

一种基于 F S M 的告警事件关联方法

唐 勇¹, 张 欣², 周明天¹

(1. 电子科技大学 计算机科学与工程学院, 四川 成都 610054; 2. 电子科技大学 通信与信息工程学院, 四川 成都 610054)

摘 要: 对告警事件进行关联处理, 去除冗余告警, 是网络管理需要解决的一个关键问题。如果考虑事件间的时间关系, 问题将变得更为复杂。因此, 在充分考虑事件间的时间关系基础上, 提出了一种基于有限状态机 (FSM) 的事件关联模型, 并利用该模型设计了一个告警关联处理器, 它能够正确地实现事件关联, 有效减少冗余告警的发生。

关键词: 网络管理; 告警事件关联; 有限状态机

中图法分类号: TP393 文献标识码: A 文章编号: 1001-3695(2006)09-0243-04

FSM Based Alarm Event Correlation

TANG Yong¹, ZHANG Xin², ZHOU Ming-tian¹

(1. College of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu Sichuan 610054, China; 2. College of Communication & Information Engineering, University of Electronic Science & Technology of China, Chengdu Sichuan 610054, China)

Abstract: Alarm event correlation, which can reduce redundant alarm events, is a key problem to network management. If considering time relationship between events, the problem becomes more complicated. Under the consideration of time factor, a correlation model based on FSM is put forward, and alarm correlation processor is implemented, which correlates the events correctly and reduces the redundant alarms efficiently.

Key words: Network Management; Alarm Event Correlation; FSM

随着现代网络技术的发展, 网络日趋庞大和复杂, 网管系统的故障冗余告警问题也日益突出。因此进行告警事件关联研究, 去除冗余告警, 是网络管理领域一个重要的研究课题。目前, 有许多方法可以用于告警事件关联^[1~4]。例如, 基于因果关系的关联、基于编码的关联、基于模型推理的关联、基于事例推理的关联、基于规则的关联、模糊逻辑、贝叶斯网络、神经网络、数据挖掘、有限状态机 (FSM)、Petri 网络等。

基于对各种方法的比较分析, 本文提出了一种基于 FSM 的告警事件关联方法, 即在利用事件驱动来选择时间窗口的基础上, 采用 FSM 实现告警事件关联。由于 FSM 本身也是事件驱动的, 因此可以方便地实现关联, 并易于编程实现。本文利用 FSM 进行了事件关联建模, 并利用该模型设计了能够处理复杂网络事件的告警事件关联模块。

1 基于 FSM 的事件关联方法

1.1 告警事件的定义

为使用 FSM 进行告警事件关联建模, 需要定义告警事件。在总结文献[5~9]的基础上, 并结合实际经验, 利用巴柯斯范式 (BNF) 定义一个告警事件如下:

< Event > :: = < PrimitiveEvent > | < CompositeEvent >
< PrimitiveEvent > :: = < EventName >, < EventID >,
< TimeStamp >, < EventPriority >,

< EventSource >, < AttributeList >
< CompositeEvent > :: =
< PrimitiveEvent > < EventBiOP > < PrimitiveEvent > |
< CompositeEvent > < EventBiOP > < PrimitiveEvent > |
< CompositeEvent > < EventBiOP > < CompositeEvent > |
< PrimitiveEvent > < EventBiOP > < CompositeEvent > |
< EventUniOP > < CompositeEvent > |
< EventUniOP > < PrimitiveEvent >
< EventBiOP > :: = && | ~ | →
< EventUniOP > :: = n
< EventName > :: = String
< EventID > :: = Integer
< TimeStamp > :: = String
< EventPriority > :: = Integer
< EventSource > :: = String
< AttributeList > :: = (< Attribute > < RelationOP > < Value >) *
< Attribute > :: = String
< RelationOP > :: = > | < | = | > = | < = | ! =
< Value > :: = Integer | String | Any

