

## **ОПТИМАЛЬНАЯ РЕМАРШРУТИЗАЦИЯ ПОТОКОВ В КОМПЬЮТЕРНЫХ СЕТЯХ С ТЕХНОЛОГИЕЙ MPLS**

**Е.Ю. ЗАЙЧЕНКО**

Рассмотрена задача оперативного управления трафиками разных категорий в сетях с технологией MPLS. Данная задача сформулирована как задача ремаршрутизации потоков разных классов сервиса при отказах каналов и узлов сети и обеспечении заданного качества обслуживания. Построена математическая модель задачи и разработан ее алгоритм, позволяющий оперативно определить новые маршруты требований, получивших отказ в обслуживании и перераспределить их по новым маршрутам так, чтобы обеспечить максимальную величину передаваемого скорректированного потока. Для оценки эффективности предложенного алгоритма оперативного управления трафиком и ремаршрутизации потоков в сети были проведены экспериментальные исследования. Решена задача оптимального выбора пропускных способностей и распределения потоков. Смоделированы отказовые состояния и осуществлена ремаршрутизация потоков по предложенному алгоритму, определено новое распределение потоков и общая величина скорректированного потока по каждому классу сервиса. Анализ результатов подтвердил, что применение оперативного управления трафиком в виде ремаршрутизации позволяет существенно повысить общую пропускную способность сети и величину передаваемого потока сети в случае отказов элементов сети.

### **ВВЕДЕНИЕ**

В последние годы задачи разработки моделей и методов для оптимального управления трафиком в сетях с перспективной технологией многопротокольной коммутации меток (MPLS) представляют большой интерес в связи с широким практическим внедрением этой технологии в глобальных компьютерных сетях в развитых странах, в том числе и в Украине. Оперативное управление трафиками различных классов сервиса в сетях MPLS реализуется в маршрутизаторах сети, так называемых LSR (Label switching routers — маршрутизатор коммуникации по меткам), в которых реализованы дополнительные функции TE [1–2]. LSR с функциями инжиниринга трафика имеют две отдельные базы данных: традиционную — LSD (Link-state Database) и специальную — TED (Traffic Engineering Database) для реализации функций управления трафиком. Управление трафиком осуществляется путем ремаршрутизации потоков, передаваемых по виртуальным путям, так называемым LSP (Label switching pathes) [1–3] в случае перегрузок, вызванных отказами оборудования или каналов связи. Для практической реализации

этих функций оперативного управления необходима разработка эффективных методов и алгоритмов ремаршрутизации путем реконфигурации маршрутов в сети MPLS в случае отказов элементов сети.

**Цель работы** — разработка математической модели и алгоритма решения задачи оптимальной ремаршрутизации трафиков в сети MPLS и его исследования.

## ПОСТАНОВКА И МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЗАДАЧИ ОПТИМАЛЬНОЙ РЕМАРШРУТИЗАЦИИ ПОТОКОВ В СЕТИ

Как известно, важной особенностью сети с технологией MPLS является наличие различных классов сервиса (CoS), каждый из которых обслуживается в маршрутизаторах (LSR) с соответствующим приоритетом. Кроме того, вводятся показатели качества обслуживания потоков различных классов: среднее время доставки пакетов ( $T_{cp}^{(k)}$ ), вариация времени доставки пакетов, а также доля потерянных пакетов [1–3]. Поэтому соответствующая математическая модель задачи ремаршрутизации должна учитывать указанные особенности технологии MPLS.

Пусть задана сеть MPLS со структурой  $G = \{X, E\}$ ,  $X = \{x_j\}$ ,  $j = \overline{1, n}$  — множество узлов сети (УС),  $E = \{r, s\}$  — множество каналов связи (КС) сети, заданы также пропускные способности (ПС) всех КС  $\mu_{rs}$ ,  $(r, s) \in E$ , и матрица требований  $H(k) = \|h_{ij}(k)\|$   $i, j = \overline{1, n}$ ,  $h_{ij}(k)$  — интенсивность потока  $k$ -го класса сервиса, который необходимо передавать из узла  $i$  в  $j$  (Кбит/с), распределение потоков всех классов  $F(k) = [f_{rs}(k)]$ , где  $f_{rs}(k)$  — величина потока класса  $k$ , передаваемого по КС  $(r, s)$  и соответствующего матрице  $H_{\Sigma}(k)$ .

