基于深度强化学习的物联网智能路由策略

丁瑞金[1]，高飞飞[1]，邢玲[2]
（1. 清华大学自动化系，北京 100084；2. 河南科技大学，河南 洛阳 471023）

摘　要：随着物联网时代的到来，万物互联的传输模式引发数据量爆炸式增长，给传统路由协议带来了严峻挑战。阐述了在数据量急剧增长的情况下，已有路由协议的局限性，并将路由选择问题重新建模为马尔可夫决策过程。在此基础上，采用深度强化学习方法为每项数据传输任务选择下一跳路由器，从而在避免数据堵塞的前提下尽可能缩短数据传输路径长度。仿真结果表明，所提方法能够显著降低数据堵塞概率，增大网络吞吐量。

关键词：深度强化学习；路由；物联网；网络堵塞
中图分类号：TN915
文献标识码：A
doi: 10.11959/j.issn.2096−3750.2019.00097

Intelligent routing strategy in the Internet of things based on deep reinforcement learning

DING Ruijin[1], GAO Feifei[1], XING Ling[2]
1. Department of Automation, Tsinghua University, Beijing 100084, China
2. Henan University of Science and Technology, Luoyang 471023, China

Abstract: At the era of the Internet of things, networking mode that connects everything would bring tremendous increase in the data volume and challenge the traditional routing protocols. The limitations of the existing routing protocols was analyzed when facing the data explosion and then the routing selection problem was re-modeled as a Markov decision process. On this basis, the deep reinforcement learning technique was utilized to choose the next-hop router for data transmission task in order to shorten the transmission path length while network congestion was avoided. The simulation results demonstrate that the congestion probability can be reduced significantly and the network throughput can be enhanced by the proposed strategy.

Key words: deep reinforcement learning, routing, Internet of things, network congestion

1　引言

随着信息技术的不断发展，数据的传输和交换已不仅是计算机、智能手机等特定设备之间的行为，越来越多的设备甚至物品将通过各种方式和接口接入互联网，“物联网”概念因此兴起。所谓“物联网”[1]，是一种将身边的一切物品都纳入网络的技术，能够提高人们感知周围环境、了解事物状态的能力，为生活带来便利。环境反向散射技术[2-3]的发展使物联网规模进一步扩大到微小无源节点。据统计，2017 年物联网设备数量已达 84 亿，超过了目前全球人口数量总和，预计在 2020 年将达 300 亿[4]。高速率物联网器件的研究[5]使物联网将迎来数据量爆炸式增长，这对传统路由协议提出了严峻挑战。

传统路由协议如 OSPF、IS-IS 和 RIP 等[6-8]基于计算最短路径[9]原理进行数据传递，其在路由选择时未考虑每个路由器剩余的缓存大小等信息。当数据量急剧增长时，可能出现某一个或多个路由器被多条数据传输任务同时选中的情况，这将造成网络数据堵塞，降低网络吞吐量，增加数据传输时延。现有的路由协议未加入智能元素，

收稿日期：2018−10−25；修回日期：2019−03−07

不能根据网络的实际状态来调整路由策略。

近年来，随着计算能力的增长，人工智能技术得到了飞速发展。其中，最有代表性的技术之一是深度学习[10]，已在图像处理领域得到成功应用。鉴于深度学习的强大能力，学者尝试将其应用到网络路由选择问题上。Kato 等[11]利用深度神经网络来模拟传统路由协议，使得网络不需要通过互相通信获取整个网络的拓扑结构并计算最短路径，而是可以直接通过神经网络获取下一跳路由器选择。然而，这并不能解决传统路由协议在面对数据量增长时发生的堵塞问题。Tang 等[12]提出利用卷积神经网络判断当前所选的路径组合是否会引起堵塞，如果当前所选路径会造成堵塞，则重新选择路径。这种方法能够有效降低网络堵塞的概率，但缺点是需要为每一条可能的传输路径组合训练神经网络，且对于路径有不能回传的限制。当网络规模增大时，所需的神经网络数目将呈指数级增长，因此，当面对物联网等未来大规模网络时，这种方法较难实现。

深度神经网络本质上是对函数关系的拟合，在路由选择这种决策类问题中并不适用，而另一种人工智能技术即强化学习[13]，则被认为更适合解决决策类问题。参考文献[14]中提出用 Q 学习方法选择路由节点。然而，传统强化学习往往只能解决状态空间和动作空间较小的问题，而在大规模网络中，状态空间或动作空间往往巨大，甚至趋近于无穷。在路由选择问题中，路由器剩余缓存大小的可能情况很多，使得直接应用传统强化学习方式不能很好地解决路由选择问题。因此，本文将通过结合深度强化学习来解决由于数据量增多引起的网络堵塞概率过高问题。目前，尚没有其他方法采用深度强化学习来解决路由问题。

2　问题描述与系统模型

为了更好地描述路由选择问题及所提算法，本文考虑了网络结构如图 1 所示。数据源如计算机、基站和服务器等产生数据后，将其送入相连接的源路由器中。假设整个网络以时隙为单位进行路由选择，每个时隙长度设为 1。在每个时隙内，每项数据传输任务均选择下一跳路由器，并将数据分组送出，存储在下一跳路由器缓存中。上述过程不断重复直到数据分组到达目标路由器。当所要传输的数据分组大小超过了下一跳路由器剩余缓存大小时，就会导致网络堵塞。

