Chapter 4

Planning Protocol

In this chapter we will present the planning protocol for ONA. The planning protocol is suitable for a wide range of ONA connections, but its implementation may depend on special requirements of actual network environments, applications, etc. The planning protocol presented here was developed for Panda active networks application. We will present the requirements of the planning protocol and show how these requirements were met in the current planning implementation.

4.1 Requirements of the Planning Protocol

The planning protocol contains two kinds of planning: incremental planning and central planning. Both processes start simultaneously, initiated by the source node when it begins to receive data packets from a local user application. The incremental planning uses only local planning that is only aware of local network conditions and adapter availability. Thus, incremental planning is fast and does not require any sophisticated resource reservation policy; if a necessary resource is available, the local planner can use it immediately for the connection. Central planning should replace the incremental plan

if a session requires the transfer of more than some hundreds of data packets. Short sessions should be satisfied with incremental planning only.

Central planning depends on the efficiency of three protocols:

- Planning data collection
- Plan calculation
- Plan deployment

In the most general case, the planning procedure involves connection nodes, the planning site, and adapter storage sites. The planning protocol should run sufficiently fast to be able to serve real-time applications.

The planning protocol starts with the exploration of network conditions around the connection nodes. The network resources found are considered a fact of life, and a new session should adjust to resources. However, resources of end-nodes of the connection can be redistributed by an issuer of the session. Adding another data transfer session affects the previously initiated sessions. The source node contains a resource manager (RM) that monitors the sessions. If there are plenty of resources, it allows the new session. If there are not enough resources for the new session, the RM uses a priority profile to reject the new session, or to preempt one of the previous sessions to release the resources for the new one, or to initiate replanning of all sessions to fit. Fair resource management is another requirement of the planning protocol.

As mentioned above, central planning involves multiple parties that can be located remotely for the source node and may be untrustworthy. The planning protocol should provide security for the sake of nodes, user application, and user data. The latter

requirement of planning securely is described in Section 4.13. However, it is not implemented in the current system.

4.2 Planning Protocol Design

The ONA planning protocol consists of the following steps:

1. Planning data must be collected the nodes of a connection and delivered to a planning node.
2. The planning node must calculate a plan and send it to an adapter storage site.
3. Adapter storage must deliver adapters to the connection nodes.

The planning procedure described above is presented in Figure 4.1. The connection consists of nodes A, B, C, D, and E. Node P runs the planner for the connection. Node S is the adapter storage node. Node A is the source node that initiates the connection. The planning request goes through the connection nodes, each of which contributes its planning data to the request. Node E sends the complete collection of planning data to node P, which runs the planner and sends the plan to node S and node A. Node S sends the adapters to the connection nodes according to the plan. The connection nodes send their acknowledgements to the adapter storage node S and to the source node A. When node A receives all acknowledgments, all adapters have been delivered to nodes that need them and have been deployed at these nodes. At this point, node A can use the plan to send data.

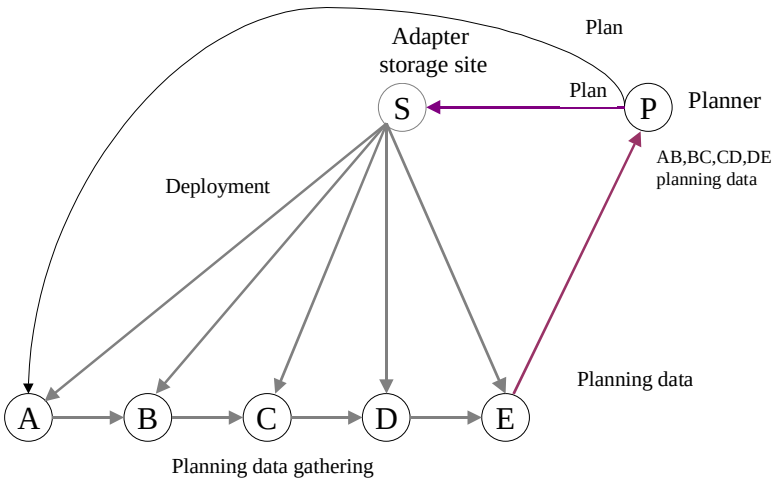


Figure 4.1: Planning procedure

4.3 Where to Run the Plan Calculation

The speed requirement of the planning protocol requires that plan calculation occurs on the node that first has all necessary planning data. As the process of data collection starts at the source node, the first node that has all planning data is the destination node (or a node prior to the destination node). The resource management requirement encourages putting the planner on the source node, where the RM can use the collected planning data and the planner to make decisions about how to treat the new data transfer session and the previously opened sessions. We chose the source node as the location for the planner in Panda.

Another question is that of where to put the adapter storage site. Some adapters can be stored on any network node; other adapters can be stored at special adapter storage

servers with a specialized access to users. But the interest of the user is to have the adapter storage site on the same node as the planner to reduce the latency and insecurity of the planning protocol. Thus, we chose the source node as the location for the adapter storage in Panda.

4.4 The Implementation of a Planning Protocol in Panda

In our Panda implementation the planner and the adapter storage site are located on the source node, which simplifies some steps of the planning procedure. The Panda planning procedure is presented on Figure 4.2. Node A initiates the connection by sending a request to node E through nodes B, C, and D, in phase 1. The picture shows that node E sends the planning data directly to source node A, where the planner and adapter storage sites are located in phase 2. The plan is handed to the adapter storage site located on node A, which deploys the chosen adapters on the connection nodes in phases 3 and 4. The connection nodes send only one set of acknowledgments to source node A to acknowledge that the adapters are received, and the plan is deployed.

The planning protocol is different for incremental planning and central planning. Thus, the insecure points of both protocols are different and require different solutions.

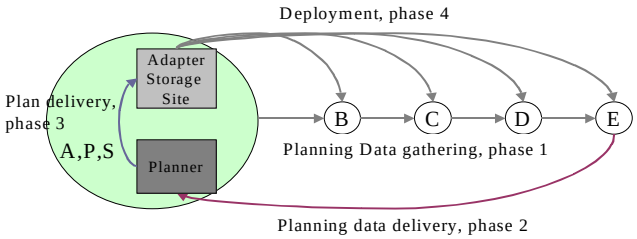


Figure 4.2: PANDA planning procedure

4.5 User Preferences, Data Stream Characteristics, and Planning Data

A node's planning data consists of its execution power, characterized by the number of adapters that the node is able run simultaneously. The planning data for links consists of bandwidth measured in Kbps, and a binary security metric: 1 standing for a secure link, 0 for an insecure link.

