**Protocol Bridging in WCF**

**Table of Contents**

[Introduction 2](#_Toc464060156)

[Protocol Bridging 2](#_Toc464060157)

[Routing Configuration 3](#_Toc464060158)

[Summary 6](#_Toc464060159)

# 

# Introduction

Bridging => the action of putting a bridge over RIVER

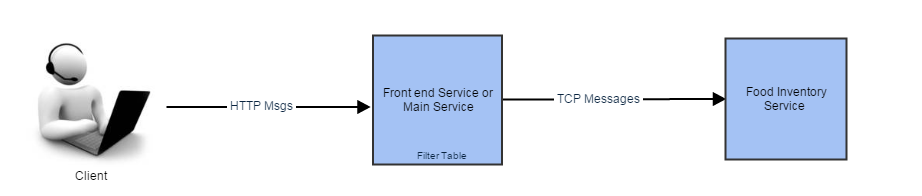
In the above sentence, you can find that bridge has been constructed over the river with the intention of traffic flow shouldn’t be interrupted at any cause. Yes, Pupils travelling from the road transport finds difficult to continue their journey as because of the River comes in their way. Bridge solves the interruption and provides a smooth flow of traffic to reach the other end. As like that, data communication shouldn’t be interrupted across the services as each service owns a different set of protocol standards like, one might use TCP communication, other might use HTTP communication. Here we see Protocol Bridging feature from WCF to route the communication across a different set of protocol services. Let’s discuss this in detail

# Protocol Bridging

Protocol Bridging is a one of the important features in WCF to route the data from the client to the

Services when both using a different set of protocols. (E.g. Client=>basicHTTBinding, WCF Service =>

netTcpBinding).



In the above snapshot, you can find that client sends messages in an HTTP format to the router

Service, but the internal services which are connected to the main router do not accept messages in a

HTTP format rather it accepts only in a TCP format. So, the router switches different binding

according to the internal service and sends the message. Client will get the response messages in the

HTTP format. It’s a complete responsibility of Router to read the data in a source protocol and switch

Into the destination protocol based on their need. It can also be useful like when you don’t want some

Complex services actual url shouldn’t be exposed to the client. In this case, you can provide only the

main router service url to the client, the main router service then redirects the messages to the actual

service based on the message filtering.

# Routing Configuration

As mentioned in the previous snapshot we shall configure two services, which are Router or Front end

Service and internal service. Internal service will accept messages in a TCP format, Router Service will

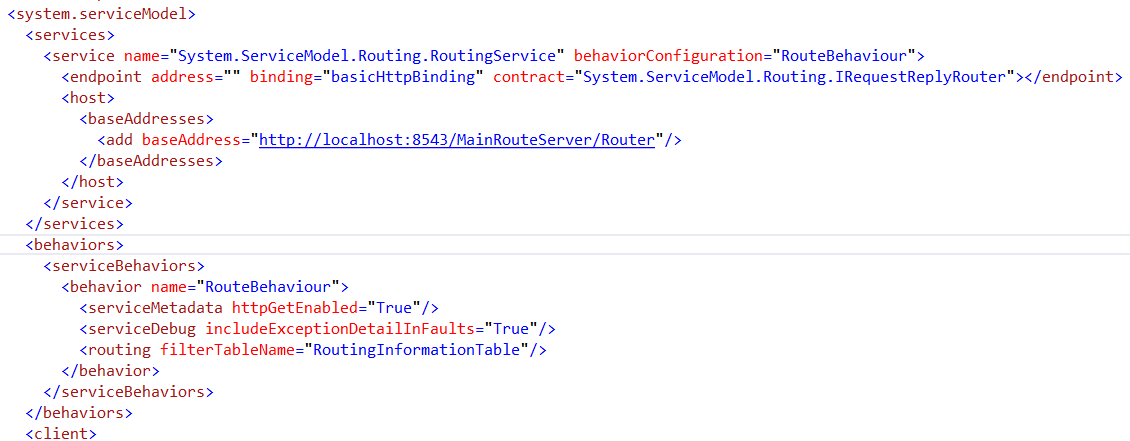
accept messages in an HTTP format.

Step 1: As a first step, we shall configure the internal service named FoodInventory Service which accepts messages in a TCP format. AS mentioned in the below snapshot you can configure the Service Library.



You can find that service Library created with the address “net.tcp://localhost:8523/FoodInventoryLib/Service” and the binding is netTCPBinding . Once the Service Library created, it can be hosted. In this case, we will do the Self-Hosting. If you need any assistance for Self-Hosting, Please refer my article “Hosting in WCF”.

Step 2: Once the internal service configured, we need to configure the Router Service or Front-end service. As mentioned in the below snapshot you can configure the Routing Service which routes the messages based on the message content, Filters. If you need any assistance on configuring Routing Service, please refer my article “Configuring WCF Routing”.



We have to remember the four important configurations in creating Router service configuration. I will suggest you remember the simple sentence to recollect the routing server configuration, which is “Address the Filter Table and Filter End”. Yes, **Address** = > you shouldn’t forget to update the Server address, in our case it is <http://localhost:8543/MainRouteServer/Router>, a client will use this address to send all the messages. Next one is **Filter Table** => once you define the address; don’t forget to declare the FilterTableName in Service Behaviour Tag, and then creating the same Filtertable in Routing Tag. In our case it is “RoutingInformationTable” Next one is **Filter** =>Create and add the Filter in the filtertable.In our case it is “MatchAllFilter”, Final one is **End** => you need to create the endpoint and associate the same in the filtertable. The above configuration says that if the message filter is applied successfully you can redirect the message to the endpoint listed. In our case, if any incoming message contains the endpoint address as [http://localhost:8543/MainRouteServer/Router,](http://localhost:8543/MainRouteServer/Router,the) the message will be redirected to the service net.tcp://localhost:8523/FoodInventoryLib/Service. Wherever the service you want to route, just simply update the name of the binding with the service URL. In our case, you can find that endpoint has been created under the client tag with the address and binding details. WCF internally switches (bridges) the protocol into TCP format and sends the messages to the internal service. Once the internal service responded to the Main router service it again switches (bridges) the protocol into HTTP format before sending to the client.

Hope you have enjoyed the learning; please update the comments/likes as a token of appreciation.

# Summary

Protocol bridging is an important feature in WCF routing. It will be useful when you need the data to be travelled among the various set of protocol services.