Математические структуры и моделирование
2010, вып. 21, с. 113–118

УДК 621.394

МОДЕЛИРОВАНИЕ СИСТЕМЫ ПРОТИВОДЕЙСТВИЯ DOS-АТАКАМ

С.В. Белим, С.Ю. Белим

В работе проводится моделирование системы противодействия DOS-атакам с помощью случайного уничтожения пакетов во входном буфере. Исследуется несколько возможных режимов работы системы.

Введение

Одним из трех аспектов информационной безопасности является доступность, то есть отклик системы на запрос за заданный промежуток времени. Для обеспечения доступности компьютерной системы необходимо принимать меры по обеспечению устойчивости системы к всплеску интенсивности поступающей информации. Как правило, соответствующие службы учитываются уже при проектировании системы. Принято говорить об архитектуре системы обработки входящих пакетов.

Простейшей является архитектура OQ (Output Queuing), в которой используется несколько входных линий и столько же обработчиков пакетов, причем каждый обработчик работает со своей входной линией. Архитектура OQ не подразумевает никаких дополнительных средств обеспечения доступности, кроме расчета устойчивого режима с помощью теории массового обслуживания.

В качестве примера можно привести архитектуру VOQ (Virtual Output Queuing), которая предполагает наличие алгоритма посылки поступающих пакетов обработчикам по циклу [1]. Другая архитектура, рассматриваемая в работах [2, 3], основывается на том, что 90% трафика составляет протокол TCP и сосредотачивается на контроле TCP-пакетов.

Данная статья посвящена моделированию состояния буфера входящих пакетов в рамках архитектуры FOQ (Feedback Output Queuing), предложенной в работе [4]. Данная архитектура предполагает обратную связь обработчика пакетов с входным буфером. Обработчики следят за состоянием буфера и уничтожают пакеты для предотвращения переполнения.

Омский государственный университет им. Ф.М. Достоевского.
E-mail: sbelim@omsu.ru

1. Постановка задачи

Рассмотрим компьютерную систему, обрабатывающую пакеты, поступающие во входной буфер длины L. Рассмотрение начнем с простой ситуации поступления пакетов с постоянной скоростью u. Более точно, пусть в буфер в единицу времени поступает u пакетов. Как обычно при моделировании компьютерных систем, время считаем дискретным, отсчитываемым по системному таймеру. Интервал между двумя «тиками» таймера примем за единицу. Будем считать, что поступающие пакеты обрабатываются системой с постоянной скоростью w пакетов в единицу времени. Обозначим количество пакетов в буфере в момент времени n через x_n, тогда без дополнительных систем уничтожения пакетов состояние буфера описывается разностным уравнением

$$x_{n+1} - x_n = u - w.$$

Или, вводя обозначение $v = u - w$, получаем рекуррентное соотношение

$$x_{n+1} = x_n + v.$$

Таким образом, задача разбивается на два случая: $v <= 0$ и $v > 0$. В первом случае ($v <= 0$) дополнительные средства очистки буфера не нужны, так как $x_{n+1} >= x_n$ для любого момента времени n. Этот результат достаточно очевиден, так как в этом случае скорость обработки пакетов не меньше скорости поступления пакетов ($u <= w$), и, как следствие, не происходит переполнение буфера. Обратный случай ($v > 0$), наоборот, приводит к росту количества пакетов в буфере с течением времени, что приводит к переполнению буфера. Как известно, такой способ нарушения работоспособности компьютерной системы получил название DOS-атаки.

Один из возможных способов борьбы с переполнением буфера был предложен в работе [7]. Основная идея метода состоит в уничтожении случайно выбранных пакетов во входном буфере в случае возрастания скорости поступления пакетов. Безусловно, при таком подходе существует не нулевая вероятность уничтожения «полезных» пакетов. Однако при отсутствии ответа на запрос серверы посылают повторный запрос. Вероятность же случайного уничтожения всех пакетов одного сервера достаточно мала.