User preferences consist of the list of the user's preferred methods for handling network problems. For example, the user may prefer to deal with low bandwidth for video data by dropping its resolution, rather than converting it to black-and-white. User preferences also describe the network requirements. For example, a user application can claim the security for the data stream, meaning that no packet should go through an insecure link without being encrypted.

The stream characteristics are basically the properties of the stream. These characteristics include the required bandwidth for the stream. If the planner finds that the required bandwidth is less than the actual bandwidth of a connection link, it identifies this

situation as a network problem that requires reducing of the amount of data to make it fit in the available link bandwidth. Another characteristic is the security of the stream, i.e. whether the stream is encrypted by a user application or not. If the stream is not encrypted but the user preferences require security, this could be a reason for the planner to apply encryption. *Compressability* attributes are a very important set of stream characteristics. There are the ability of the stream to be compressed using various techniques, such as Lempel Ziv compression, color filtering, quality filtering, etc. The format of the data is also indicated in the stream characteristics. The characteristics are dependent upon each other. For example, if the data packet is encrypted, all kinds of compressability are equal to 0. If the application compressed the data packet using a Lempel Ziv technique, compressability is also equal to 0 because the data is in a format unusable by filters (such as color drop, etc.). But if the data is decompressed somewhere, all its compressability will be restored. User data may not have a particular compressabilty. Then, compressability is not in the stream characteristics and cannot be used by the planner to reduce the amount of data. For example, if a data is not a video stream, it does not have color or quality compressability. These characteristics are not mentioned in the stream characteristics, and the planner will not use color or resolution drop filters on the data. A compressor can be applied to data that has a proper type of compressability at a value greater than 0.

An example of planning data is presented in Appendix F. An example of user preferences and stream characteristics is presented in Appendix E.

4.6 Planning Data Collection

We presume that any active network node collects and stores planning data about its neighbors and adjacent links. Thus, a local plan can be created without a special planning information gathering procedure.

The circumstances are different for central planning, which must collect planning information about all nodes and links that participate in the connection. The process of planning data gathering should occur on-line, during the handshaking phase, before the connection is established.

The user application starts the data transfer. The source node intercepts the application data packages and stores them in Panda's buffer module. At the same time it initiates the planning protocol. It creates a request for planning. The request contains

- User application solution preferences
- User application data stream characteristics obtained from a user profile
- Planning data on the first connection link collected by the source node offline

The source node forwards the request to a neighbor. The neighbor adds the planning data on the next node on the path to the destination node and the link between them. When the request reaches the destination node it contains the user application preferences, user application data stream characteristics, and the planning data about the connection nodes and the links between them. The destination node sends the data to the planning node, which is the source node in our implementation (as shown in Section 4.2). Since the user application solution preferences and data stream characteristics are already

located on the planning site, they need not be sent back there. But they must be sent to connection nodes to support incremental planning.

The planning data collection is implemented in the special *PACentralPlanning* capsule that sequentially visits the connection nodes and collects the data. The capsule is a mobile component inherited by Panda from ANTS (as mentioned in Chapter 2).

The code of *PACentralPlanning* is presented in Appendix A.

4.7 Calculation of a Plan

The calculation of a plan requires the stream characteristics, user application preferences, and planning data. This data is provided by the planning data collection protocol. The plan calculation consists of adapter selection, adapter ordering, and plan optimization.

The detailed algorithm of the plan calculation is presented in the next chapter. Once the plan is calculated, the adapter storage site, which is the source node in our implementation, runs the deployment protocol.

The planner is an independent component that is used by the planning protocol through a specified interface.

4.8 Deployment of a Plan

The last step of the planning protocol is the plan deployment.

Adapters are serializable chunks of code. Panda puts them in separate packages and sends them to the nodes that require them. When a node successfully deploys an

adapter, it sends an acknowledgement to the source node. The source collects these acknowledgements until they all arrive. Then the source node starts sending data packets -- unless it is already sending them using an incremental plan. In latter case, it sends data packets using the newly deployed central plan. The data stream packets go through the adapters located on the connection nodes. Every data packet carries the plan that must be applied. The plan is necessary to instruct the connection node which adapter and in which order to execute the adapter on the data of the packet. The adaptation modifies the data stream, improving its QoS.

4.9 Incremental Planning Protocol

The incremental planning protocol starts at the same time as does the central planning protocol. Normally it lasts longer than the planning data collection phase of the central planning protocol. It completes near the end of the central plan calculation and provides service during central plan deployment. Once the central plan is deployed it requires the incremental plan.

The source node sends the incremental planning request implemented as *PAIncrementalPlanning* capsule. The request initiates the local planning process on the source node and the second connection node. The local planner selects the adapters necessary to handle the network conditions between the two nodes. The planner chooses the adapters that are locally available. The adapters do not need to be delivered through the network, but they need to be registered on the local Panda nodes. Once the local plan is calculated and deployed, the second node runs the local planning for the next link. The

planning process involves the second and third connection node. The process continues, until the local planning process is executed for the last connection link. Once the incremental planning is completed, the destination node sends an acknowledgement to the source node. Once the source node receives the acknowledgement, it starts the data transfer using the incremental plan. Normally, the user data transfer cannot be started before the incremental plan is completed. However, sometimes the central planning process finishes faster; then the incremental plan is disregarded. This is a relatively rare event, but it can occur on network nodes that have high execution power and links have enough bandwidth to deliver the designated adapters fast.

The code of *PAIncrementalPlanning* is presented in Appendix B.

4.10 Elements of Resource Management

The problem of resource management is very complicated. We have reduced our model to resource management for planning only and have made some assumptions that allowed us to limit of the scope of this problem. We implemented resource management in our planning system.

4.10.1 Assumptions of resource management

To simplify the RM design we make a number of assumptions:

1. *Enough resources are available for monitoring and planning.* We assume that there are enough link resources in nodes and links to send any planning message to another node, executional resources to run planning, etc.

