

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-257481

(P2007-257481A)

(43) 公開日 平成19年10月4日(2007.10.4)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 13/00 (2006.01)	G06F 13/00 351N	2C061
G06F 3/12 (2006.01)	G06F 3/12 D	5B021
B41J 29/38 (2006.01)	G06F 3/12 K	5B089
B41J 29/46 (2006.01)	B41J 29/38 Z	
	B41J 29/46 Z	

審査請求 未請求 請求項の数 22 O L (全 21 頁)

(21) 出願番号 特願2006-83209 (P2006-83209)
 (22) 出願日 平成18年3月24日 (2006.3.24)

(71) 出願人 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号

(74) 代理人 100090538
 弁理士 西山 恵三

(74) 代理人 100096965
 弁理士 内尾 裕一

(72) 発明者 山内 久幸
 東京都大田区下丸子3丁目30番2号キヤ
 ノン株式会社内

Fターム(参考) 2C061 AP01 BB10 HH03 HJ08 HK17
 HN02 HN15 HV09 HV35
 5B021 AA01 BB10 EE01 NN19
 5B089 GA01 GA11 GA13 GA21 HA06
 JA35 KA13

(54) 【発明の名称】 印刷デバイス

(57) 【要約】

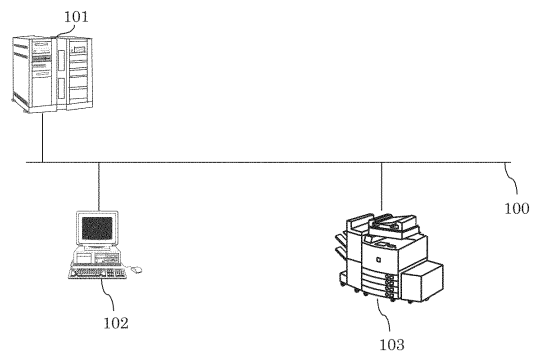
【課題】 印刷デバイスのイベント発行時に、無駄なイベント送信を行なうことを軽減する。

【解決手段】 イベント送信元となるデバイスは、イベント送信失敗あて先リストを保持する。

イベント送信が失敗した場合には、失敗カウントログを残してイベント送信失敗あて先リストへ追加する。

このイベント送信失敗あて先リストのカウント数が一定の上限を超えた場合には、そのホストに次回イベントを送信するタイミングのときに、イベント送信を行わない、または、イベント発行宛先の登録を削除する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

少なくとも1つ以上のネットワークサービスを利用・提供可能であり、そのサービスに関連するイベント発行および送受信が可能な印刷デバイスにおいて、前記印刷デバイスはイベント発行・通知手段を有し、

前記印刷デバイスは、イベント発行対象となるイベント送信先ホストの宛先情報が含まれるイベント発行宛先リストと、

イベント発行に失敗したホストの宛先情報と、イベント発行に失敗した回数を含むイベント送信失敗宛先リストを保持することが可能であって、

前記印刷デバイスは、前記イベント発行宛先リストに、イベント発行対象となるイベント送信先ホストの宛先情報を登録・削除することが可能であって、 10

前記印刷デバイスは、前記イベント発行宛先リストに登録されているホストの宛先情報に基づき、イベントを送信しイベント発行することが可能であって、

さらに、前記印刷デバイスから発行したイベントを、イベント送信先ホストが受信できるか否かを判断する手段と、前記判断手段によってイベントを前記イベント送信先ホストが受信できないと判断した場合には、イベント送信失敗宛先リストに、そのホストの宛先情報と受信に失敗した結果を登録する手段を有するイベント発行履歴登録手段を備えることを特徴とする印刷デバイス。

【請求項 2】

請求項1に記載の印刷デバイスにおける、イベント発行・通知手段において、 20

前記印刷デバイスは、イベント発行対象となるイベント送信先ホストの宛先情報が含まれるイベント発行宛先リストと、

イベント発行に失敗したホストの宛先情報と、イベント発行に失敗した回数と、イベント発行に失敗した時刻情報を含むイベント送信失敗宛先リストを保持することが可能であって、

前記印刷デバイスは、前記イベント発行宛先リストに、イベント発行対象となるイベント送信先ホストの宛先情報を登録・削除することが可能であって、

前記印刷デバイスは、前記イベント発行宛先リストに登録されているホストの宛先情報に基づき、イベントを送信しイベント発行することが可能であって、

前記印刷デバイスから発行したイベントを、イベント送信先ホストが受信できるか否かを判断する手段と、前記判断手段によってイベントを前記イベント送信先ホストが受信できないと判断した場合には、イベント送信失敗宛先リストに、そのホストの宛先情報と受信に失敗した結果を時刻情報とともに登録する手段を有するイベント発行履歴登録手段を備えることを特徴とする印刷デバイス。 30

【請求項 3】

請求項1、又は請求項2に記載の印刷デバイスにおけるイベントを受信できるか否かを判断する手段は、イベント発行・通知を行う時のコネクション型の通信において、イベント送信先ホストに対する通信コネクションを確立することができなかつたときに、イベント受信ができなかつたことを判断する手段を備えることを特徴とする印刷デバイス。

【請求項 4】

請求項1、又は請求項2に記載の印刷デバイスにおけるイベントを受信できるか否かを判断する手段は、イベント発行・通知を行う時のコネクション型の通信において、イベント送信先ホストとの通信コネクションを確立し、イベントデータを送信した後に、イベント送信先ホストから、そのイベントデータ受信にエラーが発生した旨の応答を返したときに、イベント受信ができなかつたことを判断する手段を備えることを特徴とする印刷デバイス。 40

【請求項 5】

請求項1、又は請求項2に記載の印刷デバイスにおけるイベントを受信できるか否かを判断する手段は、イベント発行・通知を行う時のコネクション型の通信において、イベント送信先ホストとの通信コネクションを確立し、イベントデータを送信した後に、イベン 50

ト送信先ホストが、一定時間の間に、その送信イベントに対する応答を返さなかったときに、イベント受信ができなかったことを判断する手段を備えることを特徴とする印刷デバイス。

【請求項 6】

請求項 1、又は請求項 2 に記載の印刷デバイスにおけるイベントを受信できるか否かを判断する手段は、イベント発行・通知を行う時のコネクションレス型の通信において、イベント送信後にイベント送信先ホストから送信されるイベントに対する応答として、イベント受信にエラーが発生した旨の応答を返したときに、イベント受信ができなかったことを判断する手段を備えることを特徴とする印刷デバイス。

【請求項 7】

請求項 1、又は請求項 2 に記載の印刷デバイスにおけるイベントを受信できるか否かを判断する手段は、イベント発行・通知を行う時のコネクションレス型の通信において、イベント送信後にイベント送信先ホストから送信されるイベントに対する応答が一定時間の間に返却されなかったときに、イベント受信ができなかったことを判断する手段を備えることを特徴とする印刷デバイス。

10

【請求項 8】

請求項 1、又は請求項 2 に記載のイベント発行履歴登録手段は、請求項 3 から 7 に記載の判断手段によって判断されたイベント送信の結果に応じて、イベント送信失敗宛先リストにイベント送信先のホストを登録するか否かを決定することが可能な手段を備えることを特徴とする印刷デバイス。

20

【請求項 9】

請求項 1、又は請求項 2 に記載のイベント発行履歴登録手段は、イベント送信に失敗した場合に、発行するイベントの内容に応じて、イベント送信失敗宛先リストにイベント送信先のホストを登録するか否かを決定することが可能な手段を備えることを特徴とする印刷デバイス。

【請求項 10】

請求項 1、又は請求項 2 に記載の印刷デバイスにおける、前記イベント発行履歴登録方法において、イベント受信に失敗したホストが前記イベント送信失敗宛先リストにすでに登録されている場合には、前記イベント送信失敗宛先リストのそのホストのイベント受信失敗回数のカウントを増やす手段を有するイベント発行履歴登録手段を備えることを特徴とする印刷デバイス。

30

【請求項 11】

請求項 2 に記載の印刷デバイスにおける、前記イベント発行履歴登録方法において、イベント受信に失敗したホストが前記イベント送信失敗宛先リストにすでに登録されている場合には、前記イベント送信失敗宛先リストのそのホストのイベント受信失敗回数のカウントを増やす手段と、そのイベント受信に失敗した時刻情報を更新する手段を有するイベント発行履歴登録手段を備えることを特徴とする印刷デバイス。

