[MC-NPR]:
.NET Packet Routing Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- Microsoft does not claim any trade secret rights in this documentation.
- Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. If you are interested in obtaining a patent license, please contact protocol@microsoft.com.
- The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary

Date	Revision History	Revision Class	Comments
08/10/2007	0.1	Major	Initial Availability
09/28/2007	0.2	Minor	Updated the technical content.
10/23/2007	0.3	Minor	Updated the technical content.
11/30/2007	0.3.1	Editorial	Revised and edited the technical content.
01/25/2008	0.3.2	Editorial	Revised and edited the technical content.

Copyright © 2008 Microsoft Corporation.

Table of Contents

Copyright © 2008 Microsoft Corporation.

1 Introduction

This document specifies the .NET Packet Routing Protocol, which defines a SOAP message header for indicating that a message can safely be treated as a packet or datagram.

In most common scenarios, a message **router** must assume that a relation to the transport connection exists and preserve that relation across inbound and outbound transport connections. Message-level packet routers can test for the presence of this header when the transport protocol is not inherently safe for use with **packet routing**.

Messages that are transmitted over a connection-oriented protocol often have some relation to the underlying transport connection. For example, there may be a dependency on the underlying transport connection to maintain message ordering, provide a back-channel for replies, or provide a consistent routing path. Other times, messages may have no such relation to the underlying transport connection and a router may be free to treat the message as an individual packet. Treating the message as a packet enables optimizations such as coalescing messages together or load balancing across multiple back-ends to improve reliability, scalability, and performance.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Endpoint

The following terms are specific to this document:

.NET Message: A full and well-formed SOAP message.

.NET Packet: A **.NET message** that includes routing-related SOAP headers, if any.

Node: An **endpoint** in the computer network that can receive, send, or process a SOAP message, as specified in [SOAP1-2.1].

PacketRoutable Element: The special XML element <PacketRoutable /> (XML namespace "http://schemas.microsoft.com/ws/2005/05/routing"), which must be inserted into the SOAP header so the message is considered a packet.

Packet Routing: The process of determining the next destination **node** for a **.NET packet** and sending that packet to it.

Router: An **endpoint** that receives messages from a **node** and forwards those messages to the next **node** based on a **routing algorithm**.

Routing Algorithm: An algorithm used to determine the next destination **node** for a specific packet. The algorithm MAY use a **routing table** or some other approach to determine a suitable next destination **node**.

Routing Table: A table used by a **.NET packet** router that determines the next destination for a specific message. The entry in the table can be static, or it can be dynamically populated.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

Copyright © 2008 Microsoft Corporation.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[SOAP1-2.1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J. J., Nielsen, H. F., Karmarkar, A. and Lafon, Y., " SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) ", W3C Recommendation 27, April 2007, http://www.w3.org/TR/soap12-part1

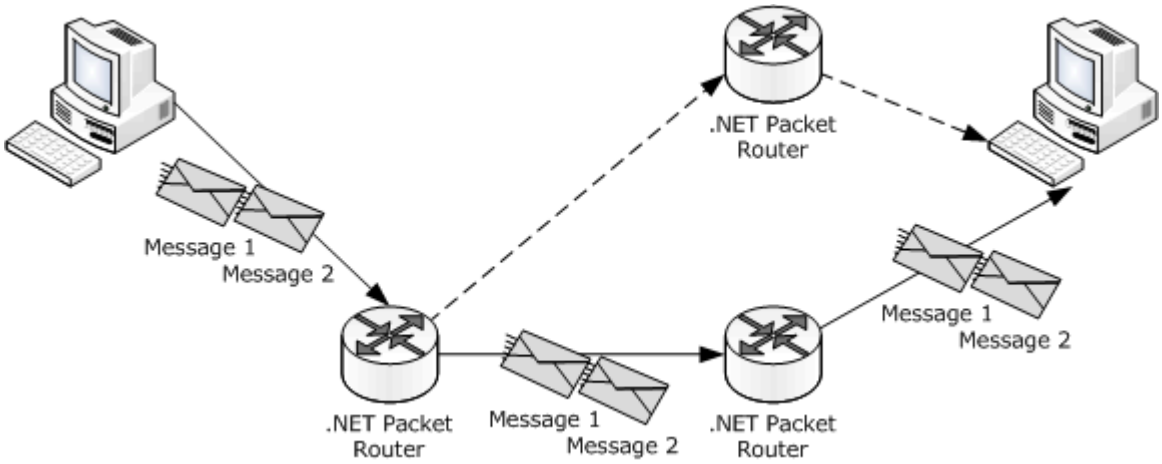
1.2.2 Informative References

No informative references are cited for this protocol.

