A HYBRID TOPOLOGY ARCHITECTURE FOR P2P FILE SHARING SYSTEMS

J. P. Muñoz-Gea, J. Malgosa-Sanahuja, P. Manzanares-Lopez,
J. C. Sanchez-Aarnoutse, A. M. Guirado-Puerta
Department of Information Technologies and Communications,
Polytechnic University of Cartagena,
Campus Muralla del Mar, 30202, Cartagena, Spain

Keywords: Peer-to-peer, structured networks, unstructured networks, application layer multicast.

Abstract: Over the Internet today, there has been much interest in emerging Peer-to-Peer (P2P) networks because they provide a good substrate for creating data sharing, content distribution, and application layer multicast applications. There are two classes of P2P overlay networks: structured and unstructured. Structured networks can efficiently locate items, but the searching process is not user friendly. Conversely, unstructured networks have efficient mechanisms to search for a content, but the lookup process does not take advantage of the distributed system nature. In this paper, we propose a hybrid structured and unstructured topology in order to take advantages of both kind of networks. In addition, our proposal guarantees that if a content is at any place in the network, it will be reachable with probability one. Simulation results show that the behaviour of the network is stable and that the network distributes the contents efficiently to avoid network congestion.

1 INTRODUCTION

The main characteristic of an overlay network is that all the computer terminals that shape it are organized defining a new network structure overlayed to the existent one. They are purely distributed systems, and can be used in a lot of interesting fields: for example, to transmit multicast traffic in a unicast network (like Internet), technique known as Application Layer Multicast (ALM). However, the most popular overlay networks are peer-to-peer (P2P) networks, commonly used to efficiently download large amounts of information. In this last scenario there are two types of P2P overlay networks: structured and unstructured.

The technical meaning of structured is that the P2P overlay network topology is tightly controlled. Such structured P2P systems have a property that consistently assigns uniform random *NodeIDs* to the set of peers into a large space of identifiers. With this identifier, the overlay network places the terminal in a specific position into a graph. On the other hand, in unstructured P2P networks the terminals are located in the overlay network by one (or several) *rendez-vous* terminals with network management functions.

Although unstructured P2P networks require the presence of one controller (*rendez-vous*) at least, they have the advantage that the information searching process supports complex queries (it is a similar methodology to that used to search for information in Google and supports keyword and phrases searching). That does not happen when the P2P network is structured. In this case the advantages are that it enables efficient discovery of data items and it doesn't require any central controller. In addition, it is also much easier to reorganize when changes occur (registering and leaving terminals) and, consequently, the overlay network is more scalable and robust. Section 2 describes in depth the searching and location process of structured and unstructured networks.

In this work we try to design a file-sharing system that shares the advantages of both types of P2P networks. The users locate the contents in an unstructured way. If this search fails, the system will use an application layer multicast service (given by a structured P2P network), to locate the terminal that owns the searched information. It is necessary to remark that, with the system proposed in this work, the location of any existing content always success.

There are several proposals that try to support sophisticated search requirements, like (Garcés-Erice et al., 2003)(Mislove and Druschel, 2004)(Castro et al., 2002a). These proposals organize P2P overlays into a hierarchy, and they have a high degree of complexity.

Muñoz-Gea J., Malgosa-Sanahuja J., Manzanares-Lopez P., Sanchez-Aarnoutse J. and Guirado-Puerta A. (2006).
A HYBRID TOPOLOGY ARCHITECTURE FOR P2P FILE SHARING SYSTEMS.
In *Proceedings of the First International Conference on Software and Data Technologies*, pages 319-324
DOI: 10.5220/0001315303190324

The remainder of the paper is organized as follows: Section 2 describes the main characteristics of both, unstructured and structured networks. Section 3 describes the system proposed in this paper in detail. Section 4 summarizes the more relevant contributions of the proposed solution. Section 5 shows the simulation results and finally, Section 6 concludes the paper.

