
Алгоритм адаптивной настройки параметров протокола маршрутизации AODV

С.П. Шипицин

Пермский национальный исследовательский политехнический университет

Аннотация: Большое разнообразие применений беспроводных самоорганизующихся сетей и высокие требования к их качеству обуславливают невозможность создания универсального протокола маршрутизации, одинаково хорошо функционирующего во всех случаях. Каждый раз при развёртывании новой сети приходится специально подбирать его параметры, однако это неудобно и зачастую неэффективно. В настоящей статье предлагается алгоритм-надстройка над протоколом маршрутизации AODV, позволяющий автоматически адаптировать параметры протокола под динамически изменяющиеся параметры сети. Показана перспектива предложенного подхода. Предложена архитектура нейронной сети, позволяющей осуществлять управление протоколом маршрутизации, обозначена область дальнейших исследований.

Ключевые слова: AODV, протокол маршрутизации, адаптивная маршрутизация, ad hoc сеть, параметрическая адаптация, имитационное моделирование.

Протокол маршрутизации Ad Hoc On-Demand Distance Vector (AODV) предназначен для маршрутизации в беспроводных самоорганизующихся (ad hoc) сетях и функционирует на прикладном уровне, используя в качестве транспортного протокола UDP [1, 2]. Поскольку сфера применения ad hoc сетей достаточно широка [3, 4], выработать универсальное решение для эффективной маршрутизации невозможно, и специалистам каждый раз при развёртывании сети приходится специально подбирать параметры протокола [5, 6]. В реализации AODV таких параметров двадцать пять. Некоторые из них вычисляются, однако большая часть должна быть вручную задана специалистом, развёртывающим сеть, перед её запуском. Ниже перечислена часть этих параметров:

- HelloInterval – интервал генерации сообщений HELLO;
- TtlStart – начальное значение «времени жизни» (time to live, TTL) для пакетов RREQ;
- TtlIncrement - увеличение TTL пакетов RREQ на каждой итерации поиска (расширяющееся кольцо);

- RreqRetries - максимальное число дополнительных попыток передачи пакета RREQ;
- RreqRateLimit - максимальное число отправляемых в секунду пакетов RREQ;
- RerrRateLimit - максимальное число отправляемых в секунду пакетов RERR;
- NodeTraversalTime - эмпирическая оценка среднего полного времени прохождения пакетом одного транзитного участка сети (хопа);
- ActiveRouteTimeout - период времени, в течение которого маршрут считается корректным;
- NetDiameter - максимальное число хопов между двумя узлами;
- MaxQueueLen - максимальное количество буферизуемых пакетов;
- MaxQueueTime - максимальное время пребывания пакета в очереди;
- AllowedHelloLoss - количество пакетов HELLO, которые могут быть потеряны на корректном маршруте.

Очевидно, что вручную задавать все эти параметры при развёртывании каждой конкретной сети (а это необходимо для обеспечения высокого качества маршрутизации) – задача нетривиальная. Влияние отдельных параметров ещё можно оценить, однако с их совокупным вкладом в качество маршрутизации для каждого отдельного случая ситуация гораздо сложнее. Есть решения [7-9], автоматизирующие этот процесс, но они не решают другой проблемы – характеристики сети могут меняться в процессе работы, и параметры протокола перестают отвечать изменившимся условиям. Представленные на рис. 1, 2 графики скомпонованы из результатов симуляции сети в сетевом симуляторе ns-3 и наглядно иллюстрируют эти сложности. Как видно, существенные коррективы вносит даже само расположение узлов, а в случае с мобильной сетью заранее можно определить разве что примерный сценарий их взаимодействия [10].