1.3 Protocol Overview (Synopsis)

Packet routing is the predominant form of routing on the Internet. Packet routing does not preserve connection-oriented features such as routing paths, message-delivery order, or the presence of a back channel for replies. Circuit routing does preserve those features, but circuit routing requires more resources to route messages (see Figure 1). In addition, some application protocols do support treating application messages as datagram packets, while other application protocols do not. However, if a router receives an application message over a connection, the router must infer that the presence of the connection is significant and use circuit routing. However, existing circuit routing schemes are inflexible and do not provide the optimal scalability or efficiency that is provided with packet routing.

The .NET Packet Routing Protocol enables a **.NET message** originator to specify that a particular message-delivery requirement does not need to be retained, and that the message body need not be processed and understood by routers. If a router understands the XML element, and the message is carrying that XML element, packet routing should be used for delivering the message (see Figure 2).



Copyright © 2008 Microsoft Corporation.

Figure 1: Circuit routing of .NET messages

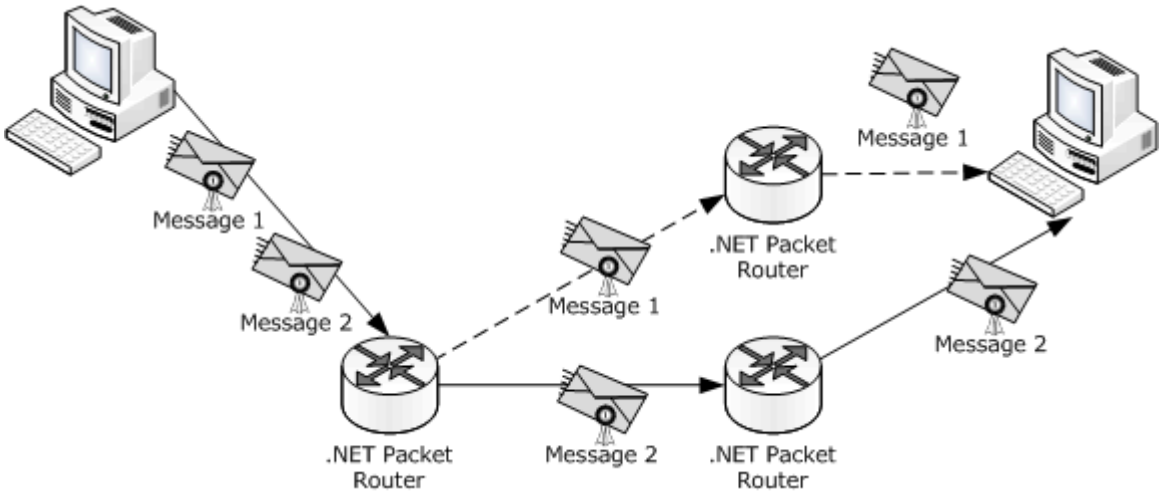


Figure 2: Packet routing of .NET packets

1.3.1 Message Type

For a **.NET message** to use packet routing, the message MUST be a well-formed and complete SOAP message, and should have the <PacketRoutable> XML element in the SOAP header, as specified in section 1.7.

1.4 Relationship to Other Protocols

The .NET Packet Routing Protocol supports only SOAP-formatted packets. The communication protocol between the sender and a router or between a router and the receiver MUST use a SOAP-supported transport protocol, such as TCP/IP or HTTP/S. Figure 3 shows the different layers in a router's stack. At the bottom of the stack is the actual physical communication medium.