2 P2P OVERLAY NETWORKS

The topology in a P2P structured overlay network is algorithmly fixed. Both, the nodes and the contents, have assigned an identifier (*NodeID* and *Key* respectively) belonging to the same scope. These P2P systems use a *hash* function applied to a MAC or IP terminal address and to the data content respectively, to generate these identifiers. The overlay network organizes its peers into a graph that maps each data *Key* to a peer, so that content is placed not at random peers but at specified locations. This structured graph enables efficient discovery of data items using the given *Keys*: a lookup algorithm is defined and it is responsible for locating the content, knowing its identifier only. However, in its simple form, this class of systems does not support complex queries. They only support exact-match lookups: one needs to know the exact *Key* of a data item to locate the node(s) responsible for storing that item. In practice, however, P2P users often have only partial information for identifying these items and tend to submit broad queries (e.g., all the articles written by "John Smith") (Garcés-Erice et al., 2004). Some examples of P2P structured networks are: CAN, Chord, Tapestry, Kademlia and Viceroy (Stoica et al., 2003), (Zhao et al., 2004), (Maymounkov and Mazières, 2002).

Unstructured P2P networks are composed of nodes that are linked to the network without any previous knowledge of the topology. The terminals need to know beforehand the location of a central controller, also denoted *rendez-vous* point, responsible for including them within the overlay network and for storing their contents list. The overlay networks organize peers in a random graph in a flat or hierarchical manner (e.g., Super-Peers layer). The search requests are sent to the *rendez-vous* node, and this evaluates the query locally on its own content, and supports complex queries. If the content is not located in the *rendez-vous*, most of the available networks use flooding or random walks or expanding-ring Time-To-Live (TTL) search on the graph to query content stored by overlay peers. This is inefficient because queries for content that are not widely replicated must be sent to a large fraction of peers, and there is no coupling between topology and data items'location (Lua et al., 2005). Some examples of P2P unstructured networks are: Gnutella, FastTrack/Kazaa, BitTorrent and eDonkey 2000 (Lua et al., 2005).

In sum, for a human being, the searching process is easier in an unstructured network, since this is made using patterns of very high level (like in Google, for example). Nevertheless, there exists much inefficiency in the location process of the content. In structured networks, exactly the opposite happens: the location is quasi-immediate, but the searching process is more tedious.

3 DESCRIPTION OF THE SYSTEM

Our proposal tries to define a hybrid system. Therefore, the user can search contents using more or less general parameters and later choosing among the elements that satisfy the searching criterion that content which he wishes to download, like in unstructured networks. Nevertheless, the network will be organized in a structured way, which will facilitate the location of the contents.

All the nodes are immersed in a structured overlay network (anyone of the previously mentioned types). In addition, the nodes divide automatically into different sub-groups, in a more or less uniform way, surrounding a *rendez-vous* node. This node has the best peformances in terms of CPU, bandwidth and reliability (see Section 3.2). When searching for a content, the user will send the search parameters to its *rendez-vous*, and this will return information about who has the contents in this sub-group.

All the *rendez-vous* nodes of the network are going to be members of a multicast group defined within the same structured network. This way, if the search fails, the *rendez-vous* node will send the request to the rest of *rendez-vous* nodes in a multicast way. Fig. 1 describes the general architecture of the system.

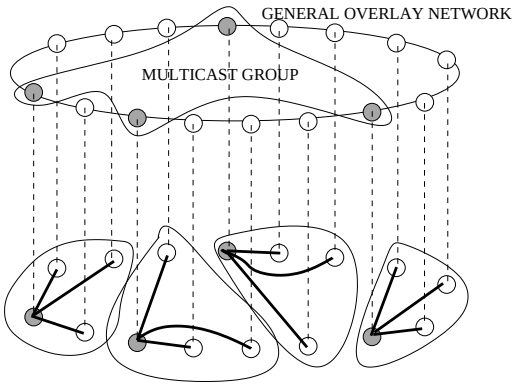

Figure 1: General architecture of the system.

3.1 Obtaining the Identifiers and Joining the General Network

Every node needs to obtain a *NodeID*. In this work, this identifier is obtained applying a hash function (MD5 or SHA-1) to its MAC or IP address. In the same way each node also needs to obtain a *SubgroupID* that identifies the sub-group to which the node is going to belong. We propose to use a previously well-known server to obtain this identifier. Each sub-group will have a maximum number of nodes, and the nodes will be assigned by order to each one of the sub-groups until completing their maximum capacity. When the existing sub-groups are completed new sub-groups will be created.