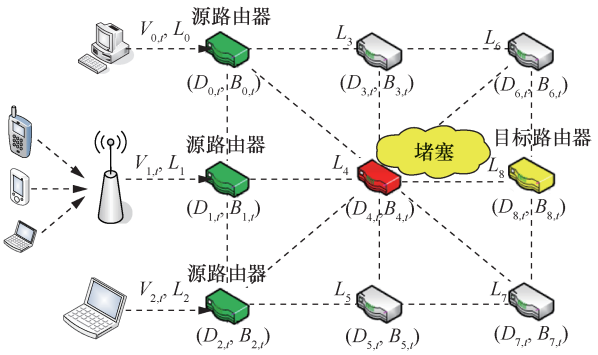

图 1　网络结构

在图 1 所示的网络结构中，L_0、L_1、L_2 作为源路由器接收数据源产生的数据，并将其传送到目标路由器 L_8 中。若 L_0、L_1、L_2 均选择最短路径，则 L_4 将同时被 3 路数据选作下一跳路由器。若 3 路数据总量与路由器缓存相比较小时，则不会引起网络堵塞；若数据源产生的数据分组很大，使得 L_4 的缓存不足以存储数据时，就会发生堵塞，这种情况在未来网络大数据流量下极易出现。此时，传统协议一般会重新选择替代的路由器（在图 1 中为 L_3 和 L_5），这会增加传输时延。若未来出现类似甚至相同的情况，即 L_0、L_1、L_2 再次同时传输大量数据到 L_8，传统路由协议仍会为 3 路数据选择 L_4，将再次引起堵塞。即传统路由协议不能从曾经发生的堵塞中学习经验教训，因此，不能根据实际网络状态选择跳转的路由器。

网络中的路由器数量设为 N，路由器集合记为 $\mathcal{L}$。$\mathcal{L}$ 能够划分为彼此不相交的 3 个集合 $\mathcal{L}=\mathcal{L}_s \cup \mathcal{L}_d \cup \mathcal{L}_n$，其中，$\mathcal{L}_s$、$\mathcal{L}_d$、$\mathcal{L}_n$ 分别表示源路由器、目标路由器和普通路由器的集合，并且 N_s、N_d、N_n 分别表示三者的数量。所有路由器均参与数据传输过程，网络是否堵塞取决于每个路由器中存储的数据大小和每个路由器中剩余的缓存大小。令 $D_t=\{D_{1,t},\cdots,D_{N,t}\}$ 表示每个路由器在 t 时隙的数据总量，$B_t=\{B_{1,t},\cdots,B_{N,t}\}$ 表示每个路由器在 t 时隙的剩余缓存大小。数据源在每个时隙新产生的数据会直接进入网络，也会对路由的选择产生影响。令 $V_t=\{V_{1,t},\cdots,V_{N_s,t}\}$ 表示 t 时隙新产生的数据，数据产生过程假设为泊松过程。则在 t 时隙内，整个网络的共同状态被定义为 (V_t,D_t,B_t)。数据的传输过程将改变网络的共同状态，如一个大小为 f 的数据分组从 L_i 传到 L_j，则在下一时隙 $t+1$，L_i 和 L_j 所缓存的数据量以及其剩余缓存的大小 $(D_{i,t+1},D_{j,t+1},B_{i,t+1},B_{j,t+1})$ 会有 6 种情况，将 $(D_{i,t+1},D_{j,t+1},B_{i,t+1},B_{j,t+1})$ 记为 K，如式(1)所示。

$$K=\begin{cases}(D_{i,t}+V_{i,t}-f,D_{j_t},B_{i,t}-V_{i,t}+f,B_{j,t}) & ,i\in\mathcal{L}_s,j\in\mathcal{L}_d\\ (D_{i,t}+V_{i,t}-f,D_{j,t}+V_{j,t}+f,B_{i,t}-V_{i,t}+f,B_{j,t}-SV_{j,t}-f) & ,i\in\mathcal{L}_s,j\in\mathcal{L}_s\\ (D_{i,t}+V_{i,t}-f,D_{j,t}+f,B_{i,t}-V_{i,t}+f,B_{j,t}-f) & ,i\in\mathcal{L}_s,j\in\mathcal{L}_n\\ (D_{i,t}-f,D_{j_t},B_{i,t}+f,B_{j,t}) & ,i\notin\mathcal{L}_s,j\in\mathcal{L}_d\\ (D_{i,t}-f,D_{j,t}+V_{j,t}+f,B_{i,t}+f,B_{j,t}-V_{j,t}-f) & ,i\notin\mathcal{L}_s,j\in\mathcal{L}_s\\ (D_{i,t}-f,D_{j,t}+f,B_{i,t}+f,B_{j,t}-f) & ,i\notin\mathcal{L}_s,j\in\mathcal{L}_n\end{cases}\tag{1}$$

除了网络的共同状态，对于每个数据传输任务而言，其数据分组的位置和大小也将影响下一跳路由器的选择。为了表示这些特征，采用改进的独热编码。用一个长度为 N 的向量 $\boldsymbol{P}_t$ 表示 t 时隙数据分组大小和所处的位置。当数据分组在路由器 i 时，向量的第 i 个元素为该数据分组大小，其余元素均为 0。独热编码能够帮助计算机理解数据分组的位置和大小信息。对于每个数据传输任务来说，其状态可以表示为数据分组的位置、大小信息与网络共同状态的和，即 $S_t=(V_t,D_t,B_t,\boldsymbol{P}_t)$。

