

基于动态分段网络编码的 DTN 高效路由算法

姚玉坤, 陈 曦, 余志龙, 徐亚伟

(重庆邮电大学 移动通信技术重庆市重点实验室, 重庆 400065)

摘 要: 针对 DTN 现有的动态分段网络编码 (Dynamic Segmented Network Coding, DSNC) 路由算法在编码包头部存在冗余字段、ACK 反馈阶段存在冗余开销和节点因无序转发编码包而影响数据包端到端时延等问题, 提出基于动态分段网络编码的 DTN 高效路由算法—ERBNC (Efficient Routing Based on dynamic segmented Network Coding). 该算法通过采取压缩编码包头部字段、删除冗余的 ACK 分组、设置编码包的发送优先级等措施以减少网络开销和数据包平均端到端时延。仿真结果表明, 与 DSNC 算法相比, ERBNC 算法在网络开销和端到端时延方面的性能均得到改善。

关键词: DTN; 路由算法; 网络编码; 动态分段; 冗余开销

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2015)03-0136-06

An Efficient Routing Algorithm for DTN Based on Dynamic Segmented Network Coding

YAO Yu-kun, CHEN Xi, YU Zhi-long, XU Ya-wei

(Chongqing Key Lab of Mobile Communications Technology,
Chongqing University of Post and Communications, Chongqing 400065, China)

Abstract: To solve the redundant overhead during the transmission of ACK packets and coding packet header, and the end to end affected by the node disorder forwards the coded packets, an Efficient Routing Algorithm for DTN based on Dynamic Segmented Network Coding (ERBNC) is proposed. By discarding the redundant ACK and coding packets in ACK transfer stage, and reducing the coding packet header fields, sets different priorities for coding packets to be forwarded, ERBNC had reduced the network overhead as well as the end-to-end delay of data packets. Simulation results show that, the performance of network overhead and the end-to-end delay of ERBNC algorithm are improved compared with DSNC.

Key words: DTN; routing algorithms; network coding; dynamic segmented; network overhead

1 引言

随着网络技术的发展, 出现了一类新型网络, 如军用 Ad hoc 网络、车载 Ad hoc 网络等。这类网络的特点是节点能量有限、稀疏且具有移动性, 网络链路会出现间歇性连接, 没有稳定的端到端路径。因此传统的路由算法 (如 AODV) 无法适用于该网络。在此背景下, 人们提出延迟容忍网络 (Delay and Disruption Tolerant Network, DTN)^[1] 的概念, DTN 是一

种在一些特定网络环境下会经常出现网络断开现象, 导致数据在传输过程中不能确保端到端路径的无线移动自组织网络。目前 DTN 主要应用于深空通信、野生动物监测与追踪、偏远地区的网络接入等领域。

对于没有稳定端到端路径的 DTN 来说, 设计有效的数据转发策略至关重要。由于网络编码^[2] 技术允许节点对数据进行编码后再转发, 可提高网络吞吐量。故将网络编码与路由算法相结合已成为

收稿日期: 2014-07-30; 修回日期: 2014-09-22

基金项目: 重庆市自然科学基金资助项目 (cstc2012jjA40040); 长江学者和创新团队发展计划资助 (IRT1299)

DTN 中设计数据转发策略的一个重要研究方向. 基于动态分段网络编码 (Dynamic Segmented Network Coding, DSNC) 路由算法^[3]能有效降低数据编码的复杂度, 但该算法中目的节点需要向各节点反馈 ACK 分组, 会带来一定的网络开销. 为解决 DSNC 算法网络开销较大等问题, 本文提出一种基于动态分段网络编码的 DTN 高效路由算法—ERBNC (Efficient Routing Based on dynamic segmented Network Coding), 并对性能进行了理论分析和仿真实验验证.

2 相关工作

大量的研究人员对如何将网络编码与 DTN 中的路由算法相结合这一问题, 进行了大量的研究. 文献^[4]中将随机线性网络编码 (Random Linear Network Coding, RLNC) 和传染路由结合在一起, 提出了基于网络编码的传染路由算法 (Network Coding Based Epidemic Routing, NCER), 并通过理论分析和仿真实验验证了 NCER 路由对于网络性能的提升. 但该算法的主要缺点是将大量的数据包编码在一起会导致编码向量长度增加, 相应的解码时延也会增加. 文献^[5]中针对多数据流并存情况下的流间编码问题给出明确的编码方案, 提出在 NCER 路由协议的基础上, 根据数据包目的节点的分类情况, 对多条数据流同时传输时的数据编码方式进行了改进, 有效地解决了多数据流并存时引起的编码向量长度增加的问题, 减少了目的节点等待解码的时间.