以上范式用递归的方式, 定义了事件、简单事件、复合事件和事件的运算。事件分为简单事件和复合事件: 一个简单事件由六元组 (EventName, EventID, TimeStamp, EventPriority, EventSource, AttributeList) 唯一决定; 复合事件是由一系列的简单事件、复合事件经过事件的运算得到的一种事件。事件的运算就是事件关联的方法, 复合事件的构成过程就是事件关联的过程。

本文定义了一系列运算符来表示事件的运算, 包括: →

是二元运算符, $E_1 \rightarrow E_2$ 表示在不超过一定的时间间隔 T 内, 事件 E_1 发生在事件 E_2 之前; n 是一元运算符, $n(E_1)$ 表示在 E_1 第一次发生后的 t 内, E_1 如果再次重复发生 n 次, 将 n 次压缩为一次; $\&\&$ 是二元运算符, $E_1 \&\&E_2$ 表示两个事件在不超过 t 内相继发生; $\vee$ 是二元运算符, $E_1 \vee E_2$ 表示两个事件之一发生, 或者两个事件均发生; $\sim$ 是二元运算符, $E_1 \sim E_2$ 表示两个事件在不超过 t 内, E_1 发生, E_2 并不发生。

为便于处理, 将定时器溢出事件 Timeout 也定义为一个告警事件, 其事件名(EventName) 为 Timeout。

1.2 告警事件关联的类型与方法

关联表示了两个或更多实体之间具有相互联系的情况, 关联的结果有两种: 信息语义内容增加; 独立单元总数缩减。对告警事件关联可作以下形式化定义^[4]: 告警事件 α 与告警事件集合 $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ 关联, 表示为 $\alpha \Rightarrow \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ 。

告警事件关联的类型可以根据需要进行定义^[4]。可定义如下关联类型:

- (1) 告警压缩。将多个告警压缩为一个告警: $[A, A, A, \dots, A] \Rightarrow A$ 。
- (2) 告警过滤。如果告警 A 的 $P(A)$ 值不属于合法集合 U , 则过滤掉告警 A : $[A, P(A) \notin U] \Rightarrow \phi$ 。
- (3) 告警抑制。在高优先级告警 A 发生时, 抑制低优先级告警 B : $[A, B] \Rightarrow A$ 。
- (4) 告警泛化。用告警的超类代替该告警: $[A, A \subset B] \Rightarrow B$ 。
- (5) 告警特化。用告警的特定子集代替该告警: $[A, A = B] \Rightarrow B$ 。
- (6) 告警时序关系。关联的告警依赖于告警发生时间顺序, 告警 A, B 顺序发生时, 则会发生告警 C : $[A \text{ T } B] \Rightarrow C$ (T 表示时间顺序 Before/After)。

1.3 FSM的概念及表示

FSM定义为如下六元组 M :

$M = (I, O, S, s_0, \delta, \lambda)$

I 是有限非空的输入符号集合。

O 是有限非空的输出符号集合。

S 是有限非空的状态集合。

$s_0 \in S$ 是初始状态。

$\delta: S \times I \rightarrow S$, 是状态转移函数。当有限状态机 M 处于状态 $s \in S$ 时, 接收到输入 $i \in I$, 转移到下一个状态 $\delta(s, i) \in S$ 。

$\lambda: S \times I \rightarrow O$, 是输出函数。当有限状态机 M 处于状态 $s \in S$ 时, 接收到输入 $i \in I$, 则输出 $\lambda(s, i) \in O$ 。

可以使用有向图来形象地表示 M : 一个有向图 $G = (V, E)$ 是一个有序的二元组, 其中 $V \neq \phi$, 是 G 的顶点集合, E 是笛卡尔积 $V \times V$ 的多重子集, 其元素称为有向边。用图 G 这样来表示一个 FSM: V 与 S 一一对应, 有向边表示了状态的转移关系。并且在有向边上进行标注: 在有向边 $s_1 s_2$ 上标上一个二元有序对 (i, o) , 用 i/o 来表示, $i \in I, o \in O$, 表示了 $\delta(s_1, i) = s_2, \lambda(s_1, i) = o$ 。

