A Generic and Modular Accounting and Charging System for Peer-to-Peer Applications *

David Hausheer[1], Jan Gerke[1], Burkhard Stiller[2,1]
[1]ETH Zürich, TIK, Switzerland
[2]University of Zürich, IFI, Switzerland
{hausheer,gerke,stiller}@tik.ee.ethz.ch

Abstract: Accountability is one of the keys to success for emerging P2P systems. In the absence of appropriate accounting and charging mechanisms, P2P applications may suffer from free-riders which benefit from other peers without contributing themselves. This paper describes a new Accounting and Charging system for P2P applications which separates generic accounting and charging functionality from individual underlying accounting mechanisms. The presented approach enables the design and implementation of different accounting schemes, independently from core accounting functionality. The approach has been implemented and successfully used with several accounting schemes in different P2P applications.

1 Introduction

P2P applications like file-sharing benefit from scalability and performance of completely decentralized P2P systems. However, existing P2P systems are vulnerable against selfish behavior and provide little support for commercial applications. To alleviate this problem, recent P2P applications apply specific accounting mechanisms to enforce balance between resource usage and consumption by individual peers. However, most of these approaches such as BitTorrent's tit-for-tat mechanism (cf. [Coh03]) or eMule's credit system are file sharing-oriented and thus not easily applicable to other decentralized applications, *e.g.*, distributed storage systems or grid computing applications.

A generic and modular Accounting and Charging (A&C) system has been designed and implemented in the scope of the MMAPPS project. The system is a key component within a middleware for P2P applications and supports the implementation of different accounting schemes. It provides generic accounting and charging functionality such as account configuration, session maintenance, and tariff evaluation, independently from any underlying accounting mechanism. Thus, the system enables the use of a variety of accounting schemes with different properties for potentially any P2P application.

*The authors would like to acknowledge all MMAPPS project partners, in particular Nicolas Liebau, Ben Strulo, and Jeff Farr. More information about the project is available at http://www.mmapps.org/.

2 Design Principles

The main goal of an A&C system is to provide accountability, i.e. to offer mechanisms that make the use of services or resources offered within a P2P application accountable. As such it gives peers incentives to contribute own resources and serves as a basis to punish selfish behavior like free-riding. Vital accounting mechanisms are the processing of accounting events describing the service units used and the application of respective tariffs, as well as storage and aggregation of accounting data to keep track of the peers' balances. The proposed A&C system operates on the basis of *accounting sessions* and distinguishes two types of accounts, *session accounts* and *peer accounts*. A session, usually initiated by a consumer peer, refers to the use of a particular service, *e.g.*, the download of a particular file or the use of computing power. For every session, there is a Service Level Agreement (SLA), typically proposed by the provider peer, including a tariff that specifies how the session is accounted for. The SLA is agreed upon and signed by both session partners and stored within the A&C system, an instance of which is installed on every peer within a particular application. While session accounts are used to store and aggregate accounting data within a particular session, *e.g.*, the amount of MBs downloaded or the number of CPU cycles used, peer accounts provide the ability to aggregate information from several sessions to keep track of the total amount services provided and used by a particular peer. Note that peer accounts may also hold other information, *e.g.*, a peer's reputation for trust.

Tariffs are used to process accounting events which are generated by the service instances running on both peers within a session. Accounting events contain service-specific information and are analyzed and evaluated by a specific *tariff formula* which may include several *tariff parameters*. Tariff parameters can be static or computed dynamically during run-time, *e.g.*, depending on the time of day or the balance of a particular account. Based on the result from a tariff evaluation a generic *balance update* is created and forwarded to the appropriate account. In addition, accounts can be configured to report back their balance to tariffs at specific conditions, *e.g.*, at the beginning or end of a session, when a particular threshold has been reached or after a certain time has passed. This enables a variety of settlement schemes like pre-payment, post-payment, or more fine-grained schemes.

An important feature of the A&C system is the ability to use different *accounting schemes*. The goal of an accounting scheme is to provide individual mechanisms for book keeping of accounts, i.e. storage and aggregation of balance updates and their exchange with other peers. Accounting schemes may implement any specific type of accounting, from simple local or centralized accounting to more sophisticated decentralized or token-based accounting. As such they usually fulfill specific requirements with respect to efficiency, scalability, and security, among which there is always a trade-off. [Hau04] describes several accounting schemes that have been implemented into the A&C system. Depending on the requirements, different schemes may be used for individual accounts. For compatibility reasons, however, it is important that all peers within an application use the same schemes. Therefore, every application has to provide a configuration policy that specifies which accounting schemes are used to hold what type of accounts.