2. *Knowledge about session streams is available.* We assume that knowledge is available concerning user application data streams. For example, applications can support multiple data streams that send data to the same destination node. We want to know how to group and prioritize the data streams by users, applications, etc. The more information available, the more effective monitoring will be. The Panda planner using this data is able to make decisions about what to do with running sessions when a new session starts, i.e., to stop them or replan them.
3. *Resource sensing messages have the ability to reserve resources until the planning process completes.* We assume that there is a mechanism to prevent the resources found and relied on by one planner from being also found and relied on by another planner.
4. *Each Panda node has an access to a service that monitors available resources on this node, all adjacent links, and all neighboring nodes.* We assume that the Panda observation component (POC) collects the information about adjacent link and node resources. The POC respects the reservations made by planners.
5. *The connections that do not start or end on the node that plans a new connection cannot be revised by that node planner.* Any connection is influenced by other connections that involve at least one node that belongs to the first connection. The connections that start or end on a node where a new session is about to start can be revised by the node planner (as shown in Figure 4.3). Node A is a source node for a new session AD. Node A is also the source node for session AB and the destination node for session EA. The resource distribution for the new session AD is also

affected by the session CF. The resource manager of node A can reconsider the resource distribution for sessions AB, EA, and AD, but not for CF. The sessions of high priority can be exempted from the revision. The planner can terminate a session of lower priority to start a new connection. The planner can reject a new connection if there are not enough resources and no sessions to terminate. The planner can replan a number of sessions of appropriate priority to give more space to the new connection.

4.10.2 Resource management

According to assumption 5, the planner must take into account all sessions that start or end on the sender. In the current implementation the planner is located on the source node, therefore the message that collects the resource information makes the trip to the destination and returns to the source node. The RM makes the decision to reject the new data stream or accept it through the preemption of another stream or through replanning of other streams if there are not enough resources for all. After a connection is terminated, the resources it used should be released and reused by other connections. The RM periodically detects inactive connections and decommissions their plans. The released resources can be distributed among active connections or wait for new connections.

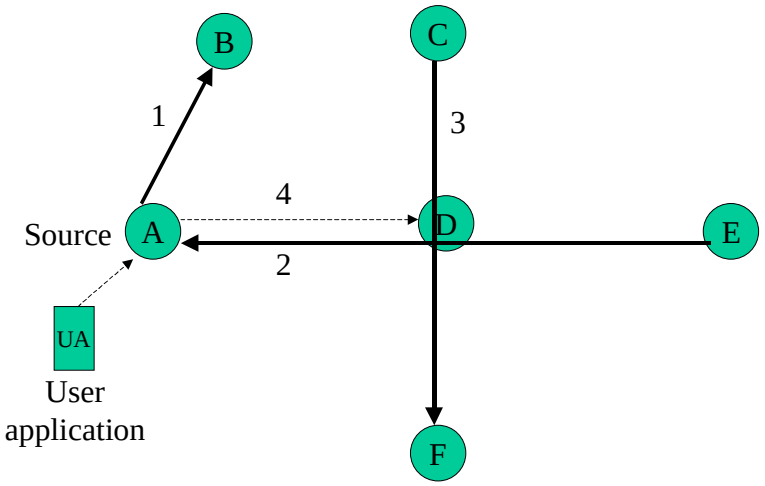


Figure 4.3: Resource redistribution among sessions. Sessions 1, 2, and 3 influence the new session 4. The planner on node A revises only sessions 1 and 2, and ignores 3 upon the request of the user application.

In Panda the following primitives are implemented:

- Add flow
- Reject flow
- Preempt flow
- Replan flow

The policy of how to treat the flows – to add, to reject, to preempt, or to replan – is formulated in the user preferences.

Add a new connection if there are enough resources. If the RM finds that there are enough spare connection resources for the new flow, it permits the flow to be added. The protocol for the adding the new flow is shown in Figure 4.4.

Reject the new connection if its priority is lower than the priority of all active connections. If the RM finds that there are not enough spare connection resources, it can reject the new flow. The protocol for rejection of a new flow is shown in Figure 4.5.

Preempt the Flow. If the RM finds that there are not enough spare connection resources and the priority of the new flow is higher than the priority of some existing flows, it can preempt these flows. The RM returns the list of the flows that should be preempted. These flows continue the data transfer while the new flow performs its planning and deployment. According to the assumption 1, there are always enough resources for the planning and deployment of a new flow. When the plan of the new, high-priority flow is computed, the flows that should be preempted are stopped and the entire planning procedure is repeated, starting with a sensing of the whole channel again. Those flows take their chances, relying on the fact that the new flow might not use all spare resources of the connection; the RM might reject some of these flows. The flow preemption protocol is shown in Figure 4.6. As an option, the preempted flows can be completely shut down by the RM.

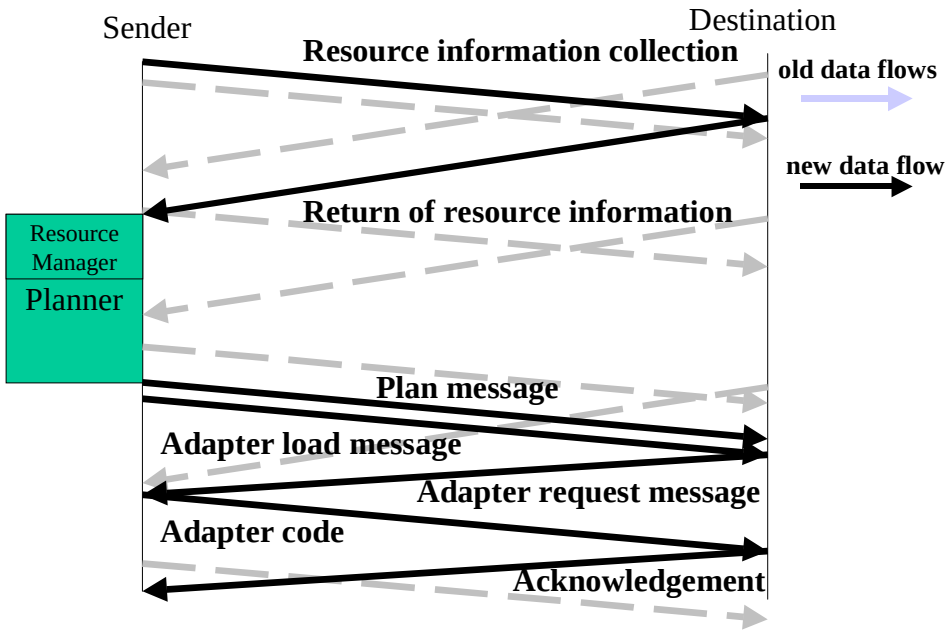


Figure 4.4: Add Flow: The resource manager permits establishing a new flow transparent to all other data transfer flows that occur in the channel.

