(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 698 37 180 T2** 2008.02.21

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 909 056 B1**
(21) Deutsches Aktenzeichen: **698 37 180.1**
(96) Europäisches Aktenzeichen: **98 108 915.4**
(96) Europäischer Anmeldetag: **15.05.1998**
(97) Erstveröffentlichung durch das EPA: **14.04.1999**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **28.02.2007**
(47) Veröffentlichungstag im Patentblatt: **21.02.2008**

(51) Int Cl.[8]: ***H04L 12/24*** (2006.01)
H04L 12/26 (2006.01)

(30) Unionspriorität:
947219 **08.10.1997** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Hewlett-Packard Development Co., L.P., Houston, Tex., US

(74) Vertreter:
Schoppe, Zimmermann, Stöckeler & Zinkler, 82049 Pullach

(72) Erfinder:
Walker, Anthony, Fort Collins, CO 80525, US;
Pulsipher, Eric A., Fort Collins, CO 80525, US;
Smith, Darren D., Fort Collins, CO 80526, US

(54) Bezeichnung: **Korrelation von Netzwerkverwaltungs-Ereignissen in Umgebungen mit inaktiven Netzelementen**

Beschreibung

Gebiet der Erfindung

[0001] Die Erfindung bezieht sich auf Netzwerkverwaltungssysteme und insbesondere auf eine Technik zum Unterscheiden zwischen defekten Netzwerkelementen und Netzwerkelementen, die auf Grund der defekten Elemente unzugänglich sind. Anschließend werden gesammelte Ereignisdaten einem Netzwerkadministrator auf eine einfache, deutliche und handhabbare Weise präsentiert.

Hintergrund der Erfindung

[0002] Netzwerkverwaltungssysteme wie z.B. das Produkt OpenView Network Node Manager sind dahin gehend entworfen, eine Netzwerktopologie (d.h. eine Liste aller Netzwerkelemente in einer Domain, ihren Typ und ihre Verbindungen) zu entdecken, die Gesundheit jedes Netzwerkelements zu überwachen und Probleme dem Netzwerkadministrator zu melden. OpenView Network Node Manager (NNM) ist ein Produkt, das von Hewlett-Packard Company, Palo Alto, Kalifornien, vertrieben wird.

[0003] Die Überwachungsfunktion eines derartigen Systems wird üblicherweise durch ein spezialisiertes Computerprogramm durchgeführt, das jedes Netzwerkelement periodisch abfragt und Daten sammelt, die etwas über die Gesundheit des Netzwerkelements aussagen. Ein Überwachungsprogramm läuft üblicherweise an einem einzelnen Host. Bei verteilten Netzwerken können Überwachungseinrichtungen jedoch an verschiedenen Knoten in dem Netzwerk laufen, wobei jede Überwachungseinrichtung ihre Ergebnisse einer zentralisierten Anzeige meldet.

[0004] Ein Netzwerkadministrator betrachtet eine Präsentation der Netzwerkgesundheit auf der Anzeige. Wenn ein Netzwerkelement ausfällt, identifizieren die dem Netzwerkadministrator präsentierten Informationen im Idealfall Folgendes: 1) welches Element eine Fehlfunktion aufweist; 2) welche anderen Netzwerkelemente durch eine Fehlfunktion beeinflusst werden – d.h. welche funktionstüchtigen Netzwerkelemente auf Grund einer ausfallenden Vorrichtung über das Netzwerk hin unzugänglich sind; und 3) welche unzugänglichen Netzwerkelemente für die Produktivität einer Organisation, die sich auf das Netzwerk stützt, von kritischer Bedeutung sind (somit hat ein Wiederherstellen ihrer Verfügbarkeit für den Netzwerkadministrator eine hohe Priorität).

[0005] Bei vielen handelsüblichen Netzwerkverwaltungsprodukten werden diese drei gesonderten Informationsklassen zu einer Klasse zusammengefügt. Da das Versagen eines einzelnen Netzwerkelements dazu führen kann, das tausende von Elementen (Knoten und Schnittstellen) plötzlich unzugänglich wer den, wird der Netzwerkadministrator (NA) mit Informationen überwältigt. Folglich kann der NA eine beträchtliche Zeit dazu brauchen, die Vielzahl von empfangenen Informationen zu analysieren und die Grundursache des Ausfalls und seine Auswirkung auf die Organisation zu ermitteln.

[0006] Wenn ein Netzwerkelement ausfällt und viele weitere Knoten unzugänglich werden, fährt eine Überwachungseinrichtung üblicherweise fort, sowohl die funktionierenden Knoten als auch die unzugänglichen Knoten abzufragen. Das Überwachen erfolgt üblicherweise unter Verwendung von ICMP-Pings (Internet Control Message Protocol Echo_Request), SNMP-Nachrichten (SNMP = Simple Network Management Protocol) oder IPX-Diagnostikanforderungen. Diese Aktivitäten werden nachfolgend als „Anfragen" bzw. „Pings" bezeichnet. Wenn ein Netzwerkelement zugänglich ist, benötigen diese Anfragen eine Größenordnung von Millisekunden zur Verarbeitung. Wenn jedoch ein Netzwerkelement unzugänglich ist, kann eine Anfrage Sekunden bis zu einer Zeitsperre benötigen.

[0007] Dies führt zu einer Flut von störendem Netzwerkverkehr, und folglich verschlechtert sich die Leistungsfähigkeit eines Netzwerks (z.B. das Überwachungsprogramm kann langsamer ablaufen – bis zu dem Punkt, dass es bezüglich seiner geplanten Abfragen „funktionierender" Knoten in Rückstand gerät. Dies kann zu einer noch weiteren Netzwerkverschlechterung führen.).

[0008] Ein Produkt, das die obigen Probleme zu lösen versucht, ist das von Seagate Software, Scotts Valley, Kalifornien, vertriebene Produkt NerveCenter. Jedoch enthält das Produkt NerveCenter kein Überwachungsprogramm. Ergebnisse werden somit dadurch erreicht, dass der NA dazu gezwungen wird, das Netzwerk unter Verwendung einer anwendereigenen Topologiebeschreibungssprache manuell zu beschreiben. Diese Aufgabe ist für Netzwerke jeglicher praktischen Größe unpraktisch. Ferner verlangen Veränderungen an dem Netzwerk, dass ein NA an der Topologiebeschreibung (manuell) äquivalente Änderungen vornimmt.

[0009] Ein weiteres Produkt, das die obigen Probleme zu lösen versucht, ist der von Hewlett-Packard Company, Palo Alto, Kalifornien, vertriebene OpenView Network Node Manager$_{5.01}$. Versionen des OpenView Network Node Manager vor und einschließlich der Version 5.01 (NNM$_{5.01}$) enthalten ein als netmon bezeichnetes Überwachungsprogramm, das ein Netzwerk, wie es oben beschrieben ist, überwacht. NNM$_{5.01}$ unterstützt Umgebungen, die ein einziges netmon enthalten, und unterstützt ferner verteilte Umgebungen, die mehrere netmon-Prozesse enthalten. In einer verteilten Umgebung laufen eine Mehrzahl von netmon-Prozessen an verschiedenen Sammelstations-Hosts (Collection Station hosts), von denen jeder Topologie- und Statusinformationen an eine zentralisierte Verwaltungsstation (Management Station) (die auf einem anderen Host in dem Netzwerk läuft) kommuniziert, wo Informationen dem NA präsentiert werden.

[0010] Zur Vereinfachung der Beschreibung wird ein Großteil der folgenden Beschreibung im Kontext nicht-verteilter Umgebungen bereitgestellt. Fig. 1 veranschaulicht ein kleines Netzwerk **100** mit netmon, das auf MGR HOST N **110** läuft und unter Verwendung der Netzwerkschnittstelle Nr. 1 von MGR HOST N auf das Netzwerk **100** zugreift. Netmon entdeckt das Netzwerk **100** unter Verwendung von ICMP und SNMP und speichert die Topologie in die Topologiedatenbank **118** (topo DB) durch Dienste, die durch den ovtopmd-Datenbankserver **116** bereitgestellt werden. Die ipmap/ovw-Prozesse **104** sind mit ovtopmd **116** verbunden **106**, und sie wandeln Topologieinformationen in eine graphische Anzeige **108** um, die alle entdeckten Netzwerkelemente, ihre Verbindungen und ihren Status zeigt.

[0011] Netmon ermittelt den Status jedes Netzwerkelements **124**, **128**-**136**, indem es an denselben ein Ping ausführt (z.B. unter Verwendung von ICMP). Wenn eine Ping-Antwort durch ein bestimmtes Netzwerkelement **124** zurückgegeben wird, dann ist das Element Aktiv bzw. Verfügbar (Up). Andernfalls ist das Element **128** Inaktiv bzw. Nicht Verfügbar (Down). Falls das Element **124** Aktiv ist, dann zeigt ipmap/ovw **104** das Element als grün an (durch einen leeren Kreis in Fig. 1, **108**, und in Fig. 3, **302**, vermittelt). Falls das Element **128** Inaktiv ist, wird es als rot angezeigt (durch einen ausgefüllten Kreis in Fig. 1, **108**, und in Fig. 3, **304**, vermittelt). Ferner ist es auch möglich, dass ein Knoten oder eine Schnittstelle einen Unbekannt (Unknown)-Status aufweist und als blau (durch einen unterteilten Kreis in Fig. 3, **306**-**312**, vermittelt) angezeigt wird. Die Fälle, in denen Unbekannt durch eine herkömmliche Netzwerküberwachungseinrichtung verwendet wird, sind rar.

[0012] Zusätzlich zu der Topologieanzeige enthält NNM ein Ereignissystem **114** zur Kommunikation von Knotenstatus-, Schnittstellenstatus- und anderen Informationen unter NNM-Prozessen **120**, **204** und Dritte-Hilfsmitteln **206** (Fig. 2). Diese Ereignisse werden dem NA unter Verwendung des Hilfsmittels xnmevents.web-Ereignisbrowser **120** (als Liste von Ereignissen **122** in einer chronologischen Reihenfolge) angezeigt.

[0013] Bei Fig. 1 wurde die Schnittstelle B.1 des Knotens Router B **128** inaktiv und hat bewirkt, dass die Knoten Router B **128**, Bridge_C **130**, X **132**, Y **134** und Z **136** plötzlich unzugänglich wurden. Dies bewirkt, dass die folgenden Ereignisse durch netmon emittiert werden, wenn es erfasst, dass diese Knoten **128**-**136** und ihre Schnittstellen inaktiv sind.
Schnittstelle C.2 Inaktiv
Schnittstelle C.1 Inaktiv
Schnittstelle B.1 Inaktiv
Schnittstelle B.2 Inaktiv
Schnittstelle Z.1 Inaktiv
Schnittstelle Y.1 Inaktiv
Schnittstelle X.1 Inaktiv

[0014] Man beachte, dass die Schnittstelle-Inaktiv-Ereignisse in der zufälligen Reihenfolge, in der netmon die Schnittstellen abfragt, emittiert werden. Dies verstärkt die Schwierigkeit des NA beim Ermitteln der Ursache eines Ausfalls unter Verwendung des Ereignisse-Browsers. Der Status jedes Knotens **124**, **128**-**136** und jeder Schnittstelle wird auch auf dem ovw-Bildschirm **108** angezeigt. Wie zuvor erwähnt wurde, werden alle unzugänglichen Knoten und Schnittstellen in der Farbe rot (d.h. einem ausgefüllten Kreis) angezeigt.

[0015] In einem echten Netzwerk mit tausenden von Knoten auf der anderen Seite des Router_B **128** ermöglicht keine Anzeige (ovw **108** oder xnmevents.web **120**) dem NA, die Ursache eines Versagens und die Dringlichkeit des Wiederbelebens entscheidender Knoten in einem kurzen Zeitraum zu ermitteln. Außerdem weist dieses System **100** die zuvor beschriebenen Verschlechterungen der Netzwerkleistungsfähigkeit auf, da netmon weiterhin unzugängliche Knoten **130**-**136** abfragt.

[0016] Die US-A-5,436,909 offenbart ein Netzwerkverwaltungssystem, das eine Benutzerschnittstelle, ein virtuelles Netzwerk und eine Vorrichtungskommunikationsverwaltungseinrichtung umfasst. Das virtuelle Netz-

werk umfasst Modelle, die Netzwerkentitäten darstellen und Beziehungen modellieren, die Beziehungen zwischen Netzwerkentitäten darstellen. Jedes Modell umfasst Netzwerkdaten, die sich auf eine entsprechende Netzwerkentität beziehen, und eine oder mehrere Störungshandhabungseinrichtungen zum Verarbeiten der Netzwerkdaten, um Benutzerinformationen zu liefern. Das System führt eine Fehlereingrenzungstechnik durch, bei der der Fehlerstatus einer Netzwerkvorrichtung unterdrückt wird, wenn ermittelt wird, dass die Vorrichtung nicht defekt ist. Benutzeranzeigen umfassen hierarchische Positionierungsansichten und topologische Ansichten der Netzwerkkonfiguration. Netzwerkvorrichtungen werden auf den Anzeigen durch Multifunktions-Ikone dargestellt, die es dem Benutzer ermöglichen, zusätzliche Anzeigen auszuwählen, die ausführliche Informationen bezüglich unterschiedlicher Aspekte der entsprechenden Netzwerkvorrichtung zeigen.

[0017] Die Aufgabe der vorliegenden Erfindung besteht darin, Probleme mit Netzwerkelementen auf eine Weise zu präsentieren, die die Grundursache eines Problems deutlich anzeigt, wobei es einem NA ermöglicht wird, rasch zu beginnen, an einer Lösung des Problems zu arbeiten.

[0018] Diese Aufgabe wird durch eine Netzwerküberwachungseinrichtung gemäß Anspruch 1, durch eine Netzwerkverwaltungseinrichtung gemäß Anspruch 3 und durch ein Verfahren gemäß Anspruch 6 gelöst.

[0019] Die vorliegende Erfindung liefert ein System und ein Verfahren zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen.

[0020] Ferner liefert die vorliegende Erfindung eine Einrichtung zum Unterdrücken und/oder Korrelieren von Netzwerkereignissen, um 1) die Informationsschwemme, die ein NA auf ein Ausfallen eines Netzwerkelements hin empfängt, zu reduzieren, und 2) eine Einrichtung zu liefern, anhand derer der NA unterdrückte Informationen auf geordnete Weise betrachten kann.

[0021] Außerdem stattet diese Erfindung einen NA mit einer Netzwerküberwachungseinrichtung aus, die sehr stark an Kundenanforderungen anpassbar ist, wodurch eine Anzahl von Formaten zum Betrachten von Informationen geliefert werden.

