MANET 网络激励节点协作的信任评估路由协议

许智君[1,2,3]，胡琪[1,3]，张玉军[1]，叶新铭[2]

（1. 中国科学院 计算技术研究所，北京 100190；

2. 内蒙古大学 计算机学院，内蒙古 呼和浩特 010021；3. 中国科学院 研究生院，北京 100049）

摘 要：针对移动自组网（MANET）中自私节点可能不愿意协作并拒绝为其他节点转发数据这一问题，提出一种基于信任评估的路由协议（TDSR）以激励节点协作并参与数据转发。信任值被用于评估节点的转发行为，低转发率的节点将被信任评估机制排除出网络，并被屏蔽一定时间后才能复活。仿真结果显示，在存在不协作节点的网络中，TDSR 仅付出一定的延时开销就可具备明显高于 DSR 的数据分组转发率。

关键词：移动自组网；协作；网络性能；信任评估

中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2012)07-0027-09

Trust evaluation routing protocol to enforce cooperation in mobile ad hoc networks

XU Zhi-jun[1,2,3], HU Qi[1,3], ZHANG Yu-jun[1], YE Xin-ming[2]

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;

2. College of Computer Science, Inner Mongolia University, Hohhot 010021, China;

3. Graduate School of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Mobile ad hoc networks (MANET) were based on the assumption that all participants cooperate and forward packets for others. Unfortunately, selfish nodes might be not willing to cooperate and refuse to forward packets. A trust evaluation routing protocol based on DSR (TDSR) was proposed to enforce cooperation and encourage forwarding packets in MANET. The trust value was used to evaluate packet forwarding behaviors. The nodes with low packet forwarding rate were excluded by the node and path trust evaluation. These nodes must wait some time for recovery. Simulation results show that TDSR has obviously higher packet delivery ratio than DSR in face of non-cooperated nodes at the expense of the performance degradation on the delay.

Key words: MANET; cooperation; network performance; trust evaluation

1 引言

移动自组网（MANET, mobile ad hoc network）是一种无固定基础设施支持、高度自治、具有动态性和多跳性的特殊对等网络，这些特点使 MANET 得到日益广泛的应用。在 MANET 中，节点自组织转发数据分组，所有网络功能依靠节点间的相互协作来实现，节点的协作性直接影响 MANET 网络各方面的性能。

一般来说，自私和贪婪是节点缺乏协作性的主

收稿日期：2011-08-29；**修回日期**：2012-04-11

基金项目：国家自然科学基金资助项目（60803139，61100177）；国家科技支撑计划基金资助项目（2008BAH37B07）；内蒙古自然科学基金资助项目（2011MS0902）

Foundation Items: The National Natural Science Foundation of China (60803139, 61100177); The National Key Technologies R&D Program of China (2008BAH37B07); The Natural Science Foundation of Inner Mongolia (2011MS0902)

要原因[1~3]。本文主要针对自私节点来激励节点协作并参与数据转发。一方面，由于移动节点受到各种资源（电量、带宽、计算能力、存储空间等）的限制，为了保存自己的资源，自私节点会拒绝为其他节点提供分组转发服务；另一方面，移动自组网是一个协作网络，自私节点需要其他节点为其转发数据。因此，自私节点往往不会单纯的丢弃所有分组，而是有选择的丢弃部分数据分组以此来获取网络资源。本文方案在如下自私行为存在的情况下仍能保持较高的网络性能：1）参与路由过程，但不转发数据分组；2）不参与路由过程，除非自身为源或目的节点：3) 以一定比例选择性的丢弃数据分组。

本文提出了激励节点协作的信任评估路由协议，用信任度作为节点分组转发行为的评价指标，提出节点和路径信任评估机制，用于路由建立、路由维护以及路由决策，提高数据分组转发率。为了进一步提高网络性能，提出结合复活机制和惩罚机制的增强方案以优化路由开销和延时开销。

MANET 网络节点协作性是一个十分活跃的研究领域，已有一定的研究成果。现有解决方案主要分为 3 大类[4]：基于虚拟货币的方案、基于博弈论的方法和基于信誉值的方案。

基于虚拟货币的方案将虚拟货币作为转发数据的奖励分发给节点，所有节点都需要虚拟货币来发送自己的数据分组。典型的方案包括 Nuglets[5]、ad hoc-VCG[6]和 Sprite[7]。这类方案往往需要依赖防篡改硬件或者第 3 方结算中心，因此缺乏独立性和灵活性，不便于应用实施。

基于博弈论的方法将转发行为建模为一个博弈过程。在这场博弈中，所有节点决定自己的最优决策。典型的方案包括 GTFT[8]、Catch[9]和文献[10]方案。这类方案需要节点收集大量的网络信息会导致巨大的网络负载。

基于信誉值的方案是解决 MANET 网络节点协作性问题最为成熟的一类解决方案[3]。Watchdog[1]最早提出 Watchdog 和 Pathrater 2 种技术来处理节点的恶意转发行为，但该方案没有相应的惩罚机制。CONFIDANT[11]和 Watchdog 类似，并进一步将不协作节点隔离出了网络。CORE[12]引入 3 种类型的信任值：主观信誉值、间接信誉值和函数信誉值，并综合 3 种信誉值得到最终的信任值，CORE 通过这种协作监听的方式来激励节点协作，但该方案对于节点行为的改变不够敏感。Friends and foes[13]方案在信任管理中提出 3 种独立的关系：朋友、敌人和自私节点，通过维护这 3 种关系和相应的状态变量来激励节点协作，但方案会消耗大量的内存。基于信誉值的方案有一个共同的缺陷，这些方案往往带来很大的开销。