Replan of resources among equal-priority connections. If the RM finds that there are not enough spare connection resources, and the priority of the new flow is not higher than the priority of the flows that are already running, the RM can make the decision to replan the group of equal priority flows with fair resource redistribution among flows. Thus, the new connection can be added to the list of running flows. The process of replanning old flows and planning new ones occurs simultaneously. Switching to the new plans occurs when the new plans are deployed.

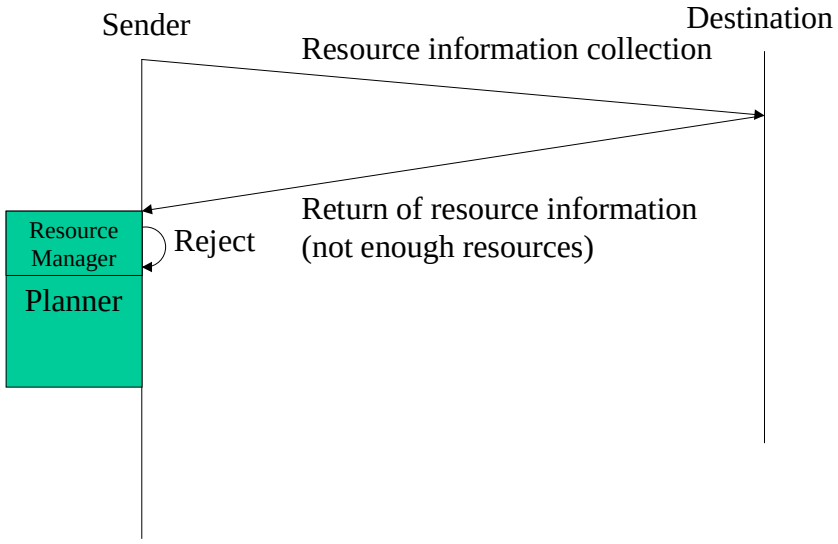


Figure 4.5: Reject flow. The lack of resources and lack of sessions that could be preempted make the resource manager decide to reject the new session.

The replanning process occurs on one node, which can be source for some nodes and destination for other nodes. To switch to a new plan on the remote source node, the node sends special messages with a newly calculated plan for that connection. If the plan successfully switches to the remote node, the remote node sends an acknowledgement back to the node where the plan was calculated. New plans are switched on upon their deployment. If the new plan is not acceptable at the remote source node, the remote source node should replan its flow with respect to the new resource distribution. The protocol of replanning is shown on Figure 4.7.

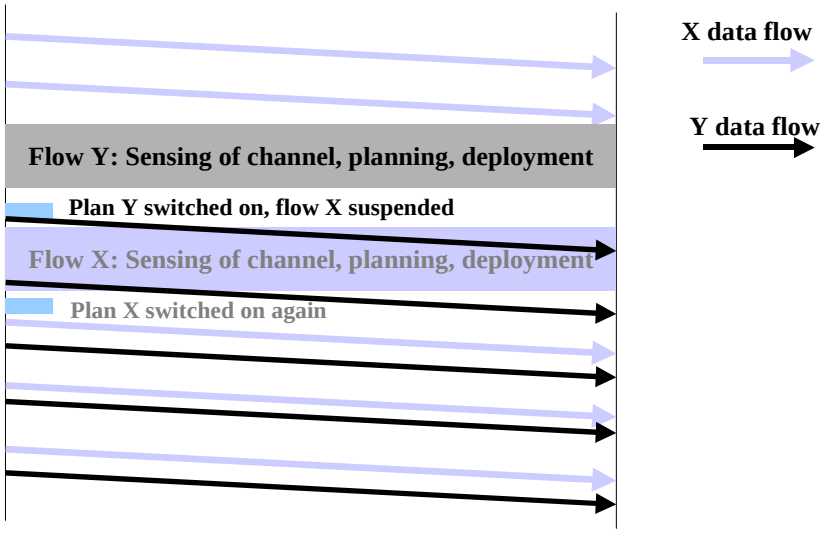


Figure 4.6: Preemption of flow X by flow Y: The planner calculates a new plan and deploys it. After the switch to new plan Y, the plan for flow X should be replanned and redeployed.

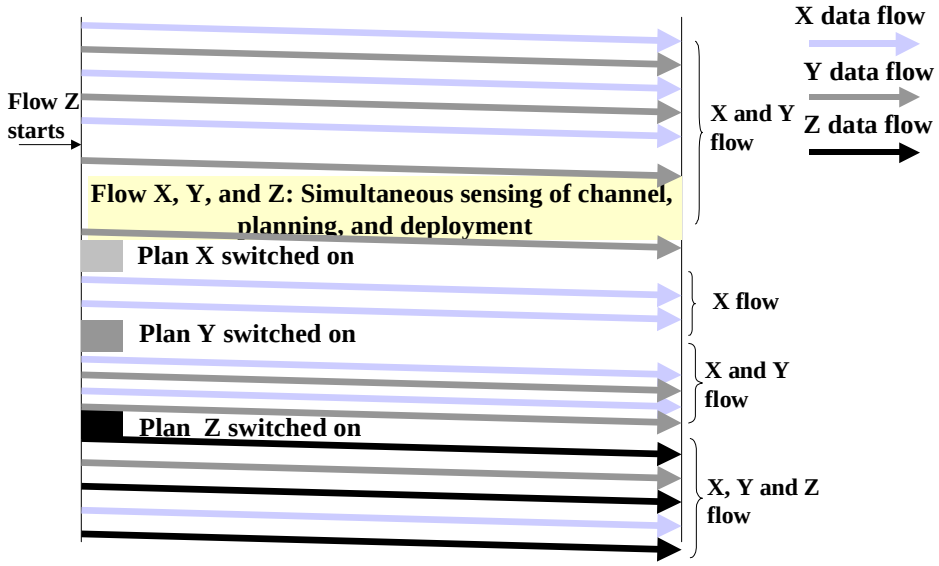


Figure 4.7: Replanning of flows X and Y with flow Z added: The planner calculates new plans for flows X, Y, and Z and deploys them. After switching on X and Y, the new plan Z can also be switched on too.