【請求項 12】

請求項 10、又は請求項 11 に記載のイベント受信失敗回数のカウントを増やす手段は、請求項 3 から請求項 7 に記載の判断手段によって判断された結果、またはイベントの内容に応じて、受信失敗回数のカウントを増やすか否かを決定する手段を備えることを特徴とする印刷デバイス。

40

【請求項 13】

請求項 11 に記載のイベント受信に失敗した時刻情報を更新する手段は、請求項 3 から請求項 7 に記載の判断手段によって判断された結果、またはイベントの内容に応じて、イベント受信に失敗した時刻情報を更新するか否かを決定することが可能な手段を備えることを特徴とする印刷デバイス。

【請求項 14】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、登録された前記送信失敗宛先リストに登録されており、かつ、あらかじめ設定された、

50

既定のイベント発行宛先に登録されているホストにイベント送信をする場合に、そのホストのイベント送信失敗カウン트가設定された回数の制限を越えているかを判断するイベント発行判断手段と、超えた場合には、イベント送信を行わないイベント発行手段を備えることを特徴とする印刷デバイス。

【請求項 15】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、登録された前記送信失敗宛先リストに登録されており、かつ、あらかじめ設定された、既定のイベント発行宛先に登録されているホストにイベント送信をする場合に、そのホストのイベント送信失敗カウン트가設定された回数の制限を越えているかを判断する請求項 14 に記載のイベント発行判断手段と、設定された回数を超えた場合には、イベント送信を行なうタイミングを毎回ではなく、一定回数の間隔をあけて送信するように変更するイベント発行手段を備えることを特徴とする印刷デバイス。

10

【請求項 16】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、登録された前記送信失敗宛先リストに登録されており、かつ、あらかじめ設定された、既定のイベント発行宛先に登録されているホストにイベント送信をする場合に、そのホストのイベント送信失敗カウン트가設定された回数の制限を越えているかを判断する請求項 14 に記載のイベント発行判断手段と、設定された回数を超えた場合には、イベント送信を行なうタイミングを毎回ではなく、予め定めた数の倍数ごと間隔をあけてイベント送信するように変更するイベント発行手段を備えることを特徴とする印刷デバイス。

20

【請求項 17】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、登録された前記送信失敗宛先リストに登録されており、かつ、あらかじめ設定された、既定のイベント発行宛先に登録されているホストにイベント送信をする場合に、そのホストのイベント送信失敗カウン트가設定された回数の制限を越えているかを判断するイベント発行判断手段と、超えた場合には、送信するイベントの内容に応じてイベント送信を行わないイベント発行手段を備えることを特徴とする印刷デバイス。

【請求項 18】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、登録された前記送信失敗宛先リストに登録されており、かつ、あらかじめ設定された、既定のイベント発行宛先に登録されているホストにイベント送信をする場合に、請求項 14 に記載のイベント発行判断手段によって送信を行なわないと判断された場合には、前記既定のイベント発行宛先からそのホストの登録を削除するイベント発行宛先削除手段を備えることを特徴とする印刷デバイス。

30

【請求項 19】

規定のイベント発行宛先に登録されているホストに対して、イベント送信に失敗し、送信失敗宛先リストに登録されたばあいにも、請求項 18 に記載のイベント発行宛先削除方法による宛先削除を行なうことを抑止することが可能な、イベント発行宛先削除抑止手段を備えることを特徴とする印刷デバイス。

【請求項 20】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、前記送信失敗宛先リストに登録されたホストに、イベントの送信に失敗してから一定時間経過した後、そのホストに対してホストの存在確認をおこなうイベントを発行または、通信を行ない、イベント発行が可能か否かをチェックするイベント発行可否のチェック手段と、前記チェック手段によってイベント発行が可能であると判断した場合には、イベント送信失敗宛先リストからそのホストを削除するイベント送信失敗宛先リストからの削除手段と、

40

さらに、イベント送信失敗宛先リストに登録されているホストが、請求項 18 に記載のイベント発行宛先削除手段によって、規定のイベント発行宛先リストから削除されていた場合に、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗

50

宛先リストからそのホストを削除するとともに、再びイベント発行を行なうことができるように規定のイベント発行宛先リストへ再登録するイベント発行宛先再登録手段を備えることを特徴とする印刷デバイス。

【請求項 2 1】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、前記送信失敗宛先リストに登録されたホストに対して手動でホストの存在確認をおこなうイベントを発行または、通信を行ない、イベント発行が可能か否かをチェックするイベント発行可否のチェック手段と、前記チェック手段によってイベント発行が可能であると判断した場合には、イベント送信失敗宛先リストからそのホストを削除するイベント送信失敗宛先リストからの削除手段と、

10

さらに、イベント送信失敗宛先リストに登録されているホストが、請求項 1 8 に記載のイベント発行宛先削除手段によって、規定のイベント発行宛先リストから削除されていた場合に、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗宛先リストからそのホストを削除するとともに、再びイベント発行を行なうことができるように規定のイベント発行宛先リストへ再登録するイベント発行宛先再登録手段を備えることを特徴とする印刷デバイス。

【請求項 2 2】

請求項 1、又は請求項 2 に記載のデバイスにおけるイベント発行履歴登録方法によって、前記送信失敗宛先リストに登録されたホストに、イベントの送信に失敗してから一定時間経過した場合や、あるいは手動で、そのホストに対して存在確認をおこなうイベントを発行または、通信を行ない、イベント発行が可能か否かをチェックするイベント発行可否のチェック手段と、前記チェック手段によってイベント発行が可能であると判断した場合には、イベント送信失敗宛先リストのカウントを減らす手段と、カウント情報が 0 になったら、イベント送信失敗宛先リストからそのホストを削除するイベント送信失敗宛先リストからの削除手段と、

20

さらに、イベント送信失敗宛先リストに登録されているホストが、請求項 1 8 に記載のイベント発行宛先削除手段によって、規定のイベント発行宛先リストから削除されていた場合に、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗宛先リストのカウントを減らす手段と、カウント情報が 0 になったら、イベント送信失敗宛先リストからそのホストを削除するとともに、再びイベント発行を行なうことができるように規定のイベント発行宛先リストへ再登録するイベント発行宛先再登録手段を備えることを特徴とする印刷デバイス。

30

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、ネットワーク上に接続された印刷デバイスのイベント通知の手段に関する。

【背景技術】

【0 0 0 2】

近年、インターネットなど大規模なネットワークの普及に伴い、ネットワーク技術が飛躍的に向上している中で、XML、SOAPなどのWWW関連の技術を用いて、ネットワークで接続された機器間の連携によって、さまざまなネットワーク上のサービスを利用できるようにしたWebサービスと呼ばれる技術が注目を集めている。その中でも、ネットワーク上に接続された端末から、ネットワーク上からの印刷要求を受け付け、印刷を実行することが可能な複数の印刷デバイスを登録あるいは検索し、所望の機能を有した印刷デバイスから印刷を行うプリントサービスと呼ばれるサービスが利用されつつある。

40

【0 0 0 3】

このようなサービスを提供する機器は、サービスを開始していることを通知するとともに、機器情報やサービス内容をネットワーク上に公開する。ネットワークに接続されている端末装置はそれらの公開されている情報を取得、あるいは検索することによって、ネットワーク上のサービスにアクセスすることが可能となる。

50

【0004】

上記のようなプリントサービスなど各種サービスを提供する印刷デバイスのサービスを利用する場合において、印刷デバイスの印刷ジョブのステータスが変化した場合には、印刷デバイスは、印刷ジョブを要求したホストや、あらかじめ登録されているホストに対して、印刷ジョブのステータス変化を告げる旨のイベントを発行する。このようなイベント通知によって、印刷デバイスのプリント処理終了などのジョブ状態を検知することができる。

【0005】

また、同様に、紙/トナー切れや紙ジャム発生時などによって印刷デバイスが印刷不可になるなど、デバイス自身のステータスが変化した場合にも、デバイスの状態変化を告げる旨のイベントを発行する。このようなイベント通知によって、サービスを利用するホストは印刷デバイスの状態変化を検知することができる。

10

【0006】

特にネットワーク上に複数の印刷デバイスが接続され、おのこの印刷デバイスが印刷サービスなどのサービスを提供しているような環境においては、サービス提供機器である複数の印刷デバイスを、統括的にサーバなどのデバイス管理用機器が、各デバイスの状態に不具合が無いか監視している。各印刷デバイスは、あらかじめ登録しておいた管理サーバに対して印刷デバイスのエラー発生等の状態をイベントによって通知する。これによって印刷デバイスのエラー発生等の状態変化を統括的に管理することができる。

