2014120116 이진규

Network/Virus on a Network

1.1. 모델 개요

선택한 Virus on a Network 모델은 컴퓨터들이 연결되어 있는 네트워크 상에서 바이러스가 퍼지는 상황에 대한 모델링이다. 바이러스들은 네트워크를 타고 다른 컴퓨터로 전염되며, 설정에 따라 컴퓨터들은 바이러스에 감염될 수도, 바이러스 감염에 저항하거나 면역이 될 수도 있다.

1.2. 동작원리

이 모델은 바이러스 감염으로 시작한다. 사전에 설정되어 있는 감염된 컴퓨터들은 주변에 연결된 컴퓨터들을 감염시키려 시도한다. 설정된 확률에 따라 바이러스에 감염이 되면, 감염된 컴퓨터들은 바이러스에 걸렸는지 스스로 확인한다. 감염된 것이 확인 되었을 때, 컴퓨터는 정상으로 회복되지 못하거나, 회복되거나, 면역상태가 되는 세 가지 동작 중 한 가지를 실행한다. 회복되지 못하면 회복 될 때까지 검사를 반복하고, 회복이 되면 또 다시 감염이 될 때까지 기다리고 있게 된다. 주목할 것은 면역이 되었을 때 인데, 면역 상태가 되면 주변 컴퓨터와 연결이 끊어져서 앞으로 바이러스에 감염될 가능성이 사라진다.

1.3 사용 방법

1. Setup 버튼은 슬라이더에 표시된 값에 따라 모델을 세팅한다.
2. Go 는 모델을 계속해서 실행하는 버튼이다.
3. Number-of-nodes 는 초기에 설정되는 컴퓨터의 개수를 의미한다.
4. Average-node-degree 는 각각의 컴퓨터가 몇개의 컴퓨터와 연결되어 있을지를 설정한다.
5. Initial-outbreak-size 는 초기에 감염된 컴퓨터의 숫자를 결정한다.

6. Virus-spread-chance 는 바이러스에 감염된 컴퓨터가 주변 다른 컴퓨터를 감염시킬 확률을 의미한다.
7. Virus-check-frequency 는 바이러스에 감염된 컴퓨터가 스스로가 감염되었는지 확인하는 빈도를 말한다.
8. Recovery-chance 는 바이러스에 감염된 것을 확인한 컴퓨터가 회복되는 확률을 의미한다.
9. Gain-resistance-chance 는 바이러스에 감염된 것을 확인한 컴퓨터가 면역상태가 되는 확률을 의미한다.

- 이 모델은 실행되는 동안 슬라이더의 변경사항이 모델에 영향을 주도록 구성되어 있다. Go 버튼이 눌러져 있는 동안 슬라이더의 값을 변경할 수 있고, 이 변경사항은 모델에 즉각적으로 반영된다.

1.4 주목할 사항

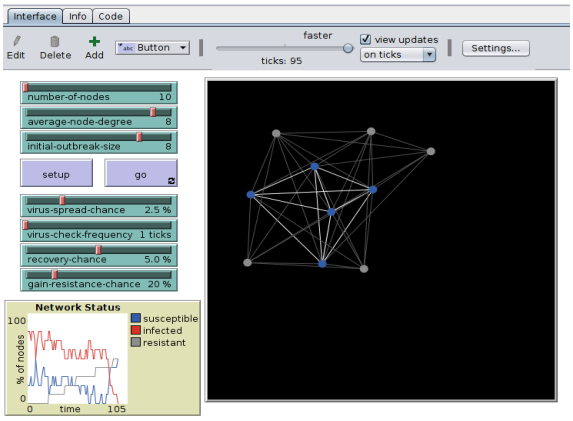
이 모델의 경우 감염되어 있는 컴퓨터가 하나도 남지 않게 되었을 때 동작을 중지한다. 일시적으로 모든 컴퓨터가 감염되어도 모델의 동작이 멈추지 않는데, 이는 컴퓨터가 감염상태를 확인하고 정상으로 회복이나 면역상태로의 전환을 시도하기 때문이다. 따라서, 어느 조건에서 모든 컴퓨터가 감염상태에서 회복되어 모델의 동작이 중지되는지 확인하는 것이 중요하다. 이를 현실에 적용하면 컴퓨터들이 백신을 사용해 바이러스 검사를 하고, 백신의 업데이트를 통해 바이러스로부터 벗어나는 것이라 할 수 있다.