При этом, общая величина потока всех классов в КС  $(r, s)$  будет равна:

$$f_{rs} = \sum_{k=1}^K f_{rs}(k). \quad (1)$$

В случае, когда обслуживание потоков различных классов  $k$  происходит с относительными приоритетами  $\rho_k$ , убывающими с ростом номера класса, т.е.  $\rho_1 > \rho_2 > \dots > \rho_k$ ,  $k = 1, K$ , то средняя задержка пакетов  $k$ -го класса задается следующим выражением [5]:

$$T_{cp,k} = \frac{1}{H_{\Sigma}^{(k)}} \sum_{(r,s) \in E} \frac{f_{rs}^{(k)} \sum_{i=1}^k f_{rs}^{(i)}}{\left( \mu_{rs} - \sum_{i=1}^{k-1} f_{rs}^{(i)} \right) \left( \mu_{rs} - \sum_{i=1}^k f_{rs}^{(i)} \right)}. \quad (2)$$

Введены также ограничения на значения показателя качества (QoS) для всех классов в виде:

$$T_{cp}^{(k)} \leq T_{зад,k}, \quad (3)$$

где  $T_{\text{cp}}^{(k)}$  — средняя задержка пакетов  $k$ -го класса (сек.);  $T_{\text{зад},k}$  — ограничение на эту величину.

Известны также маршруты коммутации меток LSP  $\{P_{ij}(k)\}$  для каждого соединения (пары)  $(i, j)$ , которые устанавливаются с помощью протокола RSVP или SNMP [1–2].

Допустим, что отказал КС  $(r_i, s_i)$  — отказовое состояние обозначим как  $z_i$ . Требуется реконфигурировать все маршруты отказавшего соединения КС  $(r_i, s_i)$  таким образом, чтобы в максимальной степени удовлетворить соответствующие требования, получившие отказ в обслуживании при сохранении остальных соединений по объему трафика и заданному качеству QoS —  $T_{\text{cp}}$ . Назовём эту задачу *задачей оптимальной ремаршрутизации* сети MPLS при отказах.

Математическая модель данной задачи имеет следующий вид [5].

Требуется найти такое распределение потоков  $[f_{rs}(k)]$ , при котором обеспечивается:

$$H_{\Sigma} = \sum_{(i,j):(r_i,s_j) \in P_{ij}} h_{ij}^{(\text{корр})} \rightarrow \max \quad (4)$$

при условиях

$$T_{\text{cp}}(F_{\text{корр}}^{(k)}) \leq T_{\text{зад}}, \quad k = \overline{1, K}, \quad (5)$$

где  $F_{\text{корр}}^{(k)}$  — скорректированный поток  $k$ -го класса сервиса после реконфигурации.

## ОПИСАНИЕ АЛГОРИТМА ОПТИМАЛЬНОЙ РЕМАРШРУТИЗАЦИИ СЕТИ MPLS

Метод основан на том свойстве оптимального решения, что поток  $k$ -го класса сервиса в силу формулы (2) не влияет на задержку потоков более высокого уровня приоритета, а именно, классов:  $1, 2, \dots, k-1$ .

Алгоритм состоит из 2-х этапов [5].

На первом этапе определяются все требования (соединения)  $(i, j)$ , которые использовали КС  $(r_i, s_i)$ , и отключаются временно от сети и пересчитываются потоки в КС  $F^{(k)} = [f_{rs}^{(k)}]$ ,  $(r, s) \in E$ .

На втором — определяются резервы по ПС всех КС и оптимальным образом перераспределяются потоки отказанных требований так, чтобы обеспечить достижение критерия  $H_{\Sigma} \rightarrow \max$  (назовём их отказовыми требованиями).