【0007】

又、別の従来例としては、特許文献1をあげることが出来る。

20

【特許文献1】特開平11-266248号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

しかし、このようなイベントの発行・通知を行なう場合に、イベント受信側ホストが、何らかの問題によってイベントを受信することができない事態が発生する。

【0009】

このような状況下で頻りにイベント通知を行なう必要がある場合には、そのイベント発行の必要があるごとに毎回、不通となっているホストに対して無駄なイベント送信を行なう事となる。また、不通となっているホストに無駄なイベント送信をすることでネットワークトラフィックが増大するといった問題があった。

30

【課題を解決するための手段】

【0010】

少なくとも1つ以上のネットワークサービスを利用・提供可能であり、そのサービスに関連するイベント発行および送受信が可能な印刷デバイスにおいて、前記印刷デバイスはイベント発行・通知手段を有し、

前記印刷デバイスは、イベント発行対象となるイベント送信先ホストの宛先情報が含まれるイベント発行宛先リストと、

イベント発行に失敗したホストの宛先情報と、イベント発行に失敗した回数と、イベント発行に失敗した時刻情報を含むイベント送信失敗宛先リストを保持することが可能であって、

40

前記印刷デバイスは、前記イベント発行宛先リストに、イベント発行対象となるイベント送信先ホストの宛先情報を登録・削除することが可能であって、

前記印刷デバイスは、前記イベント発行宛先リストに登録されているホストの宛先情報に基づき、イベントを送信しイベント発行することが可能であって、

前記印刷デバイスから発行したイベントを、イベント送信先ホストが受信できるか否かを判断する手段と、前記判断手段によってイベントを前記イベント送信先ホストが受信できないと判断した場合には、イベント送信失敗宛先リストに、そのホストの宛先情報と受信に失敗した結果、または結果と時刻情報を登録する手段を有するイベント発行履歴登録

50

手段を有する。

【0011】

さらに、前記印刷デバイスにおけるイベントを受信できるか否かを判断する手段は、イベント発行・通知を行う時のコネクション型の通信において、

イベント送信先ホストとの通信コネクションを確立し、イベントデータを送信した後に、イベント送信先ホストから、そのイベントデータ受信にエラーが発生した旨の応答を返したときに、イベント受信ができなかったことを判断する手段と、

イベント送信先ホストとの通信コネクションを確立し、イベントデータを送信した後に、イベント送信先ホストが、一定時間の間に、その送信イベントに対する応答を返さなかったときに、イベント受信ができなかったことを判断する手段と、

10

さらに、イベント発行・通知を行う時のコネクションレス型の通信において、

イベント送信後にイベント送信先ホストから送信されるイベントに対する応答として、イベント受信にエラーが発生した旨の応答を返したときに、イベント受信ができなかったことを判断する手段と、

イベント送信後にイベント送信先ホストから送信されるイベントに対する応答が一定時間の間に返却されなかったときに、イベント受信ができなかったことを判断する手段を有する。

【0012】

さらに、前記デバイスのイベント発行履歴登録手段は、

イベント送信の結果に応じて、イベント送信失敗宛先リストにイベント送信先のホストを登録するか否かを決定することが可能な手段と、

20

イベント発行・通知を行う時のコネクション型、またはコネクションレス型の通信におけるイベント送信に失敗した場合に、発行するイベントの内容に応じて、イベント送信失敗宛先リストにイベント送信先のホストを登録するか否かを決定することが可能な手段と、

イベント受信に失敗したホストが前記イベント送信失敗宛先リストにすでに登録されている場合には、前記イベント送信失敗宛先リストのそのホストのイベント受信失敗回数を負やす手段、またはそのイベント受信に失敗した時刻情報を更新する手段を有する。

【0013】

また、前記イベント受信失敗回数を増やす手段は、前記のイベント送信先ホストが受信できるか否かを判断する判断手段によって判断された結果に応じて、失敗回数増やすか否かを決定するが可能である。

30

【0014】

また、前記イベント受信に失敗した時刻情報を更新する手段は、前記のイベント送信先ホストが受信できるか否かを判断する判断手段によって判断された結果に応じて、イベント受信に失敗した時刻情報を更新するか否かを決定することが可能である。

【0015】

また前記デバイスは、登録された前記送信失敗宛先リストに登録されているホストにイベント送信をする際に、そのホストのイベント送信失敗カウントが設定された回数の制限を越えているかを判断するイベント発行判断手段を有し、さらに回数の制限を超えた場合には、

40

イベント送信を行わないイベント発行手段と、

イベント送信を行なうタイミングを毎回ではなく、一定回数の間隔をあけて送信するように変更するイベント発行手段と、

イベント送信を行なうタイミングを毎回ではなく、所定の数の倍数ごと間隔をあけてイベント送信するように変更するイベント発行手段と、

イベントの内容に応じてイベント送信するか否かを決定するイベント発行手段を有する。

【0016】

また、前記印刷デバイスは、前記イベント発行履歴登録方法によって、登録された前記

50

送信失敗宛先リストに登録されており、かつ、あらかじめ設定された、既定のイベント発行宛先に登録されているホストにイベント送信をする場合に、前記イベント発行判断手段によって送信を行わないと判断された場合には、前記既定のイベント発行宛先からそのホストの登録を削除するイベント発行宛先削除方法を有する。

【0017】

また、前記印刷デバイスは、前記イベント発行履歴登録方法によって、前記送信失敗宛先リストに登録されたホストに、イベントの送信に失敗してから一定時間経過した後、または手動でそのホストに対してホストの存在確認をおこなうイベントを発行し、イベントの送信が可能か否かをチェックするイベント送信可否のチェック手段と、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗宛先リストからそのホストを削除するイベント送信失敗宛先リストからの削除手段と、さらに、イベント送信失敗宛先リストに登録されているホストが、イベント発行宛先削除手段によって、規定のイベント発行宛先リストから削除されていた場合に、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗宛先リストからそのホストを削除するとともに、再びイベント発行を行なうことができるように規定のイベント発行宛先リストへ再登録するイベント発行宛先再登録手段を有する。

10

【0018】

また、前記デバイスは前記イベント発行履歴登録方法によって、前記送信失敗宛先リストに登録されたホストに、イベントの送信に失敗してから一定時間経過した場合や、あるいは手動で、そのホストに対して存在確認をおこなうイベントを発行し、イベントの送信が可能か否かをチェックするイベント送信可否のチェック手段と、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗宛先リストのカウント情報をデクリメントする手段と、カウント情報が0になったら、イベント送信失敗宛先リストからそのホストを削除するイベント送信失敗宛先リストからの削除手段と、さらに、イベント送信失敗宛先リストに登録されているホストが、イベント発行宛先削除手段によって、規定のイベント発行宛先リストから削除されていた場合に、前記チェック手段によってイベント送信に成功した場合には、イベント送信失敗宛先リストからそのホストを削除するとともに、再びイベント発行を行なうことができるように規定のイベント発行宛先リストへ再登録するイベント発行宛先再登録手段も有する。

20

【0019】

上記手段によって、イベントを受信することができないホストに対して、無駄なイベント送信を行なう事を回避することができる。特に、頻繁にイベント送信を行なう必要がある場合には、ネットワークトラフィックを軽減できる。特に、デバイスがイベントの送受信を行なう際には、送信側ホストであるデバイスが、不通状態の受信側ホストからのイベントレスポンス受信待ち状態になることを回避し、デバイスの処理負荷を軽減することができる。

30

【発明の効果】

【0020】

以上説明したように、本発明によれば、印刷デバイスがイベント発行する更に、イベント発行左記のホストが、何らかの理由によって、イベントを受信することができない場合に、無駄なイベント送信を行なう事を回避することができる。特に、頻繁にイベント送信を行なう必要がある場合には、ネットワークトラフィックを軽減できる。また、デバイスがイベント送受信を行なう際には、送信側ホストであるデバイスが、不通状態の受信側ホストからのイベントレスポンス受信待ち状態になることを回避し、デバイスの処理負荷を軽減することができる。

40

【発明を実施するための最良の形態】

【0021】

次に、本発明の詳細を実施例の記述に従って説明する。

【0022】

以下に本発明の実施例を示す。

50

【0023】

図1は本発明のサービスの利用・イベント通知を行う一実施の形態の構成を示す図である。

【0024】

本実施形態は、サービス監視するサーバ端末装置101、サービスを利用するクライアント端末装置102と、サービス提供機器である印刷デバイス103とから構成される。そして、これらのサーバ/クライアント端末装置101、102、印刷デバイス103は、ネットワーク100を介して接続され、相互にデータの通信を行うことができるようになっており、全てネットワーク対応デバイスである。なお、ここではサーバ装置101、クライアント端末装置102、印刷デバイス103を1つのみ示しているが、これらは複数であっても良い。

