 TCP/IP 스택에 의해 처리된 이후에, 출력 후크에 첨부된 제 2 필터는 목적지를

복원하므로, 승인(acknowledgement) 또는 응답이 원래의 목적지로부터 온것으로 보여질 것이다. 이러한 프로세스는 때때로 매스캐레이딩 인(masquerading in)이라고 불리운다. 이러한 접근은 병행(concurrent) 세션들을 지원한다. 상기 제어기가 상기 패킷을 직접 처리하는지 또는 상기 패킷을 원래의 목적지로 중계하는 지는, 패킷의 타입 및 세션의 이력(history)에 의존한다.

[0055] 실제적인 동작에서, 툴/센서는 호스트로의 연결을 시도할 것이다. 상기 중간매개 디바이스인 상기 제어기는 패킷을 호스트로 전송하려고 할 것이다. IP 전송 함수(IP send function)는 목적지 패킷 및 목적지 포트를 검색한다. 만일, 목적지가 발견되었다면, 상기 함수는 제어기 IP인 패킷 소스 IP를 테이블 내의 툴 IP로 변경하며 목적지 포트(제어기의 소스 포트일 수도 있다)를 테이블로 복사한다. 상기 툴은 상기 패킷에 응답할 것이며 수신 함수(receive function)는 상기 응답을 입수할 것이다. 하지만, 이제 상기 테이블은 완전하다. 따라서, 검색은 성공할 것이며 상기 패킷을 스택으로 다시 보낼(redirect) 것이다. 이러한 프로세스는 때때로 매스캐레이딩 아웃(masquerading out)으로 불리운다.

[0056] ARP 엔트리들이 오래되면, 상기 툴 또는 상기 호스트는 그들의 MAC 어드레스를 요구할 것이다. 예를 들어, 상기 툴은 호스트 IP의 MAC 어드레스를 요구할 것이지만, 상기 툴은 상기 호스트에는 연결되어 있지 않으며 제어기에 연결되어 있다. 상기 제어기는 호스트의 IP 어드레스를 그 자신의 테이블 엔트리들에서 검색하며, 호스트에 대한 적절한 MAC 어드레스로 ARP 요청 메시지에 대해 응답한다. 이것이 바로 디바이스 드라이버 내에서의 ARP 후크이다. 제어기에 의해 제공되는 MAC 어드레스는 호스트의 진정한 MAC 어드레스이거나 또는 호스트로부터의 ARP 요청에 응답하는 툴의 진정한 MAC 어드레스이다. 패킷의 여행은 TCP/IP 스택에 연결된 모듈(hsmsfs.o 디바이스 드라이버)의 내부에서 일어난다.

[0057] 만일, 어떤 이유로 인해 상기 제어기가 동작을 정지하거나 시스템 록업(system lockup)을 경험한다면, 오작동 방지 기능이 활성화될 수도 있다. 상기 직렬 스위치 데몬이 시스템을 구동하고 있으며 상기 시스템이 정상으로 보여지는 한, 상기 제어기는 온라인을 유지할 것이며 상기 호스트 및 상기 툴은 상기 제어기를 통해 연결될 것이다. 만일 상기 데몬이 고장난다면, 상기 스위치는 오작동 방지로 이행될 것이며 상기 호스트 및 상기 툴은 서로간에 직접적으로 연결될 것이다. 선택적으로(optionally), 데몬은 제어기 동작의 안정상태를 평가할 수도 있으며, 고장을 예측하거나 고장을 감지하여 적절한 대응을 할 수도 있다. 예측된 고장의 경우, 상기 호스트는 통보받을 수 있으며 임의적으로는 상기 제어기의 상태에 대한 감지가 수정될 수도 있다. 오작동 방지 스위치가 발동할 때(when the failsafe switch is thrown), 현재 직접적으로 연결된 세션의 참가자들은 호스트로부터 툴로의 연결을 재시도 또는 재설정 할 것이다. 연결이 재 설정되었을 때, 오작동 방지 동작은 완수된다. 기계적인 중계기에 대한 일 실시예에 있어서, 스위치에 대한 전원이 상실되면 상기 중계기는 오작동 방지 세팅을 맡으며 상기 호스트를 툴로 직접 연결한다.

[0058] 호스트가 연결을 상실한 때에는, 세션의 재시작이 시도되어야 한다. ARP 테이블이 최신의 것이므로, 직접적인 연결이 손쉽게 설정된다. 세션을 재시작하는 것은 때때로 로버스트 메시징(robust messaging)이라고 언급된다.

