

AED: 一种用于 DTN 的增强型 Earliest-Delivery 算法

周晓波 卢汉成 李津生 洪佩琳

(中国科学技术大学电子工程与信息科学系 合肥 230027)

摘要: DTN(Delay-Tolerant Network)是人们在近年出现的众多网络实例中抽象出来的网络模型。在这种网络模型中,经常出现长时间的网络分割,以至于在给定的时间段内端到端的路径不一定存在。由于DTN这种独特的拓扑性质,其路由机制和传统的网络模型有很大不同。该文详细研究了DTN的一种路由算法ED(Earliest-Delivery)算法,然后引入了对节点运动精确性(相对于时间)的考虑,给出了精确性因子 $p(\Delta)$ 的定义,并以之作为ED算法中“链接”(contact)¹⁾代价的权重,进而提出一种更稳健的路由算法——AED(Advanced ED)。实验表明,当节点运动不确定性增加时,AED维持较好的性能,而ED的性能则明显下降。

关键词: DTN; 路由算法; Oracle; ED; AED; 时间精确性

中图分类号: TN915

文献标识码: A

文章编号: 1009-5896(2007)08-1956-05

AED: Advanced Earliest-Delivery Algorithm Used in DTN

Zhou Xiao-bo Lu Han-cheng Li Jin-sheng Hong Pei-lin

(Dept. EEIS., University of Science and Technology of China, Hefei 230027, China)

Abstract: The concept of DTN (Delay-Tolerant Network) is abstracted from many new emerging network instances, which experience frequent long-duration partitions. This particular topology feature makes DTN's routing mechanism very different from typical Internet, in which the existence of a path from source to destination is always guaranteed. This paper focuses on one of DTN's routing algorithm — ED(Earliest-Delivery). A Time-Precision factor is employed to modify the cost of a “contact”, and then a new algorithm named AED(Advanced ED) is introduced. Otherwise, this paper describes a new stochastic process model that is used to implement AED.

Key words: DTN; Routing algorithm; Oracle; ED; AED; Time precision

1 引言

在一些特殊网络环境中,例如功率受限的无线移动、传感器、卫星和星际通信,网络中的节点之间的通信信道不稳定,有很大的延时和延时抖动,甚至可能很长时间处于断开状态。在这种情形下,节点在把信息传递到特定的目的地的过程中会遇到很多的困难:

长延时(long latency) 端到端的延时非常大而且不稳定。这个延时不仅仅来自低的信道容量,更主要是因为节点间的信道可能存在间歇性的、长时间的断开状态。

瞬时不可达(temporally unreachable) 即端到端的路径在某个时间段可能不存在,网络长时间处于分割状态,这也是前一点中提到的节点间信道不稳定性积累起来的表现。

有限存储(finite buffer) 即中间节点的存储空间十分有限。由于数据要经历长的延时,其在网络中滞留的时间很长,这就决定了网络中的存储空间会被很快消耗掉。另外,DTN

的应用场景中,节点功率和性能是受限的,例如传感器网络,这样使得存储问题更加突出。

动态非结构化拓扑(dynamic & unstructured topology) 网络拓扑是非结构化的并随时间变化。在多数情况下,DTN中的节点之间不具有层次关系,即是非结构化的;而另一个方面,节点间信道的不断建立和拆除造成了网络拓扑的不断变化。

人们从具有如上几种特性的网络中抽象出 DTN 的概念^[1]。DTN模型改变了传统网络中的路由协议及算法的基本前提:在通信期间,源节点和目的节点之间存在一条相对稳定的路径。因此,应该为DTN提出新的路由模型及算法,文献[2]中作者为DTN下的路由提出了一个研究框架,并给出了几种路由模型和相关算法,本文在其中一种模型的基础上提出新的路由算法。

文献[2]提出了 DTN 的路由框架,并对可能的模型进行了全面的分析:首先它对 DTN 下的节点先验知识进行了分类,提出了 5 种强度的 Knowledge Oracle(即先验知识),如表 1 所示,然后分析并修正了各种已有的路由算法,使之用于 DTN,并用实验检验了它们性能,如表 2 所示。文献[3]

