IPP Notification and Notification Services White Paper

Hugo Parra; Novell, Inc.
October 6, 1999

The intent of this paper is to supplement the discussions members of the IPP WG have had recently on the subject of an http-based notification protocol for IPP and the role of notification delivery services in IPP. Other documents similar to this one are expected from other IPP participants with their point of view on the same subject.

Notification Server-less Configuration

Figure 1 shows a possible notification scenario as described in the IPP notification spec (Tom's doc). It depicts how, in a simple configuration, all an IPP Client has to do to be notified of printer events of interest is to ask the printer through IPP (via Create-Subscription or a job creation operation). The only requirement for this to work is that the printer supports notification and that the client requests to be

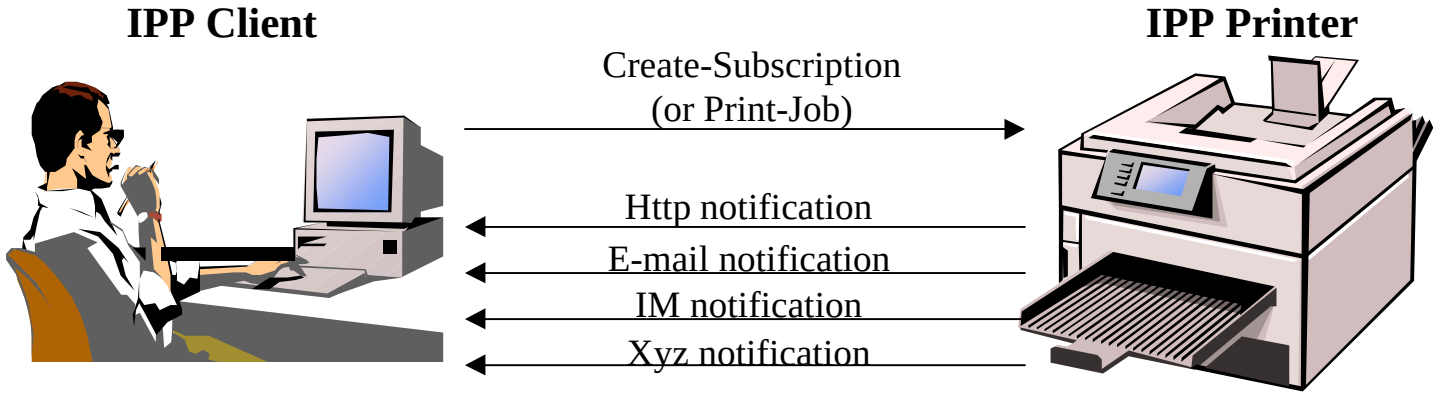

Figure 1

notified using one of the delivery methods supported by the printer. The Client finds out what delivery methods the printer supports by reading the 'notify-schemes-supported' printer attribute.

Notification Services

It's obvious, from recent discussions, that we're all familiar with notification services that vary widely in functionality and methodology. I believe IPP should make reasonable efforts to facilitate interoperability with the notification environments IPP printers are likely to be deployed in. While it might be a good thing to select a particular delivery method as "mandatory" for all IPP printers to implement, it doesn't make any sense to limit the type of delivery methods or notification environments IPP printers can support, unless allowing such requires a radical departure from our current model and acceptable design practices. Ultimately, it's not what notification service makes the most sense to some of us or how an IPP printer might interface with the notification service of our dreams that matters, but how IPP printers will work with the notification environments they're most likely to encounter.

A Notification Delivery Service

I highlight next the capabilities of a notification service like the one widely deployed in NetWare's print services. Others have mentioned different notification services that we should make sure IPP can play with as well. Those familiarized with other systems should make sure IPP's notification model accommodates interoperability with those systems.

An IPP printer would want to use a notification delivery service (a notification service with capabilities like the one I'm describing here) for the following reasons:

a) To support delivery methods the printer itself can't afford or doesn't wish to implement.

b) To support legacy, proprietary notification delivery methods.
c) To offload having to generate human readable notifications in different languages. [Optional]
d) To offload having to store subscriptions and to match events with interested parties and delivery methods. [Optional]

Notice that the notification delivery service is neutral with respect to the whole issue of notifications being able to cross Internet firewalls.