[0059] 오작동 방지에 대한 이러한 실시예는 매우 많은 장점을 가지고 있으며, 이러한 장점들 중 일부는 대안적인 실시예들에서도 볼 수 있다. 제 1 또는 제 2 디바이스(호스트 또는 툴)에 어떤 소프트웨어도 부가될 필요가 없다. 상기 툴과 상기 호스트는 동일한 서브네트 상에 있으므로, 오작동 방지 직접 연결의 경우에 이들은 통신이 가능할 것이다. 중간 매개 디바이스 또는 제어기는 툴 및 호스트처럼 동일한 서브네트상에 있다. 상기 호스트 및 툴은, 만일 상기 제어기가 오작동 방지 모드에서 시작한다면, 서로를 볼 수 있다(see each other). 동작이 정상모드에서 오작동 방지 모드로 스위칭될 때, 활성화된 세션(active session)은 툴, 호스트 또는 양자 모두에 의해서 자동적으로 재시도 될 것이다.

[0060] 제 2 작동 실시예

[0061] 제 2 실시예에서는, 호스트 및 툴 사이에서의 ARP 요청 메시지들에 대해 제어기가 응답함이 없이도, 선택적인 브리징이 구현된다. ARP 메시지들 및 호스트로부터 툴로 향하는 다른 선택된 메시지들 또는 그 반대의 경우(vice-versa)의 메시지들은 제어기의 일 측으로부터 다른 일 측으로 브리지되는바, 뒤범벅된 모드에서 동작하는 네트워크 인터페이스로부터 또 다른 네트워크 인터페이스로 통과하며 그리고 재전송된다. 상기 툴은 호스트로부터 포워딩된(forwarded) APR 요청 메시지들에 응답하며, 그 반대의 경우도 마찬가지이다. 상기 제어기는, ARP 요청 메시지들 및 인지된 소스 및 목적지를 갖는(예를 들면, 테이블 검색(look-up)에 기초한) 다른 메시지들을 선택적으로 브리지하며 포워딩한다. 유용한 테이블은 앞에 기재되어 있으며, 브리징은 선택적이다. 인지되지 않은 목적지를 갖는 패킷은 커널 후크에 의해 처리되지 않지만, 보통의 TCP/IP 스택 프로세스를 통해 통과하는 것이 허용되는 바, 이후에 상기 패킷은 드롭(drop)될 수도 있다. 몇몇 메시지들을 필터링 처리(filtering out)하

는 것은 네트워크 상에서 싸이클들을 감소시킨다. 예를 들어, 만일 호스트가 다섯개의 제어기를 통해 다섯개의 툴들에 연결된다면, 첫번째 제어기는 2번째 내지 5번째 툴로 향하는 몇몇 메시지들 또는 모든 메시지들을 안전하게 필터링 처리할 수 있다. 툴 사이드 네트워크 상의 트래픽은 상당히 감소될 수 있다. 또 다른 일례로서, 상기 제어기는 상기 툴로부터의 데이터, 호스트가 기대하고 있거나 또는 호스트가 요청할 수 있는 옵션을 갖고 있지만 아직 요청하고 있지는 않는 데이터를 획득하고 저장할 수도 있다. 상기 제어기는 상기 제어기로 향하는 데이터 트래픽을 안전하게 필터링 처리(filter out)할 수 있으며 호스트 사이드 네트워크 상의 트래픽을 감소시킬 수 있다. 선택적인 브리징에 적용될 수도 있는 하나의 물은, 제어기에 의해 인식되어지며 제어기와 통신하는 호스트 및 목적지로 패킷들만을 포워딩(forward)하는 것이다. 또한, 선택된 유형의 브로드캐스트 메시지들이 브리징될 수도 있다. 전술한 커널 후크는, 네트워크 인터페이스들 사이에서 패킷들을 전송하는데 필요한 지원과 패킷들을 브리징하는데 필요한 지원을 제공하기 위해 채택된다.

[0062] 패킷들을 처리함에 있어서, HSMS 구성 포트에 우선순위가 주어지므로, 연결은 블루 박스 상의 로컬 프로세스에 의해 처리된다. ICMP, UDP, ARP, TCP(HSMS 이외의 포트들) 패킷들은, 목적지 IP가 로컬 테이블내에 존재한다면, 2개의 네트워크 인터페이스들 사이에서 포워딩된다. 따라서, 행동양식(behavior)은 브리징과 유사하지만, 네트워크 내에서 싸이클들(cycles)을 피하기 위해서 테이블 내의 IP 어드레스들에 선택적으로 제한된다. ARP 제한은 더욱 엄격하여, 소스 및 타겟 IP는 포워딩되기 전에 소스 및 타겟 IP 모두가 테이블 내에 존재하여야 하며, 이는 곧 오직 툴과 호스트만이 제어기 네트워크 인터페이스들을 가로질러서 서로간에 ARP를 할 수 있음을 의미한다.