### 1-й этап.

1. Находим все требования  $(i, j)$ , проходившие через отказанные соединения КС  $(r_i, s_i)$ . Обозначим их множество как  $P_{r_i, s_i} = \{(i, j) : (r_i, s_i) \in P_{ij}\}$ .

2. Временно отключаем передачу информации для требований множества  $P_{r_i, s_i}$  и вычисляем новые значения потоков —  $F^H(k) = [f_{rs}^H(k)]$ :

$$f_{rs}^H = \begin{cases} f_{rs} - \sum_{(i,j):(r_i, s_i) \in \Pi_{ij}} h_{ij}, & \text{где } (i, j) \in P_{r_i, s_i}, \\ f_{rs}, & \text{в остальных случаях.} \end{cases}$$

Это выполняется следующим образом:

- Находим первое требование  $(i_1, j_1) \in P_{r_i, s_i}$ .
- Полагаем  $h_{i_1, j_1} = 0$  и вычисляем новое распределение потоков:

$$f_{rs}^H(k) = \begin{cases} f_{rs}(k) - h_{i_1, j_1}, & \text{если } (r, s) \in \Pi_{i_1, j_1}, \\ f_{rs}(k), & \text{в остальных случаях,} \end{cases} \quad (6)$$

где  $\Pi_{i_1, j_1}$  — маршрут передачи потока требования  $(i_1, j_1)$ .

• Проверка условия:  $P_{r_i, s_i} \setminus (i_1, j_1) \neq \emptyset$ . Если ДА, то на шаг 1 и повторяем шаги 1–3 до исчерпания множества  $P_{r_i, s_i}$ . В результате получим новое распределение потоков всех классов  $F^H(k) = |f_{rs}^H(k)|$ , включающее потоки только от требований, не получивших отказа в обслуживании  $(i, j) \setminus P_{r_i, s_i}$ .

- Определяем резервы по ПС всех КС:

$$Q_{\text{рез}, r, s} = \mu_{rs} - \sum_{k=1}^K f_{rs}^H(k). \quad (7)$$

Переходим ко второму этапу.

### 2-й этап.

Для требований множества  $P_{r_i, s_i}$  находим новые маршруты (реконфигурируем) так, чтобы обеспечить выполнение условий:

$$T_{\text{ср}}(F_{\text{корр}}^{(k)}) \leq T_{\text{зад}}, \quad k = \overline{1, K}, \quad (8)$$

и при этом  $\sum_{(i,j) \in P_{r_i, s_i}} h_{ij}^{(\text{корр})} \rightarrow \max$ .

Второй этап состоит из  $k$  подэтапов, на каждом из которых осуществляем реконфигурацию маршрутов и переопределение потоков для  $k$ -го класса. При этом, с учетом свойств оптимального решения, перераспределение потоков отказавших требований различных классов осуществляется в порядке убывания приоритетов до полного исчерпания резерва ПС каналов и контроля выполнения ограничений (8).

### Подэтап 1.

1-я итерация.

1.  $k = 1$ . Сначала распределяем отказанные требования класса 1.

2. Находим условную метрику  $\frac{\partial T_{\text{ср}}}{\partial f_{rs}^{(1)}} | f_{rs}^H(1)$ .

3. Находим кратчайшие пути  $\Pi_{ij}^{\min}(1)$  для всех отказанных требований класса  $k = 1$ .

4. Выбираем первое требование  $(i_1, j_1) \in P_{r_i, s_i}$  такое, что  $l(\Pi_{i_1, j_1}^{\min}) = \min_{(i, j)} l(\Pi_{i, j}^{\min})$ .

