

الوحدة الأولى

أساسيات برمجة الشبكات

Basics of Network Programming

• . مقدمة

• 1.1 تمهيد

- تبحث هذه الوحدة في الوسائل الأساسية التي يتم من خلالها تبادل المعلومات عبر الشبكة. ويتمحور هذه البحث حول دراسة الكيفية التي يتم من خلالها تبادل البيانات بين عقدتي اتصال بشكل عام وبين حاسوبين بشكل خاص.
- يحدد البروتوكول تعريفاً الأسلوب الذي يتم به التخاطب بين عقدتين، وتتطلب برمجة هذا الأسلوب بشكل عام الاستفادة من مفهوم التقسيم إلى طبقات الذي سمح بهيكله وتقيس برمجة الشبكات، وفتح الباب واسعاً أمام الدارسين والمصممين ليطوروا ويعدلوا الخدمات المقدمة

الأهداف

- يتوقع من الدارس بعد الانتهاء من دراسة هذه الوحدة أن يكون قادرا على أن:
 1. يناقش الوسائل الأساسية التي يتم من خلالها تبادل المعلومات عبر الشبكة.
 2. يميز بين البروتوكولات المختلفة الموزعة على الطبقات السبعة لنظام الشبكات المفتوح.
 3. يناقش آليات العنونة المستخدمة في مختلف الطبقات.
 4. يناقش المبادئ الأساسية لبرمجة الخادم /الزبون.
 5. يناقش مفهوم البرمجة الموزعة

2. بروتوكولات طبقات الشبكات

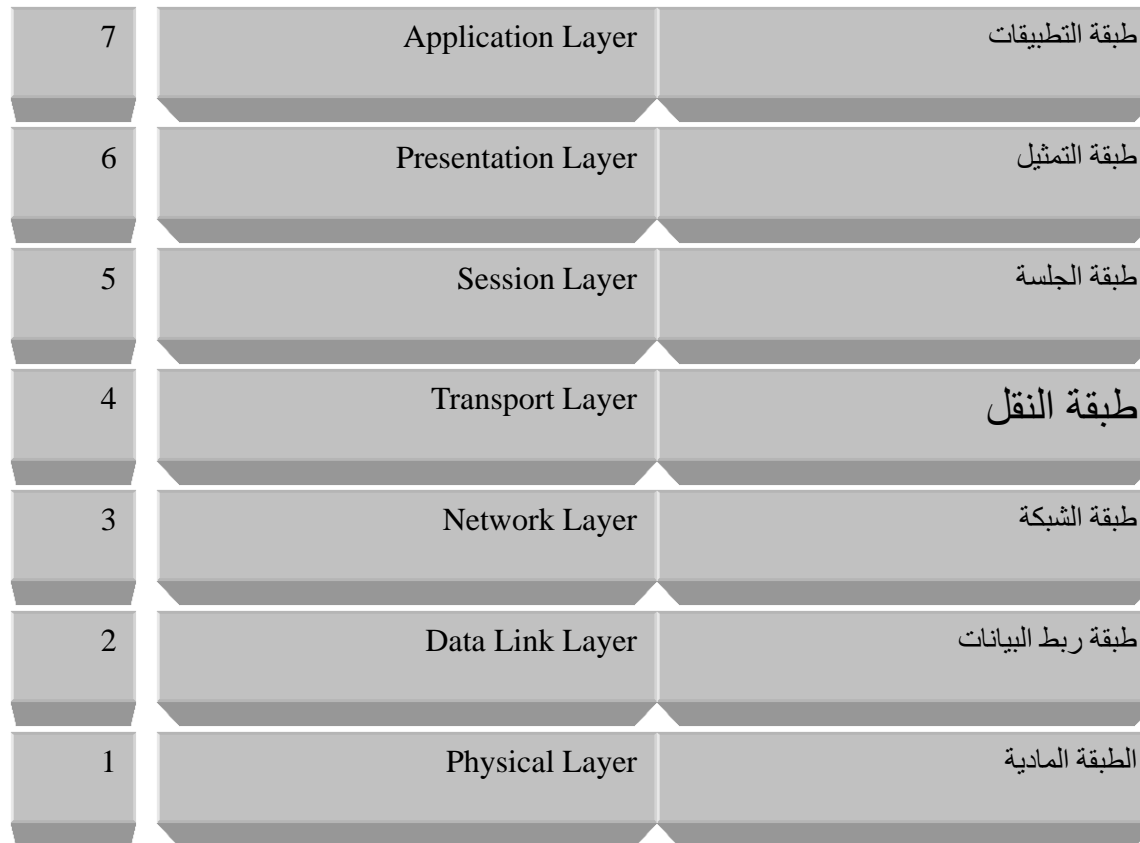
1.2 مفهوم الطبقات

- ولد مفهوم تقسم فعاليات الشبكة إلى طبقات من رحم هندسة النظم Software Engineering والبرمجة الهيكلية Structured Programming التي سعت إلى تشكيل وحدات وظيفية وبرمجية قابلة لإعادة التشكيل والتطوير والتعديل.
- وقد تبنت فلسفة بناء الكينونات Objects والتوجيه بالكينونات Object Orientation التقسيم إلى طبقات لتضفي أبعاداً جديدة على برمجيات الشبكات وتسمح بإطلاق العنان لآفاق جديدة من سرعة التطوير ربما لم تكن ممكنةً لولا ذلك.

- نستطيع تقديم مفهوم تقسيم فعاليات الشبكة إلى طبقات Layering على أنه:

1. تحقيق المهمات الموكلة للشبكة من خلال مجموعة خدمات.
2. تصنيف الخدمات وفقاً للوظائف التي تؤديها وتحديد مستويات للخدمة أو طبقات.
3. إنشاء معمارية تسمح لكل طبقة بالاستفادة من خدمات الطبقات التي دونها وتقديم الخدمات إلى الطبقات التي فوقها
4. معالجة كل طبقة أو خدمة بشكل مستقل تقريباً
5. تعريف الحدود بين الطبقات من خلال واجهات تسمح بتبادل البيانات بين الطبقات عبر منافذ Ports.

