فصل سوم

شبکه‌های پویا

۳-۱ مقدمه

تاکنون در خصوص موارد استفاده ns-2 و همچنین مفاهیم پرکاربرد و اساسی آن و همچنین دستورات مربوط به آن مطالبی بیان نمودیم.

در این بخش درباره شبکه‌های پویا و مسیریابی در صورت خرابی گره‌ها کار خواهیم کرد. در شبکه‌های بزرگ و واقعی تعداد گره‌ها بالا است و امکان خرابی نیز وجود دارد.

۳-۲ ایجاد یک توپولوژی بزرگ

ما نام این مثال خود را example3.tcl خواهیم گذاشت. همیشه توپولوژی در ابتدا ایجاد خواهد شد و برای اینکه توپولوژی‌های بزرگ را به سادگی بسازیم با یک راهکار این کار را انجام خواهیم داد. کد زیر هفت گره را ایجاد کرده و آنها را در آرایه ()n ذخیره می‌کند.

```
for {set i 0} {$i < 7} {incr i} {
set n($i) [$ns node]
}
```

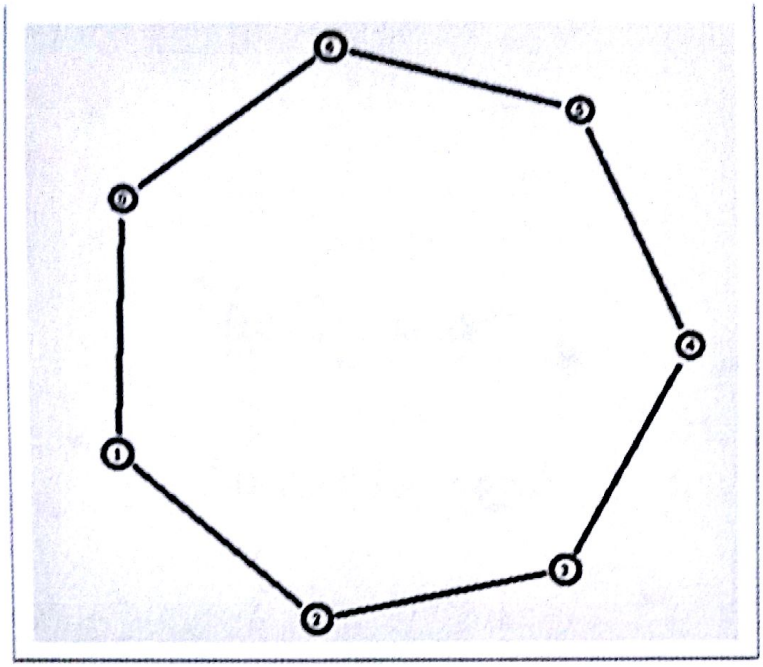
توجه داشته باشید که آرایه‌های این چنینی و نظیر آن در TCL نیازی به تعریف شدن در ابتدا ندارند.

حال کد زیر همان هفت گره را به صورت توپولوژی دایره‌ای به هم متصل می‌کند. البته کمی اینجا پیچیده شده است.

```
for {set i 0} {$i < 7} {incr i} {
$ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail
}
```

حلقه for تمامی گره‌ها به جز گره آخر را (که به گره اول متصل است) به هم متصل می‌کند.

وقتی که شما اسکریپت را اجرا می‌کنید ممکن است در nam در ابتدا شکل کمی عجیب باشد ولی اگر دکمه "re-layout" را فشار دهید شکل شبیه به زیر خواهد شد.



۳-۳ خرابی لینک

در گام بعد ما می‌خواهیم که گره n(0) به گره n(3) داده‌ها را ارسال کند.

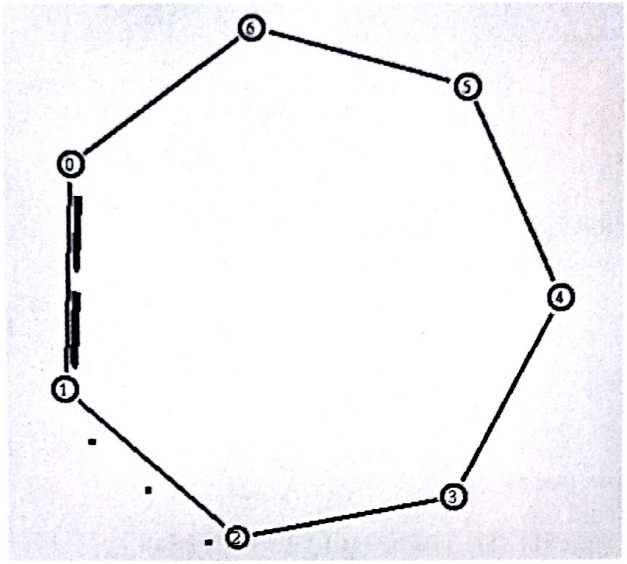
```
#Create a UDP agent and attach it to node n(0)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0
$ns connect $udp0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

کدهای بالا شبیه به همان‌هایی هستند که در قسمت‌های قبل استفاده کرده‌ایم، با این تفاوت که در اینجا از آرایه گره‌ها استفاده می‌شود. اگر اسکریپت را اجرا کنید خواهید دید که ترافیک از کوتاه‌ترین مسیر یعنی از گره ۰ به ۳ از بین گره‌های ۱ و ۲ عبور می‌کند. حال ما یک قابلیت جدید اضافه می‌کنیم که در آن لینک بین گره ۱ و ۲ (که برای ارسال ترافیک استفاده می‌شد) برای یک ثانیه down یا از کار می‌افتد.