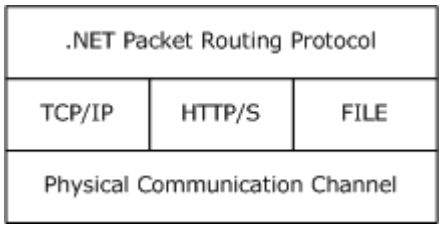


Figure 3: Dependency stack in a .NET packet router

1.5 Prerequisites/Preconditions

The .NET Packet Routing Protocol assumes the following:

1. The .NET Packet Routing Protocol is not dependent on any specific transport protocol.
2. The .NET packet is a self-contained, independent data entity that carries sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network.

Copyright © 2008 Microsoft Corporation.

3. Packet routers along the path from the sender to the receiver do not have any knowledge of the message delivery guarantees, such as ordered delivery or reliable messaging.

4. Communication between the sender and a router uses transport-protocol–specific acknowledgment to signal successful or failed delivery of a message. For example, in the case of HTTP, an HTTP status code 202 is sent back from the router if the message was successfully received. An HTTP status code 500 can be sent back when for any reason the router fails to receive the message. Communication between a router and the destination follows the same rules.

5. The .NET Packet Routing Protocol does not define any recovery mechanisms for communication failure scenarios as specified in the assumption above.

1.6 Applicability Statement

The .NET Packet Routing Protocol can be used when an application determines that circuit routing is consuming more resources than necessary. The application requests a downgrade to packet-routing by applying a special XML element to the message's SOAP header. For more information, see section 2.1.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- Protocol Versions: The .NET Packet Routing Protocol applies to SOAP messages that include the additional XML element <PacketRoutable /> with a namespace of "http://schemas.microsoft.com/ws/2005/05/routing".

  Protocol Versions: This protocol requires the use of SOAP messaging version 1.1, SOAP messaging 1.2, or later versions.

  Capability Negotiation: The .NET Packet Routing Protocol does not support negotiation of the version to use. Instead, an implementation must be configured to process only messages with the specific XML element and namespace that are described in this document.

The .NET Packet Routing Protocol applies to SOAP messages that are formatted based on the released SOAP versions 1.1, 1.2, or later versions. It is assumed that router-specific implementation is to handle other protocols that could hold the special XML element specified in this document.

Moreover, this document references valid, well-formed, and complete SOAP messages that carry the special XML element <PacketRoutable /> with the specific namespace "http://schemas.microsoft.com/ws/2005/05/routing".

An implementation is not compliant with this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined herein. A SOAP **node** MUST NOT use the "http://schemas.microsoft.com/ws/2005/05/routing" XML Namespace identifier within SOAP Envelopes unless it is compliant with this specification.

1.8 Vendor-Extensible Fields

The .NET Packet Routing Protocol does not specify any extensions or extensible fields, by default. However, vendors and implementers may choose to extend the protocol by including additional attributes or child XML elements for special handling of messages. Such extensions may include but are not limited to the following examples.

- Adding retry attributes in case of point-to-point delivery failures.

Copyright © 2008 Microsoft Corporation.

- Adding preferred routes information.
- Adding an attribute that specifies the maximum number of hops in a route and a counter to monitor the maximum number of hops already traversed.
- Supporting streamed versus buffered routing of messages.

It is required, however, that any extension or implementation must provide the basic and default behavior specified in this protocol document when the router does not understand a specific extension.

1.9 Standards Assignments

There are no particular standard assignments for this protocol at this time.

Copyright © 2008 Microsoft Corporation.

2 Messages

The following sections specify how .NET Packet Routing Protocol messages are transported and the common data types.

2.1 Transport

A .NET Packet Routing Protocol router MUST use a SOAP-supported transport protocol for receiving a .NET packet message. Furthermore, the router MUST use a SOAP transport protocol for forwarding the packet to the next destination. There are no restrictions on the use of any specific SOAP transport protocol.

A client that communicates with a .NET packet router MUST insert the special XML element <PacketRoutable /> (XML namespace "http://schemas.microsoft.com/ws/2005/05/routing"), also called the **PacketRoutable element** in this document, into the SOAP header so that it can be treated as a packet for routing.