Zusammenfassung der Erfindung

[0022] Bei der Lösung der vorstehenden Aufgaben haben die Erfinder eine Netzwerküberwachungseinrichtung zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen ersonnen. Die Netzwerküberwachungseinrichtung umfasst ein oder mehrere computerlesbare Speicherungsmedien (z.B. CD-ROM, Floppy-Disk, Magnetband, Festplattenlaufwerk usw.) und einen computerlesbaren Programmcode, der in dem einen oder den mehreren computerlesbaren Speicherungsmedien gespeichert ist. Der computerlesbare Programmcode umfasst 1) einen Code zum Entdecken der Topologie einer Mehrzahl von Netzwerkelementen, 2) einen Code zum periodischen Abfragen einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen zugeordnet sind, 3) einen Code zum Berechnen oder Validieren eines kritische-Route-Attributs (criticalRoute-Attributs) für jede der Mehrzahl von Netzwerkschnittstellen, und 4) einen Code zum Analysieren eines Status von Netzwerkschnittstellen, die durch das kritische-Route-Attribut einer betreffenden Schnittstelle (IIQ – interface in question), die nicht auf eine Abfrage reagiert, identifiziert wurden. Elemente des Codes zum Entdecken der Topologie einer Mehrzahl von Netzwerkelementen und des Codes zum periodischen Abfragen einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen zugeordnet sind, sind in der US-Patentschrift 5,185,860 von Wu mit dem Titel „Automatic Discovery of Network Elements" und in der US-Patentschrift 5,276,789 von Besaw et al. mit dem Titel „Graphic Display of Network Topology" offenbart.

[0023] Die oben beschriebene Netzwerküberwachungseinrichtung (und Systeme und Verfahren zum Verwenden derselben) liefern viele Vorteile im Vergleich zu bisherigen Implementierungen von Netzwerküberwachungseinrichtungen.

[0024] Ein erster Vorteil ist eine automatische Topologie. Um die Grundursache eines Netzwerkversagens ordnungsgemäß zu identifizieren, erfordert ein Netzwerkausfall 1) eine Eingabe von einem topologischen Modell des Netzwerks, und 2) den aktuellen Status jedes Elements in dem Netzwerk. Bisher musste der NA diese Topologie manuell beschreiben. Der hierin offenbarte Entwurf verwendet Topologie- und Statusinformationen, die bereits erzeugt wurden.

[0025] Ein zweiter Vorteil ist eine topologische Anzeige. Eine topologische graphische Anzeige von Informationen ist viel hilfreicher beim Unterstützen eines Netzwerkadministrators in seinen Bemühungen, die Grund-

ursache eines Netzwerkausfalls zu identifizieren und seine Prioritäten zu setzen. Die graphische Anzeige (ovw), die bei dem bevorzugten Ausführungsbeispiel der Erfindung verwendet wird, präsentiert deutlich einen Netzwerkelementstatus in drei Kategorien:

- Funktionierende Knoten und Schnittstellen werden in grün angezeigt, um einen Aktiv-Status anzugeben.
- Grundursachen-Ausfälle und unzugängliche Schnittstellen an kritischen Serverknoten werden in rot angezeigt, um einen Inaktiv-Status anzuzeigen.
- Nicht-entscheidende, unzugängliche Schnittstellen werden in blau angezeigt, um einen Unbekannt-Status anzugeben.

[0026] Ein dritter Vorteil ist eine neue Art und Weise, Ereignisse anzuzeigen. Die Anzeige des Ereignisbrowsers **120** (Fig. 1) von Knoten- und Schnittstellenstatusinformationen **122** ist viel sinnvoller beim Unterstützen des Netzwerkadministrators in seinen Bemühungen, die Grundursache eines Netzwerkausfalls zu identifizieren und Prioritäten zu setzen. Sekundärausfallereignisse (d.h. Ereignisse, die auf unzugängliche Schnittstellen hinweisen) werden nicht mit Primärausfällen (d.h. Ereignissen, die auf tatsächlich ausgefallene Schnittstellen hinweisen) und unzugänglichen kritischen Knoten angezeigt. Sekundärausfälle sind über einen „Drill-Down"-Prozess beobachtbar.

[0027] Ein vierter Vorteil ist die Netzwerkleistungsfähigkeit. Bei vergangenen Entwürfen nimmt die Netzwerkleistungsfähigkeit ab, wenn ein Versagensfall auftritt, da auf Grund fehlgeschlagener Anfragen nach unzugänglichen Netzwerkelementen viel mehr Netzwerkverwaltungsnachrichten emittiert werden. Außerdem geraten die Netzwerküberwachungsprozesse bezüglich ihres Zeitplans in Verzug, da Ausfälle zu Auszeiten führen, die viel langsamer sind als erfolgreiche Anfragen. Der vorliegende Entwurf umfasst Rückkopplungs-Abfragealgorithmen (backoff polling algorithms) für einen Netzwerküberwachungsdienst, um diese Situationen zu vermeiden, wann immer es möglich ist.

[0028] Ein fünfter Vorteil ist eine Fähigkeit, Netzwerkelemente zu klassifizieren. Netzwerkelemente sind in zwei Kategorien unterteilt, regulär und kritisch, um dem System zu helfen, dem NA auf eine Weise Informationen anzuzeigen, die seine Prioritäten widerspiegelt. Dies wird bewerkstelligt, indem ein Mechanismus bereitgestellt wird, mit dem der Netzwerkadministrator ein „Filter" definiert, das Router, wichtige Server und andere Netzwerkelemente, die der NA als wichtig erachtet, beschreibt.

[0029] Ein sechster Vorteil ist die Skalierbarkeit. Andere Systeme sind bezüglich der Architektur zentralisiert und sind nicht zu großen Unternehmensnetzwerken skalierbar. Die vorliegende Implementierung baut auf der verteilten OpenView-Architektur auf und liefert eine neue Funktionalität für „große" Netzwerke.

[0030] Ein siebter Vorteil ist eine Fähigkeit, willkürliche Topologien zu handhaben. Algorithmen, die versuchen, eine Grundursache eines Netzwerkelements zu finden, werden auf Grund der Komplexität bei Kundennetzwerkkonfigurationen (z.B. können sie Schleifen und ein dynamisches Routen enthalten) oft hinters Licht geführt. Algorithmen in der vorliegenden Implementierung sind „Bottoms up" bzw. von unten nach oben programmiert, und sie werden nicht dahin gehend hinters Licht geführt, zu denken, dass ein Netzwerkelement inaktiv ist, wenn ein redundanter Router versagt.

[0031] Ein achter Vorteil besteht darin, dass das System extrem konfigurierbar ist. Ein Netzwerkadministrator darf somit Kompromisse machen, die sein Netzwerk, seinen Arbeitsstil und seine Leistungsfähigkeit optimieren. Andere Systeme sind tendenziell manuell konfigurierbar und/oder starr.

[0032] Ein letzter Vorteil ist ein „Ereignisordnen". Während Netzwerkausfallssituationen wird die Verwirrung durch Implementierungen, die unzugängliche Netzwerkelemente in willkürlicher Reihenfolge entdecken, noch verstärkt. Die vorliegende Implementierung enthält neue Warteschlangenalgorithmen, um Ausfälle in einer vorhersehbaren Reihenfolge zu entdecken. Diese Vorhersehbarkeit ist sowohl für den Netzwerkadministrator als auch andere Ereigniskorrelationsprozesse, die ein Benutzer oder Dritte einrichten könnten, hilfreich.

[0033] Diese und weitere wichtige Vorteile und Ziele der vorliegenden Erfindung werden nachfolgend in der beiliegenden Beschreibung, den beiliegenden Zeichnungen und den beiliegenden Patentansprüchen näher erläutert oder werden auf Grund derselben offensichtlich.

Kurze Beschreibung der Zeichnungen

[0034] Ein veranschaulichendes und derzeit bevorzugtes Ausführungsbeispiel der Erfindung ist in den Zeichnungen veranschaulicht, bei denen:

[0035] Fig. 1 ein Blockdiagramm einer $NNM_{5.01}$-Netzwerkadministratoranzeige ist;

[0036] Fig. 2 ein Blockdiagramm eines $NNM_{5.01}$-Ereignisverteilungssystems ist;

[0037] Fig. 3 eine graphische Anzeige der Netzwerkelementgesundheit ist;

[0038] Fig. 4 ein Blockdiagramm eines bevorzugten Ereignisverteilungssystems ist;

[0039] Fig. 5 ein Blockdiagramm einer bevorzugten Netzwerkadministratoranzeige in einer [<>,Down,Unknown,True]-Konfiguration ([<>,inaktiv,unbekannt,wahr]-Konfiguration) ist;

[0040] Fig. 6 ein Flussdiagramm ist, das die Funktionsweise einer ECS-Router-Inaktiv-Schaltung veranschaulicht;

[0041] Fig. 7 ein Blockdiagramm einer bevorzugten Netzwerkadministratoranzeige in einer [<serverFilter>,Down,Unknown,True]-Konfiguration ([<server-Filter>,inaktiv,unbekannt,wahr]-Konfiguration) ist; und

[0042] Fig. 8 ein Blockdiagramm einer bevorzugten Netzwerkadministratoranzeige in einer [<serverFilter>,Unknown,Ignore,True]-Konfiguration ([<serverFilter>,inaktiv,unbekannt,ignorieren,wahr]-Konfiguration) ist.

Beschreibung des bevorzugten Ausführungsbeispiels

[0043] Eine Netzwerkverwaltungsvorrichtung zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen in einem Netzwerk und zum Präsentieren dieser Informationen gegenüber einem Netzwerkadministrator in einem leicht verständlichen Format ist in Fig. 5, Fig. 7 und Fig. 8 gezeigt. Die Vorrichtung umfasst allgemein einen Anzeigeprozess **104**, **120** und eine Netzwerküberwachungseinrichtung **110**, die mittels eines oder mehrerer Ereignisbusse **114** verbunden sind. Die Netzwerküberwachungseinrichtung **110** umfasst eine Einrichtung zum Entdecken der Topologie einer Mehrzahl von mit derselben verbundenen Netzwerkelementen **124**, **128**-**136**, eine Einrichtung zum periodischen Abfragen einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen **124**, **128**-**136** zugeordnet sind, eine Einrichtung zum Berechnen oder Validieren eines kritischeRoute-Attributs für jede der Mehrzahl von Netzwerkschnittstellen und eine Einrichtung zum Analysieren des Status von Netzwerkschnittstellen (z.B. N.1, A.1, A.2, B.1, B.2, C.1, C.2, X.1, Y.1, Z.1), die durch das kritischeRoute-Attribut einer betreffenden Schnittstelle (IIQ), die nicht auf eine Abfrage antwortet, identifiziert werden.

[0044] Desgleichen kann ein computerimplementiertes Verfahren zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen und zum Präsentieren dieser Informationen gegenüber einem Netzwerkadministrator in einem leicht verständlichen Format die Schritte des 1) Entdeckens der Topologie einer Mehrzahl von Netzwerkelementen **124**, **128**-**136**, 2) des periodischen Abfragens einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen **124**, **128**-**136** zugeordnet sind, 3) des Berechnens oder Validierens eines kritischeRoute-Attributs für jede der Mehrzahl von Netzwerkschnittstellen, und 4) des Analysierens des Status von Netzwerkschnittstellen, die durch das kritischeRoute-Attribut einer betreffenden Schnittstelle (IIQ), die nicht auf eine Abfrage antwortet, identifiziert werden, umfassen. Nachdem ein Verfahren und eine Vorrichtung zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen im Allgemeinen beschrieben wurden, werden das Verfahren und die Vorrichtung nun ausführlicher beschrieben.

[0045] Das bevorzugte Ausführungsbeispiel der Erfindung ist dahin gehend entworfen, in Verbindung mit dem Produkt OpenView Network Node Manager für allein stehende und verteilte Umgebungen zu arbeiten (hiernach „NNM"). Der OpenView Network Node Manager ist ein Produkt, das von Hewlett-Packard Company, Palo Alto, Kalifornien, vertrieben wird. Dieses Produkt wird in einer Reihe von Endbenutzerhandbüchern, die durch HP-Teilenummern J1136-90000, J1136-90001, J1136-90002, J1136-90004 und J1136-90005 identifiziert sind, und in einer Reihe von Entwicklerhandbüchern, die durch HP-Teilenummern J1150-90001, J1150-90002, J1150-90003 und J1150-90005 identifiziert sind, ausführlich beschrieben.

[0046] Der erste Teil dieser Beschreibung erörtert, wie das System in einer nicht verteilten Umgebung arbeitet. Eine nicht verteilte Umgebung ist eine Umgebung, die aus einer Verwaltungsstation **110** und keinen Sammelstationen besteht, wobei ein NNM-netmon-Prozess an der Verwaltungsstation **110** läuft. Die ovw- und ovevents.web-NA-Anzeige läuft an der Verwaltungsstation **110**.

Das kritischeRoute-Attribut

[0047] Netmon entdeckt die Topologie eines Netzwerkes **100** genau wie bei $NNM_{5.01}$. Jedoch berechnet netmon während Netmons Konfigurationsabfrage (die üblicherweise einmal pro Tag erfolgt) und während der ersten Statusabfrage jedes Knotens (nachdem netmon zu laufen beginnt) ein als kritischeRoute (critical route) bezeichnetes Pro-Netzwerkschnittstelle-Attribut.

[0048] Das kritischeRoute-Attribut ist eine Sequenz von ovtopmd-DB-Objektidentifizierern, die der Route entsprechen, die ein Netzwerkpaket nehmen könnte, wenn es von netmon an eine bestimmte Schnittstelle gesendet würde. Das kritische-Route-Attribut verfolgt den Pfad der dazwischen liegenden Netzwerkschnittstellen.

[0049] Die folgende Liste zählt die kritischeRoute-Werte für jede Netzwerkschnittstelle in Fig. 1 auf.

Netzwerkschnitt-stelle	Kritische Route
N.1	N.1
A.1	N.1, A.1
A.2	N.1, A.1, A.2
B.1	N.1, A.1, A.2, B.1
B.2	N.1, A.1, A.2, B.1, B.2
C.1	N.1, A.1, A.2, B.1, B.2, C.1
C.2	N.1, A.1, A.2, B.1, B.2, C.1, C.2
X.1	N.1, A.1, A.2, B.1, B.2, C.1, C.2, X.1
Y.1	N.1, A.1, A.2, B.1, B.2, C.1, C.2, Y.1
Z.1	N.1, A.1, A.2, B.1, B.2, C.1, C.2, Z.1

[0050] KritischeRoute kann sogar dann berechnet werden, wenn das Netzwerk Schleifen enthält, da zu dem Zeitpunkt, da die Berechnung durchgeführt wird, lediglich eine Route vorliegt, die ein Paket nehmen würde. Bei der Berechnung der kritischeRoute wird, wenn mehrere Möglichkeiten existieren, Routen in demselben Netzwerk oder Teilnetz vor Routen, die außerhalb des Netzwerks verlaufen, Vorrang gegeben. Vorrang wird auch Routen, die Routerknoten enthalten, vor anderen Nicht-Router-„Multihomed"-Knoten gegeben.