由上述讨论得知，网络可以用有向图 $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$ 表示，其中，$\mathcal{V}$ 表示路由器节点的集合，$\mathcal{E}$ 表示路由器之间链路的集合。数据传输任务依据网络的状态以及数据分组的位置和大小来选择动作，即选择当前路由器与下一跳路由器之间的链路。如 L_i 中的数据要传输到 L_j，即 $\text{link}(i,j)\in\mathcal{E}$ 被选为要执行的动作。此外，数据可以从 L_i 传输到 L_j，也可同时从 L_j 传输到 L_i。然而，对于数据传输任务来说，并不是所有链路都可以被选择，只能选择连接当前数据分组所在位置的链路作为有效动作。因此，在数据传输过程中，有效动作集合会随着数据分组位置的转移而改变。

3　路由选择马尔可夫决策过程

为了使用强化学习方法解决路由选择问题，需要将路由选择问题建模为马尔可夫决策过程。马尔可夫决策过程是包含奖励和决策的马尔可夫过程，可以用一个元组 $<\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma>$ 表示，具体如下。

1) $\mathcal{S}$ 表示所有状态的集合。

2) $\mathcal{A}$ 表示所有动作的集合，即网络结构中的所有链路，包括正向链路和反向链路。

3) $\mathcal{P}$ 是状态转移概率矩阵，由于传输本身是确定性过程，所以状态转移概率矩阵主要由数据分组产生的概率分布决定。

4) $\mathcal{R}(s,a,s')$ 是由状态 s 执行动作 a，然后转换成状态 s' 情况下所收获的奖励。

5) $\gamma\in[0,1)$ 是衰减系数，表示过去的奖励和现在的奖励之间的比重关系。

马尔可夫决策过程如图 2 所示，任务在每个时隙内基于当前状态选择动作，执行完该动作后，路由网络将下一状态和相应的奖励反馈给任务。不断重复上述选择动作，反馈奖励的过程就是马尔可夫决策过程。

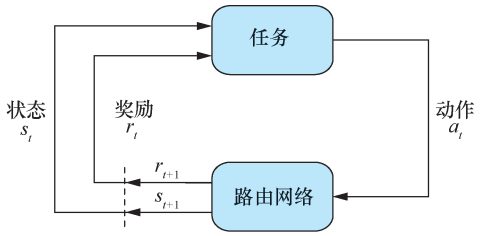

图 2　马尔可夫决策过程

3.1　马尔可夫性

马尔可夫性代表未来仅与现在有关而与过去无关，即过去的所有信息都蕴含在现在的状态中。强化学习的马尔可夫性一般表示为

$$\Pr(s_{t+1}\mid s_0,a_0,s_1,\cdots,s_t,a_t)=\Pr(s_{t+1}\mid s_t,a_t)\tag{2}$$

在路由选择问题中，由于当前状态是对过去状态的汇总，所以具有马尔可夫性。

3.2　奖励函数

奖励函数是对在每个状态下执行动作的量化评估。由于算法的目的是要尽量避免发生堵塞而不是一味地追求最短路径，因此，奖励函数应被设计为鼓励避免网络堵塞并在此基础上寻求最短路径，即奖励函数应该惩罚引起网络堵塞的动作。如每个任务只能选择从当前数据分组所在路由器出发的链路，所以奖励函数也应该惩罚无效的动作。此外，奖励函数还要能够统计路径的长度。综上所述，奖励函数 $\mathcal{R}(s,a,s')$ 应设置为

$$\mathcal{R}(s,a,s')=\begin{cases} r_c, & \text{发生堵塞} \\ r_e, & a\text{是无效动作} \\ 0, & \text{到达目标路由器} \\ -1, & \text{其他} \end{cases} \tag{3}$$

其中，r_c 是当动作 a 引起网络堵塞时获得的奖励，r_e 是当动作 a 是无效动作时获得的奖励。r_c 和 r_e 均是绝对值较大的负数，其目的是尽量避免堵塞并选择有效动作。−1 用来记录数据分组在网络中跳转的次数。所以为了能够尽量避免堵塞和无效动作的选取，需要最大化累积期望奖励，用 $\mathcal{R}_t=\mathcal{R}(s_t,a_t,s_{t+1})$ 表示在 t 时隙的奖励。

3.3　价值函数

价值函数用来量化每个状态的价值，G_t 表示在 t 时隙的累积衰减奖励。

$$G_t=R_{t+1}+\gamma R_{t+2}+\cdots+\gamma R_{t+T}=\sum_{k=0}^{T-1}\gamma^k R_{t+k} \tag{4}$$