2. Постоянный поток пакетов

Для рассматриваемой модели со случайным уничтожением пакетов рекуррентное соотношение примет вид:

$$x_{n+1} = x_n + v - d_n.$$

Здесь $v > 0$, а d_n – количество случайно уничтожаемых пакетов в момент времени n. Случайно уничтожаемые пакеты отбрасываются без обработки.

Существенным вопросом является выбор последовательности d_n, которая, очевидно, должна зависеть от заполнения буфера. Рассмотрим несколько возможных случаев:

1. $d_n = \alpha x_n$ – количество уничтожаемых пакетов прямо пропорционально количеству пакетов в буфере ($0 < \alpha < 1$). Рекуррентное соотношение будет иметь вид:

$$x_{n+1} = (1 - \alpha)x_n + v.$$

Для его решения необходимо составить соответствующее однородное уравнение. Выпишем состояние буфера в предыдущий момент времени:

$$x_n = (1 - \alpha)x_{n-1} + v$$

и, выразив из него v, подставим в предыдущее рекуррентное соотношение:

$$x_{n+1} = (2 - \alpha)x_n - (1 - \alpha)x_{n-1}.$$

Общее решение данного уравнения имеет вид:

$$x_n = C_1 + C_2(1 - \alpha)^n,$$

где C_1 и C_2 – константы. Начальное состояние системы – $x_0 = 0$. Для момента времени $n = 1$ из рекуррентного соотношения получаем значение $x_1 = v$. Можем найти частное решение, удовлетворяющее начальным условиям:

$$x_n = \frac{v}{\alpha}(1 - (1 - \alpha)^n).$$

В пределе больших n получаем

$$\lim_{n->\infty} x_n = \frac{v}{\alpha}.$$

Причем для всех n переполнение буфера не будет происходить, если для всех моментов времени n выполняется $x_n < L$, откуда получаем условие на коэффициент α:

$$\alpha > \frac{v}{L}.$$

2. $d_n = \alpha(x_n - x_{n-1})$ – количество уничтожаемых пакетов прямо пропорционально количеству поступающих в буфер пакетов ($0 < \alpha < 1$). Рекуррентное соотношение будет иметь вид:

$$x_{n+1} = (1 - \alpha)x_n + v - \alpha x_{n-1}.$$

Для его решения необходимо составить соответствующее однородное уравнение. Выпишем состояние буфера в предыдущий момент времени:

$$x_n = (1 - \alpha)x_{n-1} + v - \alpha x_{n-2}$$

и, выразив из него v, подставим в предыдущее рекуррентное соотношение:

$$x_{n+1} = (2 - \alpha)x_n - (1 - 2\alpha)x_{n-1} - \alpha x_{n-2}.$$

Общее решение данного уравнения имеет вид:

$$x_n = C_1 + C_2 n + C_3(-\alpha)^n,$$

где C_1, C_2 и C_3 – константы. Начальное состояние системы – $x_0 = 0$. Для момента времени $n = 1$ $x_1 = u$, для $n = 2$ из рекуррентного соотношения получаем значение $x_2 = u + v$. Можем найти частное решение, удовлетворяющее начальным условиям:

$$x_n = \frac{u - v}{(1+\alpha)^2} + \frac{2\alpha u + v}{(1+\alpha)} n + \frac{v - u}{(1+\alpha)^2}(-\alpha)^n.$$

Несложно заметить, что x_n растет со временем, однако не монотонно за счет слагаемого $(-\alpha)^n$. Тем не менее для любого заданного размера буфера L существует момент времени k такой, что $x_k > L$.

3. Случайный поток пакетов

Рассмотрим случай, в котором количество прибывающих пакетов v_n, а следовательно, и количество пакетов в буфере x_n являются случайной величиной с гауссовым распределением. Выпишем средние значения случайной величины:

$$< v_n >= v, \ < v_n v_k >= \delta_{nk}\sigma_v,$$

где δ_{nk} – символ Кронекера.