Figure 2 illustrates the different levels at which an IPP printer might wish to interface with a notification delivery service. The following scenarios are possibilities afforded by this multi-level support:

Scenario 1. The IPP printer implements one or more delivery methods but wishes to have access to additional, potentially proprietary, delivery methods supported by a notification delivery service available

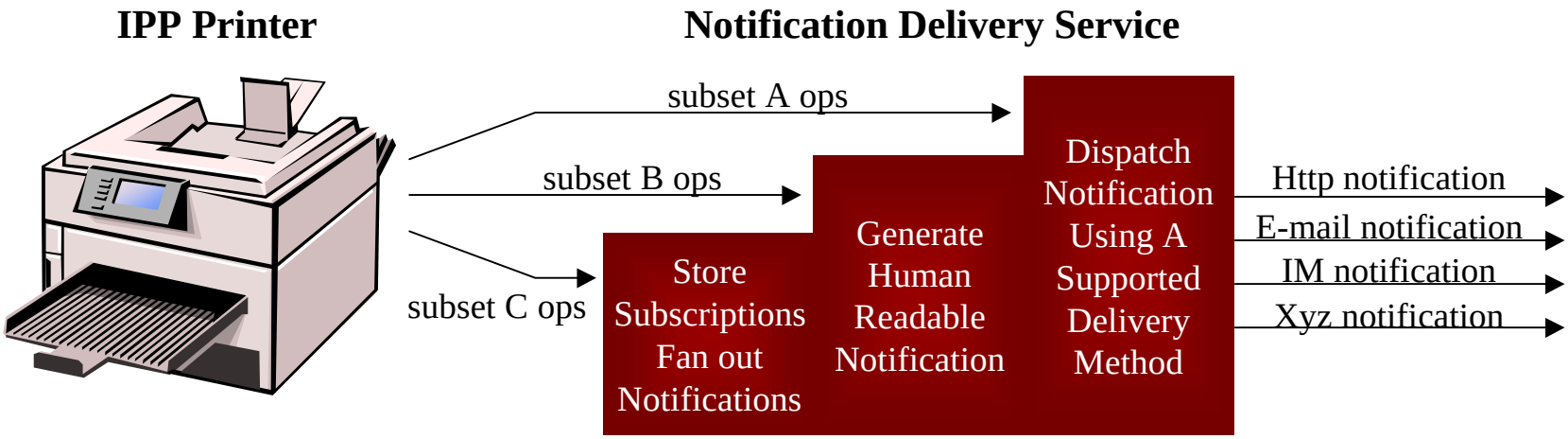

Figure 2

on the network. In the case of NetWare, these delivery methods may include a) Pop-up notification, b) Log-file via NCP, c) RPC-based notification to existing printing administrator and user tools, d) e-mail notification, and e) HTTP-based notification. The IPP printer only needs to implement the *subset A* operations of the Notification Delivery Service protocol, which is almost identical to the HTTP-Based Notification Protocol.

Scenario 2. The IPP printer is capable of storing several subscriptions but can't afford to provide human-consumable notifications in any language but English. The printer only implements the mandatory *x* delivery method which is insufficient in its current environment. By implementing the *subset B* operations of the Notification Delivery Service protocol it can extend its capabilities and meet the needs of its users.

Scenario 3. The IPP printer supports the mandatory single "per-job" and "per-printer" Subscription and is unable to meet its demand for notification. By implementing the *subset C* operations of the Notification Delivery Service protocol, such a printer can ask the notification server to store subscriptions (persistently or otherwise) in its behalf and generate the appropriate notifications for all interested parties each time the printer reports an event.

Figure 3 shows an IPP printer using notification delivery services available on the network to provide an IPP client with full notification support. A clear advantage of this setup is that the IPP client doesn't need to know what notification capabilities the IPP printer supports natively and what capabilities it's brokering out to the network. The client's interaction to the printer is exactly the same as the one depicted in Figure 1.