نظام الوصل المفتوح Open System Interconnect OSI



الشكل (1): طبقات نظام الوصل المفتوح

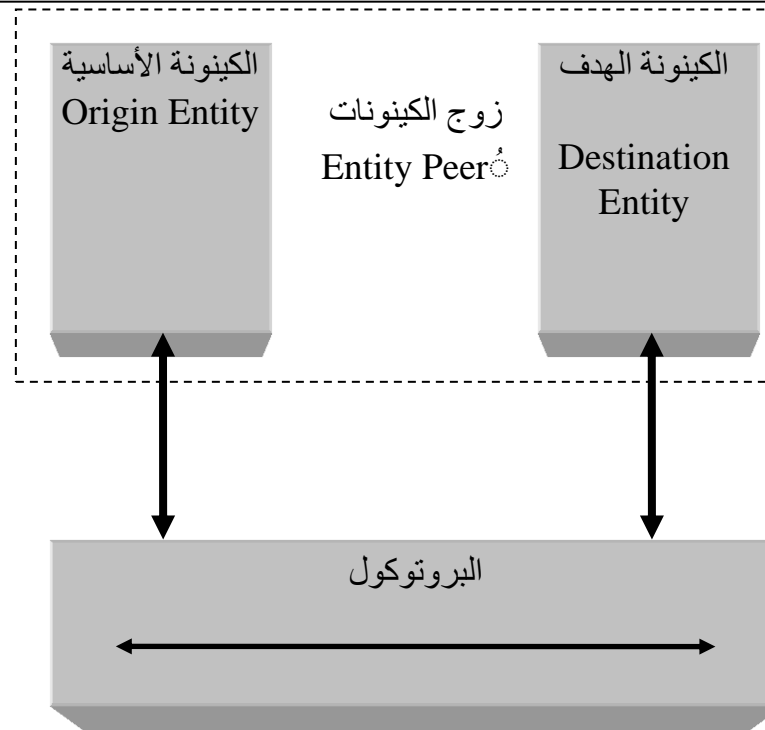
تحتوي كل طبقة من طبقات نظام الوصل الداخلي المفتوح OSI على:

1. بيانات خاصة بها

2. مجموعة من الكينونات Entities أو العمليات الفعالة
Processes تتوزع على:

- عمليات برمجية كالتوابع والإجراءات.
- عمليات إلكترونية تنجز من خلال العتاد الصلب كرقاقات الإدخال الإخراج الذكية

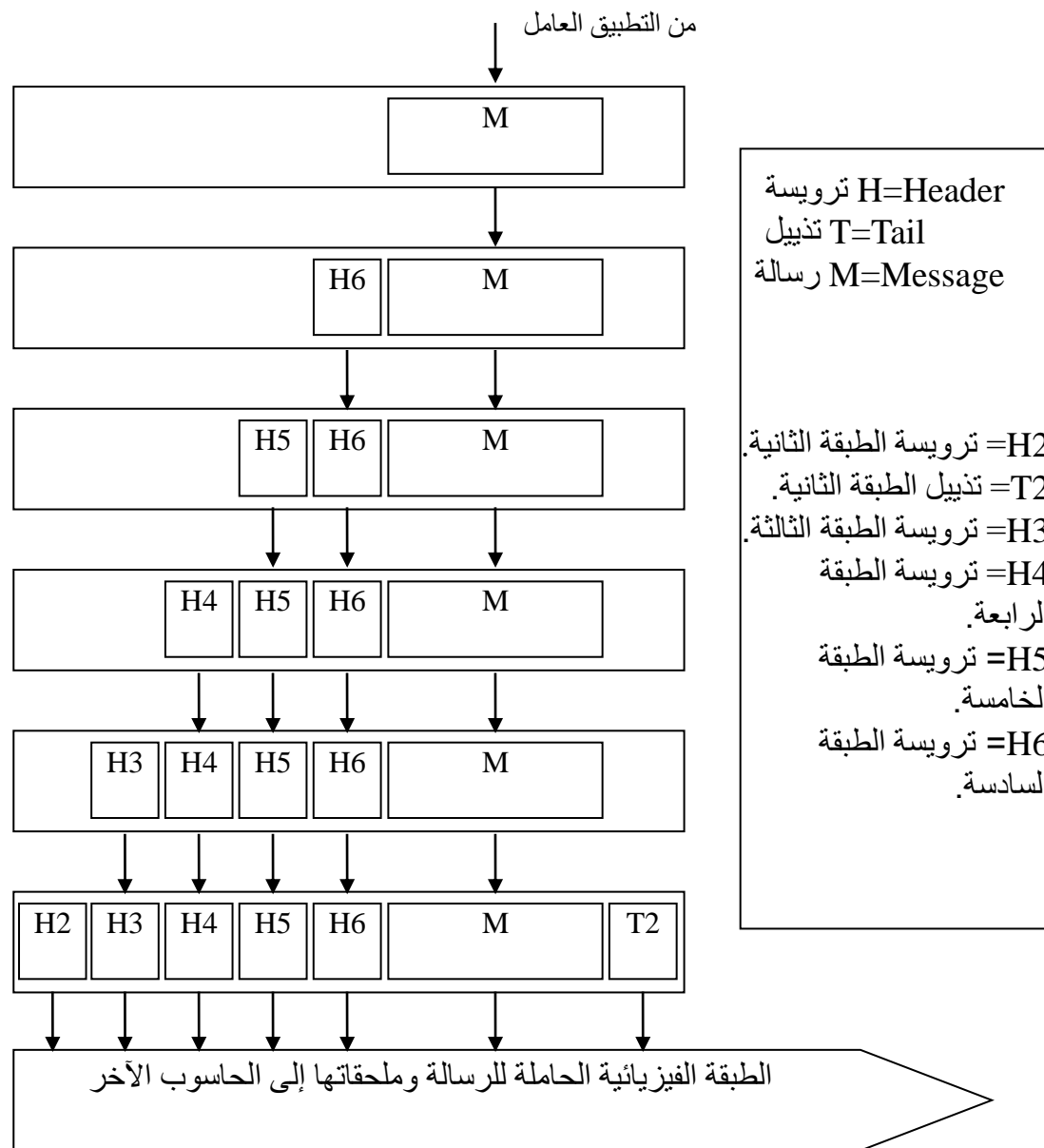
وسنتعامل مع مصطلحي الكينونات والعمليات على أنهما
يشيران إلى نفس المدلول رغم أن بعض المراجع تحصر
العمليات بالفعاليات البرمجية فقط.



الشكل (2): الكيونات أو العمليات عند الاتصال بين العقد

- ويطلق على الكينونات في الطبقات المتماثلة في جهازين
يتبادلان البيانات بزواج الكينونات أو العمليات Entity or
Process Pair.

- أما فيما يتعلق بالبيانات، فيجري توزيعها على رسالة
Message ومجموعة من التذييلات والترويسات
Headers & Tails كما هو موضح في الشكل (3).



