Chapter 6

Proposed Nearest PIM-SM Extension

Researches on anycast routing to date (Katabi, D., 1999; Park, Vincent D. and Macker, Joseph P., 1999; Park, Vicent D. and Macker, Joseph P., 1999b; Wei Jia et al., 1999; Wei Jia et al., 2000a; Wei Jia et al., 2000b) did not involve the PIM-SM multicast protocol, although it is widely implemented in the current internetworking world. Rather, most of the discussions are about the CBT, a rarely used model.

PIM-SM is chosen as the multicast routing protocol for this thesis because the problem of not having such a good network topology will not be so severe for PIM-SM. The reason behind this is that PIM-SM does not send any multicast flows to network areas where there are no group members. Besides, PIM-SM can scale well and is more reliable than normal CBTs with the uses of Candidate-RP. However, some modifications on the PIM-SM are needed for the use of anycast routing.

6.1 RIPng Extension

RIPng extension in this thesis is built based on the idea proposed by Park, Vicent D. and Macker, Joseph P. (1999b). Anycast service is treated as a virtual node and each router will have a route entry to that virtual node. When a request for the anycast service is made, the routers will route the packets towards the nearest virtual (see Figure 6.1).

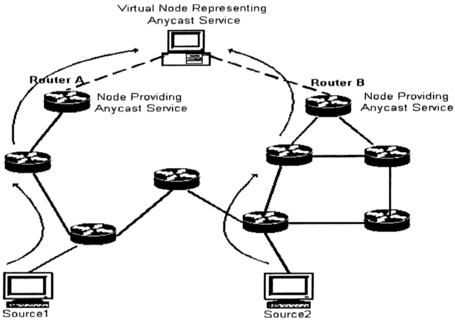

Figure 6.1 *A scenario of RIPng extension for anycast routing*

In Figure 6.1, the anycast packets sent from Source1 will be routed through Router A and the anycast packets sent from Source2 will eventually be routed through Router B. The anycast packets sent from Source2 will not be routed through Router A as it is further than the route through Router B and vice versa for Source1.

6.2 PIM-SM Extension

According to Dong Xuan et al. (2000), multi-path approach for anycast routing can load balance the traffic better than single-path routing in the case of heavy traffic (refer to Section 5.3 for issues of single-path and multi-path routing). However, the results may not be totally agreeable. The multicast routing protocol used by Dong Xuan et al. (2000) is CBTs, a model that may have problems when not having a good network topology. This thesis later found that the performance of the anycast routing is very much dependent of the network topology, the senders and the receivers as well.

PIM-SM is chosen for several reasons:

1) Problems of not having such a good network topology will not be so severe for PIM-SM
2) More reliable with the use of C-RP
3) More scalable
4) A lot of implementations in the real internetworking world
5) It is loop-free

Some modifications to the PIM-SM are needed for the anycast routing. For example, the switching to the shortest-path tree method is omitted in this thesis. This is based on the assumption that the communication between the sender and the receiver will be in unicast after a connection is established. The continuous flow of the traffic between the sender and the receiver is beyond the scope of this thesis. The arguments of how the communication of the sender and the receiver transform to unicast will not be discussed in this thesis.

The terminology used by the PIM-SM extension is as follows:

When receiving a unicast/anycast packet

If the receiving router has group information of the anycast group

Forward the packet towards the node that joins the anycast group, according to the load-balancing scheme applied

Else

Forward the packet towards the Rendezvous Point (RP) for that anycast group

The problem for PIM-SM extension, as well as other multicast routing protocol is that it will not work well when there are many off-tree hits.

On-tree Hits

On-tree hits means that the router that the sender is directly attached to has joined the multicast tree, thus the packets can be routed to the receivers directly.

Off-tree Hits

Alternatively, off-tree hits mean that the router that the sender is directly attached to has not joined the multicast tree and has no information of the receivers, so the router will have to route the packets towards the RP for the group.

6.3 Proposed Nearest PIM-SM Extension

Using multicast routing for anycast will only perform better than shortest-path routing under heavy load conditions and when there are many on-tree hits (refer to previous section). Under other conditions, it will not perform better than shortest-path routing. Beside, by sending all the packets towards the RP, the RP might be overloaded and fail to respond properly. Although PIM-SM does provide failure handling through C-RP, it is more desirable to avoid it from happening in the first place. Based on these observations, this thesis proposes to use shortest-path routing for off-tree hits, rather than sending all the packets towards the RP. To accomplish this, this thesis proposed a new method, nearest PIM-SM Extension, a modification to the PIM-SM extension discussed earlier.

The objective of the proposed nearest PIM-SM extension is to obtain equivalent performance of using shortest-path tree while inherits the load-balancing abilities of the PIM-SM. The proposed protocol aims to reduce the effects of hot spot and traffic concentration around the RP. It will not only improve the performance of the anycast routing, but also its reliability.