10

【0025】

また、本実施例では、サービス提供機器は印刷サービスにおけるイベント通知を行う印刷デバイスのみを記述しているが、提供するサービスはスキャンサービス、ストレージサービスなどの他のサービスであっても良い。また、本実施例では、サービス提供機器から端末装置へのイベント通知方法について述べるが、サーバ/クライアント端末装置が本発明によるイベント通知を行なう事が可能であっても良い。

【0026】

図2は、一般的な端末装置のハードウェア構成を示すブロック図である。図2において、200はコンピュータPCであり、図1におけるサーバ/クライアント端末装置101、102、103と同等である。PC200は、CPU201を備え、ROM202またはハードディスク(HD)211に記憶された、あるいはフロッピー(登録商標)ディスクドライブ(FD)212に格納されるプログラムを実行し、システムバス204に接続される各デバイスを総括的に制御し、本実施形態の各手段が構成される。

20

【0027】

203はRAMで、CPU201の主メモリ、ワークエリア等として機能する。205はキーボードコントローラ(KBC)で、キーボード(KB)209や不図示のマウス等のポインティングデバイスからの指示入力を制御する。

【0028】

206はCRTコントローラ(CRTC)で、CRTディスプレイ(CRT)210の表示を制御する。207はディスクコントローラ(DKC)で、ブートプログラム(起動プログラム:パソコンのハードやソフトの実行(動作)を開始するプログラム)、複数のアプリケーション、編集ファイル、ユーザファイルそしてネットワーク管理プログラム等を記憶するハードディスク(HD)211、及びフロッピー(登録商標)ディスク(FD)212とのアクセスを制御する。

30

【0029】

208はネットワークI/F制御部で、LAN213を介して、印刷デバイスなど、他のネットワーク機器と双方向のデータのやり取りを行う。なお、本実施形態においては、LAN213は図1におけるネットワーク100と同じものである。

【0030】

図3は、本発明の実施例における印刷デバイスのハードウェア構成を示すブロック図である。図3において、300は印刷デバイスであり、図1における印刷デバイス103と同等である。印刷デバイス300は、CPU(中央処理装置)301を備えている。このCPU301は、ROM(リードオンリーメモリ)302またはハードディスク310に格納されているプログラムを実行して、システムバス304に接続される各デバイスを総括的に制御し、本実施形態の各手段が構成される。303はRAM(ランダムアクセスメモリ)で、CPU301の主メモリ、ワークエリアとして機能する。305は、プリンタI/F制御部であり、プリンタ306を制御する装置である。上記MFP300のCPU301、ROM302またはハードディスク310に記憶されたプログラムにより、本実施形態の各手段が構成される。307は不揮発性のメモリNVRAMであり印刷デバイスの各種設定値を保存するためのものである。パネル制御部308は、オペレーションパネル309を制御し、各種情報の表示、使用者からの指示入力を行なう。ネットワークI/F制御部311は、LAN312とのデータの送受信を制御する。なお、本実施形態においては、LAN310は図

40

50

1におけるネットワーク100と同じものである。

【0031】

図20は本実施例でデバイス103が所有するユーザインターフェース部分であり、図3のオペレーションパネル309に相当する。313は大型タッチパネルで、ユーザはタッチパネルを操作することで、各種設定を行うことが可能である。図の画面はコピーの待機画面である。314はテンキーボタンで、数字などの値を入力するのに使う。Sのボタンはサービスボタンで、このボタンを押下することにより、タッチパネル上に各種サービス画面が出現し、コピー以外のサービスを行うことができる。Rのボタンは設定ボタンで、このボタンを押下することによって、タッチパネル上に、各種設定画面が出現し、パラメータの設定を行うことができる。

10

【0032】

図4は、本発明の実施例におけるサーバ/クライアント端末装置200のソフトウェア構成を示すブロック図である。印刷アプリケーション401は、ユーザの指示操作によって印刷を実行するアプリケーションであり、印刷処理制御部へ印刷データを渡す。印刷処理制御部402は、受信したデータを印刷可能なデータに変換するとともに、印刷部数、両面印刷等の印刷に関する様々な制御を行なう。イベント処理部403は本実施例におけるイベント通知処理を実行するための処理を行なうモジュールであり、イベントの送受信に関する処理を行なう。ネットワークドライバ408は、図2におけるネットワークI/F制御部208を制御し、ネットワーク409とのデータ送受信の制御を行う。TCP/IP制御部407は、TCP/IPプロトコルによるデータの送受信及び制御を行なう。HTTPクライアントは、HTTPプロトコルのクライアント機能を有しており、上位アプリケーションの指示によりHTTPリクエストパケットを印刷デバイスに返信するとともに、印刷デバイスから送信された、HTTPレスポンスパケットを解析して、データをSOAP制御部404、サービス検索処理部403や印刷処理制御部402等の上位アプリケーションにわたす処理を行なう。SOAP制御部404はSOAP(Simple Object Access Protocol)と呼ばれるプロトコルを制御するモジュールであり、端末装置から受信したXML(eXtensible Markup Language)形式のデータをXMLパーサ405を使用して解析し、印刷処理制御部402の適切なモジュールを呼び出したり、イベント処理部403へ印刷デバイスから送信されるイベントのデータを渡すと同時に、印刷デバイスに返すべきデータをXMLデータに変換し、HTTPサーバ制御部506を介して印刷デバイスに返信する制御を行なう。XMLパーサ505はXML形式データを入力とし解析結果を出力するモジュールである。

20

30

【0033】

図5は、本発明の実施例における印刷デバイス300のソフトウェア構成を示すブロック図である。プリンタ制御部501は、図3におけるプリンタI/F制御部305を制御して、図3におけるプリンタ306へのデータ送信、排紙制御などを行う。印刷処理制御部502は、受信したデータを印刷可能なデータに変換するとともに、印刷部数、両面印刷等の印刷に関する様々な制御を行なう。イベント処理部503は本実施例におけるイベント通知を実行するための処理を行なうモジュールであり、イベントに関する処理を行なう。ネットワークドライバ508は、図3におけるネットワークI/F制御部311を制御し、ネットワーク509とのデータ送受信の制御を行う。TCP/IP制御部507は、TCP/IPプロトコルによるデータの送受信及び制御を行なう。HTTPサーバは、HTTPプロトコルのサーバ機能を有しており、端末装置から受信したHTTPリクエストパケットを解析して、データをSOAP制御部504、サービス検索処理部503や印刷処理制御部502等の上位アプリケーションにわたすと共に、上位アプリケーションの指示によりHTTPレスポンスパケットを端末装置に返信する制御を行なう。SOAP制御部504はSOAP(Simple Object Access Protocol)と呼ばれるプロトコルを制御するモジュールであり、端末装置から受信したXML(eXtensible Markup Language)形式のデータをXMLパーサ505を使用して解析し、印刷処理制御部502の適切なモジュールを呼び出したり、イベント処理部503へ端末装置から送信されるイベントのデータを渡すと同時に、端末装置に返すべきデータをXMLデータに変換し、HTTPサーバ制御部506を介して端末装置に返信する制御を行なう。XMLパーサ505はXML形式データを入力とし解析結果を出力するモジュールである。

40

50

【 0 0 3 4 】

以下に本実施例における、クライアント端末装置から印刷デバイスであるMFPによって印刷を行う場合の印刷デバイスの動作について述べる。

【 0 0 3 5 】

図 6 は、本実施例における HTTP 上の SOAP を用いてクライアント端末装置から印刷デバイスに送信される CreateJob と称されるパケットデータの一例である。このデータは XML 形式で記述されている。CreateJob パケットは、印刷デバイスに対し、ジョブ（印刷）の開始を指示するコマンドであり、要求元のユーザ名（<requesting-user-name>タグ）やジョブの処理に関する指示（<job-instruction>タグ）等の情報が記述されている。<job-instruction>タグには印刷部数を設定する <copies>タグ、両面印刷を設定する <sides>タグ、印刷の仕上げを設定する <finishing>タグなどが含まれ、印刷デバイスはこれらのタグに設定された値に基づき、ジョブの処理を行う。さらに、<job-instruction>タグにはオプションとして <notification-instruction>タグを含めることも可能である。この <notification-instruction>タグにはジョブに関する通知情報が記述される。図 6 の例においては通知情報として、通知先を設定する <notification-recipient>タグおよび通知条件を設定する <event>タグが記述されている。印刷デバイスはこれらのタグに設定された値に基づいてイベント送信処理を行う。