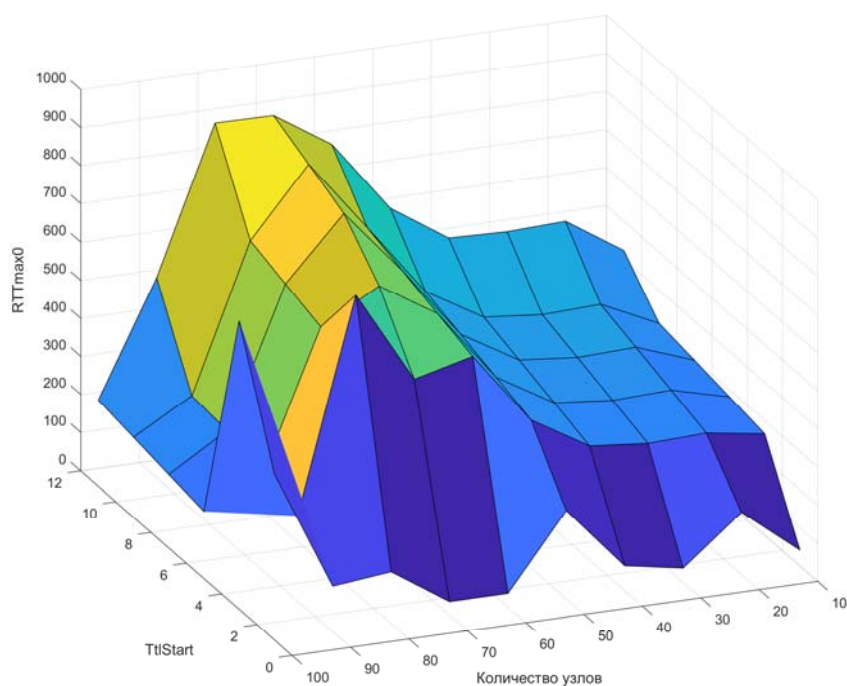


Рис. 1. – Влияние параметра TtlStart на скорость установки соединения в сети со случайным расположением узлов

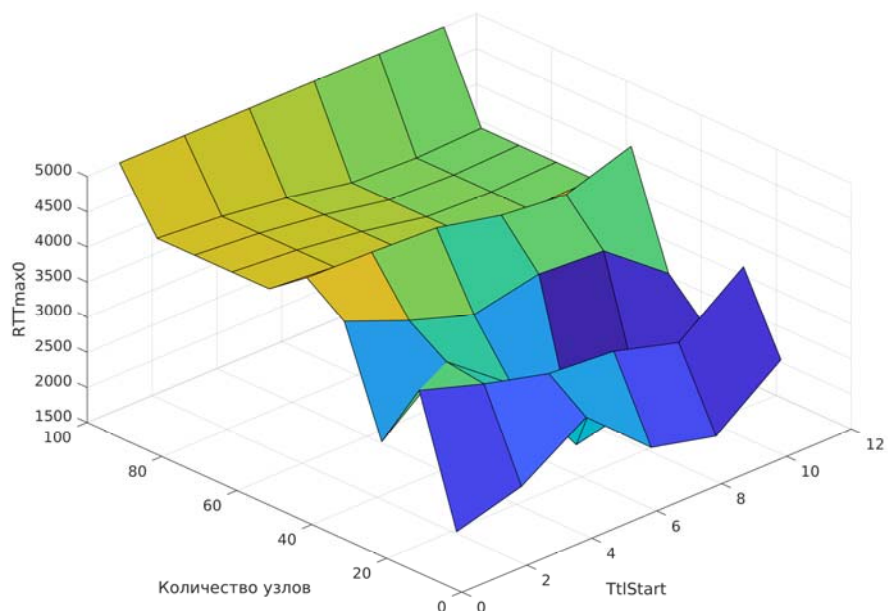


Рис. 2. – Влияние параметра TtlStart на скорость установки соединения в сети с линейным расположением узлов

Предлагаемый подход состоит в том, чтобы над самим протоколом установить управляющую надстройку, которая измеряет параметры текущего соединения и в зависимости от своих настроек меняет значения параметров протокола для следующего соединения (рис. 3).