文献^[6]中通过理论分析和大量的仿真实验对 DTN 中基于网络编码的单播传染路由进行了性能评估. 实验结果表明在传染路由算法中采用随机网络编码, 能够明显获得性能上的提升. 文献^[7]中针对在传染路由算法中采用随机网络编码会存在冗余编码包而带来额外的网络开销这一问题, 提出了冗余度的概念, 利用冗余度来控制编码包的传输过程, 降低了冗余编码包所带来的额外的网络开销. 但该算法在利用冗余度控制编码包的传输过程中, 并没有考虑编码包间的线性相关问题, 仍会存在一定量冗余的编码包.

文献^[8]中证明机会链路^[9]持续时间的大小由网络中节点的运动特性和节点所采用的邻居发现机制所决定. 故在有限的传输带宽下, 当两个节点相遇时, 往往只有一部分需要传输的数据包能够被传送到相遇节点, 因此如何选择被传输的数据包来传送决定了网络中的数据传输性能.

3 网络模型及问题描述

3.1 网络模型

本文采用同 DSNC 相同的网络模型, 如图 1 所示. 该模型由一个源节点 S 、 N 个中间节点和一个目的节点 D 组成. 对本模型给出如下假设前提:

(1) 源节点采用 Epidemic 机制向目的节点传输数据.

(2) 源节点采用随机线性网络编码对源数据包进行编码处理;

(3) 节点间建立的机会链路的带宽只允许每次传输一个编码包;

(4) 源节点和目的节点具有足够的缓存, 中间节点缓存容量受限.

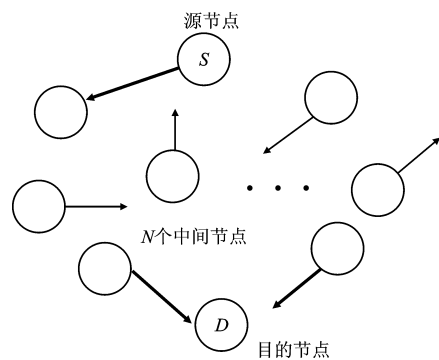


图1 ERBNC 算法的网络模型

3.2 DSNC 算法的问题分析

DSNC 算法根据目的节点反馈的 ACK 分组将数据进行动态分段编码传输. 具体过程为源节点将每段的编码包通过中间节点传输至目的节点. 目的节点接收到编码包后按段进行解码, 每当把属于同一个段的编码包全部解码成功后, 就以传染路由的方式将 ACK 分组反馈至源节点. 当源节点接收到目的节点反馈的属于前一个段的 ACK 分组后, 即可停止对当前段的编码工作并开始对下一个段的数据进行编码发送.

该算法存在以下 3 个问题:

(1) 目的节点通过编码包头部 (头部格式见图 2, 字段长度单位为比特) 中的 `Seg_size` 字段 (意指本段含有的编码包总个数) 判断某段的编码包是否全部解码完成. 这样的操作需要每个编码包头部中带有 8 bit 的 `Seg_size` 字段. 实际上利用头部中的其它字段也可判断某段的编码包是否全部解码完成. 因此包头中 `Seg_size` 字段是冗余的, 会带来一定的开销.

0	8	16	24	32	40	48可变长度	可变长度
Seg_ID	Cp_ID	TTL	Seg_size	Src_ID	Dest_ID	CV	PK_ID

图 2 DSNC 算法编码包头格式

(2)当两个节点相遇后,通过交换索引信息各自判断出对方缓存中没有而自身缓存中有的编码包,并将这些编码包随机选择发送给对方节点.但因节点的相遇时间较短且节点间建立的机会链路的带宽受限,不能保证所有的编码包都能发送到对方节点.在此过程中,如果对方节点不是目的节点,剩余生存时间 TTL 值小的编码包却被最先发给对方节点,那么该编码包可能会在后续的传输过程中不断地被传送到其他中间节点,直到生存时间结束(即 TTL 为 0)也没有被发送到目的节点,这种情况会影响数据包的端到端时延.