10

【 0 0 3 6 】

図 7 は、図 6 の CreateJob パケットに対する応答パケットデータの一例である。本データも図 6 と同様に XML 形式で記述され、本実施例においては HTTP 上の SOAP を用いて送受信される。CreateJob 応答パケットには、CreateJob コマンドに対する結果コード（<result-code>タグ）や生成されたジョブの識別子（<job-id>タグ）、印刷用ポートの URI（<data-sink-uri>タグ）等の情報が含まれている。

20

【 0 0 3 7 】

図 8 は、クライアント端末装置が印刷を行う場合における本実施例の印刷デバイスの動作を示すフローチャートである。

【 0 0 3 8 】

以下、図 8 を用いて、本実施例の印刷デバイスの動作を説明する。

【 0 0 3 9 】

ホストコンピュータから例えば図 6 に示したような CreateJob パケットが印刷デバイスに送信されると、印刷デバイスは S801 において、CreateJob に記述された XML データを解析し、S802 に進み解析結果にエラーがないか判定する。エラーがなければ、S803 において印刷データ受信用の印刷ポートを生成する。S803 の処理が終わると S804 へ進み、CreateJob パケットに対するレスポンスの XML データの生成を行なう。このときに印刷データ受信用に生成したポートの URI を <data-sink-uri>タグの値として設定する。例えば、図 5 の例のような URI が XML データに埋め込まれる。CreateJob レスポンスデータの生成が終了すると S805 へ進み、そのデータを SOAP を用いてホストコンピュータに送信する。ホストコンピュータは、その後、<data-sink-uri>タグで指定された URI に対して HTTP の POST メソッドを用いて印刷データを送信する。図 9 は HTTP POST メソッドによる印刷データ転送パケットの一例である。印刷デバイスは S806 において印刷ポートに到着したデータを受信し、適切な処理をしながらプリンタに印刷する。S806 において、印刷データの受信が正常に終了すると印刷デバイスは S807 において図 10 に示した例のような HTTP レスポンスパケットをホストコンピュータに送信し、印刷用ポートを削除（クローズ）して印刷動作を終了する。一方、S802 において CreateJob に記述された XML データにエラーがあると判定された場合は S808 へ進み、エラーレスポンスデータを生成する。図 11 はエラーレスポンスの例である。次に S805 においてエラーレスポンスをホストコンピュータに送信すると、ホストコンピュータは印刷データを送信することはせず、処理を終了する。

30

40

【 0 0 4 0 】

次に、上記の本実施例における印刷サービスを提供する印刷デバイスの状態変化によるサーバ端末装置へのイベント通知手段、イベント失敗リスト登録手段、およびイベント発

50

行宛先の登録削除手段について述べる。

【0041】

図12は印刷デバイスがイベント通知を行う場合における本実施形態の印刷デバイスの動作を示すフローチャートである。

【0042】

以下、図12を用いて、本実施形態の印刷デバイスの動作を説明する。

【0043】

本実施形態における印刷デバイスは図1に示す印刷デバイス103であり、サーバ装置は図1に示すサーバ装置101、クライアント端末は図1に示すクライアント端末102である。

【0044】

印刷デバイスは図8に示した印刷処理での印刷ジョブのステータス変化や、紙ジャム、紙/トナー切れなどのエラーによる印刷デバイスの状態変化があった場合には、その状態変化をイベント発行によってサーバ装置あるいはクライアント端末に通知する。

【0045】

特に、紙/トナー切れなどのエラーによる印刷デバイスの状態変化があった場合には、図13に示すような、あらかじめイベント発行先のホストに関する情報が登録されているイベント発行宛先リストを保持しておき、そのリストに登録されているホストに対してイベントを発行する。このリストへの登録は、図3に示したオペレーションパネル等からユーザによって宛先が入力され、登録される。

【0046】

一方、リストからの削除についても同様にオペレーションパネル等からユーザによって削除可能である。

【0047】

本実施例では、イベントを発行するホストに関する情報としてIPアドレスをイベント発行宛先リスト記載してあるが、その他、MACアドレス、FQDN名などホストを識別できる情報であれば、どのような情報でもよい。

【0048】

印刷デバイスは前記したような、イベント発行の必要があった場合には、イベント発行宛先リストに登録されているホストに対してイベント発行する前に、ステップS1201において、そのホストが、図14に示したようなイベント送信失敗宛先リストに登録されているか否かを判断する。イベント送信失敗リストには、以前にイベント発行に失敗したホストに関する情報と、失敗したカウント数、およびイベント送信に失敗した時刻情報が記載されている。本実施例では、前記ホストに関する情報としてIPアドレスをイベント通知リスト記載してあるが、その他、MACアドレス、FQDN名などホストを識別できる情報であれば、どのような情報でもよい。

【0049】

また、本実施例では、イベント発行宛先リストとイベント送信失敗宛先リストの2つのリストを保持しているが、イベント発行宛先リストに、送信に失敗したカウント数、およびイベント送信に失敗した時刻情報などを記載することによって、イベント送信失敗宛先リストを保持しない構成をとっても良い。

【0050】

(すなわちイベント発行宛先リスト=イベント送信失敗宛先リスト)

イベント送信失敗リストにそのホストが登録されていない場合には、次のステップへ進み、ステップS1205において図15に示したようなイベントパケットデータを生成し、ステップS1206においてイベントパケットデータを送信することで、イベントを発行する。

【0051】

イベント発行先であるサーバ装置やクライアント装置のホストが、イベント受信可能であった場合には、図16に示したようなイベントの応答パケットデータが返信されるので、ステップS1207においてステップS1206で発行したイベントの応答が返信されたかを確認することで、イベントの発行に成功したか否かの判断を行い、成功した場合には処理を終

10

20

30

40

50

了する。

【0052】

本実施例におけるイベント発行が成功したか否かの判断は、TCPプロトコルなどのコネクション型の通信によって、イベント発行先ホストであるサーバ装置との通信コネクションを確立し、その確立したコネクションにおいて、発行したイベントに対する応答があるか否かで判断を行なっているが、通信コネクションそのものが確立できなかったり、イベント応答としてイベントを受信できなかった旨のエラーなどが返却された場合や、応答待ちのタイムアウトが発生した場合も、イベント発行が失敗したと判断しても良い。

【0053】

また、イベント発行に関する通信は、UDPプロトコルなどのコネクションレス型の通信であってもよく、イベントデータの送信後、同プロトコルで発行したイベントに対する応答を受け取ることで、イベント発行が成功したか否かを判断しても良い。 10

【0054】

従って、コネクション型の通信と同様に、イベント送信後にイベント送信先ホストから送信されるイベントに対する応答として、イベント受信にエラーが発生した旨の応答を返したときや、イベントに対する応答が一定時間の間に返却されなかったときに、イベント受信ができなかったときには、イベント発行が失敗したと判断しても良い。

【0055】

ステップS1207においてイベントの送信に失敗したと判断した場合には、ステップS1208において、そのホストがイベント送信失敗リストに登録されているか否かの判断を行い、イベント送信失敗リストに登録されていない場合には、ステップS1209においてイベント送信失敗宛先リストにそのホストを登録するとともに、ステップS1210においてイベント送信失敗宛先リスト内の、イベント送信失敗カウントを増やし、さらにその時刻情報を登録して処理を終了する。 20

【0056】

本実施例では、ステップS1207においてイベント送信に失敗した何れの場合も、イベント送信失敗リストに登録する、あるいはイベント送信失敗カウントを増やす構成をとっている。

【0057】

しかし、たとえば「通信のタイムアウトが発生した場合には、イベント送信失敗リストに登録しない、あるいはイベント送信失敗カウントを増さない」など、イベント発行の際のエラーの内容によって、イベント送信失敗リストへの登録やイベント送信失敗カウントを増やすか否かを判断しても良い。 30

【0058】

また、本実施例では何れのイベントにおいてもイベント送信失敗リストに登録する、あるいはイベント送信失敗カウントを増やす構成をとっている。

【0059】

しかし、イベント送信に失敗したとき、たとえばイベントの内容が

「印刷終了通知」である場合には、イベント送信失敗リストに登録する、あるいはイベント送信失敗カウントを増やす。 40

【0060】

「印刷デバイスの設定変更」である場合には、イベント送信失敗リストに登録しない、あるいはイベント送信失敗カウントを増やさない。

【0061】

など、イベントの内容によって、イベント送信失敗リストへの登録やイベント送信失敗カウントを増やすか否かを判断しても良い。

【0062】

ステップS1208において、イベント送信失敗リストに登録されている場合には、ステップS1209をスキップし、ステップS1210においてイベント送信失敗宛先リスト内の、イベント送信失敗カウントをインクリメントし、さらにその時刻情報を更新して処理を終了する 50

