# Реализация виртуальной сети на плоскости данных SDN

**Igor Burdonov**[1,2,5], **Nina Yevtushenko**[1,3] **and Alexandre Kossatchev**[1,4]

[1] Ivannikov Institute for System Programming of the RAS, Moscow, Russia
[2] igor@ispras.ru
[3] evtushenko@ispras.ru
[4] kos@ispras.ru
[5] This work was supported by RFBR project 20-07-00338 A.

**Abstract**. В статье исследуется реализация виртуальных сетей на плоскости данных SDN, моделируемой графом физических связей между узлами сети. Виртуальная сеть задается как множество упорядоченных пар хостов (отправитель, получатель), а реализуется множеством путей хост-хост, однозначно определяющим настройки коммутаторов. Показано, что любое множество пар хостов можно реализовать на связном графе без возникновения бесконечного движения пакетов по циклу и без дублирующих путей, когда хост получает один и тот же пакет несколько раз. Однако могут возникать непредусмотренные пути, когда хост получает пакет, ему не предназначенный. С другой стороны, показано, что в некоторых случаях реализация без непредусмотренных путей неизбежно приводит к дублированию или зацикливанию пакетов. Ставится вопрос: на каком графе любое множество пар хостов можно реализовать без зацикливания, дублирования и непредусмотренных путей? Предлагается достаточное условие и выдвигается гипотеза о том, что это условие является также необходимым.

## 1. Introduction

В настоящее время программно-конфигурируемые сети (SDN) являются одной из основных технологий виртуализации сети [1][2][3][4][5]. На плоскости данных пересылаются пакеты между хостами через промежуточные коммутаторы. Это моделируется графом физических связей, вершинами которого являются хосты и коммутаторы, а ребра соответствуют физическим связям между ними. Настройку коммутаторов выполняют SDN-контроллеры, устанавливая для каждого коммутатора набор правил работы. Правило определяет, каким соседним вершинам графа пересылается принятый коммутатором пакет в зависимости от того, от какого соседа пакет пришёл, и от вектора параметров в заголовке пакета [6]. Тем самым, настройка коммутаторов сети определяет множество путей от хоста к хосту, по которым и будут пересылаться пакеты.

Здесь возникают задачи двух уровней. 1) Как реализовать заданное множество путей хост-хост через подходящие настройки коммутаторов? 2) Как реализовать заданное множество пар (хост, хост) через подходящие пути хост-хост в графе физических связей?

Известно, что при решении 1-й задачи возникают три эффекта. 1a) Могут появляться циклы, по которым пакеты будут передаваться бесконечно и, более того, бесконечно размножаться. 1b) Могут появляться непредусмотренные пути хост-хост. Эти проблемы исследовались в

[4][5]. 1c) Могут появляться дублирующие пути, из-за чего хост-адресат получает один и тот же пакет не один, а несколько раз.

2-я задача сводится к 1-й задаче выбором подходящего множества путей, по возможности избегая указанных эффектов. Здесь возникают следующие вопросы. 2a) Можно ли заданное множество пар хостов реализовать на графе, т.е. выбрать подходящее множество путей, по возможности без указанных эффектов? 2b) Можно ли любое множество пар хостов реализовать без указанных эффектов на данном графе физических связей?

В качестве ответа на вопроса 2a) в статье показывается, что любое множество пар хостов можно реализовать на связном графе без возникновения циклов и дублирования, однако могут возникать непредусмотренные пути, т.е. пути, соединяющие непредусмотренные пары хостов. С другой стороны в статье демонстрируется, что в некоторых случаях реализация заданного множества пар хостов без непредусмотренных путей неизбежно приводит к возникновению дублирования или циклов. В статье устанавливается достаточное условие положительного ответа на вопрос 2b) и выдвигается гипотеза о том, что это достаточное условие является необходимым.

## 2. Preliminaries

Графом физических связей (далее просто графом) будем называть связный неориентированный граф $G = \{V, E\}$ без кратных ребер и петель, где $V$ – множество коммутаторов и хостов, $E \subseteq V \times V$ – множество ребер, моделирующих физические связи между коммутаторами и хостами. Если не оговорено противное, под графом будет пониматься именно такой граф. Поскольку ребро, соединяющее вершины $a$ и $b$, неориентированное и нет кратных ребер, его можно обозначать как $ab$, так и $ba$. Поскольку нет петель, ребер вида $aa$ в $E$ нет. Поскольку нет кратных ребер, путь как последовательность смежных ребер однозначно задается последовательностью вершин $a_1 \ldots a_n$, через которые он проходит. Путь, начинающийся в вершине $a$ и заканчивающийся в вершине $b$, называют $ab$-путем. Если путь проходит по ребру $ab$ из $a$ в $b$, то будем говорить, что он проходит дугу $ab$. Если $a$ и $b$ хосты, $ab$-путь, в котором все вершины, кроме первой вершины $a$ и последней вершины $b$, коммутаторы, будем называть *полным*. Путь, в котором вершины (дуги) не повторяются, называется *вершинно-простым* (*реберно-простым*). Вершины графа будем обозначать строчными буквами $a$, $b$, $c$,…$x$, $y$, $z$, пути – жирными строчными буквами $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{r}$,…, а множества путей – прописными буквами – $P$, $Q$, $R$, …

Мы будем предполагать, что каждый хост $x$ подсоединен ровно к одному коммутатору [3]. Поэтому хост – это терминальная вершина графа, т.е. вершина степени 1. Если коммутатор $a$ имеет степень 1 и соединен с вершиной $b$, то любой полный путь, проходящий через $a$, имеет вид …$bab$…; удаляя из него все циклы $bab$, получаем путь, не проходящий через $a$. Это значит, что такой коммутатор «лишний», и достаточно рассматривать графы, в которых терминальные вершины – это только хосты. Множества хостов и коммутаторов обозначим через $H$ и $S$, соответственно; $H \cup S = V$, $H \cap S = \varnothing$.