本文使用 E 表示所有告警事件的集合。包括简单事件、复合事件和定时器溢出事件, ε 为空事件, $I = E, O = E \cup \{\varepsilon\}$ 。

2 事件关联的 FSM建模

要^[2,5]。文献[5]提出了一种较好的基于 Petri 网的关联方法, 考虑到了时间因素, 但没有确定时间窗口的起始和大小的选取, 一般采取等分窗口和大尺度时间窗口的方式。但复杂网络中告警发生时间在网络稳定情况下难以预测, 待关联事件不一定落在所选的时间窗口内, 这必将导致关联失效, 而大尺度时间窗口将会由于告警事件的丢失导致关联不正确。文献[2]也指出了时间窗口在事件管理中的不足。鉴于以上由时间窗口引发的问题, 本文提出了一种基于事件驱动来选择时间窗口的 FSM事件关联方法: 在某个关键告警事件出现时触发时间窗口大小的设定, 窗口尺寸的选择依赖于相关告警事件的时间间隔概率分布, 这可以通过对历史数据进行统计分析得到, 采用满足系统要求的落在窗口外的丢失事件的概率 $p < \varepsilon$ 来设计窗口尺寸。

使用该方法, 可以有效避免以上由时间窗口引发的问题。下面对该 FSM事件关联的建模方法进行详述。

2.1 基于 FSM的事件关联建模

对于前面定义的事件关联关系, 本文建立了相应的 FSM模型。一共建立了六种 FSM模型, 即一种简单事件模型和五种复合事件模型。下面对这六种模型的意义进行阐述。重点以简单事件、二元操作符 $Event_1 \rightarrow Event_2$ 、一元操作符 $n(Event)$ 加以详述。以下模型中, t 为时间窗口, 其选取方式按照本文前述方法进行。

图 1 为一个简单事件的 FSM模型。对于事件集合 Event 可以采用复杂的逻辑表达式来表示。

图 2 为关联规则 $Event_1 \rightarrow Event_2$ 的 FSM模型, 表示事件 $Event_1$ 先发生, 在事件 $Event_1$ 发生后的不超过 t 的时间间隔内, 事件 $Event_2$ 发生。 $Event_1$ 为驱动事件。此 FSM模型的所有可能情况为:

- (1) 事件 $Event_1$ 先发生, 在不超过 t 时间内, $Event_2$ 也发生;
- (2) 事件 $Event_1$ 先发生, 在超过 t 时间内, 事件 $Event_2$ 不发生;
- (3) 事件 $Event_2$ 先发生。