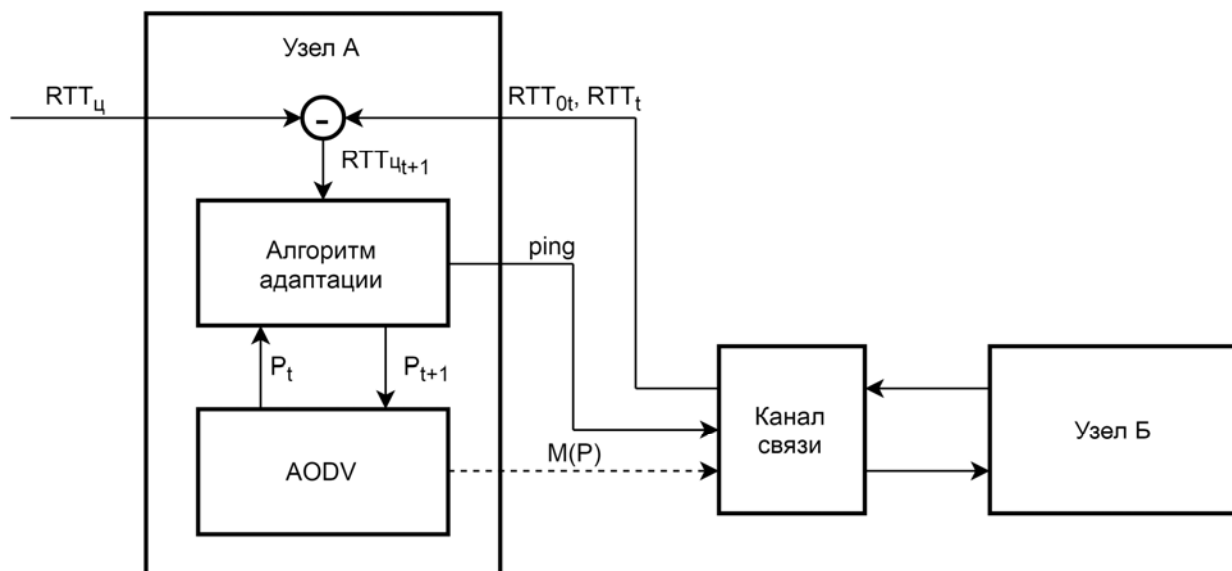


Рис. 3. – Схема функционирования алгоритма адаптации протокола

Здесь RTT – round-trip time (круговая задержка) – время доставки пакета с подтверждением; P – совокупность значений параметров протокола; $M(P)$ – маршрутизация по параметрам P ; t – текущий момент времени.

Надстройка сама по себе не влияет на работу протокола. Когда какое-либо приложение запрашивает передачу данных, AODV, как обычно, выстраивает маршрут до узла назначения. Разница в том, что теперь время построения маршрута фиксируется. Далее, после построения маршрута, включается надстройка: она отправляет по этому маршруту ICMP-запросы (аналогично ping). Объём этих пакетов невелик, поэтому большой нагрузки на канал связи не создаётся. В результате опроса целевого узла можно получить следующие параметры соединения:

- RTT_0 – время установки соединения;
- RTT_{min} – минимальное время доставки пакета;
- RTT_{avg} – среднее время доставки пакета;

- RTT_{\max} – максимальное время доставки пакета (при открытом маршруте);

- LOSS – процент потерь пакетов.

После сопоставления совокупности этих данных RTT_{0t} , RTT_t с целевыми значениями соответствующих параметров соединения $RTT_{ц}$ принимается решение о корректировке значений рабочих параметров протокола. Скорректированные значения параметров применяются к протоколу, но построенные и функционирующие на текущий момент маршруты не разрываются: это не имеет смысла. Вместо этого при очередном построении маршрута начинается новая итерация адаптации, и теперь параметры либо уже оптимальны, либо близки к таковым, поскольку сама сеть в общем случае меняется, но меняется всё же медленнее, чем обновляются маршруты.

Если учесть тот факт, что этот процесс должен выполняться на всех узлах параллельно, получим в результате сеть, где значения параметров на узлах будут различаться градиентно в зависимости от близости к геометрическому центру сети, плотности распределения узлов, количества узлов, средней скорости и так далее.

В качестве основного блока управляющей надстройки целесообразным представляется принять нейронную сеть (рис. 3). Здесь IL – входной слой; PCL – слой с частичной связностью; FCL – полносвязный слой; OL – выходной слой. В данном случае ситуация несколько парадоксальна, поскольку, с одной стороны, речь идёт о задаче регрессии, поэтому обучение без учителя невозможно, с другой, вручную или даже автоматизированно размечать пары «параметры протокола – параметры соединения» таким образом, чтобы выдавать совокупность новых параметров соединения – бессмысленно.