(3)当目的节点将某段的编码包全部解码成功后,要采用传染路由的方式将 ACK 分组反馈至源节点.在该过程中,会出现当源节点已经接收到某段编码包的 ACK 分组,但网络中的某些中间节点还会继续携带该 ACK 分组进行传染泛洪的情况,这时就存在 ACK 分组的冗余传输问题.

4 ERBNC 算法

ERBNC 算法提出编码包头字段压缩、编码包转发优先级和减少冗余 ACK 分组 3 个新策略来解决 DSNC 算法存在的上述 3 个问题.

4.1 编码包头字段压缩策略

压缩编码包头字段的方法是:将编码包头中 8 bit 的 Seg_size 字段改成只需 1 bit 的标识位 Flag.当 Flag 为 0 时,说明该段数据的编码工作还未结束.当 Flag 为 1 时,说明该段数据的编码工作已结束.

当目的节点收到编码包后,若查看到头中 Flag 为 1,此时执行如下操作获知该段编码包的总数:

设第 i 段的编码包的总数为 X ,该段 Flag 为 1 的编码包头中 Pk_ID(源数据包序号)的最大值为 m ,最小值为 n ,则

$$X = m - n + 1 \quad (1)$$

目的节点获知该段的编码包总数后,将已解码出的源数据包的个数和 X 进行对比,若两者相等,则说明目的节点对该段的编码包已全部解码成功.

4.2 编码包转发优先级策略

当两个节点 A 、 B 相遇后,通过交换索引信息各自进行判断,将自身缓存中有而对方节点缓存中没有的编码包按段号大小升序排列,然后根据编码包的地址和剩余生存时间 TTL 的不同对编码包的发送顺序做出如下调整:

(1)若对方是目的节点,则 TTL 值越小的编码包其转发优先级越高,将编码包按 TTL 值由小到大的顺序依次发送给对方节点.

这是因为编码包的 TTL 值越小说明该编码包剩余的生存期越短,而目的节点又没有该编码包,所以该编码包应优先发送给目的节点,以避免因该编码包这次没有被发送到目的节点,以致于在后续传输过程中直到生存期到期也没有被发送到目的节点,从而影响数据包的端到端时延.

(2)若对方不是目的节点,则 TTL 值越大的编码包其转发优先级越高,将编码包按 TTL 值由大到小的顺序依次发送给对方节点.

这是因为编码包的 TTL 值越大说明该编码包剩余的生存期越长,在网络中传播的越少,越需要在网络中尽快地传播.故应优先发送 TTL 值大的编码包,使其在网络中被广泛的传播,增大发送给目的节点的速度.而将 TTL 值小的编码包作为最后来发送,是为了尽可能地保留 TTL 值小的编码包,以便在其有限的生存期内可以被发送到目的节点,减小数据包的端到端时延.

4.3 减少冗余 ACK 分组策略

因为源节点采用动态分段编码的方式,只有源节点收到目的节点反馈的属于前一个段的 ACK 分组后,即可停止对当前段的编码工作并开始对下一个段的数据进行编码发送.由此可知 ACK 分组的段号和源节点进行下一个段编码的该段段号之间的差值为 2.进而可推出若中间节点缓存中最大的段号为 i ,那么段号比 i 小 2 的所有的段都已解码成功且都成功地通知到了源节点,中间节点无需再对该 ACK 分组进行传播.

基于上述原因 ERBNC 算法设计了一种 ACK 分组冗余减少策略:设节点 A 、 B 各自的两缓存区中最大的段号分别为 a 、 b ,ACK 分组所对应的段号为 c .在 ACK 分组传播的过程中,当节点 A 收到节点 B 发送的索引信息,判断出自身缓存中有而对方缓存中没有的 ACK 分组后,执行如下操作.

第一步:若 B 为源节点,则 A 无需经过任何判断,直接将自身缓存中有而 B 缓存中没有的 ACK

分组发送给 B , 并从缓存中删除该 ACK 分组. 否则执行第二步.

第二步: 判断 $\max(a, b) - c \geq 2$ 是否成立. 若成立, 说明自身缓存中有而 B 缓存中没有的 ACK 分组已经送至源节点, 故 A 不再给 B 发送该 ACK 分组, 并从缓存中删除该 ACK 分组; 否则执行第三步.

第三步: 按照正常的传染路由机制将 ACK 分组进行转发前传, 对方节点收到 ACK 分组后, 删除缓存中相应段的编码包.