PrimärAusfälle (primaryFailures) gegenüber SekundärAusfällen (secondaryFailures)

[0051] Nachdem für jede Netzwerkschnittstelle ein kritischeRoute-PrimärAusfall-Schnittstellen von SekundärAusfall-Schnittstellen zu unterscheiden. Gemäß der Verwendung hierin ist eine PrimärAusfall-Schnittstelle eine Schnittstelle, die ausgefallen ist, wohingegen eine SekundärAusfall-Schnittstelle eine Schnittstelle ist, die auf Grund einer ausgefallenen Schnittstelle unzugänglich ist.

[0052] Angenommen, dass netmon Netzwerkschnittstellen in derselben Reihenfolge abfragt, wie die Ereignisse in dem Ereignisbrowser **120** in Fig. 1 angezeigt sind. Das heißt:

Schnittstelle C.2
Schnittstelle C.1
Schnittstelle B.1
Schnittstelle B.2
Schnittstelle Z.1
Schnittstelle Y.1
Schnittstelle X.1

[0053] Vor der Statusabfrage der Schnittstellen des Knotens BRIDGE_C sind alle Knoten **124**, **126**-**136** und Schnittstellen Aktiv, und sie werden in der Farbe grün auf der ovw-Abbildung **104**/**108** angezeigt. In dem Ereignisbrowser **120** befinden sich keine Schnittstelle-Inaktiv-Ereignisse. Wenn netmons Ping der Schnittstelle C.2 eine Zeitbegrenzung auslöst, weiß netmon, dass die Schnittstelle C.2 unzugänglich ist. Es weiß noch nicht, ob die Schnittstelle C.2 unzugänglich ist, weil sie auf Grund eines Hardware-/Softwareausfalls physisch inaktiv ist oder weil eine Verbindungsschnittstelle inaktiv ist.

[0054] Bei $NNM_{5.01}$ würde netmon einfach den Status der Schnittstelle unter Verwendung der ovtopmd-API **116** auf Kritisch (Critical) einstellen. Ovtopmd **116** würde dann den Status der Schnittstelle in der Topologiedatenbank **118** ändern und das Schnittstellelnaktiv-Ereignis (interfaceDown-Ereignis) aussenden.

[0055] Als Teil des aktuellen Verfahrens analysiert netmon den Status von Schnittstellen entlang der kritische-Route für die betreffende Schnittstelle (IIQ) und versucht, zu ermitteln, welche Schnittstelle den Hardware-/Softwareausfall enthält. Wie zuvor erwähnt wurde, wird die Schnittstelle, die auf Grund ihres eigenen HW/SW-Ausfalls unzugänglich ist, als PrimärAusfall-Schnittstelle betrachtet. Die Schnittstellen, die auf Grund des PrimärAusfalls unzugänglich sind, werden als SekundärAusfall-Schnittstellen betrachtet. Bei diesem Szenario haben wir die folgende Klassifizierung von Schnittstellen (unter der Annahme, dass Schnittstelle B.1 ausgefallen ist):

Schnittstelle	Klassifizierung des Ausfalls
N.1	Weist keinen Ausfall auf.
A.1	Weist keinen Ausfall auf.
A.2	Weist keinen Ausfall auf.
B.1	Primärer Ausfall.
B.2	Sekundärer Ausfall.
C.1	Sekundärer Ausfall.
C.2	Sekundärer Ausfall.
X.1	Sekundärer Ausfall.
Y.1	Sekundärer Ausfall.
Z.1	Sekundärer Ausfall.

Vor-kritischeRouteWarteListe-Klassifizierungsalgorithmus

[0056] Wenn netmons Ping der Schnittstelle C.2 eine Zeitbegrenzung auslöst, untersucht netmon den Im-Speicher-Status jeder Schnittstelle entlang des kritischeRoute-Pfades für die IIQ (Schnittstelle C.2). Wenn irgendeine Schnittstelle entlang dieses Pfades Inaktiv (Kritisch) ist, dann ist die IIQ eine SekundärAusfall-Schnittstelle. Falls die IIQ eine SekundärAusfall-Schnittstelle ist, ändert netmon ihren Status unter Verwendung des libtopm changeSecondaryFailureStatus()-API-Aufrufs zu Unbekannt.

[0057] Ovtopmd **116** verändert den Status der Schnittstelle in der Topologiedatenbank **116** und emittiert eine spezielle SekundärAusfallSchnittstelleInaktiv-Nachricht, wie im Abschnitt 3.1.6 beschrieben ist. Dann stellt netmon die interne Darstellung der Schnittstelle in eine Langsamer-Ping-Liste (slowPingList) und setzt seine normale Statusabfrageverarbeitung anderer Schnittstellen in dem Netzwerk fort.

kritischeRouteWarteListe-Klassifizierungsalgorithmus

[0058] Falls der pre-kritischeRouteWarteListe-Schnittstellenausfallsklassifizierungsalgorithmus nicht in der Lage ist, eine Schnittstelle (die nicht die IIQ ist), die bereits inaktiv ist, entlang der kritischeRoute zu finden, so muss der Status aller Schnittstellen entlang der kritischeRoute(IIQ) überprüft werden, um zu gewährleisten, dass eine dieser Schnittstellen nicht ausgefallen ist, seit sie zum letzten Mal geprüft wurde.

[0059] Um dies zu erleichtern, sichert netmon die Tatsache, dass die Schnittstelle C.2 unzugänglich ist, indem es die Darstellung dieser IIQ aus der normalen PingListe zu einer als kritischeRouteWarteListe bezeichneten neuen Warteschlange verschiebt. Diese Liste weist die folgenden Charakteristika auf:

- Eine beliebige Anzahl von Schnittstellen kann auf der Liste platziert sein.
- Lediglich die erste Schnittstelle auf der Liste wird durch den kritischeRouteWarteListe-Algorithmus verarbeitet. Alle anderen Schnittstellen werden einfach beibehalten.
- Wenn sich eine Schnittstelle auf dieser Liste befindet, befindet sie sich auf keiner anderen Liste. Dies verhindert ein Verarbeiten dieser Schnittstelle durch andere netmon-Aktivitäten.
- Diese Liste weist Datenstrukturen auf, um ein sequentielles Durchgehen der kritischeRouteWarte-Liste(IIQ) zu ermöglichen. Dies ist auf Grund der verketteten Beschaffenheit von netmon wichtig. Nachdem es ein Ping an eine bestimmte Schnittstelle auf der kritischeRoute(IIQ) gesendet hat, erfüllt netmon andere Aufgaben, während es darauf wartet, dass die Ping-Antwort zurückkehrt oder dass eine Zeitbegrenzung erfolgt.

[0060] Netmon befragt jede Schnittstelle entlang der kritische-Route im Namen der betreffenden Schnittstelle (IIQ), indem es einen Ping sendet, und zwar jede Schnittstelle einzeln, wobei mit der Schnittstelle des netmon-Knotens (an der Verwaltungsstation **110**) begonnen wird, bis es eine Schnittstelle findet, die inaktiv ist, oder bis es wieder an der unzugänglichen Schnittstelle der IIQ ankommt.

[0061] Wenn eine Schnittstelle (die nicht die IIQ ist) Inaktiv ist, dann wird sie als PrimärAusfall-Schnittstelle

verarbeitet. Der Status der Schnittstelle wird unter Verwendung der $NNM_{5.01}$ libtopm-API zu Kritisch verändert. Ovtopmd verändert den Status der Schnittstelle in der Topologiedatenbank **118** und emittiert eine spezielle PrimärAusfallSchnittstellelnaktiv-Nachricht. Dann bewegt netmon die interne Darstellung der Schnittstelle von der kritischeRouteWarte-Liste zu der Langsamer-Ping-Liste und setzt seine normale Statusabfrageverarbeitung anderer Schnittstellen in dem Netzwerk **100** fort.

[0062] Wenn ein PrimärAusfall (der nicht die IIQ ist) gefunden wird, dann ist der Ausfall der IIQ ein SekundärAusfall und wird so verarbeitet, wie dies oben für SekundärAusfälle beschrieben wurde. Netmon verändert den Zustand der IIQ unter Verwendung des libtopm SekundänAusfall()-API-Aufrufs zu Unbekannt. Ovtopmd **116** verändert den Status der Schnittstelle in der Topologiedatenbank **118** und emittiert eine spezielle SekundärAusfallSchnittstelleInaktiv-Nachricht, wie nachfolgend beschrieben wird. Dann setzt netmon die interne Darstellung der Schnittstelle auf die Langsamer-Ping-Liste und setzt seine normale Statusabfrageverarbeitung anderer Schnittstellen in dem Netzwerk **100** fort.

[0063] Wenn keine PrimärAusfall-Schnittstelle entlang der kritischeRoute(IIQ) gefunden werden kann, kehrt die kritische-RouteWarteListe-Verarbeitung schließlich zu der IIQ-Schnittstelle zurück. Wenn dies eintritt, wissen wir, dass die IIQ ein PrimärAusfall ist. Für die IIQ folgt eine reguläre PrimärAusfall-Verarbeitung.

[0064] Der Status der Schnittstelle wird unter Verwendung der $NNM_{5.01}$ libtopm-API zu Kritisch geändert. Ovtopmd **116** verändert den Status der Schnittstelle in der Topologiedatenbank **118** und emittiert eine spezielle PrimärAusfallSchnittstellelnaktiv-Nachricht, wie nachfolgend beschrieben wird. Dann verschiebt netmon die interne Darstellung der Schnittstelle aus der kritischeRouteWarteListe zu der Langsamer-Ping-Liste und setzt seine normale Statusabfrageverarbeitung anderer Schnittstellen in dem Netzwerk fort.

Entnahme von Daten aus der kritischeRouteWarteListe

[0065] Während die kritischeRouteWarteListe verarbeitet wird, fährt netmon fort, andere Schnittstellen in dem Netzwerk **100** gemäß der Steuerung durch die PingListe abzufragen. Manche von diesen können unzugänglich sein und durch die obigen Algorithmen auf der kritischeRouteWarteListe landen. Viele dieser Schnittstellen könnten SekundärAusfall-Schnittstellen sein, die auf dieselbe PrimärAusfall-Schnittstelle zurückzuführen sind. Es wäre sehr ineffizient, den Status der gesamten kritischeRoute für jede SekundärAusfall-Schnittstelle zu überprüfen.

[0066] Zu dem Zeitpunkt, zu dem die erste SekundärAusfall-Schnittstelle identifiziert und verarbeitet wurde, wurde auch die PrimärAusfall-Schnittstelle identifiziert und verarbeitet. Dies bedeutet, dass es nun möglich ist, zu ermitteln, ob die neue IIQ (IIQ2) eine PrimärAusfall-Schnittstelle oder eine SekundärAusfall-Schnittstelle ist, indem der Im-Speicher-Status jeder Schnittstelle entlang der kritischeRoute (IIQ2) unter Verwendung des VorkritischeRouteWarteListe-Klassifizierungsalgorithmus untersucht wird, und eine Zeitverschwendung durch ein Senden zusätzlicher Pings zu vermeiden.

[0067] Wenn ermittelt werden kann, dass IIQ2 ein SekundärAusfall ist, dann erfolgt eine SekundärAusfall-Verarbeitung, die Schnittstelle wird aus der kritischeRouteWarteListe entnommen und in die Langsamer-Ping-Liste platziert, und keine Überprüfung des kritischeRoute-Status ist erforderlich. Andernfalls wird die Verarbeitung für die IIQ2 auf ähnliche Weise wie eine Verarbeitung der IIQ fortgesetzt.

Schnittstelle-Inaktiv-Ereignisse für PrimärAusfälle und SekundärAusfälle

[0068] Die obigen Erläuterungen legen nahe, dass die ovtopmd-Hintergrundroutine **116** für die Veränderung des Status der Netzwerkelemente **124**, **128**-**136** in der Topologiedatenbank **118** und für das Aussenden darauf bezogener Ereignisse verantwortlich ist. Sie tut dies im Namen anderer Prozesse, wenn diese sie anweisen, die liptopm-API zu verwenden.

[0069] Es erfolgt keine Änderung dieser API für PrimärAusfall-Ereignisse, und PrimärAusfall-Ereignisse verwenden das $NNM_{5.01}$-Ereignisformat ohne Änderung. Jedoch müssen für SekundärAusfall-Ereignisse Informationen über den SekundärAusfall kommuniziert werden, und außerdem muss die Primär-Ausfall-Schnittstelle identifiziert werden. Dieses zusätzliche Erfordernis ist notwendig, so dass ein Ereigniskorrelationssystem wie z.B. ECS **408** (Fig. 4; von Hewlett-Packard Company, Palo Alto, Kalifornien, vertrieben) zwischen den zwei Arten von Ereignissen unterscheiden und dieselben korrelieren und/oder unterdrücken kann.

[0070] Dies wird bewerkstelligt, indem ein zusätzliches Varbind (SNMP-Varbinds sind ausführlich in „Simple

Book" von Marshall Rose beschrieben) zu dem regulären PrimärAusfall-Ereignisformat hinzugefügt wird. Dieses zusätzliche Varbind wird als ...PrimärAusfallUuid bezeichnet und enthält den Ereignis-UUID (Universally Unique IDentifier (universell einzigartiger Identifizierer) ist eine Kennung, die für jedes Ereignis einzigartig ist) des entsprechenden PrimärAusfall-Ereignisses. UUID ist eine Kennung, die für jedes Ereignis über jeglichen Computer in einem Netzwerk hinweg einzigartig ist. Es gibt einen separaten API-Aufruf, SekundärAusfallStatusVerändern (changeSecondaryFailureStatus)(), in libtopm, der zum Verändern des Status des Netzwerkelements verwendet wird. Die Parameter des API-Aufrufs sind identisch mit dem für PrimärAusfälle verwendeten Aufruf, plus die Hinzufügung des ovwDbId des PrimärAusfall-Netzwerkelements (und das Verhalten ist anders).