As is usual in any structured network a node needs to know at least one address of another node in the overlay network. The previous server can also provide this information. Finally, the node will have to link to the P2P overlay network, using the mechanism imposed by the structured network.

3.2 Joining the Sub-group

Each node of the sub-group will be able to establish a TCP connection with its *rendez-vous* node, and they will send their content list to it. Each sub-group is identified by a *SubgroupID*. Initially, a node looks for its *rendez-vous*. To do this, it uses the structured network to locate the node which *NodeID* fits with its *SubgroupID*. This node knows the IP address of the *rendez-vous* node of its sub-group. The last step consists of transmitting this information to the requester node. Note that in this way the system builds an unstructured network by using an underground structured network. In addition, this last property allows us to define the *rendez-vous* nodes dynamically and to guarantee the stability of the network throughout time.

3.3 Management of the Hierarchy

When the new node finds its *rendez-vous*, it notifies its resources of bandwidth and CPU. The *rendez-vous* nodes control the nodes that are linked to their subgroup and they form an ordered list of future *rendez-vous* candidates: the longer a node remains connected (and the better resources it has), the better candidate it becomes. This list is transmitted to all the members of the sub-group, and when the *rendez-vous* fails, the first node in the list becomes its successor. Later, it must inform all sub-group members that this node is now the new *rendez-vous*. Also, it must to modify this information in the node which *NodeID* fits with its *SubgroupID*.

3.4 Management of the *Rendez-Vous* Nodes

All the *rendez-vous* nodes are members of a multicast group defined at application level. When a node becomes *rendez-vous*, it must be linked to this multicast group, in order to spread the unsuccessful searches to the rest of sub-groups. Structured P2P networks can be used to implement an application layer multicast service, for example CAN-Multicast (Ratsanamy et al., 2001), Chord-Multicast (El-Ansary et al., 2003) and Scribe (Castro et al., 2002b). Each one uses a different P2P overlay and it can implement the multicast service using flooding (CAN-Multicast, Chord-Multicast) or the construction of a tree (Scribe). Anyone of the previous methods provides an efficient mechanism to identify and to send messages to all the members of a group.

Our proposal uses Chord-Multicast. It is not necessary that the multicast process reaches all the group members before sending the searching results to the requester node. When one node responds affirmatively to a request it sends to the requester's *rendez-vous* the coincidences of the search in its database. Next, this *rendez-vous* gives back immediately the IP address and the corresponding metadata to the requester node. Therefore, the requester node obtains the searching results as soon as possible.

3.5 Registering the Shared Files

In a similar way to KaZaA, when a node establishes connection with its *rendez-vous* it sends the metadata of those files that it wants to share. This allows the *rendez-vous* to maintain a data base including the identifiers of the files that all the nodes of the subgroup are sharing and the corresponding IP address of the node that contains them. The information sent by the node includes the name of the file, its size and its description.

3.6 Search

When a user wishes to make the search of certain content, his node sends a request on the TCP connection established with its *rendez-vous*. For each coincidence of the search in the data base, the *rendez-vous* gives back the IP address and the corresponding metadata.

If the search fails, the user has the possibility of asking for to its *rendez-vous* node that tries to contact with other *rendez-vous*. The identification of those nodes is simple, since all belong to the same application layer multicast group.

4 ADVANTAGES OF THE SYSTEM

Next we are going to describe some of the contributions of the system proposed in this work. First, it is necessary to emphasize that all the nodes are assigned to a sub-group and not to a server. The nodes are able to automatically find the *rendez-vous* responsible for their sub-group.

It is also necessary to emphasize that this system is able to manage the heterogeneity of the network too. The most stable nodes and those with better benefits will become *rendez-vous* nodes, which will increase the network performances.

On the other hand, the application layer multicast service provides an effective way to share information among *rendez-vous* nodes. In this way the maintenance of multicast group is practically made in an automatic mode. In addition, this guarantees that any content in the network can be located by any user.

Finally, the searches will be made in a simple way, similar to those made in current unstructured file-sharing applications.

