昆明学院学报　2010,32(3):76~78　CN 53-1211/G4　ISSN 1674-5639
Journal of Kunming University

基于 NS-2 的距离向量选路算法仿真

钱开国
(昆明学院 物理科学与技术系,云南 昆明 650031)

摘要:路由协议是计算机网络组网的关键技术和教学难点,距离向量的选路算法对计算机网络初学者来说更加抽象. 通过 NSG 脚本生成工具可以方便快捷的建立 NS-2 网络模拟器的仿真场景来仿真距离向量选路算法,运行时可直接观察到实际计算机网络节点动态交换路由信息、建立路由表和数据传送的过程,同时 trace 文件记录了仿真过程中节点交换的信息包及其传送时间.

关键词:路由表;距离向量;选路算法;仿真
中图分类号:TP393;TP391.9　**文献标识码**:A　**文章编号**:1674-5639(2010)03-0076-03

The Simulation for Distance Vector Routing Algorithm Based on NS-2

QIAN Kai-guo
(Department of Physics Science and Technology, Kunming University, Yunnan Kunming 650031, China)

Abstract: The routing algorithm is a key technology and teaching difficult points for computer networks. The research of Distance Vector Routing Algorithm is abstract for beginners. The Distance Vector Routing Algorithm, one of the routing algorithms, is introduced, designed and implemented a simulation scene for DV algorithm through NSG on NS-2 networks simulator, and then directly viewed the routing process for Distance Vector Routing Algorithm, Meanwhile, the trace file will record exchanging packets and sending time.
Key words: routing table; distance vector; routing algorithm; simulation

计算机网络中,路由(Routing)算法就是用来决定将 IP 封包从源主机节点传送到目的主机节点的最佳路径(Route)的方法. 在进行数据传输之前,路由器之间要通过一定的信息交换建立路由表,路由便根据路由表来进行. 路由算法按不同的标准有全局选路算法、分散式选路算法、静态路由算法和动态路由算法. 在动态链路算法中,比较常用的是距离向量算法(distance vector, DV). 本文在分析了距离向量算法的基础上,设计了在 NS-2 下的仿真和实现.

1　距离向量路由算法

距离向量路由算法[1]是个一迭代、异步和分布式的算法. 每个节点都要从一个或者多个直接相连的邻居接收某些信息,执行计算,然后将计算结果发回给邻居,这个过程一直要持续到邻居节点之间没有更多的信息要交换,节点之间不要求同步进行操作.

1.1 基本思想

网络抽象为赋权图[2] $G=(V,E:w(e))$,其中 V 是顶点集合,代表计算机网络中路由器,计算机等设备. E 是边的结合,代表计算机网络中设备之间的链路. $w(e)$ 表示链路上的通信代价,实际模型如图 1 所示.

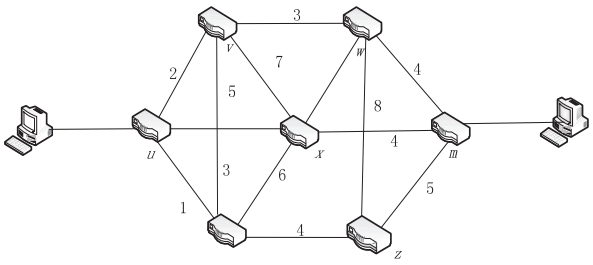

图1　计算机网络抽象模型

令 $d_x(y)$ 为从节点 x 到节点 y 的最低通信代价,则根据 Bellman-Ford 方程:

$$d_x(y)=\min\{w(x,y)+d_v(y)\}\ , \qquad (1)$$

可以找到节点 x 到节点 y 的最低通信代价的链路,以此为依据每个节点通过交换信息建立自己的路由表. 基本思想如下:

每个节点 x 以 $\boldsymbol{D}_x(y)$ 开始对在 G 中的所有节点,估计从它到节点 y 的最低链路通信代价. $\boldsymbol{D}_x=\boldsymbol{D}_x(y)$ 是节点 x 的距离向量,该向量是从 x 到 y 的链路通信代价的估计向量. 而每个节点 x 维护下例数据:

1)对于每个邻居节点,从 x 到直接相连邻居 y 的通信代价 $w(x,y)$;

2)节点 x 的距离向量,即 $\boldsymbol{D}_x=\boldsymbol{D}_x(y)$,包括了 x 到 G 中所有节点的链路通信代价的估计值;

3)它的每个邻居的距离向量,即 x 的每个邻居

收稿日期:2009-11-30
作者简介:钱开国(1979—),男,云南丽江人,讲师,硕士研究生,主要从事无线传感器网络、计算机网络研究.

$\boldsymbol{D}v = \boldsymbol{D}v(y)$.