5. Проверяем возможность передачи его в полном объеме по пути  $\Pi_{i_1, j_1}^{\min}$ :

$$h_{i_1, j_1} < Q_{\text{рез}}(\Pi_{i_1, j_1}^{\min}), \quad (9)$$

где  $Q_{\text{рез}}(\Pi_{i_1, j_1}^{\min})$  — свободная ПС маршрута  $\Pi_{i_1, j_1}^{\min}$ :

$$Q_{\text{рез}}(\Pi_{i_1, j_1}^{\min}) = \min_{(r, s) \in \Pi_{i_1, j_1}^{\min}} \{\mu_{rs} - f_{rs}\}. \quad (10)$$

Если условие (10) выполняется, то распределяем полностью поток требования  $h_{i_1, j_1}$  по маршруту  $\Pi_{i_1, j_1}^{\min}$  и находим скорректированное распределение потоков (РП):

$$f_{rs}^H(k) = \begin{cases} f_{rs}^H(1) + h_{i_1, j_1}, & \text{если } (r, s) \in \Pi_{i_1, j_1}^{\min}, \\ f_{rs}^H(1), & \text{в остальных случаях.} \end{cases} \quad (11)$$

Иначе переходим на шаг 6.

6. Полагаем  $h_{i_1, j_1}^{(a)} = Q_{\text{рез}}(\Pi_{i_1, j_1}^{\min}) - \Delta$ . Здесь  $h_{i_1, j_1}^{(a)}$  — доля требования  $h_{i_1, j_1}$ , передаваемая по маршруту  $\Pi_{i_1, j_1}^{\min}$ ,  $\Delta$  — некоторая заданная величина.

7. Находим скорректированное распределение потоков:

$$f_{rs}^{\text{корр}}(k) = \begin{cases} f_{rs}^H(1) + h_{i_1, j_1}^{(a)}, & \text{если } (r, s) \in \Pi_{i_1, j_1}^{\min}, \\ f_{rs}^H(1), & \text{в остальных случаях.} \end{cases} \quad (12)$$

Проверяем выполняется ли ограничение на  $T_{\text{ср}, 1}$ :

$$T_{\text{ср}}(F_{\text{корр}}^{(1)}) \leq T_{\text{зад}, 1}. \quad (13)$$

Если условие (13) выполняется, то  $P_{r_i, s_i}^H = P_{r_i, s_i} \setminus (i_1, j_1)$ , иначе на шаг 9.

8. Проверка условия:  $P_{r_i, s_i}^H \neq \emptyset$ . Если ДА, то переходим к следующей итерации и распределяем очередное требование класса  $k = 1$ . Иначе на шаг 9.

9. Конец первого подэтапа. Переходим ко второму подэтапу  $k = 2$ . На этом этапе перераспределяем отказанные требования класса  $k = 2$  так, чтобы не нарушалось условие:

$$T_{\text{ср}}(F_{\text{корр}}^{(2)}) \leq T_{\text{зад}, 2}. \quad (14)$$

**Подэтап 2** аналогичен подэтапу 1. Поэтому его описание можно опустить. Отказанные требования КС 2 перераспределяем до тех пор, пока либо они не будут распределены полностью, либо начнет нарушаться условие (8).

Тогда конец подэтапа 2 и переходим к подэтапу 3.

**Подэтап  $k$ .** На этом подэтапе находим реконфигурированные маршруты для отказанных требований  $k$ -го класса:  $(i, j) \in P_{r_i, s_i}^{(k)}$ .

Пусть  $F^0(k) = [f_{rs}^0(k)]$  — вектор многопродуктового потока  $k$ -го класса, после выполнения  $(k-1)$ -го подэтапа и при этом  $T_{cp}(F(k) | F(1), \dots, F(k-1)) < T_{зад, k}$ .

1-я итерация.

1. Находим условную метрику  $l_{rs}(k) = \frac{\partial T_{cp}}{\partial f_{rs}(k)} | f_{rs}^0(k)$ .

2. Находим кратчайшие пути в метрике  $l_{rs}(k)$  для всех отказанных требований  $k$ -го класса —  $\Pi_{ij}^{\min}(k)$ .

3. Ищем такое требование  $(i_k, j_k) \in P_{r_i, s_i}^{(k)}$  для которого  $l(\Pi_{i_k, j_k}^{\min}(k)) = \min_{(i, j) \in P_{r_i, s_i}^{(k)}} l(\Pi_{i, j}^{\min}(k))$ .