2005-12-22 收到, 2006-06-19 改回

¹⁾DTN中,用术语“链接”(Contact)来描述两个节点之间的通信信道。本文中一律使用“链接”代替“链路”,“连接”,“会话”等说法。

表 1 节点 Knowledge Oracle

Oracle	描述
0: 无	当前节点不知道任何先验知识, 仅仅从自身所经历的历史信息和当前信息决策路由
1: 链接信息概要	当前节点对网络整体的链接信息有一个统计上的了解, 如各节点间延时的平均值等
2: 链接信息	当前节点明确的知道各节点间的链接信息, 包括何时建立何时拆除, 链接的信道容量等
3: 链接信息 + 队列信息	除了链接信息还知道各节点上的队列长度信息
4: 链接信息 + 队列信息 + 通信需求	在链接信息和队列信息的基础上还知道各节点的业务信息, 例如何时有数据发送, 发往哪里等

表 2 几种代表性路由算法

算法	描述	Oracle needed
DD(Direct Delivery)	源节点等待和目的节点的链接的到来, 直接传递数据	Oracle 0
EP(Epidemic)	当前节点把数据复制 N 份, 分别发送给链接最先到来的 N 个节点, 如果当前不止 N 个链接存在, 则随机选取	Oracle 0
FC(First Contact)	当前节点与第一碰到的节点交换缓存的数据	Oracle 0
MED(Min Expected Delay)	以各链接平均延时作为代价的 Dijkstra 算法	Oracle 1
ED(Earliest Delivery)	以各链接的瞬时延时为代价的修正的 Dijkstra 算法	Oracle 2
EDLQ(ED with Local Queue)	ED 的一种特殊情况, 当前节点知道自己的队列长度	Oracle 2
EDAQ(ED with All Queue)	ED 的一种特殊情况, 当前节点知道所有节点的队列长度	Oracle 3
LP(Linear Program)	当前节点详细掌握网络任意时刻的状况	Oracle 4

将 Erasure-coding 用于 Epidemic 路由模型中, 把数据分为 N 片, 当接收端收到其中任意 M 片 ($M < N$) 就能恢复原数据; 该论文从理论上给出了 N 和 M 对性能极限的影响并进行了实验验证。文献[4]对文献[2]中的几种算法进行了全面的实验, 并提出了一种链接状态维护机制。文献[5]针对一个特定的应用场景, 对节点的行为(主要体现在移动性)进行归类, 并在此基础上建立虚拟空间, 随后的路由算法都建立在该虚拟空间上。文献[6]提出了 DTN 中的多播语义模型。

本文接下来的内容安排如下: 第 2 节分析 ED 算法并提出改进的方案, 即 AED; 第 3 节介绍实验验证及结果分析; 最后在第 4 节总结。

2 改进 ED 算法

DTN 中路由算法的选择与其所在的应用场景密切相关。

虽然无需先验知识的路由算法例如 FC, EP 和 DD 等更具有普适性, 但由于 DTN 多用于如传感器等能量和功能受限的器件组网, 能根据实际情况尽可能多地利用系统信息, 以实现尽可能好的路由效果的算法才更有实用价值。

另外, 文献[2]中所提出的几种 Knowledge Oracle 在实际中很难被实现, 例如“Oracle3 和 Oracle4”要当前节点知道所有节点的队列信息, 这几乎是不可能的——网络中节点之间的状态信息交互不可能如此的频繁, 而且这种状态信息交互的本身也会影响到各节点的状态。

基于上面两点的考虑, 本文将利用 Oracle2, 即网络各节点的链接信息, 来进行分析。

2.1 ED 算法介绍

Dijkstra 算法中, 节点间的 Metric 通常用发送数据包的代价来表示, 而且这种代价一般是固定值。而 ED(Earliest-Delivery)算法中, 两节点间的代价随时间变化, 即代价转变为代价方程。一般情况下代价和延时成正比关系, 因此研究中直接采用延时作为代价。

