Reprinted from the

Proceedings of the
Ottawa Linux Symposium

June 26th–29th, 2002
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc.*
Stephanie Donovan, *Linux Symposium*
C. Craig Ross, *Linux Symposium*

Proceedings Formatting Team

John W. Lockhart, *Wild Open Source, Inc.*

Mobile Cluster Computing Using IPv6

Abdul Basit

Chin-Chih Chang

Wichita State University

Dept. of Computer Science

Wichita, KS 67260-0083, USA

{axbasit, chang}@cs.twsu.edu   http://wireless.cs.twsu.edu

Abstract

Clusters play a major role in scientific computing. They eliminate the need of supercomputers. Communication protocols for cluster computing define efficiency and performance measures for overall system design.

Using IPv6 as a protocol for cluster computing gives benefits such as

- Network load balancing can make use of IPv6 Congestion/Non-Congestion traffic mechanisms (e.g. for handling real-time data requests on clusters).

- Geographically distributed cluster system can make use of embedded IPv6 route optimizations.

- IP Anycast service has the ability to choose the topologically closest server available for handling the request. So it can be used to effectively load balance the network traffic.

- IPv6 Authentication Headers (AH) can be used to define what workstations are allowed to join the cluster.

Moreover, Mobile IPv6 [5] allows transparent geographic mobility without affecting the present connections. This behaviour may be useful in building a clustered environment consisting of mobile agents in which a mobile device (cellphones, PDA's, etc.) submits a computation request to be performed on some local cluster accessible by the Internet. The uses for mobile cluster computing (MCC) can be determining a person's geographical location, mobile business operations (shopping via a cellphone), handling a distributed robot by some mobile device, observing weather information by means of mobile nodes and sending the data to the cluster for future weather prediction (like predicting tornadoes), etc. Issues like timeliness could be better solved by using Mobile IPv6.

This paper covers IPv6, its extension header mechanism, QoS, security, existing network transition, use of IPv6 for cluster computing and mobile cluster computing using IPv6 and its possible *nix [1] implementations.

---

[1]The next generation UNIX

1 Introduction

High performance distributed computing is always an interesting topic for researchers. During the past decade personal computers have become increasingly powerful, and the communication bandwidth between them is increasing day by day so researchers made a collection of interconnected computers working together as a single system formerly known as 'cluster.' A cluster provides similar (or sometimes better) performance and fault tolerance as the traditional mainframes or supercomputers.

The Internet is growing rapidly since almost 7 years, its continuous growth introduced many problems like IP address space shortage, IP mobility etc. A new protocol named 'IPng – IP next generation' was introduced in 1994 to solve problems in the existing Internet infrastructure. IPng is termed as 'IPv6 – IP version 6' in 1998 [3], IPv6 is expected to replace current IPv4 protocol in year 2005. IPv6 offers many new features like extension header mechanism, IP mobility, IP Anycast, IP route optimization, etc. It provides a very large IP address space up to 340,282,366,920,938,463,374,607,431,770,000,000 IP addresses [4].

Many research-oriented IPv6 backbones like 6BONE, 6TAP, and 6REN are fully operational. IPv6 protocol is designed in such a way that existing networks running IPv4 can natively migrate to IPv6 or they can use IPv6 without fully migrating to it. IPv6 simplifies the header format resulting in less bandwidth usage. IPv6 has embedded support for mobility, priority-based data handling (e.g. video streaming), and security.

According to our definition, mobile cluster computing (MCC) refers to a new paradigm,
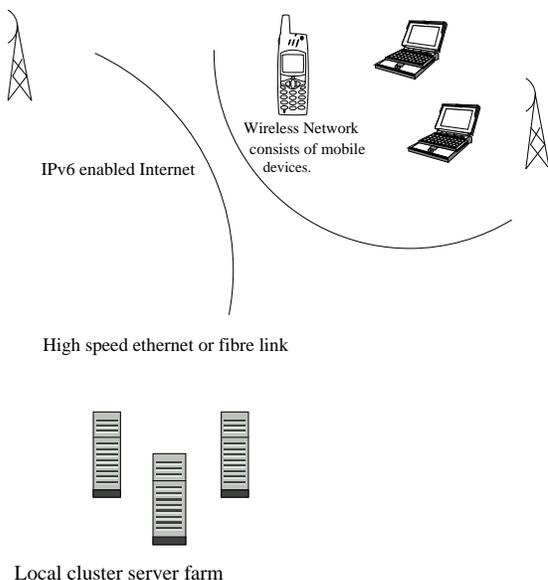

