

УДК 621.391

В. В. Титов, магистрант

И. А. Кайсина, аспирант

А. В. Абилов, канд. техн. наук, доц., декан факультета

М. А. Ламри, аспирант

E-mail: kafsts@istu.ru

Кафедра «Сети связи и телекоммуникационные системы»

Ижевский государственный технический университет имени М. Т. Калашникова

Обзор методов улучшения доставки потоковых данных в беспроводных самоорганизующихся сетях

В статье рассматриваются различные технологии QoS и методы контроля нагрузки применительно к самоорганизующимся сетям. Подробно описаны механизмы устранения и предотвращения перегрузки. Анализируется реализация контроля нагрузки в протоколе TCP.

Ключевые слова: самоорганизующиеся сети, передача данных, QoS, контроль нагрузки.

Контроль нагрузки и QoS

Контроль перегрузки и качество обслуживания – это две проблемы, настолько тесно связанные между собой, что улучшение одного означает улучшение другого, а игнорирование одного обычно означает игнорирование другого. Большинство методов предотвращения или устранения перегрузок также улучшают качество обслуживания в сети. Важной проблемой в сети с коммутацией пакетов является перегрузка [5]. Перегрузка в сети может возникнуть, если нагрузка на сеть, т. е. количество пакетов, отправленных в сеть, больше, чем емкость сети, определяющей количество пакетов, которое может обработать сеть. Контроль перегрузки относится к механизмам и методам контроля нагрузки и ее удержания ниже пропускной способности.

Контроль перегрузки может либо предотвратить переполнение до того, как оно произойдет, либо устранить перегрузку после того, как она произошла. В общем случае можно разделить механизмы управления перегрузками на две широкие категории: контроль перегрузки по разомкнутому контуру (предотвращение) и контроль перегрузки по замкнутому контуру (устранение), как показано на рис. 1.

К механизмам предотвращения (Open-loop) относятся политики, предотвращающие перегрузку до ее возникновения. В этих механизмах

управление перегрузкой осуществляется либо источником, либо получателем [8]:

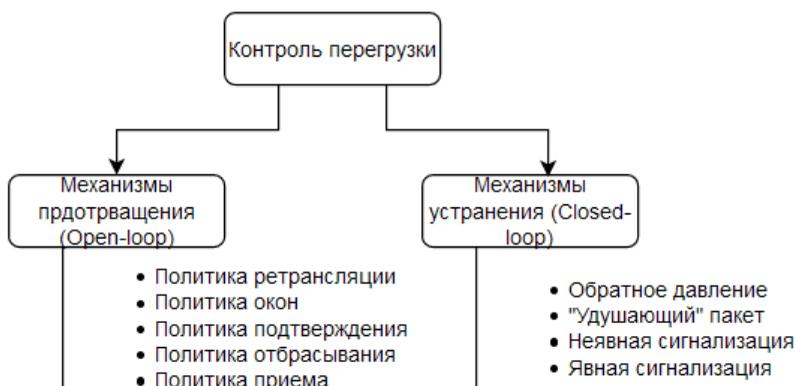


Рис. 1. Механизмы управления перегрузками

- Политика ретрансляции – оптимизация таймеров ретрансляции повышает эффективность и в то же время предотвращает перегрузку.
- Политика окон – выбор корректного размера окна у отправителя.
- Политика подтверждения – если получатель не подтверждает каждый пакет, который он получает, это может замедлить отправителя и помочь предотвратить перегрузку.
- Политика отбрасывания – отбрасывание незначительных пакетов не повредит целостности передачи.
- Политика приема – коммутаторы в потоке трафика сначала проверяют потребность потока в ресурсах, прежде чем пустить его в сеть. Маршрутизатор может отказать в установлении соединения, если в сети существует перегрузка или существует возможность будущей перегрузки.

Механизмы устранения (Closed-loop) пытаются облегчить перегрузку после того, как она произошла.