1.5. 시도할 사항

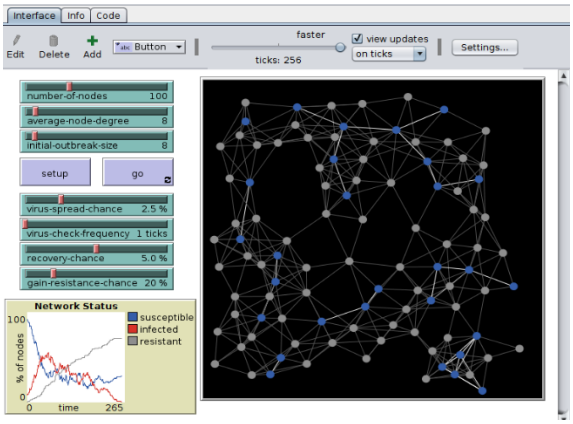
파라미터들의 값을 각각 변화시켜본다. 그리고 이를 통해 만들어지는 모델 실행결과의 변화를 관측한다.

2. 1 다양한 설정값을 이용한 NetLogo 시뮬레이션 결과

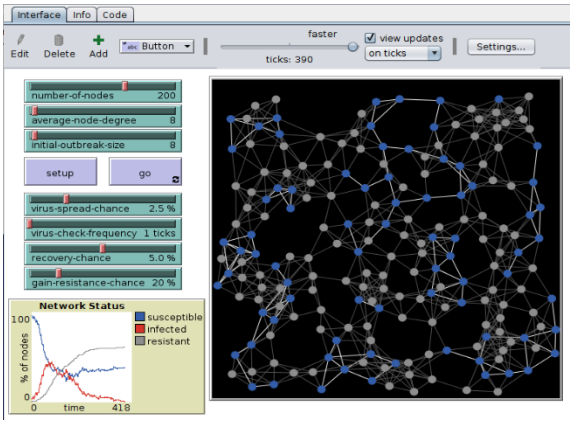
2.1.1 컴퓨터 개수에 따른 실행 결과의 변화



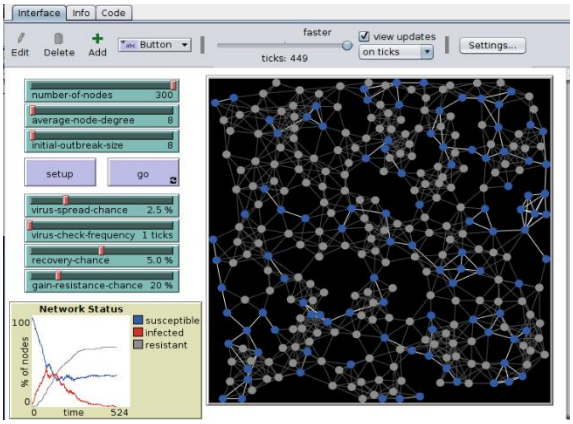
<number-of-nodes 를 10 으로 설정한 결과화면>



<number-of-nodes 를 100 으로 설정한 결과화면>



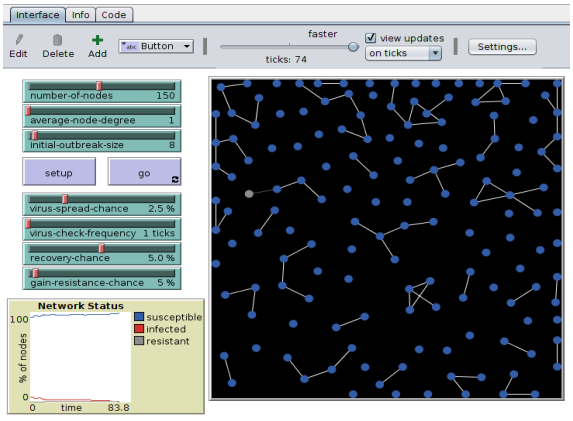
<number-of-nodes 를 200 으로 설정한 결과화면>



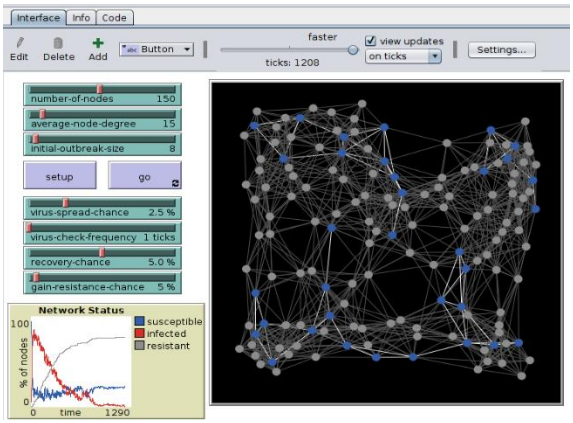
<number-of-nodes 를 300 으로 설정한 결과화면>

초기 컴퓨터 수가 많아질수록 감염까지 걸리는 시간이 증가하는 경향성을 보인다. 이는 감염시켜야 할 컴퓨터가 많을수록 바이러스가 퍼지는데 시간이 오래 걸린다는 사실을 보여준다.