В общем случае правило коммутатора $b$ имеет вид $\sigma abc$, где $a$ и $c$ соседи $b$, а $\sigma$ вектор параметров заголовка пакета, которые можно использовать в правилах. Такое правило означает, что коммутатор $b$, получив пакет с вектором $\sigma$ от соседа $a$, пересылает его соседу $c$. Предполагается, что коммутатор не меняет $\sigma$. Тем самым, для вектора $\sigma$ порождаются полные пути вида $a_1 \ldots a_n$, где для $i = 2..n - 1$ в коммутаторе $a_i$ есть правило $\sigma a_{i-1} a_i a_{i+1}$. Если есть два правила $\sigma abc$ и $\sigma abc`$, где $c \neq c`$, говорят, что пакет *клонируется*, т.е. пересылается обоим соседям $c$ и $c`$.

Заданное множество $P$ полных путей однозначно определяет минимальный набор правил коммутаторов, порождающий все пути из $P$. Однако это не значит, что порождаются только пути из $P$. Будем говорить, что два пути *сливаются* на дуге $ab$ в вершине $a$, если у них дуга $ab$ общая и не первая, а непосредственно предшествующие ей дуги $ca$ и $c`a$ разные (т.е. $c \neq c`$), и *разделяются* после дуги $de$ в вершине $e$, если у них дуга $de$ общая и не последняя, а непосредственно следующие дуги $ef$ и $ef`$ разные (т.е. $f \neq f`$).

Цикл порождается, если полный $xy$-путь проходит через некоторую дугу дважды, т.е. путь имеет вид $\boldsymbol{paqer}(\boldsymbol{aqer})^*\boldsymbol{aqes}$, где отрезок $\boldsymbol{p}$ начинается в хосте $x \neq a$, отрезки $\boldsymbol{p}$ и $\boldsymbol{r}$ не заканчиваются в одной вершине, после этих отрезков в пути следует коммутатор $a$, отрезок $\boldsymbol{aqer}$ проходится один или несколько раз, после коммутатора $e$ отрезки $\boldsymbol{r}$ и $\boldsymbol{s}$ не начинаются в одной вершине, и отрезок $\boldsymbol{s}$ заканчивается в хосте $y \neq e$. Двигаясь вдоль пути, мы видим, что в вершине $a$ путь сливается сам с собой, потом в вершине $e$ разделяется сам с собой, а затем это слияние и разделение происходит ещё раз (если путь проходит цикл $k$ раз, то будет $k + 1$ раз как разделение после слияния, так и слияние после разделения). Пакеты будут не только бесконечно ходить по циклу $\boldsymbol{aqer}$, но и бесконечно клонироваться в вершине $e$, так что хост $y$ будет получать бесконечное число клонов пакета.

Путь, который не сливается сам с собой, это реберно-простой путь. Для отсутствия циклов необходимо, чтобы все пути множества $P$ были реберно-простыми. Но этого не достаточно. Если два реберно-простых полных пути из $P$ после слияния на дуге $ab$ разделяются (после этой же или другой дуги), т.е. имеют вид $x\boldsymbol{p}ab\boldsymbol{q}y$ и $x`\boldsymbol{p`}ab\boldsymbol{q`}y`$ с разными начальными и конечными хостами $x \neq x`$ и $y \neq y`$, то порождаются и новые пути $x\boldsymbol{p}ab\boldsymbol{q`}y`$ и $x`\boldsymbol{p`}ab\boldsymbol{q}y$. Эта операция порождения новых путей называется замыканием по дугам, а результат замыкания по дугам всех пар путей из $P$ обозначается $P{\downarrow}{\uparrow}$ [4][5]. Очевидно, $P \subseteq P{\downarrow}{\uparrow}$. Если $P \neq P{\downarrow}{\uparrow}$, т.е. $P$ не замкнуто по дугам, то возникают непредусмотренные пути. В частности, могут возникнуть не реберно-простые пути и, следовательно, циклы. Появление циклов в замыкании по дугам множества полных путей всегда свидетельствует о бесконечности этого замыкания и, тем самым, наличии дублирования. В конечном замкнутом по дугам множестве полных реберно-простых путей циклов нет.