其中，T 为终止步数。

策略 π 是指给定状态下动作的概率分布。策略决定了在当前状态下数据任务该如何选择下一跳路由器，并且这个策略是稳定的，不随时间而改变。定义 v_π 为在策略 $\pi(a\,|\,s)$ 下的状态价值函数，这是累积衰减奖励从状态 s 开始在当前策略下的期望值，即

$$v_\pi(s)=\mathrm{E}_\pi[G_t\,|\,S_t=s] \tag{5}$$

强化学习算法的目标是寻找一种策略能够最大化价值。此外，价值函数还有一种形式被称为动作状态价值函数。动作状态价值函数较直观，其直接衡量在状态 s 下每个动作的价值，即在状态 s 下执行动作 a 所获得的累积衰减奖励的期望。定义为

$$Q_\pi(s,a)=\mathrm{E}_\pi[G_t\,|\,S_t=s,A_t=a] \tag{6}$$

动作状态价值函数可以被分为两个部分。

1) 下一时隙的直接奖励。

2) 衰减后下一状态下的下一动作价值。

$$Q_\pi(s,a)=\mathrm{E}_\pi[R_{t+1}+\gamma Q_\pi(S_{t+1},A_{t+1})\,|\,S_t=s,A_t=a] \tag{7}$$

最优动作状态价值函数是所有策略中使得动作状态价值函数最大化的函数，即

$$Q_*(s,a)=\max_\pi Q_\pi(s,a) \tag{8}$$

最优动作状态价值函数存在且唯一，相关证明在 3.4 节。显然，如果能够知晓最优动作状态价值函数，则在每一状态下直接选择价值最高的动作就能获得最优策略。即最优策略为

$$\pi_*(a\,|\,s)=\begin{cases} 1, & a=\arg\max\limits_{a\in\mathcal{A}} Q_*(s,a) \\ 0, & \text{其他} \end{cases} \tag{9}$$

3.4　贝尔曼方程

根据式(7)，将最优动作状态价值函数 $Q_*(s,a)$ 分解为两部分

$$Q_*(s,a)=R(s,a)+\gamma\mathrm{E}_{s'\sim\mathcal{P}_{ss'}^a}[\max_{a'}Q_*(s',a')] \tag{10}$$

式(10)被称为贝尔曼方程。定义贝尔曼算子 $\mathcal{B}$ 是将 Q 转换为定义在 $\mathcal{S}\times\mathcal{A}$ 空间上的函数 $\mathcal{B}Q$

$$\mathcal{B}Q_*(s,a)=R(s,a)+\gamma\mathrm{E}_{s'\sim\mathcal{P}_{ss'}^a}[\max_{a'}Q_*(s',a')] \tag{11}$$

结合式(10)和式(11)，贝尔曼方程可以被重新表示为 $Q_*=\mathcal{B}Q_*$，因此，最优动作状态价值函数的存在性和唯一性等价于贝尔曼方程不动点的存在性和唯一性。为了证明不动点的存在性和唯一性，引入引理 1。

引理 1　贝尔曼算子 $\mathcal{B}$ 具有收缩性，即对任意两个不同的动作状态价值函数 Q_1 和 Q_2，$\|\mathcal{B}Q_1-\mathcal{B}Q_2\|_\infty\leqslant\gamma\|Q_1-Q_2\|_\infty$ 成立。其中，$\|\bullet\|_\infty$ 表示无穷范数。

证明　对于任意 $(s,a)\in\mathcal{S}\times\mathcal{A}$，有

$$\begin{aligned} & \mathcal{B}Q_1(s,a)-\mathcal{B}Q_2(s,a) \\ = & \left(R(s,a)+\gamma\mathrm{E}_{s'\sim\mathcal{P}_{ss'}^a}[\max_{a'}Q_1(s',a')]\right)- \\ & \left(R(s,a)+\gamma\mathrm{E}_{s'\sim\mathcal{P}_{ss'}^a}[\max_{a'}Q_2(s',a')]\right) \\ = & \gamma\mathrm{E}_{s'\sim\mathcal{P}_{ss'}^a}\left[\max_{a'}Q_1(s',a')-\max_{a'}Q_2(s',a')\right] \\ \leqslant & \gamma\mathrm{E}_{s'\sim\mathcal{P}_{ss'}^a}\left[Q_1(s',a_1(s'))-Q_2(s',a_1(s'))\right] \\ & \left(\text{where } a_1(s')=\arg\max_{a'}Q_1(s',a')\right) \\ \leqslant & \gamma\max_{s',a'}\left(Q_1(s',a')-Q_2(s',a')\right) \\ \leqslant & \gamma\|Q_1-Q_2\|_\infty \end{aligned} \tag{12}$$

同理，$\mathcal{B}Q_2(s,a)-\mathcal{B}Q_1(s,a)\leqslant\gamma\|Q_1-Q_2\|_\infty$，因此，有 $\|\mathcal{B}Q_1-\mathcal{B}Q_2\|_\infty\leqslant\gamma\|Q_1-Q_2\|_\infty$。

定理 1　动作状态价值函数存在且唯一。

证明　如上所述，最优动作状态价值函数的存在性和唯一性等价于贝尔曼方程的存在性和唯一性。

存在性　对于任意 Q_1 和 Q_2，由于引理 1 中所证明的贝尔曼算子的衰减性，有

$$\begin{aligned}\| \mathcal{B}^k Q_1 - \mathcal{B}^k Q_2 \|_\infty &\leqslant \gamma \| \mathcal{B}^{k-1} Q_1 - \mathcal{B}^{k-1} Q_2 \|_\infty \\ &\leqslant \cdots \leqslant \gamma^{k-1} \| \mathcal{B} Q_1 - \mathcal{B} Q_2 \|_\infty \\ &\leqslant \gamma^k \| Q_1 - Q_2 \|_\infty \xrightarrow{k\to\infty} 0\end{aligned} \tag{13}$$