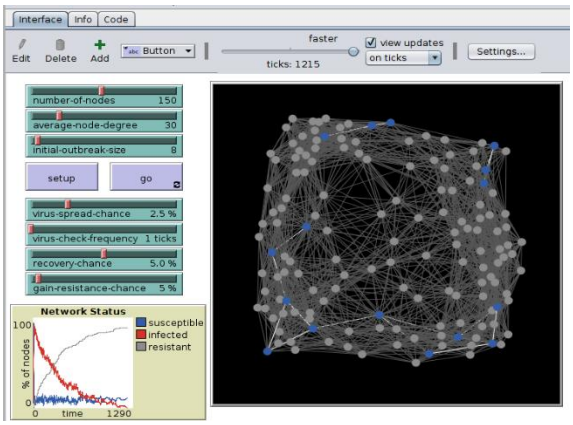
2.1.2 연결된 컴퓨터의 수에 따른 실행 결과의 변화



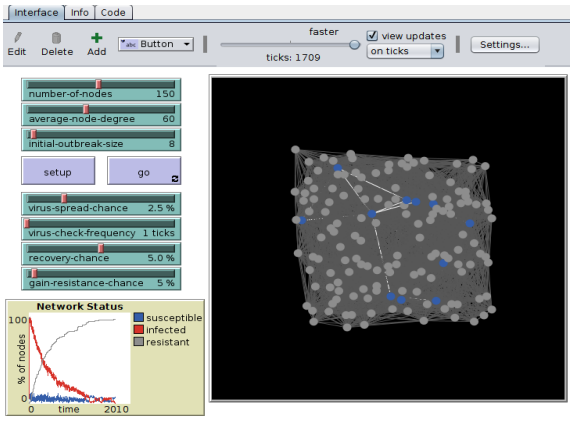
<average-node-degree 를 1 로 설정한 결과화면>



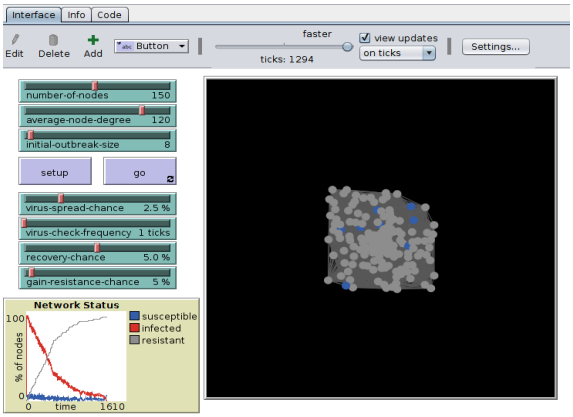
<average-node-degree 를 15 로 설정한 결과화면>



<average-node-degree 를 30 으로 설정한 결과화면>



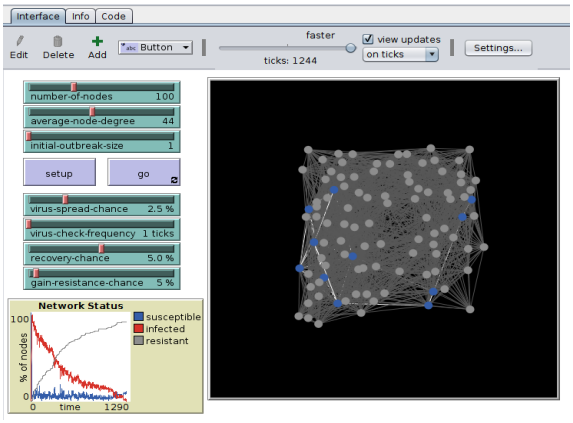
<average-node-degree 를 60 으로 설정한 결과화면>



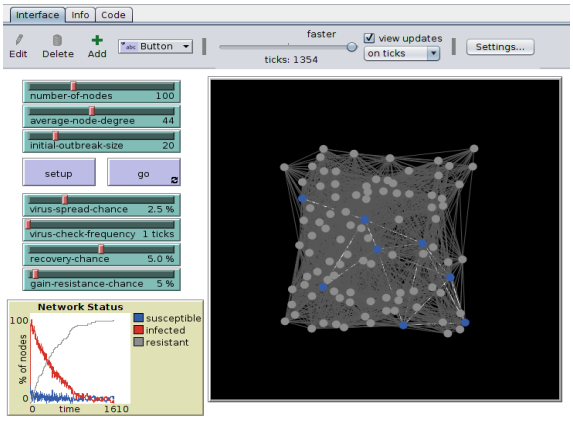
<average-node-degree 를 120 으로 설정한 결과화면>

초기에 설정되어 있는 연결된 네트워크의 수가 극히 적을 때를 제외하고는 설정 값의 변화가 모듈 실행결과의 차이를 만들지 않는다.

