

УДК 681.324

Г.А. КУЧУК, А.А. МОЖАЕВ, И.В. ИЛЬИНА

Харьковский университет Воздушных Сил им. Ивана Кожедуба, Украина

МЕТОД ОПРЕДЕЛЕНИЯ БАЗОВОГО МНОЖЕСТВА ПУТЕЙ ПЕРЕДАЧИ ИНФОРМАЦИИ В КОМПЬЮТЕРНОЙ СЕТИ СИСТЕМЫ КРИТИЧЕСКОГО ПРИМЕНЕНИЯ

Предложен метод определения базового множества путей передачи информации в компьютерной сети системы критического применения. Рассмотрено условие постоянного отсутствия циклов, позволяющее избежать «зацикливания» информации. Проведена сравнительная оценка эффективности метода по сравнению с известными. Определено, что при отсутствии флуктуаций трафика разработанный метод сопоставим с известными методами по времени задержки передачи информации и в несколько раз оказывается более эффективным при резких флуктуациях входного трафика.

базовое множество путей передачи информации, компьютерная сеть, система критического применения, маршрутизация, управление сетевыми ресурсами

Введение

Постановка проблемы. На современном этапе развития систем критического применения для достижения мировых стандартов качества услуг (*QoS*) необходимо совершенствовать методы и средства управления сетевыми ресурсами с учетом тенденции развития информационных технологий и глобализации компьютерных сетей. Эти сети должны обеспечивать передачу различного рода информации (звуковой, видео, данных и др.) с гарантией качества обслуживания. Для решения этой задачи целесообразно применение в компьютерных сетях систем критического применения новых методов управления сетевыми ресурсами.

Анализ литературы [1 – 6] показал, что в настоящее время применяемые на практике протоколы управления сетевыми ресурсами базируются на следующих методах обмена служебной информации:

- дистанционно-векторные;
- состояния связей.

Отметим, что методы состояния связей отличаются более быстрой сходимостью и менее «склонны» к образованию петель маршрутизации, чем дистанционно-векторные методы. С другой стороны, в сравне-

нии с последними они характеризуются большей вычислительной сложностью. Реализация методов состояния связей в условиях временных ограничений требует большей производительности и памяти используемых вычислительных средств. Комбинирование применения рассмотренных методов маршрутизации позволяет компенсировать их взаимные недостатки и повысить эффективность использования компьютерных сетей в сравнении со случаем использования каждого метода в отдельности.

При решении задачи определения множества $\mathcal{N}_{\text{баз}}$ путей передачи информации в компьютерной сети системы критического применения для УС i и j из множества \mathcal{R} [3] сначала необходимо найти кратчайшее «расстояние» (минимальное время передачи информационных пакетов) $T_{i,j \min}$ от источника i к адресату j и множество $S_j^{(i)}$ узлов k , ближайших к УС i и лежащих по направлению движения потока к j (множество «узлов-преемников») в порядке уровней иерархии дерева допустимых маршрутов множества U :

$$T_{i,j \min} = \min_{k \in \mathcal{R}} \{ T_{k,j} + t_{i,k} \}; \quad (1)$$

$$S_j^{(i)} = \{k \mid k \subset \mathfrak{R}\}, \quad (2)$$

где $t_{i,k}$ – «расстояние» от узла i к узлу k ;

$T_{k,j}$ – «расстояние» от узла k к адресату j .

При решении поставленной в (1)–(2) задачи известными алгоритмами поиска кратчайших путей [2] в большинстве практических случаев приходится сталкиваться с проблемой «зацикливания» информации в найденных путях («зацикливания» маршрутов). Это приводит к увеличению времени передачи информационных пакетов, а зачастую и их потере. Избежать «зацикливания» информации можно введя ограничения (условие постоянного отсутствия циклов), представленные в виде выражений:

$$T_{k,j} \leq T_{i,j \min}; \quad (3)$$

$$T_{k,j \min} \leq T_{i,k,j}, \quad k \subset \mathfrak{R}, \quad (4)$$

где $T_{k,j \min}$ – кратчайшее «расстояние» (минимальное время передачи информационных пакетов) от узла k к адресату j ;

$T_{i,k,j}$ – «расстояние» (время передачи информационных пакетов) от узла i к адресату j через узел k .