。

【0063】

本実施例では、ステップS1207においてイベントの送信に失敗したと判断したイベント送信先ホストが既にイベント送信失敗宛先リストに登録されている場合には、毎回、イベント送信失敗宛先リストの時刻情報を更新しているが、初回にイベント送信に失敗した時刻のみを登録するような構成をとっても良い。

【0064】

また、ステップS1201において、イベント送信失敗宛先リストに、イベントを送信するホストが登録されている場合には、ステップS1202へ進み、ステップS1202において、イベント送信失敗リストのそのホストのイベント送信失敗カウンターの値が、あらかじめ設定してある上限数を超過しているか否かの判断を行なう。カウンター数が上限を超えていない場合には、ステップS1205へ進み、前記したステップS1205からステップS1210までの処理を行なう。

10

【0065】

ステップS1202において、カウンター数が上限を超えていない場合には、ステップS1203へ進む。

【0066】

この場合、イベント送信を行わずに処理を終了する。

【0067】

本実施例では、印刷デバイスは、図3のオペレーションパネルからユーザによって、イベント送信失敗のカウンター数が一定の値を超えた場合に『イベント発行宛先からホストを削除する』という設定を行なうことができる。ステップS1203において、この設定がされているか否かの判断を行ない、削除の設定がされている場合にはステップS1204において、イベント発行宛先リストからこのホストを削除し、処理を終了する。

20

【0068】

本実施例では、イベント送信失敗リストのイベント送信失敗カウンターの値が一定の上限値を超えてしまったホストに対しては、イベント送信を行わない構成をとっているが、たとえば、上限値を5回と設定されている場合には、さらに失敗した数をカウントし

6回目	イベントを送信する
7回目	イベントを送信しない
8回目	イベントを送信する
9回目	イベントを送信しない

30

… …

などのように一定回数の間隔をあけて、イベント送信を行ったり、

5+1回目	イベントを送信する
5+2回目	イベントを送信する
5+4回目	イベントを送信する
5+8回目	イベントを送信する

… …

などのように一定倍数の回数ごとにイベント送信を行なうなど、イベントの発行間隔を変更するような構成をとっても良い。

40

【0069】

さらに、イベントの内容を解析し、その内容に応じてイベント送信を行わない構成をとっても良い。

【0070】

また、イベント発行先ホストのイベント発行宛先リスト登録時に、一定の上限値を超えた場合においても、イベント発行宛先リストからの削除を抑止するような設定をオペレーションパネルから行なうことで、リストからの削除しないようにする構成をとっても良い。

。

【0071】

50

本実施例では、前記した図12のステップS1202からS1203の処理によって、イベント送信失敗リストに登録されているホストに対して、リスト内の時刻情報を元に、イベント送信に失敗してから一定時間経過している場合には、ホスト存在確認を行なうイベントを発行し、そのホストがイベント受信可能であるかの検証を定期的に行ない、そのホストがイベント受信可能であった場合には、イベント送信失敗リストから削除する。

【0072】

また、イベント送信失敗リストに登録されているホストが、ステップS1204の処理によって、イベント発行宛先リストから削除されてしまっていた場合に、ホスト存在確認ができた場合には、そのホストをイベント発行宛先リストへ再登録を行なう。

【0073】

図15は、本実施例におけるHTTP上のSOAPを用いて印刷デバイスからサーバ装置あるいはクライアント端末に送信されるイベントパケットデータの一例である。このデータは図12のステップS1205で作成されるイベントパケットデータでありXML形式で記述されている。イベントパケットには、Notifyメソッドが用いられる。イベントパケットはサーバ装置あるいはクライアント端末に対し、ジョブ（印刷）または印刷デバイスの状態変化を通知するコマンドであり、イベントの内容（<trigger-event>タグ）やイベント発生時刻（<trigger-time>タグ）、イベントのID情報（<notification-id>タグ）等の情報が記述されている。本実施例でのイベントパケットデータには、ジョブの状態を通知するイベントの例（図15-A）とデバイスの状態を通知するイベントの例（図15-B）を記述している。ジョブの状態をイベント通知する場合には、<trigger-event>タグ内にジョブの終了（"job-completed"）やジョブのキャンセル（"job-canceled"）等のジョブのステータスを表す文字列が記述される。

【0074】

また、デバイスの状態をイベント通知する場合には、<trigger-event>タグ内にデバイスの状態変化を表す文字列"device-state-changed"が記述され、さらに<device-status>タグ内にデバイスの状態を表す文字列（"stop" etc.）と、<device-state-reason>タグ内にデバイスの状態変化の理由（紙ジャムやトナー残量が少ない）などを表す文字列（"paper-jam"、"toner-low"）が記述される。

【0075】

図16は、図15のイベントパケットに対する応答パケットデータの一例である。本データも図16と同様にXML形式で記述され、本実施例においてはHTTP上のSOAPを用いて送受信される。イベント応答パケットにはNotifyResponseメソッドが用いられ、イベントパケットのNotifyコマンドに対する結果コード（<result-code>タグ）が含まれている。正常にイベントを受信した場合には（<result-code>タグ内に"ok"が記述される。

【0076】

図17は印刷デバイスがイベント送信失敗リストに登録されているホストの存在確認を行なう動作と、イベント送信失敗リストに登録されているイベント発行宛先リストから削除されたホストをイベント発行宛先リストへ再登録する場合における、本実施形態の印刷デバイスの動作を示すフローチャートである。

【0077】

以下、図17を用いて、本実施形態の印刷デバイスの動作を説明する。

【0078】

本実施形態における印刷デバイスは図1に示す印刷デバイス103であり、サーバ装置は図1に示すサーバ装置101、クライアント端末は図1に示すクライアント端末102である。

【0079】

印刷デバイスは、一定時間間隔で図17に示した以下のステップの処理を定期的に行なう。この処理を実行するか否かの設定は図3に示したオペレーションパネルからユーザによって設定することが可能である。また、その時間間隔もユーザが指定・設定することが可能である。

【0080】

10

20

30

40

50

また、何れの時間が経過している場合にもデバイスのオペレーションパネルなどから手動で、上記の処理を実行させてもよい。

【0081】

まず、印刷デバイスはステップS1701において、イベント送信失敗宛先リストからイベント送信に失敗したホストの情報を取得する。

【0082】

次に、ステップS1702においてステップS1701でのホスト情報の取得に成功したか否かの判断を行い、成功した場合には次のステップに進む。ホスト情報の取得に失敗あるいは、イベント送信失敗宛先リストにホストの情報がない場合には、処理を終了する。

【0083】

次にステップS1703において、ステップS1701において取得したイベント受信の確認対処となるホストがイベント受信に失敗した時刻が現在時刻から一定時間経過しているか否かの判断を行なう。

【0084】

ステップS1703においてそのホストがイベント受信に失敗した時刻から、前記した設定時刻よりも時間が経過している場合には、次のステップへ進む。また、設定時刻よりも時間が経過していない場合には、ステップS1701に戻り、次のホスト情報を取得し、前記した同様の処理を行なう。

【0085】

S1704において、図18に示したようなホストの存在確認を行なうイベント発行のためのイベントパケットデータを作成し、S1705においてホストの存在確認イベントの発行送信処理を行なう。

【0086】

存在確認イベント発行先であるサーバ装置やクライアント装置のホストが、イベント受信可能であった場合には、図19に示したようなイベントの応答パケットデータが返信されるので、ステップS1706において、ステップS1705で発行したイベントの応答が返信されたかを確認することで、イベントの発行に成功したか否かの判断を行い、成功した場合には次のステップへ進む。また、イベント送信に失敗した場合には、S1710において、イベント送信失敗宛先リストの時刻情報を更新し、ステップS1701に戻り、次のホスト情報を取得し、前記した同様の処理を行なう。