[0063] 제 3 작동 실시예

[0064] 대안적으로, 상기 제어기 또는 중간매개 디바이스는, 직접적으로 통신할 수 있는 디바이스들의 IP 어드레스들을 추정(assume)할 수도 있으며, 패킷들을 적절히 재패키지(repackage)하고 중계할 수도 있다. 이러한 접근법은, 다른 타입의 디바이스들의 전통적인 기능들(functions)하고는 매우 다른데, 왜냐하면 제어기의 네트워크 인터페이스들이 고유의 하드웨어 계층 식별자(hardware layer identifier) 또는 MAC 어드레스를 갖기 때문이다. MAC 어드레스는, 네트워크들을 위해 전형적으로 구현되는 패킷 어드레싱 체계의 일부분이다. 전형적인 네트워크 인터페이스에서, 네트워크 계층(예를 들면, IP)은 데이터 링크 계층(예를 들면, MAC)에 링크되어 있는바. 이는 OSI 모델의 7 계층에 명시된 바와같다. 네트워크 프로토콜, 예를 들면 ARP, 는 네트워크 상의 디바이스들로 하여금 IP 및 MAC 어드레스들을 상관(correlate)하도록 한다. 패킷 전송의 데이터 리던던시(data redundancy) 특징들은, 패킷의 데이터 링크 계층으로부터의 정보를 부분적으로 사용하여 인코딩되는 체크 코드들을 포함한다. 예를 들어, 2개의 네트워크 인터페이스들이 동일한 IP 어드레스를 사용하는 개별적인 네트워크 세그먼트들(segments)에 연결할 때, 이들은 그럼에도 불구하고 별개의 MAC 어드레스들을 갖는다. 제어기가 호스트로부터 툴로 패킷을 중계하기 위해서는, 또는 그 역의 경우도 마찬가지인데, 제어기가 툴의 IP 어드레스 신원(IP address identity)을 차용할 때에, 상기 제어기는, 제어기의 MAC 어드레스 대신에 툴의 MAC 어드레스를 사용하여 인코딩된 올바른 체크 코드들로 상기 패킷을 재패키지 하여야 한다. 하지만, 이것이 오작동 방지의 구현에 있어서 충분한 것은 아니다. 직접적으로 통신할 수 있는 디바이스들은 그들의 네트워크 계층을 데이터 링크 계층 테이블들(예를 들면, ARP 테이블들)로 업데이트 해야 하는데, 이는 페일 오버(fail over) 이후에 유효한 MAC 어드레스를 보정하기 위해서이다. 오작동 방지 모드에서 전술한 오작동 방지 스위치가 툴과 호스트를 직접 연결할 때에는, 패킷들은 호스트로부터 툴로 직접적으로 흐르지 않을 것인데, 왜냐하면 상기 호스트 및 툴이 서로간에 가지고 있는 MAC 어드레스들은 제어기의 MAC 어드레스들을 포함하기 때문이다. 이러한 점을 극복하기 위해서, 적어도 하나의 호스트 또는 툴은 그 자신의 네트워크 계층을 데이터 링크 계층 테이블로 업데이트 해야한다. 상기 호스트, 툴 또는 양쪽 모두는 그들의 테이블들을 자주 업데이트 하도록 적용될 수도 있으며, 또는 그들의 테이블들을 업데이트함으로써 드롭된 세션에 응답하도록 적용될 수도 있다. 파기된 세션(broken session)은 업데이트된 테이블을 가지고 재 설정될 수 있으며 또는 새로운 직접적인 세션이 시작될 수 있다.

[0065] 실시예들에 대한 추가적인 논의

[0066] 본 발명은 방법으로서 실행될 수도 있으며 또는 상기 방법을 실행하는 장치로 적용될 수도 있다. 동일한 방법이 가령 SenseLink 와 같은 스마트 센서 또는 어댑터의 관점으로부터 보여질 수 있으며 또는 가령, 블루박스(Blue Box)와 같은 제어기의 관점으로부터 보여질 수 도 있다. 본 발명은 컴퓨터의 도움을 받는 센서 제어를 수행하기 위한 논리가 수록되어 있는 매체와 같은 제조 물품(article of manufacture) 일 수도 있다.

[0067] 첫번째 방법 실시예는, 중간 매개 디바이스의 고장시에 제 1 및 제 2 디바이스들 사이에서 직접 연결을 투명하게(transparently) 설정하는 방법에 관한 것이다. 이 방법은 제 1 및 제 2 디바이스들을 중간 매개 디바이스의

제 1 및 제 2 네트워크 인터페이스들에 연결하는 것을 포함한다. 상기 제 1 및 제 2 네트워크 인터페이스들은, 상기 제 1 및 제 2 디바이스들과 관련된 데이터 링크 계층 식별자들과는 명백히 다른 데이터 링크 계층 식별자들을 갖는다. 정상 동작 모드에서, 상기 방법은 제 1 및 제 2 디바이스들 사이에서 어드레스된 메시지들을 처리하는 것을 더 포함하며, 반면에 상기 중간 매개 디바이스는 자신의 지속적인 안정 동작(healthy operation)을 오작동 방지 스위치에게 반복적으로 시그널링한다.