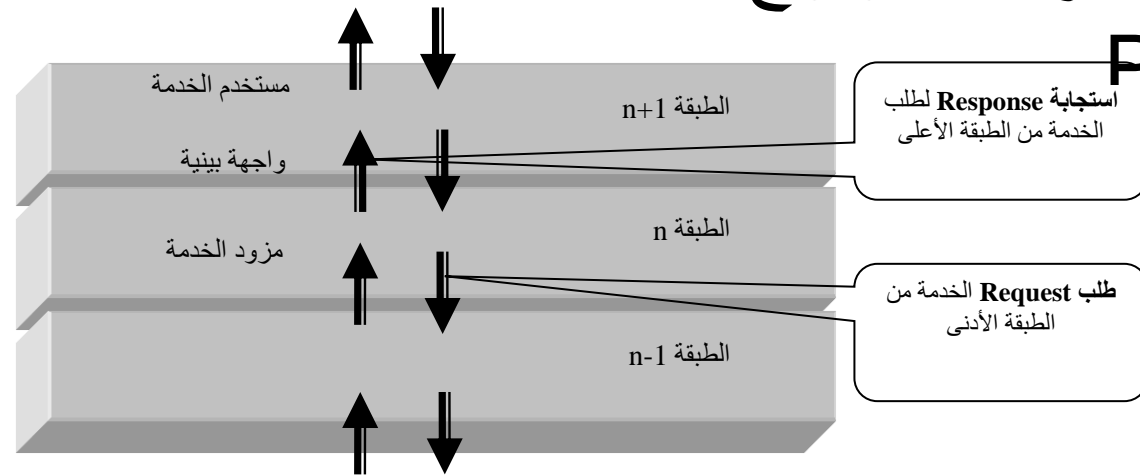
الشكل (3): تشكل الرسالة في طور الإرسال في طبقات OSI

• 3. الخدمات Services

- يرتبط مفهوم الخدمة Service والبروتوكول Protocol ارتباطاً وثيقاً ببعضهما البعض من ناحية وبنظام ومحددات الطبقات من ناحية أخرى.

• الواجهة البينية Interface:

- هي واجهة اتصال طبقتين متجاورتين، و من خلالها يحدث تبادل البيانات عبر نقاط ولوج الخدمة Service Access