Найдем соответствующие средние значения для количества пакетов в буфере. Выберем количество уничтожаемых пакетов в виде $d_n = \alpha x_n$. Тогда рекуррентное соотношение примет вид:

$$x_n = (1 - \alpha)x_{n-1} + v_n.$$

Усреднение его по времени приводит к выражению

$$< x_n >= (1 - \alpha) < x_{n-1} > + v.$$

Отсюда можно получить выражение

$$< x_n >= \frac{v}{\alpha}(1 - (1 - \alpha)^n).$$

То есть при надлежащем выборе коэффициента α, как и в случае постоянного потока пакетов, можно добиться стабильной работы системы без переполнения буфера. Однако при случайной интенсивности поступления пакетов возможны резкие всплески количества пакетов в буфере, поэтому необходимо также оценить дисперсию случайной величины x_n:

$$(x_{n+1})^2 = (1 - \alpha)^2 (x_n)^2 + 2(1 - \alpha)x_n v_n + (v_n)^2.$$

Выражая v_n из рекуррентного выражения для x_n, получаем следующее соотношение:

$$(x_{n+1})^2 = (1 - \alpha)^2 (x_n)^2 + 2(1 - \alpha)(x_n)^2 - 2(1 - \alpha)^2 x_n x_{n-1} + (v_n)^2.$$

Усредняя это выражение по времени, получаем

$$< (x_{n+1})^2 >=$$

$$= (1-\alpha)^2 < (x_n)^2 > +2(1-\alpha) < (x_n)^2 > -2(1-\alpha)^2 < x_n x_{n-1} > + < (v_n)^2 > .$$

Предполагая, что x_n имеет гауссово распределение со средними значениями:

$$< x_n >= \frac{v}{\alpha}(1-(1-\alpha)^n, \ < x_n x_k >= \sigma_x \delta_{nk}.$$

Откуда получаем

$$\sigma_x = \frac{\sigma_v}{2-(2-\alpha)^2}.$$

Следовательно, дисперсия наполнения буфера прямо пропорциональна дисперсии интенсивности поступающих пакетов.

По хорошо известному из теории вероятностей правилу «трех сигм» для нормального распределения с вероятностью 0.9973 случайная величина x_n будет попадать в интервал $[< x_n > -3\sigma_x, < x_n > +3\sigma_x)$. Следовательно, чтобы с вероятностью 0.9973 не происходило переполнение буфера, необходимо, чтобы $< x_n > +3\sigma_x < L$. Отсюда получаем более жесткое, чем в случае постоянного потока, условие на коэффициент α:

$$\frac{v}{\alpha} + 3\frac{\sigma_v}{2-(2-\alpha)^2} < L.$$

Данное неравенство сводится к неравенству третьей степени, которое всегда имеет решение. Значение параметров входящего потока v и σ_v может быть определено экспериментально. Соответственно настройка системы производится выбором значения α. Рассмотренные соотношения не позволяют получить точное оптимальное значение параметра α, а только определяют его граничные значения.

4. Заключение

Таким образом, задача построения системы защиты от DOS-атак с помощью механизма уничтожения случайным образом входящих пакетов во входном буфере разрешима. Степень надежности системы может варьироваться с помощью изменения параметра системы, отвечающего за активность уничтожения пакетов. При этом следует учитывать, что активность системы защиты снижает скорость обработки информации. Но это общеизвестный факт, касающийся всех систем защиты, к нему уже все привыкли и он не считается недостатком. Кроме того система случайного уничтожения входящих пакетов не является гарантированной защитой от DOS-атак. Но и в этом случае мы сталкиваемся с общеизвестным фактом: гарантированной защиты не бывает в принципе.

Литература

1. Nong G., Hamdi M. On the provisioning of Quality of Service guarantees for input queud switches //IEEE Communications Magazine. 2000. V. 38(12). P.62–69.
2. Jacobson V. Congestion avoidance and control //Proceeding of ACM SIGCOMM'88. 1988. Stanford. P.314–329.
3. Stevens W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms // IETF RFC 2001, January 1997.
4. Firoiu V., Zhang X., Gunduzhan E., Christin N. Providing service guarantees in high-speed switching systems with feedback output queuing. // arXiv:cs/0406019v1.