Figure 1: Mobile cluster network

in which mobile clusters or traditional clusters work together with a set of mobile nodes to carry out a specific task. Mobile cluster computing refers to an environment that offers flexibility in terms of mobility and extendibility, cluster security and a robust mobile network for handling geographically independent high performance computing requests.

IETF standardized mobile IP to handle Internet mobility. However, there exist two standards for mobile IP, they are referred as 'Mobile IPv4' and 'Mobile IPv6.' The Mobile IPv6 protocol offers many benefits over Mobile IPv4 protocol like providing 'Dynamic home agent discovery' mechanism, IP Anycast service, route optimization, etc [7, 6]. In this paper, we use Mobile IPv6 as a working protocol in our definition of 'mobile cluster computing.'

A mobile network typically consists of desktop PC's, wireless devices (such as cellphones, etc.), PDA's, server farms, cluster applications, etc. as shown in Figure 1.

1.1 Mobile and local clusters

A *mobile cluster* refers to a set of interconnected mobile nodes by means of wireless network while a *local cluster* refers to a set of interconnected workstations by means of high speed Ethernet or fibre link. Any cluster node is allowed to migrate from a mobile cluster to a local cluster or from a local cluster to a mobile cluster if this migration is allowed by a *S-node* or *Switching node*. A S-node is any node in a cluster that handles the operation of cluster transition.

Multiple S-nodes can exist either in local or mobile clusters. S-nodes maintain a security map. This security map defines how many nodes are allowed to join a particular cluster at some time. The security map can either be static or dynamic. Static security map is defined initially by the local cluster administrator or by the default policy. The nodes in the security map will grow dynamically if some node can be authenticated by a S-node properly. A S-node performs the network level authentication. A S-node can make use of the AH header of IPv6 packet for authenticating cluster nodes. S-nodes exchange their security map periodically. An IP Authentication Header is shown in Figure 2.

1.2 S-Node cluster authentication mechanism

The *IP Security* (*IPSec*) is designed to provide high quality cryptography-based security for IPv4 and IPv6. The IPv6 protocol has IPSec embedded in it and provides a separate feature different from IPv4. The objectives for using IPSec are to provide integrity, repudiation, confidentiality, encryption, etc. for IP and upper-level layers. Use of IPSec will minimize the need of security for different applications, such as ssh. IPSec provides two traffic security

protocols: AH and Encryption Security Payload (ESP). IPSec provides security services at the IP layer. It can select required security protocols, determine the algorithms used for the service, and choose cryptographic keys.

A S-node will act as a 'security gateway' for some particular cluster. A security gateway refers to an intermediate system that implements IPSec. S-node can use AH to provide authentication of the sender of data.

For example, a node, A, wants to join a cluster, C_1. Both the node A and the S-node for C_1 speak IPv6. Whenever the node A wants to be authenticated by C_1, it creates an IP extension header along with IPv6 packet and sends to the S-node. Upon receiving the packet that contains an IP AH header, the S-node determines the appropriate Security Association (SA) based on the destination IP address, security protocol in AH and the Security Parameters Index (SPI). A Security Association is a security object shared between two nodes, which contains the data mutually agreed upon for operation on the designated cryptographic algorithm (typically including a key).

The S-node will fetch the source IP address and match it in the specified security map. If this IP matches, the S-node will check authentication data, perform security checking and appropriately grant or deny a ticket to a cluster node according to the designated algorithm. Once the node gets a ticket, it can join that particular cluster. Each ticket has its own pre-defined lifetime.