- Протодавление – перегруженный узел перестает получать данные от непосредственного вышестоящего узла или узлов. Это может привести к тому, что вышестоящий узел или узлы будут перегружены и они, в свою очередь, отклонят данные от своих вышестоящих узлов или узла.
- Обратное давление – начинается с узла и распространяется в противоположном идущим данным направлении, к источнику.

- «Удушающий» пакет – это пакет, отправленный узлом источнику, чтобы сообщить ему о перегрузке.
- Неявная сигнализация – источник предполагает, что где-то в сети существует перегрузка по иным причинам.
- Явная сигнализация – узел, который испытывает перегрузку, может явно послать сигнал источнику.

Реализация контроля нагрузки

При обычной передаче данных размер окна определяется получателем и непосредственно сетью. Когда в сети возникает перегрузка, она будет оказывать влияние на размер окна получателя. У отправителя есть следующая информация: предполагаемый размер окна получателя и пороговый размер окна перегрузки. Можно сказать, что размер окна является минимальным из этих двух. Рассмотрим несколько примеров реализации контроля нагрузки.

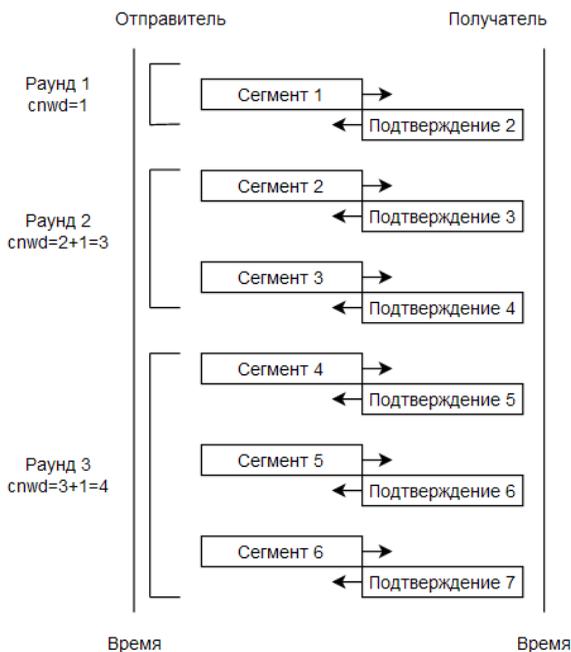


Рис. 2. Механизм предотвращения перегрузки

Для предотвращения перегрузки применяется аддитивное увеличение. Если мы начнем с алгоритма медленного запуска, размер окна перегрузки увеличивается экспоненциально. Чтобы избежать заторов до того, как это произойдет, нужно замедлить этот экспоненциальный рост. ТСР располагает еще одним алгоритмом, называемым избеганием перегрузки, который подвергается аддитивному увеличению вместо экспоненциального. Когда размер окна перегрузки достигает порога медленного запуска, фаза медленного запуска останавливается и начинается аддитивная фаза. В этом алгоритме каждый раз, когда доставляются все сегменты, помещаемые в окно (первый раунд), размер окна перегрузки увеличивается на единицу. Чтобы продемонстрировать это, применим данный алгоритм к тому же сценарию, что и медленный запуск. Хотя алгоритм предотвращения перегрузки обычно запускается, когда размер окна намного больше единицы. На рис. 2 показана эта идея.

В этом случае после получения отправителем подтверждений для всего окна сегментов размер окна увеличивается на один сегмент. Если смотреть на размер $cnwnd$ со стороны раундов (round), в ходе которых происходит изменение размера окна, то обнаружим, что скорость аддитивна, как показано на рис. 3.

Начало	⇒	$cnwnd=1$
После первого раунда	⇒	$cnwnd=1+1=2$
После второго раунда	⇒	$cnwnd=2+1=3$
После третьего раунда	⇒	$cnwnd=3+1=4$

Рис. 3. Аддитивное увеличение размера окна

При использовании алгоритма предотвращения перегрузки размер окна увеличивается аддитивно до момента обнаружения этой перегрузки.