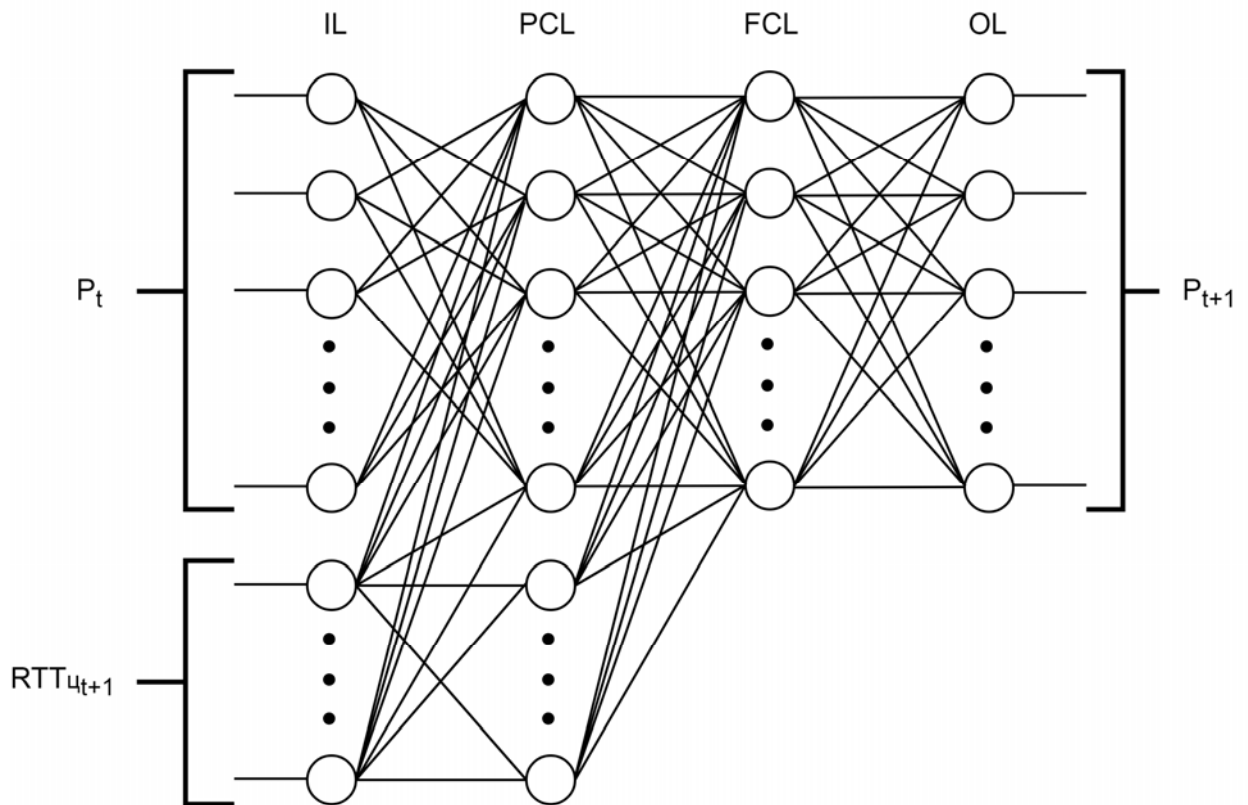


Рис. 4. – Архитектура нейронной сети для управления протоколом

Целесообразно обучать слои данной сети отдельно. PCL, от нейронов которого требуется просто то же значение соответствующего параметра протокола, что и на входе, можно обучить на выборке из симулятора даже без использования алгоритма. FCL и OL обучать придётся уже при симуляции работы алгоритма с обратной связью, но без необходимости обучения всей сети сразу.

Заключение

В рамках настоящей статьи рассмотрены зависимости параметров соединения в мобильных самоорганизующихся сетях от параметров протокола маршрутизации и характеристик самой сети. Предложен алгоритм адаптивной перенастройки параметров протокола AODV в процессе работы сети без необходимости её остановки и вмешательства технических специалистов с перспективной реализацией в виде нейронной сети. В

дальнейшем данный алгоритм будет реализован программно, обучен и всесторонне протестирован.