2 S-node selection algorithm

The question arises that how to select an appropriate S-node when a cluster fires up and stars

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Next Header  | Payload Len |        RESERVED          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|           Security Parameters Index (SPI)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|              Sequence Number Field             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                                |
+          Authentication Data (variable)         |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
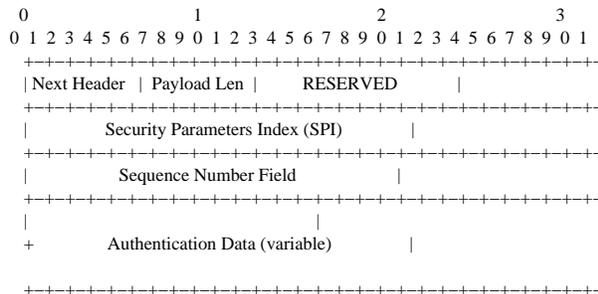
Figure 2: IP Authentication Header

working. To solve this problem, we propose
the following election mechanism.

1. The local cluster administrator specify a
   password/key in the security map if the
   default security policy is not employed.

2. Whenever a cluster is made operational,
   every node in this cluster computes a ran-
   dom number using the password in the
   security map and multicasts this random
   number to all nodes in this cluster.

3. Every node will receive the multicast from
   other nodes within $2n$ time where n is a
   random time to wait.

4. Every node compares each random num-
   ber it receive with the local random num-
   ber. If no random number from other
   nodes is larger than the local one, then this
   node will multicast a packet telling other
   nodes that it is selected as the S-node for
   a designated cluster.

5. The elected S-node can further delegate
   access control handling functions to other
   nodes. The receiving nodes will be syn-
   chronized with the security map of the pri-
   mary S-node. IP AH headers should be
   used to detect spoofed packets.

3 Dynamic clustering based on type of network traffic

Dynamic clustering refers to the fact that a
cluster can distinguish between different types
of network traffic. This behaviour is useful in
performing network-based load balancing for
different types of network data, such as per-
forming network level load balancing based on
traffic class.

3.1 Types of traffic

Traffic class is specified in the IPv6 header
to identify and distinguish between different
classes or priorities of IPv6 packets. In the
original specification IPv6 splits traffic into
two categories [2]:

1. *Congestion controlled traffic* may be ar-
   bitrarily delayed under conditions of con-
   gestion.

2. *Non-congestion controlled traffic* would
   be discarded under conditions of conges-
   tion

In our design we keep the notation defined in
the original specification until the new speci-
fication has been finalized. In that document
congestion controlled traffic can be further cat-
egorized as shown in the following table where
larger priority value indicates higher priority:

Pri.	Description	Example
0	Uncharacterised traffic	Custom use
1	Filler traffic	netnews
2	Unattended data traffic	email
3	Reserved	
4	Attended data traffic	FTP, NFS
5	Reserved	
6	Interactive traffic	telnet, X
7	Internet control traffic	routing, ICMP

Table 1: Congestion-controlled Traffic

Non-congestion controlled traffic is the type of traffic for which constant data rate and constant delay is required such as real-time audio and video transmission. The priority values 8 through 15 are used to specify this type of traffic.

3.2 Using flow labels to perform priority level traffic handling

Priority level traffic handling in IPv6 is based on flow label mechanism where flow refers to a sequence of packets transmitted by some transmission protocol from source to destination with some QoS requirements. It can enable a cluster to either handle different data on different clusters or be optimized for a specific type of data such as video streaming [8, 1]. Using flow labels, we can route different network traffic through a different route. Previously in IPv4, TOS (Type of service) field offers little control over network traffic.