所以，$\mathcal{B}^k Q_1$ 和 $\mathcal{B}^k Q_2$ 均会分别收敛到一些不动点 $\bar{Q}$，不动点的存在性得证。

唯一性　采用反证法，假设贝尔曼方程有两个不同的不动点 $\bar{Q}_1 \neq \bar{Q}_2$，根据引理 1，有

$$\begin{aligned}\| \bar{Q}_1 - \bar{Q}_2 \|_\infty &= \| \mathcal{B} Q_1 - \mathcal{B} Q_2 \|_\infty \\ &\leqslant \gamma \| Q_1 - Q_2 \|_\infty\end{aligned} \tag{14}$$

进一步，可得出

$$\begin{aligned}\| \bar{Q}_1 - \bar{Q}_2 \|_\infty &\leqslant \gamma \| \bar{Q}_1 - \bar{Q}_2 \|_\infty \\ &\leqslant \cdots \leqslant \gamma^k \| Q_1 - Q_2 \|_\infty \xrightarrow{k\to\infty} 0\end{aligned} \tag{15}$$

注意 $\gamma \in [0,1)$，因此 $\| \bar{Q}_1 - \bar{Q}_2 \|_\infty = 0$，所以 $\bar{Q}_1 = \bar{Q}_2$，此结论与有两个不同不动点的假设矛盾。

综上所述，贝尔曼方程的不动点存在且唯一，所以最优动作状态价值函数存在且唯一。

3.5　Q 学习

直接计算最优动作状态价值函数 Q_* 非常困难，可以通过 Q 学习用迭代方法近似 Q_*。其迭代步骤为

$$\begin{aligned}&Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \\ &\alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]\end{aligned} \tag{16}$$

其中，α 为更新步长。式(16)中的迭代更新步骤使得动作状态价值函数 $Q(s,a)$ 向 $R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$ 更新。即用下一状态下的动作价值来更新动作状态价值函数，此方法被称为自举法。

为了尽可能准确地获得某一状态下每个动作的价值，强化学习算法必须尝试各种可能动作。但是如果算法仅选择最大价值的状态，则没有被试过的动作将很难再被选到。因此，除了利用已经知道的知识，还要学会探索新动作。ϵ−贪婪策略经常被用来实现探索。

$$a_t = \begin{cases} \arg\max_a Q(s,a), & \text{概率 } 1-\epsilon \\ \text{完全随机选取}, & \text{概率 } \epsilon \end{cases} \tag{17}$$

其中，ϵ 是随机选取动作的概率，对于动作的完全随机选取保证了充足的探索。并且 Q 学习是一种离线策略学习算法，即估计和更新动作状态价值函数所使用的策略不一致。具体来说，在 Q 学习中，实际执行的动作是采用 ε−贪婪策略选取的，而更新时采用贪婪策略估计下一动作的价值，如式(16)所示。

4　基于深度 Q 网络的路由选择算法

根据前文对状态的定义，可以发现状态空间非常大，导致 Q 学习方法难以应用，因此借助基于深度 Q 网络（DQN, deep Q-network）[15]来解决这一问题。

4.1　DQN

DQN 是 Q 学习算法的拓展，而 Q 学习算法本质上是一种查表法，当状态空间 $\mathcal{S}$ 非常大时，查表法不适用。根据前文对 V_t、D_t、B_t 和 $\boldsymbol{P}_t$ 的定义可知，状态的可能情况非常多。DQN 可借助深度神经网络来表示动作状态价值函数，从而解决状态空间过大的问题。

令 θ 表示神经网络参数，则动作状态价值函数可以表示为 $Q(s,a;\theta)$。DQN 中神经网络结构如图 3 所示，神经网络的输入为状态，输出是每个动作的价值。当神经网络结构确定后，θ 可以直接代表整个动作状态价值函数。因此，对动作状态价值函数的更新就是对 θ 的更新。

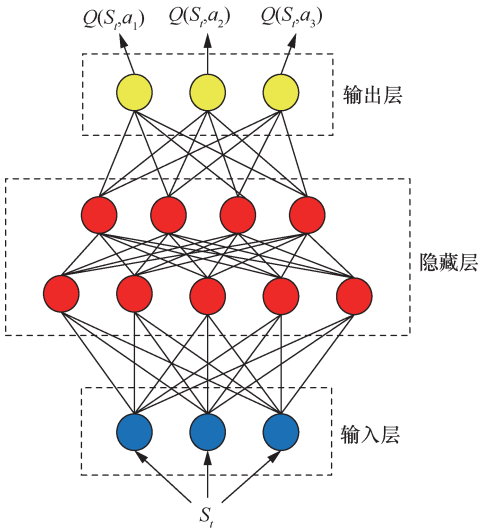

图 3　DQN 中神经网络结构

DQN 采用自举法产生训练目标，即优化 $R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta)$ 和 $Q(S_t, A_t; \theta)$ 之间的误差。因此，损失函数定义为

$$J(\theta) = [R_{t+1} + \gamma \max_\alpha Q(S_{t+1}, \alpha; \theta) - (S_t, A_t; \theta)]^2 \tag{18}$$

对网络参数的更新步骤为

$$\begin{aligned}&\theta_{t+1} = \theta_t + \nabla Q(s,a;\theta) \times \\ &\alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta) - Q(S_t, A_t; \theta) \right]\end{aligned} \tag{19}$$