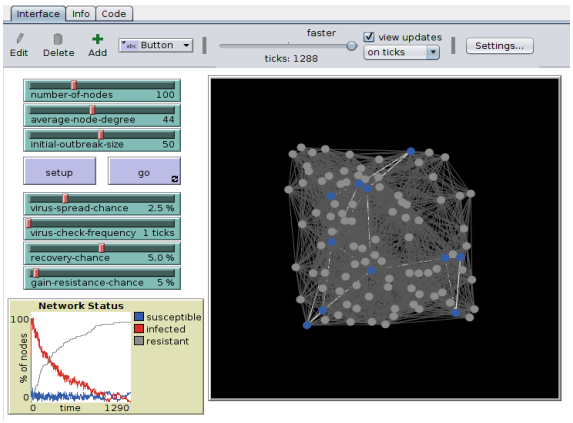
2.1.3 초기 설정된 감염된 컴퓨터의 수에 따른 실행 결과의 변화



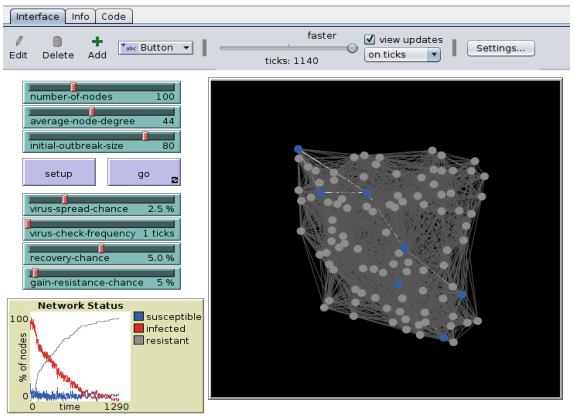
<initial-outbreak-size 를 1 로 설정한 결과화면>



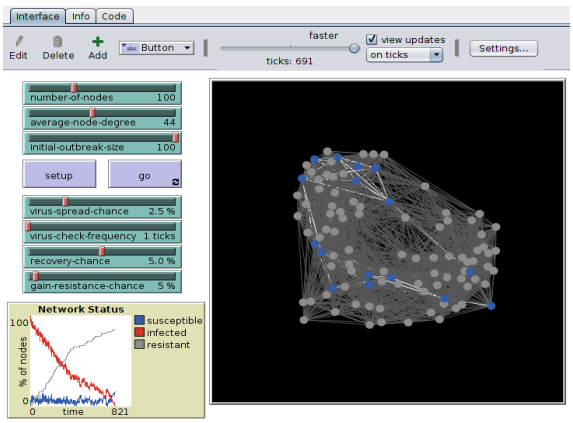
<initial-outbreak-size 를 20 으로 설정한 결과화면>



<initial-outbreak-size 를 50 으로 설정한 결과화면>



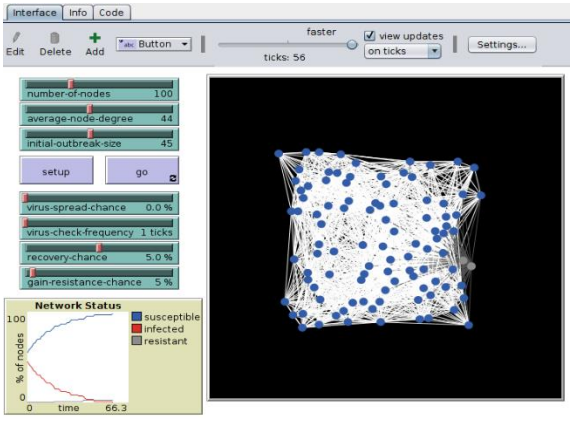
<initial-outbreak-size 를 80 으로 설정한 결과화면>



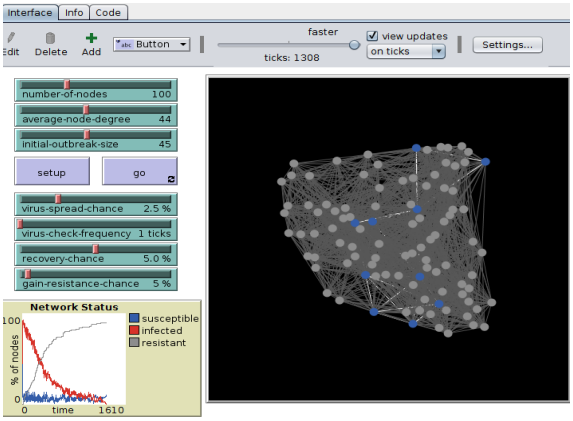
<initial-outbreak-size 를 100 으로 설정한 결과화면>

2.1.2 와 같이 초기 바이러스 설정 값을 변화시킨다고 해서 모듈 동작의 결과가 변화하지 않는다. 이를 실제 컴퓨터로 생각해보면, 바이러스는 초기에 빠른 속도로 확산되기 때문에, 처음 감염된 컴퓨터의 수가 얼마나 되는지는 중요하지 않다는 것을 말해준다.