[0071] Wenn der PrimärAusfall-API-Aufruf StatusVerändern (changeStatus)() durchgeführt wird, verändert ovtopmd **116** den Status in der Topologiedatenbank **118** und sendet das entsprechende Ereignis wie bei $NNM_{5.01}$ aus. Außerdem zeichnet es in seinem Prozessspeicher den UUID des Inaktiv-Ereignisses auf.

[0072] Wenn der SekundärAusfall-API-Aufruf SekundärAusfallStatusVerändern (changeSeconaryFailureStatus) () durchgeführt wird, verändert ovovtopmd **116** den Status in der Topologiedatenbank **118** und konstruiert ein Ereignis wie bei $NNM_{5.01}$. Außerdem nimmt es den ovwDbId-Parameter und schlägt den entsprechenden PrimärAusfall-Ereignis-UUID nach und erzeugt einen PrimärAusfallUuid-Varbind, um ihn in das SekundärAusfall-Ereignis aufzunehmen. Dann emittiert es das Ereignis. Der UUID des SekundärAusfall-Ereignisses wird zur möglichen späteren Verwendung in dem Prozessspeicher von ovovtopmd aufgezeichnet.

Langsamer-Ping-Liste

[0073] Die Langsamer-Ping-Liste ermöglicht, dass netmon seine Abfrage von inaktiven Schnittstellen durchführt, ohne in Bezug auf Schnittstellen, die aktiv sind, in Verzug zu geraten. Durch das Trennen von Inaktiv-Schnittstellen (die wahrscheinlich auch das nächste Mal, wenn sie abgefragt werden, inaktiv sind) von Aktiv-Schnittstellen ist netmon in der Lage, den Netzwerkadministrator von Übergängen von Aktiv zu Inaktiv rechtzeitig zu benachrichtigen. Netmon unternimmt dann auch weniger Neuversuche für Schnittstellen auf der Langsamer-Ping-Liste, wodurch es die Zeit und die Netzwerkbandbreite, die beim Durchführen dieser Operationen „verschwendet" wird, beschränkt.

PMD/ECS-Ereignisverteilung

[0074] Fig. 1 ist eine vereinfachte Veranschaulichung der $NNM_{5.01}$-Architektur, die einen Ereignissystembus **114** umfasst. Ein Ereignissystembus **114** existiert eventuell gar nicht. Vielmehr kann eine Anschlussdosenverbindung von jedem kommunizierenden Hilfsmittel **120**, **202**-**206** zu und von dem PMD-Prozess (PMD = Post Master Daemon, Postmaster-Hintergrundroutine) **102** existieren (siehe Fig. 2). Ein Sender **202** sendet ein Ereignis an den PMD-Prozess **102**, und der PMD **102** verteilt das Ereignis an jeden Zuhörer **120**, **204**, **206**. Die Verbindungen können bidirektional sein, so dass jeder Prozess **102**, **120**, **202**-**296** ein Zuhörer und ein Urheber sein kann.

[0075] Bei dem bevorzugten Ausführungsbeispiel des hierin präsentierten Systems wird das Ereignissystem **200** verbessert, indem ein Ereigniskorrelationssystem (ECS – Event Correlation System) **408** in den PMD-Prozess **406** integriert wird (siehe Fig. 4). ECS **408** ist ein weiteres Produkt, das von der Hewlett-Packard Company vertrieben wird. Dieses Produkt wird in dem durch die HP-Teilenummer J1095-90203 identifizierten „ECS 2.0 Designer's Reference Manual" ausführlich beschrieben. Dieses Handbuch ist hiermit durch Bezugnahme mit Bezug auf alles, was es offenbart, in das vorliegende Dokument aufgenommen. Fig. 4 veranschaulicht die PMD/ECS-Architektur **502**. Alle Ereignisse, die in die PMD **406** fließen, fließen in die ECS-Maschine **408**, die die Ereignisse auf folgende Weise manipulieren kann:

- Ereignisse können einfach unverändert durch die ECS-Maschine **408** gelangen.
- Ereignisse können einen gewissen Zeitraum in der ECS-Maschine **408** gespeichert und später freigegeben werden.
- Ereignisse können durch die ECS-Maschine **408** unterdrückt werden. Das heißt, dass sie hereinkommen, aber nicht hinausfließen.
- Unabhängig von einer Unterdrückung können Ereignisse mit anderen Ereignissen korreliert werden. Das heißt, dass ein Attribut an das Ereignis angehängt wird, das ein Mutterereignis spezifiziert. Dies ermöglicht die neue DrillDown-Funktionalität in dem Ereignisse-Browser **120**.
- Zusätzlich zu einem oder statt eines Ereignisses, das in das ECS **408** eintritt, können neue Ereignisse erzeugt werden.
- Ereignisse können über einen längeren Zeitraum im ECS **408** gesichert und als Zustandsinformationen

verwendet werden, um eine Interpretation der Bedeutung nachfolgender Ereignisse zu unterstützen.

• Ereignisse können Anfragen von Daten, die sich außerhalb der PMD **406** und der ECS-Maschine **408** befinden, auslösen, um die Interpretation der Bedeutung des aktuellen Ereignisses zu unterstützen.

[0076] Der Ereignisse-Browser **120**, manche NNM-Prozesse und die meisten Dritte-Hilfsmittel sind mit dem Korreliertes-Ereignis-Bus **402** verbunden. Dieser Bus **402** kann manche Ereignisse unterdrücken oder verzögern lassen, wenn die Logik in den Schaltungen **408** der Maschine dahin gehend ausgelegt ist. Viele der ursprünglichen NNM-Prozesse **204**/**206** sind mit dem Rohes- + Korreliertes-Ereignis-Bus **404** verbunden, so dass sie alle Ereignisse in dem System sehen können.

[0077] Eine detaillierte Beschreibung dessen, wie die Logik in der ECS-Maschine **408** organisiert ist, geht über den Umfang des vorliegenden Dokuments hinaus. Kurz gesagt ist die Logik in zwei Schichten organisiert. Die erste Schicht ist ein Graphikdatenflussentwurf (ECS-Schaltung), der unter Verwendung eines Entwickler-GUI und einer Folge von ECS-Schaltungsknoten verschiedener Typen mit verschiedenen Funktionalitäten erstellt wird. Jeder Knoten der Schaltung kann ein Computerprogramm enthalten, das in der Ereigniskorrelationsbeschreibungssprache (ECDL – Event Correlation Description Language) geschrieben ist. Eines der ECS-Schaltungselemente wird als Kommentarknoten (Annotate Node) **412** bezeichnet und wird dazu verwendet, den entsprechenden Kommentarserver (Annotation Server) **510**, der sich außerhalb des ECS **408** befindet, bezüglich Daten, die nicht in dem ECS **408** enthalten sind, zu befragen.

Benutzerkonfigurationsattribute

[0078] Das oben beschriebene Ereignissystem kann durch den Benutzer dahin gehend konfiguriert werden, eines von vielen möglichen Verhalten zu erhalten, mit verschiedenen Kompromissen in Bezug auf die Leistungsfähigkeit und Nutzbarkeit. Die folgende Liste beschreibt die wichtigsten konfigurierbaren Attribute:

• Critical_Node_Filter_Name <String>

Dieser netmon-Parameter spezifiziert ein Topologiefilter, das es netmon ermöglicht, zwischen einem kritischen Knoten und einem regulären Knoten zu unterscheiden. Ereignisse für kritische Knoten können korreliert sein, werden jedoch niemals durch ECS **408** oder netmon unterdrückt, selbst wenn der Ausfall sekundär ist.

• Critical_Node_Sec_Status <Down|Unknown>

Dieser netmon-Parameter beschreibt den neuen Status, der für das StatusVerändern-Ereignis (changeStatus-Ereignis) für einen kritischen Knoten mit einem SekundärAusfall zu verwenden ist. PrimärAusfälle erhalten immer den Status Inaktiv, ungeachtet dessen, ob der Knoten kritisch oder regulär ist.

• Normal_Node_Sec_Status <Down|Unknown|Ignore>

Dieser netmon-Parameter beschreibt den neuen Status, der für das StatusVerändern-Ereignis für einen Regulärer-Knoten-SekundärAusfall zu verwenden ist. Falls der Wert ignorieren (ignore) ist, dann ist der Status des Knotens nicht verändert. Er wird auf der Abbildung als Aktiv verbleiben, obwohl er unzugänglich ist.

• Sec_Fail_Event_Suppress_Switch <False|True>

Dieser Boolesche netmon-Parameter wird über eine Router-Inaktiv-Kommentarserver-Schnittstelle **410** an ECS **408** kommuniziert und informiert das ECS **408**, ob SekundärAusfälle für normale Knoten (d.h. nicht-kritische Knoten) unterdrückt werden sollen.

[0079] Erörterungen in diesem Dokument, die sich auf diese Benutzerparameter beziehen, beziehen sich in der obigen Reihenfolge auf dieselben. Beispielsweise gibt [<>, Down, Ignore, True] kein Filter an, Critical_Node_Sec_Status=Down, Normal_Node_Sec_Status=Ignore und Sec_FailEvent_Suppress_Switch=True.

Verhalten mit [<>, Down, Unknown, True]-Konfiguration

[0080] Fig. 1 veranschaulicht das Systemverhalten für $NNM_{5,01}$. Fig. 5 veranschaulicht das Systemverhalten für das hierin offenbarte System, wobei die [<>, Down, Unknown, True]-Konfiguration ausgewählt ist. Bei diesem System hat netmon erkannt, dass die Schnittstelle B.1 der PrimärAusfall ist und Schnittstellen B.2, C.1, C.2, X.1, V.1 und Z.1 SekundärAusfälle sind. Da B.1 eine PrimärAusfall-Schnittstelle ist, wird ihr der Status Kritisch verliehen, und sie wird in ovw **104** als rot angezeigt. Ferner wird ein Schnittstelle-B-Inaktiv-Ereignis emittiert.

[0081] Da kein Filter spezifiziert wurde (Critical_Node_Filter_Name=" "), werden alle Knoten als normal erachtet (d.h. keine Knoten werden als Kritisch erachtet), und das Attribut Critical_Node_Sec_Status wird nicht verwendet. Da Normal_Node_Sec_Status=Unknown, werden alle SekundärAusfall-Schnittstellen an allen

Knoten **124**, **128**-**136** als blau angezeigt, um einen Unbekannt-Status darzustellen (d.h. als durchgestrichenen Kreis in der Anzeige **108** der Fig. 5).

[0082] Das Ereignis changeStatus Unknown wird seitens netmon/ovtopmd **110**/**116** für alle SekundärAusfall-Schnittstellen emittiert, wenn netmon sie erst einmal als SekundärAusfall-Schnittstellen erkannt hat. Jedoch unterdrückt das ECS **408** die Ereignisse, denn Sec_Fail_Event_Suppress_Switch=True. Deshalb erscheinen die SekundärAusfall-Ereignisse nicht in dem Ereignisbrowser xnmevents.web **120** auf der Obere-Ebene-Anzeige **522**.

[0083] Eine neue Funktionalität in dem Ereignisbrowser ermöglicht, dass der Benutzer eine Menüoption aufruft, die SekundärAusfälle hervorbringt, die dem ausgewählten Obere-Ebene-Ereignis zugeordnet (mit demselben korreliert) sind. In diesem Fall bringt ein Auswählen von Interface B.1 Down und ein Aufrufen von „Show Correlated Events" („Zeige korrelierte Ereignisse") einen weiteren Dialog zum Vorschein, der die darauf bezogenen SekundärAusfall-Ereignisse zeigt.

[0084] Wenn man Fig. 1 mit Fig. 5 vergleicht, wird man erkennen, dass Probleme, die in Abschnitt „Hintergrund" dieser Offenbarung umrissen werden, durch die Architektur und Konfiguration der Fig. 5 gelöst werden. Die ovw-Anzeige **104** identifiziert die funktionierenden Knoten und Schnittstellen (in grün), die PrimärAusfälle (in rot) und alle SekundärAusfälle (in blau). Die Ereignisbrowser-Anzeige **522** ist nicht mit SekundärAusfällen **524** überhäuft, und sie identifiziert problemlos die eine Instandhaltung benötigende Schnittstelle gegenüber dem NA.

Critical_Node_Filter_Name und der ECS-Kommentarserver

[0085] Das Critical_Node_Filter_Name-Attribut kann durch den Benutzer spezifiziert werden und definiert zwei Klassen (kritisch und regulär) von Netzwerkknoten unter Verwendung der NNM5.01-Netzverbindungsfiltersprache. Diese Sprache ermöglicht es Benutzern, eine Teilmenge von Elementen in der Topologiedatenbank **118** auf der Basis von Attributen in der Datenbank zu beschreiben. Beispielsweise könnte man ohne weiteres eine Gruppe spezifizieren, die aus allen Routern **124**, **128** und Knoten mit der ipAdresse 15.1.2.* besteht.

[0086] Dieser Mechanismus ist vorgesehen, um es dem Benutzer zu ermöglichen, Netzwerkelemente zu identifizieren, deren Zugänglichkeit für die Produktivität der Organisation wesentlich ist. Beispielsweise könnten die Router **124**, **128** und Server kritisch sein, die Arbeitsstationen **132**-**136** und PCs aber eventuell nicht. Schnittstellen, die unzugänglich sind und PrimärAusfälle sind, wird immer ein Inaktiv-Status verliehen, ungeachtet dessen, zu welcher Klasse sie gehören. Wenn jedoch eine Schnittstelle unzugänglich ist und ein SekundärAusfall ist, so kann ein Kritischer-Knoten-Filter dazu verwendet werden, das Verhalten des Systems zu beeinflussen.

[0087] Wenn eine Schnittstelle unzugänglich ist und sich an einem kritischen Knoten (gemäß der Definition durch das Filter und der Auswertung durch netmon) befindet, so definiert der Critical_Node_Sec_Status-Attributwert den Status, der der Schnittstelle tatsächlich verliehen wird. Die möglichen Werte sind Inaktiv und Unbekannt. Dieses Attribut wird durch netmon ausgewertet, was die Entität ist, die ovtopmd 116 anweist, den Status der Schnittstelle zu verändern.

[0088] Wenn eine Schnittstelle unzugänglich ist sich an einem regulären Knoten (gemäß der Definition durch das Filter und der Auswertung durch netmon) befindet, so definiert der Normal_Node_Sec_Status-Attributwert den Status, der der Schnittstelle tatsächlich verliehen wird. Die möglichen Werte sind Inaktiv, Unbekannt und Ignorieren. Wiederum wird dieses Attribut durch netmon ausgewertet, was die Entität ist, die ovtopmd **116** anweist, den Status der Schnittstelle zu verändern.