其中，α 为学习率。

当训练深度神经网络时，假设输入数据是独立同分布的。然而，强化学习的训练数据源于自身，如果按照路由选择过程的时间顺序产生的数据来训练，则数据之间具有很强的相关性。因此，需要采用经验回放方式来减轻相关性。具体来说，将路由选择过程中出现的状态、动作以及奖励值等存入记忆池，训练时随机从记忆池中选取数据进行训练，能够很好地降低因时间间隔太近带来的相关性。

如式(19)所示，计算目标与计算当前动作状态价值函数的参数 θ 一样，同样会带来相关性。因此设置一个目标网络，其网络参数为 θ^{-}。目标网络 $\hat{Q}$ 每隔 N_u 步从原始估计动作状态价值函数的神经网络复制一次参数，因此对动作状态价值函数的更新可改为

$$\begin{aligned}&\theta_{t+1}=\theta_t+\nabla Q(s,a;\theta)\times\\&\alpha\left[R_{t+1}+\gamma\max_a\hat{Q}(S_{t+1},a;\theta)-Q(S_t,A_t;\theta)\right]\end{aligned}\tag{20}$$

4.2　基于 DQN 的路由选择算法

DQN 是一种对单智能体的强化学习算法，在路由选择问题中，每个时隙内会有若干任务要选择下一跳路由器。为了解决这一问题，假设存在有足够计算能力的中央控制器，能够获取数据源产生的数据信息并指挥每个路由器将缓存中的数据发送给下一跳路由器。数据传输是一个确定性过程，即动作 a 所引起的对于每个路由器数据量 D_t 和剩余缓存大小 B_t 的改变通过式(1)求得，而不需要与每个路由器进行通信。因此，得知网络的最初状态和每个时隙新产生的数据分组即可获知每个时隙的网络状态。

即使两个数据传输任务的数据分组大小一样并且存储于同一路由器中，也会由于其目标不同而选择不同的数据传输链路。即使神经网络的输入状态均相同，其输出的动作也不一定相同。因此，按照数据可能传输的目标路由器个数来设置神经网络，即 N_d 个神经网络。传向同一目标路由器的数据传输任务采用同一个神经网络计算动作状态价值函数，并以此选择数据。

在中央控制器中设置一个任务队列 QT，用来存储需要完成的数据传输任务。在时隙 t 的最开始，数据源产生数据分组并将其传入所连接的源路由器中。中央控制器获取数据传输任务并将其添加到 QT 末尾，同时根据新产生的数据计算当前每个路由器保存的数据总量 D_t 和剩余缓存大小 B_t。此外，由于中央控制器知道数据传输任务的位置和大小，因此对每个数据传输任务来说，其状态已知。

依次从 QT 队列中选取任务，并根据其目标路由器选取对应的神经网络，输入状态并输出每个动作的价值，然后采用 $\epsilon-$贪婪策略选择动作。若选择的动作无效，即不是从当前位置出发的链路，则中央控制器先将此情况当作终止情况，并将反馈奖励值 r_e 保存到记忆池作为训练数据，然后重新选择价值最高的有效动作继续数据传输任务，以保证每次实际选择链路的有效性。根据这个动作，中央控制器很容易计算该数据传输任务的下一状态，可能的情况如下。

1) 如果下一跳路由器是目标路由器，则数据传输任务完成，奖励值为 0。

2) 如果该动作引起了网络堵塞，则任务被终止并被抛弃，产生奖励值 r_c。

3) 在其他情况下，中央控制器更新完状态后将当前任务重新添加到队列 QT 末尾，并且返回奖励值−1。

中央控制器将元组状态、动作以及奖励（下一状态）存入记忆池。神经网络从对应的记忆池中随机采样进行训练，重复上述过程直到队列中的每个任务都选择了一个动作。最后，中央控制器将当前时隙对每个任务的指令传给所有路由器，路由器根据指令将存储在其缓存中的数据分组发送到对应的下一跳路由器中。

5　仿真与分析

本文采用深度学习框架 Keras（TensorFlow 后端）编写，计算平台为英特尔酷睿 i7-8700k CPU，内存为 32 GB，CPU 为英伟达 GTX1070，操作系统为 Ubuntu 16.04。

本节从复杂度的角度比较了所提 DQN 方法与基于深度学习方法，还比较了传统基于最短路径的路由协议与所提基于 DQN 的路由算法的表现。考虑如图 1 所示的网络结构，L_0、L_1、L_2 接收数据源产生的数据，其数据产生过程为泊松过程，每个路由器的缓存大小设置为 45 MB，将用于探索的 $\epsilon-$贪婪策略中的 ϵ 值设置为 0.9。根据之前对状态的定义，神经网络的输入层有 $3\times N+N_s$ 个神经元，