[0068] 오작동 방지 스위치에서, 중간 매개 디바이스로부터의 안정 동작에 대한 반복적인 신호가 중단되면, 상기 방법은 중간 매개 디바이스를 바이패스(bypass)하는 데이터 연결을 이용하여 제 1 및 제 2 디바이스들을 직접 연결함으로써 오작동 방지 모드로 스위칭하는 것을 포함한다. 이러한 데이터 연결은 제 1 및 제 2 네트워크 인터페이스들을 바이패스 할 수도 있다. 상기 안정 동작에 대한 반복된 신호들은 연속적인 회로 신호일 수도 있으며 또는 분리된 주기적인 신호들일 수도 있다.

[0069] 제 1 실시예의 일 양상에 따르면, 오작동 방지모드로 스위칭하는 것은 릴레이 클릭(relay click)과 같은 가청 신호를 만드는 것일 수도 있다. 또 다른 양상에 따르면, 제 1 및 제 2 네트워크 인터페이스들은 수신된 모든 패킷들을, 제 1 디바이스 또는 제 2 디바이스 또는 중간 매개 디바이스로 어드레스된 패킷들을 처리하는 필터로 통과시키도록 적응될 수도 있다. 적어도 몇몇 패킷들에 대해서는, 패킷들을 처리한다는 것은, 제 1 디바이스 또는 제 2 디바이스로부터의 패킷들을 제 2 디바이스 또는 제 1 디바이스로 중계하는 것을 포함하며, 중계된 패킷들의 소스 식별 필드(source identification field) 내에서 제 1 또는 제 2 네트워크 인터페이스의 데이터 링크 식별자 대신에 실제 소스의 데이터 링크 식별자를 특정하는 것을 포함한다.

[0070] 실시예의 또 다른 양상은 상기 반복적인 시그널링을 어떻게 구현하는가에 관한 것이다. 이는 반복적으로 메시지를 오작동 방지 스위치로 보내는 중간 매개 디바이스 상에서 구동되는 데몬에 의해 구현될 수도 있으며 또는, 오작동 방지 스위치로부터의 조회(inquiry)를 요청하고 상기 조회에 응답하는 중간 매개 디바이스 상에서 구동되는 데몬에 의해 구현될 수도 있다. 대안적으로, 상기 데몬은 연속적인 회로를 제어할 수 있다.

[0071] 실시예들은 또한, 하나의 물리적인 통신 링크 또는 2개 이상의 물리적인 통신 링크들을 이용하여, 상기 제 1 디바이스에 응답하는 복수개의 제 2 디바이스들을 포함할 수도 있다. 2개 이상의 물리적인 통신 링크들이 이용되는 경우, 하나는 이더넷(Ethernet) 세그먼트가 될 수도 있으며 다른 하나는 직렬 연결일 수도 있다. 상기 직렬 연결은 가령 RS-232 또는 RS 422 와 같은 IEEE 기준을 따를 수도 있다.

[0072] 작동에 있어서, 제 1 및 제 2 디바이스들 사이에서 직접 통신 세션을 설정하는 것은, 네트워크 어드레스를 데이터 링크 계층 어드레스에 상관시키는 제 1 디바이스 또는 제 2 디바이스내의 임의의 테이블을 갱신하는 것을 필요로 하지 않을 수도 있다. 즉, ARP 테이블들은 업데이트될 필요가 없을 수도 있다. IP 어드레스와 MAC 어드레스를 상관시키는 임의의 테이블들을 업데이트하는 것은 필요할 수도 있다. 상기 제 1 디바이스, 제 2 디바이스 및 중간 매개 디바이스는 모두 동일한 네트워크 서브네트워크내에서 작동할 수도 있다.

[0073] 또 다른 방법 실시예는, 호스트 및 툴 사이에 접속된 지능형 제어기가 고장난 이후에 상기 호스트 및 툴 사이에서 통신 링크를 설정하는 페일오버(failover) 방법에 대한 것이다. 이러한 방법은, 다른 경우라면 상기 툴에 의해 사용되는(otherwise used by the tool) 제 1 하드웨어 레벨 식별자를 지능형 제어기의 제 1 포트에서 채택하는 것을 포함하며, 상기 호스트에게 상기 제 1 포트가 상기 제 1 하드웨어 레벨 식별자를 이용하여 어드레스되었다고 공표(advertising)하는 것을 포함한다. 상기 방법은 또한, 다른 경우라면 상기 호스트에 의해 사용되는 제 2 하드웨어 레벨 식별자를 지능형 제어기의 제 2 포트에서 채택하는 것을 포함하며, 상기 툴에게 상기 제 2 포트가 상기 제 2 하드웨어 레벨 식별자를 이용하여 어드레스되었다고 공표하는 것을 포함한다. 또한 상기 방법은, 상기 지능형 제어기에서 상기 제 1 및 제 2 하드웨어 레벨 식별자들의 공표들을 가로채는 것을 포함하는바, 이에 의해 상기 호스트 및 툴이 직접 공표 통신(direct advertising communication)을 하지 않는다. 동작에 있어서, 상기 지능형 제어기는 상기 지능형 제어기가 신뢰되어야 함을 오작동 방지 스위치에게 반복적으로 시그널링한다. 오작동 방지 스위치는 상기 반복적인 시그널링이 실패할 경우 상기 지능형 제어기를 바이패스하는 통신 링크로 상기 호스트 및 상기 툴을 직접적으로 연결하도록 적응된다. 그에 따라 상기 호스트 및 툴은 직접 공표 통신하게 되며 상기 제 1 및 제 2 하드웨어 레벨 식별자들의 공표들은 상기 호스트 및 툴 사이의 통신 링크의 설정을 용이하게 한다.