[0089] Ein Wert von Inaktiv oder Unbekannt führt zu einem Verhalten für reguläre Knoten, das zu dem Verhalten von kritischen Knoten analog ist. Jedoch weist ein Wert von ignorieren netmon an, diese unzugängliche Schnittstelle an einem regulären Knoten zu ignorieren. Das heißt, verändere nicht den Status der Schnittstelle und sende keinerlei Ereignisse bezüglich dieser Schnittstelle aus. Sie bleibt auf der Abbildung als Aktiv, obwohl sie unzugänglich ist.

[0090] Diese Konfiguration ist nützlich, wenn es wünschenswert ist, Netzwerkverkehr und NNM-Leistungsfähigkeit bezüglich Knoten, die für die Produktivität der Organisation nicht entscheidend sind, zu minimieren. In dieser Situation setzt netmon die Schnittstelle trotzdem auf die Langsamer-Ping-Liste, so dass Netzwerkverkehr weiter minimiert wird und eine netmon-Statusabfrage weiterhin nach Plan verläuft.

[0091] Dieses Filter wird durch netmon verwendet (gemäß der obigen Beschreibung), wenn es entdeckt, dass eine SekundärAusfall-Schnittstelle Inaktiv ist. Ferner wird es durch die ECS-Maschine **408** benötigt, um zu bestimmen, ob das entsprechende Ereignis unterdrückt werden sollte. Da netmon bereits dahin gehend eingerichtet ist, zwischen kritischen und regulären Knoten zu unterscheiden, ist es sinnvoll, netmon diese Unterscheidung an die Router-Inaktiv-Schaltung in der ECS-Maschine **408** kommunizieren zu lassen.

[0092] Es tut dies über den durch das ECS **408** bereitgestellten Kommentarserver-Mechanismus **412**. Immer dann, wenn eine Schaltung im ECS **408** wissen muss, ob ein Ereignis, das sie empfangen hat, einem kritischen Knoten oder einem regulären Knoten entspricht, fließt das Ereignis in einen entsprechenden Kommentarschaltungsknoten **412**, der die Anfrage unter Verwendung von UNIX®-Domain-Anschlussdosen an den Kommentarserverprozess **510** sendet. Bei Windows® NT wird ein anderer Mechanismus als eine UNIX®-Domgin-Anschlussdose verwendet.

[0093] Die Anfrage kommt an dem Router-Inaktiv-Kommentarserver **510** an, der das ovwDbld-Argument durch seine Filterauswertungseinrichtung laufen lässt und das Boolesche Ergebnis an den Kommentarschaltungsknoten **412** in der ECS-Schaltung **408** zurücksendet. Dieser jeweilige Kommentarserver ist in netmon eingebaut (siehe Fig. 4).

[0094] Man beachte, dass sich in einem verteilten System, bei dem mehrere netmons laufen, das Critical_Node_Filter_Name-Attribut an der Verwaltungsstation **510** von dem Wert an einer Sammelstation unterscheiden kann.

ECS-Router-Inaktiv-Schaltung

[0095] Obwohl die ECS-Maschine **408** eine beträchtliche Leistung aufweist, wird sehr wenig dieses Potentials durch die Router-Inaktiv-Schaltung verwendet, da ein Großteil der Logik aus Gründen der Leistungsfähigkeit in den netmon-Prozess platziert wurde. Fig. 6 veranschaulicht die Schaltungslogik **600**.

[0096] Falls das Ereignis kein StatusVerändern-Ereignis (KnotenInaktiv oder Schnittstellelnaktiv) **602**/**604** ist oder falls das Ereignis ein PrimärAusfall (da keine zusätzlichen Varbinds vorliegen) **606** ist, so wird das Ereignis unmittelbar an andere ECS-Schaltungselemente **608**/**626** weitergeleitet. Andernfalls fließt das Ereignis in den SekundärAusfall-Pfad **610**, wo es mit dem PrimärAusfall-Ereignis **612** korreliert wird. Diese Korrelation ist nichts weiter als ein Protokollieren eines Attributs in einer Protokolldatei, die das Mutterereignis des aktuellen Ereignisses identifiziert. Dies ist möglich, da der Ereignis-UUID des Mutterereignisses (des PrimärAusfall-Ereignisses) in dem zusätzlichen Varbind vorliegt. Diese Korrelation ermöglicht ein Drill-Down mit dem Ereignisse-Browser.

[0097] An diesem Punkt muss die Schaltung entscheiden, ob das Ereignis unterdrückt werden sollte. Wenn das Ereignis einem kritischen Knoten **620** entspricht, wird das Ereignis nicht unterdrückt, da es wichtig ist, dass der NA weiß, dass dieser wichtige Server oder Router unmittelbar repariert wird **618**. Die Schaltung ermittelt, ob der Knoten kritisch oder regulär ist, indem sie den in netmon **616** eingebetteten Router-Inaktiv-Kommentarserver befragt.

[0098] Wenn das Ereignis einem regulären Knoten **622** entspricht, dann untersucht die Schaltung den Wert des Sec_Fail_Event_Suppress_Switch-Attributs und verhält sich entsprechend **624**/**626**/**628**. All diese Attribute sind in netmons Konfiguration konfiguriert. Dieses Attribut wird nicht wirklich durch netmon verwendet. Es wird lediglich durch die ECS-Schaltung **600** verwendet. Deshalb wird der Wert dieses Attributs über die Kommentarserver-Schnittstelle auch an ECS kommuniziert.

Verhalten bei [<serverFilter>, Down, Unknown, True]-Konfiguration

[0099] Fig. 7 veranschaulicht ein Systemverhalten bei der [<serverFilter>, Down, Unknown, True]-Konfiguration. Diese Konfiguration unterscheidet sich von der Konfiguration der Fig. 5, da der Benutzer unter Verwendung des Critical_Node-Filter_Name-Attributs ein Filter spezifiziert hat. Das Filter in Fig. 7 wurde dahin gehend entworfen, Knoten Z als kritischen Knoten zu identifizieren. Beispielsweise kann die Produktivität der Organisation von der Verfügbarkeit eines an dem Knoten Z laufenden Anwendungsservers abhängen.

[0100] Bei diesem Szenario hat netmon erkannt, dass die Schnittstelle B.1 der PrimärAusfall ist, und die Schnittstellen B.2, C.1, C.2, X.1, Y.1 und Z.1 sekundär sind. Da B.1 eine PrimärAusfall-Schnittstelle ist, wird ihr der Status Kritisch verliehen, der in ovw als rot angezeigt ist, und ein Schnittstelle-B-Inaktiv-Ereignis wird emit-

tiert.

[0101] Da die Schnittstelle Z.1 eine an einem KritischerKnoten angeordnete SekundärAusfall-Schnittstelle ist, wird ihr ein Status verliehen, der durch das Critical_Node_Sec_Status-Attribut spezifiziert wird, das dahin gehend konfiguriert wurde, einen Wert von Inaktiv zu haben. Der Knoten Z und die Schnittstelle Z.1 sind in ovw als rot angezeigt, und ein Schnittstelle-Z.1-Inaktiv-Ereignis wird emittiert.

[0102] Allen verbleibenden SekundärAusfall-Schnittstellen wird der Status verliehen, der durch das Normal_Node_Sec_Status-Attribut spezifiziert ist, das einen Wert von Unbekannt aufweist. Diese Schnittstellen sind in blau angezeigt, um einen Unbekannt-Status darzustellen.

[0103] Das Ereignis changeStatus Unbekannt wird von netmon/ovtopmd für alle nicht-kritischen/SekundärAusfall-Schnittstellen emittiert, nachdem netmon sie als SekundärAusfall-Schnittstellen erkannt hat. Jedoch unterdrückt das ECS **408** die Ereignisse, da Sec_Fail_Event_Suppress_Switch=True. Deshalb erscheinen die SekundärAusfall-Ereignisse, die keinen kritischen Knoten entsprechen, nicht auf der Obere-Ebene-Anzeige **722** in dem xnmevents.web-Ereignisbrowser **120**.

[0104] Eine neue Funktionalität bei dem Ereignisbrowser **120** ermöglicht, dass der Benutzer eine Menüoption aufruft, die SekundärAusfälle **724** zum Vorschein bringt, die dem ausgewählten Obere-Ebene-Ereignis zugeordnet (mit demselben korreliert) sind. In diesem Fall zeigt ein Auswählen von „Interface B.1 Down" und ein Aufrufen von „Show Correlated Events" einen weiteren Dialog **724** zum Vorschein, der die verwandten SekundärAusfall-Ereignisse zeigt. Man beachte, dass die Schnittstelle Z.1 in der Obere-Ebene-Anzeige **722** und in der Drill-Down-Anzeige **724** erscheint. Sie erscheint in der Obere-Ebene-Anzeige **722**, da sie sich an einem Knoten befindet, der kritisch ist. Sie erscheint in der Drill-Down-Anzeige **724**, da sie auf Grund des Ausfalls der Schnittstelle B.1 unzugänglich ist.

[0105] Beim Vergleichen von Fig. 1 mit Fig. 7 kann man erkennen, dass alle drei Anforderungen, die in dem Abschnitt „Hintergrund" dieser Offenbarung identifiziert wurden, nun durch die Architektur und Konfiguration der vorliegenden Erfindung erfüllt sind. Die ovw-Anzeige **104**/**108** identifiziert die funktionierenden Knoten und Schnittstellen (in grün), die PrimärAusfall-Schnittstellen (in rot), die kritischen SekundärAusfall-Schnittstellen (in rot) und alle regulären SekundärAusfälle (in blau). Die Ereignisbrowseranzeige **722** ist nicht mit nicht-kritischen SekundärAusfällen überhäuft, und sie identifiziert ohne weiteres eine Schnittstelle, die eine Wartung benötigt, gegenüber dem NA.

Verhalten bei [<serverFilter>, Unknown, Ignore, True]-Konfiguration

[0106] Fig. 8 veranschaulicht ein Systemverhalten bei einer [<serverFilter>, Unknown, Ignore, True]-Konfiguration. Diese Konfiguration unterscheidet sich von der in Fig. 7 gezeigten Konfiguration darin, dass ein Benutzer spezifiziert hat, dass SekundärAusfälle von kritischen Knoten ein Unbekannt-Status verliehen werden sollte, und dass SekundärAusfälle an regulären Knoten ignoriert werden sollten.

[0107] Diese Konfiguration weist Vorteile und Nachteile auf. Der Hauptvorteil besteht darin, dass die System- und Netzwerk-Leistungsfähigkeit sehr gut ist, da an Sammel- und/oder Verwaltungsstationen weniger Statusänderungen und Ereignisse erzeugt werden. Die ovw-Anzeige **104**/**108** und die Ereignisbrowseranzeige **822** kommunizieren mehr von der Auswirkung des Ausfalls, da PrimärAusfälle und SekundärAusfälle von Schnittstellen von kritischen Knoten in unterschiedlichen Farben vorliegen und die Anhäufung von unwichtigen SekundärAusfällen nicht gezeigt ist.

[0108] Der Nachteil besteht darin, dass SekundärAusfälle von unwichtigen Knoten als zugänglich angezeigt werden, wenn sie nicht zugänglich sind. Dies ist ein Kompromiss, den der Netzwerkadministrator abwägen muss, wenn er diese Konfiguration wählt.

[0109] Bei diesem Szenario erkennt netmon, dass die Schnittstelle B.1 ein PrimärAusfall ist und Schnittstellen B.2, C.1, C.2, X.1, Y.1 und Z.1 sekundär sind. Da B.1 eine PrimärAusfall-Schnittstelle ist, wird ihr der Status Kritisch verliehen, und sie wird in ovw **104**/**108** als rot angezeigt. Ferner wird ein Schnittstelle-B-Inaktiv-Ereignis emittiert.

[0110] Da die Schnittstelle Z.1 ein SekundärAusfall ist, der sich an einem kritischen Knoten befindet, wird ihr ein Status verliehen, der durch das Critical_Node_Sec_Status-Attribut spezifiziert wird, das dahin gehend konfiguriert wurde, einen Unbekannt-Wert aufzuweisen. Die Schnittstelle Z.1 wird in ovw als blau angezeigt, und

ein Schnittstelle Z|Unbekannt-Ereignis wird emittiert.

[0111] Alle übrigen SekundärAusfall-Schnittstellen werden ignoriert. Es erfolgen keine Statusänderungen, und es werden keine Ereignisse emittiert. Sie werden weiterhin als grün angezeigt, was einen Statuswert von Aktiv darstellt. Jedoch geht netmon trotzdem noch in seinen Sicherungsabfragemodus.

[0112] Man beachte, dass Fig. 8 mehrere Schnittstelle-Aktiv-Ereignisse zeigt, die kollektiv zu einer Überhäufung der Ereignisanzeige führen. Diese Veranschaulichung ist etwas irreführend. Diese Schnittstelle-Aktiv-Ereignisse sind gezeigt, um zu veranschaulichen, dass anfänglich, wenn netmon die Knoten und Schnittstellen entdeckt, die Schnittstelle-Aktiv-Ereignisse übertragen werden. Jedoch sollte das nur einmal geschehen. Es geschieht nie mehr wieder, es sei denn, der Knoten wird in dem Netzwerk physisch verschoben.

[0113] Während eines typischen Betriebs sieht der NA lediglich die zwei Ereignisse, Schnittstelle-B.1-Inaktiv und Schnittstelle-Z.1-Unbekannt. Desgleichen geschehen die Schnittstelle-Aktiv-Ereignisse in den Fig. 1, Fig. 5 und Fig. 7 nur, wenn die Knoten **124**, **128**-**136** entdeckt werden. Die Anzeigen in jedem dieser vier Szenarios werden von einer Überhäufung gereinigt und sind für den NA nützlich, da sie ein fehlerhaftes Netzwerkelement genau zu lokalisieren versuchen.

[0114] Eine neue Funktionalität bei dem Ereignisbrowser **120** ermöglicht, dass der Benutzer eine Menüoption aufruft, die SekundärAusfälle zum Vorschein bringt, die dem ausgewählten Obere-Ebene-Ereignis zugeordnet (mit demselben korreliert) sind. In diesem Fall zeigt ein Auswählen von „Interface B.1 Down" und ein Aufrufen von „Show Correlated Events" einen weiteren Dialog zum Vorschein, der die verwandten SekundärAusfall-Ereignisse zeigt. Man beachte, dass die Schnittstelle Z.1 wiederum in der Obere-Ebene-Anzeige **822** und in der Drill-Down-Anzeige **824** erscheint. Sie erscheint in der Obere-Ebene-Anzeige **822**, da sie sich an einem Knoten befindet, der als kritisch erachtet wird. Sie erscheint in der Drill-Down-Anzeige **824**, da sie auf Grund des Ausfalls der Schnittstelle B.1 unzugänglich ist.