DTN 中, 数据在相邻节点间的延时来源于: 数据在当前节点排队延时 t_q , 数据在链接上的传输延时 τ 和两节点间链接建立前的等待时间 t_c , 如图 1 所示。在 DTN 中第 3 方面是最主要的; 另外, 延时的 3 个组成方面都是随时间变化的, 也就是说, 路由的决策取决于数据到达节点内队列尾部的时刻; 当然也可以考虑把路由决策时刻放在数据到达队列最前端的时刻, 而把排队延时 t_q 放在接收节点端, 这两种情况是等价的。

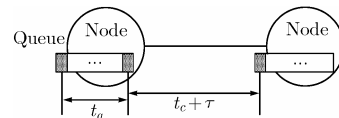


图 1 数据在相邻节点间的延时

这里定义节点 i 和节点 j 之间的代价函数为

$$c_{ij}(t) = t_q(t) + t_c(t) + \tau(t) \tag{1}$$

自变量 t 是数据到达发送节点队列尾部的时刻, 等号右边 3 项分别对应图 1 中的 3 种延时分量。

ED 算法中, 所谓的 Oracle2 是指进行路由决策的节点知道网络中所有节点两两之间的 $c_{ij}(t)$ 。如前面所述, $c_{ij}(t)$ 的主要部分是等待链接到来的延时, 那么网络拓扑随时间的变化就可以表达各节点间的 $c_{ij}(t)$, 因此, 本文用下面的数学模型来描述。网络的拓扑可以用一个随时间变化的 $N \times N$ (设网络中有 N 个节点) 的邻接矩阵 $\mathbf{A}(t)$ 来表达。 \mathbf{A} 是对称矩阵, 且假设对角线上元素取值 0, 即节点与自己是不链接的。这样, 拓扑的每一次变化都可以看作是 \mathbf{A} 的元素的变化, 那么用一个矩阵序列 $\{\mathbf{A}(t): i \in N\}$ 就可以表达一个网络随时间变化的拓扑结构。如图 2 的第一行所示。

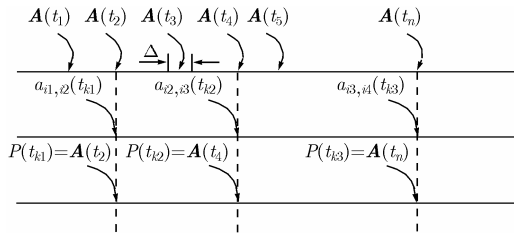


图 2 基于 Oracle2 的路由的矩阵描述

这就是 Oracle2 的一种表示方式，每个节点需要知道这个矩阵序列才能进行寻路。

设 $G=(V, E)$ 是网络的图模型。ED 算法中每次计算出一条最优路径，即图 G 中的节点序列 $(i_1, i_2, i_3, \dots, i_n)$ ，如图 2 的第 2 行所示， $a_{i,j}$ 对应 A 中的一个元素 (相邻两点之间的距离就是对应的代价 c)。由于是源路由，路径一旦计算出来就不能变化，因此 ED 算法效果的好坏很大程度上取决于图 2 中所显示的序列 $(a_{i_1, i_2}, a_{i_2, i_3}, a_{i_3, i_4}, \dots, a_{i_{n-1}, i_n})$ 关于时间的精确性。图 2 的第 3 行表示的是各节点进行转发时的网络状况，即邻接矩阵 A 的值。

文献[2]提出的 ED 对 Dijkstra 算法修改主要在：源节点到工作节点的代价值与不仅与路径中各链路的代价积累有关，而且与路由决策的时刻 T 有关。当计算路由时，所引用的代价值必须是当前时刻的值，而不是传统 Dijkstra 中的恒量。这会带来两个问题：一是路由计算的结果与路由决策的时机相关性非常大，二是路由决策之后网络中各边代价的变化必须严格遵守代价函数 $c(e, t)$ ，也就是说当前节点对 Oracle2 的掌握必须是可靠的，否则计算的路由就不是最优的，甚至导致数据包不能到达目的节点。

ED 算法的基础是 Oracle2，而从上面的算法描述可以看出路由计算的结果对路由决策时刻和每条边上的代价函数的精确性是很敏感的，这对于网络环境通常极为恶劣的 DTN 来说是不好的，因此本文试图在代价函数误差的容忍性上对 ED 算法进行改进。