Для обнаружения перегрузки используется мультипликативное уменьшение. При возникновении перегрузки пороговое значение размера окна должно быть уменьшено. Существует лишь один способ, благодаря которому отправитель может понять, что произошла перегрузка, – это необходимость повторной передачи сегмента. Однако повторная передача может происходить в одном из двух случаев: когда таймер истекает или когда получено три АСК. В обоих случаях размер порога снижается наполовину, что является мультипликативным уменьшением.

Большинство реализаций ТСР имеют две реакции [7]:

а) если происходит тайм-аут, существует более высокая вероятность перегрузки; сегмент, вероятно, был отброшен в сеть и нет никаких новостей об отправленных сегментах. В этом случае TCP реагирует сильно:

- устанавливает значение порога в половину текущего размера окна;
- устанавливает $swnd$ на размер одного сегмента;
- снова начинает фазу медленного запуска;

б) если получено три АСК, существует более слабая вероятность перегруженности; сегмент может быть отброшен, но некоторые сегменты после этого могут благополучно прибыть, т. к. получены три АСК. Это называется быстрой передачей и быстрым восстановлением. В этом случае TCP имеет более слабую реакцию:

- устанавливается значение порога равное половине текущего размера окна;
- устанавливается $swnd$, равное значению порога (в некоторых реализациях к порогу прибавляется три размера сегментов);
- начинается фаза предотвращения перегрузки.

Пример на рис. 4.



Рис. 4. Реакции протокола TCP на обнаружение перегрузки

Исходя из предположения, что максимальный размер окна 32 сегмента, устанавливается пороговое значение, равное 16 сегментам, т. е. половина максимального размера. В течение фазы медленного запуска размер окна начинает расти с единицы экспоненциально до достижения порогового значения. После этого процедура аддитивного увеличения позволяет размеру окна линейно расти, пока не наступит тайм-аут или не упрется в ограничение максимального размера окна. На рис. 4 тайм-аут наступает при размере окна, равном 20. В этот момент процедура

мультипликативного уменьшения берет на себя инициативу и уменьшает порог до значения, равного половине предыдущего размера окна. Поскольку последний зафиксированный размер окна был 20, новый порог будет равен 10. TCP вновь начинает процесс медленного запуска с размером окна 1. Далее, когда размер окна достигает значения 12, происходит событие three-ACKs. Процедура мультипликативного уменьшения повторяется снова. Пороговое значение устанавливается равным 6, и на этот раз TCP переходит в фазу аддитивного увеличения. Он остается в этой фазе до тех пор, пока не произойдет еще один тайм-аут или еще три ACK.

Применение контроля нагрузки в самоорганизующихся сетях

В самоорганизующихся сетях существуют следующие особенности [4]:

- частая смена топологии и маршрутов передачи данных;
- некоторые пакеты, которые мы отправляем, неизбежно теряются по пути;
- случайная перестановка пакетов местами;
- большие и нестабильные значения задержки.

Учитывая вышеперечисленные особенности, TCP зачастую оказывается не эффективен в таких сетях. Стандартные механизмы борьбы с перегрузками в TCP не учитывают случайный характер потерь и нестабильные характеристики канала. Как правило, при возникновении любых потерь механизм борьбы с перегрузкой резко снижает скорость передачи, вследствие чего полоса пропускания расходуется не эффективно.

На момент написания статьи предложены следующие варианты развития механизма управления перегрузками в TCP [2]:

- управление перегрузкой на канальном или сетевом уровне. Пропускная способность TCP улучшается за счет улучшения взаимодействия с канальным уровнем IEEE 802.11. Скрытые терминальные эффекты, вызванные помехами, могут влиять на производительность TCP и устраняться путем ограничения максимального размера окна отправки. Фреймворк Atra [3] состоит из трех механизмов на канальном и сетевом уровнях для улучшения производительности TCP в сетях MANET. Atra [3] повышает пропускную способность стека протоколов по умолчанию на 50–100 %;
- load-balanced – адаптивная маршрутизация в зависимости от перегрузки. В такой маршрутизации для определения состояния перегрузки и самого слабого узла маршрута используются две метрики (плотность трафика и время жизни, связанные с маршрутом). Маршрут с низкой