Решение задачи поиска множества путей, исключающих «циклы», разобьем на два этапа:

– определение кратчайшего «расстояния» $T_{i,j \min}$ от источника i к адресату j ;

– нахождение множества $S_j^{(i)}$ «узлов-преемников» на маршрутах, исключающих «цикличность», для произвольных источников i и адресатов j в соответствии с порядком вложенности множества U уровней иерархии дерева допустимых маршрутов.

1. Поиск кратчайших путей от источника к адресатам.

На первом этапе для определения кратчайшего «расстояния» $T_{i,j \min}$ целесообразно использовать алгоритм построения предварительной топологии. Алгоритм создан на основе известных алгоритмов состояния связей (OSPF, SPTA) [2]. В отличие от

известных в алгоритме построения предварительной топологии учитывается иерархичность построения компьютерной сети системы критического применения (определяются «расстояния» от «узла-источника» i до «узла-адресата» j в соответствии с существующими уровнями иерархии), что в дальнейшем позволит осуществить пропорциональное (с учетом коэффициентов ϕ_s и $\phi_s^{(c)}$ [3]) распределение информации по найденным маршрутам.

Алгоритм выполняется каждый раз при получении от соседнего узла служебного сообщения об обновлении состояния связей, при обнаружении обрыва связи, изменении конфигурации в компьютерной сети системы критического применения и в ряде других случаев.

В момент введения узла связи в структуру компьютерной сети в соответствии с алгоритмом выполняется инициализация таблицы маршрутизации. При этом все переменные, характеризующие расстояния между узлами связи и их количество, стремятся к бесконечности и пустому множеству соответственно. В каждом узле связи i , инициализирующем выполнение алгоритма построения предварительной топологии, имеется следующая информация:

– основная таблица топологии $D^{(i)}$, содержащая информацию о связях с узлами ξ , доступными узлу i ;

– таблица топологии «соседей» $D_k^{(i)}$, в которой содержится информация о состоянии связи («расстоянии» $t_{i,k}$) с соседним узлом k ;

– таблицы расстояний, содержащие информацию о «расстояниях» $T_{i,k,j}$ и $T_{k,j}$ от узла i и соседних с ним узлов k к адресатам j на основе таблиц топологии $D^{(i)}$ и $D_k^{(i)}$ соответственно;

– таблица маршрутизации, содержащая информацию о множестве $S_j^{(i)}$ «узлов-преемников» k , текущем $T_{i,k,j}$ и кратчайшем $T_{i,j \min}$ «расстояниях» до каждого адресата j .

Согласно алгоритму, в первую очередь выполняется процедура корректировки таблицы топологии соседних узлов $D_k^{(i)}$, в которой при добавлении или удалении связи (i, k) происходит изменение «расстояния» $t_{i,k}$ (соответственно добавление или очищение записи в $D_k^{(i)}$) и посылается сообщение об обновлении состояния связей соседнему узлу с информацией для основной таблицы топологии $D^{(i)}$. Это же сообщение посылается и при изменении величины «расстояния» $t_{i,k}$, что также влечет за собой обновление таблицы топологии соседних узлов $D_k^{(i)}$.

Информация обновленной таблицы $\tilde{D}_k^{(i)}$ топологии соседних узлов служит для поиска кратчайших путей и соответственно определения кратчайших «расстояний» $T_{i,j \min}$ от любого УС i до любого УС j из множества \mathfrak{R} . Такой поиск в рамках разрабатываемого метода адаптивной маршрутизации целесообразно проводить с помощью известного алгоритма Дейкстры, обладающего по сравнению с другими известными алгоритмами (Флойда, Данцига и др.) наименьшей вычислительной сложностью. Поэтому следующим этапом выполнения процедуры корректировки таблицы топологии соседних узлов является поиск кратчайшего пути по полученной таблице топологии $D_k^{(i)}$ и обновление таблицы расстояний новой информацией о «расстоянии» $T_{k,j}$ из таблицы $D_k^{(i)}$.