本文针对节点协作性问题，兼顾考虑网络性能，提出基于信誉值的解决方案。通过建立信任评估模型，对于节点和路径进行信任评估，以此激励节点参与数据转发。并结合复活机制和惩罚机制，在保证较高数据分组转发率的情况下，有效降低路由开销和延时，提高网络性能。

2 信任评估路由协议

本文在 DSR[14]的基础上提出了信任评估路由协议（TDSR, trusted Evaluation routing protocol based on DSR），旨在激励节点参与网络协作。针对不协作节点，本文引入节点和路径信任评估模型，包括节点单向及双向信任评估和路径信任评估，用信任度评价节点的包转发行为；在此基础上设计了信任评估路由协议，结合信任机制进行路由建立、路由维护和路由决策，提高数据分组转发率并通过节点信任度进行路由请求的有效减枝，控制无效信令的传输和处理。

2.1 信任评估模型

信任评估包括 2 个紧密联系的部分，即节点信任评估和路径信任评估。节点信任评估采用邻居节点的直接评估机制，路径信任评估建立在节点信任评估的基础上。

节点间的信任评估主要采取类似 Watchdog 的策略，节点在混杂模式下监听邻居节点的数据分组转发行为，据此评价其信任度。在监听机制下，节点缓存需要邻居节点转发的数据分组，在一定时间段内如果这些分组被转发则从缓存中删除，时间段结束时缓存中剩余的数据分组为没有被正常处理的数据分组，即认为被邻居节点丢弃了。统计一定时间段内邻居节点的数据分组的接收数（packet received）和数据分组丢弃数（packet dropped），二者之差为数据分组转发数（packet forwarded），统计时隙为T_{cycle}，根据统计结果得到的邻居信任表如图 1 所示，该表为节点 A 的邻居信任表，邻居节点之一为节点 B。其中，${}_{\Delta}N_{\text{r}}^{A->B}$、${}_{\Delta}N_{\text{f}}^{A->B}$ 分别是当前时隙内节点 B 的分组接收数和分组转发数，$TV_{\text{rt}}^{A->B}$、$TV_{\text{dc}}^{A->B}$ 分别是节点 A 对节点 B 的实时信

节点A的邻居信任表

Neighbor_ID	Packet_Received	Packet_Forwarded	TV_realtime	TV_decision	Recovery_Number	Wait_Time
Node_ID B	${}_{\Delta}N_{\mathrm{r}}^{A->B}$	${}_{\Delta}N_{\mathrm{f}}^{A->B}$	$TV_{\mathrm{rt}}^{A->B}$	$TV_{\mathrm{dc}}^{A->B}$	n	T_{wait}

图 1　邻居信任表

任度和决策信任度，n 是 B 被判定为不协作节点的次数，T_{wait} 是等待复活的时间（参见 2.3 节）。则 $TV_{\mathrm{rt}}^{A->B}$ 的计算方法如式（1）所示：

$$TV_{\mathrm{rt}}^{A->B}=\begin{cases}0, {}_{\Delta}N_{\mathrm{r}}^{A->B}=0\\ \dfrac{{}_{\Delta}N_{\mathrm{f}}^{A->B}}{{}_{\Delta}N_{\mathrm{r}}^{A->B}}, \text{其他}\end{cases} \tag{1}$$

$TV_{\mathrm{dc}}^{A->B}$ 在每个时隙结束时进行更新，上一个时隙结束时 A 对 B 的决策信任度记为 $TV_{\mathrm{old\_dc}}^{A->B}$，新时隙内的收分组数为 ${}_{\Delta}N_{\mathrm{new\_r}}^{A->B}$，新时隙结束时的实时信任度记为 $TV_{\mathrm{new\_rt}}^{A->B}$，则 $TV_{\mathrm{new\_dc}}^{A->B}$ 的计算方法如式（2）所示：

$$TV_{\mathrm{new\_dc}}^{A->B}=\begin{cases}TV_{\mathrm{old\_dc}}^{A->B}, \ {}_{\Delta}N_{\mathrm{new\_r}}^{A->B}=0\\ \alpha TV_{\mathrm{old\_dc}}^{A->B}+\beta TV_{\mathrm{new\_rt}}^{A->B}, \text{其他}\end{cases} \tag{2}$$

其中，α,β 分别为信任度历史效应和实时效应的权重参数，且 $\alpha,\beta\in[0,1]$，$\alpha<\beta$，$\alpha+\beta=1$，实时信任度更能够反映节点的当前行为，因此应该赋予更大的权值。初始状态所有节点的决策信任度为 1。

节点信任评估只使用直接评估策略，不考虑其他节点的推荐，仅基于以下考虑。

1) 在混杂监听模式下，只要节点 B 位于节点 A 的有效通信范围内，A 就可以监听到来自 B 的所有路由分组，即使这些分组与 A 没有直接关系[1]，因此 ${}_{\Delta}N_{\mathrm{r}\_i}$、${}_{\Delta}N_{\mathrm{f}\_i}$ 和 ${}_{\Delta}N_{\mathrm{d}\_i}$，实际上已经包含了来自其他节点的部分信息。

2) 节点的信任推荐策略需要引入额外的信令交互和网络延时，特别是对于远距离节点（remote node）的信任度的维护，需要经过多跳节点的传输和交互，在节点动态性比较高，行为反复的情况下将给系统带来比较大的负担。

