



здесь  $C_{p1}$ ,  $C_{p2}$  – соответственно изобарные тепловые емкости влажного и сухого газов;  $\Delta\tilde{T}$  - интегральный эффект Жоул-Томсона;  $C$  – удельный выход конденсата.

### Литература

1. Бекиров Т.М., Ланчаков Г.А. Технология обработки газа и конденсата. - М.: ООО «Недра-Бизнесцентр», 1999. -596 с.
2. Жила Н.П., Ключева Э.С. Методы очистки гликолей от тяжелых углеводородов и продуктов деструкции // Обз. информ. Сер. Подготовка и переработка газа и газового конденсата. -М.: ВНИИЭгазпром, 1990. -40 с.
3. Зиберт Г.К., Седых А.Д., Кащицкий Ю.А. и др. Подготовка и переработка углеводородных газов и конденсата. Технология и оборудование: Справочное пособие. - М.: ООО «Недра-Бизнесцентр», 2001. -316 с.

А.М. Сухов, Н.А. Альзинская

## ПРИМЕНЕНИЕ КЛАСТЕРНОГО ПОДХОДА ПРИ ОРГАНИЗАЦИИ МАРШРУТИЗАЦИИ В ИЕРАРХИЧЕСКИХ СИСТЕМАХ

(Самарский университет)

### 1. Введение

Маршрутизация информации - это самая простая и, пожалуй, самая сложная функция, которую выполняют сети. Обычная мудрость гласит, что для поиска путей к пунктам назначения через сложный сетевой лабиринт узлы должны коллективно находить текущее состояние топологии сети путем обмена информацией о статусе их соединений с другими узлами. Эти коммуникационные издержки считаются одними из самых серьезных ограничений масштабирования наших первичных коммуникационных технологий сегодня, в том числе Интернета и новых беспроводных и сенсорных сетей.

Однако многие сети в природе могут каким-то образом «маршрутизировать трафик» эффективно. То есть узлы в этих сетях могут эффективно найти намеченные цели связи, даже если они не обладают глобальным представлением о системе. Эксперимент Милгрэма 1969 года [1] показал классическую демонстрацию этого эффекта. Гораздо позже Клейнберг предложил первое популярное объяснение этого удивительного эффекта [2]. В его модели каждый узел, помимо того, что он является частью графа, представляющего топологию глобальной сети, находится в координатном пространстве – сетке, внедренной в евклидову плоскость. Каждый узел знает: 1) его координаты; 2) координаты его соседей; и 3) координаты адресата, записанного на пакете. Учитывая эти три части информации, узел жадно выбирает ближайший узел к месту назначения из своих ближайших узлов-соседей.



Данный подход позволяет решить одну из серьезных проблем современных сетей – это увеличение таблиц маршрутизации. Так как при использовании метода жадного продвижения таблицы маршрутизации не требуются.

Наша работа представляет технологию маршрутизации на основе кластерного подхода, которая дополняет иерархическую маршрутизацию, основанную на методе жадного продвижения. Представим иерархическую систему как некий набор узлов с возможностью организации связи между ними. Для этой системы можно построить иерархический граф связанности с выделением центрального и дочерних узлов. Все остальные связи между узлами при таком построении игнорируются.

На основании этого подхода было предложено строить различные типы адресации, опираясь на гиперболическое преобразование [3]. Маршрутизация между узлами в таких пространствах строится на основании принципа жадного продвижения [4]. В качестве расстояния выбираются разные функции, от расстояния в гиперболическом пространстве, до числа переходов по иерархическому графу.

Для того чтобы задействовать переходы между различными ветками древовидной иерархической структуры и тем самым снизить нагрузку на первые четыре уровня иерархии, мы предлагаем применить кластерный подход.

## 2. Основные проблемы адресации

Теперь необходимо попытаться найти битовую форму записи адреса, оценить его длину и размер адресного пространства для полного покрытия потребностей технологии Интернета вещей.

Рассмотрим форму записи числа  $m_1, m_2, \dots, m_i$  в виде  $m_k$  единиц с нулем в качестве разделительного знака. Данная запись требует  $N$  бит (1), что является очень неэкономичной формой записи.

$$N = m_1 + m_2 + \dots + m_i + i \quad (1)$$

Если попытаться сократить длину адреса и записать составные части адреса  $m_k$  в двоичной системе. Тогда для записи каждого числа адреса  $m_k$  потребуется  $N_k$  бит (2) с округлением до ближайшего целого числа сверху.

$$N_k = \lceil \log_2 m_k \rceil + 1 \quad (2)$$

Однако, этот подход не учитывает наличия разделительных знаков между различными частями адресной иерархии. Если бы для каждой части адреса  $m_k$  выделялось бы фиксированное число бит, как в адресном пространстве IP протокола, то введение специальных символов для разделителей бы не требовалось. Так как для уменьшения общей длины адреса используется переменное число бит для описания каждой части адреса, то требуется передавать точное место начала каждой иерархии.