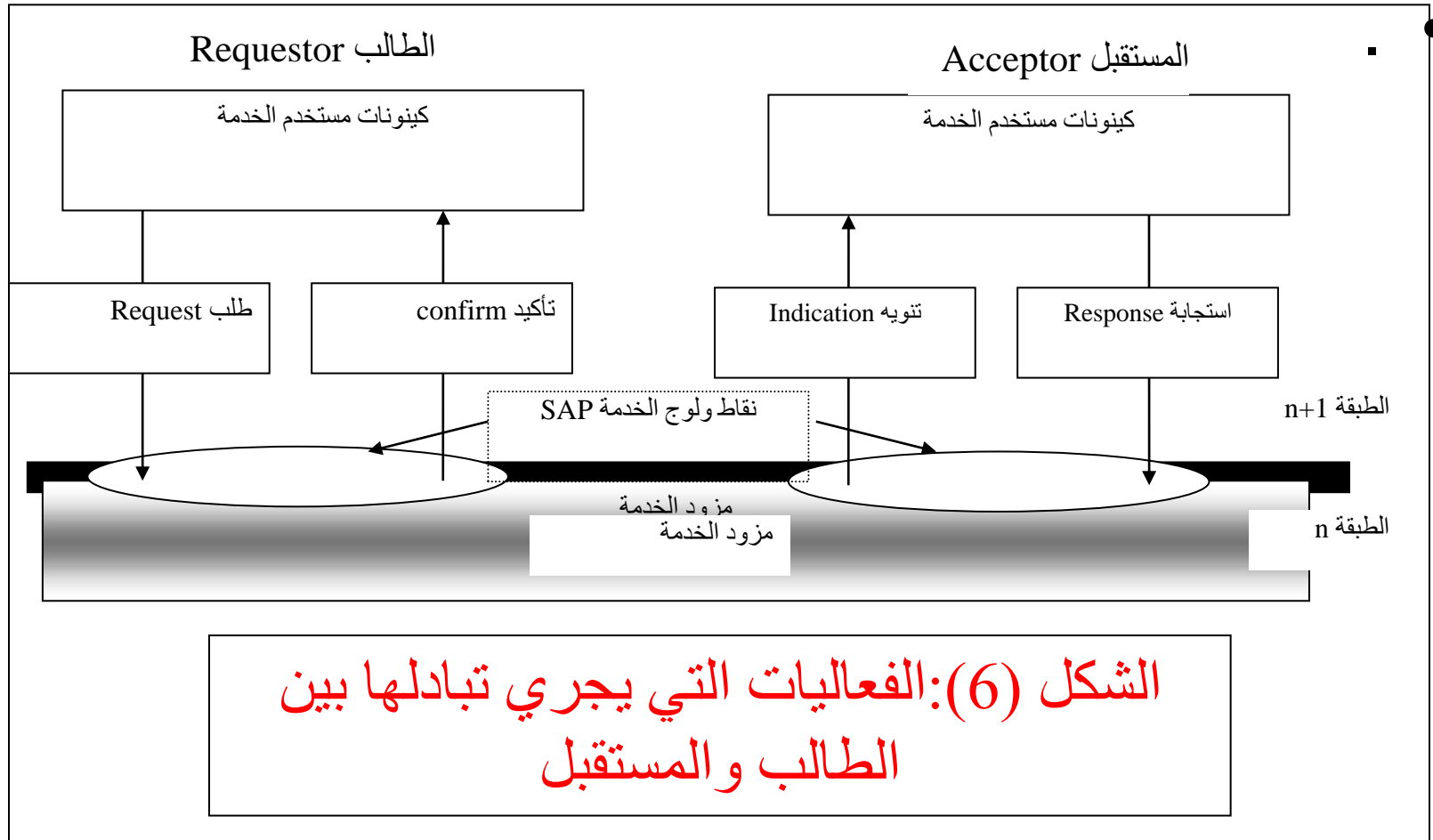


Points SAP

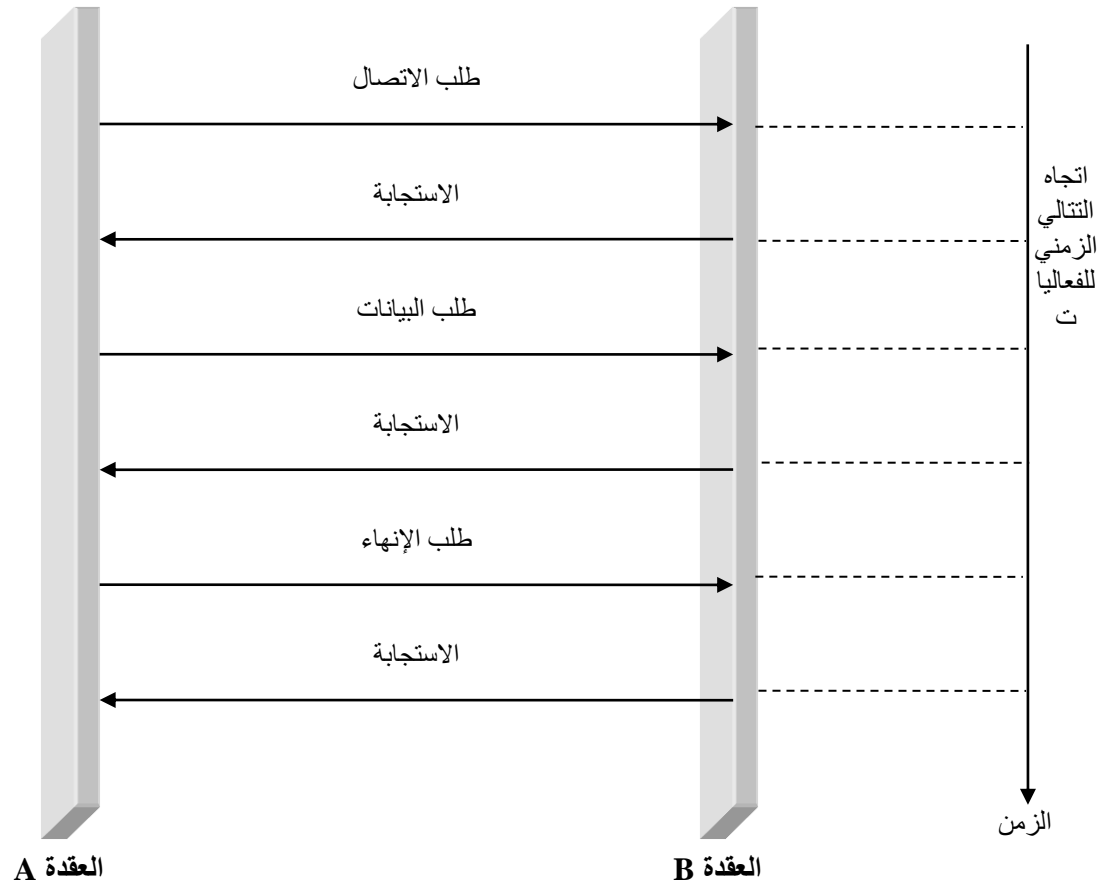
الشكل (5): مفهوم الطلب والاستجابة في البروتوكولات المقسمة إلى طبقات
برمجة شبكات

البروتوكول Protocol

هو عبارة عن مجموعة القواعد والقوانين التي تحكم تشكيل ومدلولات الرسالة أو وحدة المعلومات التي يتم تبادلها مع الكينونات الأخرى ضمن نفس الطبقة إنما عقدة أخرى



- مثال تبادل البيانات في نظام الاتصال الموجه:
- عندما تتبادل عقدتين A و B البيانات في نظام اتصال موجه Connection Oriented system لا بد من توفر الخدمات التالية:
- خدمة إنشاء الاتصال Connection Establishment.
- خدمة نقل البيانات Data Transfer.
- خدمة إنهاء الاتصال Connection Release.
- يتم تحقيق هذه الخدمات كما يبين الشكل (11) من خلال تكرار استخدام نفس المحددات كما يلي:



الشكل (11): اتصال موجه بين عقدتين

• 4. هياكل البيانات المستخدمة في البروتوكولات

Data Structure used in Protocols