[0074] 이러한 실시예의 다양한 양상들이 설명된다. 일 양상에 따르면, 상기 제 1 포트를 식별하는 제 1 논리적 어드레스, 상기 툴에 의해 사용되는 제 2 논리적 어드레스, 상기 제 2 포트를 식별하는 제 3 논리적 어드레스, 상기 호스트에 의해 사용되는 제 4 논리적 어드레스가 모두 서로 다를 수도 있다. 이러한 논리적 어드레스들은 IP 어드레스들일 수도 있으며 또는 다른 네트워크 계층 어드레스들일 수도 있다. 상기 하드웨어 레벨 식별자는 MAC

어드레스일 수도 있다. 하드웨어 레벨 식별자의 공표는 표준에 부합하는 ARP 메시지들일 수도 있다.

[0075] 다른 실시예가 또한 상기 툴을 모니터링하는 센서에 적용될 수도 있는바, 상기 지능형 제어기는 상기 호스트 및 상기 센서 사이에 접속된다. 이러한 실시예는, 다른 경우라면 상기 센서에 의해 사용되는 제 3 하드웨어 레벨 식별자를 지능형 제어기의 제 1 포트에서 채택하는 것을 포함하며 상기 호스트에게 상기 제 1 포트가 제 3 하드웨어 레벨 식별자를 사용하여 어드레스 되었다고 공표하는 것을 포함한다. 또한 이러한 실시예는 상기 센서에게 상기 제 2 포트가 제 2 하드웨어 레벨 식별자를 사용하여 어드레스 되었다고 공표하는 것을 포함하며, 지능형 제어기에서 제 3 하드웨어 레벨 식별자의 공표들을 가로채는 것을 포함하는 바, 이에 의해 상기 호스트 및 상기 센서는 직접 공표 통신을 하지 않는다. 이러한 실시예에 따르면, 반복된 시그널링(signaling)이 실패하는 경우, 상기 오작동 방지 스위치는, 상기 지능형 제어기를 바이패스하는 직접 통신 링크를 상기 호스트 및 상기 센서 사이에 설정한다. 이러한 직접 연결은 제 1 포트 및 제 2 포트를 바이패스할 수도 있다. 별개의 논리적 IP 어드레스들, 별개의 네트워크 계층 어드레스들, 하드웨어 식별자 및 표준에 부합하는 ARP 메시지들은 모두 이러한 실시예에 적용될 수도 있다.

[0076] 또 다른 방법 실시예에서는, 적어도 하나의 툴 또는 센서와 호스트 사이의 통신 링크에 페일오버 방지된 지능형 제어기 및 스위치를 도입한다. 이러한 방법은, 상기 스위치를 상기 호스트, 상기 지능형 제어기 및 상기 툴에 접속하는 것을 포함하며, 여기서 상기 스위치가 제 1 상태에 있을 때 상기 호스트 및 상기 툴은 상기 지능형 제어기와 통신하며 직접적으로는 서로 통신하지 않는다. 상기 스위치가 제 2 상태에 있을 때 상기 호스트 및 상기 툴은 직접적으로 서로 통신한다. 상기 제 1 상태에서, 상기 지능형 제어기는 상기 호스트 및 상기 툴 사이의 메시지들을 가로채고, 상기 가로채 메시지들내에서 적어도 하나의 하드웨어 레벨 소스 식별자와 적어도 하나의 하드웨어 레벨 목적지 식별자를 수정하고, 상기 수정된 하드웨어 레벨 식별자들에 대응하도록 상기 가로채 메시지들 내에서 적어도 하나의 에러 보정 코드를 수정하고, 그리고 수정된 하드웨어 레벨 식별자들 및 수정된 보정 코드를 갖는 상기 가로채 메시지들을 중계한다. 상기 제 2 상태에서, 상기 스위치는 상기 호스트와 상기 툴 사이의 직접 통신을 설정함으로써 상기 지능형 제어기를 바이패스시킨다.