Для множества полных путей $P$ через $H(P) \subseteq H{\times}H$ обозначим множество пар $xy$, для которых в $P$ есть $xy$-путь. Множество пар хостов $D \subseteq H{\times}H$, не содержащее пар вида $xx$, будем называть *нормальным*. Будем говорить, что нормальное множество $D$ (*нестрого*) *реализуется* замкнутым по дугам множеством полных путей $P$, если $D \subseteq H(P)$, и, кроме того, *реализуется без циклов*, если $P$ конечно, *строго реализуется*, если $D = H(P)$, *реализуется без дублирования*, если $P$ содержит ровно один $xy$-путь для каждой пары $xy \in D$.

Если адрес хоста-отправителя входит в вектор параметров σ, то правила для векторов параметров с разными адресами хоста-отправителя работают независимо друг от друга. Для каждого хоста-отправителя $x$ в графе можно выбрать исходящее из $x$ дерево $I_x$ кратчайших путей, ведущих во все остальные хосты. Для любого нормального множества $D$ пар хостов и любого хоста $x$ выбирается подмножество $D_x$ пар, где первый элемент пары – это хост $x$, а в дереве $I_x$ – поддерево $I_x(D)$, в котором листовые вершины – это вершины $y$ такие, что $xy \in D_x$. В исходящем дереве все пути реберно-простые (даже вершинно-простые), и нет слияния, тем самым, нет и разделения после слияния. Поэтому $I_x(D)$ замкнуто по дугам и, очевидно, строго реализует $D_x$ без циклов и дублирования, причём используются кратчайшие полные пути. Таким образом, в этом случае нет проблем со строгой реализацией без циклов и дублирования любого нормального множества пар хостов. Более того, реализация любого такого множества оказывается подмножеством одного и того же множества путей – объединения деревьев $I_x$ по всем хостам $x$. Аналогичная процедура с аналогичным результатом применима тогда, когда в вектор параметров σ входит адрес хоста-получателя. Только здесь строится входящее дерево $O_x$ для каждого хоста $x$.

Ниже мы рассматриваем случай, когда адрес хоста-отправителя и адрес хоста-получателя не входят в вектор параметров σ. Остальные параметры никак не влияют на перемещение пакетов с данным вектором σ, поэтому мы будем опускать σ в обозначении правила и писать вместо σ$abc$ просто $abc$. Иными словами, правила коммутатора (для данного вектора σ) определяют, кому должен быть послан пакет, только в зависимости от соседа, от которого пакет принят. В этом случае число правил, по которым работает коммутатор, зависит только от числа его соседей и не зависит от числа хостов в сети.

## 3. Нестрогая/строгая реализация versus циклы и дублирование

В этом разделе мы исследуем связь нестрогой и строгой реализации множества пар хостов с наличием или отсутствием циклов и дублирования.

Утверждение 1. На связном графе $G$ любое нормальное множество $D$ пар хостов нестрого реализуется без циклов и дублирования.

Доказательство. Выберем в графе $G$ произвольное остовное дерево $T$. Поскольку хост имеет степень 1 в графе $G$, все хосты будут листьями дерева $T$. Пусть $P$ множество всех кратчайших полных путей по дереву $T$. Очевидно, что все пути из $P$ вершинно-простые и, следовательно, реберно-простые, нет дублирующих путей и $P$ конечно и замкнуто по дугам. Если во множестве $P$ оставить только такие $xy$-пути, что $xy \in D$, то для полученного множества $P(D)$ будет $P(D){\downarrow}{\uparrow} \subseteq P$. Тем самым, в $P(D){\downarrow}{\uparrow}$ тоже все пути реберно-простые, нет дублирующих путей и $P(D){\downarrow}{\uparrow}$ конечно и замкнуто по дугам . По построению $D = H(P(D)) \subseteq H(P(D){\downarrow}{\uparrow})$. Следовательно, $P(D){\downarrow}{\uparrow}$ нестрого реализует $D$ без циклов и дублирования.

□

Для полного пути $\boldsymbol{p}$ через $\boldsymbol{p}^{\circ}$ обозначим путь, который получается из $\boldsymbol{p}$ применением, пока возможно, следующей операции удаления циклов: путь $\boldsymbol{p} = \boldsymbol{q}a\boldsymbol{r}a\boldsymbol{s}$ превращается в путь $\boldsymbol{p}^{\circ} = \boldsymbol{q}a\boldsymbol{s}$. Заметим, что результат операции “°”, вообще говоря, неоднозначный. Для однозначности будем считать, что удаление цикла применяется только тогда, когда каждая вершина префикса $\boldsymbol{q}$ имеет только одно вхождение в $\boldsymbol{p}$, а $\boldsymbol{r}$ не содержит вхождений вершины $a$. Иными словами, среди всех вершин, имеющих несколько вхождений в $\boldsymbol{p}$, выбирается та вершина $a$, вхождение которой встретилось первым, и удаляется цикл от этого первого вхождения $a$ до ее второго вхождения. Например, для $\boldsymbol{p} = xacbcaby$, $\boldsymbol{p}^{\circ} = xaby$ (не $xacby$). Для множества полных путей $P$ обозначим через $P^{\circ}$ множество путей, получаемых удалением циклов из всех путей $P$, т.е. $P^{\circ} = \{\, \boldsymbol{p}^{\circ} \mid \boldsymbol{p} \in P \,\}$.