Verhalten bei [<>, Down, Down, False]-Konfiguration

[0115] Diese Konfiguration zwingt ein System, sich sehr ähnlich wie $NNM_{5.01}$ zu verhalten, da allen unzugänglichen Schnittstellen ein Inaktiv-Status verliehen wird, in ovw als rot angezeigt, und keine Ereignisse unterdrückt werden (siehe Fig. 1). Das Systemverhalten unterscheidet sich auf folgende Weise:

- SekundärAusfall-Schnittstellen werden immer noch durch netmon erkannt, und ihre Knoten-Inaktiv- und Schnittstelle-Inaktiv-Ereignisse enthalten das zusätzliche Varbind.
- Obwohl die SekundärAusfälle nicht unterdrückt werden, werden sie trotzdem mit dem PrimärAusfall korreliert und sind ferner mittels Drill-Down sichtbar.
- Der Rückkopplungs-Abfragealgorithmus liefert immer noch Verbesserungen der Leistungsfähigkeit (d.h. die Langsamer-Ping-Liste wird verwendet).

Verteilte Architektur

[0116] Das hierin beschriebene System und Verfahren sind ohne weiteres an eine verteilte Umgebung, die sowohl Sammel- als auch Verwaltungsstationen umfasst, anpassbar. Eine exemplarische verteilte Umgebung, an die das offenbarte System und Verfahren ohne weiteres anpassbar sind, ist in der US-Patentanmeldung von Eric Pulsipher et al., die am 29. August 1996 eingereicht wurde, die Seriennummer 08/705,358 und den Titel „Distributed Internet Monitoring System and Method" trägt, beschrieben.

Patentansprüche

1. Eine Netzwerküberwachungseinrichtung (**510**) zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen, die folgende Merkmale aufweist:
a) ein oder mehrere computerlesbare Speicherungsmedien; und
b) einen computerlesbaren Programmcode, der in dem einen oder den mehreren computerlesbaren Speicherungsmedien gespeichert ist, wobei der computerlesbare Programmcode folgende Merkmale umfasst:
i) einen Code zum Entdecken der Topologie einer Mehrzahl von Netzwerkelementen (**124**, **128**-**136**);
ii) einen Code zum periodischen Abfragen einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen zugeordnet sind;
iii) einen Code zum Berechnen oder Validieren, für jede der Mehrzahl von Netzwerkschnittstellen, einer Liste der Netzwerkschnittstellen, die dem Pfad entsprechen, den ein Netzwerkpaket nehmen würde, wenn es von der Netzwerküberwachungseinrichtung an die betreffende Netzwerkschnittstelle gesendet würde; und

iv) einen Code zum Analysieren eines Status von Netzwerkschnittstellen, die durch die Liste von Netzwerkschnittstellen für die betreffende Netzwerkschnittstelle, die als IIQ (interface in question, betreffende Schnittstelle) bezeichnet wird, die nicht auf eine Abfrage reagiert, identifiziert werden.

2. Eine Netzwerküberwachungseinrichtung (**510**) gemäß Anspruch 1, die ferner einen Code zum Erstellen einer Liste von defekten oder ausgefallenen Netzwerkschnittstellen, die mit einer langsameren Rate als andere Schnittstellen abzufragen sind, umfasst.

3. Eine Netzwerkverwaltungsvorrichtung zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen und zum Präsentieren dieser Informationen gegenüber einem Netzwerkadministrator, wobei die Netzwerkverwaltungsvorrichtung folgende Merkmale aufweist:
a) einen Anzeigeprozess zum Präsentieren der Informationen (**104** oder **120**); und
b) eine Netzwerküberwachungseinrichtung (**510**), die folgende Merkmale aufweist:
i) eine Einrichtung zum Entdecken der Topologie einer Mehrzahl von mit derselben verbundenen Netzwerkelementen (**124**, **128**-**136**);
ii) eine Einrichtung zum periodischen Abfragen einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen zugeordnet sind;
iii) eine Einrichtung zum Berechnen oder Validieren, für jede der Mehrzahl von Netzwerkschnittstellen, einer Liste der Netzwerkschnittstellen, die dem Pfad entsprechen, den ein Netzwerkpaket nehmen würde, wenn es von der Netzwerküberwachungseinrichtung an die betreffende Netzwerkschnittstelle gesendet würde; und
iv) eine Einrichtung zum Analysieren eines Status von Netzwerkschnittstellen, die durch die Liste von Netzwerkschnittstellen identifiziert werden, für die betreffende Netzwerkschnittstelle, die als IIQ bezeichnet wird, die nicht auf eine Abfrage reagiert.
wobei der Anzeigeprozess und die Netzwerküberwachungseinrichtung mittels eines oder mehrerer Ereignisbusse (**114**) kommunizieren.

4. Netzwerkverwaltungsvorrichtung gemäß Anspruch 3, die ferner folgende Merkmale aufweist:
a) ein Ereigniskorrelationssystem (**408**), das einen Kommentarknoten (**408**) umfasst; und
b) einen Kommentarserver (**510**);
wobei der eine oder die mehreren Ereignisbusse (**114**) einen Korrelierte-Ereignisse-Bus (**402**) umfassen, der Anzeigeprozess (**120**) Ereignisdaten über den Korrelierte-Ereignisse-Bus empfängt und der Kommentarserver und der Kommentarknoten mittels eines Kommunikationskanals (**410**) kommunizieren;
wobei der Kommentarserver dahin gehend konfiguriert ist, die folgenden Kommentarinformationen zu verarbeiten:
i) einen Kommentar, der eine Kritischer-Knoten-Erkennung vorsieht;
ii) einen Kommentar, der angibt, wie unzugängliche Knoten in den Listen von Netzwerkschnittstellen verarbeitet und angezeigt werden sollten;
iii) einen Kommentar, der angibt, wie unzugängliche Knoten, die sich nicht in den Listen von Netzwerkschnittstellen befinden, verarbeitet und angezeigt werden sollten; und
iv) einen Kommentar, der eine Unterdrückung von unzugänglichen Knoten, die sich nicht in den Listen von Netzwerkschnittstellen befinden, vorsieht.

5. Netzwerkverwaltungsvorrichtung gemäß Anspruch 4, bei der:
a) das Ereigniskorrelationssystem eine Einrichtung zum Unterdrücken von Ereignissen umfasst, so dass sie nicht auf dem Korrelierte-Ereignisse-Bus (**402**) erscheinen; und
b) der Anzeigeprozess (**120**) eine Drill-Down-Schnittstelle (**522**, **722**, **822**) umfasst, durch die verschiedene unterdrückte Ereignisse zum Zweck einer Betrachtung aufgerufen werden können.

6. Ein computerimplementiertes Verfahren zum Unterscheiden zwischen defekten und unzugänglichen Netzwerkelementen und zum Präsentieren dieser Informationen gegenüber einem Netzwerkadministrator, wobei das Verfahren folgende Schritte umfasst:
a) Entdecken der Topologie einer Mehrzahl von Netzwerkelementen (**124**, **128**-**136**);
b) periodisches Abfragen einer Mehrzahl von Netzwerkschnittstellen, die der Mehrzahl von Netzwerkelementen zugeordnet sind;
c) Berechnen oder Validieren, für jede der Mehrzahl von Netzwerkschnittstellen, einer Liste der Netzwerkschnittstellen, die dem Pfad entsprechen, den ein Netzwerkpaket nehmen würde, wenn es von der Netzwerküberwachungseinrichtung an die betreffende Netzwerkschnittstelle gesendet würde; und
d) Analysieren eines Status von Netzwerkschnittstellen, die durch eine Liste von Netzwerkschnittstellen identifiziert werden, für die betreffende Netzwerkschnittstelle, die als IIQ bezeichnet wird, die nicht auf eine Abfrage reagiert.

7. Ein Verfahren gemäß Anspruch 6, bei dem der Schritt des Analysierens des Status von Netzwerkschnittstellen, die durch die Liste von Netzwerkschnittstellen für eine IIQ, die nicht auf eine Abfrage antwortet, identifiziert werden, folgende Schritte umfasst:
a) erstens, Untersuchen des In-Speicher-Status einer oder mehrerer Netzwerkschnittstellen, um zu bestimmen, ob eine durch die Liste von Netzwerkschnittstellen für die IIQ identifizierte Netzwerkschnittstelle nicht verfügbar ist, und falls dies der Fall ist, Identifizieren dieser Netzwerkschnittstelle als defekte Schnittstelle;
b) zweitens, falls eine durch die Liste von Netzwerkschnittstellen für die IIQ identifizierte Netzwerkschnittstelle nicht als defekte Schnittstelle identifiziert wurde, Verifizieren des Status einer oder mehrerer Netzwerkschnittstellen, die durch die Liste von Netzwerkschnittstellen für die IIQ identifiziert werden, um zu bestimmen, ob eine derselben eine defekte Schnittstelle ist; und
c) drittens, falls eine durch die Liste von Netzwerkschnittstellen für die IIQ identifizierte Netzwerkschnittstelle nicht als defekte Schnittstelle identifiziert wurde, Identifizieren der IIQ als defekte Schnittstelle.

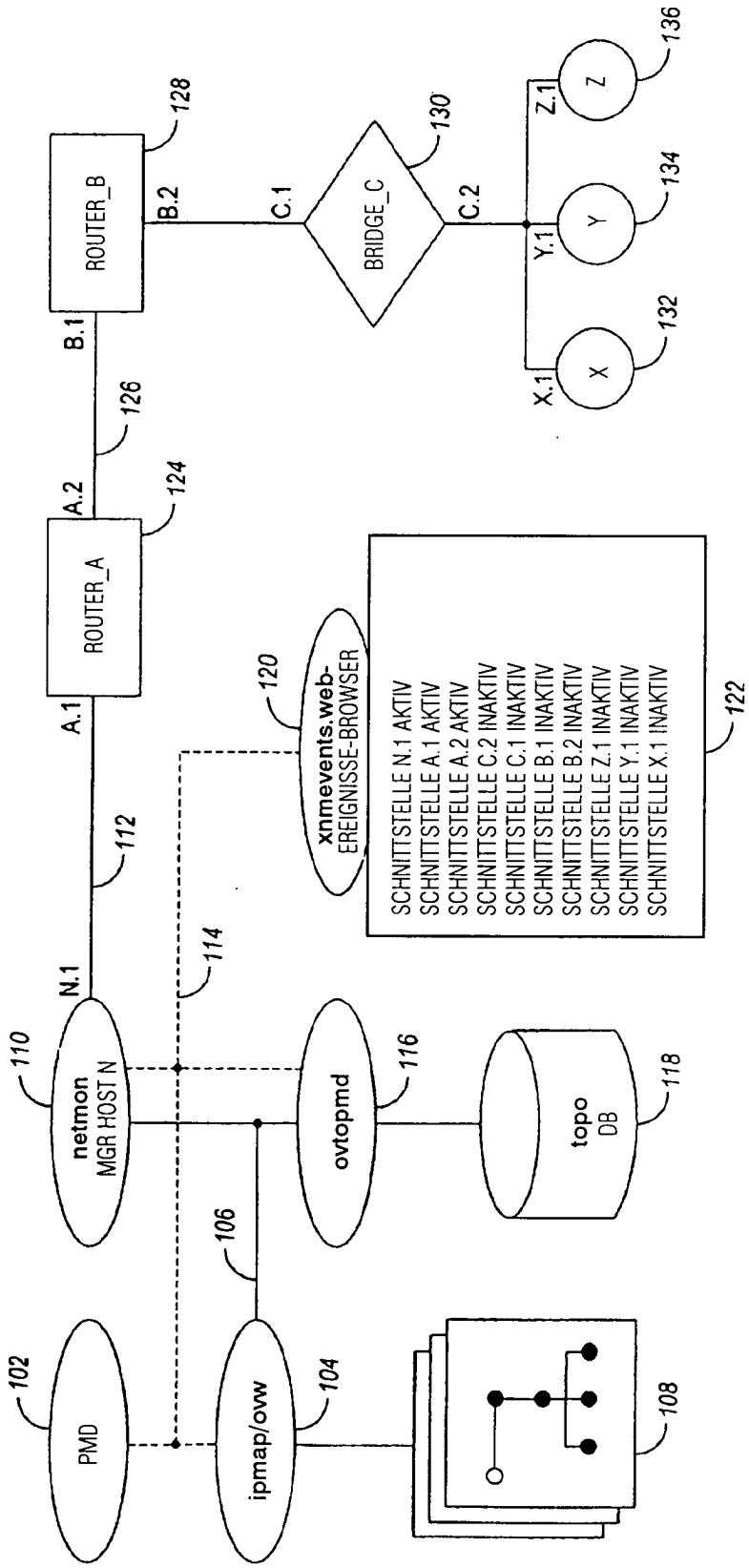
8. Ein Verfahren gemäß Anspruch 6, das ferner den Schritt des Pflegens einer Liste von Netzwerkschnittstellen, die als nicht verfügbar identifiziert wurden, die jedoch nicht als unzugänglich oder defekt eingestuft wurden, umfasst, wobei ein Algorithmus zum Identifizieren unzugänglicher Netzwerkschnittstellen folgende Schritte umfasst:
a) Untersuchen eines In-Speicher-Status einer oder mehrerer Netzwerkschnittstellen, um zu bestimmen, ob eine Netzwerkschnittstelle, die dahin gehend identifiziert ist, dass sie sich auf der Liste von Netzwerkschnittstellen befindet, die einem Pfad entsprechen, den ein Netzwerkpaket nehmen würde, wenn es von der Netzwerküberwachungseinrichtung an die Netzwerkschnittstelle für die IIQ gesendet würde, nicht verfügbar ist, und falls dies der Fall ist, Identifizieren dieser Netzwerkschnittstelle als defekte Schnittstelle; und
b) falls der obige Schritt eine Netzwerkschnittstelle erfolgreich als defekte Schnittstelle identifiziert,
i) Identifizieren der IIQ als unzugängliche Schnittstelle;
ii) Emittieren eines Ereignisses, das angibt, dass die Netzwerkschnittstelle nicht verfügbar ist; und
iii) Platzieren einer In-Speicher-Darstellung der IIQ auf einer Liste von Netzwerkschnittstellen, die mit einer langsameren Rate abzufragen sind als andere Schnittstellen.