[0077] 또 다른 방법 실시예의 일 양상에 따르면, 지능형 제어기는 적어도 하나의 제 1 및 제 2 포트를 포함하며, 제 1 포트는 호스트와 통신하도록 접속되며 상기 제 2 포트는 툴과 통신하도록 접속된다. 이러한 일 양상에서, 제 1 논리적 어드레스는 제 1 포트를 식별하며, 제 2 논리적 어드레스는 툴에 의해 사용되며, 제 3 논리적 어드레스는 제 2 포트를 식별하며, 제 4 논리적 어드레스는 호스트에 의해 사용된다. 이러한 4개의 논리적 어드레스들은 모두 서로 다르다. 이들은 IP 어드레스들일 수도 있으며 또는 다른 네트워크 계층 어드레스들일 수도 있다. 상기 또 다른 방법 실시예에서의 하드웨어 레벨 식별자들은 MAC 어드레스들일 수도 있다.

[0078] 상기 또 다른 방법 실시예는 또한 툴을 모니터링하는 센서에 적용될 수도 있다. 이러한 방법 실시예는 오작동 방지 스위치를 센서에 접속하는 것을 포함하며, 여기서 오작동 방지 스위치가 제 1 상태에 있을 때, 상기 호스트 및 상기 센서는 지능형 제어기와 통신하며 직접적으로는 서로 통신하지 않는다. 상기 오작동 방지 스위치가 제 2 상태에 있을 때, 호스트 및 센서는 서로 직접적으로 통신한다. 상기 제 1 상태에서, 상기 지능형 제어기는, 호스트 및 툴을 위해, 호스트와 센서 사이의 메시지들에 영향을 미친다. 상기 제 2 상태에서, 상기 오작동 방지 스위치는 지능형 제어기를 바이패스 시키며 상기 호스트는 센서와 직접적으로 통신한다. 또 다른 실시예의 다른 양상들은 호스트, 툴, 및 센서 통신들에 적용될 수도 있다.

[0079] 상기 지능형 제어기는 툴 및 센서로의 부착(attachment)을 위해 복수개의 포트들을 포함할 수도 있으며 포트들, 호스트, 툴, 및 센서의 논리적 어드레스들이 모두 다르다. 전술한 바와같이, 이러한 논리적 어드레스들은 IP 어드레스들일 수도 있으며, 또는 네트워크 레벨 식별자들일 수도 있다.

[0080] 상기에서 기술된 모든 방법들은, 상기 중간 매개 디바이스가 고장난 경우에 제 1 및 제 2 디바이스 사이에서 직접적인 연결을 설정하도록 적용된 오작동 방지 스위치를 갖는 중간 매개 디바이스로서 구현될 수도 있다. 일 실시예는 상기 제 1 및 제 2 디바이스와 각각 통신하도록 링크될 수 있는 제 1 및 제 2 네트워크 인터페이스를 갖는 중간 매개 디바이스를 포함하며, 상기 제 1 및 제 2 네트워크 인터페이스는 상기 제 1 디바이스 또는 제 2 디바이스에 관련되는 임의의 데이터 링크 계층 식별자들과는 구별되는 데이터 링크 계층 식별자를 갖는다. 일 실시예는 또한, 상기 중간 매개 디바이스와 통신하며 부수적으로 상기 제 1 및 제 2 디바이스와 링크되어 통신하는 오작동 방지 스위치를 포함한다. 상기 오작동 방지 스위치는 적어도 2개의 모드를 가지며, 예를 들면, 정상 동작모드와 오작동 방지 모드이다. 정상 동작 모드에서, 상기 오작동 방지 스위치는 상기 제 1 및 제 2 네트워크 인터페이스를 이용하여 상기 제 1 및 제 2 디바이스를 연결한다. 오작동 방지 모드에서, 상기 오작동 방지 스위치는 상기 중간 매개 디바이스를 바이패스함으로써, 상기 제 1 및 제 2 디바이스를 직접 연결한다. 이러한

연결은 전기적인 것일 수도 있으며 또는 격리기(isolator)를 통할 수도 있다. 상기 중간 매개 디바이스는, 안정 동작 신호를 오작동 방지 스위치로 반복적으로 통신하도록 적응된 로직 및 리소스들을 더 포함한다. 상기 오작동 방지 스위치는, 오작동 방지 모드로 스위칭함으로써 안정 동작에 대한 상기 반복된 시그널링(signaling)의 중단에 대해 응답하도록 적응된 로직 및 리소스들을 더 포함한다.

[0081] 오작동 방지 모드로 스위칭하는 것에 대한 일 양상에 따르면 가령, 릴레이 클릭과 같은 가청(audible) 신호를 생성하는 것을 포함할 수도 있다. 상기 중간 매개 디바이스는 그 자신의 로직 및 리소스들 내에, 제 1 디바이스 또는 제 2 디바이스 또는 중간 매개 디바이스로 어드레스되는 패킷들을 처리하기 위해 보유(retain)하는 필터를 더 포함할 수도 있으며, 상기 필터는 제 1 및 제 2 네트워크 인터페이스에 접속된다. 이러한 필터는 네트워크 인터페이스로 하여금 뒤범벅된 모드(promiscuous mode)에서 동작하도록 한다. 상기 필터를 구현하는 로직 및 리소스들은, 상기 중간 매개 디바이스내의 프로세서 상에 존재할 수도 있으며 또는 상기 중간 매개 디바이스의 네트워크 인터페이스들 상에 또는 근처에 존재할 수도 있다. 또 다른 양상에 따르면, 상기 중간 매개 디바이스의 로직 및 리소스들은 오작동 방지 스위치로 신호를 전송하는 데몬을 더 포함한다. 이러한 데몬은 전술한 방법들에서 설명된 임의의 모드에서 기능할 수도 있다.