IPv6 specifications state that every IPv6 header contains a 24-bit flow label field and a 4-bit traffic class field. A flow label is basically a unique identifier between 1 and FFFFFF when it is combined with the source IP address. Flow label 0 indicates that no flow label exists and this packet doesn't need any special treatment. Real-time network flow always has some flow label.
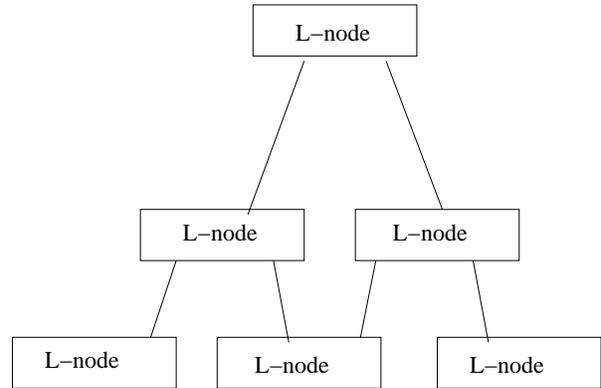

Figure 3: Hierarchal view of load balancing nodes

A router associates a flow with some state. If the router doesn't contain any state associated with flow, it may either drop the packet or forward it setting flow label to zero (0). All packets pertaining to the same flow must be originated from same source address, same destination address and same flow label.

A *L-node* refers to a *load balance node* or *load balancer* for a particular cluster. L-node will be responsible for flow control handling. There can be one or more L-nodes for a particular cluster. L-nodes can form a hierarchy to balance load from a set of cluster nodes as shown in Figure 3.

3.3 Communication mechanism between L-nodes

Every L-node upon receipt of some packet will forward the packet (workload) to its corresponding L-node or balance the traffic load if it is connected to some network (that is presence of a single L-node for balancing). L-node will balance the traffic either among different L-nodes or among different cluster-nodes according to following schemes:

- FCFS (First-Come First-Served) – The request that comes first is allocated to L-node first.

- RR (Round Robbin) – Each request gets a quantum time to run.

- Priority based – Service is based on their priority. The one with higher priority gets served first.

- IP Anycast – The L-node uses IP Anycast IPv6 service to distribute network requests to any nearby available node or cluster.

Possible failures can be minimized by suggesting that the network requests don't need to come through the top-level L-node, the network request can hit any L-node in L-node hierarchy and the request will be processed further down in hierarchy. So in the case if top level L-node fails, this will not cause the whole network to be down. In other case, if a child of a L-node fails, the request coming through parent L-node will be timed out. This won't affect a computing too much. The request can be queued when the parent detects a failure node. When the parent detect a timeout, the parent either carry that computation or forward to other L-nodes.

4 Benefits of IPv6 for roaming cluster nodes

A *roaming cluster node* usually refers to a mobile device participating in a cluster. A local node can become a mobile node at any time, but that should not affect local network connections. The transition of a local node to a mobile node should be transparent for cluster operations. They should keep processing without any interruption. One of the facts is that in IPv4 we don't have much global IP addresses left to assign to each mobile node. However, this problem is solved in IPv6 because we can uniquely identify the mobile device with at least one global IP from a very great IP address space provided by IPv6. Moreover, if we transit a local node to a mobile node in IPv4, that will definitely cause a service interruption because possible IP address change occurs when moving from one network subnet to another in IPv4. On the contrary, in IPv6, Mobile IP offers a way so that this node transition won't interrupt the cluster operation. Figure 4 shows basic communication between a mobile node and a cluster.

4.1 Mobile cluster

A mobile cluster is introduced as a set of mobile nodes that participate in a cluster computing. A mobile node can migrate to a local cluster and a local node can become mobile at any time.

We have to consider if we purely use mobile clusters. For example, one PDA sends request to another PDA for performing some task. How can IPv6 be useful in this scenario?

Our primitive observation is that due to simplified header mechanism in IPv6 (simplified headers mean that some fields in header are made optional or eliminated) it gives performance increase over IPv4. Since wireless media has a low bandwidth, this saves bandwidth. So IPv6 is more feasible than IPv4 for mobile only clustering.

4.2 Local to mobile cluster node migration mechanism

- A local node detects that it has moved by discovering a new default router.

- A local node becomes mobile node now, it performs a stateless or stateful address auto-configuration mechanism to obtain a new COA (Care of Address) for the new location. All packets destined to this COA will redirect to the mobile node through the current link. If the current link also serves some other cluster, then this node can decide to participate in both clusters (old and newly joined cluster).

- The mobile node then performs a binding update with the home agent of the old cluster.