【0087】

上記のイベント発行が成功したか否かの判断は、図12を用いて説明したイベント発行における判断と同様である。

【0088】

ステップS1706において送信先ホストが存在確認イベントの受信に成功し、レスポンスパケットデータを受信した場合には、送信ホストがイベントの受信が可能であると判断されるので、ステップS1708においてイベント送信失敗リストから削除する。さらに次のステップS1709においてイベント送信失敗リストから削除したホストが、イベント発行宛先リストに登録されているか否かを判断する。登録されていない場合には、図12に示したステップS1204において既にイベント発行宛先リストから削除されていることになるので、ステップS1709において、そのホストをイベント発行宛先リストにホストの情報を再登録する。

【0089】

ステップS1708においてイベント送信失敗リストから削除したホストが、イベント発行宛先リストに登録されている場合には、処理を行わずに、ステップS1701に戻り、次のホスト情報を取得し、前記した同様の処理を行なう。

【0090】

本実施例では、イベント送信失敗宛先リストに登録されているホストが、存在確認イベントの発行によって、イベントの受信が可能であると判断された場合には、いずれの場合も、イベント送信失敗宛先リストからの削除、または、イベント発行宛先リストへの再登

10

20

30

40

50

録を行なっているが、存在確認イベントの発行によって、ホストがイベントの受信が可能であると判断された場合には、イベント送信失敗宛先リストのイベント送信失敗カウントを減らしていき、最終的にカウントが0になったらイベント送信失敗リストからの削除や、イベント発行宛先リストにホストの情報を再登録するような構成をとってもよい。

【0091】

S1709までの処理が終了したら、ステップS1701に戻り、次のホスト情報を取得し、前記した同様の処理を行なう。

【0092】

イベント送信失敗宛先リスト内に登録されている全てのホストに対して、上記のステップを実行したら処理を終了する。

【0093】

本実施例においてイベント送信失敗宛先リスト内に登録されているホストの情報は、図3に示したオペレーションパネル等に表示可能であって、ユーザによってホストの宛先をリストから削除することも可能である。

【0094】

なお、本実施例においては、ダミーイベントを送信することによりホストの存在を確認していたが、例えば、Pingコマンドなどを使用しIPレベルでホストの存在を確認する構成とすることも可能である。送信先ホストの存在やイベント発行が可能であることが確認できれば何れの通信プロトコルを使用してもかまわない。

【0095】

図18は、本実施例におけるHTTP上のSOAPを用いて印刷デバイスからサーバ装置あるいはクライアント端末に送信されるホストの存在確認イベントの пакетデータの一例である。このデータは図17のステップS1705で作成されるイベント пакетデータであり、XML形式で記述されている。イベント пакетには、Notifyメソッドが用いられる。イベント пакетはサーバ装置あるいはクライアント端末に対し、ジョブ(印刷)または印刷デバイスの状態変化を通知するコマンドであり、イベントの内容(<trigger-event>タグ)やイベント発生時刻(<trigger-time>タグ)、イベントのID情報(<notification-id>タグ)等の情報が記述されている。本実施例におけるイベント пакетデータには、イベントの内容を表す<trigger-event>タグ内に、ホストが存在するか否か、またはイベントを受信可能であるか否かの確認を行なうためのダミーイベントを表す"dummy-event"が記述される。

【0096】

図19は、図18のイベント пакетに対する応答 пакетデータの一例である。本データも図16と同様にXML形式で記述され、本実施例においてはHTTP上のSOAPを用いて送受信される。イベント応答 пакетはNotifyResponseメソッドが用いられ、イベント пакетのNotifyコマンドに対する結果コードのみ(<result-code>タグ)が含まれている。正常にイベントを受信した場合には(<result-code>タグ内に"ok"が記述される。

【図面の簡単な説明】

【0097】

【図1】本実施例におけるサービス検索・印刷システムの構成図

【図2】端末装置のハードウェア構成ブロック図

【図3】印刷デバイスのハードウェア構成ブロック図

【図4】端末装置のソフトウェア構成ブロック図

【図5】印刷デバイスのソフトウェア構成ブロック図

【図6】印刷デバイスに送信されるCreateJob пакетデータの一例

【図7】CreateJob пакетに対する応答 пакетデータの一例

【図8】印刷実行時の印刷デバイスの動作を示すフローチャート

【図9】HTTP POSTメソッドによる印刷データ転送 пакетの一例

【図10】印刷終了時に印刷デバイスが送信するHTTPのレスポンス пакетの例

【図11】CreateJob пакетエラーのレスポンスの例

10

20

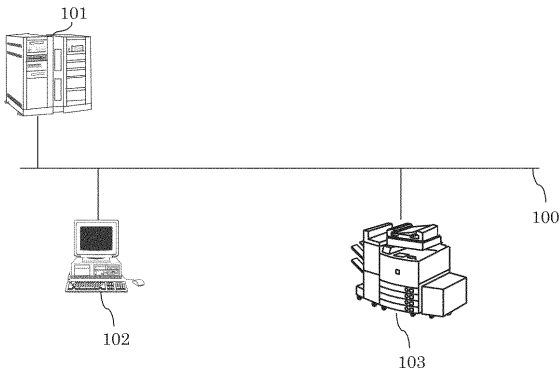
30

40

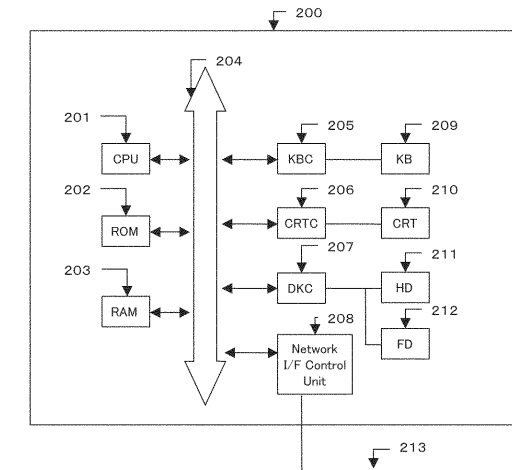
50

- 【図12】 イベント通知を行う場合における印刷デバイスの動作を示すフローチャート
- 【図13】 イベント発行宛先リストの一例
- 【図14】 イベント送信失敗宛先リストの一例
- 【図15】 イベントパッケージデータの一例
- 【図16】 イベント応答パッケージデータの一例
- 【図17】 イベント発行宛先リストへ再登録する場合の印刷デバイスの動作を示すフローチャート
- 【図18】 存在確認イベントパッケージデータの一例
- 【図19】 存在確認イベント応答パッケージデータの一例
- 【図20】 印刷デバイスが所有するユーザインターフェース部