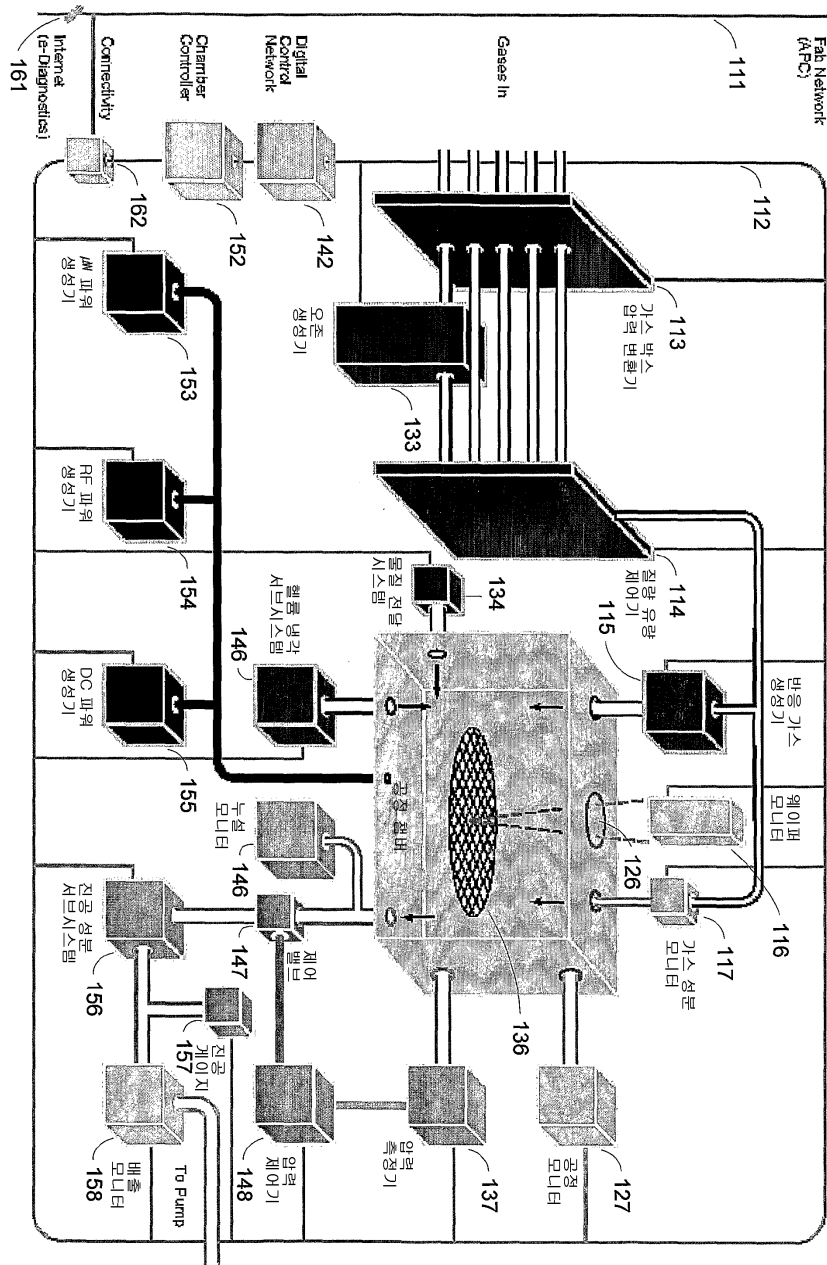
[0082] 비록, 본 발명이 전술한 상세하고 바람직한 실시예들 및 일례들을 참고로 하여 개시되었지만, 이러한 실시예들은 발명을 제한하는 의미라기 보다는 단지 예시적인 의미로 해석되어야 한다. 컴퓨터의 도움을 받는 프로세싱(computer-assisted processing)이 설명된 실시예들에 관련된다. 따라서, 본 발명은, 컴퓨터의 도움을 받는 센싱, 센서 제어를 수행하기 위한 로직(logic)을 포함하는 시스템들, 컴퓨터의 도움을 받는 센서 제어의 이득을 취하는 시스템들, 컴퓨터의 도움을 받는 센서 제어를 수행하기 위한 로직이 수록(impress)된 미디어, 컴퓨터의 도움을 받는 센서 제어를 수행하기 위한 로직이 수록(impress)된 데이터 스트림 또는, 컴퓨터의 도움을 받는 센서 제어를 수행하는 컴퓨터 접속가능한 서비스들 등의 방법들을 통해 구현될 수도 있다. 변형예들 및 조합들은 해당 기술분야의 당업자에게는 자명한 것이며, 이러한 변형예들 및 조합들은 본 발명의 기술적 사상의 범위와 후술될 청구범위의 범위내에 존재할 것이다.