The terminology used by the proposed nearest PIM-SM extension is as follows:

When receiving a unicast/anycast packet

If the receiving router has group information of the anycast group

Forward the packet to the nodes that join the anycast group, according to the load-balancing scheme applied

Else

Forward the packet to the nearest interface having the anycast group address, like a normal unicast route table lookup

The proposed anycast routing protocol follows the IPv6 addressing scheme by Hinden, R. and Deering, S. (1998) that an anycast address space is allocated from a unicast address space and thus they are indistinguishable from each other.

6.4 Load-balancing Scheme

Shortest-path

Shortest-path is a typical single-path routing approach. The received anycast packets will be routed towards the nearest downstream router in the anycast group, according to the routing protocol's measure of distance.

Shortest-path is the simplest approach and is easy to implement. However, it has some weak points. It cannot make use of the existence of an alternative route even though the shortest-path is very congested at the moment. The packets will still be sent towards the shortest-path and high packets loss will be experienced. This drawback greatly reduces the reliability of the anycast routing using this scheme.

Round Robin

Round robin is a typical multi-path routing approach. Each interface that has anycast members in the downstream will be selected accordingly in turns.

The benefit of using round robin is that it can distribute the traffic load evenly among all the available routes. This behavior can greatly improve the performance of the anycast routing when the traffic load is extremely high. However, this scheme has a setback. The action of distributing the traffic load evenly to all the available routes degrades the performance of the anycast when the traffic load is either low or moderate.

Fuzzy Shortest-path

In this scheme, the shortest-path will be fuzzified. Fuzzification is implemented to minimize the need to perform complex calculations and decision-making by introducing some linguistic rules and membership functions. Fuzzy logic will help in deciding whether to use that route or choose an alternative route. Fuzzy rule base is used here.

The terms used to describe the remaining queue length percentage (compare to full-size queue length) are "very low", "low", "moderate", "high" and "very high". The terms used to describe the queue-length change percentage are "decreasing fast", "decreasing", "zero", "increasing" and "increasing fast". The terms for output congestion level (y) are "very low", "low", "moderate", "high" and "very high".

$T(q)$ = { very low(qVL), low (qL), moderate (qM), high (qH), very high (qVH) }

$T(\Delta q)$ = { decreasing fast (dqDF), decreasing (dqD), zero (dqZ), increasing (dqI), increasing fast (dqIF) }

$T(y)$ = { very low (congVL), low (congL), moderate (congM), high (congH), very high (congVH) }

The membership function for the fuzzy set (very low (qVL), low (qL), moderate (qM), high (qH), very high (qVH) and etc.) listed above should be defined with proper shape and position. Usually a triangular function $f(x{:}x_0, a_0, a_1)$ or a trapezoidal function $g(x{:}x_0, x1, a_0, a_1)$ is chosen as the membership function because these functions are suitable for real-time operation. As shown in Figure 6.2, $f(x{:}x_0, a_0, a_1)$ and $g(x{:}x_0, x1, a_0, a_1)$ are given by

$$f\left(x:x_0,a_0,a_1\right)=\begin{cases}\dfrac{x-x_0}{a_0}+1 & for\ x_0-a_0<x<x_0\\ \dfrac{x-x_0}{a_1}+1 & for\ x_0<x<x_0+a_1\\ 0 & otherwise\\ \dfrac{x-x_0}{a_0}+1 & for\ x_0-a_0<x\leq x_0\end{cases}$$

$$f\left(x:x_0,x_1,a_0,a_1\right)=\begin{cases}1 & for\, x_0<x\leq x_1\\ \dfrac{x_1-x}{a_1}+1 & for\, x_1<x<x_1+a_1\\ 0 & otherwise\end{cases}$$

where x_0 in $f(.)$ is the center of triangular function; x_0 (x_1) in g(.) the left (right) edge of the trapezoidal function; and a_0 (a_1) is the left (right) width of the triangular or the trapezoidal function.