- هياكل البيانات هي الطريقة التي تنظم بها البيانات آخذة بعين الاعتبار العناصر المخزنة وعلاقة هذه العناصر مع بعضها البعض. وسنستعرض فيما يلي المصطلحات التي اتفق عليها مصمموا البروتوكولات لوحداث البيانات المستخدمة في البروتوكولات المختلفة في كدسة بروتوكولات نظام OSI.

• 1.4 **بناء هياكل البيانات المستخدمة من الأنماط القياسية**

Standard Data Structure

- تبني هياكل البيانات اعتماداً على الأنماط الأساسية للبيانات التي تحددها لغة البرمجة المستخدمة والتي بدورها تشكل حجر الأساس لهياكل البيانات المركبة والأكثر تعقيداً.

• 2.4 الرسائل Messages

- تبني الرسائل Messages من هياكل البيانات البسيطة والمركبة.

• 3.4 عناصر وعمليات البروتوكول الأساسية

- تختلف الرسائل المتبادلة بين البروتوكولات باختلاف وحدات بيانات البروتوكول Protocol Data Units PDU التي تتكون كما أسلفنا من البيانات إضافةً إلى الترويسات والتذييل كما وضح ذلك لنا الشكل (1).
- تعرف وحدة بيانات البروتوكول PDU على أنها مجموعة البيانات التي يتعامل معها البروتوكول.

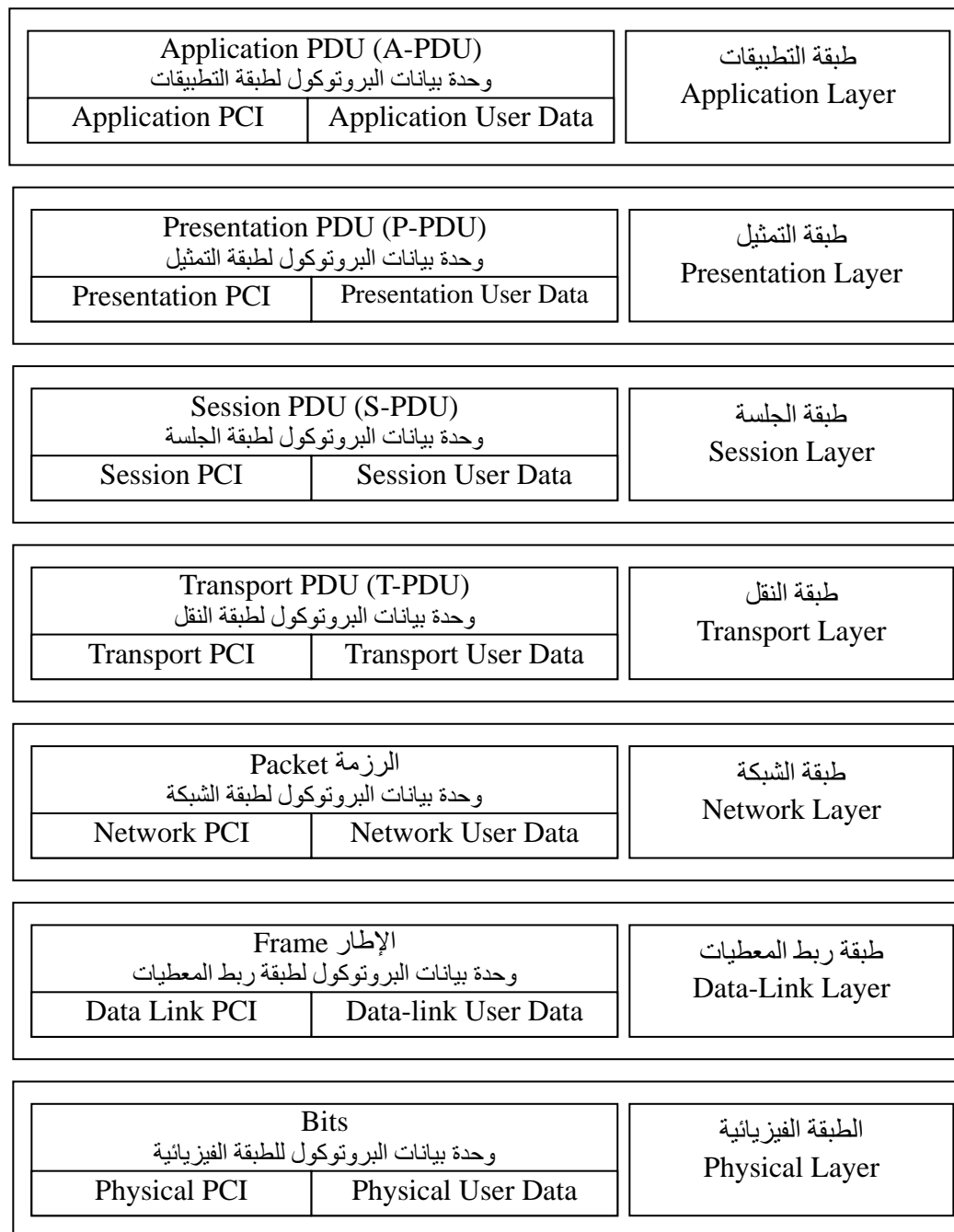
وبناءً عليه قسمت وحدة بيانات البروتوكول PDU إلى:

1. معلومات البروتوكول التحكمية Protocol Control
Information PCI

2. بيانات المستخدم User Data

ويتم التحكم بتدفق المعلومات بمجموعة من الفعاليات التي تعرف البروتوكول وتعرف هذه الفعاليات ضمن ما يسمى

معلومات التحكم بالبروتوكول Protocol Control
Information PCI

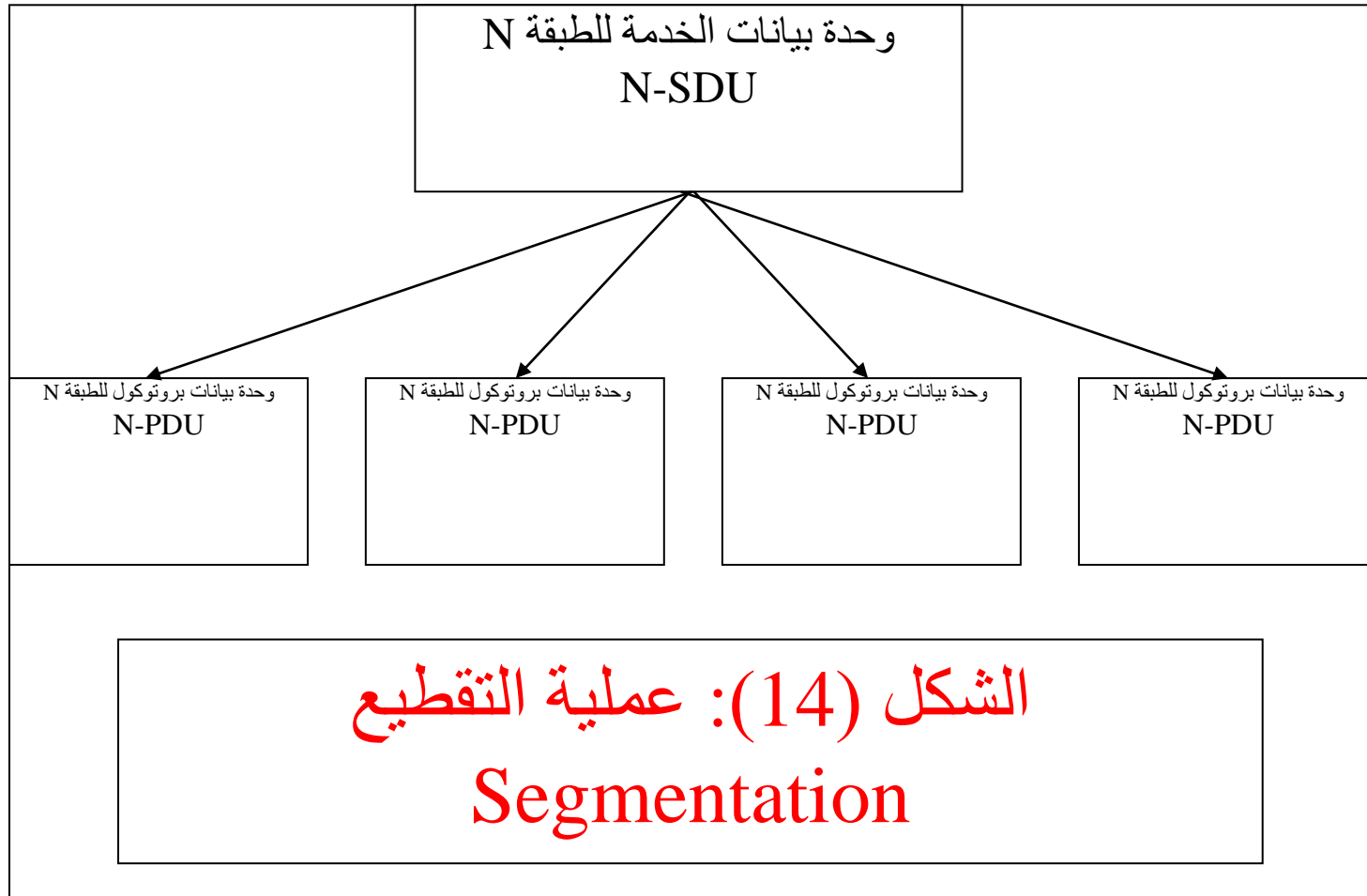


الشكل (13): وحدات البيانات في البروتوكول OSI

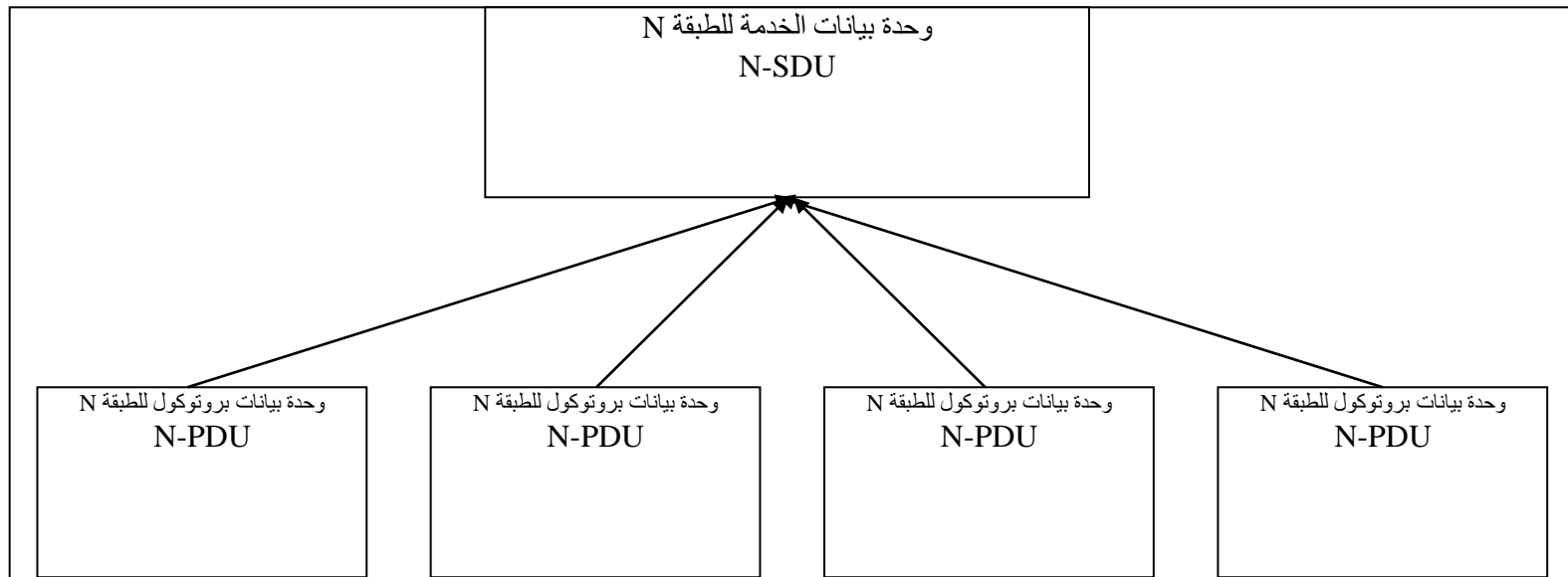
• 4.4 الفعاليات الأساسية للبروتوكول

- تخضع وحدات بيانات البروتوكول n-PDU ووحدات بيانات الخدمة n-SDU لمجموعة من العمليات التي تمكن البروتوكول من التغلب على مشكلة تفاوت طول الرسائل المتبادلة بين العقد وبالتالي الانتظار الطويل عند ورود رسائل طويلة مما يسمح بتحسين أداء تبادل البيانات عند تعدد الرسائل من خلال ست عمليات يمكننا تلخيصها بما يلي:

1. التقطيع Segmentation الشكل (14): وهو عملية فصل وحدات بيانات الخدمة في طبقة ما N (N-SDU) إلى مجموعة من وحدات بيانات البروتوكول المخصصة للطبقة N (N-PDU).

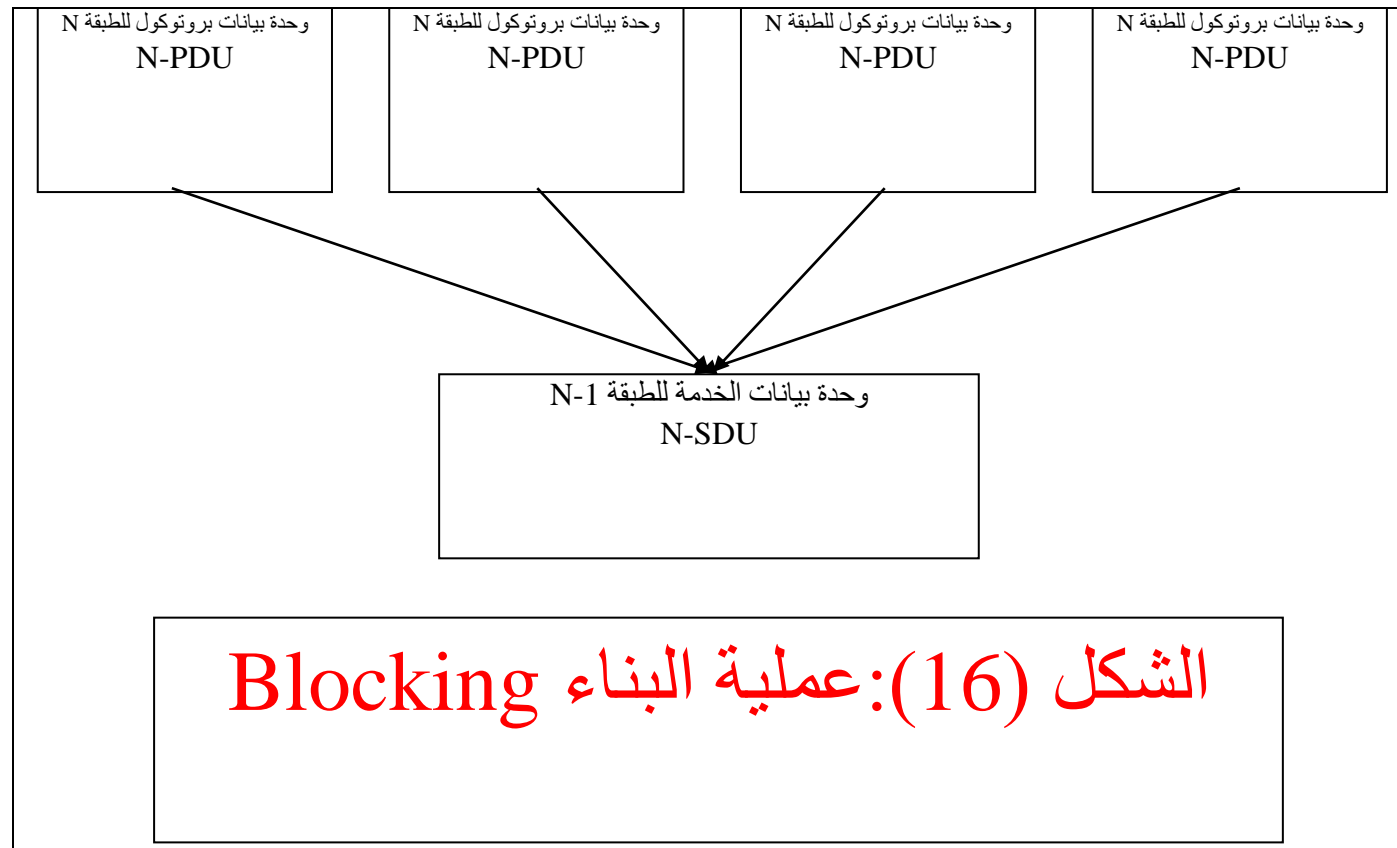


2. إعادة التجميع Reassembly الشكل (15): وهي عملية إعادة تركيب وحدات بيانات البروتوكول المختلفة للطبقة N (N-PDU) التي تمت تجزئتها في مرحلة التقطيع إلى وحدة بيانات خدمة (N-SDU) من جديد. وعملية إعادة التجميع هي العملية المعاكسة للتقطيع.