도면의 간단한 설명

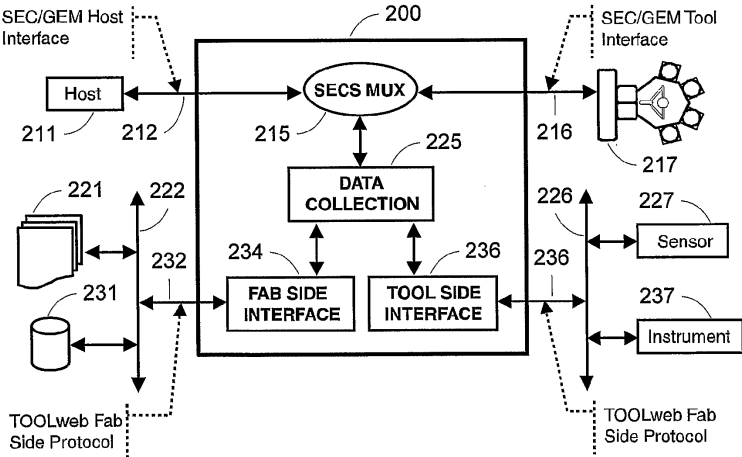
- [0008] 도1은 본 발명의 일 양상이 실제로 사용가능한 환경을 도시한 도면이다.
- [0009] 도2는 툴들, 센서들, 및 툴 호스트와 통신하는 제어기의 블록도이다.
- [0010] 도3은 툴들, 센서들, 및 툴 호스트와 통신하기 위해 싱글 타입 통신 채널을 사용하는 제어기를 도시한 블록도이다.
- [0011] 도4는 멀티 타입 통신 채널들을 사용하는 것을 도시한 도면이다.
- [0012] 도5는 네트워크에 오작동 방지 스위치를 부가한 블록도이다.
- [0013] 도6은 또 다른 네트워크에 오작동 방지 스위치를 부가한 블록도이다.

도면

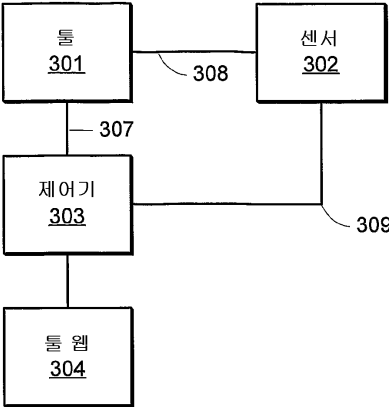
도면1



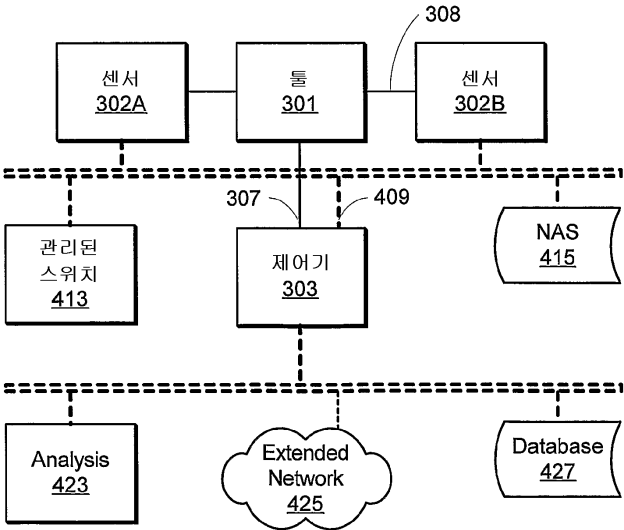
도면2



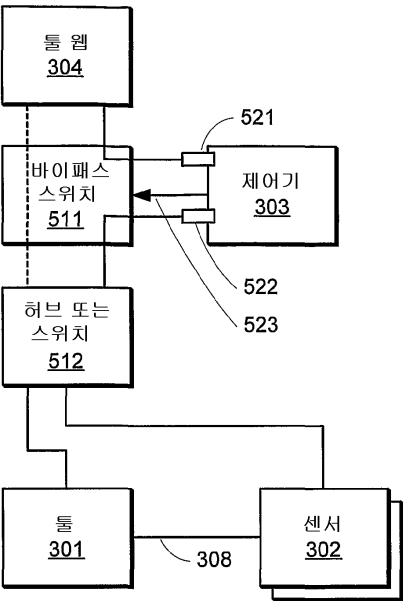
도면3



도면4



도면5



도면6