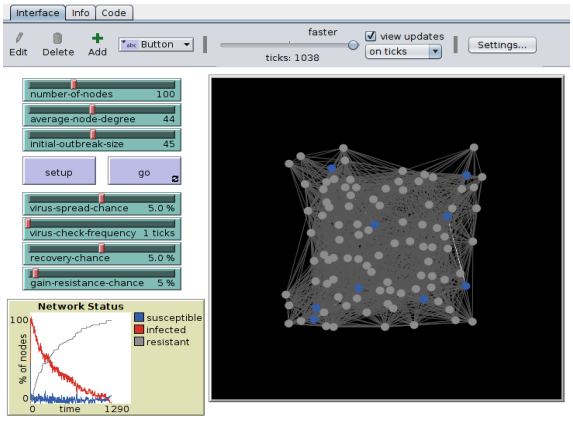
2.1.4 감염확률에 따른 실행 결과의 변화



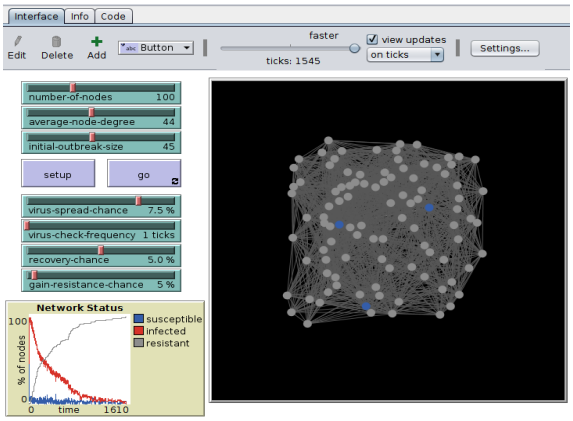
<virus-spread-chance 를 0%로 설정한 결과화면>



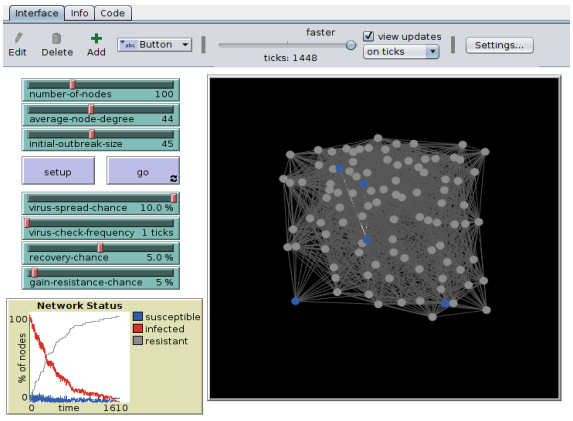
<virus-spread-chance 를 2.5%로 설정한 결과화면>



<virus-spread-chance 를 5%로 설정한 결과화면>



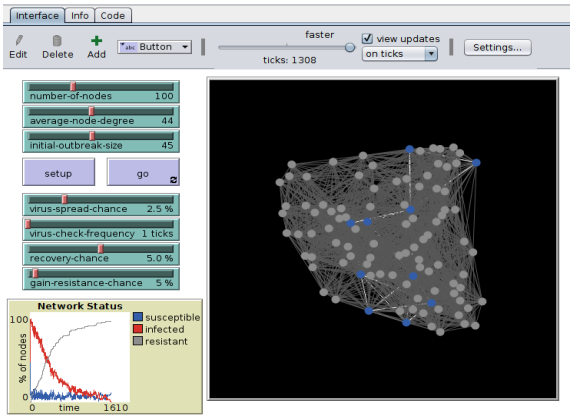
<virus-spread-chance 를 7.5%로 설정한 결과화면>



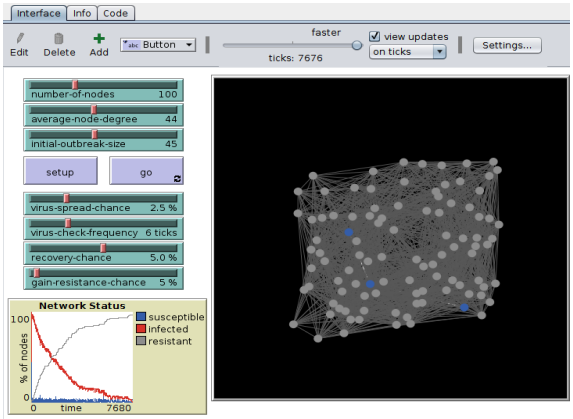
<virus-spread-chance 를 10%로 설정한 결과화면>

감염확률이 0%인 경우를 제외하고는 실행결과가 모두 같았다. 이는 바이러스의 감염성공확률이 바이러스 확산에 차이를 만들어 내지 않는다는 것을 보여준다.