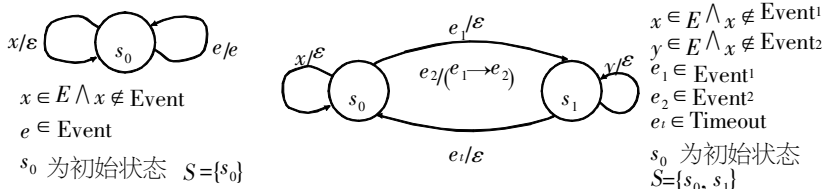


图 1 简单事件

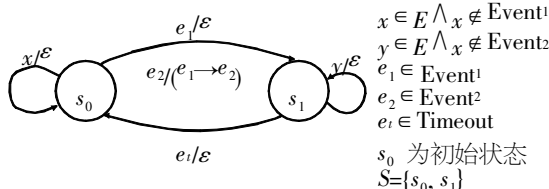


图 2 $Event_1 \rightarrow Event_2$

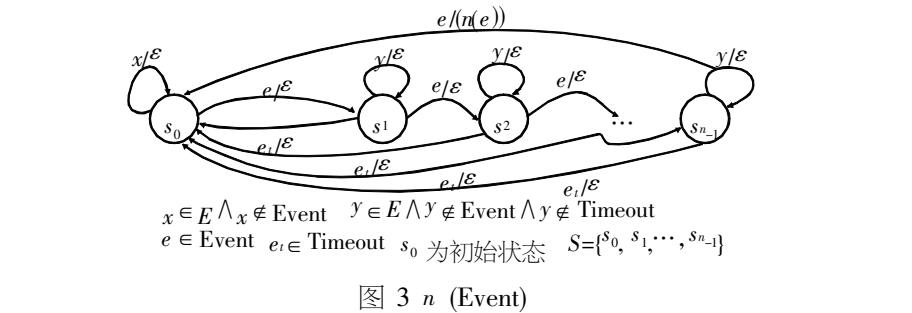
图 3 为关联规则 $n(Event)$ 的 FSM模型, 表示在事件 $Event$ 第一次发生后, 在未来的时间间隔 t 内, 将事件 $Event$ 的 n 次发生压缩为一次发生。第一个 $Event$ 为驱动事件。此 FSM模型的所有可能情况为:

- (1) 事件 $Event$ 发生后, 恰有 n 的整数倍的事件 $Event$ 发生;
- (2) 事件 $Event$ 发生后, 事件 $Event$ 发生不是 n 的整数倍。

图 4 为关联规则 $Event_1 \vee Event_2$ 的 FSM模型。

图 5 为关联规则 $Event_1 \&\&Event_2$ 的 FSM模型, 表示事件 $Event_1, Event_2$ 相继发生的时间间隔不超过 t 。 $Event_1, Event_2$ 为驱动事件。此 FSM模型的所有可能情况是:

- (1) 事件 $Event_1$ 先发生, 在不超过 t 时间内, 事件 $Event_2$ 也发生;



- (2) 事件 Event_2 先发生, 在不超过 t 时间内, 事件 Event_1 也发生;
- (3) 事件 Event_1 先发生, 在超过 t 时间内, 事件 Event_2 不发生;
- (4) 事件 Event_2 先发生, 在超过 t 时间内, 事件 Event_1 没发生。

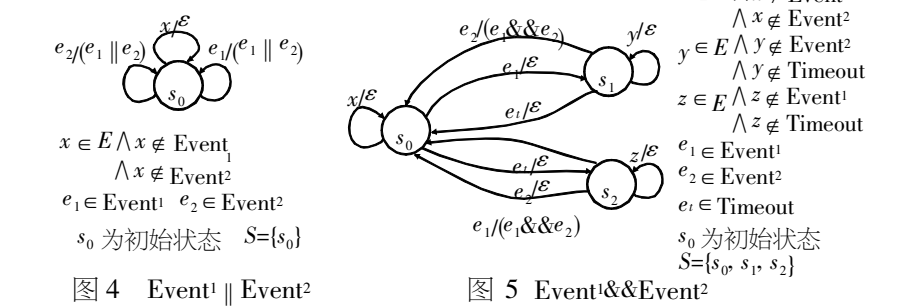
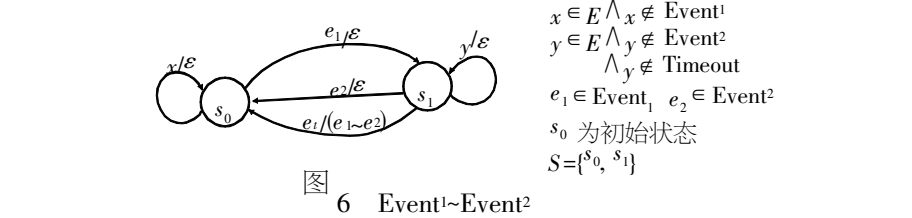


图 6 为关联规则 $\text{Event}_1 \sim \text{Event}_2$ 的 FSM 模型, 表示在事件 Event_1 发生后的时间间隔 t 内, 事件 Event_2 不发生。 Event_1 为驱动事件。此 FSM 模型的所有可能情况是:

- (1) 事件 Event_1 发生后, 在超过 t 时间内, Event_2 不发生;
- (2) 事件 Event_1 发生后, 在不超过 t 时间内, Event_2 也发生;
- (3) 事件 Event_2 先发生。



2.2 利用 FSM 模型处理告警事件

使用上面建立的六种 FSM 模型中的一种或模型的相互组合, 可以完成 1.2 节所列所有告警事件关联类型的处理。其中, 模型中的事件可以是简单事件或复杂事件。

- (1) 使用 $n(\text{Event})$ 的 FSM 模型, 如果告警事件 Event 多次发生, 可将 $n(\text{Event})$ 映射为一个简单事件, 完成告警压缩的关联处理;
- (2) 使用简单事件的 FSM 模型, 可用 $P(\text{Event}) \cup$ 限定事件的类型, 完成告警过滤;
- (3) 使用 $\text{Event}_1 \&\& \text{Event}_2$ 的 FSM 模型, 如果 Event_1 和 Event_2 同时发生, 将关联的结果映射为高优先级告警事件 Event_1 , 抑制了低优先级告警 Event_2 , 完成告警抑制;
- (4) 如果 $\text{Event}_1 \subset \text{Event}_2$, 使用 $\text{Event}_1 \sim \text{Event}_2$ 的 FSM 模型, 当 Event_1 或 Event_2 发生, 可将 $\text{Event}_1 \sim \text{Event}_2$ 映射为 Event_2 , 完成告警泛化;
- (5) 如果 $\text{Event}_1 \supset \text{Event}_2$, 使用 $\text{Event}_1 \sim \text{Event}_2$ 的 FSM 模型, 当 Event_1 或 Event_2 发生, 将 $\text{Event}_1 \sim \text{Event}_2$ 映射为 Event_2 , 完成告警特化;
- (6) 使用 $\text{Event}_1 \rightarrow \text{Event}_2$ 的 FSM 模型, 如果 Event_1 和 Event_2

先后发生, 可将关联结果 $\text{Event}_1 \rightarrow \text{Event}_2$ 映射为简单事件 Event_3 , 完成告警时序关系的关联处理。

使用上述关联模型的组合, 可实现更复杂的关联处理。

3 告警关联处理器的实现

根据上文建立的告警事件关联的 FSM 模型, 设计了告警关联处理器。告警关联处理器使用组件的设计思想, 使用多线程的消息驱动方式进行设计, 它首先根据关联模型设计出上述六种事件关联方式, 然后对其进行组合, 从而实现复杂的事件关联。

在实现中, 还需要定时服务器和事件接收器。定时服务器负责发送定时器溢出事件给相应的订购者; 事件接收器负责从网络中接收网络告警事件。

3.1 单个 FSM 的程序实现

利用定义好的状态机, 可以容易地进行编程实现。其一般流程如下:

- (1) 输入事件 e 到达;
- (2) 确定当前状态 s , 判断状态转移和事件输出情况, 如果输出为空事件, 并且下一个状态 s_n 与 s 相同, 不做任何动作; 否则转(3);
- (3) 根据相应的 λ, δ 函数, 确定输出事件并完成状态转换。

一种事件关联方式对应于一种状态机实现, 一次事件关联实例对应于一个状态机实例。以下为 $\text{Event}_1 \rightarrow \text{Event}_2$ 为例的伪码编程实现。事件相关函数为 $\text{Ev}_1 \text{LeadEv}_2$ 。

```
class MachineEv1LeadEv2{
    Event1 event1;
    Event2 event2;
    State s;
    ev1LeadEv2( E e );
    ... ;
MachineEv1LeadEv2::ev1LeadEv2( E e )
{
    if ( this->s = s0 ) {
        if ( e == m.event1 ) { 请求定时器溢出事件; 修改 this->s 为 s1; }
        else{ 不做任何事 }
    } else if ( this->s = s1 ) {
        if ( e == m.event2 ) { 输出事件( event1 → event2 ); 修改 this->s 为 s0; }
        else if ( e == m.Timeout ) { 修改 this->s 为 s0; }
        else{ 不做任何事 }
    } else { 初始化状态 this->s 为 s0 }
}
```

