Chapter 9

Conclusion

This dissertation describes the current status of planning for open architectures. It presents the most critical planning problems that ONA faces and describes our planning approach, the design and the implementation of a planning system, and the planning system performance results. Implementation of the planning algorithm presented here was done in the context of active networks, and performance tests show that the system is feasible and suitable for real-time applications.

9.1 Contributions

Because ONA technologies are complex, many applications will not be coded to take advantage of ONA capabilities. The data streams sent by such applications can gain the benefits of ONA technologies provided an automated system can determine the proper choice and placement of ONA adaptations. This dissertation has demonstrated that it is possible to build an automated planning system that is quick, effective, and extensible. The planning procedure implemented in this work collects the planning data, executes the planning algorithm to calculate a plan, and deploys the plan.

The planning procedure uses a planning algorithm which is based on a heuristic search in the space of all possible plans. We have shown that our planner implementation outperforms simple alternatives such as an exhaustive search of all possible adaptation deployments or treating the problems of each link separately. Depending on the complexity of the problem, our heuristic planning method can be thousands of times faster than exhaustive search, and can handle realistic cases with problem complexities too large for exhaustive search to be used at all. It took at most, 160 milliseconds to calculate a plan in the tested cases using our planning algorithm. Using our working units as a figure of merit, our centralized planning system produced plans that were as much as 100% better than unsophisticated incremental planning. The heuristic planning algorithm produced optimal plans (as evaluated using heuristic search) in at least 99% of tested cases.

We implemented a planner design for active networks. Panda is an example of an active network system that provides an adaptation service for end-to-end connections and automated planning for the selection and deployment of adaptations. The planning process presumes that a search for a feasible plan in the space of all possible plans will succeed. The complexity of the search depends on the scale of the space of possible plans. We believe that in a practical system the plan space is very large, making automated planning a complex artificial intelligence problem. Fast but inefficient incremental planning and slower but more efficient centralized planning were combined in the planning procedure implemented here. This design allowed the data transfer to start up to 400 milliseconds faster. Thus, short sessions (less than a hundred packets) may not require central planning because some tens of packets will be served by the incremental plan before a central plan is activated. Some factors can make the life of the incremental plan even longer -- for example, a slow or faulty planner or adapter storage site, a slow network, etc.

The implemented planning procedure supports some functions of network resource management. The planning system adds and rejects new connections, and preempts and replans old connections according to priorities of the connections and the network resource availability. The resource management component of the planning procedure can reduce the chaos in the active network resource distribution. The planner first learns what resources are necessary to support a connection and where they are needed. The planning resource management receives resource data from the planner and uses a RSVP-like service (see [Braden01]) or automated configuration network management (see [Konstantinou02]) for resource reservation before a connection is established.

We also presented the design of the secure planning procedure. The security of the planning procedure is based on the public key cryptography authentication of signaling between all participants of the planning procedure: connection nodes, a planner, and adapter storage sites. The planning procedure should not be a source of additional computer insecurity.

We demonstrated that the planning system can significantly improve the performance of real-world applications using metrics meaningful to their users. In the tests reported here, our automated planner doubled the peak signal-to-noise ratio of a video stream at certain points in the transmission compared to standard networks or unsophisticated planning, and (after a ramp-up period) always provided better PSNR than standard networks. Planning was also applied to other application data streams. We used encryption for the RAT application that transmits audio data. The quality of voice and music audio streams was downgraded but remained recognizable. The planning procedure can be applied to any kind of data streams given that the planner knows their characteristics and requirements. Encryption increases the delay of the data stream, thereby making QoS worse. In some cases the overhead added by encryption of a data stream should be compensated for by latency-reducing techniques (for example, compression, etc.)

The measurements presented in this dissertation were made in three dimensions: applications with different data generation intensity, computers with different CPU power running Panda nodes, and network links with different bandwidth and security levels. The planning procedure chooses a plan faster in the cases of less intensive user applications and more powerful machines running Panda nodes. The planning procedure completes planning 3 to 4 times faster if adapters are predeployed on the connection nodes. It takes longer to deploy larger adapters, and their deployment may suffer if network conditions are too poor. The larger number of adapters used, the more delay in deployment. It makes optimization of the plan (e.g., redundant adapter dropping) absolutely necessary.