同时应该注意到，这里的节点信任度是单向的，因为 A 监听到的 B 的行为与 B 监听到的 A 的行为是不同的，2 个节点间的信任度应该综合考虑双向信任度。记节点 A 与节点 B 之间的双向决策信任度为 $TV_{\mathrm{dc}}^{A\leftrightarrow B}$，则 $TV_{\mathrm{dc}}^{A\leftrightarrow B}$ 为 2 节点间单向信任度的平均值：

$$TV_{\mathrm{dc}}^{A\leftrightarrow B}=(TV_{\mathrm{dc}}^{A->B}+TV_{\mathrm{dc}}^{B->A})/2 \tag{3}$$

路径信任度根据节点间双向决策信任度进行评估，记从源节点 S 到目标节点 D 的第 i 条路径为 $Path_i$，该路径的信任度为 pt_i，该路径中的节点分别为 $Node_0, Node_1, \cdots, Node_n$，则 $Path_i$ 的信任度为路径中节点间双向决策信任度的最小值，即

$$pt_i=\min(TV_{\mathrm{dc}}^{N_0\leftrightarrow N_1}, TV_{\mathrm{dc}}^{N_1\leftrightarrow N_2}, \cdots, TV_{\mathrm{dc}}^{N_{n-1}\leftrightarrow N_n}) \tag{4}$$

2.2　路由协议设计

路由协议 TDSR 包括路由建立（route discovery）和路由维护（route maintenance）2 个过程。2 个过程中综合考虑节点和路径信任度，并设计了最佳路由决策算法。

某时刻节点间的路由场景如图 2 所示，节点间双向决策信任度标注在一跳路径上，在 DSR 路由协议中，源节点根据收到路由响应的时间决定使用哪条路径，若 $S\to A\to D$ 的路由响应首先返回则选择该路径。但实际上由于节点 A 的分组丢失率很高，导致这条路径的数据分组转发率很低，而数据分组转发率较高的路径 $S\to C\to F\to D$ 被丢弃。因此路由建立和路由决策应该综合考虑路径信任度进行。

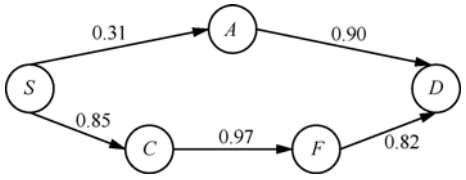

图 2　路由场景实例

2.2.1　路由建立过程

TDSR 的路由建立过程如下（* 表示广播，● 表示单播）。

Step 1

$Source\to *$:

$<REQUEST, S, D, id, NTT_S, (pt_0), pt_{\min}>$

φ（0<φ<1）为衡量节点/路径可信和不可信的阈值，如取 φ=0.3，则信任度大于 0.3 的节点/路径可信，否则不可信，φ 的取值可根据具体应用对安全性和可靠性的需求进行调节。源节点广播路由请求 RREQ，包括源节点 ID、目标节点 ID、RREQ 标识 id、源节点的邻居信任表 NTT_S，以及初始路径信任度 $pt_0=1$。

Step 2

$Intemediate\ node_i \rightarrow *$:

$< REQUEST, S, D, id, \{N_1, \cdots, N_i\}, (pt_0, \cdots, pt_i), NTT_i, pt_{\text{cur\_min}} >$

中间节点 i 收到广播的 RREQ，从本节点 NTT_i 中提取对前一跳节点的决策信任度 $T_{\text{dc}}^{i->(i-1)}$，从 RREQ 中 NTT_{i-1} 中提取前一跳节点对本节点的决策信任度 $T_{\text{dc}}^{(i-1)->i}$，据此计算双向决策信任度 $T_{\text{dc}}^{(i-1)<->i}$；剪枝策略为：如果 $T_{\text{dc}}^{(i-1)<->i}$ 大于 φ，且本节点不在已有的中间路径节点列表中，则将本节点 ID 增加到 RREQ 的中间节点列表中，否则直接丢弃该 RREQ；对于未被丢弃的 RREQ，若 $T_{\text{dc}}^{(i-1)<->i}$ 小于路由请求中当前路径信任度 $pt_{\text{cur\_min}}$，则用 $T_{\text{dc}}^{(i-1)<->i}$ 取代 $pt_{\text{cur\_min}}$，否则 $pt_{\text{cur\_min}}$ 不变；取得的 $pt_{\text{cur\_min}}$ 作为本节点计算出的路径信任度 pt_i 增加到路径信任度列表中；最后用 NTT_i 替换 RREQ 中的 NTT_{i-1}，然后 rebroadcast 路由请求。类似 DSR，中间节点只处理第一次收到的来自相同源节点的 RREQ，从而防止 RREQ 洪泛耗费网络资源。

一种典型情况是中间节点缓存了一条当前目的节点的可达路径，则取 $pt_{\text{cur\_min}}$ 和缓存路径的 pt_{min} 的最小值作为该路径的信任度，直接返回路由响应 RREP。减枝策略除了可以避免路由请求的洪泛，还对不协作节点起到惩罚作用。具体来说：节点只处理来自信任度高于一定值（φ）的节点发来的 RREQ，对于出于自私目的不参与协作的节点，监听的结果可以发现这类节点，并通过减枝不处理其 RREQ，当不协作节点有数据发送需求的时候，其他节点也不会为其提供服务。这样就可以激励有数据发送需求的节点参与协作，提供包转发服务。

Step 3

$Destination\ node \rightarrow \bullet$:

$< REPLY, D, S, id, \{N_1, \cdots, N_{(m-1)}, N_m\}, pt_{\text{min}} >$

假设路径长度为 m，当 RREQ 到达指定的目标节点时，目标节点根据 RREQ 中指定的路径回复路由响应给源节点，内容包括目标节点 ID，源节点 ID，RREQ 标识 id，路径节点列表，以及本路径的信任度；将路由请求 id 增加到路由响应中主要目的是为了防止节点通过发回过期的 RREP 发起 relay attack[9]，为此同时需要增加 id 有效期定时器，并对有效的 id 进行缓存。

Step 4

$Intermediate\ node_i \rightarrow Intermediate\ node_{i-1}$:

$< REPLY, D, S, id, \{N_1, \cdots, N_{(i-1)}, N_i, \cdots, N_m\}, pt_{\text{min}} >$

当路由响应到达中间节点时，每个中间节点确定 id 是否仍然有效，若有效则继续向前一跳节点转发 RREP；否则认为请求已经过期，直接丢弃该响应。

Step 5

$Intermediate\ node_1 \rightarrow S$:

$< REPLY, D, S, id, \{N_1, N_2, \cdots, N_m\}, pt_{\text{min}} >$

源节点收到路由响应以后，根据路径信任度进行路由决策，决策算法如图 3 所示。

```
运行于每个节点的分组处理函数（部分），输入参数：节点收到的分组
Recv (pkg packet){
if（收到的分组是 RREP && 本节点是路径源节点）{
    获取路由分组中的路径；
获取路径信任度 pt_min ；
    if ( pt_min > φ ){
        缓存这条路径；
        在缓存路径中寻找信任值大于 φ − Δ 的路径集合；
        // Δ 是远小于 φ 的数
        在该集合中选取路径最短的路径；
        返回被选路径；
        //返回用于后续数据分组转发的路径
    }//end if ( pt_min > φ )
    else
        无效路径，直接丢弃；        //信任度小于阈值的路径不可信
}//end if（收到的分组是 RREP && 本节点是路径源节点）
}
```

图 3 路由决策算法

2.2.2 路由维护过程

TDSR 的路由维护，除了包括类似 DSR 中由于拓扑变化或信道冲突导致的节点不可达以外，还包括由于节点信任度变化到阈值 $\phi(\phi \in (0,1))$ 以下导致的节点不可信。节点信任度变化引起的路由维护过程如下。

节点信任评估每隔时间 T_{cycle} 进行一次，当路径中的某中间节点计算得到的与下一跳邻居节点的双向决策信任度降低到阈值 ϕ 以下，则认为该节点不可被信任，于是返回路由错误消息 RERR 至源节点；收到 RERR 消息，源节点将缓存中所有包含该不可信节点所在路径全部删除。为防止节点行为动态性较强导致的频繁路由维护，ϕ 的设置应该比路由建立过程中的阈值 φ 小。

2.3　结合惩罚机制的节点复活机制

节点信任度一旦下降到阈值 φ 以下，就被作为不协作节点而屏蔽，此后不能成为被选路径上的节点，也无法发起新的路由请求。这种情况下，网络中实际有效的节点数会随着系统运行而逐渐减少，导致网络连通性变差，系统性能下降，特别是信令开销和延时增大。因此设计不协作节点复活机制，在不协作节点被屏蔽一定时间以后使其复活，同时引入相应的惩罚机制，激励节点合作进行数据分组转发。具体方案如下。

设置节点失效单位时间 T_{recover}，即一旦节点失效，最少需要等待 T_{recover} 才能恢复。邻居信任表记录成为不协作节点的次数 n，以及等待复活的时间 T_{wait}。一旦节点的信任度降到阈值 φ 以下，启动该邻居节点对应的定时器，初值设为 $T_{\text{wait}}=n\times T_{\text{recover}}$，即不协作行为更频繁的节点需要等待更长的时间才能复活。复活后节点的决策信任度初始值的设定与 n 相关。设系统信任度阈值为 $TV_{\text{threshold}}$，邻居信任表中节点 i 对应的不协作节点次数为 n_i，复活后的信任度初始值设为 $TV_{\text{cur}}+0.1+\Delta^n(\Delta\in(0,0.9-TV_{\text{cur}}))$。复活机制中的对 T_{wait} 和信任度初值的设置体现了对节点不协作行为的惩罚。

2.4　信任度评估的一致性说明

TDSR 的节点信任度是局部信任值，节点信任度不是全局唯一的，同一节点被不同的邻居节点评估值会不同。如图 2 所示，相对于节点 C，节点 S 和 F 互为隐藏节点，对于 C 同样的转发行为，由于隐藏节点造成的冲突，可能计算得到 $TV_{\text{dc}}^{S\to C}$ 和 $TV_{\text{dc}}^{F\to C}$ 不同。但是这种差异符合实际的链路差异，实际上反映了 S 经 C 转发数据和 F 经 C 转发数据因为不同的隐藏节点冲突造成的转发成功率不同。

当网络内节点传输碰撞较大时，即使节点转发了数据，由于冲突其邻居节点不能监听到转发数据，从而导致其节点评估值降低。这造成节点的真实协作意愿与信任度的不一致，但是评估得到的路径信任度与真实的链路情况是一致的。即信任度较低的链路其分组转发率也较低。同时，对于传输碰撞较大的链路，降低其信任度，减少数据转发也有利于降低冲突，恢复链路质量。