В привычной нам IP системе маршрутизации эту роль играет параметр «маска». Этот параметр показывает, сколько бит в адресе не изменяется, то есть задает родительский узел и описывает соответствующий кластер. В нашем случае мы можем передавать число бит  $p_k$ , которое равно числу бит  $N_k$  соответствующего значения иерархического адреса  $m_k$ , где  $p_k = \lceil \log_2 N_k \rceil + 1$ .



Теперь оценим битовую длину адреса, которая потребуется для нужд технологий Интернета вещей. Во-первых, необходимо оценить глубину адресного пространства. Для этого разделим адресное пространство на 2 части – глобальную и локальную. Глобальные адреса служат для привязки локального сегмента к глобальной сети. Сейчас эту роль в основном выполняют IPv4 адреса, но так как их уже не хватает, то для расширения адресного пространства применяются технологии NAT.

Во-вторых, оценим глубину иерархического адресного пространства исходя из факта, что число переходов (hop) команды ping ограничено 30. Соответствующий выбор центрального узла может сократить это число вдвое. В качестве центрального узла целесообразно выбрать самую популярную точку обмена трафиком, расположенную в Нью-Йорке, США. При этом можно ограничиться 15 уровнями иерархии, чтобы дойти до каждого домохозяйства или другого локального узла, к которому будут присоединяться локальные сети для удовлетворения местных потребностей.

При таком подходе максимальная длина адреса без разделения на уровни иерархии оценивается в 60 бит, а с учетом разделения – это ещё порядка 40 бит. Это больше, чем существующее адресное пространство IPv4. Однако средняя длина адреса будет значительно короче. Абсолютное число адресов будет иметь длину до 32 бит. Еще один огромный плюс подобной маршрутизации – это отсутствие огромных маршрутизирующих таблиц.

### 3. Основные положения межкластерной маршрутизации

Под *кластером* будем понимать совокупность устройств и всех их дочерних узлов. На иерархическом графе это отдельная скелетная ветвь, начинающаяся с родительского узла. В наших обозначениях кластер можно записывать как родительский узел  $m_1, m_2, \dots, m_i$ .

В стандартном методе жадного продвижения мы всегда движемся вверх по иерархии, в то время как кластерный метод основан на поиске возможных путей через дочерние узлы. Вопрос в том, как можно найти такие переходы без значительного увеличения вычислительной сложности алгоритма и времени установления соединения.

Тот вариант присвоения адресов, что описан нами в статье [5], накладывает сильные ограничения на межкластерные переходы. На первом этапе присвоения адресов мы нарезали окрестности относительно начала маршрута, а на втором построили сами кластеры. Поэтому межкластерный переход может происходить между узлами одной окрестности или соседних. Причем, если узел лежит в основании «старшего» кластера, то он может быть связан с устройством младшего кластера той же окрестности или старшей окрестности. Устройства младшего кластера могут быть связаны с устройствами старшего кластера той же окрестности или с понижением номера окрестности.

Каждому межкластерному переходу можно сопоставить его уровень. Уровень межкластерного перехода будем определять как число участков маршрута по иерархической структуре между двумя соседними узлами. То есть для



двух узлов  $m_1, \dots, m_k, m_{k+1}, \dots, m_i$  и  $m_1, \dots, m_k, n_{k+1}, \dots, n_l$  уровень межкластерного перехода  $Y$  равен

$$Y = i + l - 2k \quad (3)$$

Заметим при этом, что  $i$  и  $l$  не могут различаться больше, чем на единицу, как это было показано выше.

Особое внимание должно быть уделено поиску межкластерных переходов на первых четырех уровнях иерархии, как это показано на Рис. 1. Эти уровни являются наиболее загруженными и именно для них необходимо использовать альтернативные межкластерные маршруты. Поэтому необходимо специальным образом сохранять информацию о переходах, у которых  $k$  из уравнения (3) меньше или равно 3.