5 SIMULATIONS

One of the advantages of this system, commented previously, is that any content present in the network could be located by any user. Nevertheless, the searches of contents present in the same sub-group will be faster and more efficient than when the searches need to use other *rendez-vous* nodes.

There are several interesting parameters that is necessary to quantify. First, the probability that the requested content is registered in the *rendez-vous* of the node's sub-group. Second, the evolution of the previous parameter throughout time. Since the users are making successive searches of contents in other sub-groups of the network, these automatically will be registered in their own *rendez-vous* node, increasing the value of this probability. Finally, it is also interesting to find out the average number of *rendez-vous* nodes that will be consulted in order to locate a content.

In order to quantify the previous parameters a simulator in C language has been programmed. The contents are classified in three classes based on the degree of interest that they can motivate in the users ("very interesting", "interesting" and "of little interest"). At the beginning, the available contents are distributed in a random way among all the nodes of the network. As has been mentioned before, the *rendez-vous* share information using a Chord-Multicast procedure.

The simulation results show the probability that a content is located in the same sub-group as the requester node, as well as the average and maximum number of *rendez-vous* consulted until content location. All these results are obtained based on the number of simulation iterations. In each one, all the nodes of the network ask for a content that they do not have.

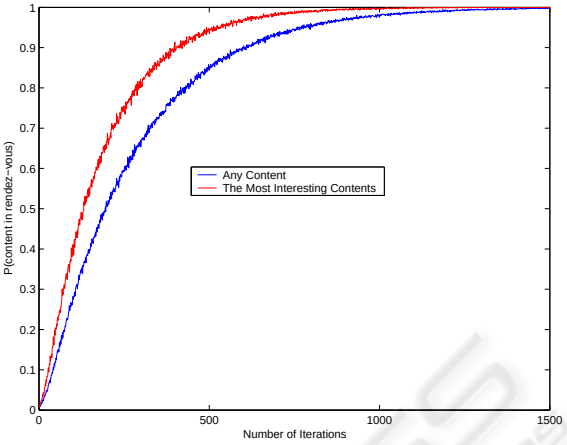

Figure 2: Probability that a content is located in the same sub-group as the requester node.

Figures 2, 3 and 4 present the simulation results corresponding to a network with 12,800 different contents and 6,400 nodes, with 128 *rendez-vous* nodes.

Fig. 2 shows the probability that the content is in the same requester's sub-group, for both the most interesting contents and for any content. It is observed that this probability grows as the number of iterations increases, but converging to a value of one, which assures that our system is stable. This also indicates that our architecture assures that, in a few steps, the contents will be equally distributed among all the sub-groups. It is also possible to observe that in the transitory, the probability of finding an interestig content in the *rendez-vous* increases more quickly than the probability of finding any content.

Fig. 3 shows the average number of *rendez-vous* consulted to find a content. It is observed that the number of consulted *rendez-vous* quickly decreases, and when the number of iterations reaches 500 this value converges to one, which indicates that the content is in the same sub-group as the requester node. This shows us that the load coming from other sub-groups is minimal. It is also observed that this parameter decreases more quickly in the case of the more interesting contents than in the case of other contents.

Finally, Fig. 4 shows, in linear scale, the maximum number of *rendez-vous* consulted to locate any content. This parameter oscillates a lot in the initial transitory, but when it finishes it converges to values near the unit, agreeing practically with the average number.

Next, we are going to check the effect that both the

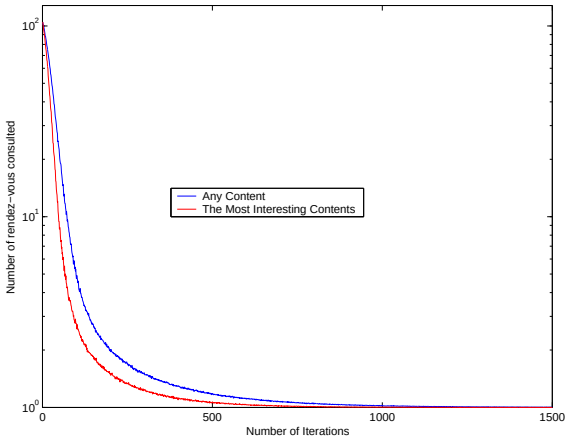

Figure 3: Average number of *rendez-vous.* consulted until content location (in semilogarithmic scale).