3　性能分析

使用 NS-2 进行系统仿真，模拟场景参数设置如表 1 所示，未列出的参数采用 NS-2 v2.33 中对 DSR 的默认设置，该版本 DSR 实现了文献[15]中的多数优化方案，如链路状态缓存（link state cache）[16]，流状态控制（flow state control）[17]等。TDSR 在该优化 DSR 版本上进行修改并与之比较，关闭了部分优化选项（如表 1 所示），TDSR 和 DSR 在这些优化选项的选择上保持一致。同时通过增大分组大小（packet size）反映增加的安全参数域及节点处理延时[18]，增加 RREQ 超时（timeout）来弥补安全处理延时，避免不必要的路由请求重传。

表 1　　仿真参数说明

仿真参数	值
仿真时间	900s
CBR 初始化时间	900s 中的前 180s
节点数	*N*
源/目的节点对数	30%×*N*
最大移动速率	20m/s
源数据模式(每节点)	1 packet/s
路由缓存	64
非正常节点数	*M*
抢救(salvage)控制	关
流状态控制	关
Snoop	关
IFQ 长度	1 000
应用数据负载	512byte
DSR RREQ 最大超时时间	10s
TDSR RREQ 最大超时时间	20s

场景运行 900s，其中前 180s 产生数据分组。基本设置是 50 个节点，拓扑大小 1 500m× 300m[11,18,19]，网络规模变化，拓扑大小也同比例变化，保证拓扑中节点的稠密程度相当。源/目的节点对数为网络规模即节点总数的 30%，如 50 个节点的场景中，源/目的节点对为 15，拓扑大小 1 500m× 300m；200 个节点的场景中，源/目的节点对为 60，拓扑大小 3 000m× 600m。节点的相对传输距离为 250m。

每种场景下协议运行关注以下 5 个参数[19]。

①数据分组转发率（PDR, packet delivery ratio），即指定目的节点应用层数据分组的接收百分比，当网络中存在多对源和目的节点的情况下，平均数据

分组转发率定义为

$$\overline{PDR}=\frac{\sum_{DN=1}^{0.3N}\text{目的节点接收的数据分组}}{\sum_{SN=1}^{0.3N}\text{源节点发送的数据分组}}\times 100\%$$

（*SN*：源节点，*DN*：目的节点）

②每个成功接收的数据分组的路由层平均信令开销（RMO, route-level message overhead per correct delivery）：

$$\overline{RMO}=\frac{\sum\text{路由层信令开销}}{\sum\text{成功接收的数据分组}}\times 100\%$$

③所有成功建立路径的平均路径长度（ARL, average route length per correct delivery）：

$$\overline{ARL}=\frac{\sum\text{成功建立路径的路径长度}}{\sum\text{建立路径数}}\times 100\%$$

④数据分组在路由层的端到端平均传输延时（ETD, end-to-end transmission delay）：

$$\overline{ETD}=\frac{\sum\text{路由层接收数据分组的端到端延时}}{\sum\text{成功接收的数据分组数}}\times 100\%$$

⑤应用层所见的数据分组端到端平均发送延时（EED, end-to-end delay on application layer）：

$$\overline{EED}=\frac{\sum\text{应用层接收数据分组的端到端延时}}{\sum\text{成功接收的数据分组数}}\times 100\%$$

其中，应用层端到端平均发送延时不仅包括路由层的端到端平均传输延时，还包括数据分组在源节点处缓存等待发送的时间。

对于很多 MANET 典型应用，如移动数据合作转发、区域办公场合、会议展览场合、车载网络和战场网络等，数据分组转发率是一个非常重要的参数，而 TDSR 的最大优势在于，在存在不协作节点的环境中仍可以保证较高的数据分组转发率。下面对不同场景下协议运行的仿真结果进行分析。

3.1 无不协作节点的协议运行状况

考察无不协作节点的静态场景中，DSR/TDSR 随网络规模变化的运行状况，设置停留时间 *pause time* = 900s，协议运行状况如图 4 所示。在无不协作节点的静态场景中，网络规模在 100 个节点以内时，DSR 与 TDSR 都能保持较高的数据分组转发率（>96%）、较低的信令开销（<10%）和端到端延时（<200ms）。但随着网络规模的增大，DSR 在 100 个节点处性能发生转折，数据分组转发率下降，信令开销和端到端延时增长迅速，这是由于冲突造成了大量的路径重建；而 TDSR 仍可以保持较好的路由性能，这是由于 TDSR 增加了 RREQ Timeout 来弥补安全处理延时，避免了不必要的 RREQ 重传，减少了网络冲突。同时，随着网络规模的增大，DSR 和 TDSR 的平均路径长度都逐渐增加，二者的平均传输延时在网络规模小于 100 个节点时也缓慢增长，即二者具有类似的变化趋势。此外，该场景下两种协议的应用层平均发送延时与路由层平均传输延时基本一致，即数据分组在源节点处等待被发送的时间很小，路径建立速度快。

同时，从图中可以看出，网络规模较小时 DSR 与 TDSR 表现相当，为了考察 2 种协议由于不协作节点而不是网络冲突造成的路由性能差异，需要选择一种基准场景，即 2 种协议运行状况都比较好、网络冲突造成的影响比较小的场景。因此下面的仿真将网络规模设定为 50 个节点。

然后考察无不协作节点的动态场景中，DSR/TDSR 随节点动态情况不同而变化的运行状况，设定网络规模为 50，协议运行状况如图 5 所示。随着节点动态性的减弱，协议性能有不同程度的提升，数据分组转发率逐渐升高，信令开销逐渐下降。具体来说，TDSR 与 DSR 相比数据分组转发率相当，平均信令开销在 *Pause Time* = 0~700s 范围内更低，应用层平均发送延时与路由层平均传输延时在 *Pause Time* = 100~900s 范围内相当，在 *Pause Time* = 0 时较大，这时由于在节点动态性较强、断路情况较频繁时，TDSR 需要等待更长的时间才能进行路径重建；延时和信令开销的起伏是由于网络冲突和断路的综合影响，而延时的变化趋势与平均路径长度的变化趋势是一致的。同样的，在无不协作节点的动态场景中，2 种协议的应用层平均发送延时与路由层平均传输延时基本一致。