输出层有 N_a 个神经元。

神经网络数目比较如表 1 所示，表 1 比较了本文提出的 DQN 方法与参考文献[12]中基于深度学习方法所需的神经网络数目。参考文献[12]中的方法首先需要人为确定可能的路径组合，然后为每一种路径组合训练一个神经网络，以判断当前网络状态下该路径组合是否会引起网络堵塞。假设路径不允许回传，图 1 中 L_0 产生的数据一旦往右或者往下传输后，便不允许再往上或者往左传；如果允许回传，可能的路径组合将有无穷种，显然不能应用到实际问题中。基于不能回传的设定，L_0 产生的数据共有 $L_0 \to L_4 \to L_8$ 、$L_0 \to L_3 \to L_6 \to L_8$ 、$L_0 \to L_3 \to L_4 \to L_8$ 和 $L_0 \to L_1 \to L_4 \to L_8$ 4 条路径可供选择，对称地，L_2 产生的数据也有 4 条路径，而 L_1 产生的数据只有 $L_1 \to L_4 \to L_8$ 一条路径。即便如此，该算法也需要为 $4 \times 1 \times 4 = 16$ 种路径组合训练 16 个神经网络。当网络规模增大时，需要训练的神经网络数目呈指数级增长。本文提出的基于 DQN 的路由算法需要训练的神经网络数目与网络中目标路由器的数目保持一致，大大增强了算法的扩展性。由于需要运算的神经网络数目很少，因此训练完成后，网络的运算量较低，与传统路由器相比，不会明显增加所需要的运算量。

表 1　　神经网络数目比较

方法	神经网络数目
本文算法	1
参考文献[12]不考虑回传	16
参考文献[12]考虑回传	∞

堵塞概率与训练轮数关系如图 4 所示。图 4 比较了传统路由协议与基于 DQN 的路由选择算法引起网络堵塞的概率，衰减系数γ设置为 0.9，数据产生的泊松过程参数为 15 MB。如图 4 所示，传统路由协议的堵塞概率很高，与之形成对比的是基于 DQN 的路由选择算法的网络堵塞概率显著降低，并维持在一个很低的水平，这是因为后者能够从过去发生的堵塞中吸取教训，并找到规避堵塞的方法。ϵ–贪婪策略具有随机性，会增大堵塞概率。在实际训练中为了探索，仍采取 ϵ –贪婪策略。采取贪婪策略的堵塞概率比采取 ϵ –贪婪策略的堵塞概率略低。当神经网络训练完成后，在实际网络运行时，使用贪婪策略，探索仅针对训练阶段。

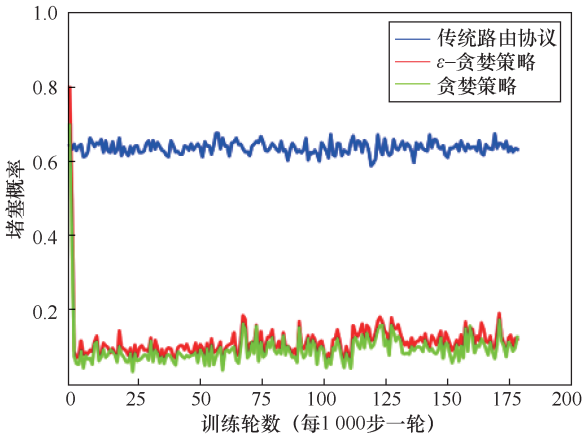

图 4　堵塞概率与训练轮数关系

堵塞概率与数据产生速率关系如图 5 所示。其中，基于 DQN 的路由选择算法的曲线是由经过一定训练轮数后的神经网络计算得到。当数据产生速率较低时，两种方法的堵塞概率都很低；当数据产生速率升高时，传统路由协议的堵塞概率上升很快，而基于 DQN 的路由选择算法则能维持在较低水平。

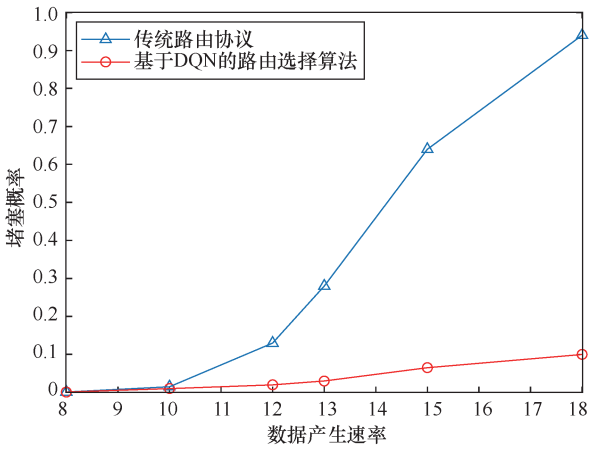

图 5　堵塞概率与数据产生速率关系

网络吞吐量与数据产生速率关系如图 6 所示。当网络空闲时，传统路由协议与本文提出的算法性能相差无几；当网络负载加大时，由于堵塞概率增加，导致尽管数据产生速率提高了，但吞吐量反而下降。基于 DQN 的路由选择算法由于堵塞概率并没有显著升高，其吞吐量仍然随着数据产生速率的提高而增大。

基于 DQN 的路由选择算法的无效动作概率与训练轮数关系如图 7 所示。尽管选到无效动作后会重新选取有效动作，但是这会增加计算量。可以看到，经过训练后，基于 DQN 的路由选择算法几乎不会再选到无效动作。