4.10.3 Periodic decommissioning of inactive connections

In many cases with legacy user applications, Panda cannot detect when a particular session is over. So Panda nodes periodically identify inactive connections and delete them. When a particular connection is taken away from the list of active connections, its resources are released and can be used by other active connections. The RM runs the replanning process for the list of active connections.

The active plans and inactive plans are stored on source and destination nodes only. A node decommissions only those plans for which it is the sending node. For the rest of the connections in which it participates, it sends a message to the senders to initiate replanning. This may not be necessary, as other nodes run their plan-decommission procedures also.

The protocol of replanning after plan decommissioning is shown on Figure 4.8.

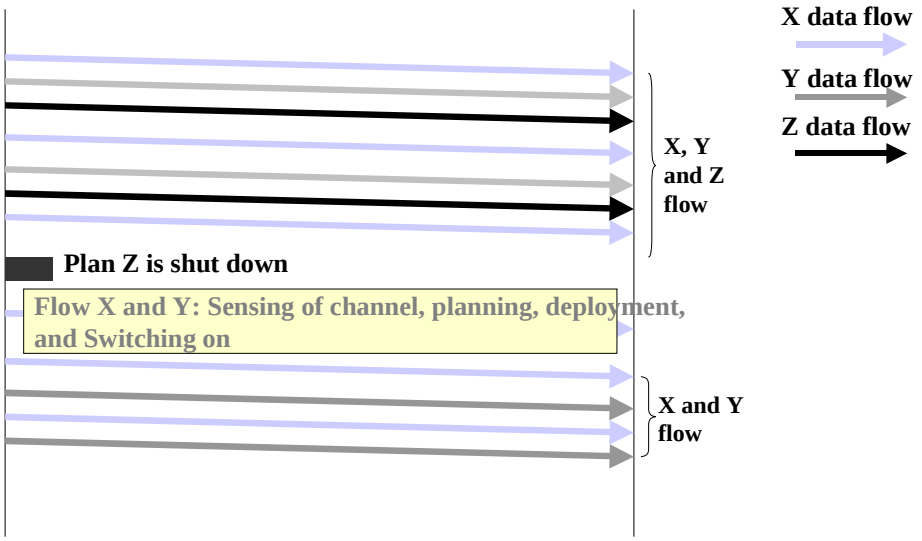


Figure 4.8: Replanning of flows X and Y when Flow Z is shut down: The planner recalculates new plans for flows X and Y and redeploys them.

4.11 Replanning

If the RM on any Panda node using the POC component detects the reduction of spare connection resources, it sends a request to replan to the sources of all data packets that arrive (Figure 4.9). The process of replanning starts with a sensing of the channel by the senders. The connection resources will be distributed on a FCFS basis. The sources that are closer to the problematic node will have the advantage of capturing the resources first.

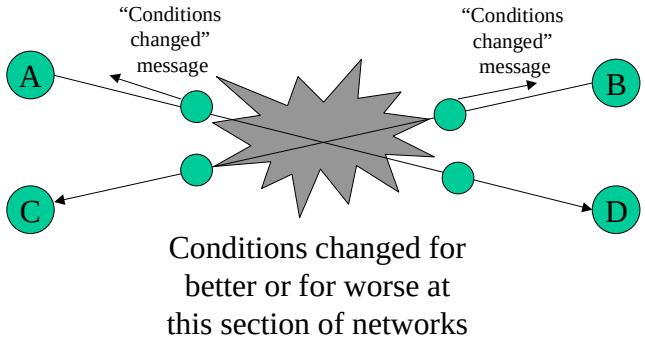


Figure 4.9: Replanning if conditions change at a particular section of networks. Border Panda nodes inform the senders about the change and initiate the replanning process described before. The scheme presumes that some method of guaranteeing fairness is implemented.

This approach raises the issue of fairness with respect to different distances between the problem node and the sender and how to share the spare resources between different senders. It also requires cooperation between different connections. The

implemented solution does not pretend to be general; it offers basic functionality only, which can be improved by future research.

4.12 User Access to Resources

The resources of the network should be distributed fairly. This raises the question of proper user authentication. The planning protocol must make ensure that the right user uses the right resource. This problem is addressed in very early documents on ONA planning [Zegura98].

4.13 Security in ONA Planning

The issue of security is very important for ONA functionality. As with any distributed system, Panda can be attacked or corrupted. Major problems of security lie in:

- protection of user data from untrustworthy ONA nodes or adapters
- protection of ONA node from malicious user applications or adapters, including node resource hogging
- other attacks, for example specific viruses designed to affect the functionality of the ONA node

Most of these problems are already addressed in ONA designs. [Murphy01] describes SANTS, a solution for AN security problems including authentication, a policy system, and a certificate distribution scheme. The misbehavior of malicious adapters toward an ONA node is handled by sandboxing and code verification techniques

developed in Java, the major programming language for AN design. Resource hogging is handled by fair resource allocation with the Active Resource Protocol (ARP), an RSVP-like protocol designed for AN [Braden01]. However, scalable techniques that protect user data using ONA capabilities are not well developed yet. Security techniques that support proper ONA functioning must be properly chosen, because security measures are resource-costly by their nature and must be concentrated at particular insecure points.

The Panda planning procedure is particularly vulnerable to attacks because it requires the cooperation of multiple nodes that participate in the connection establishment and data transfer. The nodes that participate in the plan data gathering can report false network information, the plan calculation can be run on a node that is not trustworthy, adapter storage sites may not be trustworthy, or the adapters can be substituted during delivery by malicious sites.

This section describes a number of security problems in ONA planning and outlines ways to solve them.

4.13.1 Classification of Panda planning security issues

The ONA planning procedure can suffer from a variety of security problems. They are defined as follows:

1. An untrustworthy ONA node that belongs to the connection may be a problem during all stages of the planning procedure and data transmission. If an ONA node reports false data on the state of adapter execution capabilities and adjacent link conditions, it will cause the wrong plan to be calculated. This leads to deployment of the wrong adapters, which will affect user data transmission. The ONA node can also

sabotage the deployment of adapters or deploy the wrong adapters, which can also affect the data transfer or the integrity of data.