На этом заканчивается выполнение процедуры корректировки таблицы топологии соседних узлов. Остальные шаги представляют собой выполнение процедуры корректировки основной таблицы топологии $D^{(i)}$. При этом создается собственное дерево кратчайшего пути по информации о топологии, передаваемой от соседних узлов.

Выполнение этой процедуры начинается с сохранения «старой» таблицы топологии $D^{(i)}$ с целью ее дальнейшего сравнения с обновленной таблицей

$\tilde{D}^{(i)}$. Далее для получения информации о состоянии связи со всеми соседними узлами («узлами-преемниками») элементы таблицы $D^{(i)}$ проверяются на предмет «наличия» («отсутствия») информации о некотором текущем узле ξ первого уровня иерархии U для произвольного «узла-источника» i и состояниях связи с этим узлом. В случае «наличия» такой информации выполнение алгоритма продолжается с первого шага, в противном случае информация об узле ξ и состоянии связи с ним заносится в таблицу $D^{(i)}$.

Если некоторый текущий узел ξ такой, что $T_{\xi,j} + t_{i,\xi} \leq \min_{k \in \mathfrak{R}} \{T_{k,j} + t_{i,k}\}$, т.е. маршрут через узел ξ является кратчайшим, то принимается решение об изменении кратчайшего маршрута передачи информации, и соответствующие изменения вносятся в таблицу $D^{(i)}$. В результате перебора всех узлов ξ по уровню иерархии u_α происходит обновление записей о состоянии связей («расстоянии» $t_{i,\xi}$) и формируется «обновленная» таблица $\tilde{D}^{(i)}$. По результатам изменений «расстояния» $T_{i,k,j}$ обновляются соответствующие таблицы расстояний.

По результатам сравнения «старой» таблицы топологии $D^{(i)}$ с вновь сформированной («обновленной») таблицей $\tilde{D}^{(i)}$ происходит оповещение «узлов-преемников» об изменениях в $D^{(i)}$.

В результате в таблице $D^{(i)}$ формируется информация о множестве $\aleph_{\text{баз}}$ маршрутов передачи информации для любой пары УС i и j из множества \mathfrak{R} .

Узлы связи переходят в состояние ожидания очередного события и в момент прихода сообщения об обновлении состояния связей алгоритм повторяется.

2. Нахождение «узлов-преемников» на маршрутах к каждому адресату.

Для нахождения множества путей передачи информации, исключая «цикличность», на этом

этапе используется алгоритм нахождения «узлов-преемников» на маршрутах к каждому адресату, структурная схема которого представлена на рис. 1.

В отличие от известных алгоритмов [2, 4 – 6], в которых не учитывается текущее состояние УС («АКТИВНОЕ» или «ПАССИВНОЕ») и в этой связи необходима синхронизация обмена служебными сообщениями о состоянии связей по всей сети, в алгоритме нахождения «узлов-преемников» на маршрутах к каждому адресату такой синхронизации подлежит только один переход между соседними узлами. Это значительно упрощает работу узлов связи и сокращает время сходимости алгоритмов.

После проверки состояния УС (в «АКТИВНОМ» состоянии в УС выполняется процедура корректировки таблицы топологии соседних узлов или корректировки основной таблицы топологии) и (при необходимости) перевода УС в «ПАССИВНОЕ» состояние производится поиск множества путей передачи информации и определяется множество $\mathcal{N}_{\text{баз}}$, соответствующее первому уровню иерархии U .

Для выполнения условия постоянного отсутствия циклов с помощью ограничений (3) и (4) в множестве $\mathcal{N}_{\text{баз}}$ исключаются маршруты, приводящие к такому результату и формируется множество $\tilde{\mathcal{N}}_{\text{баз}}$ маршрутов, исключаящих «зацикливание» информации.