- The home agent of the old cluster registers this binding and then sends back the binding acknowledgement.

- The home agent of the old cluster will now intercept the packets for migration by using 'proxy neighborhood discovery' protocol. *Proxy neighborhood discovery* means that the home agent multicasts a neighborhood advertisement onto the home link (cluster link) on behalf of the mobile node. The home agent itself also replies to neighbor solicitation on behalf of the mobile node. Each intercepted packet is then tunneled to the new COA address of the migrated node using IPv6 encapsulation.

- This triangular routing can be avoided by means of '*route optimization*' in which a mobile/migrated node can send binding update to any correspondent node. So the correspondent node can later send packet directly to the new COA of migrated node instead to the cluster home agent. But in

this way the cluster/newly migrated node can not participate in two clusters at one time. If the mobile node sends packet to any other node, it sends packet directly to its destination (not to home agent), it sets the source address of this packet to the COA, it also includes 'home address' destination option.

4.3 Stateless vs. stateful address auto-configuration for mobile nodes

In stateless address auto-configuration, no central server is configured for a host to obtain its interface address (COA in our case) and or configuration information such as DNS servers, gateways, etc. The stateless mechanism allows a host to generate its own address using a combination of locally available information advertised by routers. Routers advertise prefixes (like /64) that identify the subnet(s) associated with this link, whereas the host generates an identifier that uniquely identifies the host on some subnet (usually it uses MAC 48-bit address for first 48-bits). The final address is computed by combining the two. In the absence of router advertisements, the host will only generate link local address. With only link-local address [2], the host can only communicates with other hosts on the same link.

In stateful address auto-configuration, a host obtains interface address and or configuration parameters by contacting a central server (e.g. DHCPv6 server). The server maintains a database that keeps track of assigned COA (IP address).

---

[2] A link-local address refers to a non-routable address that is unique and valid only on the local network. The link-local address has a prefix of fe80::/64 and it is used for contacting hosts and routers on the same network only. The addresses are not visible or reachable from different subnets.
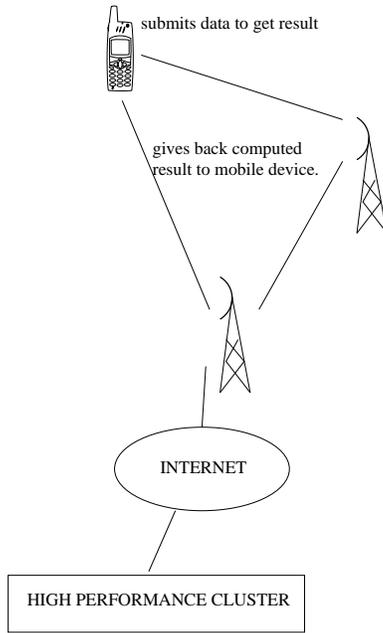
submits data to get result

gives back computed result to mobile device.

INTERNET

HIGH PERFORMANCE CLUSTER

Figure 4: Basic communication between a mobile node and a cluster

4.4 Availability of a computing node

Wireless communication is thought to be an unreliable one because of occasional omission and arbitrary failures caused by the intrinsic property of non-uniform wireless signal. IPv6 could alleviate this situation. For example, if a computing node is beyond the range of the cluster, this node is thought to leave the cluster in IPv4. But in IPv6 the node outside of the range of the cluster can be detected if it is adopted by the remote network via Mobile IP. This could guarantee the computing capability.

We can use the following scheme to seamlessly achieve a cluster computing task in respect to leaving and joining nodes:

1. If a node leaves the current cluster, the task assigned to it will be queued in the requesting agent.

2. The requesting agent continue to carry out

its computation in the cluster and wait for some pre-defined time for the leaving node to join the cluster through the Foreign Agent (FA).

3. If the leaving node is detected within the pre-defined time, the queued task will be carried out by the rejoining node. Otherwise, the task will be carried by other available nodes.

This scheme won't be so feasible in IPv4.