第四步: 若节点 A 、 B 的缓存中都没有 ACK 分组, 那么节点 A 、 B 均采用相同的处理操作: 对于自身缓存中有而对方缓存中没有的编码包先进行筛选和删除处理, 即筛选出段号比对方缓存中段号小 2 的编码包进行删除操作, 不再发送给对方节点. 然后把筛选后剩余的编码包按照转发优先级发送给对方节点.

同理, 节点 B 收到节点 A 发送的索引信息后采用上述相同的操作步骤进行处理.

4.4 ERBNC 算法的运行步骤

ERBNC 算法主要包括源节点编码、中间节点转发编码包和目的节点解码三个阶段. ERBNC 路由算法各阶段的具体运行步骤如下所示.

(1) 源节点编码

源节点采用递增的编码方式对每个段中的数据块进行随机线性编码操作. 设当前源节点正对第 i 段的数据包进行编码, 按源节点是否收到 ACK_{i-1} 分组分两种情况讨论:

① 源节点收到第 $i-1$ 段的 ACK_{i-1} 分组, 则源节点将最后一个编码包包头中的标识位 Flag 置为 1 后停止对第 i 段数据包的编码, 开始对第 $i+1$ 段数据包进行编码;

② 若第 i 段的编码包个数已达到上限值 M (注: 第 1 段的编码包个数需预定义), 但源节点仍未收到第 $i-1$ 段的 ACK_{i-1} 分组, 则源节点将最后一个编码包包头中的标识位 Flag 置为 1 后停止第 i 段的编码, 进行等待, 直到收到第 $i-1$ 段的 ACK_{i-1} 分组后再开始对第 $i+1$ 段数据进行编码.

(2) 中间节点进行编码包转发与缓存管理

当两个节点相遇后, 交换彼此的索引信息, 将自己有而对方节点没有的编码包按段号大小升序排列, 再按照 4.2 节中所述的编码包转发策略将这些编码包按序发送给对方节点. 其中, 各个中间节点采用相同的缓存处理策略, 如图 3 所示.

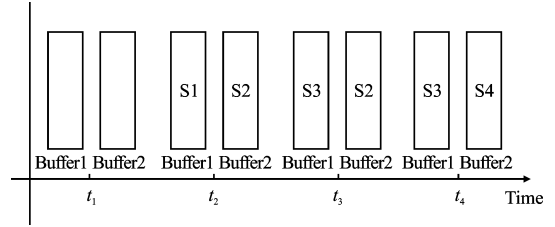


图3 ERBNC 算法的中间节点缓存处理策略

图 3 表示中间节点有两个缓存区 Buffer 1 和 Buffer 2. 每个缓存区一次只服务一个段的数据. t_1 时刻, 某个中间节点未遇到其他节点, 此时两个缓存区都是空的; t_2 时刻, 该节点与其他节点相遇, 收到了第 1、2 段的编码包, 此时 Buffer 1 存放第 1 段的编码包, Buffer 2 存放第 2 段的编码包; t_3 时刻, 该节点收到第 3 段的编码包, 此时 Buffer 1 删除第 1 段的编码包来存放第 3 段的编码包; t_4 时刻, 该节点收到第 4 段的编码包, 此时 Buffer 2 删除第 2 段的编码包来存放第 4 段的编码包. 以此类推, 直到完成所有数据的传输工作.

(3) 目的节点解码

目的节点收到编码包后, 根据段号的不同分别对编码包进行高斯消元解码. 例如, 目的节点接收到属于同一段的 3 个编码包分别为 $A_1 = a_1 P_1$ 、 $A_2 = b_1 P_1 + b_2 P_2$ 、 $A_3 = c_1 P_1 + c_2 P_2 + c_3 P_3$. 令 F 表示编码系数矩阵、 P 表示源数据包矩阵、 A 表示编码包矩阵, 即

$$F = \begin{bmatrix} a_1 & 0 & 0 \\ b_1 & b_2 & 0 \\ c_1 & c_2 & c_3 \end{bmatrix}, P = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}, A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}$$

那么编码包矩阵可表示为如下形式:

$$A = FP \quad (2)$$

采用高斯消元的方法, 通过下面公式 (3) 进行解码获得源数据包.

$$P = F^{-1}A \quad (3)$$

解码完成后, 目的节点根据 4.1 节所述的方法判断某段的编码包是否全部解码完成, 如果解码完成, 则将该段的 ACK 分组进行传染路由发送. 在 ACK 分组传输的过程中, 采用 4.3 节中所述的冗余减少策略, 可减少 ACK 分组传输过程中因冗余问题所带来的网络开销.