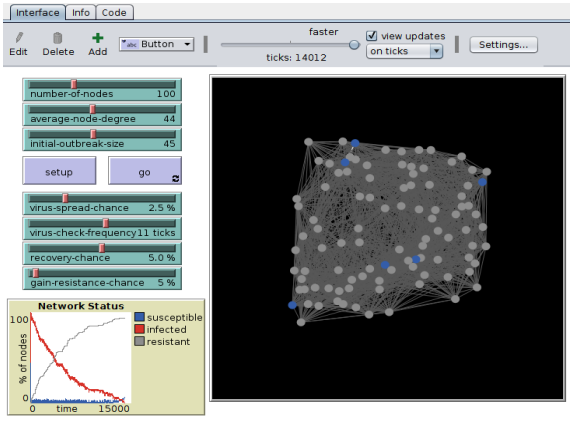
2.1.5 바이러스 감염확인 빈도에 따른 실행결과의 변화



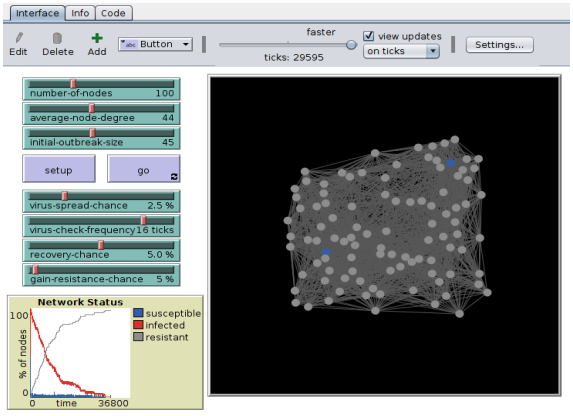
<virus-check-frequency 를 1tick 으로 설정한 결과화면>



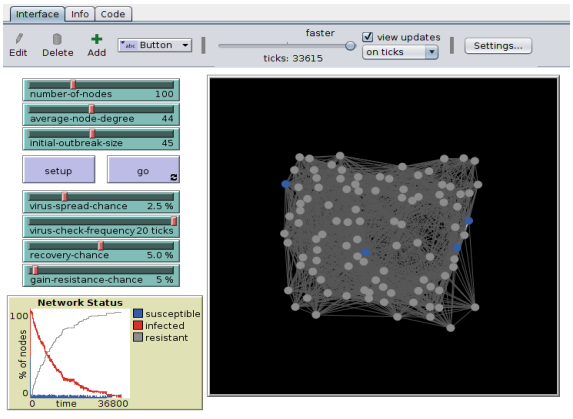
<virus-check-frequency 를 6ticks 으로 설정한 결과화면>



<virus-check-frequency 를 11ticks 으로 설정한 결과화면>



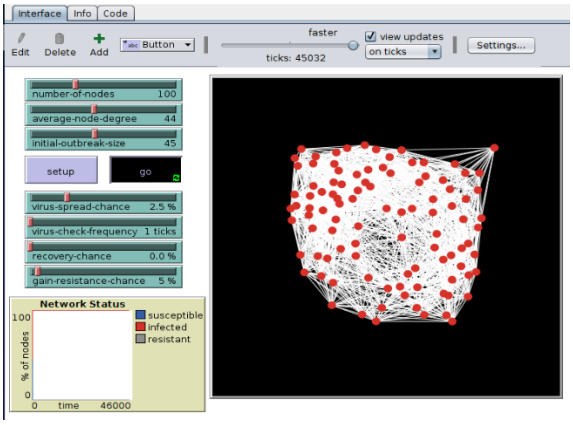
<virus-check-frequency 를 16ticks 으로 설정한 결과화면>



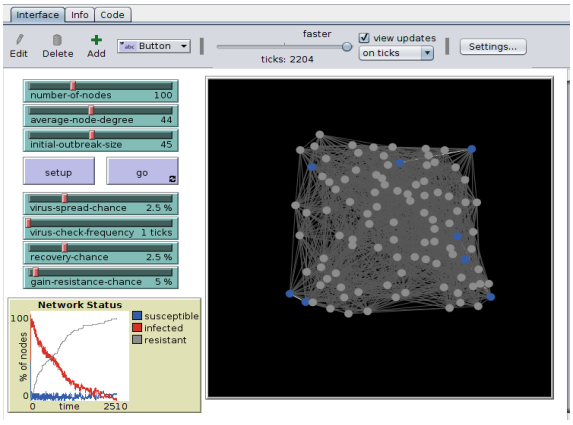
<virus-check-frequency 를 20ticks 으로 설정한 결과화면>

바이러스 감염을 확인하는 간격이 늘어날수록 모든 컴퓨터가 감염에서 벗어나기까지 소요되는 시간도 점점 늘어난다. 따라서, 바이러스를 자주 확인하는 것이 바이러스 확산을 막는데 중요하다는 것을 알 수 있다.