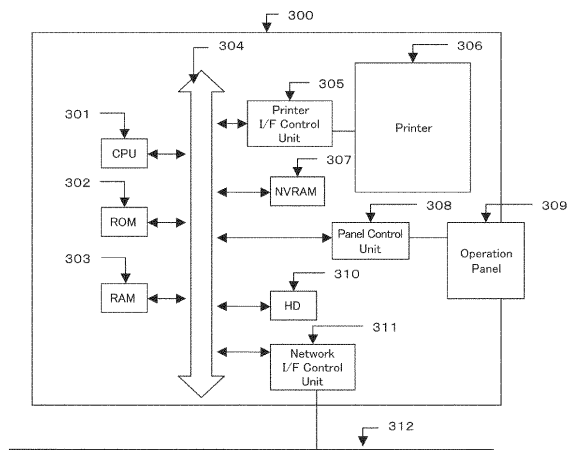
【図1】



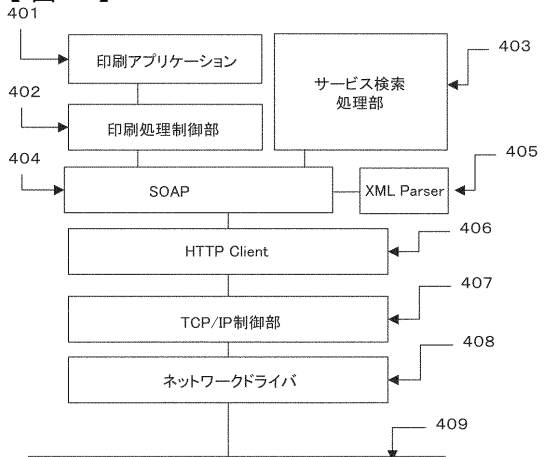
【図2】



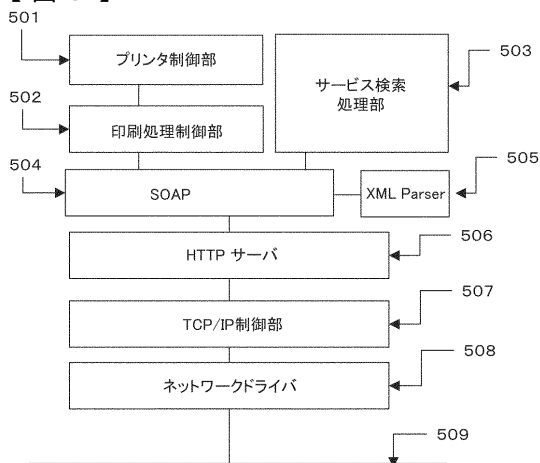
【図3】



【 図 4 】



【 図 5 】



【 図 6 】

```

<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope"
  env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
  xmlns:enc="http://www.w3.org/2001/12/soap-encoding">
  <env:Body>
    <bm:create_job xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
      <requesting-user-name>user1</requesting-user-name>
      <job-instruction>
        <job-instruction-params>
          <job-name>sample-job</job-name>
          <optional-attributes-fidelity>>false</optional-attributes-fidelity>
          <copies>1</copies>
          <sides>two-sided-long-edge</sides>
          <finishings enc:arrayType="bm:Finishing[1]">
            <finishing>staple</finishing>
          </finishings>
          <document-format>image/tiff</document-format>
        </job-instruction-params>
        <notification-instruction>
          <notification-recipient>http://192.168.1.5/event</notification-recipient>
          <events enc:arrayType="bm:Event[2]">
            <event>job-completed</event>
            <event>job-canceled</event>
          </events>
        </notification-instruction>
      </job-instruction>
    </bm:create_job>
  </env:Body>
</env:Envelope>

```

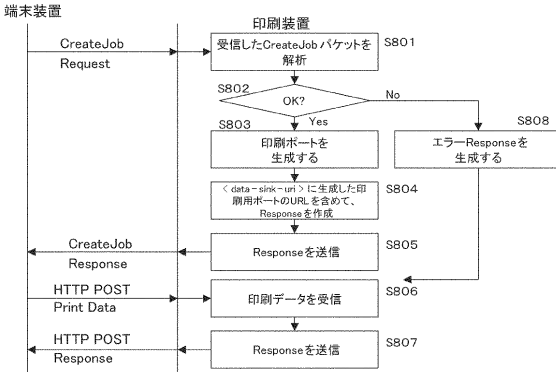
【 図 7 】

```

<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope"
  env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
  xmlns:enc="http://www.w3.org/2001/12/soap-encoding">
  <env:Body>
    <bm:create_jobResponse xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
      <result-code>ok</result-code>
      <job-id>1</job-id>
      <notification-subscription-id>1</notification-subscription-id>
      <data-sink-uri>
        http://192.168.1.4/job001
      </data-sink-uri>
    </bm:create_jobResponse>
  </env:Body>
</env:Envelope>

```

【 図 8 】



【 図 9 】

```

POST /job001 HTTP/1.1
Host: Canon-Printer
Content-Length: nnn

(...data...)

```

【 図 10 】

```

HTTP/1.1 200 OK
Connection: close

```

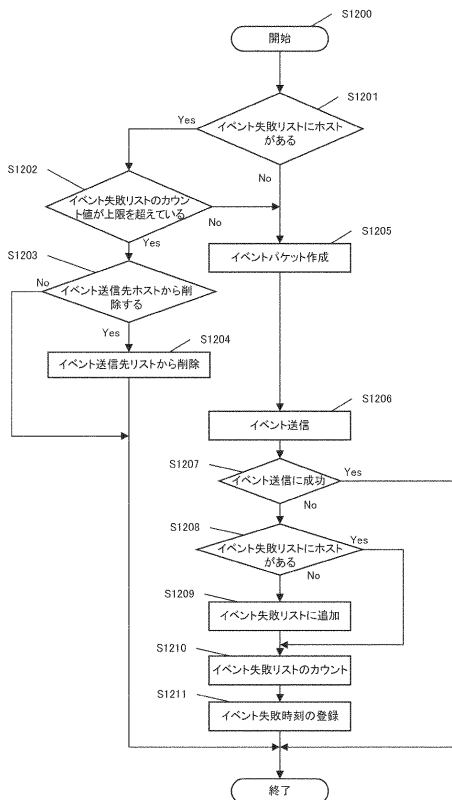
【 図 1 1 】

```

<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope"
  env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
  xmlns:enc="http://www.w3.org/2001/12/soap-encoding">
  <env:Body>
  <env:Fault>
  <faultcode>env:Sender</faultcode>
  <faultstring>Sender Error</faultstring>
  <detail>
  <bm:fault-information xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
  <result-code>conflicting-attributes</result-code>
  </bm:fault-information>
  </detail>
  </env:Fault>
  </env:Body>
  </env:Envelope>

```

【 図 1 2 】



【 図 1 3 】

イベント通知先
111. 111. 0. 1
111. 111. 0. 2
⋮

【 図 1 4 】

イベント通知先	カウンタ	時刻
111. 111. 0. 1	5	2003/12/1 10:00
111. 111. 0. 2	1	2003/12/1 9:00
⋮		

【 図 1 5 】

```

<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope"
  env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
  xmlns:enc="http://www.w3.org/2001/12/soap-encoding">
  <env:Body>
  <bm:notify xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
  <notification>
  /* job に関するイベント */
  <trigger-event>job-completed</trigger-event> ← A
  /* device status に関するイベント */
  <trigger-event>device-state-change</trigger-event>
  <device-state>stopped</device-state>
  <device-state-reason>paper-jam</device-state-reason> ← B
  <trigger-time>33333</trigger-time>
  <notification-id>001</notification-id>
  </notification>
  </bm:notify>
  </env:Body>
  </env:Envelope>

```

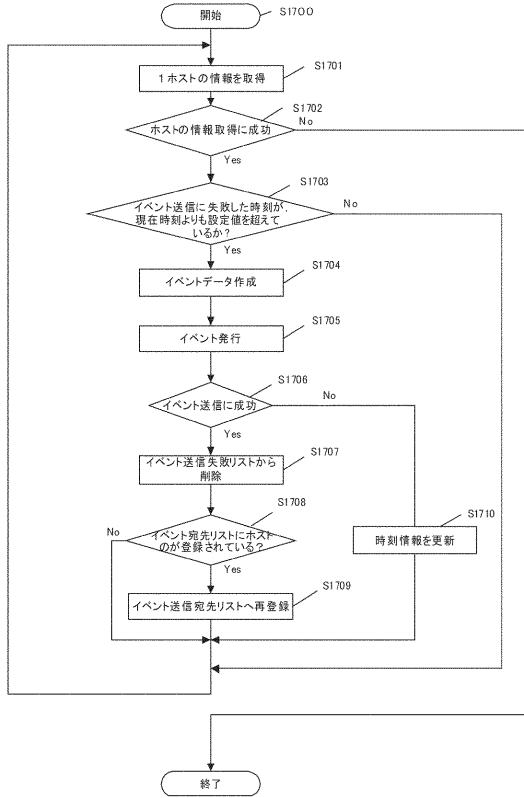
【 図 16 】

```

<env:Envelope xmlns:env="http://www. w3. org/2001/12/soap-envelope"
env:encodingStyle="http://www. w3. org/2001/12/soap-encoding"
xmlns:enc="http://www. w3. org/2001/12/soap-encoding">
  <env:Body>
    <bm:notifyResponse xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
      <notification>
        <result-code>ok</result-code/>
      </notification>
    </bm:notifyResponse>
  </env:Body>
</env:Envelope>

```

【 図 17 】



【 図 18 】

```

<env:Envelope xmlns:env="http://www. w3. org/2001/12/soap-envelope"
env:encodingStyle="http://www. w3. org/2001/12/soap-encoding"
xmlns:enc="http://www. w3. org/2001/12/soap-encoding">
  <env:Body>
    <bm:notify xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
      <notification>
        <trigger-event>dummy-event</trigger-event>
        <trigger-time>33333</trigger-time>
        <notification-id>001</notification-id>
      </notification>
    </bm:notify>
  </env:Body>
</env:Envelope>

```

【 図 19 】

```

<env:Envelope xmlns:env="http://www. w3. org/2001/12/soap-envelope"
env:encodingStyle="http://www. w3. org/2001/12/soap-encoding"
xmlns:enc="http://www. w3. org/2001/12/soap-encoding">
  <env:Body>
    <bm:notifyResponse xmlns:bm="urn:schemas-bmlinks-jp:service:bmlinks-1-2">
      <notification>
        <result-code>OK</result-code/>
      </notification>
    </bm:notifyResponse>
  </env:Body>
</env:Envelope>

```

【 図 20 】