Для определения возможности обеспечения на маршрутах найденного множества $\tilde{\mathcal{N}}_{\text{баз}}$ своевременной передачи информации о воздушной обстановке на следующем этапе определяются коэффициенты $\phi_s^{(\text{баз})}$ распределения потока информации по s -му

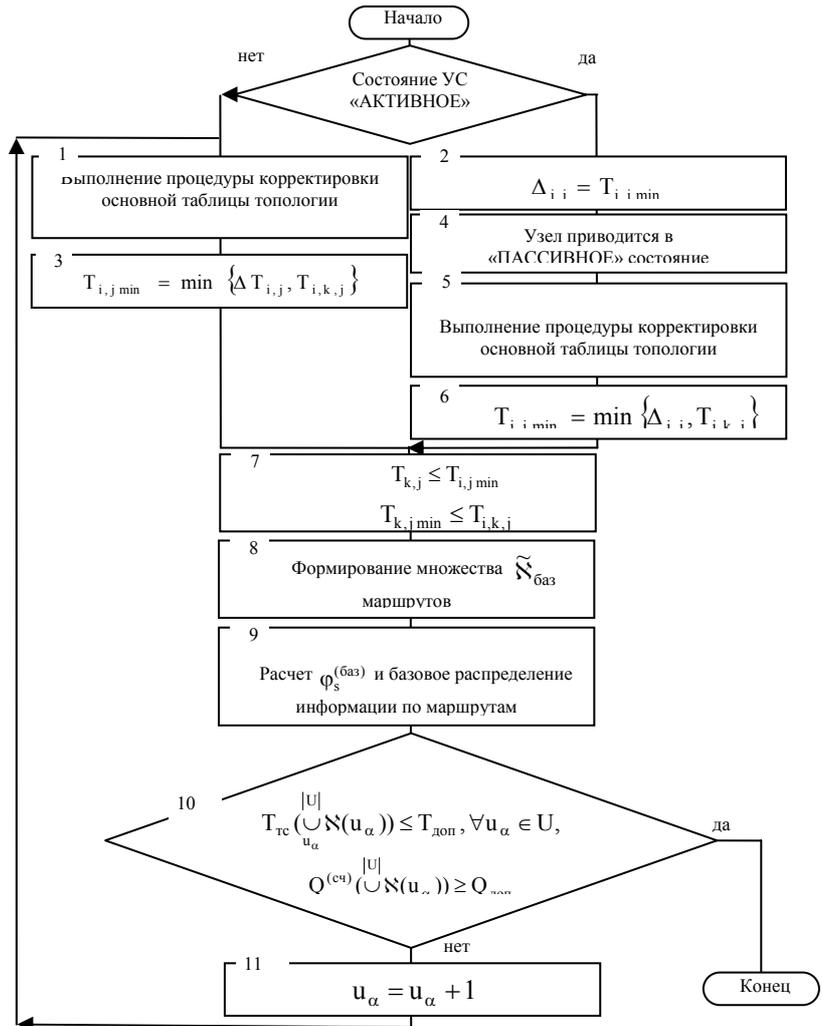


Рис. 1. Структурная схема алгоритма нахождения «узлов-преемников» на маршрутах к каждому адресату

маршруту, в соответствии с которыми осуществляется базовый вариант передачи информации. При обеспечении требований своевременности передачи цифровой информации с помощью найденного множества маршрутов принимается решение о его использовании для передачи. В противном случае итерационно осуществляется «наращивание» множества $\mathcal{N}_{\text{баз}}$ путем увеличения используемых для передачи уровней иерархии u_α дерева допустимых маршрутов. Конечным результатом выполнения алгоритма нахождения «узлов-преемников» на маршрутах к каждому адресату является формирование множества $\mathcal{N}_{\text{баз}}$ с минимально необходимым числом u_α уровней иерархии дерева выбранных маршрутов, исключаящих «зацикливание» информации.

На рис. 2 приведені графіки залежності часу Δt затримки від часу при статическому і флюктуаційно змінюючомуся трафіку для різних алгоритмів пошуку множини шляхів передачі інформації (алгоритма Галлагера (крива 1) [4], модифікованого дистанційно-векторного алгоритма MDVA (крива 2) [5] і запропонованого алгоритму знаходження «вузлів-приймачів» на маршрутах до кожного адресату (крива 3)).