When a packet is received by a .NET packet router, and the SOAP header includes the PacketRoutable XML element, the router MUST process the PacketRoutable XML element.

When sending the received message, if the PacketRoutable XML element exists in the received message, the .NET packet router MUST ensure that it also exists in the outgoing packet.

This document does not specify how to process any custom third-party extensions or attributes to this protocol when they are processed by a .NET packet router. For consistency and compliance with this protocol, a .NET Packet Routing Protocol router MUST ensure the transfer of the XML element to the outbound packet in its basic form, as specified in section 2.2.1.

2.2 Message Syntax

2.2.1 PacketRoutable Element Syntax

For a .NET message to be treated as a .NET packet, a special XML element MUST be placed within its SOAP header. The format of this XML element is as follows:

```
<PacketRoutable xmlns="http://schemas.microsoft.com/ws/2005/05/routing
">
</PacketRoutable>
    Name : PacketRoubale
    Namespace: http://schemas.microsoft.com/ws/2005/05/routing
```

There is no specific location for this XML element within the SOAP header.

In addition, the PacketRoutable element MAY carry the SOAP **mustUnderstand** attribute whose interpretation is specified in [SOAP1-2.1]. See Table 1 for details on how the **mustUnderstand** attribute MUST be treated by a router that can process the .NET Packet Routing Protocol, and by routers that cannot process the .NET Packet Routing Protocol.

	Value of mustUnderstand		
	TRUE	FALSE	Unspecified

Copyright © 2008 Microsoft Corporation.

	Value of mustUnderstand		
.NET packet routers	Treat Message as a Packet	Treat Message as a Packet	Treat Message as a Packet
Other routers	Generate a Fault as specified in [SOAP1-2.1] section 2.4	Ignore the XML element	Ignore the XML element

Copyright © 2008 Microsoft Corporation.

3 Protocol Details

The client side of the .NET Packet Routing Protocol MUST implement the insertion of the <PacketRoutable> XML element into the SOAP header of the message for a .NET Packet Routing Protocol router to treat it as a packet. Figure 4 represents the state transition of a SOAP message into a .NET packet.

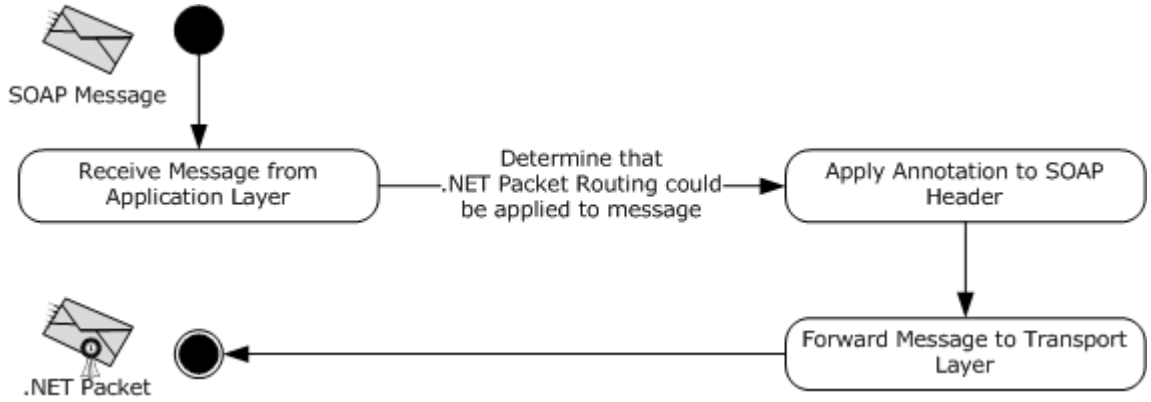


Figure 4: State diagram of client processing of a message

3.1 .NET Packet Router Server Details

3.1.1 Abstract Data Model

Every .NET Packet Routing Protocol router MUST implement processing to receive and send the <PacketRoutable> XML element.

When a packet is received by a .NET packet router, and it contains a SOAP header that includes the PacketRoutable XML element, the router MUST process the XML element and downgrade the routing to packet routing, applying any **routing algorithm** to determine the next destination of the packet. At this stage, an NPR router MAY also process any vendor-specific extension (such as hop count).