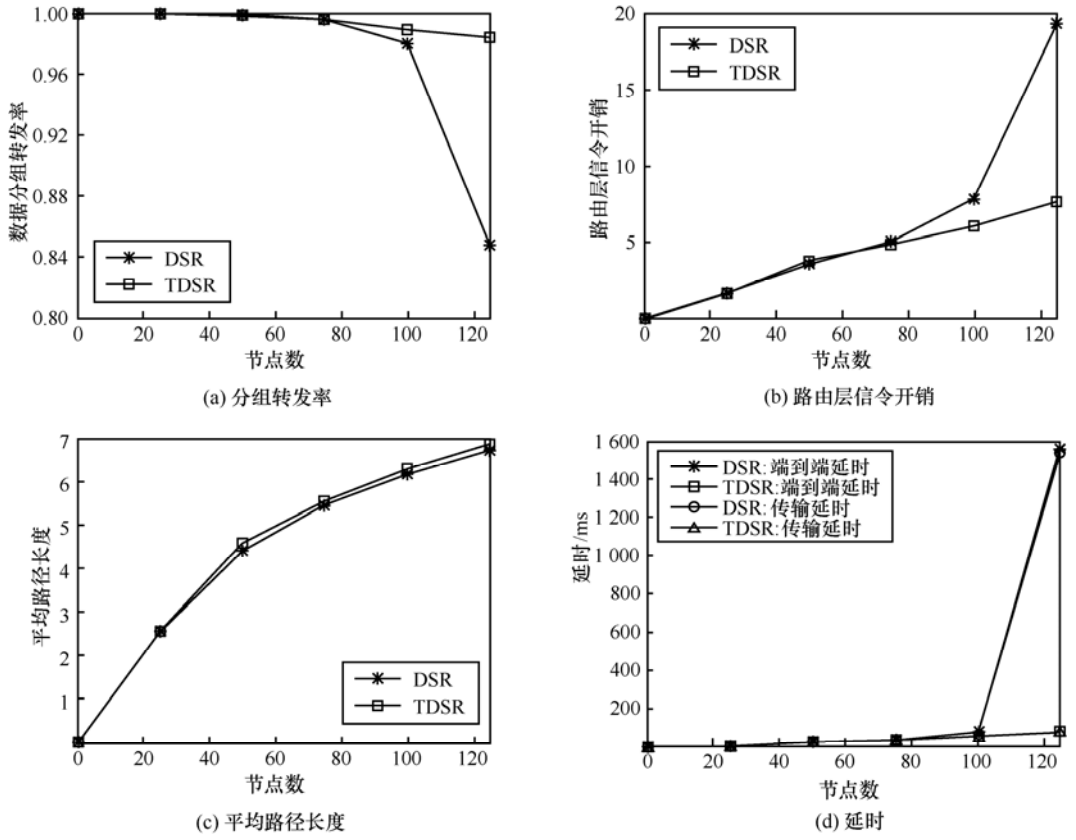

图 4　无不协作节点静态场景

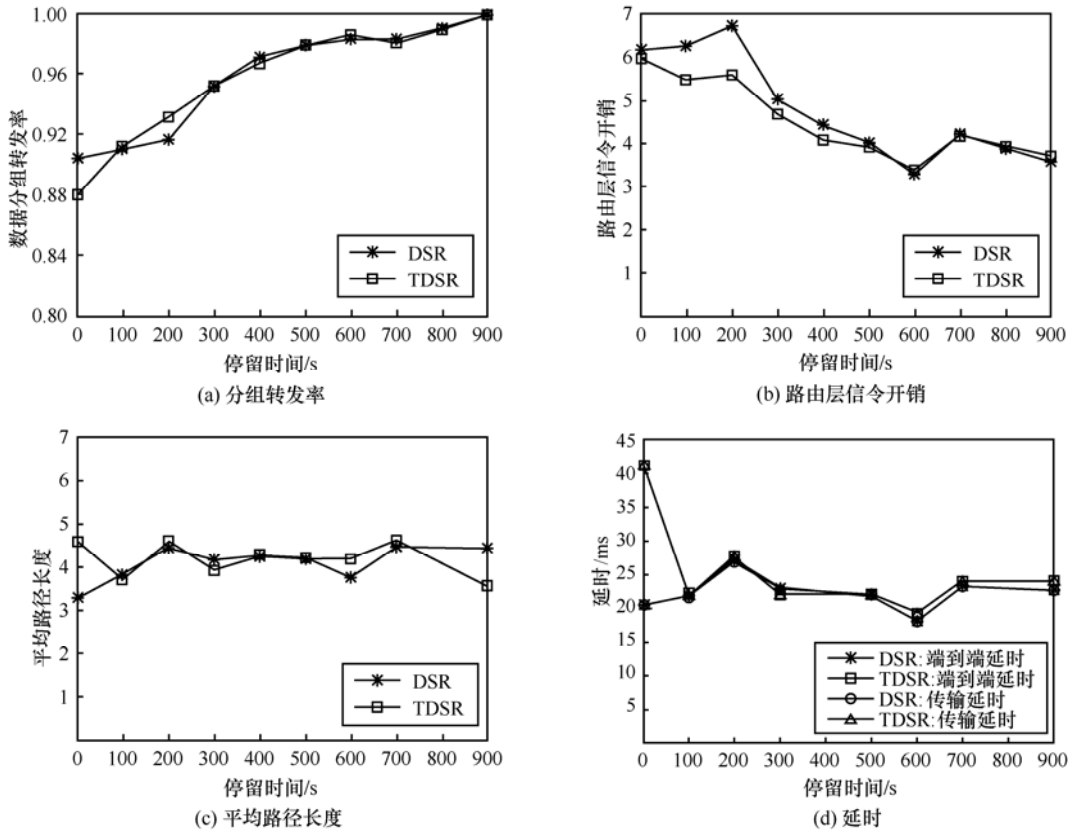

图 5　无不协作节点动态场景（$N = 50$）

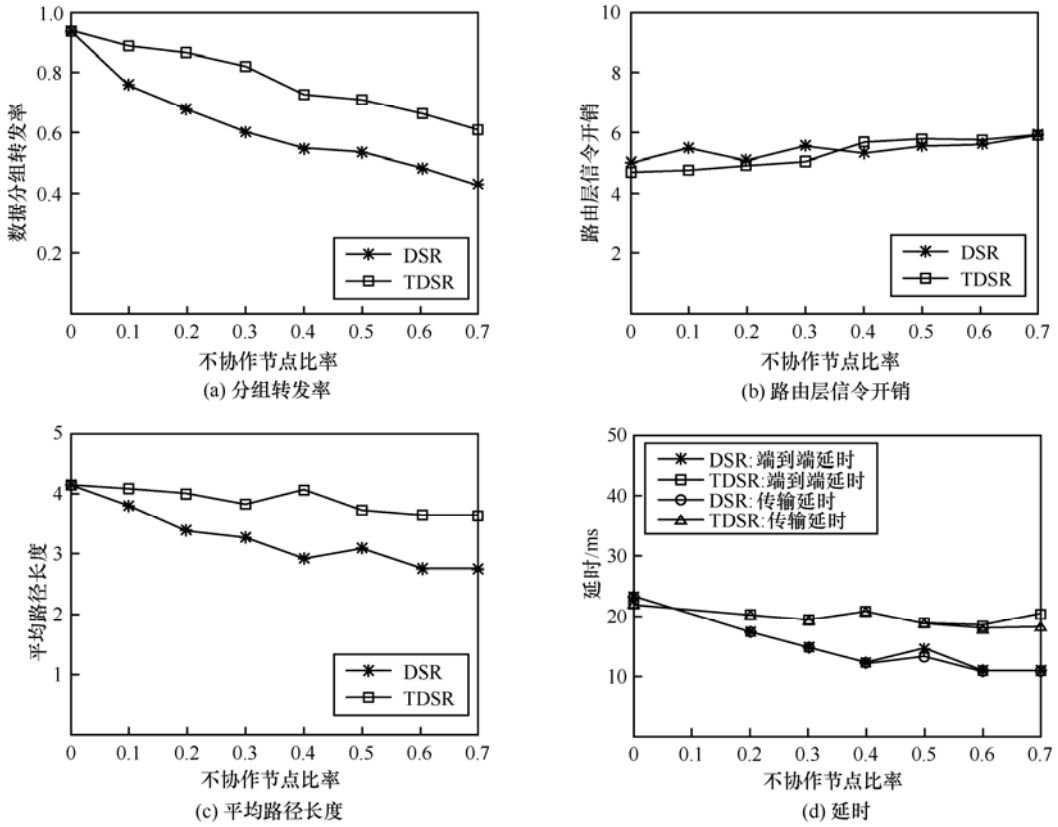

图 6　存在不协作节点动态场景（$N = 50$，*Pause Time* = 300s）

后面将考察 *Pause Time* = 300s，即节点具有一定动态性、性能比较理想的情况下，2 种协议随不协作节点数目变化的性能。

3.2　存在不协作节点的协议运行状况

本仿真使用选择错误模型（SEM, select error model）实现节点的不协作分组丢失，着重分析 TDSR 抵抗不协作节点的性能。比较 DSR 和 TDSR 在网络规模固定（$N = 50$），动态状况相同（*Pause Time* = 300s）的情况下，数据分组转发率随不协作节点数增加而变化的情况。路由决策信任度阈值 φ=0.4，路由维护信任度阈值 ϕ=0.3，$\Delta = 0.3$。不协作节点的选取随机确定（网络规模的 0~70%范围内变化），不协作节点的分组丢失率在一定范围内随机变化。仿真结果如图 6 所示。

从图中可以看出，TDSR 的数据分组转发率比 DSR 更高，这是由于 TDSR 通过避免将不协作节点选入路径，保持了较高的数据分组转发率；当不协作节点数小于 35%时，由于 TDSR 引入了减枝策略，平均信令开销比 DSR 少；但由于路由维护中的信任度变化引入了额外的信令开销，可能引发新的路由请求，因此不协作节点数大于 35%以后，TDSR 的平均信令开销基本与 DSR 相当；同时由于需要绕过不协作节点，TDSR 的平均路径长度比 DSR 大，二者的路由层平均传输延时的变化趋势与平均路径长度相当。而 TDSR 由于需要避开不协作节点，建路延时随着不协作节点数增加而增大，体现在 TDSR 的应用层平均发送延时与路由层平均传输延时随着不协作节点数目的增多差距变大。即 TDSR 随着不协作节点数增加平均路径长度和延时的降低没有 DSR 明显，这是由于 TDSR 的节点复活机制使得网络的活跃节点数没有随着不协作节点数目的增多而显著降低，保持了一定的网络连通性和网络直径。总体来看，TDSR 平均延时大于 DSR，但仍保持低于 25ms，对于时延要求不是很严格的 MANET 应用是可以接受的。