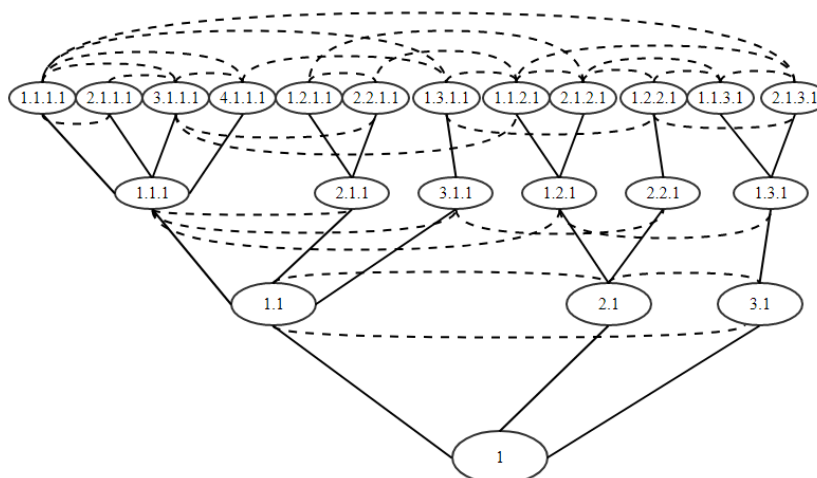


Рисунок 1– Структура межкластерных переходов

Можно предложить следующий алгоритм учета таких межкластерных переходов. Каждый узел  $m_1, m_2, \dots, m_i$  опрашивает своих соседей об адресации. Для каждого соседа рассчитывается уровень межкластерного перехода и находится общий родительский кластер  $m_1, \dots, m_k$  и число  $k$ . Если число  $k \leq 2$ , то информацию о таком переходе и его источнике следует хранить на родительских узлах кластеров  $m_1, \dots, m_{k+1}$ .

Таким образом, все узлы первых трех уровней иерархии будут хранить информацию о возможных межкластерных переходах. На основании этой информации можно будет построить таблицу маршрутизации. Эта таблица позволит обходить наиболее загруженные верхние уровни иерархии. Размер такой таблицы будет очень сильно ограничен из-за ограниченного числа узлов первых трех уровней иерархии.

#### 4. Заключение

В настоящей работе рассматривается маршрутизация на основе принципа жадного продвижения в гиперболическом пространстве. Для предотвращения переполнения каналов старших уровней иерархии предложено использовать кластерный подход. Дано определение кластера, описаны направления межкластерных переходов. Для каждого узла и его соседей предложено находить метрику межкластерного перехода. Если эта метрика описывает межкластерный



переход на первых трех уровнях иерархической адресации, то на родительском узле кластера ведется специальный учет таких переходов и их источников. На основании межкластерных переходов для первых трех уровней иерархии строится специальная таблица маршрутизации, которая позволяет не перегружать древовидные иерархические каналы.

### Литература

1. Milgram, S. An experimental study of the small world problem/ J. Travers and S. Milgram// *Sociometry*. – 1969. – P. 425–443.
2. Kleinberg R. Geographic routing using hyperbolic space //INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. – IEEE, 2007. – С. 1902-1909.
3. Cvetkovski, A. Hyperbolic embedding and routing for dynamic graphs / A. Cvetkovski and M. Crovella // *IEEE INFOCOM*. – 2009. – P. 1647-1655.
4. Papadopoulos, F. Network mapping by replaying hyperbolic growth / F. Papadopoulos, C. Psomas, D. Krioukov. // *IEEE/ACM Transactions on Networking (TON)*. – 2015. – P. 198-211.
5. Sukhov, A. M. Greedy forwarding for hyperbolic space in MANET / S. V. Timofeeva, D. M. Sarkisian, A. M. Sukhov, S. A. Zuev // *Telecommunication Forum (TELFOR)*. – 2017. – P. 1-4.

М. С. Светлов<sup>1</sup>, А. А. Львов<sup>2</sup>, Д. В. Кленов<sup>2</sup>, Н. С. Вагарина<sup>2</sup>

### АЛГОРИТМЫ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ДЛЯ КОДА НА ОСНОВЕ КОДОВОГО СИГНАЛЬНОГО ПРИЗНАКА В ИНФОРМАЦИОННЫХ КАНАЛАХ СО СТИРАНИЕМ

(<sup>1</sup>Институт проблем точной механики и управления РАН,

<sup>2</sup>Саратовский государственный технический университет  
имени Гагарина Ю.А.)

#### I. Введение

Одной из наиболее актуальных задач синтеза информационных каналов (ИК) является обеспечение требуемого уровня их информационной надежности в условиях действия случайных импульсных помех большой интенсивности, когда  $i_{pn} = f_{pn}/f_c \geq 3$  ( $i_{pn}$  — интенсивность случайной помехи;  $f_{pn}$ ,  $f_c$  — частоты импульсной помехи и кода, соответственно). ИК состоит из устройств кодирования (КУ), декодирования (ДКУ) и канала связи (КС). Под действием помех возможно искажение передаваемых по КС кодовых комбинаций (КК). Существует два возможных вида ошибок: трансформации и стирания. При ошибках трансформации одна КК переходит в другую возможную КК. При ошибках стирания образуется КК, недопустимая с точки зрения выбранного кода. В зависимости от типа возможных ошибок, ИК делятся на каналы с трансформациями, каналы со стираниями и каналы общего вида, в которых