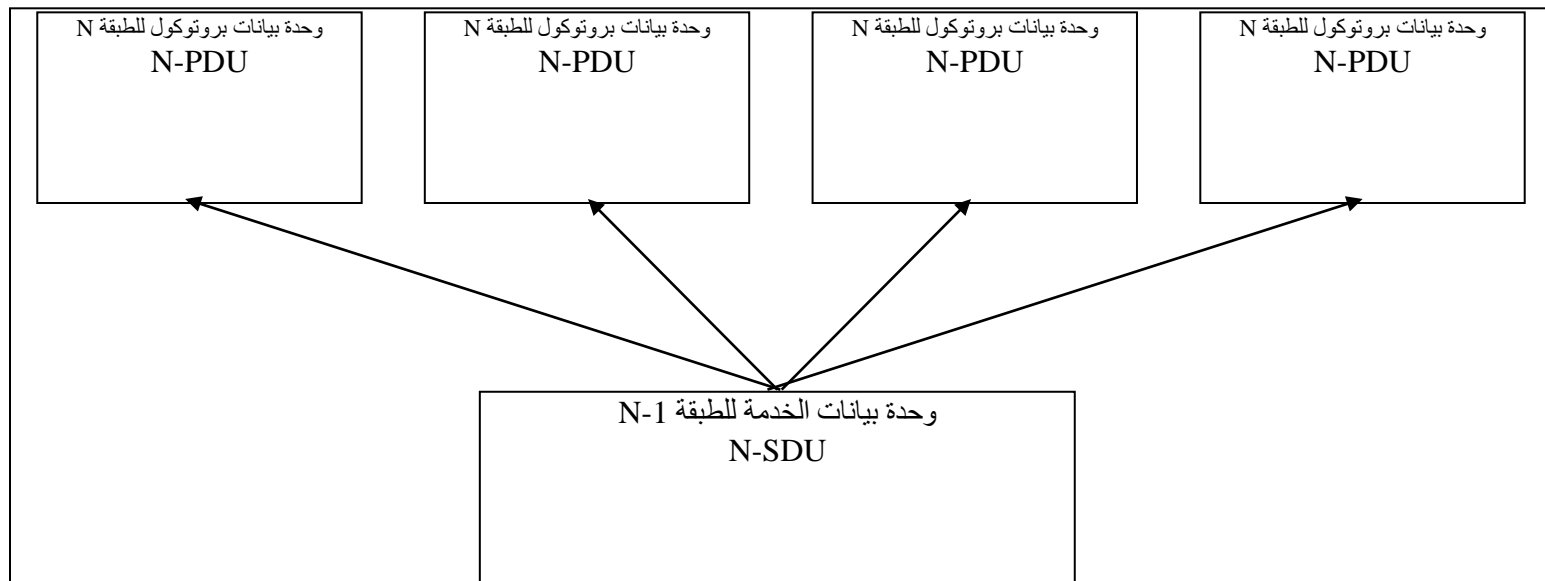


الشكل (15): عملية إعادة التجميع Reassembly

التكتيل Blocking: الشكل (16) التي تسمح بإنشاء عدة وحدات بيانات خدمة SDUs والتي قد تأتي من خدمات مختلفة في وحدة بيانات بروتوكول كبيرة ضمن نفس الطبقة التي نشأت منها وحدات بيانات الخدمة.

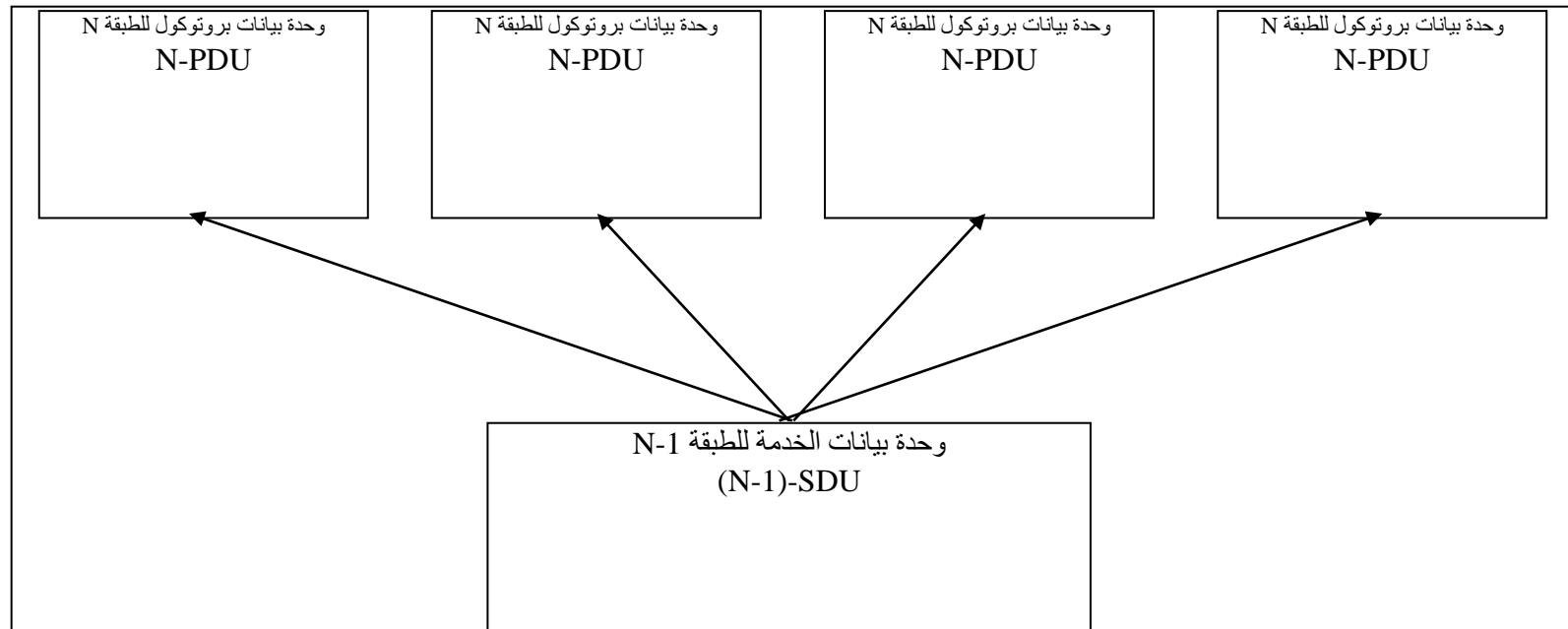


التفكيك Unblocking الشكل (17): هي عملية تفكيك وحدة بيانات البروتوكول PDU التي تم إنشاؤها في عملية البناء إلى وحدات بيانات الخدمة في نفس الطبقة.