9. Ein Verfahren gemäß Anspruch 8, bei dem der Algorithmus zum Identifizieren von unzugänglichen Netzwerkschnittstellen folgende Schritte umfasst:
a) falls keine der durch die Liste von Netzwerkschnittstellen für die IIQ identifizierten Netzwerkschnittstellen als defekte Schnittstelle identifiziert wurde, Überprüfen des Status einer oder mehrerer Netzwerkschnittstellen, die durch die Liste von Netzwerkschnittstellen für die IIQ identifiziert wurden, durch
i) Bewegen einer In-Speicher-Darstellung der IIQ und aller durch die Liste von Netzwerkschnittstellen für die IIQ identifizierten Netzwerkschnittstellen zu der Liste von Netzwerkschnittstellen, die als nicht verfügbar identifiziert wurden, die jedoch nicht als unzugänglich oder defekt eingestuft wurden, und Entfernen dieser Netzwerkschnittstellen aus allen anderen Listen, die für eine Netzwerküberwachungseinrichtung, die dieses Verfahren ausführt, zugänglich sind;
ii) sequentielles Durchgehen der Liste von Netzwerkschnittstellen, die als nicht verfügbar identifiziert wurden, die jedoch nicht als unzugänglich oder defekt eingestuft wurden, Abfragen jeder Netzwerkschnittstelle, um zu bestimmen, ob es sich dabei um eine defekte Schnittstelle handelt; und
b) falls eine durch die Liste von Netzwerkschnittstellen für die IIQ identifizierte Netzwerkschnittstelle nicht als defekte Schnittstelle identifiziert wurde, Identifizieren der IIQ als defekte Schnittstelle, andernfalls Identifizieren der IIQ als unzugängliche Schnittstelle.

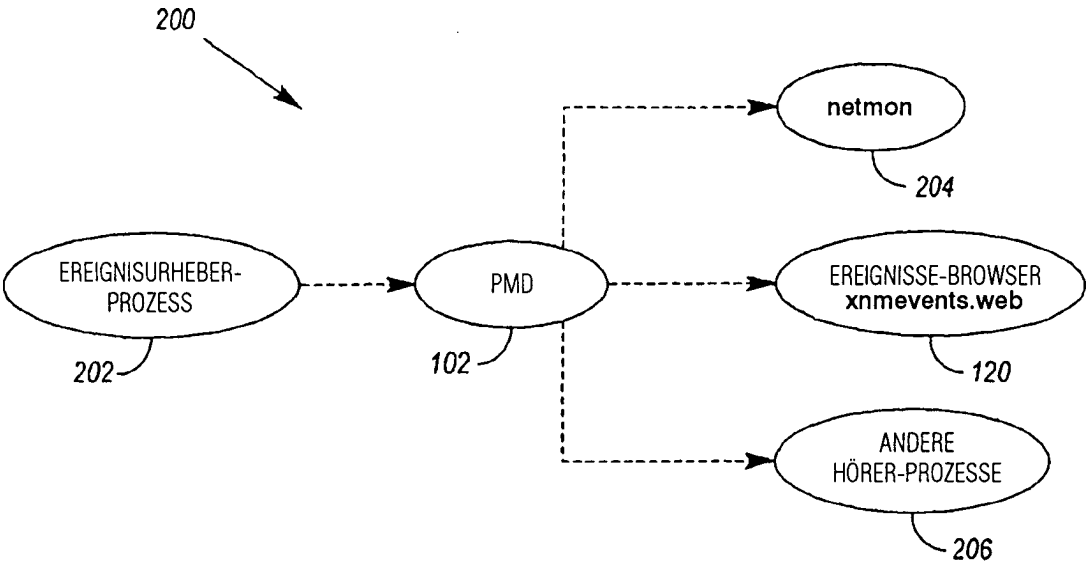
10. Ein Verfahren gemäß Anspruch 6, das ferner den Schritt des Verwendens einer Netzwerkschnittstellen-Liste einer IIQ zum Identifizieren einer IIQ als defekte Schnittstelle oder als unzugängliche Schnittstelle umfasst.