плотностью трафика и максимальным временем жизни выбирается для передачи пакетов;

– управление перегрузкой в протоколе мультиточковой маршрутизации может обеспечить балансировку нагрузки и улучшить качество обслуживания за счет обнаружения нескольких путей и поддержания этих путей. Механизм управления перегрузкой в MLBCC (методика распределения нагрузки по мультиточковому каналу для контроля перегрузки) [1] обнаруживает перегрузку в узле-кандидате, используя входящую и исходящую скорости. Методика распределения нагрузки [1] также распределяет трафик по нескольким путям с использованием узлов шлюзов, которые выбираются с использованием стоимости пути и стоимости канала.

Заключение

В статье проведен обзор методов улучшения доставки данных в самоорганизующихся сетях. Подробно рассмотрен контроль нагрузки как часть качества обслуживания QoS и механизмы с политиками устранения и предотвращения перегрузки. Также проанализирована реализация контроля нагрузки в протоколе TCP. Это позволит в дальнейшем существенно повысить качество передачи данных и снизить количество ошибок в ad-hoc-сетях.

Работа выполнена при финансовой поддержке РФФИ (проект №19-29-06076).

Список литературы

1. Mallapur, S. V., Patil, S. R., Agarkhed, J. V. Load balancing technique for congestion control multipath routing protocol in MANETs // *Wireless Personal Communications*. 2017. Vol. 92. No. 2. Pp. 749-770.
2. Kanellopoulos, D. Congestion control for MANETs: An overview // *ICT Express*. 2019. Vol. 5. No. 2. Pp. 77-83.
3. Anantharaman, V. TCP performance over mobile ad hoc networks: a quantitative study / V. Anantharaman, S. J. Park, K. Sundaresan, R. Sivakumar // *Wireless Communications and Mobile Computing*. 2004. Vol. 4. No. 2. Pp. 203-222.
4. Кучерявый, А. Е. Теоретические и практические направления исследований в области летающих сенсорных сетей / А. Е. Кучерявый, А. Г. Владыко, Р. В. Киричек // *Электросвязь*. – 2015. – № 7. – С. 9–11.
5. Кучерявый, А. Е. Летающие сенсорные сети – новое приложение интернета вещей / А. Е. Кучерявый, А. Г. Владыко, Р. В. Киричек // *Актуальные проблемы инфотелекоммуникаций в науке и образовании. IV Международная научно-техническая и научно-методическая конференция : сб. науч. ст. в 2 т. – 2015. – С. 17–22.*

6. *Meitis, D., Vasiliev, D., Abilov, A.* Simulation of MANETs routing protocols for UAVs // Fourth Forum of Young Researchers. In the framework of International Forum “Education Quality – 2014”: Conf. proceedings (April 23, 2014). Izhevsk, Russia : Publ. House of Kalashnikov ISTU, 2014. Pp. 358–363.

7. *Pleisch, S.* MISTRAL: efficient flooding in mobile ad-hoc networks / S. Pleisch, M. Balakrishnan, K. Birman, R. Renesse // MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. Florence, Italy : Association for Computing Machinery, 2006. Pp. 1–12. DOI: 10.1145/1132905.1132907.

8. *Forouzan, A.* Data communications and networking. McGraw-Hill Higher Education, 2006. 1171 p.

V. V. Titov, master student

I. A. Kaisina, postgraduate student

A. V. Abilov, CSc in engineering, associate professor

M. A. Lamri, postgraduate student

E-mail: kafsts@istu.ru

Department of Communication Networks and Telecommunication Systems
Kalashnikov Izhevsk State Technical University, Izhevsk, Russian Federation

Overview of Methods to Improve Delivery of Streaming Data in Wireless Ad-hoc Networks

The article discusses various QoS technologies and load control methods applied to self-organizing networks. Mechanisms for eliminating and preventing overloading are described in detail. The implementation of load monitoring in the TCP protocol is also analyzed.

Keywords: self-organizing networks, data transmission, QoS, load monitoring.