3 Implementation

The presented A&C system is embedded as a key component within a middleware for P2P services [GS05], a prototype of which has been implemented. On every peer within an application an instance of the middleware has to be installed and configured appropriately. As depicted in Figure 1, the A&C system has a core component which is further separated into an accounting configuration (AC) module and session management (SM) module. While the AC module enforces configuration policies provided by different applications, the SM module maintains accounting sessions and holds copies of the corresponding SLAs.

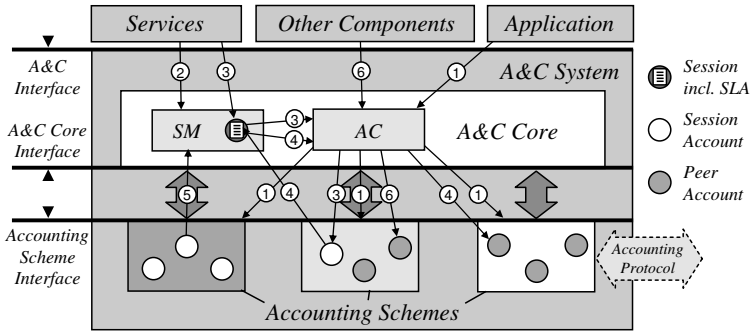

Figure 1: Accounting and Charging System

The A&C system comprises three main interfaces. The *A&C interface* is the only one that can be accessed from other middleware components. It is used to configure the A&C system, notify new sessions, send accounting events, and to query accounts. The second interface, called *A&C core interface*, is used by accounting schemes to report back the balance of an account and to notify a third party peer about a new session. Finally, the *accounting scheme interface* needs to be implemented by every individual accounting scheme. This interface offers the following methods: The *configureScheme* method is used to initialize the scheme with vital information like the peer ID of the local peer and further scheme-specific parameters. The methods *createPeerAccount* and *createSessionAccount* are used to create peer or session accounts, respectively. Finally, *notifyBalanceUpdate* and *queryAccount* are used to update or query accounts which are hold by a particular scheme.

All communication between instances of the A&C system on different peers is encapsulated within the specific accounting schemes deployed. Hence, accounting schemes are using their own accounting protocol to setup connections and talk to other peers. Figure 1 gives an overview on the A&C system and its interactions with other middleware components and different accounting schemes.

When a new application is started, an application-specific configuration policy is handed over to the A&C system (1). The AC module uses the policy to instantiate the respective accounting schemes and creates a peer account for the local peer using the *createPeer-Account* method. Depending on the nature of the scheme, the account is either allocated

locally, on a central server or a set of remote peers. Note that peer accounts may be used in several applications if they use the same scheme.

Whenever a new session is notified by a service (2), the SM module creates a new accounting session and keeps the provided SLA which includes the tariff. In addition, the SLA is forwarded to the AC module to create a new session account. Upon each accounting event sent by the service (3), the SM module selects the corresponding tariff, which evaluates the service-specific information described in the event and generates zero or more balance updates that will be forwarded to the AC module. Based on the configuration policy, the balance updates are sent to the appropriate accounting schemes using the *notifyBalanceUpdate* method. Depending on the settlement scheme, accounts may report back their balance to the A&C core (4) using the *notifyAccountBalance* method. Again, the SM module selects the corresponding tariff which may generate further balance updates that either affect session or peer accounts. As mentioned above, accounts may also be hold on third-party peers. To enable a remote account to notify its balance, an accounting session needs to exist on the remote A&C core containing the appropriate tariff. For this purpose the accounting scheme can create a new session on the remote peer using the A&C core interface (5).

Finally, any external component can query an account using the *queryAccount* method (6). The AC module forwards the query to the accounting scheme which holds the respective account. The scheme uses a given listener interface to reply the current account balance.

4 Conclusions

The A&C system presented follows a modular approach, enabling the flexible use of different accounting schemes for a variety of P2P applications. Through the use of tariffs, the system is able to support various settlement schemes and can handle service-specific events in a generic fashion. The A&C system has been implemented and was successfully used with different accounting schemes in a set of applications, *e.g.*, a file-sharing application and a collaborative radiology diagnosis application. Further accounting schemes are envisaged to be implemented in future, increasing the variety of accounting mechanisms supported.

References

[Coh03] B. Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003.

[GS05] J. Gerke and B. Stiller. A Service-Oriented Peer-to-Peer Middleware. In *14. Fachtagung Kommunikation in Verteilten Systemen 2005 (KiVS 05)*, To appear, February 2005.

[Hau04] D. Hausheer. Accounting Patterns, Deliverable D10.1 of the MMAPPS Project. Technical report, March 2004.