1.2 路由算法

在这个算法中,每个节点不时地向它的每个邻居发送它的距离向量拷贝. 当节点 x 从它的任何邻居接收到一个新的距离向量,它保存 v 的距离向量,使用公式(1)更新自己的距离向量,如果节点 x 的距离向量发生了改变,节点 x 将向它的每个邻居发送它更新的距离向量. 所以节点以异步的方式工作,最后收敛于 $Dx(y)$. 算法如下:

初始化

G 中的每个目标节点 y

$D_x(y) = c(x,y)/*$ 如果 x 的直接邻居,则 $D_x(y)$ 为∞

每个邻居 w

$D_w(y) = \infty$

每个邻居 w

发送距离向量 $D_x = [d_x(y): y \text{ in } g]$ 到 w

循环

G 中的每个 y:

$D_x(y) = \min_v\{c(x,y) + D_v(y)\}$

如 $D_x(y)$ 改变

发送距离向量 $D_x = [D_x(y): y \text{ 在 } G]$ 到所有邻居.

2　仿真设计和实现

2.1 仿真场景

设计如图 2 所示的仿真场景,假设每个节点之间的通信链路代价都是 1,节点 n0 将数据传送到 n4,n0 到 n4 之间有 3 条链路. 链路 1:n0-n5-n6-n4 链路代价是 3;链路 2:n0-n1-n2-n3-n4 链路代价是 4;链路 3:n0-n7-n8-n9-n10-n4 链路代价是 5.

仿真时间为 7 s. 2 s 时,n0-n5 链路发生故障,即链路 1 断开;3 s 时,n0-n1 链路发生故障,即链路 2 断开;4 s 时,n0-n7 发生故障,链路 3 断开;5 s 时,n0-n5 链路恢复,即链路 1 恢复;6 s 时,n0-n1 链路恢复,即链路 2 恢复. 通过 NS-2 的 NAM 界面[3]观测路由表的建立和数据包的路由过程.