Утверждение 2. Пусть $P$ множество полных путей. Тогда $P^{\circ}$ состоит из вершинно-простых путей и связывает те же самые пары хостов, что множество $P$: $H(P^{\circ}) = H(P)$. Если $P$ замкнуто по дугам, то замыкание по дугам $P^{\circ}{\downarrow}{\uparrow}$ связывает те же самые пары хостов: $H(P^{\circ}{\downarrow}{\uparrow}) = H(P^{\circ})$, т.е. замыкание по дугам $P^{\circ}{\downarrow}{\uparrow}$ добавляет во множество $P^{\circ}$ только дублирующие пути. Если $P$ конечно и замкнуто по дугам, то $P^{\circ}$ конечно и замкнуто по дугам.

Доказательство. Очевидно, что удаление из пути всех циклов делает путь вершинно-простым. Поэтому $P^{\circ}$ состоит из вершинно-простых путей. Очевидно, что операция удаления одного цикла из одного пути не меняет $H(P)$. Следовательно, цепочка таких операций также не меняет $H(P)$. Следовательно, $P^{\circ}$ связывают те же самые пары хостов, что множество $P$: $H(P^{\circ}) = H(P)$.

Докажем, что если множество $P$ замкнуто по дугам, то $H(P^{\circ}{\downarrow}{\uparrow}) = H(P^{\circ})$. Действительно, пусть во множестве $P^{\circ}$ есть $xy$-путь $\boldsymbol{p}$ и $x^{\backprime}y^{\backprime}$-путь $\boldsymbol{q}$, которые получены удалением циклов из $xy$-пути $\boldsymbol{p}^{\backprime}$ и $x^{\backprime}y^{\backprime}$-пути $\boldsymbol{q}^{\backprime}$, соответственно, имеющихся во множестве $P$. Очевидно, что если пути $\boldsymbol{p}$ и $\boldsymbol{q}$ имеют общую дугу, то эта дуга также общая для путей $\boldsymbol{p}^{\backprime}$ и $\boldsymbol{q}^{\backprime}$. Поскольку $P$ замкнуто по дугам, оно содержит также $xy^{\backprime}$-путь и $x^{\backprime}y$-путь. Тем самым, после удаления циклов в $P^{\circ}$ также будет $xy^{\backprime}$-путь и $x^{\backprime}y$-путь. Поэтому $H(P^{\circ}{\downarrow}{\uparrow}) = H(P^{\backprime})$.

Пусть $P$ конечно и замнуто по дугам. Тогда, очевидно, $P^{\circ}$ также конечно. Докажем, что $P^{\circ}$ замкнуто по дугам. Пусть $P^{\circ}$ содержит пути $\boldsymbol{p}ab\boldsymbol{q}$ и $\boldsymbol{p}_1ab\boldsymbol{q}_1$ с общей дугой $ab$. Эти пути получены удалением циклов из путей $\boldsymbol{r}$ и $\boldsymbol{r}_1$, соответственно, имеющихся в $P$. Поскольку при удалении циклов любая непустая последовательность между любыми двумя вхождениями $a$ и $b$ может быть полностью удалена только вместе с удалением вхождения $a$ и/или $b$, пути $\boldsymbol{r}$ и $\boldsymbol{r}_1$ можно представить в виде $\boldsymbol{p}^{\backprime}ab\boldsymbol{q}^{\backprime}$ и $\boldsymbol{p}^{\backprime}_1ab\boldsymbol{q}^{\backprime}_1$, соответственно, где $\boldsymbol{p} = \boldsymbol{p}^{\backprime\circ}$, $\boldsymbol{p}_1 = \boldsymbol{p}^{\backprime}_1{}^{\circ}$, $\boldsymbol{q} = \boldsymbol{q}^{\backprime\circ}$,

$\boldsymbol{q}_1 = \boldsymbol{q}`_1{}^\circ$. Поскольку $P$ замкнуто по дугам, в нем имеются пути $\boldsymbol{p}`ab\boldsymbol{q}_1`$ и $\boldsymbol{p}`_1ab\boldsymbol{q}`$. А тогда в $P^\circ$ имеются пути $\boldsymbol{p}ab\boldsymbol{q}_1 = \boldsymbol{p}`^\circ ab\boldsymbol{q}_1`^\circ = (\boldsymbol{p}`ab\boldsymbol{q}_1`)^\circ$ и $\boldsymbol{p}_1ab\boldsymbol{q} = \boldsymbol{p}`_1{}^\circ ab\boldsymbol{q}`^\circ = (\boldsymbol{p}`_1ab\boldsymbol{q}`)^\circ$. Следовательно, множество $P^\circ$ замкнуто по дугам.