2. A malicious planning node that may or may not belong to the connection is a problem because the node can calculate a bad plan, which will affect the data transfer.

3. A false plan is a problem because it will cause the deployment of the wrong adapters or increase the latency of deployment, etc. False plans arise when an attacker switches or modifies the plan on the way from the planning node to the deployment node.

4. An untrustworthy adapter storage site is a problem because it can disrupt up the whole deployment process.

5. An untrustworthy adapter causes problems when an attacker substitutes a malicious adapter for the adapter chosen by the planner.

6. Poor adapter design is a problem if an untrustworthy adapter designer releases adapters with backdoors or with poor design, because they will not provide proper service to the connection.

7. Improper requests for adapters (similar to an attempt to purchase a drug without a proper prescription) may be a problem in the future if the power or side effects of a particular adapter can be used for some malicious purposes or without proper license. For example, an escrow adapter designed for authorized use can only be used for the illegal decryption of user data; or a user may not be eligible to expend the amount of resources needed by the chosen adapter.

The problem of secure planning is different for incremental and central planning. Secure incremental and central planning procedures were presented in Sections 4.6.3 through 4.6.6. Problems 6 and 7 were discussed in Sections 4.6.7 and 4.6.8.

4.13.2 Security of incremental planning protocol

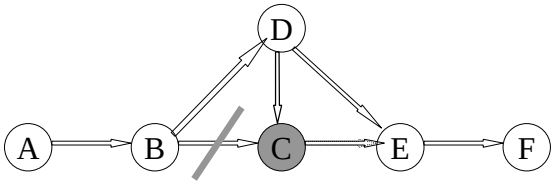
Incremental planning does not suffer from most of the planning insecurities outlined above. Recall that incremental planning occurs on every pair of adjacent nodes during the connection establishment. Incremental planning consists of the calculations of local plans that cover the planning node, its next-neighbor node, and the link between them. We assume that the connection nodes calculate local plans using their own planning facility. Thus, if a node trusts another node, it also trusts its planner. If all adapters chosen by the local plan are already deployed on the two nodes, the adapters are all trustworthy, since these nodes have verified adapter trustworthiness before deploying them. That is why problems 2, 3, 4, and 5 from the previous section are not applicable to incremental planning. However, the problem of untrustworthy Panda nodes remains an important issue here.

Let us assume that the source node can trust the connection nodes to reasonably judge their neighbors on the subject of trustworthiness. During the connection establishment, any connection node can build a local plan toward the trustworthy neighbor and refuse to build a plan toward an untrustworthy one. Then we can apply incremental planning. If the source node does not trust the ability of the connection nodes to judge the trustworthiness of their neighbors, the incremental planning procedure cannot be used, and the session must wait until the centralized planning procedure is completed.

If a connection node does not trust its neighbor that it wants to use as a next node to the destination, it can try to avoid it. It can use some other trustworthy node from its direct neighborhood. If there is another trustworthy node, other than the previous connection node, it can forward the connection initiative to that node. There is a chance that the node has its own way of choosing trustworthy nodes to reach the destination. Figure 4.10 presents this situation.

In Figure 4.10, node B does not trust node C. Node B forwards the incremental planning initiative to node D. Three possibilities may occur here. First, node D perfectly trusts node C and uses it as a next node, thereby ignoring node B's suspicion. Security policies of the connection can forbid it. Second, node D may not have any way to reach the destination, and incremental planning fails. Third, node D is able to reach node F through node E.

If the incremental planning procedure cannot find a secure path to the destination, it should be stopped. The connection establishment should wait until the central planning procedure is completed.



Connection through links AB, **BC**, CE, EF is not possible because B does not trust C:
1. Connection AB, BD, **DC**, CE, EF is possible (B trusts D, D trusts C).
2. If B forbids D to trust C and DE does not exist, the secure connection cannot be made.
3. Connection AB, BD, DE, EF is possible.

Figure 4.10: Incremental planning procedure

4.13.3 Untrustworthy ONA node for central planning protocol

An untrustworthy Panda node can be a serious problem for Panda functionality:

- It can falsify planning data requested by the planner.
- It can mistreat the planner instructions.

We assume the misbehavior of ONA nodes can be detected using some observational technique. This technique can be some type of automated monitoring of node behavior, or it can just be long-term statistics collected about the node's participation in various connections. Such observation makes it possible to judge the trustworthiness of a node. We assume that it is possible to implement a service that can answer questions about the trustworthiness of a particular node. The ONA connection then can avoid the use of untrustworthy nodes. These nodes should be treated as normal Internet nodes. The harm that this type of node can do to the connection as an Internet node, such as packet delay

or packet dropping, can be handled by conventional Internet techniques. An ONA-capable untrustworthy node that is on the path of the connection can increase insecurity to the adjacent links, and extra security measures (such as encryption of the data) should be provided. We assume that there exists a protocol called *tête-a-tête* that determines, in real time, planning data between two trustworthy nodes when one or more untrustworthy ONA nodes between them need to be omitted. This technique is presented in Figure 4.11. Nodes A and C run the tete-a-tete protocol exempting node B from the connection, treating it as a normal Internet node. Normally, the tete-a-tete protocol uses a third trustworthy party that controls the dialog between A and C. It allows stronger control of malicious behavior of untrustworthy node B. For example, the third party can help to deploy an encryption adapter or distribute an encryption key between A and C.

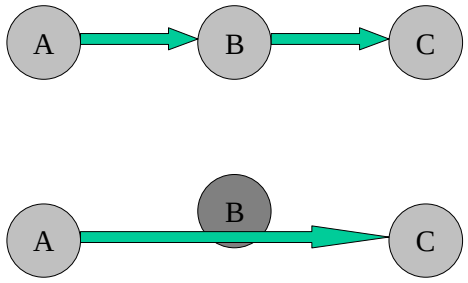


Figure 4.11: Exemption of an intermediate node(s) in planning data collection via the tête-a-tête protocol

All techniques for physical avoidance of the untrustworthy node are also applicable.