5 仿真实验及结果分析

5.1 仿真场景与参数设置

本文采用 OPNET 14.5 版本搭建平台进行仿

真实验. 在 $1\,500\text{ m} \times 300\text{ m}$ 的矩形区域范围内随机分布 50 个节点, 并从 50 个节点中随机选取两个节点作为源和目的节点. 各节点的 MAC 层传输速率根据 IEEE 802.11b 标准设置为 11 Mbit/s. 网络的业务模型是源节点通过多个中间节点将 1 000 个数据包发送至目的节点. 本文根据节点通信半径的不同, 设置了 5 个仿真场景, 仿真实验中设置的主要仿真参数如表 1 所示.

表 1 主要仿真参数

参数	取值
移动模型	Random Waypoint
通信半径/m	10, 25, 50, 75, 100
仿真时间/s	3 000
数据包大小/kB	1

5.2 仿真结果分析

5.2.1 网络开销

由于本算法中 Hello 分组、SV 分组和 Request 分组与原算法相比没有发生变化, 故本文仿真中的网络开销只统计在网络运行时间内, 所有节点发送的编码包和 ACK 分组所包含的总比特数, 其值为

$$C_{\text{total}} = N_D \times S_D + N_A \times S_A \quad (4)$$

式中, C_{total} 为总的网络开销, S_D 和 N_D 分别对应编码包的长度和总的发送个数; S_A 和 N_A 分别对应 ACK 分组的长度和总的发送个数. DSNC 算法和 ERBNC 算法的网络开销的仿真结果如图 4 所示.

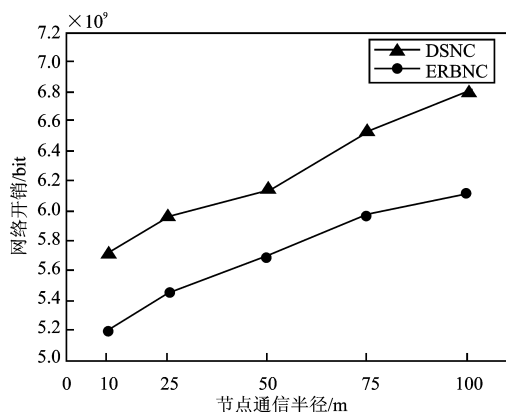


图 4 网络开销对比

由图 4 可看出, ERBNC 算法的网络开销在各场景中均低于 DSNC 算法. 这是因为 ERBNC 算法不仅对编码包包头格式的字段进行了压缩处理, 用一个 1 bit 的 Flag 代替 DSNC 算法中 8 bit 的 Seg_size, 减少了编码包在传输过程中的包头比特开销, 还在 ACK 传输阶段, 采取了 ACK 分组冗余减少策

略, 减少了冗余 ACK 分组的传输, 使得网络开销得到了减少.

5.2.2 数据包平均端到端时延

数据包的端到端时延包括数据包在节点内存储并携带的时间以及在节点之间传递的时间. 该值与节点的通信范围及数据包所经历的中间节点等因素有关. 其值为

$$T_{\text{avg}} = \frac{\sum_{i=1} T_i}{N_{\text{num}}} \quad (5)$$

式中, T_{avg} 为数据包平均端到端时延, T_i 为第 i 个到达目的节点的数据包的时延; N_{num} 为已到达目的节点的数据包个数. DSNC 算法和 ERBNC 算法的平均端到端时延的仿真结果如图 5 所示.

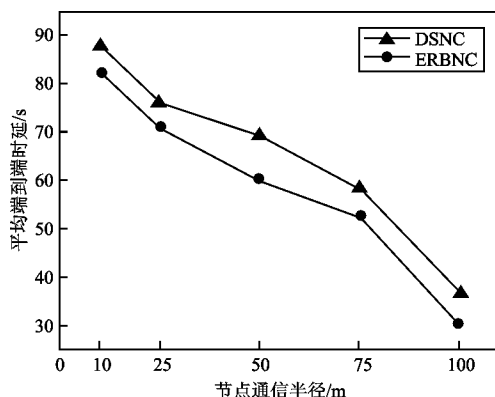


图 5 数据包平均端到端时延对比

由图 5 可看出, ERBNC 算法的数据包平均端到端时延在各场景中均低于 DSNC 算法. 这是因为 ERBNC 算法根据编码包的地址和 TTL 值的不同对编码包的发送顺序作出调整, 既避免了 TTL 值小的编码包本次没有被发送到目的节点却又在后续传输的过程中直到生存期到期也没有被发送到目的节点的情况, 又使得 TTL 值较大的编码包能较快地在网络中传播, 从而减少了数据包的端到端时延.