2.2 AED 算法的提出

文献[2]中指出，采用哪种 Metric 在 DTN 中的路由协议的研究中很重要，因为 Metric 是对网络各种参数的数学概括，如果它不能体现网络的特性，就不能很好的为路由算法服务。在 ED 算法中利用的是每两节点间的链接的代价函数作为 Metric，这是传统路由算法经常采用的，但是在 DTN 中，网络环境变化很快，带宽和拓扑结构都是动态的，此时仅仅采用延时可能不足以描述网络的参数特性。

例如如图 3，当节点 A 向节点 D 发送数据包 (用灰色矩形表示) 时，用 ED 算法计算得到的路径是 $A \rightarrow B \rightarrow C \rightarrow D$ ，总的代价是 0.31；其中，虚线表示当前时刻处于断开状态的链接。如果，当数据包到达节点 B 时， B 和 C 之间的链路按预期建立起来，而 C 和 D 之间由于 C 的“不守时”造成了意外的断开，或者 C 和 D 之间的断开时刻提前到来，这样就造成了数据包无法按预期到达目的地。当然，某个节点的状

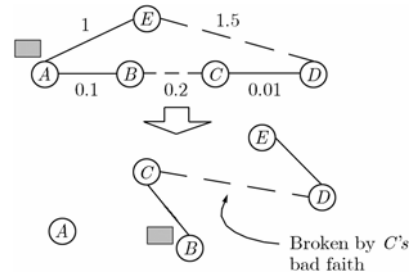


图 3 节点的时间精确性对路由的影响

态突然改变不太可能发生，因此更有可能出现的情况是：路径中每个节点造成一个小的时间上的误差 (延时或超前) Δ_i ，如果不断的积累可能造成在某个节点上实际路由与预期的路由错位。

下面来分析一下 ED 算法下，一个数据包经历整条路径的过程中被丢弃的概率。设在一次路由计算中，形成了一条路径 (用经过的节点表示，包括源和目的节点) $\{i_1, i_2, i_3, \dots, i_n\}$ ，那么，数据包在各节点间的转发过程是：当数据包处于节点 i_k 时，经过了一个时间段 T_k ，在时刻 $t_{k, k+1}$ 节点 i_k 和 i_{k+1} 之间的链接建立，数据包从节点 i_k 发送到 i_{k+1} ，忽略数据包在节点间的传送时间和排队时间，数据包经历的延时就是 T_k 。如此继续，直到目的节点。如图 4 所示：时间轴上的标度 $t_{ik, ik+1}$ 的意思是节点 i_k 和 i_{k+1} 的链接到来的时刻，如果只考虑等待延时，则该时刻也是数据包从节点 i_k 向 i_{k+1} 发送的时刻；图中 t_{i_2, i_3} 和 t_{i_3, i_4} 相距很近，表明当数据包到达节点 i_3 时 i_3 和 i_4 的链接已经存在，数据包能立即被发送到节点 i_4 ，在我们的分析中这两点被当做一点来处理。

如果路径中节点 i_k 的运动和预期有一定的误差，直接影响到的是 T_{k-1}, T_k 和 T_{k+1} 的值，也就是 $t_{k, k+1}$ 和 $t_{k-1, k}$ 在时间轴上的位置发生移动，我们假设这种影响是正态的，即 $t_{k, k+1}$ 和 $t_{k-1, k}$ 的值，在以其原始值 (即节点 k 运动和预期一致的情形) 为中心的附近正态分布，并假设方差为 σ^2 ，如图中所示。

基于图 4，我们来讨论该路径上的丢包概率。当节点运动很精确时，丢包可以认为只来自于队列溢出，把节点 i_k 上丢包的概率用 $P(T_k)$ 表示，其中 T_k 是数据包在节点 i_k 上滞留的时间，显然 T_k 越大丢包的概率越大。这样易得总体的丢包概率为

$$P = 1 - \prod_{k=1}^{n-1} (1 - P(T_k)) \tag{2}$$

当节点运动不精确时，丢包来自两个方面，一是队列溢出，另一个是链接到来的时刻发生错位，例如如图 4 中 t_{i_5, i_6} 和 t_{i_6, i_7} 交换位置，这种情况下丢包是必然的。对于队列溢出丢包，