4　结束语

不协作节点的存在对 MANET 路由的安全性和可靠性造成威胁，特别是会导致数据分组转发率的下降。针对这种情况，本文提出 MANET 网络激励节点协作的信任评估路由协议，通过节点和路径信任评估机制衡量节点和路径的数据分组转发行为，

并引入路由请求减枝策略，以此为基础进行路由建立、路由维护和路由决策，同时设计结合惩罚机制的节点复活机制，避免网络连通性随不协作节点数增大而显著变差。仿真结果表明，在无不协作节点的静态场景中，DSR 在 100 个节点规模时发生性能转折，而 TDSR 此时仍可以保持较好的路由性能；固定网络规模为 50 个节点，在无不协作节点的动态场景中，TDSR 具有与 DSR 类似的性能，特别是信令开销在 *Pause Time*<700s 时较小；固定 *Pause Time*=300s，在存在不协作节点的动态场景中，TDSR 比 DSR 的数据分组转发率显著提高，信令开销与 DSR 相当，但延时开销高于 DSR。

参考文献：

[1] MARTI S, GIULI T, LAI K, *et al*. Mitigating routing misbehavior in mobile ad hoc networks[A]. Proc MobiCom'00[C]. NY, USA, 2000. 255-265.

[2] DJAHEL S, NAIT-ABDESSELAM F, ZHANG Z. Mitigating packet dropping problem in mobile ad hoc networks: proposals and challenges[J]. Communications Surveys & Tutorials, 2010, 99:1-15.

[3] YU H, SHEN Z, MIAO C, *et al*. A survey of trust and reputation management systems in wireless communications[J]. Proceedings of the IEEE, 2010, 98(10): 1755-1772.

[4] YOO Y, AGRAWAL D P. Why does it pay to be selfish in a MANET[J]. IEEE Wireless Communications, 2006, 13(6): 87-97.

[5] BUTTYÁN L, HUBAUX J P. Enforcing service availability in mobile ad-hoc networks[A]. Proc MobiHoc'00[C]. Boston, USA, 2000. 87-96.

[6] ANDEREGG L, EIDENBENZ S. Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents[A]. Proc MobiCom'03[C]. NY, USA, 2003. 245-259.

[7] ZHONG S, CHEN J, YANG Y R. Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks[A]. Proc IEEE INFOCOM 2003[C]. USA, 2003. 1987-1997.

[8] SRINIVASAN V, NUGGEHALLI P, CHIASSERIN C F, *et al*. Cooperation in wireless ad hoc networks[A]. Proc IEEE INFOCOM 2003[C]. USA, 2003. 808-817.

[9] MAHAJAN R, RODRIG M, WETHERALL D, *et al*. Experiences applying game theory to system design[A]. Proc ACM PINS'04[C]. NY, USA, 2004. 183-190.

[10] WEI H Y, GITLIN R D. Incentive scheduling for cooperative relay in WWAN/WLAN two-hop-relay network[A]. Proc IEEE Wireless Communication and Network Conference 2005[C]. LA, USA, 2005. 1696-1701.

[11] BUCHEGGER S, BOUDEC J Y L. Performance analysis of the CONFIDANT protocol: cooperation of nodes fairness in dynamic ad hoc networks[A]. Proc MobiHoc'02[C]. NY, USA, 2002. 226-236.

[12] MICHIARDI P, MOLVA R. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks[A]. Proc of Communication and Multimedia Security 2002[C]. Portoroz, Slovenia, 2002. 107-121.

[13] MIRANDA H, RODRIGUES L. Friends and foes: preventing selfishness in open mobile ad hoc networks[A]. Proc 23rd International Conference on Distributed Computing Systems Workshops 2003[C]. USA, 2003. 440-445.

[14] JOHNSON D B, MALTZ D A. Dynamic source routing in ad hoc wireless networks[A]. Mobile Computing[C]. Kluwer, 1996. 153-181.

[15] JOHNSON D B, MALTZ D A, HU Y C. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4[S]. IETF RFC 4728, 2007.

[16] HU Y C, JOHNSON D B. Caching strategies in on-demand routing protocols for wireless ad hoc networks[A]. Proc MobiCom'00[C]. NY, USA, 2000. 231-242.

[17] HU Y C, JOHNSON D B. Implicit source routing for on-demand ad hoc network routing[A]. Proc MobilHoc'01[C]. NY, USA, 2001. 1-10.

[18] HU Y C, PERRIG A, JOHNSON D B. Ariadne: a secure on-demand routing protocol for ad hoc networks[J]. Wireless Network, 2005, 11(2): 21-38.

[19] MIGUEL M C, CASTRO M, NIGHTINGALE E B, *et al*. Virtual ring routing: network routing inspired by DHTs[A]. Proc SIGCOMM'06[C]. NY, USA, 2006. 351-362.

作者简介：

许智君（1974-），男，内蒙古兴和人，中国科学院博士生，主要研究方向为 MANET 网络和无线 mesh 网络。

胡琪（1984-），女，湖北武汉人，中国科学院硕士生，主要研究方向为 MANET 网络。

张玉军（1976-），男，河北衡水人，中国科学院副研究员、博士生导师，主要研究方向为下一代网络。

叶新铭（1943-），男，内蒙古海拉尔人，内蒙古大学教授、博士生导师，主要研究方向为计算机网络和分布式系统、协议测试和形式化描述等。