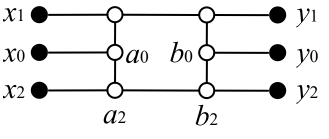
□

Заметим, что все условия на множество $P$ полных путей в утверждении 2 необходимы. Если множество $P$ не замкнуто по дугам, то замыкание по дугам $P^\circ\downarrow\uparrow$ может связывать дополнительные пары хостов: $H(P^\circ\downarrow\uparrow) \supseteq H(P^\circ)$, т.е. замыкание по дугам $P^\circ\downarrow\uparrow$ может добавить во множество $P^\circ$ не только дублирующие пути. Если множество $P$ конечно, но не замкнуто по дугам, то множество $P^\circ$ конечно, но может быть не замкнуто по дугам. В обоих случаях примером может служить множество $P = \{ xaby, x`aby` \}$, где $x \neq x`$ и $y \neq y`$: $P^\circ = P$, $P^\circ\downarrow\uparrow = \{ xaby, x`aby`, xaby`, x`aby \}$. Если множество $P$ замкнуто по дугам, но бесконечно, то множество $P^\circ$ может быть не замкнуто по дугам. Примером может служить множество $P = Q\downarrow\uparrow$, где $Q = \{ xabcdy, x`cdaby` \}$, $x \neq x`$ и $y \neq y`$: множество $P$ содержит не реберно-простой путь $xabcdaby`$, $P^\circ = Q$ и $P^\circ\downarrow\uparrow = P$.

Из утверждения 2 следует, что любое множество $D$ пар хостов, строго реализуемое без циклов, может быть строго реализовано множеством вершинно-простых путей. Достаточно вместо конечного замкнутого по дугам множества $P$ полных путей, строго реализующего $D$, т.е. $D = H(P)$, взять множество $P^\circ$ вершинно-простых путей.

<u>Утверждение 3.</u> Строгая реализация не всегда возможна без дублирования: существует граф, на котором некоторое нормальное множество пар хостов строго реализуется только с дублированием.

Доказательство. Рассмотрим пример на figure 1. Множество $D$ строго реализуется конечным замкнутым по дугам множеством путей $P$, которое содержит дублирующие пути $x_0a_0a_1b_1b_0y_0$ и $x_0a_0a_2b_2b_0y_0$. Пусть замкнутое по дугам множество путей $P`$ строго реализует множество $D$. Допустим, оно не содержит дублирующих путей. Поскольку $D$ содержит 7 пар хостов, $P`$ строго реализует $D$ и в $P`$ нет дублирующих путей, $P`$ должно содержать ровно 7 путей. Тогда по утверждению 2 мы можем выбрать множество $P`$, состоящее из вершинно-простых путей. Для того чтобы попасть из хоста $x_i$ в хост $y_j$, путь должен пройти либо по дуге $a_1b_1$, либо по дуге $a_2b_2$. Пусть $m_i$ путей, $i = 1, 2$, проходят дугу $a_ib_i$, и эти пути начинаются в $n_i$ хостах и заканчиваются в $k_i$ хостах. Тогда $n_1 + n_2 = 3$, $k_1 + k_2 = 3$, $m_1 + m_2 = 7$. Из замкнутости по дугам множества $P`$ следует $n_1k_1 = m_1$, $n_2k_2 = m_2$. Отсюда $n_1k_1 + (3 - n_1)(3 - k_1) = 7$, что влечет $2n_1k_1 = 3(n_1 + k_1) - 2$. Поскольку $k_1 \leq 3$ и $n_1 \leq 3$, имеем: $3k_1 = 2$ для $n_1 = 0$, $k_1 = -1$ для $n_1 = 1$, $k_1 = 4$ для $n_1 = 2$, $3k_1 = 7$ для $n_1 = 3$. Каждое из этих уравнений не имеет решения в целых неотрицательных числах или противоречит условию $k_1 \leq 3$. Мы пришли к противоречию, следовательно, наше допущение не верно, и в $P`$ есть дублирующие пути.




$D = \{ x_1y_1, x_1y_0, x_2y_2, x_2y_0, x_0y_1, x_0y_0, x_0y_2 \}$

$P = \{ x_1a_1b_1y_1, x_1a_1b_1b_0y_0, x_2a_2b_2y_2, x_2a_2b_2b_0y_0,$
$x_0a_0a_1b_1y_1, x_0a_0a_1b_1b_0y_0, x_0a_0a_2b_2b_0y_0, x_0a_0a_2b_2y_2, \}$

**Figure 1.** Множество $D$ строго реализуется только с дублированием

□

<u>Утверждение 4.</u> Строгая реализация не всегда возможна без циклов: существует граф, на котором некоторое нормальное множество пар хостов строго реализуемо, но только бесконечными замкнутыми множествами путей.

Доказательство. Рассмотрим пример на figure 2. Множество $D$ строго реализуется замыканием по дугам $P\downarrow\uparrow$ множества путей $P$. Но в $P$ есть пути $x_1a_1b_1c_2c_1a_2b_2y_2$ и $x_2a_2b_2d_2d_1a_1b_1y_1$, которые в $P\downarrow\uparrow$ порождают не реберно-простой путь $x_1a_1b_1c_2c_1a_2b_2d_2d_1a_1b_1y_1$ (проходит дважды по дуге $a_1b_1$), т.е. $P\downarrow\uparrow$ бесконечно. Пусть замкнутое по дугам множество путей $P`$ строго реализует множество $D$. Допустим, оно конечно. Тогда по утверждению 2 мы можем выбрать множество $P`$, состоящее из вершинно-простых путей. Такой путь обходит цикл коммутаторов либо по часовой стрелке, либо против часовой стрелки.