In the bigger picture, this work contributes to the creation of responsive middleware that rearranges its internal architecture to serve a user application. The results of this work extend the area of automated responsiveness of middleware, making a wider range of network services available to an average user.

9.2 Discussion

We defined the structure of adapter data that consists of five major components: problem ID, solution method, effect, cost, and preconditions and postconditions. The list of preconditions and postconditions can very drastically from adapter to adapter. If an adapter defines fewer preconditions than defined by stream characteristics and previously used adapters, that adapter is not sensitive to particular conditions. However, a particular adapter can require preconditions that are not defined by stream characteristics or by other adapters' postconditions. This kind of an adapter pre-/post-condition inconsistency is indefiniteness, but not necessarily an adapter conflict. Whenever possibble, planner should try to avoid whenever possible the use of such adapters because the planner cannot verify adapter consistency and data semantics correctness.

The design of adapters should remain as simple as possible to facilitate planning, because combining complicated adapters makes the planning process complex, and adapter consistency and data semantics correctness becomes problematic. In our model, simple adapters form an adaptation layer that directly handles user data and can be optimized by the planner specifically for a particular connection.

Another problem is the heterogeneity of planning data. Various nodes may support an uneven number of link and node resource attributes. For example, some links may lack bandwidth or security information; some nodes may lack their execution capability information, etc. The planner can assume that these links and nodes have a particular (average or minimal) amount of resources specific to the network where a connection occurs. If the planner assumes wrong, a connection QoS can be insufficient to support application requirements, or the calculated plan may not be optimal or feasible. The planner should have access to the network statistics to be able to make correct assumptions with higher probability.

The centralized planning paradigm requires a certain level of network stability. As our experiments show, centralized replanning can require some seconds. If the state of the network is stable for less than the time necessary for replanning, the centralized plan becomes obsolete before it can be used, or it can be used only for a very short time. In this case, incremental planning must be used. Such an incremental plan is never decommissioned before the connection terminates. Whenever a change occurs on one of the connection links, a local replanning process occurs, and a replanning message is sent to the source node, which switches back to the incremental plan and starts centralized replanning. The replanning can be started with the full-scale planning data selection or, for acceleration of the process, just the problematic link change can be sent to a planner that keeps the planning data for the life of the connection. This approach can save 120 to 150 milliseconds in the replanning process, but the centralized planner must keep the planning data for all connections it serves. Other techniques can be used to accelerate the planning process in unstable networks. For example, it can be assumed that the number of states of network instability during a connection is limited. Then all central plans that were calculated during the connection are not decommissioned until the end of the connection. If network conditions return to the state that has a correspondent centralized plan calculated and deployed, the source node reswitches to that plan instead of performing full-scale replanning. Another potential way to accelerate the replanning process is to choose only predeployed adapters for newly calculated plans. Massive reuse of previously used plans and predeployed adapters will be a major source of acceleration in the planning process.

In the future the service of adapter and partial order plan distribution can grow to the same scale the a secret or public key distribution services existing today. The services will be available on-line for subscription and use. The access to multiple distantly deployed services will require a much better quality of networks and accounting facilities, but user application requirements grow very fast also.

The planning procedure presented in this work is applicable not only for open network architectures, but also for other environments. For example, real-time rescue/military software systems use ad hoc networks that support highly prioritized and diverse traffic that carries video and audio data to diverse devices, often in wireless-communication-hostile terrain. The planned adaptation and rerouting will improve QoS of these networks. Peer-to-peer systems tend to use a distributed infrastructure for their services. These services provide adaptations, information and rerouting to peer-to-peer connections. The number of these services grows with every year, and these services participate in more and more complex interaction. In the peer-to-peer community, composability and coordination of these services require planning procedures that will make the combined use of the services feasible and efficient [Venkatasubramanian02].

This makes the ONA planning approach presented here desirable in peer-to-peer communications. It is an example of how the technology of open network architectures penetrates into conventional networks. Remote code invocation systems such as Rover, RMI, and Java Beans that are able to run code remotely can benefit from this planning approach if the code that they run on other hosts is in the form of adapters similar to those that are described in this dissertation. These systems will make their choice of what adapter to use and where to use it by applying the planning procedure presented here. This approach is especially valuable if the adapters that are chosen by these systems are automatically selected and ordered, and the large scale of a system makes manual planning problematic.