4.13.4 Secure central planning procedure

The ONA central planning protocol consists of a number of phases: planning data gathering, plan calculation, and deployment procedure. The participants are the source node, the destination node, the intermediate nodes, the planning node, and the adapter storage site. The source node that initiates the connection chooses the destination node, planning node and adapter storage site. These nodes can be chosen for their trustworthiness. The only nodes that cannot be freely chosen by the source node are the intermediate nodes. We assume there will be a service, *Untrustworthy Node Check (UNC),* that is able to answer "yes" or "no" to the question of whether a particular node chosen for connection by the ONA routing protocol is trustworthy. The service will be located on what will be called UNC nodes. How the central planning procedure uses the UNC service is presented in Figure 4.12.

In this figure, node A initiates the connection with node E. Nodes B, C, and D are intermediate nodes chosen by the ONA routing protocol. After all planning data is collected at node E, it is sent to node L, which then approves the trustworthiness of the intermediate nodes. Then the planning data is sent to node P for calculation of the plan. Node P sends the plan to node S where the adapters are stored. Node S runs the deployment protocol with the connection nodes. The connection nodes acknowledge the successful deployment of adapters to node A. Node A starts the user data transfer. If the UNC service does not approve a particular node, the UNC asks adjacent trustworthy nodes to run the tete-a-tete protocol to determine the planning data between them. The

determined planning data is then sent back to the UNC node. Following this, the UNC node resumes the planning process. This protocol is presented on Figure 4.13.

Another alternative is for each connection node to verify its neighbor's trustworthiness using UNC. Alternative paths that avoid untrustworthy nodes can be found. This approach requires a special routing table that would allow optimized alternative paths through trustworthy nodes. If a node is surrounded by untrustworthy nodes, the traffic must still go through the untrustworthy nodes.

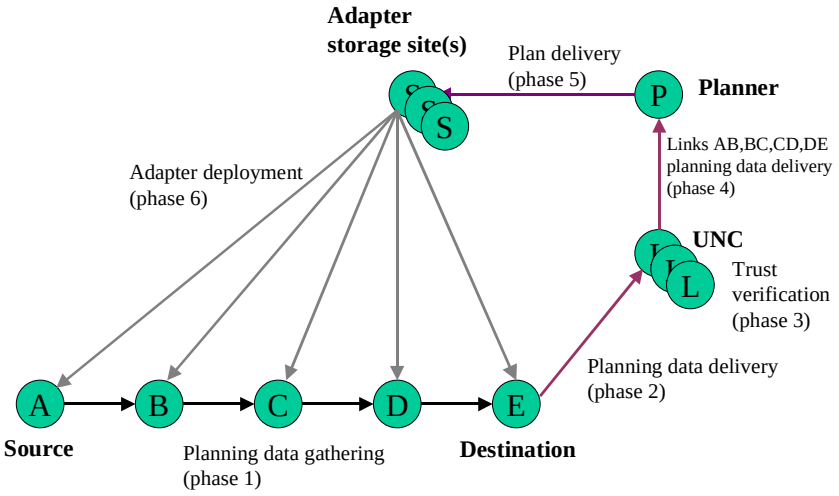


Figure 4.12: Central planning procedure with UNC service

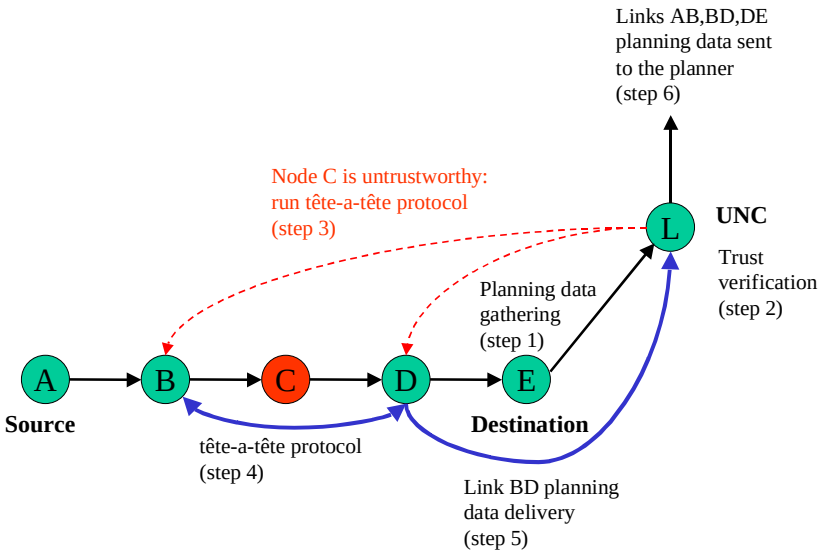


Figure 4.13: UNC node does not approve node C as a trustworthy node

The security of the protocol in Figure 4.12 requires verification of the identity of trustworthy sites and verification that data was not changed by an attacker during the delivery. If necessary, the data can be also encrypted. Verification of the identity and the protection of data can be achieved using electronic signatures of the data by the participating trustworthy sites (whose public key is known to all participating parties).

4.13.5 Secure central planning protocol

The secure central planning protocol is presented in Figure 4.12. The destination node is chosen by a user. When source node A begins the planning process, it chooses one or more UNC nodes, a planning node, and one or more adapter storage sites (AS), nodes E, L, P, and S respectively. Node A creates a planning request that contains the following:

1. Source node ID
2. Node B ID (A's next node to the destination)

3. Session ID (unique ID that prevents the re-use of any planning request attribute)
4. UNC ID (node L, which, in reality, can be more than one node)
5. Planner ID (node P)
6. AS ID (node S, which, in reality, can be more than one node)
7. Adjacent planning data (planning data collected by the ONA observation component and stored on adjacent nodes; in this case it is AB planning data)
8. User application preferences

Node A signs the planning request using its private key and sends it to node B. Node B receives the signed request, verifies A's signature using A's public key, and stores request attributes 1, 3, 4, 5, and 6. The attribute IDs will be used by node B in the subsequent steps of connection establishment. Thus, node B will inform the source node that the adapters are deployed, node B will then verify that a plan comes from the proper planner and adapters from the proper adapter storage site, etc. Node B adds the following items to the original request:

9. Node C ID (B's next node to the destination)
10. Link BC planning data

Node B signs the resulting request with its private key and forwards it to node C. Nodes C and D repeat this procedure, by turn, and forward the request to node E. Node E verifies the signature of node D, stores the request attributes 1, 3, 4, 5, and 6, signs the resulting request with its public key and forwards it to UNC node L. This procedure

prevents an attacker from changing the collected planning data during the delivery to node L.