Impact On the IPP Model and Protocol

I agree with Carl-Uno's stand that the IPP WG is not in the business of inventing a full-fledged Notification

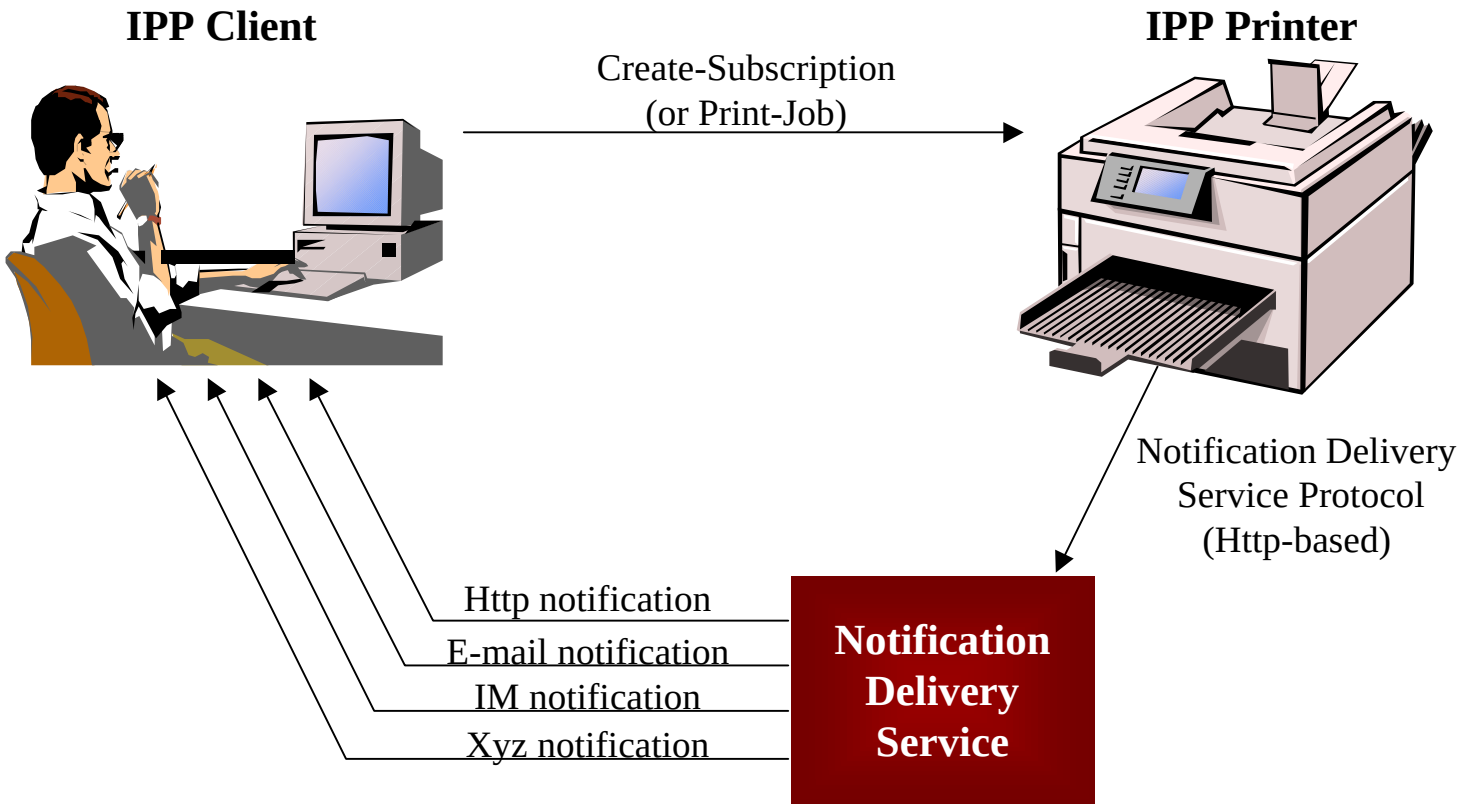

Figure 3

Service; what's important is to know what's required for IPP to support available services.