5.2.3 数据包传送成功率

因为 ERBNC 和 DSNC 两种算法都只考虑的是单源单目的的网络场景模型, 在本次仿真中, 分别统计源节点的发包个数和目的节点解码成功的数据包个数. 故本文仿真中数据包传送成功率是指目的节点解码获得的数据包的个数占网络中源节点产生并发送的数据包总个数的比例, 其值为

$$D_{\text{rate}} = \frac{D}{S} \quad (6)$$

式中, D_{rate} 为数据包传送成功率; D 为成功到达目的

节点的数据包的个数;S为源节点产生并发送的数据包的个数.其中,在5个场景的仿真中,两种算法中的目的节点通过解码后均获得1000个数据包.DSNC算法和ERBNC算法的数据包传送成功率仿真结果如表2所示.

表2 2种算法的数据包传送成功率对比

通信半径	成功率大小路由算法	DSNC/%	EERBNC/%
10		100	100
25		100	100
50		100	100
75		100	100
100		100	100

由表2可知,DSNC和ERBNC两种算法在仿真时间内,在不同的仿真场景中的数据包传送成功率都达到了100%.这是因为在这两个算法中都是基于泛洪机制,编码包在目的节点处均可全部解码成功,使得目的节点成功获得源节点发送的全部的数据包,具有较好的可靠性.

6 结束语

针对DSNC路由算法在编码包包头存在冗余字段和ACK反馈阶段存在冗余开销等问题,本文提出ERBNC路由算法,通过压缩编码包包头字段,减少冗余的ACK分组,有效降低了网络开销,同时根据编码包的目的地和生存时间对其发送顺序作出调整,减少了数据包平均端到端时延.仿真实验结果表明,与DSNC算法相比,ERBNC算法在网络开销和端到端时延方面的性能均得到了提高.未来我们的研究将在如何减少编码包在传输过程中的冗余问题方面展开.

参考文献:

- [1] K Fall. A delay-tolerant network architecture for challenged internets[C]//Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. Karlsruhe, Germany: ACM, 2003: 27-34.
- [2] R Ahlswede, N Cai, S-Y R Li, et al. Network information flow[J]. IEEE Transactions on Information Theory, 2000(46): 1204-1216.
- [3] D Zeng, S Guo, H Jin, et al. Dynamic segmented network coding for reliable data dissemination in delay tolerant networks[C]// Communications (ICC), 2012 IEEE International Conference on. Ottawa, Canada: IEEE, 2012: 63-67.

- [4] Y LIN, B LIANG, B LI. Performance modeling of network coding in epidemic routing[C]// Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp 2007). San Juan: ACM Press, 2007: 67-74.
- [5] 白云飞. 基于链路代价综合评估和网络编码的延迟容忍网络路由优化研究[D]. 北京:北京邮电大学,2012.
- [6] Seung-Keun Yoon, Zygmunt J Hass. Application of linear network coding in delay tolerant networks[C]// Second International Conference on Ubiquitous and Future Networks (ICUFN). Jeju Island, Korea: IEEE, 2010: 338-343.
- [7] B Zhao, W Peng, Z Song, et al. Towards efficient and practical network coding in delay tolerant networks[J]. Computers & Mathematics with Applications, 2012, 63(2): 588-600.
- [8] S Qin, G Feng, Y Zhang. How the contact-probing mechanism affects the transmission capacity of delay-tolerant networks[J]. Vehicular Technology, IEEE Transactions on, 2011, 60(4): 1825-1834.
- [9] Yunfeng Lin, Baochun Li and Ben Liang. Efficient network coded data transmissions in disruption tolerant networks[C]// The 27th Conference on Computer Communications. Phoenix, AZ, USA: IEEE, 2008: 1508-1516.

作者简介:



姚玉坤 女,(1964-),教授.研究方向为网络管理及应用及网络编码.



陈曦 女,(1989-),硕士研究生.研究方向为网络编码. E-mail: xiflycn@sina.cn.

余志龙 男,(1989-),硕士研究生.研究方向为网络编码.

徐亚伟 男,(1992-),硕士研究生.研究方向为网络编码.