1. Пусть $u_1u_2$-путь $\boldsymbol{p}_1$ идет по часовой стрелке, тогда $\boldsymbol{p}_1 = u_1d_1a_1b_1c_2c_1a_2b_2d_2u_2$.

   1.1. Если $x_2y_2$-путь $\boldsymbol{p}_2$ идет по часовой стрелке, тогда $\boldsymbol{p}_2 = x_2a_2b_2y_2$; $u_1u_2$-путь $\boldsymbol{p}_1$ и $x_2y_2$-путь $\boldsymbol{p}_2$ имеют общую дугу $a_2b_2$, что в замыкании по дугам порождает $u_1y_2$-путь, но $u_1y_2 \notin D$. Значит $\boldsymbol{p}_2$ идет против часовой стрелки, тогда $\boldsymbol{p}_2 = x_1a_2c_1c_2b_1a_1d_1d_2b_2y_2$.

   1.2. Если $x_1y_1$-путь $\boldsymbol{p}_3$ идет по часовой стрелке, тогда $\boldsymbol{p}_3 = x_1a_1b_1y_1$; $u_1u_2$-путь $\boldsymbol{p}_1$ и $x_1y_1$-путь $\boldsymbol{p}_3$ имеют общую дугу $a_1b_1$, что в замыкании по дугам порождает $u_1y_1$-путь, но $u_1y_1 \notin D$. Значит $\boldsymbol{p}_3$ идет против часовой стрелки, тогда $\boldsymbol{p}_3 = x_1a_1d_1d_2b_2a_2c_1c_2b_1y_1$.

   1.3. Но тогда $x_2y_2$-путь $\boldsymbol{p}_2$ и $x_1y_1$-путь $\boldsymbol{p}_3$ имеют общую дугу $d_2b_2$, что в замыкании по дугам порождает путь $x_1a_2c_1c_2b_1a_1d_1d_2b_2a_2c_1c_2b_1y_1$, который не является реберно-простым, так как дважды проходит по дуге $a_2c_1$, т.е. порождается цикл.

Следовательно, $u_1u_2$-путь $\boldsymbol{p}_1$ идет против часовой стрелки, тогда $\boldsymbol{p}_1 = u_1d_1d_2u_2$.

2. В силу симметрии аналогично доказывается (рассматривая $v_1v_2$-путь, $x_1y_1$-путь и $x_2y_2$-путь), что $v_1v_2$-путь $\boldsymbol{p}_4$ идет против часовой стрелки, тогда $\boldsymbol{p}_4 = v_1c_1c_2v_2$.

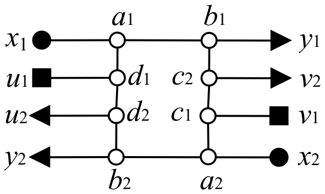
3. Если $x_1y_2$-путь $\boldsymbol{p}_5$ идет против часовой стрелки, тогда $\boldsymbol{p}_5 = x_1a_1d_1d_2b_2y_2$; $u_1u_2$-путь $\boldsymbol{p}_1$ и $x_1y_2$-путь $\boldsymbol{p}_5$ имеют общую дугу $d_1d_2$, что в замыкании по дугам порождает $u_1y_2$-путь, но $u_1y_2 \notin D$.

Следовательно, $\boldsymbol{p}_5$ идет по часовой стрелке, тогда $\boldsymbol{p}_5 = x_1a_1b_1c_2c_1a_2b_2y_2$.

4. В силу симметрии аналогично доказывается (рассматривая $x_2y_1$-путь и $v_1v_2$-путь), что $x_2y_1$-путь $\boldsymbol{p}_6$ идет по часовой стрелке, тогда $\boldsymbol{p}_6 = x_2a_2b_2d_2d_1a_1b_1y_1$.

Но тогда пути $\boldsymbol{p}_5$ и $\boldsymbol{p}_6$ имеют общую дугу $a_2b_2$ и в замыкании по дугам порождают путь $x_1a_1b_1c_2c_1a_2b_2d_2d_1a_1b_1y_1$, который не является реберно-простым, так как дважды проходит по дуге $a_1b_1$, т.е. порождается цикл.

Мы пришли к противоречию и, следовательно, наше допущение не верно, и утверждение доказано.




$D = \{ x_1y_1, x_1y_2, x_2y_2, x_2y_1, u_1u_2, v_1v_2 \}$
$P = \{ x_1a_1b_1y_1, x_1a_1b_1c_2c_1a_2b_2y_2,$
$x_2a_2b_2y_2, x_2a_2b_2d_2d_1a_1b_1y_1,$
$u_1d_1d_2u_2, v_1c_1c_2v_2 \}$

**Figure 2.** Множество $D$ строго реализуется только с циклами

□

## 4. Достаточное условие строгой реализации любого множества пар хостов без циклов и дублирования

В этом разделе мы исследуем достаточные условия на граф, позволяющие строго реализовывать без циклов и дублирования любые множества пар хостов. Если для двух путей есть слияние на дуге $ab$ и есть разделение после дуги $cd$, то будем говорить, что разделение происходит после слияния, если хотя бы в одном из этих путей сначала проходится дуга $ab$, а потом дуга $cd$. Соответственно, слияние происходит после разделения, если хотя бы в одном из этих путей сначала проходится дуга $cd$, а потом дуга $ab$. Заметим, что в линейном порядке вершин одного пути разделение после дуги $cd$ может происходить после слияния на дуге $ab$, а в линейном порядке вершин другого пути, наоборот, слияние на дуге $ab$ –после разделения после дуги $cd$, что демонстрируется примером двух путей: $xabefcdy$ и $ufcdabev$, где разными буквами обозначены разные вершины.