When receiving a message from a sending node to the NPR router, any failure should be handled based on the specific transport protocol used. For example, if HTTP was selected, an HTTP status code of 200 or 202 may signify that the message was successfully received by the router. An HTTP status code of 500 may signify that the message was not successfully received or processed. The specific handling of errors and .NET Faults is outside the scope of this document.

Copyright © 2008 Microsoft Corporation.

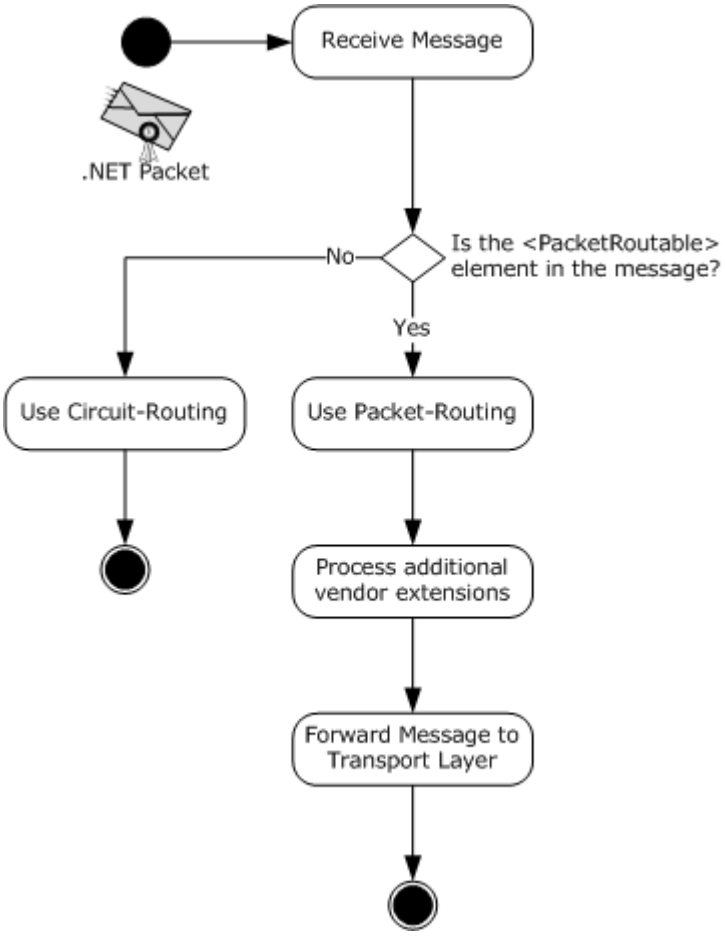


Figure 5: State diagram of the receiving end of a .NET Packet Routing Protocol router

When a .NET Packet Routing Protocol router sends a packet to a destination node, the router MUST maintain the PacketRoutable XML element in the outgoing packet, and MAY maintain other vendor-specific extensions. Any outbound transport MAY be chosen based on the available capabilities of either the sending or the receiving router. This protocol does not specify how to determine the appropriate transport protocol, or the selection process for the next destination node.

When a .NET Packet Routing Protocol router sends a packet to a destination node, any failure to deliver a message should be handled based on the specific transport protocol that is used. For example, if HTTP was selected, an HTTP status code of 200 or 202 may signify that the message was successfully received by the destination node. An HTTP status code of 500 may signify that the message was not successfully received or processed by the destination node.

Vendor extensions to the protocol MAY implement retry mechanism for resending a message upon failure. This protocol does not recommend any specific handling of such failures.

Copyright © 2008 Microsoft Corporation.