Es folgen 7 Blatt Zeichnungen

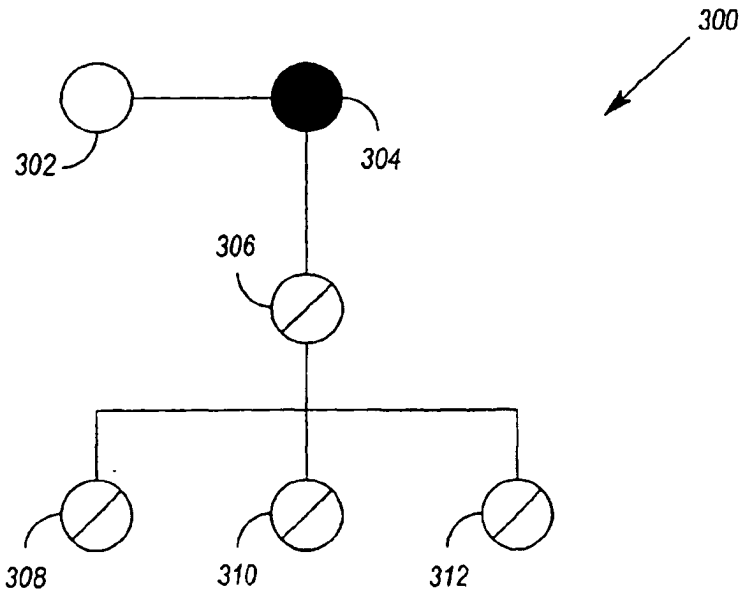
Anhängende Zeichnungen



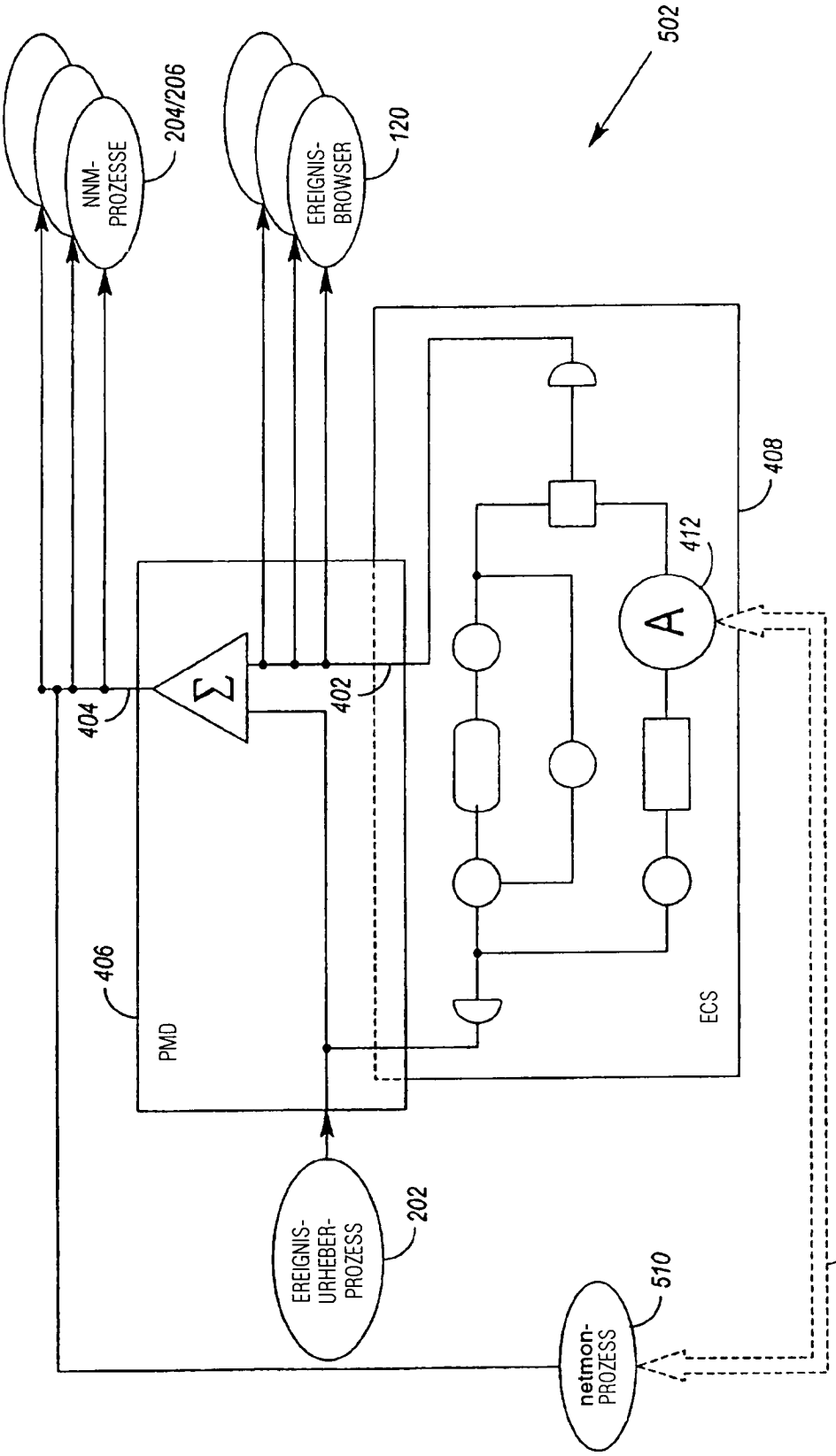
FIGUR 1



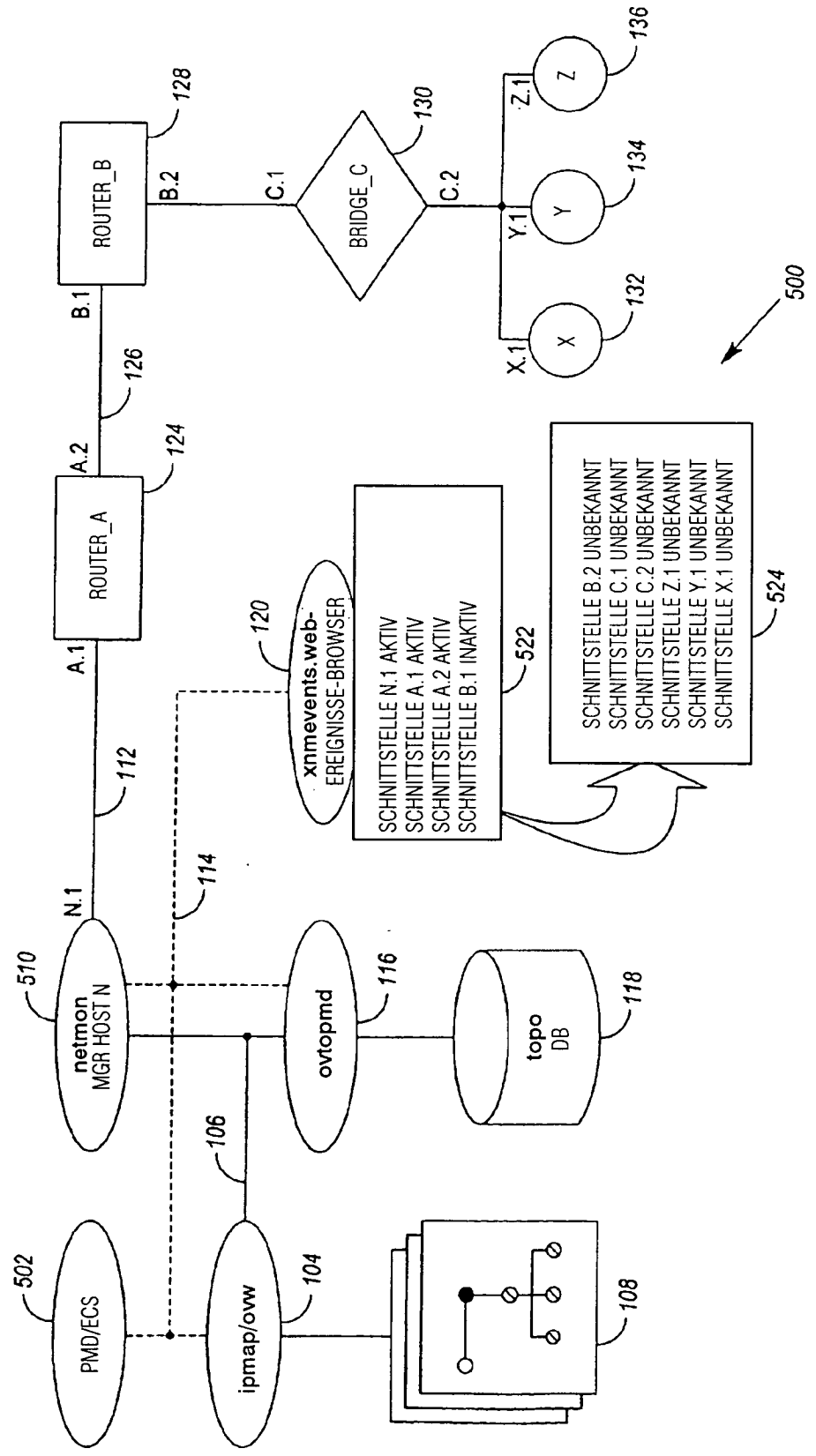
FIGUR 2



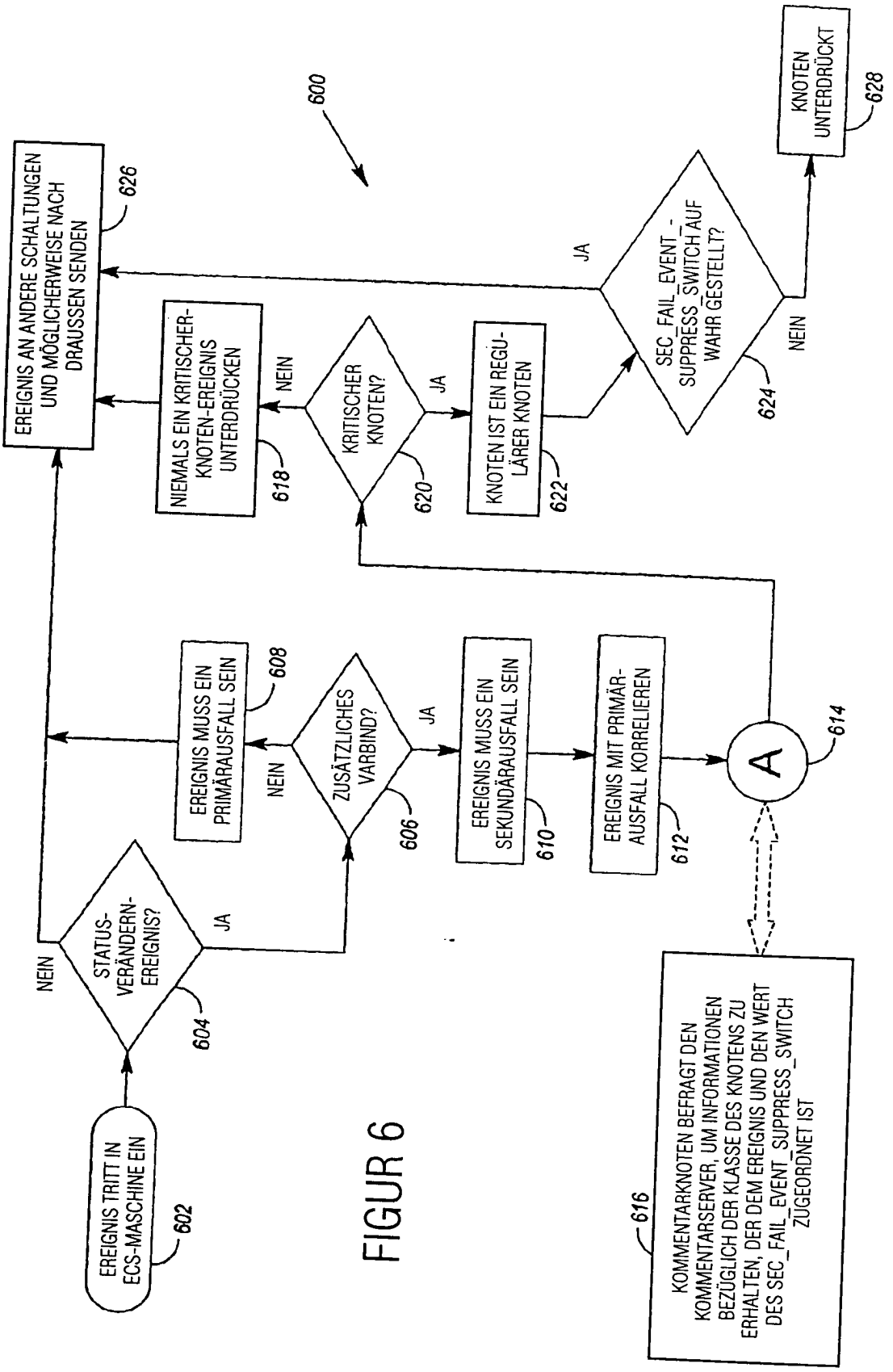
FIGUR 3


502
NNM-PROZESSE
204/206
EREIGNIS-BROWSER
120
408
412
A
Σ
402
404
406
PMD
ECS
EREIGNIS-URHEBER-PROZESS
202
netmon-PROZESS
510
410

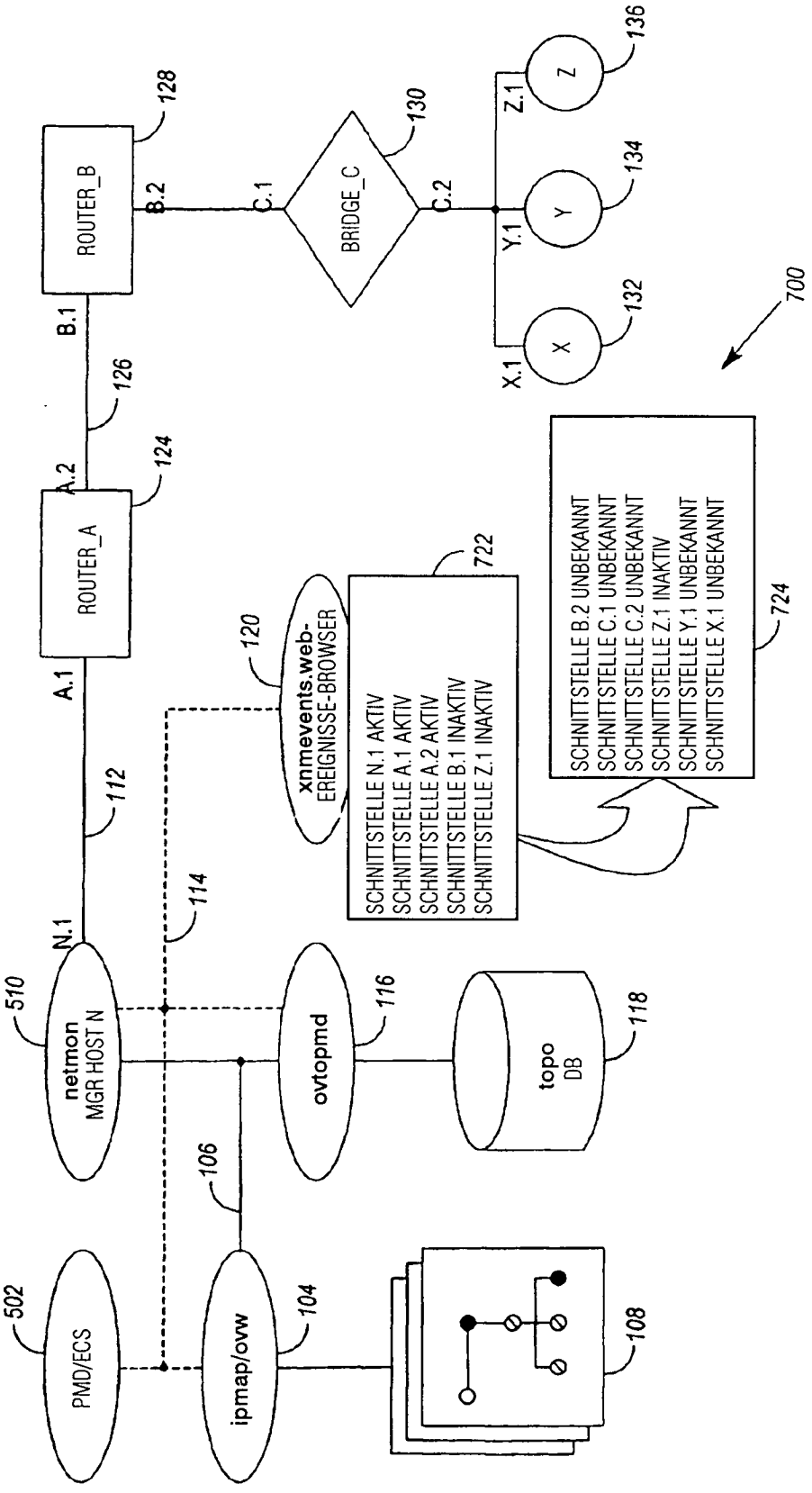
FIGUR 4


ROUTER_B
128
B.2
C.1
BRIDGE_C
130
C.2
Z.1
Z
136
Y.1
Y
134
X.1
X
132
500
B.1
126
A.2
ROUTER_A
124
A.1
120
xnmevents.web-
EREIGNISSE-BROWSER
SCHNITTSTELLE N.1 AKTIV
SCHNITTSTELLE A.1 AKTIV
SCHNITTSTELLE A.2 AKTIV
SCHNITTSTELLE B.1 INAKTIV
522
SCHNITTSTELLE B.2 UNBEKANNT
SCHNITTSTELLE C.1 UNBEKANNT
SCHNITTSTELLE C.2 UNBEKANNT
SCHNITTSTELLE Z.1 UNBEKANNT
SCHNITTSTELLE Y.1 UNBEKANNT
SCHNITTSTELLE X.1 UNBEKANNT
524
112
N.1
114
510
netmon
MGR HOST N
ovtopmd
116
topo
DB
118
106
502
PMD/ECS
ipmap/ovw
104
108

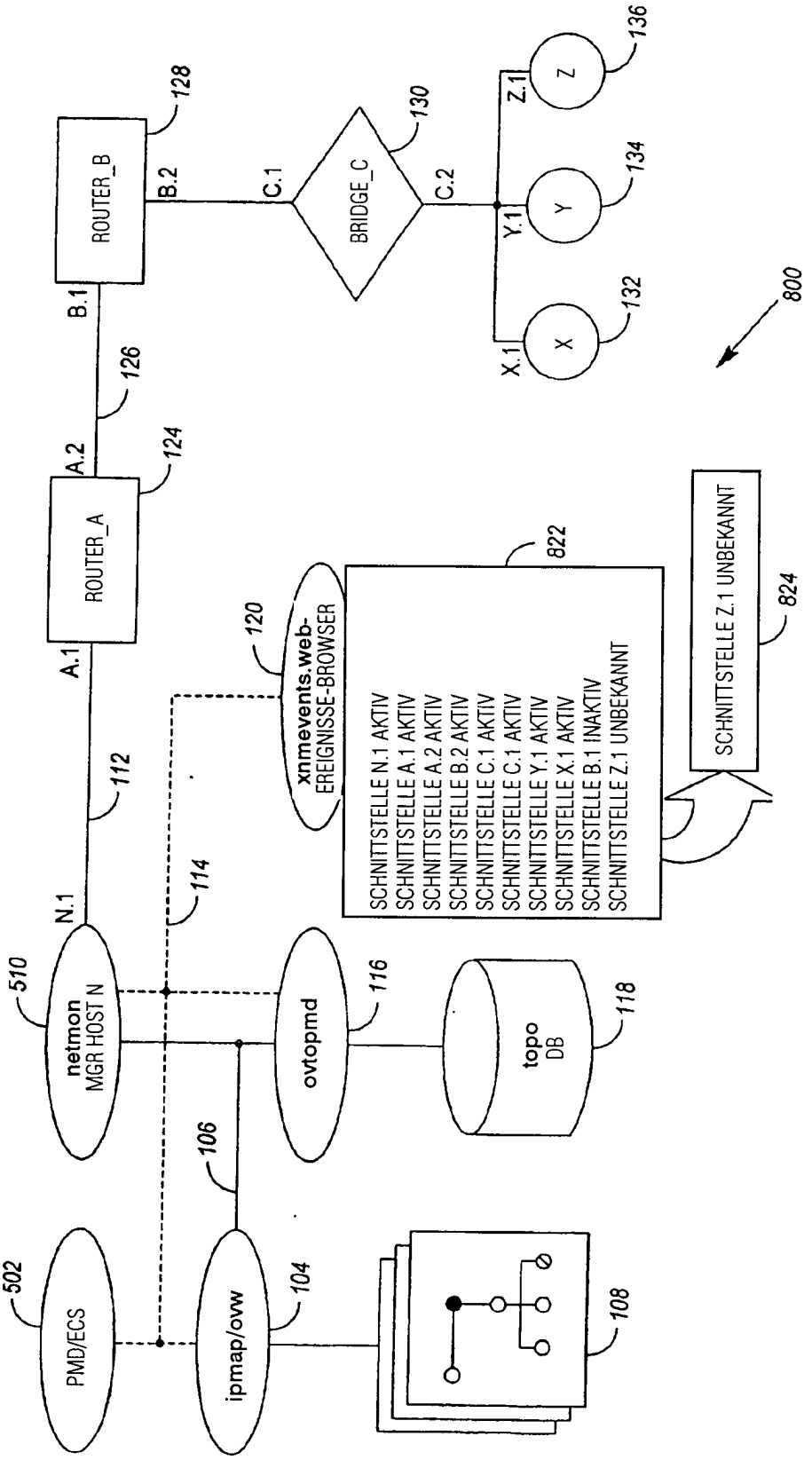
FIGUR 5


600
EREIGNIS TRITT IN ECS-MASCHINE EIN
602
STATUS-VERÄNDERN-EREIGNIS?
604
JA
NEIN
ZUSÄTZLICHES VARBIND?
606
JA
NEIN
EREIGNIS MUSS EIN PRIMÄRAUSFALL SEIN
608
EREIGNIS MUSS EIN SEKUNDÄRAUSFALL SEIN
610
EREIGNIS MIT PRIMÄR-AUSFALL KORRELIEREN
612
A
614
KOMMENTARKNOTEN BEFRAGT DEN KOMMENTARSERVER, UM INFORMATIONEN BEZÜGLICH DER KLASSE DES KNOTENS ZU ERHALTEN, DER DEM EREIGNIS UND DEN WERT DES SEC_FAIL_EVENT_SUPPRESS_SWITCH ZUGEORDNET IST
616
SEC_FAIL_EVENT_-SUPPRESS_SWITCH AUF WAHR GESTELLT?
624
JA
NEIN
KNOTEN UNTERDRÜCKT
628
EREIGNIS AN ANDERE SCHALTUNGEN UND MÖGLICHERWEISE NACH DRAUSSEN SENDEN
626
KRITISCHER KNOTEN?
620
JA
NEIN
KNOTEN IST EIN REGU-LÄRER KNOTEN
622
NIEMALS EIN KRITISCHER-KNOTEN-EREIGNIS UNTERDRÜCKEN
618

FIGUR 6


700
ROUTER_B
128
B.2
C.1
BRIDGE_C
130
C.2
Z.1
Z
136
Y.1
Y
134
X.1
X
132
B.1
126
A.2
ROUTER_A
124
A.1
112
N.1
114
120
xnmevents.web-
EREIGNISSE-BROWSER
722
SCHNITTSTELLE N.1 AKTIV
SCHNITTSTELLE A.1 AKTIV
SCHNITTSTELLE A.2 AKTIV
SCHNITTSTELLE B.1 INAKTIV
SCHNITTSTELLE Z.1 INAKTIV
SCHNITTSTELLE B.2 UNBEKANNT
SCHNITTSTELLE C.1 UNBEKANNT
SCHNITTSTELLE C.2 UNBEKANNT
SCHNITTSTELLE Z.1 INAKTIV
SCHNITTSTELLE Y.1 UNBEKANNT
SCHNITTSTELLE X.1 UNBEKANNT
724
510
netmon
MGR HOST N
ovtopmd
116
topo
DB
118
106
502
PMD/ECS
ipmap/ovw
104
108

FIGUR 7


ROUTER_B
128
B.2
C.1
BRIDGE_C
130
C.2
Z.1
Z
136
Y.1
Y
134
X.1
X
132
800
B.1
126
A.2
124
ROUTER_A
A.1
120
xnmevents.web-
EREIGNISSE-BROWSER
822
SCHNITTSTELLE N.1 AKTIV
SCHNITTSTELLE A.1 AKTIV
SCHNITTSTELLE A.2 AKTIV
SCHNITTSTELLE B.2 AKTIV
SCHNITTSTELLE C.1 AKTIV
SCHNITTSTELLE C.1 AKTIV
SCHNITTSTELLE Y.1 AKTIV
SCHNITTSTELLE X.1 AKTIV
SCHNITTSTELLE B.1 INAKTIV
SCHNITTSTELLE Z.1 UNBEKANNT
SCHNITTSTELLE Z.1 UNBEKANNT
824
112
N.1
114
510
netmon
MGR HOST N
ovtopmd
116
topo
DB
118
106
502
PMD/ECS
ipmap/ovw
104
108

FIGUR 8