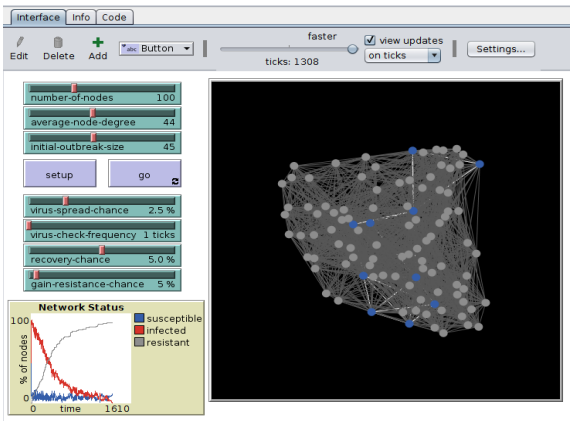
2.1.6 바이러스 감염에서 벗어나는 확률에 따른 실행결과의 변화



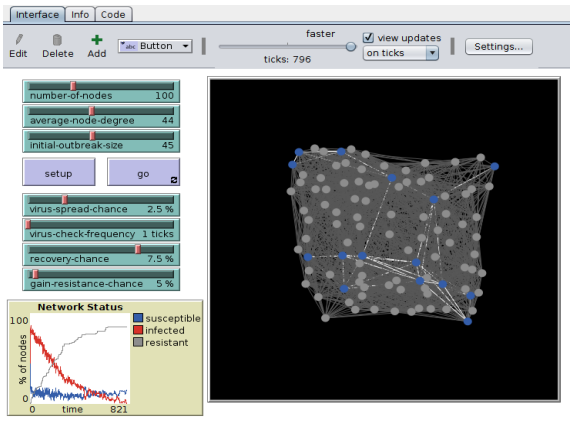
<recovery-chance 를 0%로 설정한 결과화면>



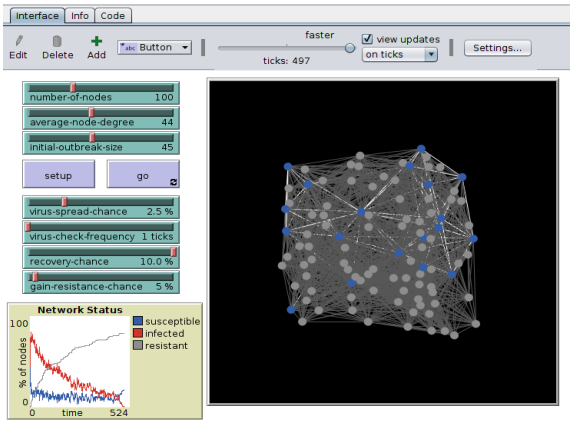
<recovery-chance 를 2.5%로 설정한 결과화면>



<recovery-chance 를 5%로 설정한 결과화면>



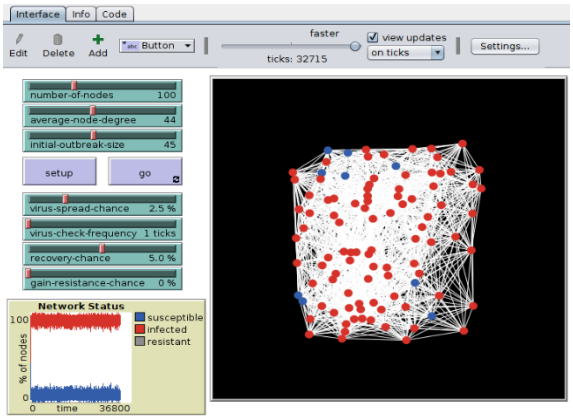
<recovery-chance 를 7.5%로 설정한 결과화면>



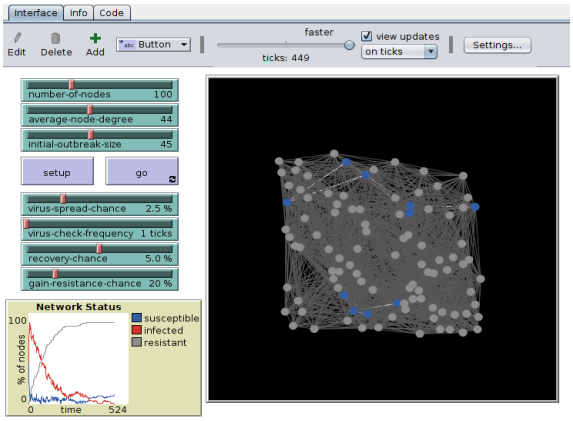
<recovery-chance 를 10%로 설정한 결과화면>

감염상태에서 회복률이 0%일 때는 모듈이 계속해서 실행된다. 그 이외의 경우에는 회복률이 올라감에 따라 감염에서 벗어나는 시간이 줄어든다. 이는 컴퓨터를 바이러스의 감염으로부터 회복시켜줄 수 있는 백신프로그램의 성능의 중요성을 알려준다.