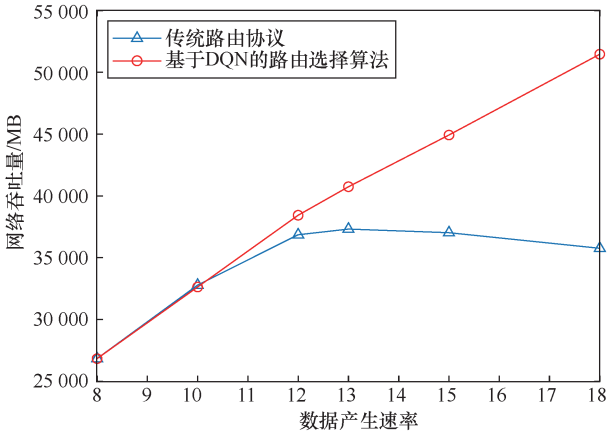

图 6　网络吞吐量与数据产生速率关系

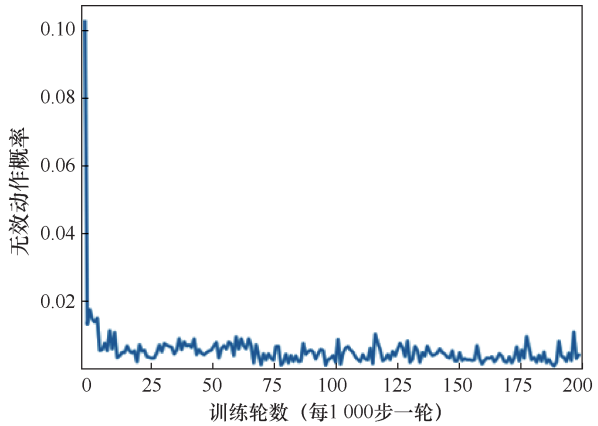

图 7　基于 DQN 的路由选择算法的无效动作概率与训练轮数关系

6　结束语

本文针对由于物联网以及未来大数据时代数据量爆炸式增长而带来的网络堵塞问题，提出了一种智能路由算法。借助深度强化学习，能够根据当前的网络状态动态选择传输的跳转路由器，从而降低堵塞概率，并提高网络吞吐量。

参考文献：

[1] 孙其博, 刘杰, 黎羴, 等. 物联网: 概念、架构与关键技术研究综述[J]. 北京: 北京邮电大学学报, 2010, 33(3): 1-9.
SUN Q B, LIU J, LI S, et al. Internet of things: summarize on concepts, architecture and key technology problem[J]. Beijing: Journal of Beijing University of Posts and Telecommunications, 2010, 33(3): 1-9.

[2] LIU V, PARKS A, TALLA V, et al. Ambient backscatter: wireless communication out of thin air[C]//ACM SIGCOMM Computer Communication Review. ACM, 2013, 43(4): 39-50.

[3] QIAN J, GAO F, WANG G, et al. Noncoherent detections for ambient backscatter system[J]. IEEE Transactions on Wireless Communications, 2017, 16(3): 1412-1422.

[4] NORDRUM A. The Internet of fewer things[J]. IEEE Spectrum, 2016, 53(10): 12-13.

[5] QIAN J, PARKS A N, SMITH J R, et al. IoT communications with M-PSK modulated ambient backscatter: algorithm, analysis and implementation[J]. IEEE Internet of Things Journal, 2019, 6(1): 844-855.

[6] FORTZ B, THORUP M. Internet traffic engineering by optimizing OSPF weights[J]. IEEE INFOCOM, 2000, 2(3): 519-528.

[7] HEDRICK C L. Routing information protocol[R]. 1988.

[8] FORTZ B, THORUP M. Optimizing OSPF/IS-IS weights in a changing world[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(4): 756-767.

[9] GRIFFIN T G, SHEPHERD F B, WILFONG G. The stable paths problem and interdomain routing[J]. IEEE/ACM Transactions on Networking (ToN), 2002, 10(2): 232-243.

[10] 孙志军, 薛磊, 许阳明, 等. 深度学习研究综述[J]. 计算机应用研究, 2012, 29(8): 2806-2810.
SUN Z J, XUE L, XU Y M, et al. Overview of deep learning[J]. Application Research of Computers, 2012, 29(8): 2806-2810.

[11] KATO N, FADLULLAH Z M, MAO B, et al. The deep learning vision for heterogeneous network traffic control: proposal, challenges and future perspective[J]. IEEE Wireless Communications, 2017, 24(3): 146-153.

[12] TANG F, MAO B, FADLULLAH Z M, et al. On removing routing protocol from future wireless networks: a real-time deep learning approach for intelligent traffic control[J]. IEEE Wireless Communications, 2018, 25(1): 154-160.

[13] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, 30(1): 86-100.
GAO Y, CHEN S F, LU X. Research on reinforcement learning technology: a review[J]. ACTA Automatica Sinica, 2004, 30(1): 86-100.

[14] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529.

[15] BOYAN J A, LITTMAN M L. Packet routing in dynamically changing networks: a reinforcement learning approach[C]//Advances in Neural Information Processing Systems. Morgan Kaufmann Publishers Inc, 1994: 671-678.

[作者简介]

丁瑞金（1994− ），男，江苏南通人，清华大学博士生，主要研究方向为 AI 及其在通信中的应用。

高飞飞（1980− ），男，陕西西安人，博士，清华大学副教授、博士生导师，主要研究方向为多天线通信以及智能信号处理技术。

邢玲（1978− ），女，四川成都人，博士，河南科技大学教授、博士生导师，主要研究方向为智能信息传输、计算机网络与多媒体技术。