4. Определяем резерв по ПС маршрута  $\Pi_{i_k, j_k}^{\min}$ :

$$Q_{рез}(\Pi_{i_k, j_k}^{\min}(k)) = \min_{(r, s) \in \Pi_{i_k, j_k}^{\min}} ((\mu_{rs} - f_{rs}) - \varepsilon), \quad (15)$$

где  $f_{rs} = \sum_{i=1}^k f_{rs}^{(i)}$ .

5. Проверка условия:

$$h_{i_k, j_k} < Q_{рез}(\Pi_{i_k, j_k}^{\min}(k)). \quad (16)$$

Если ДА, то на шаг 6, иначе на 7.

6. Распределяем поток от требования  $h_{i_k, j_k}$  в полном объеме по маршруту  $\Pi_{i_k, j_k}^{\min}$  и находим новое распределение потоков (РП):

$$f_{rs}^H(k) = \begin{cases} f_{rs}(k) + h_{i_k, j_k}, & \text{если } (r, s) \in \Pi_{i_k, j_k}^{\min}, \\ f_{rs}(k), & \text{в остальных случаях.} \end{cases}$$

Переход на шаг 9.

7. Полагаем  $h_{i_k, j_k}^{(a)} = Q_{рез}(\Pi_{i_k, j_k}^{\min}) - \Delta$ .

8. Распределяем поток от требования  $(i_k, j_k)$  величиной  $h_{i_k, j_k}^{(a)}$  по маршруту  $\Pi_{i_k, j_k}^{\min}$  и находим новое РП.

$$f_{rs}^H(k) = \begin{cases} f_{rs}(k) + h_{i_k, j_k}^{(a)}, & \text{если } (r, s) \in \Pi_{i_k, j_k}^{\min}, \\ f_{rs}(k), & \text{в остальных случаях.} \end{cases} \quad (17)$$

9. Проверка условия:  $T_{\text{cp}}(F(k) | F(1), \dots, F(k-1)) < T_{\text{зад},k}$ . Если да, то на шаг 10, или уменьшаем величину передаваемого потока требования  $(i_k, j_k)$  до значения  $h_{i_k, j_k}^*$  такого что:  $T_{\text{cp}}(F^*(k) | F(1), \dots, F(k-1)) = T_{\text{зад},k}$  и конец подэтапа  $k$ .

10. Полагаем  $P_{r_i, s_i}^{(k)} = P_{r_i, s_i}^k \setminus (i_k, j_k)$ .

Проверка условия:  $P_{r_i, s_i}^k = \emptyset$ . Если ДА, то конец подэтапа  $k$ , иначе на шаг 1 следующей итерации.

Последовательность итераций  $k$ -го подэтапа повторяется до тех пор, пока не выполняется одно из следующих условий:

- а)  $T_{\text{cp}}(F(k) | F(1), \dots, F(k-1)) = T_{\text{зад},k}$ ;
- б)  $T_{\text{cp}}(F(k+s) | F(1), \dots, F(k)) = T_{\text{зад},k+s}$ ,  $1 \leq s \leq K-k$ ;
- в)  $P_{r_i, s_i}^{(k)} = \emptyset$  и  $T_{\text{cp}}(F(k) | F(1), \dots, F(k-1)) \leq T_{\text{зад},k}$ .

В случае (а) переходим на  $k+1$  подэтап, и ищем распределение потоков  $k+1$ -го класса приоритета.

В случае (б), когда вышло на границу ограничение для менее приоритетного потока  $F(k+s)$  и  $k+s < K$ , переходим к распределению потока класса  $k+s+1$ , поскольку распределение потоков класса  $F(k+s+1)$  не влияет на среднюю задержку для более приоритетных потоков  $r < k+s+1$ .

В случае (в) переходим на  $k+1$  подэтап, как и в случае (а).

Последовательность выполняемых подэтапов заканчивается либо полным перераспределением всех отказанных требований, что мало вероятно, либо при выходе на границу по некоторым из ограничений, включая ограничение для наименее приоритетного потока, то есть:

$$\exists k, T_{\text{cp}}(F_{\text{корр}}^{(k)}) \geq T_{\text{зад},k}. \quad (18)$$

Это означает полное исчерпание всех свободных ресурсов (свободной полосы) каналов связи.