The use of open network architectures (especially active networks) is very complex. Wide use of this technology will bring to life other problems that are barely visible at this point, but that may become an issue in the future. Thus, this work suggests that poor connection link resources can be compensated for by execution resources of connection nodes through adaptation of the data stream. The node resources are considered "free" and are limited by the node resource limits. In the future, however, link and node resources may become equally valuable commodities, and planning will have to balance the use of these resources. For example, distributed data search and processing (e.g., OLAP) make node resources and the node workload more important factors of planning than link resources of the network. Adapters are customized mobile tools that should be deployed on connection nodes to search and process data, especially if the client and servers are not capable of doing heavy data processing. In addition to CPU, memory and hard drive resources, monetary cost and battery capacity (for wireless networks) have also become important factors of planning.

The two orthogonal methods for of planning – adaptation and rerouting – have their advantages and drawbacks. Rerouting planning can find the best path from a source to a destination, but this path may not be sufficient for a particular user application that requires at least a particular lever of QoS. The multi-factor routing tables [Choi99] of networks may not be flexible and efficient enough to serve all possible application requirements. The routing tables will be under permanent reconstruction in varying network conditions, therefore relying on them is problematic. Additionally, if all users try to use the "resource-richest" parts of networks, it will cause an overloading of those areas. Thus, it may be beneficial to avoid the resource-richest part of the network as well as the resource-poorest part. Adaptation planning fits an application data stream into a network connection, but in some cases it can achieve much higher QoS using resource-richer networks. Thus, it can be beneficial to use, in conjunction, both adaptation and rerouting planning. Using rerouting planning, we can get a path from a source to a destination with the most critical properties improved, then apply adaptation planning as presented in this dissertation to fit the data stream into the path. The tradeoff between richer networks and adaptation will provide more flexibility to the planning process and more space for massive use of planning in networks.

Open network architectures can make possible another way of planning possible. Instead of planning on behalf of connections (adaptation and rerouting), global planning on behalf of networks is possible. Systems that are capable of automated network configuration management began to appear rather recently (see Nestor [Konstantinou02]). This kind of planning will study major trends and interests of users' behavior with respect to geography, time, a particular event, etc., and modify the network conditions accordingly. For example, global planning can wisely locate multiple caches before a sporting event to facilitate user access during the event, or it can fairly reroute all traffic through alternative paths to avoid a massive workload on particular parts of networks. The global planning will be responsible for the maintenance and effective use of the whole-planet Internet with all potential positive and negative consequences of globalization. The necessity for global network planning is not strongly felt yet, but the addition of hundreds of millions of new users into the Internet traffic may turn this situation toward wider Internet regulation. Both planning approaches, network- and connection-oriented must cooperate to achieve the best results. For example, global planning will be defining a space where particular user applications will operate and within which the connection-based adaptation and routing planning described in this thesis will take place.

The planning mechanism presented in this work can also be used by adaptation-aware applications that are specifically designed to benefit from open network architectures and active networks. In these applications, we can expect that the user or application designer may intrude more into the planning process than in legacy applications. The user may suggest not only what methods are more preferable for the solution of the problem, but also may suggest that specific adapters, services, and planning algorithms be applied to their data. This flexibility of the planning procedure will bring more constraints into the planning procedure's resource distribution, security, fault-tolerance, and accounting.

The distributed nature of the planning mechanism described in this work adds more dimensions to the complexity of planning. Thus, resource distribution and security may require involvement of other services that observe network resources, security of networks, trustworthiness of network nodes and services (i.e., adapter storage sites, etc.), and authentication mechanisms. These services may become bottlenecks for the multiple network connections, and their use may require extra coordination and preparation for the connection establishment, sometimes even before the moment that the connection occurs. Thus, the planning process will contain a number of layers where adaptation planning will be only one of many necessary planning procedures.

Planning has been largely unexplored in the area of open network architectures. This dissertation provides a sizable first step in designing planners for the next generation of open network architectures.