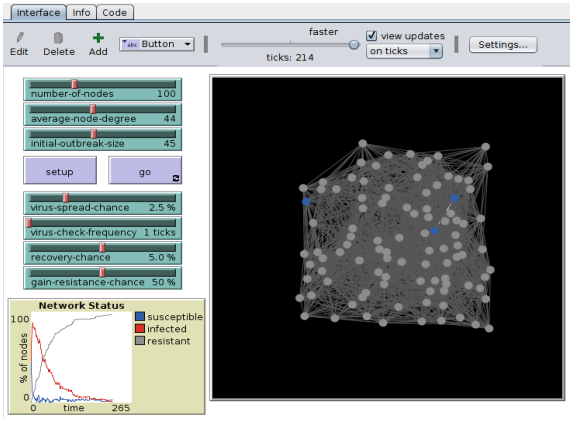
2.1.7 바이러스에서 면역이 되는 확률에 따른 실행결과의 변화



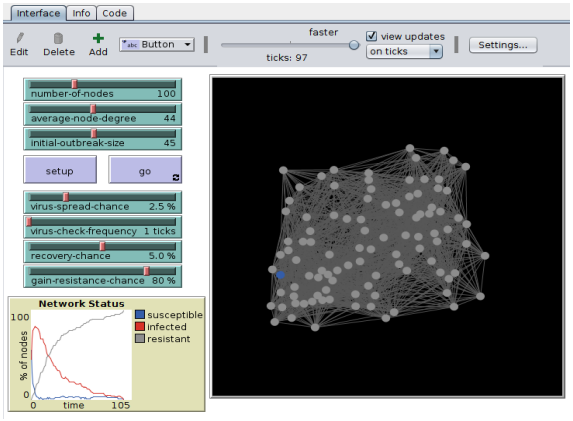
<gain-resistance-chance 를 0%로 설정한 결과화면>



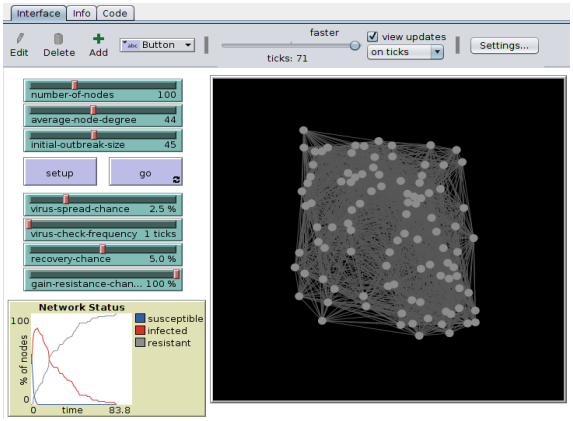
<gain-resistance-chance 를 20%로 설정한 결과화면>



<gain-resistance-chance 를 50%로 설정한 결과화면>



<gain-resistance-chance 를 80%로 설정한 결과화면>



<gain-resistance-chance 를 100%로 설정한 결과화면>

바이러스 면역률을 0%로 설정하면 감염된 컴퓨터와 정상인 컴퓨터가 일정비율을 유지하면서 모델이 계속해서 실행된다. 그리고 바이러스 면역률이 높아질수록 모든 컴퓨터가 바이러스 감염에서 벗어날 때까지 걸리는 시간이 줄어든다.

2.2 정리

위에서 시행한 다양한 시뮬레이션 결과를 종합해보면 다음과 같은 사실을 알 수 있다.

- 바이러스의 초기확산 속도는 엄청나게 빠르기 때문에 처음부터 확산을 막는 것은 어려운 일이다. 따라서, 확산을 막는 것 보다 바이러스가 퍼진 이후 얼마나 빨리 감염된 컴퓨터를 치료하는지가 더 중요하다.
- 감염된 모든 컴퓨터들을 정상으로 되돌리는데 영향을 끼치는 변수는 다양하다. 하지만 그 중에서도 가장 큰 영향을 끼치는 것은 얼마나 자주 바이러스를 검사하는 지 이다. 다른 요소들의 영향은 바이러스 검사빈도에 비교하면 아주 미미한 영향을 미칠 뿐이다. 따라서, 바이러스의 위협으로부터 벗어나기 위해서는 주기적인 바이러스 검사를 하는 것이 매우 중요하다.