В результате работы алгоритма находятся реконфигурированные пути (LSP) для соединений  $(i, j)$  получивших отказ в обслуживании из-за отказа соответствующего КС или УС. Естественно, что в силу ограниченной ПС сети при этом некоторые требования обслуживаться не будут. При этом, учитывая очередность реконфигурации, отказ в обслуживании получают наименее приоритетные соединения, а величина общего потока, передаваемого в сети после реконфигурации — максимальна.

## ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

Для оценки эффективности предложенного алгоритма оперативного управления трафиком и ремаршрутизации потоков в сети с технологией MPLS были проведены его экспериментальные исследования. Все эксперименты проводились с сетью MPLS, структура которой приведена на рисунке.

Сеть состояла из 25 узлов, 39 каналов связи и вводились 3 типа класса сервиса. По каждому классу задавалась матрица требований  $H(k)$ ,  $k = 1, 3$ . Введены ограничения на среднюю доставку пакетов по сети (QoS) для различных классов:

$$T_{\text{зад},1} = 0,1 \text{ сек}, T_{\text{зад},2} = 0,5 \text{ сек}, T_{\text{зад},3} = 1 \text{ сек}.$$

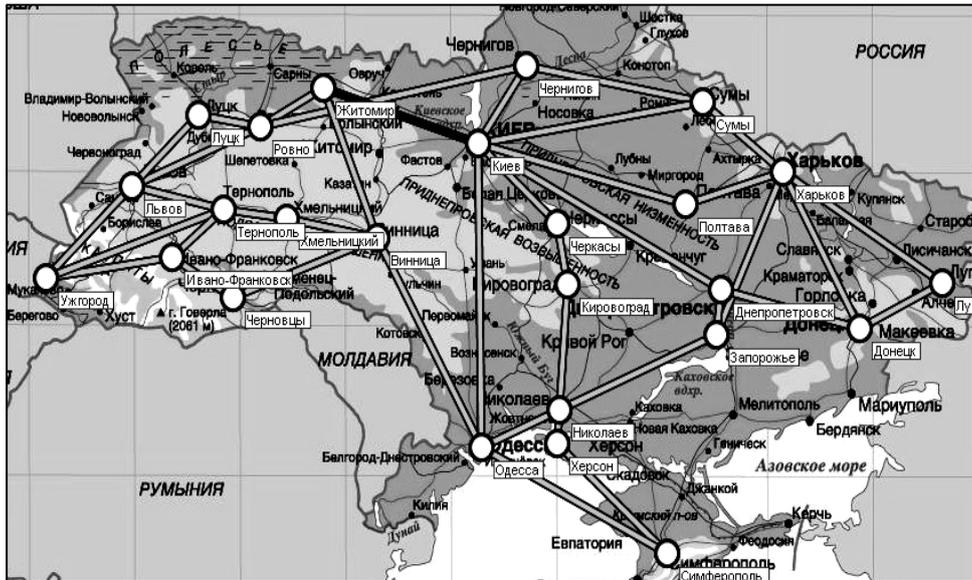


Рисунок. Топология сети

При этих ограничениях решалась задача оптимального выбора пропускных способностей и распределения потоков (ВПСРП) [5–6], в результате были определены ПС всех каналов связи и найдено оптимальное распределение потоков всех классов  $F^{(k)} = [f_{rs}^{(k)}]$ , а также была определена общая величина потока, передаваемого в сети по каждому классу  $H_{\Sigma}(k) = \sum_{(i,j) \in \Pi_{ij}}$

$k = \overline{1,3}$  при указанных ограничениях на среднюю задержку. Далее моделировались отказовые состояния: отказы одного канала связи (КС), двух каналов связи и трех КС. При этих состояниях определялись новые значения потоков —  $F^H(k) = [f_{rs}^H(k)]$  и фактическая величина передаваемого потока класса  $k$  в случае действия отказов  $H_{\Sigma}^{\Phi}(k)$ . Далее осуществлялась ремаршрутизация потоков по предложенному алгоритму и определялось новое распределение потоков  $F_{\text{корр}}^{(k)}$  и общая величина скорректированного потока по каждому классу сервиса  $H_{\Sigma, \text{корр}}^{\Phi}$  при ограничениях  $T_{\text{ср}}(F_{\text{корр}}^{(k)}) \leq T_{\text{зад},k}$ .