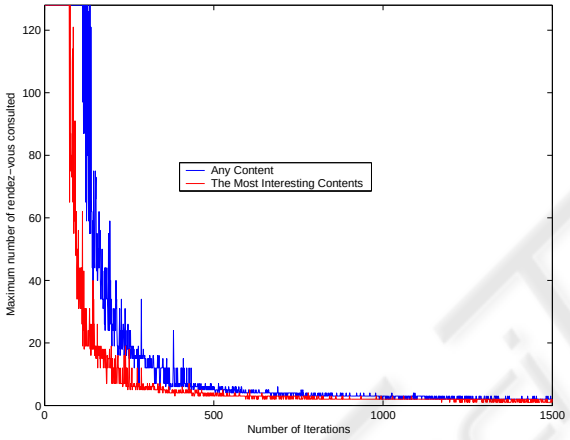

Figure 4: Maximum number of *rendez-vous.* consulted until content location (in linear scale).

number of contents and the number of *rendez-vous* nodes have on the probability that a content is located in the same requester's sub-group. Figure 5 shows the previous probability but with 6,400 and 19,200 contents. It can be observed that when the number of contents in the network diminishes the probability of finding it in the same requester's sub-group increases more quickly. On the other hand, when the number of contents in the network increases, a greater number of iterations is needed for the previous probability to reach the value of one.

Besides, Figure 6 shows the previous probability in a similiar network but with 64 and 256 *rendez-vous* nodes. It can be observed that the effect of the number of *rendez-vous* on this probability is quite similar to the effect of the number of contents. When the number of *rendez-vous* diminishes, the probability of finding a content in the same requester's sub-group

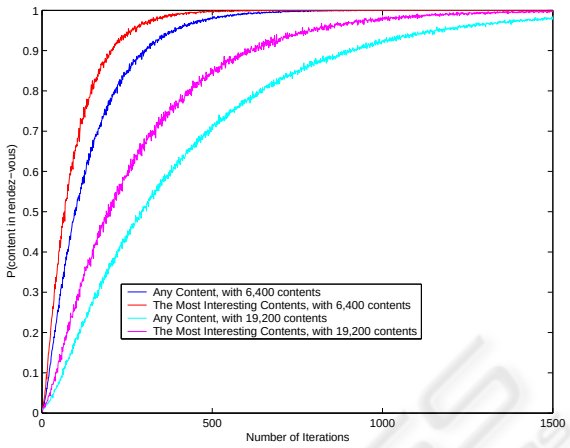

Figure 5: Probability that a content is located in the same sub-group as the requester node, with 6,400 and 19,200 contents.

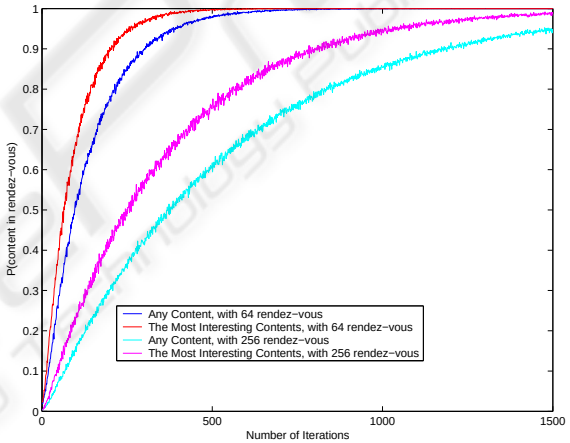

Figure 6: Probability that a content is located in the same sub-group as the requester node, with 64 and 256 *rendez-vous*.

increases more quickly. On the other hand, when the number of *rendez-vous* increases, a greater number of iterations is needed to obtain a probability close to one.

Next, we are going to compare the presented approach with the existing ones. In (Garcés-Erice et al., 2003), peers are organized into groups, and each group has its autonomous intra-group structured overlay network and lookup service. Groups are organized in a top-level structured overlay network. To find a peer that it is responsible for a key, the top-level overlay first determines the group responsible for the key; the responsible group then uses its intra-group overlay to determine the specific peer that is responsible for the key. However, due to the use of structured networks this system does not support complex queries.