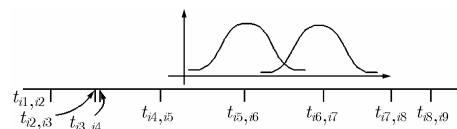


图 4 节点沿选定路径传递数据包的过程

节点 i_{k-1}, i_k 和 i_{k+1} 会受到影响：在保持 $T_{k-1} + T_k + T_{k+1}$ 不变的

情况下,重新分配 T_{k-1} , T_k 和 T_{k+1} 的值,这会改变 P 的大小。另外,如图4,在 $t_{i5,i6}$ 和 $t_{i6,i7}$ 错位的时候,丢包必然发生,这个概率为

$$\frac{1}{2\pi\sigma^2} \int_{-\infty}^{+\infty} \left[\int_{-\infty}^y e^{-\frac{(x-A)^2}{\sigma^2}} dx \right] \times \left[\int_y^{+\infty} e^{-\frac{x^2}{\sigma^2}} dx \right] dy$$

其中 A 为发生偏移的两个时刻 $t_{ik-1,ik}$ 和 $t_{ik,ik+1}$ 的原始值(即预期值)的差。这样得到总的丢包概率为

$$P = 1 - \frac{1}{2\pi\sigma^2} \int_{-\infty}^{+\infty} \left[\int_{-\infty}^y e^{-\frac{(x-A)^2}{\sigma^2}} dx \right] \times \left[\int_y^{+\infty} e^{-\frac{x^2}{\sigma^2}} dx \right] dy \times P'(T_{k-1}) \times P'(T_k) \times P'(T_{k+1}) \times \prod_{m=1}^{n-1} P(T_m) \quad (3)$$

$m \neq k-1, k, k+1$

和式(2)相比,重要差异来自积分项,而这一项必然小于1,也就是说丢包概率增加。这是当一个节点运动不精确的情形,当多个节点发生这种情况,这个增加量会更加明显。

基于以上的考虑,本文在 ED 算法的 Metric 中引入对节点运动的时间精确性的描述。这里我们定义各节点时间精确性因子 $\rho_i(\Delta)$, 如图2第1行所示: $\rho_i(\Delta)$ 表示节点 j 与 i 链接建立的实际时刻处在预定时刻 t 的 Δ -邻域中的概率。 $\rho_i(\Delta)$ 的变化特性能反映出网络各节点的整体行为对在时间上的精确性, $\rho_i(\Delta)$ 变化越平缓则精确性越低,即链接可能在更宽的时间区域内出现,这对于 ED 算法是很不利的;本文希望将 $\rho_i(\Delta)$ 作为一个权重因子去修正 ED 中节点间传输数据的代价函数,精确度越高的节点代价降低越多。这样我们修正节点间的代价函数为

$$c'_{ij}(t) = (t_q(t) + t_c(t) + \tau(t)) \times (1 - \rho_j(\Delta)) \quad (4)$$

从式(4)中可以看出,因为各节点的精确性因子不一样,两节点间的代价函数不再是对称的了。另外, Δ 的取值应尽可能地使 ρ 分布在尽可能宽的范围,这取决于各种不同类型的网络的特性参数:对于节点行为较固定,对时间的精确性较高的网络可以选择较小的,反之选择较大的。还有,不能直接用新的代价函数 $c'_{ij}(t)$ 替换 ED 算法中的 $c_{ij}(t)$, 因为它不保留延时信息,即数据到达下一个节点的时刻还必须由原始的代价函数 $c_{ij}(t)$ 来确定。

3 仿真及结论

本文用 NS 仿真工具对 DTN 中现有的路由算法进行了全面的实验比较,着重比较了 ED 和 AED 算法的性能。仿真中共有 10 个节点,每两个节点之间的链接按给定的规则建立和拆除,也就是说给定代价函数 $c_{ij}(t)$, 该函数值域为 $\{0, 1\}$ 。两节点间的链接的持续时间约 1s 量级,而处于断开状态的时间约 10s 量级。在这种网络设定下,每隔一定时间随机选取一个源节点和目的节点对,发送一个数据包。仿真过程中共发送约 2000 个数据包,每个数据包大小约 1000byte,