4.5 Timeliness Issues

In the previous section we discuss availability of mobile nodes. The situation will become complex when time constraint is imposed. Timeliness issues happen when mobile nodes within a computing cluster migrate from one subnet to another subnet in a wireless network and when time constraint is imposed [9]. To meet the QoS requirement timeliness issue needs to be considered.

In [9] it is proposed that the route optimization is delayed to a certain period and the message from the home network to a new foreign network is tunneled. Several approaches also presented in that paper. In IPv4, this problem is difficult to solve. In IPv6, we can combine their approaches and the scheme we propose for availability of a node to alleviate timeliness issue.

4.6 Mobile process migration

A *mobile process* refers to a process running in either a local or mobile node in some cluster. A mobile process differs from a local process in a way that it can either fully or partially mi-

grate from a node to another or from a node to multiple nodes.

- *Full process migration* refers to the fact that a process running on some node detaches itself from that node and injects itself in some other node.

- *Partial process migration* refers to the fact that a part of process migrates itself to some other node.

- Partial and full process migration can be bi-directional. That is from a mobile node to a local node or vice versa.

- In the case of full process migration, some level of transparency should be addressed so that the local process should not possess dangling references to the mobile process. The full migration of a mobile process may cause possible memory leaks because the process is not running inside the same machine. Hence, possible memory references should be updated. One of the proposed solution is to employ a *MM-node* (*Memory-manager node*). The purpose of this MM-node is to provide a *V-map* (*virtual map*) to map physical memory to distributed memory. Each mobile process takes references to some memory location provided by means of V-map.

- In the case of partial process migration, only a part of process or a thread of a process migrates. The parent process maintains the state for each migrated thread. If a migrated thread further spawns some thread, then this thread can not be migrated.

- Several threads of a mobile process can migrate to different nodes.

- One mobile process can only fully migrate to a node at a time.

- *P-node* (*Processing node*) refers to a node whose sole purpose is to handle incoming and outgoing processes for a particular cluster. A mobile process submits itself to P-node, and P-node will migrate this process to some other local or mobile cluster. Depending on system usage, P-node can either queue, suspend, or stop migration. P-node will migrate the mobile process to P-node of another cluster that will further migrate that process to any node or multiple nodes of its cluster.

- Using IPv6 protocol can protect some node in cluster to pretend as P-node.

- P-node or MM-node in simple words is like a router, that will multicast its address by means of IPv6 packets along with AH extension security headers to prevent spoofed packets.

5 Conclusion

We have discussed some basic issues of mobile cluster computing. We specify some basic concepts which are required in solving these problems. We further propose several solutions based on IPv6. S-nodes are designed to use AH and search the security map to authenticate a node to join a cluster. The flow label mechanism is used to handle priority level traffic. L-node is intended to balance the workload. Then we present how we could benefit from using IPv6. We build a top-down view from the cluster, to the node, and eventually to the process level. Timeliness issues are also discussed.

In this paper we identify problems and outline some design idea. Our future work is to refine our design and put it into implementation.

References

[1] Rahul Banerjee and Sumeshwar Paul Malhotra. A Modified Specification for Use of IPv6 Flow Label for Providing Efficient Quality of Service using Hybrid Approach. *IETF Internet Draft*, February 2002.

[2] S. E. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 1883*, December 1995.

[3] S. E. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460*, December 1998.

[4] R. Hinden and S. E. Deering. IP Version 6 Addressing Architecture. *RFC 2373*, July 1998.

[5] David B. Johnson and Charles Perkins. Mobility Support in IPv6. *IETF Internet Draft*, March 2002.

[6] Charles Perkins and David B. Johnson. Route Optimization in Mobile IP. *Cluster Computing*, 1(2):161–176, 1998.

[7] Charles Perkins and David B. Johnson. Route Optimization in Mobile IP. *IETF Internet Draft*, September 2001.

[8] J. Rajahalme, A. Conta, B. Carpenter, and S. Deering. IPv6 Flow Label Specification. *IETF Internet Draft*, March 2002.

[9] Haihong Zheng, Rajkumar Buyya, and Sourav Bhattacharya. Mobile Cluster Computing and Timeliness Issues. *Informatica*, 23(1):5–17, 1999.