```
$ns attach-agent $n(3) $null0
$ns connect $udp0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"_
```

درک این دو خط کار مشکلی نیست. ما در ثانیه ۱,۰ شبیه‌سازی خط بین ۱ و ۲ را غیرفعال و در ثانیه ۲,۰ فعال می‌کنیم. پس در این ثانیه پاکت‌هایی که از گره ۰ به ۳ می‌روند از بین خواهند رفت.

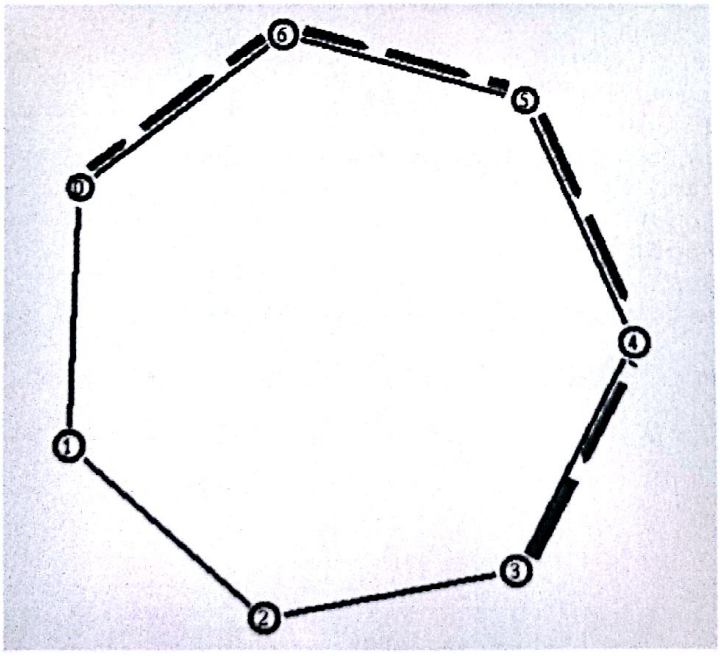
در شکل زیر این اتفاق را در nam مشاهده می‌کنید.



در ادامه ما نشان خواهیم داد که چگونه مسیریابی پویا این مشکل را حل خواهد کرد. حال خط زیر را به ابتدای اسکریپت خود، بعد از ایجاد شئ شبیه‌ساز اضافه کنید.

```
$ns rtproto DV
```

اگر اسکریپت را اجرا کنید در ثانیه شبیه‌سازی خواهیم دید که ترافیک بین گره‌های ۰ و ۳ با استفاده از مسیر گره‌های ۴، ۵ و۶ حل خواهد شد. در شکل زیر نمونه اجرا را مشاهده می‌نمایید.



در زیر کد کامل مثال ما یعنی example3.tcl به طور کامل نوشته شده است.

```
#Create a simulator object
set ns [new Simulator]
#Tell the simulator to use dynamic routing
$ns rtproto DV
```

```
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
#Define a 'finish' procedure
proc finish {} {
global ns nf
$ns flush-trace
#Close the trace file
close $nf
#Execute nam on the trace file
exec nam out.nam &
exit 0
}
#Create seven nodes
for {set i 0} {$i < 7} {incr i} {
set n($i) [$ns node]
}
#Create links between the nodes
for {set i 0} {$i < 7} {incr i} {
$ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail
}
#Create a UDP agent and attach it to node n(0)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
#Create a Null agent (a traffic sink) and attach it to node n(3)
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0
#Connect the traffic source with the traffic sink
$ns connect $udp0 $null0
#Schedule events for the CBR agent and the network dynamics
$ns at 0.5 "$cbr0 start"
$ns rtmodel-at 1.0 down $n(1) $n(2)
$ns rtmodel-at 2.0 up $n(1) $n(2)
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```