Общая величина передаваемого потоков сети  $H_{\Sigma}^{\Phi}(k)$  и  $H_{\Sigma, \text{корр}}^{\Phi}$  до и после ремаршрутизации соответственно в процентах от величины номинального потока в безотказовом состоянии  $H_{\Sigma}^{(0)}(1), H_{\Sigma}^{(0)}(2), \dots, H_{\Sigma}^{(0)}(k)$  приводится в таблице для различных классов отказовых состояний.

**Таблица.** Зависимость величины передаваемого потока для различных отказов состояний в процентах от  $H_{\Sigma}^{(0)}(k)$

Тип отказов состояний	Класс сервиса $k=1$		Класс сервиса $k=2$		Класс сервиса $k=3$	
	$H_{\Sigma}^{\Phi}(k)$	$H_{\Sigma, \text{копп}}^{\Phi}(k)$	$H_{\Sigma}^{\Phi}(k)$	$H_{\Sigma, \text{копп}}^{\Phi}(k)$	$H_{\Sigma}^{\Phi}(k)$	$H_{\Sigma, \text{копп}}^{\Phi}(k)$
Отказ 1КС	94,7	97,2	93	94,5	90	90,5
Отказ 2КС	90,5	94,3	88	89,5	85	85,5
Отказ 3 КС	84,3	88,2	82	83	74	74

Анализ приведенных результатов свидетельствует о том, что применение оперативного управления трафиком в виде ремаршрутизации позволяет существенно повысить общую пропускную способность сети и величину передаваемого потока сети в случае отказов элементов сети. При этом такое увеличение относится к потокам высокого приоритета (класс 1 и 2), что вполне объяснимо, т.к. они ремаршрутизируются в первую очередь и забирают всю резервную пропускную способность каналов связи.

## ВЫВОДЫ

- В статье рассмотрена задача ремаршрутизации потоков информации разных классов сервиса в сетях с технологией MPLS.
- Построена математическая модель данной задачи и предложен алгоритм ее решения, использующий свойства оптимального потока.
- Проведены экспериментальные исследования и определена эффективность предложенного алгоритма ремаршрутизации

## ЛИТЕРАТУРА

1. *Thomas D. Nadeua. MPLS Network Management: MIBs, Tools and Techniques.* — NY.: Morgan Kaufmann, 2003. — 529 p.
2. *Request for Comments 5151. A. Farrel, Ed, A. Ayyangar, JP. Vasseur. Inter-Domain MPLS and GMPLS Traffic Engineering. Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions Cisco Systems, Inc. February, 2008.* — <https://tools.ietf.org/html/rfc5151>.
3. *Vivek A. Advanced MPLS design and implementation.* — Indianapolis: Cisco Press, 2002. — 320 p.
4. *Гольдштейн А.Б., Гольдштейн Б.С. Технология и протоколы MPLS.* — СПб: БХВ, 2005. — 304 с.
5. *Вивек О. Структура и реализация современной технологии MPLS. Пер. С англ.* — СПб.: Изд. дом «Вильямс», 2004. — 480 с.
6. *Зайченко Е.Ю., Зайченко Ю.П. Сети с технологией MPLS: моделирование, анализ и оптимизация.* — К.: Политехника, 2008. — 240 с.
7. *Zaychenko Y., Zaychenko H. New Generation Computer Networks Survivability Analysis and Optimization // In book «Distributed Computer and Communication Networks».* — Bern: Springer International Publishing Switzerland, 2014. — P. 73–81.

Поступила 27.02.2014