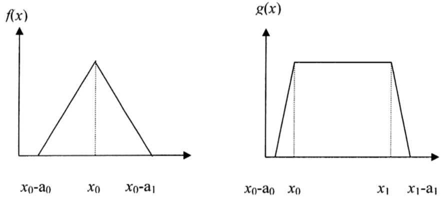

Figure 6.2 *Definitions for functions f(.) and g(.)*

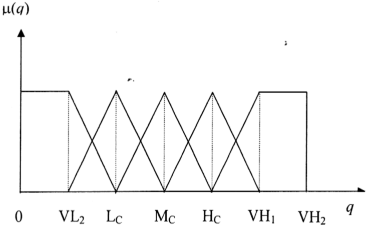

Figure 6.3 *The membership function of the term set T(q)*

Let μqVL(q), μqL(q), μqM(q), μqH(q) and μqVH(q) denote the membership function for qVL, qL, qM, qH and qVH in T(q), respectively. Let μqVL(q), μqL(q), μqM(q), μqH(q) and μqVH(q) be

$$\mu_{qVL}(q) = g(q;0,VL_2,0,VL_w)$$
$$\mu_{qL}(q) = g(q;0,L_1,L_2,L_w,L_w)$$
$$\mu_{qM}(q) = g(q;0,M_1,M_2,M_w,M_w)$$
$$\mu_{qH}(q) = g(q;0,H_1,H_2,H_w,H_w)$$
$$\mu_{qVH}(q) = g(q;VH_1,VH_2,VH_w,0)$$

Similarly, let μDF(Δq) and μD(Δq), μZ(Δq), μI(Δq) and μIF(Δq) denote the membership functions for the terms DF, DS, ZR, IS and IF in T(Δq), respectively, and define μDF(Δq) and μDS(Δq), μZ(Δq), μIS(Δq) and μIF(Δq) as

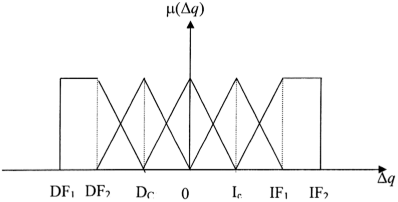

Figure 6.4 *The membership function of the term set T(Δq)*

As shown in Figure 6.4, μDF(Δq) and μIF(Δq) are trapezoidal function. However, μD(Δq), μZ(Δq) and μI(Δq) are respectively triangular function.

The membership functions associated with the terms VL, L, M, H, VH in T(y) are denoted by μVL(y), μL(y), μM(y), μH(y) and μVH(y), respectively, which are given by

$\mu_{VL}(y) = f(y;\ VL_C, 0, 0)$

$\mu_L(y) = f(y; L_C, 0, 0)$

$\mu_M(y) = f(y; M_C, 0, 0)$

$\mu_H(y) = f(y; H_C, 0, 0)$

$\mu_{VH}(y) = f(y; VH_C, 0, 0)$

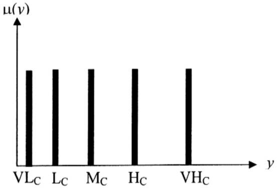

Figure 6.5 *The membership function of the term set T(y)*

Fuzzy Rule Base

The design of a rule base is two-fold: firstly, the linguistic rules ("surface structure") are set; secondly, membership functions of the linguistic values ("deep structure") are determined. The tradeoff involving the design of the rule base is to have a set of minimum number of linguistic rules representing the control surface with sufficient accuracy to achieve an acceptable performance. Recently, in the fuzzy control literature, some formal techniques for obtaining a rule base by using artificial neural networks or genetic algorithms have appeared.

/* linguistic rules */

if q is very low and dq is increasing fast then congestion level is very high; /* r1 */

if q is very low and dq is increasing then congestion level is very high; /* r2 */

if q is very low and dq is zero then congestion level is very high; /* r3 */

if q is very low and dq is decreasing then congestion level is high; /* r4 */

if q is very low and dq is decreasing fast then congestion level is high; /* r5 */

if q is low and dq is increasing fast then congestion level is very high; /* r6 */

if q is low and dq is increasing then congestion level is very high; /* r7 */

if q is low and dq is zero then congestion level is high; /* r8 */

if q is low and dq is decreasing then congestion level is high; /* r9 */

if q is low and dq is decreasing fast then congestion level is moderate; /* r10 */

if q is moderate and dq is increasing fast then congestion level is high; /* r11 */

if q is moderate and dq is increasing then congestion level is high; /* r12 */

if q is moderate and dq is zero then congestion level is moderate; /* r13 */

if q is moderate and dq is decreasing then congestion level is moderate; /* r14 */

if q is moderate and dq is decreasing fast then congestion level is low; /* r15 */

if q is high and dq is increasing fast then congestion level is moderate; /* r16 */

if q is high and dq is increasing then congestion level is moderate; /* r17 */

if q is high and dq is zero then congestion level is low; /* r18 */

if q is high and dq is decreasing then congestion level is low; /* r19 */

if q is high and dq is decreasing fast then congestion level is very low; /* r20 */

if q is very high and dq is increasing fast then congestion level is low; /* r21 */

if q is very high and dq is increasing then congestion level is low; /* r22 */

if q is very high and dq is zero fast then congestion level is very low; /* r23 */

if q is very high and dq is decreasing then congestion level is very low; /* r24 */

if q is very high and dq is decreasing fast then congestion level is very low; /* r25 */

The fuzzy shortest-path load-balancing algorithm adopts the max-min inference method for the inference engine because it is designed for real-time operation. Figure 6.6 shows an example of the max-min inference method for the fuzzy shortest-path algorithm, where rules 4 and 7, which have the same control action, "y is very high (VH)", are depicted.