Литература

1. M. Conti and S. Giordano, "Multi-Hop Ad Hoc Networking: The Reality," IEEE Commun. Mag., Feature Topic on Ad Hoc and Sensor Networks, vol. 45, no. 4, Apr. 2007, pp. 88–95.
2. Perkins C. E., Belding-Royer E., Das S. Ad Hoc On Demand Distance Vector (AODV) Routing, IETF RFC 3561 [2005-11-01]. URL: moment.cs.ucsb.edu/AODV/aodv.html. – 2003.
3. S. Giordano and W. W. Lu, "Challenges in Mobile Ad Hoc Networking," IEEE Commun. Mag., vol. 39, no. 6, 2001, p. 129.
4. Альбекова З. М., Квашурин В. О., Тутик Н. А. Анализ эволюции технологии беспроводных сетей и прогнозы развития инфокоммуникационных сетей в России // Инженерный вестник Дона. 2016. №. 4. URL: ivdon.ru/ru/magazine/archive/n4y2016/3933.
5. Карманов М. Л. Протокол маршрутизации для ad-hoc сетей // Вестник ЮУрГУ. – 2009. - № 26, с. 47-51.
6. M. Conti and S. Giordano, "Mobile Ad Hoc Networking: Milestones, Challenges, and New Research Directories," IEEE Commun. Mag., Feature Topic on Ad Hoc and Sensor Networks, no. 1, Jan. 2014, pp. 85–96.
7. García-Nieto J., Alba E. Automatic parameter tuning with metaheuristics of the AODV routing protocol for vehicular ad-hoc networks //European Conference on the Applications of Evolutionary Computation. – Springer, Berlin, Heidelberg, 2010. – pp. 21-30.
8. Haigh K. Z., Varadarajan S., Tang C. Y. Automatic learning-based MANET cross-layer parameter configuration //Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on. – IEEE, 2006. – pp. 84-84.



9. Ledy J. et al. AODV enhancements in a realistic VANET context //Wireless Communications in Unusual and Confined Areas (ICWCUCA), 2012 International Conference on. – IEEE, 2012. – pp. 1-5.

10. Бахтин А. А., Меркушев В. А. Метод локального восстановления маршрута в эпизодических сетях // Инженерный вестник Дона, 2011, №3 URL: ivdon.ru/ru/magazine/archive/n3y2011/467.

References

1. M. Conti and S. Giordano, “Multi-Hop Ad Hoc Networking: The Reality,” IEEE Commun. Mag., Feature Topic on Ad Hoc and Sensor Networks, vol. 45, no. 4, Apr. 2007, pp. 88–95.

2. Perkins C. E., Belding-Royer E., Das S. Ad Hoc On Demand Distance Vector (AODV) Routing, IETF RFC 3561. [2005-11-01]. URL: moment.cs.ucsb.edu/AODV/aodv.html. 2003.

3. S. Giordano and W. W. Lu, “Challenges in Mobile Ad Hoc Networking,” IEEE Commun. Mag., vol. 39, no. 6, 2001, p. 129.

4. Albekova Z.M., Kvashurin V.O., Tutik N.A. Inzhenernyj vestnik Dona (Rus), 2016, №4. URL: ivdon.ru/ru/magazine/archive/n4y2016/3933.

5. Karmanov M. L. Vestnik SUSU. 2009. №26, pp. 47-51.

6. M. Conti and S. Giordano, “Mobile Ad Hoc Networking: Milestones, Challenges, and New Research Directories,” IEEE Commun. Mag., Feature Topic on Ad Hoc and Sensor Networks, no. 1, Jan. 2014, pp. 85–96.

7. García-Nieto J., Alba E. European Conference on the Applications of Evolutionary Computation. – Springer, Berlin, Heidelberg, 2010. pp. 21-30.

8. Haigh K. Z., Varadarajan S., Tang C. Y. Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on. IEEE, 2006. pp. 84-84.

9. Ledy J. et al. Wireless Communications in Unusual and Confined Areas (ICWCUCA), 2012 International Conference on. IEEE, 2012. pp. 1-5.



10. Bahtin A. A., Merkushev V. A. Inzhenernyj vestnik Dona (Rus), 2011, №3. URL: ivdon.ru/ru/magazine/archive/n3y2011/467.