الشكل (17): عملية التفكيك Unblocking

الفصل Separation الشكل (19): وهي العملية المعاكسة لعملية الوصل وفيها يتم فصل وحدة بيانات الخدمة إلى عدة وحدات بيانات بروتوكول وتقديمها للطبقة التالية. أي أن عملية الفصل مشابهة لعملية التفكيك وإنما للوحدات بين الطبقات.



الشكل (19): عملية الفصل Separation

• 5. برمجة الخادم /الزبون Client/Server Programming

• 1.5 أنماط هيكليات الخادم/الزبون

• تلخيص أنماط هيكليات الخادم/الزبون في ثلاثة أنماط أساسية هي:

• 1. المعمارية ثنائية المراحل Two-Tier Architecture:

- تنشأ المعمارية ثنائية المراحل عن ارتباط ثنائي بين الخادم والزبون وهي حالياً أكثر هيكليات الخادم/الزبون انتشاراً فمعظم تطبيقات نظم قواعد البيانات الاستعلامية والبريد الإلكتروني
- وتتميز هذه البنية بكلفتها المنخفضة وبأنها تقنياً بسيطة التركيب.

2.المعمارية الهجينة أو ذات المرحلتين والنصف Hyprid :or "2 1/2-tier Architecture"

- والمبينة بالشكل (21-ب) وفيها يجري استخدام الموجهات Routers والمجمعات Hubs وغيرها من متحكمات الشبكات الذكية لكي تستعويض بها عن الخادم الوسيط وتعتبر هذه المعمارية متوسطة التعقيد والكلفة.

3.المعمارية ثلاثية المراحل Three-Tier :Architecture

- المبينة بالشكل (21-ج) التي تقسم الموارد الفيزيائية والمنطقية إلى خادمت مركزية وخادمت موزعة ومحطات عمل زبونة. وتتميز بتعقيدها التقني وبكلفتها العالية.

• 2.5 الطرق المستخدمة في برمجة الخادم /الزبون

عندما نتكلم عن البرمجة لا بد لنا أن نستعرض تقنيات البرامج وخصائصها من خلال تصنيف يأخذ بعين الاعتبار طريقة تعاملها مع العمليات والمهام التي تشكل الوحدة الأساسية في البناء البرمجي لتطبيقات الخادم/الزبون والبرمجة الموزعة وأهم هذه البرامج هي:

1. البرنامج التسلسلي Sequential Program

سلسلة متتابعة من الفعاليات (التعليمات والمتغيرات) المولدة لنتيجة، نطلق عليها العملية Process أو المهمة Task أو المسلك Thread.

2. البرنامج التعاوني Concurrent Program

عمليتان Processes أو أكثر تتصلان مع بعضهما البعض لإنجاز عملاً مشتركاً من خلال متغيرات تشاركية Shared Variables وتؤديان (تؤدي) عملهما من خلال تمرير الرسائل وتحتاجان من أجل ذلك إلى مزامنة Synchronization لفعاليتهما.

1. التطبيقات متعددة المسالك Multithreaded Application

- تنفيذ الفعاليات من خلال استخدام أكثر من مسلك Thread وغالباً ما يتم ذلك من خلال التشارك بزمان المعالجة التابع لوحدة معالجة مركزية واحدة.
- تعتبر التطبيقات متعددة المسالك وسيلة جيدة لتنظيم التطبيقات البرمجية الحديثة من خلال الخادومات Servers ونظم التقسيم إلى شرائح زمنية Time Sharing systems.

2. التطبيقات المتوازية Parallel Applications

- يختص كل معالج في التطبيقات المتوازية بتنفيذ عمليات محددة منوطة به من أجل حل مشاكل ومسائل برمجية كبيرة الحجم بسرعة أكبر.

3. التطبيقات الموزعة Distributed Applications

- تتصل العمليات مع بعضها البعض في التطبيقات الموزعة عبر شبكة مما يسمح باستثمار معلومات ومقدرات الحواسيب ذات القدرات الكبيرة مثل الخادومات الكبيرة والمتطورة من خلال

6. البرمجة الموزعة Distributed Programming

- تعرف منظمة المقاييس العالمية ISO المعالجة الموزعة أو الحساب الموزع على أنها فعاليات معالجة المعلومات حين تكون المكونات المنفصلة للنظام موجودة في أكثر من موقع أو عندما يكون هنالك ضرورة لاتصال مكونات مختلفة لتشكل منظومة أيّا كان السبب الداعي لذلك.

• 6 البرمجة الموزعة في الشبكات

- يميز النظام الموزع باعتباره مجموعة العمليات Process المرتبطة مع بعضها البعض بواسطة شبكة عن بقية الأنظمة الحاسوبية ما يلي:
- ليس هنالك معلومات عامة ولا توقيت عام.
- لا يمكن التنبؤ بالتأخير الزمني للاتصال بعملية ما.
- تعاونية العمليات Concurrency وعدم التحديدية Non-determinism فيها.
- احتمالية عالية للأخطاء المحلية.
- دخول سهل للمستخدمين غير المرخص لهم.
- تحتاج البرمجة الموزعة إلى بنية تحتية من نوع خاص، وتعتبر الإنترنت خطوة أولى نحو تطبيقات البرمجة الموزعة العامة بسبب:
- مجال الأسماء العام الذي تستخدمه Global Namespace.
- بروتوكول الاتصال الواسع الانتشار TCP/IP.
- وقد استفادت منصات برمجية هامة مثل جافا JAVA من البنية التحتية للنظم الموزعة، إلا أن كتابة تطبيقات البرمجيات الموزعة لا تزال حتى الآن أكثر صعوبة بكثير من كتابة تطبيقات البرمجيات المركزية Centralized Application.

- يزيد ضمان أمن المعلومات في البرمجيات الموزعة من تعقيدها إلى درجة كبيرة.

•

- أمثلة من الواقع العملي على البرمجيات الموزعة:

- الإجراءات المستدعاة عن بعد RPC Remote Procedure Call.

- لغة جافا Java وفروعها المختلفة:

- جافا الأصلية Original and Pure Java.

- استدعاء بالدخول عن بعد RMI Remote Access

Invocation: التي تعتبر منظومة قوية لتوزيع ومعالجة الكينونات بشكل مستقل عن المنصة البرمجية المستخدمة إلا أنها لا تعمل إلا مع الكينونات المنشأة بلغة جافا.

2.6 خصائص النظم الموزعة

تتسم النظم الموزعة ببعض الخصائص الأساسية المميزة لها والتي يمكننا تلخيصها بما يلي:

1. تعدد الحواسيب (العقد) Multiple Computers (Nodes)

2. التشارك بالموارد Resource Sharing

3. التعاونية Concurrency

4. تمرير الرسائل Message Passing