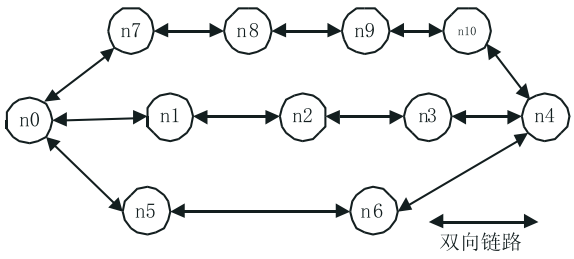

图2　网络仿真场景

2.2 仿真程序设计

仿真程序可以用 NSG[4]脚本生成器生成,也可以用文本工具执行编写. 过程:建立仿真对象,建立 6 个仿真节点,设定传输层和应用层代理以及链路间的连接,设定链路控制脚本,开始仿真运行.

```
#建立仿真对象
set ns [new Simulator]
#设定距离向量路由
$ns rtproto DV
#打开 trace 文件和 nam 文件
set tracefile [open out.tr w]
$ns trace-all $tracefile
set namfile [open out.nam w]
$ns namtrace-all $namfile
#建立 11 个仿真节点
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
set n9 [$ns node]
set n10 [$ns node]
#设定节点之间的链路
$ns duplex-link $n0 $n5 1.0Mb 10ms DropTail
$ns duplex-link $n5 $n6 1.0Mb 10ms DropTail
$ns duplex-link $n6 $n4 1.0Mb 10ms DropTail
$ns duplex-link $n0 $n1 1.0Mb 10ms DropTail
$ns duplex-link $n1 $n2 1.0Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.0Mb 10ms DropTail
$ns duplex-link $n3 $n4 1.0Mb 10ms DropTail
$ns duplex-link $n7 $n0 1.0Mb 10ms DropTail
$ns duplex-link $n7 $n8 1.0Mb 10ms DropTail
$ns duplex-link $n9 $n8 1.0Mb 10ms DropTail
$ns duplex-link $n10 $n9 1.0Mb 10ms DropTail
$ns duplex-link $n10 $n4 1.0Mb 10ms DropTail
#NAM 中的节点位置设定
$ns duplex-link-op $n0 $n5 orient right-down
$ns duplex-link-op $n5 $n6 orient right
$ns duplex-link-op $n6 $n4 orient right-up
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient right
$ns duplex-link-op $n7 $n0 orient left-down
$ns duplex-link-op $n7 $n8 orient right
$ns duplex-link-op $n9 $n8 orient left
$ns duplex-link-op $n10 $n9 orient left
$ns duplex-link-op $n10 $n4 orient right-down
#建立 TCP 连接和 FTP 业务绑定
set tcp0 [new Agent/TCP]
```

```
$tcp0 set fid_ 1
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 0.1 "$ftp0 start"
$ns at 7.0 "$ftp0 stop"
#设定链路故障和恢复控制
$ns rtmodel-at 2.0 down $n0 $n5
$ns rtmodel-at 3.0 down $n0 $n1
$ns rtmodel-at 4.0 down $n0 $n7
$ns rtmodel-at 5.0 up $n0 $n5
$ns rtmodel-at 6.0 up $n0 $n1
#定义结束过程
proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam out.nam &
exit 0
}
$ns at $val(stop) "$ns nam-end-wireless
$val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

3 仿真运行

1)开始时,大概 0.1 s 左右,节点与节点之间先交换路由信息(图中小黑点),建立路由表,如图 3 所示;

2)路由表建立后,传送数据从 n0 到 n4 的路径是链路 1:n0-n5-n6-n4,如图 4 所示;

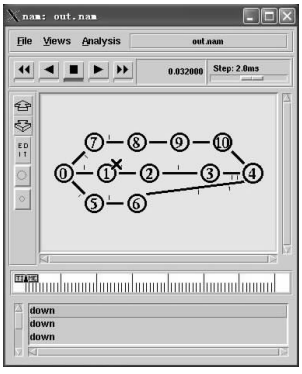

图3 节点交换信息

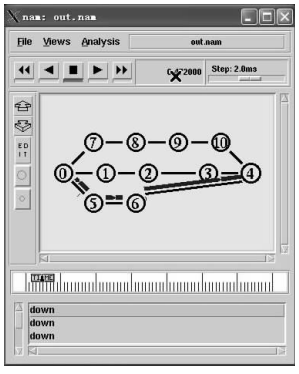

图4 约0.5 s时的数据包传送链路

3)在 2.0 s 时,根据程序控制 n1-n5 的链路发生故障,链路 1 断开. 节点之间又要交换信息重新建立路由表,如图 5 所示;

4)在 2.0 ~ 3.0 s 时,链路 1 发生故障,数据包的传输链路切换成链路 2:n0-n1-n2-n3-n4,如图 6 所示;

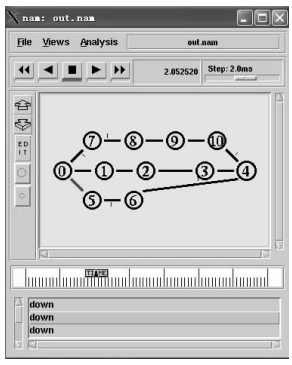

图5 第2 s时n0–n5链路故障

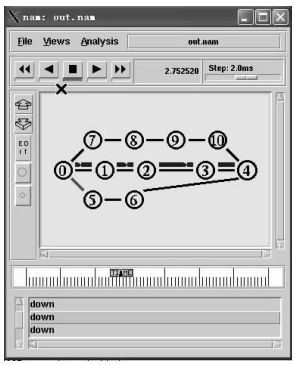

图6 约2.7 s时数据包的传输链路

5)时间在 3.0 s 时,n0-n1 的链路发生故障,节点之间又要交换信息重新建立路由表;

6)时间在 3.0 ~ 4.0 s 内,n0 带 n4 的通信链路切换成链路 3:n0-n7-n8-n9-n10-n4;

7)时间在 4 ~ 5 s 内,链路全部断开;

8)第 5 s 时,n0-n5 链路故障恢复,链路 1 通,交换信息重新建立路由表;

9)第 6 s 时,n0-n1 链路故障恢复,链路 2 通,交换信息重新建立路由表. 时间在 6 s 后,数据包传送沿着链路 1,n0-n5-n6-n4 传送.

4 结语

计算机网络路由协议[5]非常抽象,构建物理网络进行演示困难重重,而且看不到数据包收发的效果. 通过以上的仿真运行,采用网络仿真软件 NS-2 配合教学,可以清楚的呈现动态距离向量选路算法的工作过程,使学生能够直观的看到整个路由表建立和数据包收发过程,并且可以进一步分析 trace 文件深入理解整个网络协议,同时对于网络路由协议的研究也具有参考意义.

[参考文献]

[1]JAMES F,KUROSE,KEITH W,et al. 计算机网络[M]. 陈鸣,译. 北京:机械工业出版社,2007:233-245.

[2]张先迪,李正良. 图论及其应用[M]. 北京:高等教育出版社,2005.

[3]KEVIN F. The ns Manual[EB/OL]. [2009-01-02]. http://www.isi.edu/nsnam/ns/ns-documentation.htm.

[4]柯志亨,程荣祥,邓德隽. 仿真实验[M]. 北京:电子工业出版社,2009:82.

[5]申浩如,王付艳,邱莎. 基于 Wireshark 的 PPPoE 通信过程研究[J]. 昆明学院学报,2009,31(6):73-76.