记录每种路由算法下的平均延时、延时抖动、路由成功率、平均跳数和平均存储消耗等。

本实验仿真了 AED, ED, MED, DD, FC 和 EP 等 6 种路由算法的性能。对不同缓存能力下和不同时间精确性下两种情形进行了仿真。

情形 1 让各节点的缓存从 1kB 逐步增加到 10kB, 各算法的表现如图 5, 图 6 所示。

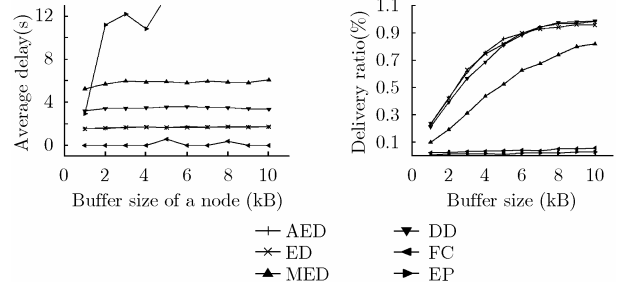


图5 随着各节点缓存能力的加大各算法下的延时 图6 随着各节点缓存能力的加大各算法下的路由成功率

从图 5 可以看出缓存能力的加大对延时影响不大。FC 算法有最好的延时性能,其次是 AED 和 ED。AED 只是比 ED 略好一点,而 FC 为什么会有如此好的性能呢? FC 规定节点把要转发的数据包简单地交给第 1 个碰到的节点,如果当前有多个节点与之相连则随机选取一个,理论上说这是不可能达到很好的性能的;事实上,从图 6 我们将看到这里如此低的延时是以很低的路由成功率为代价的,而且那些成功传递到目的节点的数据包都只经历了一跳,因此低延时就可以理解了。

从图 5 中还看到 EP 算法的延时非常大,最大值是 15.1199s。按 EP 算法,每个转发节点把数据包复制 K 份,本实验中 $K=3$,发送给最先到达的 K 个节点。理论上讲,当 K 等于网络中节点个数时,EP 算法在不要任何 Oracle 的支持下,也必然能找到那条从源到目的的最佳路径,它应该有很好的性能的。然而,EP 算法极大地消耗网络的缓存,从实验中我们看到,EP 算法下,各节点的队列始终处于饱和状态,因此,很多数据包被丢弃了,另外本实验中采用的是 FIFO 队列调度算法,这就使某些本就不能到达目的节点的数据包给那些可以到达目的节点的数据包带来了延时。改变队列调度算法是一个改进方案,但不能从根本上解决 EP 算法的性能;可以得出结论:EP 算法只适用于缓存能力很强或者业务量很稀疏的网络中。

图 6 显示的是各算法的路由成功率, AED 是最好的,只是在缓存大于 8kB 后 DD 算法的性能超过了 AED。DD 算法下每个数据包必须等到到目的节点和源节点直接相连,当缓存很大时,数据包不会因为队列满而丢弃,因此可以等待的时间就更长,到达目的节点的机会就越大,当缓存大到一定程度其路由成功率可以达到 100%;当然延时不会有很大改善。

情形 2 因为 AED 是对 ED 关于时间精确性的改进,本

实验的重点是检验时间精确性对各算法性能的影响。这里的精确性用 Δ -邻域的长度相对于相邻两事件间的平均时间间隔的比例来表达, 越大则网络整体精确性越差。图 8 的横轴 time jitter 即用链接依据 Oracle 的设定应该变化的时间间隔和实际的变化时间间隔的差除以前者得到。

图 7 显示的是随着精确性的降低, 各算法下的平均延时情况。可见精确性对延时性能的影响不大, 这是因为, 在 DTN 中, 延时更多的决定于式(1)的第 2 项 $t_c(t)$, 精确性对 $t_c(t)$ 的改变量是微不足道的。图 7 中, AED 和 ED 不相上下。