<u>Утверждение 5.</u> Если множество полных путей конечно, то отсутствие разделения после слияния достаточно, но не необходимо для замкнутости по дугам и отсутствия циклов.

Доказательство. Достаточность следует из того, что замыкание по дугам порождает новые пути только в случае разделения после слияния. Также цикл порождается не реберно-простым путем, в котором, как указано в разделе 2, имеет место разделение после слияния. Но наличие разделения после слияния не обязательно означает незамкнутость по дугам или наличие циклов, что демонстрируется следующим примером конечного замкнутого по дугам множества полных путей без циклов $P = \{ xaby, xaby`, x`aby, x`aby` \}$, где все хосты $x$, $y$, $x`$ и $y`$ разные.

□

<u>Утверждение 6.</u> Для замкнутого множества $P$ полных путей условие отсутствия слияния после разделения необходимо и достаточно для отсутствия дублирования.

Доказательство. Пусть есть два разных $xy$-пути. Поскольку каждый хост подсоединен ровно к одному коммутатору, максимальный общий префикс этих путей и максимальный общий постфикс этих путей имеет каждый длину не меньше 1, и префикс можно представить в виде $x\mathbf{p}a$, а постфикс – в виде $b\mathbf{q}y$. Тогда, очевидно, $a$ проходится в каждом из путей раньше $b$, и пути имеют вид $x\mathbf{p}a\mathbf{r}b\mathbf{q}y$ и $x\mathbf{p}a\mathbf{r}`b\mathbf{q}y$. Тем самым, эти пути разделяются в вершине $a$, и после этого сливаются в вершине $b$. Отсюда следует достаточность условия. Докажем необходимость условия. Пусть есть слияние после разделения для двух полных путей: $xy$-пути и $x`y`$-пути, причем на $xy$-пути сначала проходится вершина $a$, в которой пути разделяются, а потом вершина $b$, в которой пути сливаются. Возможны два случая.

1) Figure 3 ($a$). На $x`y`$-пути вершины $a$ и $b$ проходятся том же порядке: $ab$. Тогда пути имеют вид $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$ и $x`\mathbf{p}`a\mathbf{q}`b\mathbf{r}`y`$, где отрезки $x\mathbf{p}$ и $x`\mathbf{p}`$ заканчиваются в одной вершине $c$, отрезки $\mathbf{q}$ и $\mathbf{q}`$ начинаются в разных вершинах (и заканчиваются в разных вершинах), отрезки $\mathbf{r}y$ и $\mathbf{r}`y`$ начинаются в одной вершине $d$. Дуга $ca$ общая для этих путей, поэтому в замыкании по дугам порождается путь $x\mathbf{p}a\mathbf{q}`b\mathbf{r}`y`$. Сопоставим этот путь с путем $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$. Дуга $db$ общая для этих путей, поэтому в замыкании по дугам порождается путь $x\mathbf{p}a\mathbf{q}`b\mathbf{r}y$. Поскольку отрезки $\mathbf{q}$ и $\mathbf{q}`$ начинаются (и заканчиваются) в разных вершинах, пути $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$ и $x\mathbf{p}a\mathbf{q}`b\mathbf{r}y$ разные, следовательно, это дублирующие пути.

2) Figure 3 ($b$). На $x`y`$-пути вершины $a$ и $b$ проходятся в обратном порядке: $ba$. Тогда пути имеют вид $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$ и $x`\mathbf{p}`b\mathbf{q}`a\mathbf{r}`y`$, где отрезки $x\mathbf{p}$ и $b\mathbf{q}`$ заканчиваются в одной вершине $c$, (отрезки $\mathbf{q}$ и $\mathbf{r}`y`$ начинаются в разных вершинах, отрезки $\mathbf{q}$ и $x`\mathbf{p}`$ заканчиваются в разных вершинах,) отрезки $\mathbf{r}y$ и $\mathbf{q}`a$ начинаются в одной вершине $d$. Дуга $bd$ общая для этих путей, поэтому в замыкании по дугам порождается путь $x\mathbf{p}a\mathbf{q}b\mathbf{q}`a\mathbf{r}`y`$. Сопоставим этот путь с путем $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$. Дуга $ca$ общая для этих путей, поэтому в замыкании по дугам порождается путь $x\mathbf{p}a\mathbf{q}b\mathbf{q}`a\mathbf{q}b\mathbf{r}y$. Сопоставим этот путь с путем $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$. Поскольку отрезок $\mathbf{q}`a\mathbf{q}b$ заведомо не пуст, пути $x\mathbf{p}a\mathbf{q}b\mathbf{r}y$ и $x\mathbf{p}a\mathbf{q}b\mathbf{q}`a\mathbf{q}b\mathbf{r}y$ разные, следовательно, это дублирующие пути.