3.2 组合的告警关联处理器设计

告警关联处理器的组件主要由六个事件处理线程实现。每个事件处理线程均有一个状态机实例队列, 事件到达后, 事件处理线程在各个状态机间循环调用事件相关函数。事件分发线程则是将事件转发给下一步的处理者, 如果无下一步处理者, 表示该事件的处理已完成, 转发给最终的接收者, 即缺省事件订购者。

线程间通信采用队列形式, 除简单事件和 $\text{Event}_1 \sim \text{Event}_2$ 事件处理线程只有一个待关联事件队列外, 其他事件处理线程均有两个队列, 即待关联事件队列(包括简单事件和复合事件) 和定时事件队列。定时事件有最高的优先级, 首先处理定时事件。

告警关联处理器的设计如图 7 所示。其关联处理过程是：

(1) 事件接收者从告警源接收告警事件；

(2) 事件接收者按需将告警事件分发给六个事件处理线程，如果该事件无下一步的处理者，则分发给缺省事件订购者；

(3) 事件处理线程处理后分发给其他事件处理线程(包括该线程自身)，如果该事件无订购者，则分发给缺省事件订购者；

(4) 缺省事件订购者将告警事件转发给需要告警事件的其他系统，如数据库、浏览器界面等。

实验表明，使用 FSM 实现的告警关联处理器，能正确进行告警事件关联处理。并且，关联的有效性与时间窗口的选择密切相关，窗口越大关联越有效，但同时降低了告警的实时性。在下一步工作中，将研究使用多重时间窗口设定的方式来提高事件关联的实时性和效率。

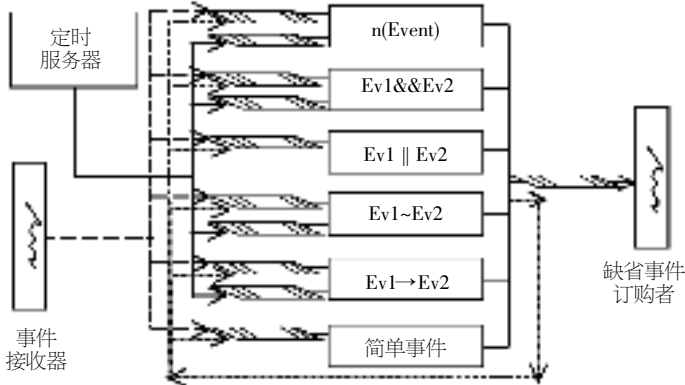


图 8 告警关联处理器

4 结束语

本文根据告警事件关联的需要，提出了一种基于 FSM 的事件关联模型，并利用该模型，设计了一个告警关联处理器，该处理器在实际的网络环境中已得到应用，正确地进行了告警事件关联，有效地减少了冗余告警的发生。该处理器具有较强的通用性，可用于各种网络管理系统，完成复杂网络的告警事件关联处理。

(上接第 242 页)

6 结束语

根据仿真实例配置的参数，所建模型中的性能指标评估结果与理论分析和实际情况一致。当网络流量和节点数目适中时，DSR 路由协议可以快速地建立路由发送数据，其准确性优于 AODV。考虑路由开销和能耗时，AODV 与 DSR 在高效性方面各有所长：AODV 分组传递率较低，端到端吞吐量较少，所以扩展性次于 DSR，因此应用于规模较小的网络中；而 DSR 的综合性能高于 AODV。

基于 NS-2 对 MANET 路由协议仿真和性能评估模型还能够实现新路由协议的嵌入及调用，具有良好的实用性和较强的通用性。

参考文献：

[1] 赵志峰，郑少仁. Ad hoc 网络体系结构研究[J] . 电信科学，2001，(1) : 14-17.