图 8 是路由成功率, 随着精确性的降低, AED 和 MED 基本维持不变, ED 和 DD 明显下降。ED 用到了网络中各节点代价函数的完全信息, 因此, 当实际代价函数与路由计算时的存在差异时就降低了路由成功率。这里就可以看出 AED 表现出了较好的对精确性的容忍能力, 平均提高 8.3%。

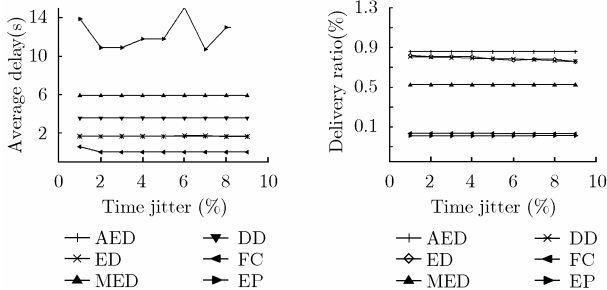


图 7 随着时间精确性变化各算法下的延时

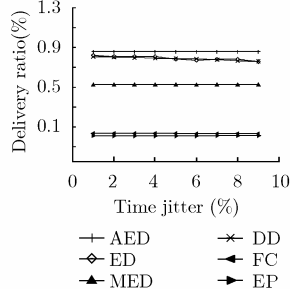


图 8 随着时间精确性变化各算法下的路由成功率

结论 (1)在两种实验情形下, AED 在维持基本性能不差于 ED 的情况下, 在路由成功率对时间精确性的容忍性上明显优于 ED 算法。(2)在综合性能上, 包括平均延时、路由成功率和缓存消耗(未给出图示)几个方面, AED 和 ED 优于其他算法; 也就是说 AED 和 ED 算法的适应面更广。(3)增大节点的缓存会明显提高各种算法的性能。(4)EP 算法只适用于缓存能力很强或业务稀疏的情形。

4 结束语

本文从路由协议运行的实际网络环境出发, 提出路由算法应该加入对 Oracle 的精确性的考虑, 并在 ED 算法的基础

通过 NS 仿真, 全面地分析了各种算法在改变节点缓存能力和改变节点时间精确性两种情形下的性能, 并证实了 AED 确实改善了 ED 对时间精确性的容忍能力。

但是本文并没有对 Oracle2 的实现进行分析, 如果能根据节点的运动精确性定义其运动模式, 并由此定义向量空间, 在这个虚拟空间中定义节点间距, 然后以之作为路由算法的 Metric, 将是一种很好的思路, 这正是我们下一步要研究的内容。

参考文献

- [1] Fall Kevin, *et al.*. A delay-tolerant network architecture for challenged Internets. Sigcomm'03, Karlsruhe, Germany, August 25-29, 2003: 27-34.
- [2] Jain Sushant, *et al.*. Routing in a delay tolerant network. Sigcomm'04, Portland, Oregon, USA, Aug. 30-Sept. 3, 2004: 145-158.
- [3] Wang Yong, *et al.*. Margaret Martonosi, Kevin Fall. Erasure-coding based routing for opportunistic networks. Sigcomm'05, workshops, Philadelphia, USA, August 22-26, 2005: 229-236.
- [4] Jones P C, *et al.*. Practical routing in delay-tolerant networks. Sigcomm'05, workshops, Philadelphia, USA, August 22-26, 2005: 237-243.
- [5] Leguay emie, *et al.*. DTN routing in a mobility pattern space. Sigcomm'05, workshops, Philadelphia, USA, August 22-26, 2005: 276-283.
- [6] Zhao Wenrui, *et al.*. Multicasting in delay tolerant networks semantic models and routing algorithms. Sigcomm'05, workshops, Philadelphia, USA, August 22-26, 2005: 268-275.

周晓波: 男, 1981 年生, 博士生, 研究方向为宽带无线网络技术和 DTN.

卢汉成: 男, 1977 年生, 博士, 讲师, 研究方向为下一代互联网网络技术和无线网络技术.

李津生: 男, 1937 年生, 教授, 研究方向为信息通信网技术.

洪佩琳: 女, 1960 年生, 教授, 研究方向为信息通信网技术.

上通过加入节点的时间精确性因子 ρ 进而提出 AED 算法。