Node L verifies the integrity of the collected planning data using the signatures of all connection nodes by applying their public keys. UNC ID indicates that node L was chosen by node A to verify the trustworthiness of the intermediate nodes. Node L approves the trustworthiness of the intermediate nodes. If one of the nodes is untrustworthy, node L asks neighboring trustworthy connection nodes to run the tete-a-tete protocol, determine planning data between, and forward the planning data again to node L. In the latter case, adjacent trustworthy nodes use the UNC ID to accept the UNC command to run the tete-a-tete protocol. Then node L modifies the planning data if necessary, signs the request with its private key, and forwards the request to node P, the planner indicated in the planning request.

Node P verifies the signature of node L to ensure that the correct node L has verified the trustworthiness of connection nodes, and verifies the signature of node A to ensure that node A chose node P as a planner, using public keys for L and A. Then node P calculates the plan using the delivered planning data, signs it with its private key, and forwards it to node S.

Node S verifies the signature of node P to ensure that the plan comes from the proper planner and verifies node A to ensure that node A chose node S as a designated AS. Recall that there can be more than one adapter storage site. Node S runs the deployment protocol. It creates packets for every connection node that contain:

1. Session ID

2. AS ID
3. Planner ID
4. Source node ID
5. Plan for the node
6. Adapters that should be deployed on the connection node

Node S signs the package and forwards it to the correspondent connection node. If necessary, it can verify the eligibility of the connection node to use some adapters. It may require extra handshaking between the connection node and node S. (A more detailed discussion on the problem of improper requests for adapters follows below.) Node S also sends the signed plan to the source node. This could also be done by node P. The message contains session ID, AS ID or planner ID, and the plan.

The connection node verifies the signature of node S (it should correspond to the AS ID), compares the session ID (so the package cannot be reused), source node ID, and plan for the node. It deploys the adapters and sends the signed acknowledgment that contains the session ID to the source node and AS node.

The source node verifies whether or not all the adapters are deployed according to the plan that was obtained from node P or node S. If all adapters are deployed, the source node begins the data transfer.

The protocol presented above precludes an unauthorized modification or malicious reuse of planning messages.

4.13.6 Poor quality and malicious design of adapters

We believe that in the future adapters will be designed and distributed by numerous parties. The adapters will be installed by individual users and large adapter warehouses. This huge number of widely and freely available adapters makes it hard to imagine that there will be any administrative procedure that could really guarantee that these adapters will do what they are designed to do. We believe that the adapters should be signed by certified adapter designers. These signatures should allow automatic tracing of the origin of any adapter by any user at any moment. The uncertain origin of an adapter or improper credentials of the originator of the adapter will cause users to not choose that adapter. Proper credentials of adapter designers will be a matter of legal obligation and reputation; as it is with other trademarks. Ideally, credential verification should contain only the verification of the designer's signature using his/her public key. The statistics on adapter package usage and adapter designers' signature verification can be handled by adapter storage sites in cooperation with clients. In the latter case, only adapter storage signature verification during a connection establishment is necessary.

4.13.7 Security measures on adapter requests

An ONA must have a mechanism that will guarantee access of the proper user to the proper adapters. For example, assume that special escrow adapters will be designed for legal purposes. These adapters will be able to decrypt any encrypted user data. Of course, these adapters would not be publicly available, and a special legal procedure to request them must be instantiated. Another example is the adapters that require a paid subscription in order to be accessed by a user.

There are a number of approaches to preventing access to these adapters. For example, the planner site or adapter storage site can ask a connection node for a proper certificate permitting access to a particular adapter.

4.13.8 Electronic signatures with public key and key distribution

The authentication procedure that was described above can be achieved using public key encryption. A typical electronic signature contains the checksum of the data encrypted with the signer's private key. To verify the authenticity of data, one must apply the signer's public key to the signature and compare the result with the checksum of the data delivered. If the two values are identical, the integrity of the data is proved. If a data package should be signed by different parties, each signer adds its data to the packet and signs the whole package, including all data and all previous signatures. This approach is presented on Figure 4.14.

Packet N was signed first, then it was delivered to the $N+1^{st}$ node and signed again by node N+1.

A public key distribution technique that is suitable for WANs should be available for ONA authentication. A key distribution and management toolkit (KMT) suitable for AN key distribution is presented in [Gudmundsson98]. Other public key infrastructures also can be used.

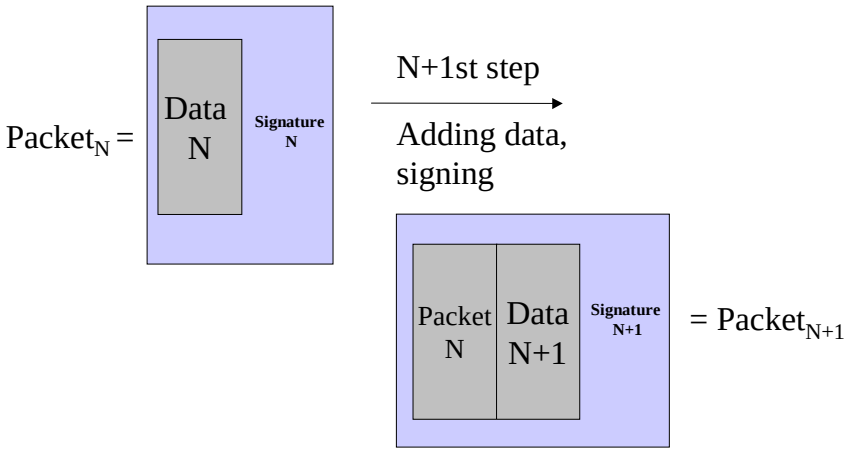


Figure 4.14: The chain of signatures

4.14 Summary

In this section we presented the requirements for the planning protocol:

- Performance
- Resource management
- Security.

We presented the implementation of the planning protocol, including issues of performance and resource management. The implementation of the secure planning protocol was left for future work.