/* linguistic rules */

if q is very low and dq is increasing fast then congestion level is very high; /* r1 */
if q is very low and dq is increasing then congestion level is very high; /* r2 */
if q is very low and dq is zero then congestion level is very high; /* r3 */
if q is very low and dq is decreasing then congestion level is high; /* r4 */
if q is very low and dq is decreasing fast then congestion level is high; /* r5 */
if q is low and dq is increasing fast then congestion level is very high; /* r6 */
if q is low and dq is increasing then congestion level is very high; /* r7 */
if q is low and dq is zero then congestion level is high; /* r8 */
if q is low and dq is decreasing then congestion level is high; /* r9 */
if q is low and dq is decreasing fast then congestion level is moderate; /* r10 */
if q is moderate and dq is increasing fast then congestion level is high; /* r11 */
if q is moderate and dq is increasing then congestion level is high; /* r12 */
if q is moderate and dq is zero then congestion level is moderate; /* r13 */
if q is moderate and dq is decreasing then congestion level is moderate; /* r14 */
if q is moderate and dq is decreasing fast then congestion level is low; /* r15 */
if q is high and dq is increasing fast then congestion level is moderate; /* r16 */
if q is high and dq is increasing then congestion level is moderate; /* r17 */
if q is high and dq is zero then congestion level is low; /* r18 */
if q is high and dq is decreasing then congestion level is low; /* r19 */
if q is high and dq is decreasing fast then congestion level is very low; /* r20 */
if q is very high and dq is increasing fast then congestion level is low; /* r21 */
if q is very high and dq is increasing then congestion level is low; /* r22 */
if q is very high and dq is zero fast then congestion level is very low; /* r23 */
if q is very high and dq is decreasing then congestion level is very low; /* r24 */
if q is very high and dq is decreasing fast then congestion level is very low; /* r25 */

The fuzzy shortest-path load-balancing algorithm adopts the max-min inference method for the inference engine because it is designed for real-time operation. Figure 6.6 shows an example of the max-min inference method for the fuzzy shortest-path algorithm, where rules 4 and 7, which have the same control action, "y is very high (VH)", are depicted.

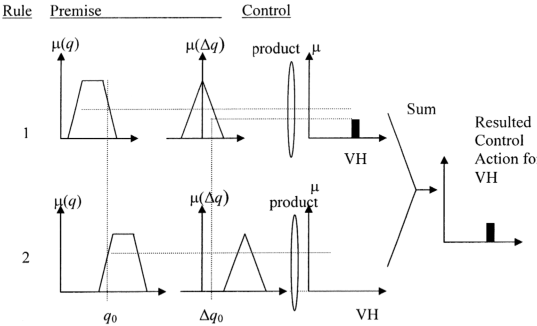

Figure 6.6 *The product-sum inference method*

In Figure 6.6, the input for q and Δq is q_0 and Δq_0. The membership values of q_0 and Δq_0 corresponding to the premise of rule 1 is q is moderate and Δq is zero are given by $\mu_{qVL}(q_0)$ and $\mu_{IF}(\Delta q_0)$, respectively. Applying the product operator, the membership value of the control action y=VH or rule 4 can be obtained, by

$$w_1 = product\lfloor \mu_q(q_0), \mu_{IF}(\Delta q_0) \rfloor$$

$$\mu_{VH}(y = VH_C)$$

where w_1 is membership value of the control action y=VH of rule 1 and VH_C is the center of function $\mu_{VH}(y)$. The membership values of rules 1 and 2 are denoted by w_1 and w_2, respectively, can be obtained in the same manner. Subsequently, applying the sum operator yields the overall membership value of the control action y=VH, denoted by w_{VH} as follows:

$$w_{VH} = \text{sum}(w_1, w_2)$$

The overall membership values of the control actions VL, L, M, H and VH denoted by w_{VL}, w_{L}, w_{M}, w_{H} and w_{VH} respectively, can be calculate in a similar way.

Tsukamoto's defuzzification method for the defuzzifier is used because of its simplicity in computation. This defuzzification method obtains a crisp value y_0 of the control action by combining w_{VL}, w_{L}, w_{M}, w_{H} and w_{VH}, shown below

$$y_0 = \frac{VL_C.w_{VL} + L_C.wL + M_C.w_M + H_C.wH + VH_C.w_{VH}}{w_{VL} + w_L + w_M + wH + w_{VH}}$$