The main advantage of this system is to reduce the expected number of hops that are required for a lookup, but in any case this is bigger than in our system, since in steady state only one hop is required.

In (Mislove and Druschel, 2004), they call an instance of a structured overlay as a *organizational ring*. A *multi-ring* protocol stitches together the *organizational rings* and implements a *global ring*. Each ring has a globally unique *ringID*, which is known by all the members of the ring. Every search message carries, in addition to a target key, the *ringID* in which the key is stored. Then, the node forwards the message in the global ring to the group that corresponds to the desired *ringId*. When a key is inserted into a organizational ring, it is necessary that a special indirection record is inserted into the global ring that associates the key with the *ringID* of the organizational ring where key is stored. However, the expected number of hops that are required for a lookup is similar to the previous work.

Finally, in (Castro et al., 2002a), it is proposed the use of a universal ring, but it provides only bootstrap functionality while each service runs in a separate P2P overlay. The universal ring provides: an indexing service that enables users to find services of interest, a multicast service used to distributed software updates, a persistent store and distribution network that allows users to obtain the code needed to participate in a service's overlay and a service to provide users with a contact node to join a service overlay.

6 CONCLUSIONS

This paper presents a hybrid P2P overlay network that makes easier for the user both the searching process and the content location. The simulation results show that in this type of networks the contents are distributed in a way that minimizes the overload on the *rendez-vous* nodes.

We have also verified that an increase of both the number of *rendez-vous* and of contents increases the number of necessary iterations to guarantee that the content is located in the same requester's sub-group.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Researh Council under project TEC2005-08068-C04-01/TCM and with funds of DG Technological Innovation and Information Society of Industry and Environment Council of the Regional Government of Murcia and with funds ERDF of the European Union.

REFERENCES

Castro, M., Druschel, P., Kermarrec, A.-M., and Rowstron, A. (2002a). One ring to rule them all: Service discovery and binding in structured peer-to-peer overlay networks. In *Proceedings of the SIGOPS European Workshop*, Saint-Emilion, France.

Castro, M., Druschel, P., Kermarrec, A.-M., and Rowstron, A. (2002b). Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications (JSAC)*, 20(8):100–110.

El-Ansary, S., Alima, L. O., Brand, P., and Haridi, S. (2003). Efficient broadcast in structured p2p networks. In *Proceedings of the Second International WorkShop on Peer-to-Peer Systems (IPTPS03)*, pages 304–314, Berkeley, CA, USA.

Garcés-Erice, L., Biersack, E. W., Ross, K. W., Felber, P. A., and Urvoy-Keller, G. (2003). Hierarchical peer-to-peeer systems. In *Proceedings of ACM/IFIP International Confeence on Parallel and Distributed Computing (Euro-Par)*, Klagenfurt, Austria.

Garcés-Erice, L., Felber, P. A., Biersack, E. W., Urvoy-Keller, G., and Ross, K. W. (2004). Data indexing in peer-to-peer dht networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems*, pages 200–208, Tokyo, Japan.

Lua, K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. (2005). A survey and comparison of peer-to-peer overlay networks schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93.

Maymounkov, P. and Mazières, D. (2002). Kademlia: A peer-to-peer information system based on the xor metric. In *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS02)*, pages 53–65, Cambridge, MA, USA.

Mislove, A. and Druschel, P. (2004). Providing administrative control and autonomy in structured peer-to-peer overlays. In *Proceedings of the Third International Workshop on Peer-to-Peer Systems (IPTPS04)*, pages 162–172, San Diego, CA, USA.

Ratsanamy, S., Handley, M., Karp, R., and Shenker, S. (2001). Application-level multicast using content-addressable networks. In *Proceedings of the Third International Workshop on Networked Group Communication*, pages 14–29, London, UK.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., and Balakirshnan, H. (2003). Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32.

Zhao, B. Y., Huang, L., Stribling, J., Rhea, S. C., Joseph, A. D., and Kubiatowicz, J. D. (2004). Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications (JSAC)*, 22(1):41–53.