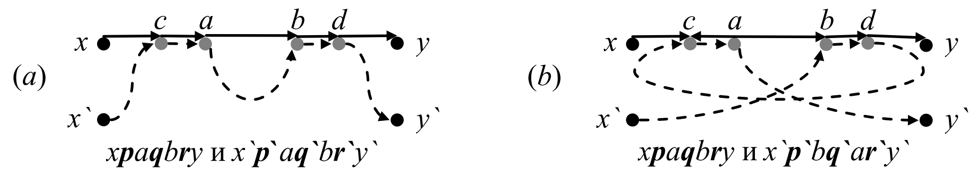



**Figure 3.** Слияние после разделения

□

Граф, в котором любое нормальное множество пар хостов можно строго реализовать без циклов, назовём *почти хорошим*. Граф, в котором любое нормальное множество пар хостов можно строго реализовать без циклов и без дублирования, назовём *хорошим*.

Конечное замкнутое множество $P$ путей, связывающее все пары хостов (т.е. $H(P)$ наибольшее нормальное множество пар хостов), назовем *почти совершенным*, если в нем нет разделения после слияния путей, и *совершенным*, если в нем, кроме того, нет слияния после разделения путей. Граф будем называть *почти совершенным* или *совершенным*, если в нем есть, соответственно, почти совершенное или совершенное множество путей.

Достаточное условие строгой реализации любого нормального множества пар хостов без циклов и дублирования теперь можно сформулировать в виде следующего утверждения.

Утверждение 7. Почти совершенный граф почти хороший, а совершенный граф хороший. При этом почти совершенное множество путей для каждого нормального множества пар хостов содержит его строгую реализацию без циклов как подмножество, а совершенное множество путей для каждого нормального множества пар хостов содержит его строгую реализацию без циклов и без дублирования как подмножество.

Доказательство. Поскольку почти совершенное множество путей конечно и в нем нет разделения после слияния, любое его подмножество также конечно и в нем нет разделения после слияния, поэтому, по Утверждение 5, оно замкнуто по дугам и не порождает циклов. По определению совершенное множество является почти совершенным, поэтому любое его подмножество также конечно, замкнуто по дугам и не порождает циклов. Поскольку в совершенном множестве путей нет слияния после разделения, в любом его подмножестве также нет слияния после разделения, и, по Утверждение 6, оно не порождает дублирование. Для любого нормального множества $D$ пар хостов в почти совершенном (совершенном) множестве $P$ путей можно выбрать подмножество $P(D)$ такое, что $H(P(D)) = D$. Множество $P(D)$ путей строго реализует множество $D$ пар хостов без циклов и, если множество $P$ совершенное, без дублирования.

□

## 5. Заключение

В статье показано, что любое множество пар хостов можно реализовать на связном графе с помощью путей, соединяющих хосты заданных пар, без возникновения циклов, по которым пакеты будут циркулировать бесконечно и бесконечно размножаться, и дублирующих путей, т.е. разных путей, соединяющих одни и те же пары хостов. Однако при этом могут возникать непредусмотренные пути, соединяющие дополнительные пары хостов, отсутствующие в заданном множестве пар. Если потребовать отсутствия непредусмотренных путей, то на некоторых графах некоторые множества пар хостов реализуются только с дублированием или циклами. Сформулированы и доказаны требования к графу, которые достаточны для реализации любого множества пар хостов без циклов (с возможным дублированием), и без циклов и без дублирования. В то же время сама возможность реализации на графе любого множества пар хостов без циклов и, тем более, без циклов и без дублирования представляется достаточно сильным требованием. Поэтому можно высказать гипотезу о том, что эти требования на граф являются также и необходимыми. Подтверждение или опровержение этой гипотезы является одним из направлений дальнейших исследований.

## References

[1] Sezer S, Scott-Hayward S, Chouhan P K, Fraser B, Lake D, Finnegan J, Viljoen N, Miller M and Rao N 2013 Are we ready for sdn? Implementation challenges for software-defined networks *IEEE Communications Magazine* **51** **7** pp 36-43

[2] López J, Kushik N, Yevtushenko N and Zeghlache D 2017 Analyzing and Validating Virtual Network Requests *Proc. ICSOFT* pp 441-6.

[3] Yevtushenko N, Kossatchev A, Lopez J, Kushik N and Zeghlache D 2018 Test Derivation for the Software Defined Networking Platforms: Novel Fault Models and Test Completeness Proc. IEEE East-West Design and Test Symposium EWDTS 2018 **8524712** pp 1-5

[4] Burdonov I B, Yevtushenko N V and Kossatchev A S 2018 Testing switch rules in software defined networks *Trudy ISP RAN/Proc. ISP RAS* vol **30** issue **6** 2018 pp 69-88 (in Russian)

[5] Burdonov I, Kossachev A, Yevtushenko N, López J, Kushik N and Zeghlache D 2019 Verifying SDN Data Path Requests *CoRR abs/1906.03101* (2019)

[6] Boufkhad Y, De La Paz R, Linguaglossa L, Mathieu F, Perino D and Viennot L 2016 Forwarding tables verification through representative header sets *arXiv preprint*

*arXiv:1601.07002 (2016)*