[2] Samir R Das，Charles E Perkins，Elizabeth M Royer. Performance Comparison of Two on-demand Routing Protocols for Ad hoc Networks [C] . Proceedings of the IEEE Conference on Computer Communications (INFOCOM)，2000.

[3] Josh Boroch，David A Maltz，David B Johnson, et al. A Performance Comparison of Multihop Wireless Ad hoc Network Routing Protocols

参考文献：

[1] ouayad Albaghdadi, et al. A Framework for Event Correlation in Communication Systems[C] . IFIP/IEEE International Conference on Management of Multimedia Networks and Services, MMNS, 2001. 271-284.

[2] Mal gorzata Steinder, Adarshpal S Sethi. Probabilistic Fault Localization in Communication Systems Using Belief Networks [J] . IEEE/ACM Transactions on Networking, 2004, 12(5) : 809-822.

[3] Malgorzata Steinder, Adarshpal S Sethi. A Survey of Fault Localization Techniques in Computer Networks[J] . Science of Computer Programming, 2004, 53(2) : 165-194.

[4] 夏海涛，詹志强. 新一代网络管理技术[M] . 北京：北京邮电大学出版社，2003.

[5] 王平，李莉，赵宏. 网络管理中事件关联检测机制的研究[J] . 通信学报，2004, 25(3) : 73-81.

[6] Ehab Al-Shaer. Active Management Framework for Distributed Multimedia Systems [J] . Journal of Network and System Management, 2000, 8(1) : 49-72.

[7] Alarm Reporting Function[S] . ITU_T X. 733-1992.

[8] Gabriel Jakobson, MarkD Weissman. Alarm Correlation: Correlating Multiple Network Alarms Improves Telecommunications Network Surveillance and Fault Management[J] . IEEE Network, 1993, (11) : 52-59.

[9] GaltonA, J C Augusto. Two Approaches to Event Definition[C] . France: Proc. of the 13th Int. Conference on Database and Expert Systems Applications, Lecture Notes in Computer Science 2453, 2002.

作者简介：

唐勇，男，云南云县人，博士研究生，研究方向为无线网络、网络计算与网络管理；张欣，女，硕士研究生，研究方向为无线网络、多媒体通信与宽带通信网；周明天，男，教授，博导，研究方向为为网络计算、分布式计算与信息安全。

[C] . Proceedings of MOBICOM 98, 1998. 85-97.

[4] 王海涛，郑少仁. 移动 Ad hoc 网络的路由协议及其性能比较[J] . 重庆邮电学院学报，2002, 14(4) : 73-77.

[5] S Corson，J Macker. MANET: Routing Protocol Performance Issues and Evaluation Considerations[S] . Internet RFC 2501, 1999.

[6] Kevin Fall，Kannan Varahan. The Ns Manual [EB/OL] . <http://www.isi.edu/nsnam/ns/ns-documentation>, 2003.

[7] Jiann-Min Ho，Janak Sanjaykumar Mehta, et al. Design and Evaluation of Mobility Models in Wireless Ad hoc Networks[EB/OL] . <http://www.cs.cmu.edu/desney/5824/>.

[8] C Perkins，E Belding-Royer，S Das. Ad hoc On-Demand Distance Vector (AODV) Routing [EB/OL] . <http://www.internet-draft/draft-ietf-manet-aodv-09.txt>, IETF MANET Working Group, 2001.

[9] David B Johnson，David A Maltz，Yih-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR) [EB/OL] . <http://www.internet-draft/draft-ietf-manet-dsr-05.txt>, IETF MANET Working Group, 2001.

[10] AODV-UU [EB/OL] . <http://user.it.uu.se/~henrik/aodv/>.

作者简介：

牛秋娜(1977-)，女，助教，硕士研究生，主要研究方向为网络通信及安全性；王美琴(1974-)，女，博士研究生，主要研究方向为网络安全与网络测试；王英龙(1965-)，男，研究员，博士研究生，主要研究方向为网络及信息安全；徐永道(1978-)，男，硕士研究生，主要研究方向为网络通信及安全技术。