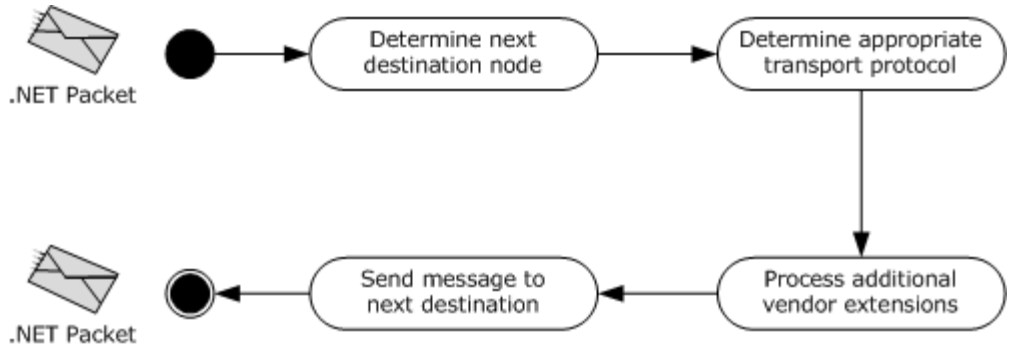


Figure 6: State diagram of the sending side of a .NET Packet Routing Protocol router

3.1.2 Timers

There are no timers associated with the .NET Packet Routing Protocol.

3.1.3 Initialization

There are no specific initializations associated with the .NET Packet Routing Protocol.

3.1.4 Higher-Layer Triggered Events

There are no higher-layer triggered events associated with the .NET Packet Routing Protocol.

3.1.5 Message Processing Events and Sequencing Rules

The .NET Packet Routing Protocol does not specify how or when an outbound connection should be opened when a router receives a packet. For example, an implementation of the .NET Packet Routing Protocol MAY require that a packet be fully received before an outbound connection is opened and the packet is forwarded. Another implementation MAY open an outbound connection and start forwarding the data in the packet as it is received on the inbound connection.

3.1.6 Timer Events

There are no timers associated with the .NET Packet Routing Protocol.

3.1.7 Other Local Events

There are no other local events associated with the .NET Packet Routing Protocol.

Copyright © 2008 Microsoft Corporation.

4 Protocol Examples

Messages may have no relation to the underlying transport connection. By inserting the PacketRoutable element into the SOAP header, packet routing would be used instead of circuit routing to deliver the message. Treating the message as a packet enables optimizations such as coalescing messages together or load balancing across multiple back ends to improve reliability, scalability, and performance. The following is an example of a SOAP message that contains the <PacketRoutable> XML element in the SOAP header.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
 xmlns:a="http://www.w3.org/2005/08/addressing">
    <s:Header>
        <PacketRoutable
             xmlns="http://schemas.microsoft.com/ws/2005/05/routing">
        </PacketRoutable>
        <a:Action s:mustUnderstand="1">
             http://tempuri.org/IService/MyOperation
        </a:Action>
        <ActivityId CorrelationId="592d7dfd-aafe-4df8-adfc-003db32efc7
c" xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostic
s">9df08e0a-a956-46d9-a859-d6571139ff0f
        </ActivityId>
        <a:To s:mustUnderstand="1">http://localhost:8080/service1
        </a:To>
    </s:Header>
    <s:Body>
        <MyOperation xmlns="http://tempuri.org/">
            <myValue>Some Value</myValue>
        </MyOperation>
    </s:Body>
</s:Envelope>
```


Copyright © 2008 Microsoft Corporation.

5 Security

The following sections specify security considerations for implementers of the .NET Packet Routing Protocol.

5.1 Security Considerations for Implementers

The .NET Packet Routing Protocol does not carry any security consideration. A vendor may extend the protocol to provide additional security considerations as long as the default and basic routing scenarios, as specified by this document, function as expected. In addition, vendors and implementers of this protocol must account for the fallback scenario when a message is routed through a .NET Packet Routing Protocol router that does not support the additional security extensions.

A vendor may extend the protocol to provide additional security considerations, as long as the default and basic routing scenarios specified by this document function as expected.

5.2 Index of Security Parameters

None.

Copyright © 2008 Microsoft Corporation.

6 Appendix A: Windows Behavior

The .NET Packet Routing Protocol has not been implemented in any Window version as of the release date of this document. Therefore, there are no specific Window behaviors for it.

Copyright © 2008 Microsoft Corporation.

7 Index

Copyright © 2008 Microsoft Corporation.