To support the flavor of Notification Delivery Services I described above, an IPP printer first needs to know when it is supposed to use the services. Exactly how this is accomplished may vary from printer to printer,

but I suggest that at the very least printers that support these type of notification services allow system administrators to configure them with the url of the service they're to employ. For this purpose I propose the 'supported-notification-delivery-services-uri' (1SetOf uri) printer attribute be defined. Printers may also be able to obtain these uri's from SLP or a directory.

When a printer's "supported-notification-delivery-services-uri" attribute is populated, the printer should contact each specified notification delivery service to get the notification schemes they support and use the information to populate its own 'notify-schemes-supported' attribute. Internally the printer should know what values of this attribute it supports natively and what values came from what notification delivery service. When a notification is generated that requires a delivery method not supported natively by the printer, the printer should send it to the appropriate notification delivery service for proper dispatching.

Notification Delivery Services Protocol

I'm currently working on the details of an HTTP-based protocol for talking to a type of notification delivery service like the one described in this paper. The protocol uses the IPP syntax and the semantics defined in the IPP Notification spec. I alluded earlier to three levels of support afforded by Notification Delivery Services; I include a short description of each one next.

Subset A Operations

IPP printers that only wish to use the Notification Delivery Services to add delivery methods to their list of 'notify-schemes-supported' may implement this subset of the protocol. Two operations make up subset A:

• ‘get-notify-schemes-supported’. An IPP printer may use this operation to get a list of the notification schemes supported by the notification delivery service. The printer should use this information to populate its ‘notify-schemes-supported’ attribute.
• ‘send-notification’. An IPP printer may use this operation to request that the notification delivery service use one of its supported delivery methods to dispatch a notification. The operation specifies the following information:
    a) The notification recipient’s url which the printer received as part the subscription.
    b) The data that comprises an “notification” as defined in the IPP Notification spec, including a human-consumable representation if the delivery method implied by the notification recipient’s url so requires it.

Subset B operations

IPP printers that, in addition to wishing to use the Notification Delivery Services to add delivery methods to their list of ‘notify-schemes-supported’, wish the notification delivery service to generate human-consumable notification data may implement this subset of the protocol. Two operations make up subset B:

• ‘get-notify-schemes-supported’. (same as in *subset A*).
• ‘send-notification-with-language’. An IPP printer may use this operation to request that the notification delivery service generate a human-consumable representation of the notification data (if necessary) and use one of its supported delivery methods to dispatch a notification. The operation specifies the following information:
    a) The notification recipient’s url which the printer received as part the subscription.
    b) The data that comprises an “notification” as defined in the IPP Notification spec without a human-consumable representation.
    c) The ‘natural-language’ the notification delivery service should use when generating any

necessary human-consumable data.

Subset C operations

IPP printers that wish the notification delivery service to store subscriptions and generate and dispatch notifications may use this subset of the protocol. At least the following four operations comprise subset C:

- 'get-notify-schemes-supported'. (same as in *subset A*).
- 'create-subscription'. An IPP printer may use this operation to register a subscription (most likely as a result of receiving a request to create a subscription itself) on to the notification delivery service. The information specified in this operation parallels the data in the 'create-subscription' operation of the notification extension to IPP. This operation returns a subscription-id and a list of the new notifications the printer needs to generate in order to satisfy this new notification request.
- 'destroy-subscription'. An IPP printer may use this operation to remove a previously registered subscription. This operation returns a list of the notifications the printer no longer needs to generate.
- 'send-target-less-notification'. An IPP printer may use this operation to send a notification to the notification delivery service and request that it find all previously-registered subscriptions by this printer that specify interest in the event. The notification delivery service is then responsible for generating any necessary human-consumable representation and dispatching the necessary notifications using the delivery methods specified in the subscriptions.