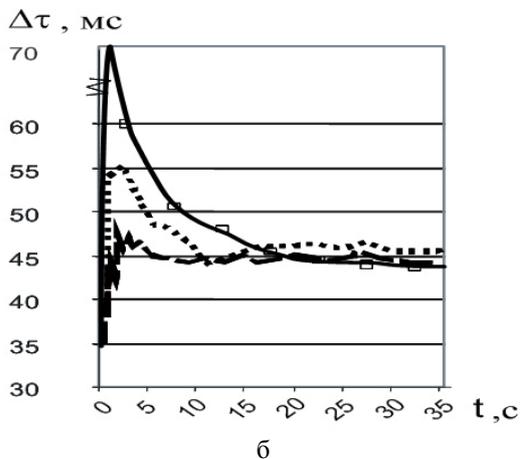
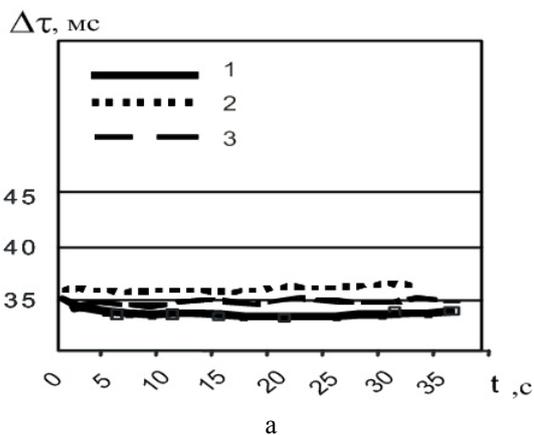


Рис. 2. Час затримки інформації:
а – при статическому трафіку $\lambda = 3$ Мбіт/с,
б – при флюктуації трафіку від 3 до 10 Мбіт/с

Видно, що запропонований спосіб більш ефективний з точки зору мінімізації інтервала стабілізації часу затримки пакетів при різких флюктуаціях трафіку (алгоритм знаходження «вузлів-приймачів» на маршрутах до кожного адресату «стабілізується» за 7 с, алгоритм Галлагера – за 30 с, модифікований дистанційно-векторний алгоритм MDVA – за 15 с) і тим самим, являється

більш практично прийнятним в реальних умовах динамічно змінюючоїся високої інтенсивності обміну інформацією.

Висновок

Таким чином, сукупність алгоритмів побудови попередньої топології і знаходження «вузлів-приймачів» на маршрутах до кожного адресату лежить в основі способу визначення базової множини шляхів передачі інформації. Запропонований спосіб при відсутності флюктуацій трафіку порівняємо з відомими [1, 4, 6] по часу Δt затримки передачі інформації (рис. 2, а) і в декілька раз стає більш ефективним при різких флюктуаціях входного трафіку (рис. 2, б).

Література

1. Королев А.В., Кучук Г.А., Пашнев А.А. Адаптивна маршрутизація в корпоративних мережах. – Х.: ХВУ, 2003. – 224 с.
2. Майніка Э. Алгоритми оптимізації на мережах і графах: Пер. з англ. / Під ред. Е.К. Масловського. – М.: Мир, 1981. – 321 с.
3. Кучук Г.А., Гахов Р.П., Пашнев А.А. Управління ресурсами інфотелекомунікацій. – М.: Фізматліт, 2006. – 220 с.
4. Gallager R. A Minimum Delay Routing Algorithm Using Distributed Computation // IEEE Trans. on Commun. – 1977. – V. COM-25, N 1. – P. 73-84.
5. Vutukury S., Garcia-Luna-Aceves J. J. MDVA: A Distance-Vector Multipath Routing Protocol // Proc. IEEE INFOCOM. – Anchorage, 2001. – P. 557-564.
6. Vutukury S., Garcia-Luna-Aceves J.J. MPATH: a loop-free multipath routing algorithm // Elsevier Journal of Microprocessors and Microsystems. – 2001. – 24 (6). – P. 319-327.

Поступила в редакцію 25.02.2008

Рецензент: д-р техн. наук А.В. Лемешко, Харківський університет Воздушних Сил ім. І. Кожедуба, Харків.